Compter les globules blancs, analyser les partitions

Mathieu Giraud

École Doctorale Sciences pour l'Ingénieur (SPI)

Centre de Recherche en Informatique, Signal et Automatique de Lille
CRIStAL, UMR 9189 CNRS, Université de Lille

Centre de Recherche Inria Lille

Compter les globules blancs, analyser les partitions

Habilitation à diriger des recherches

présentée par

Mathieu Giraud

---

soutenue publiquement le 30 mars 2016 devant le jury composé de

Gérard Assayag, rapporteur	Directeur de l'UMR STMS 9912, Ircam, Paris
Pierre Boulet	Professeur, Université Lille 1
Frédéric Davi	Professeur PUPH, Hôpital Pitié-Salpêtrière, Paris
Alain Denise, rapporteur	Professeur, Université Paris-Sud
Thierry Lecroq, rapporteur	Professeur, Université de Rouen
François Pachet	Directeur du Sony CSL Labs, Paris
Hélène Touzet, garante	Directrice de recherche, CNRS, CRIStAL, Lille

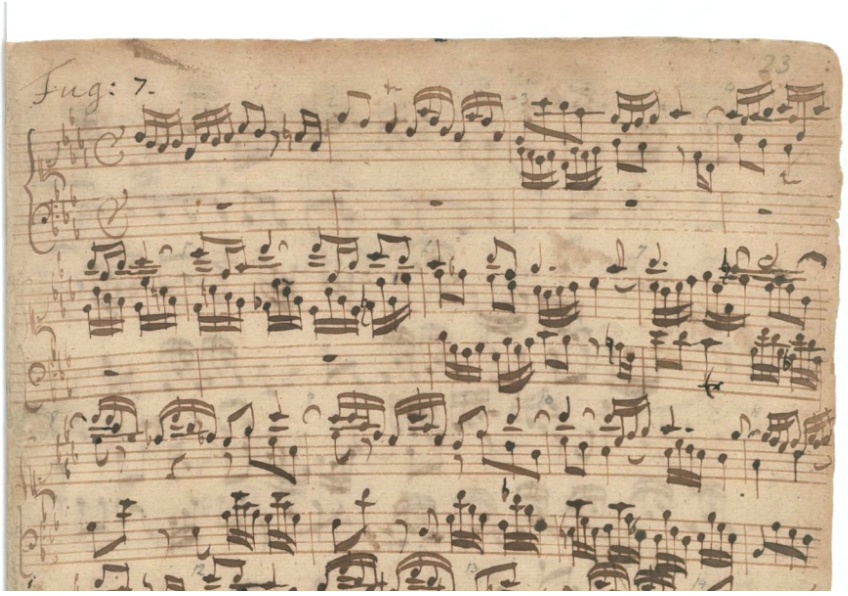


FIGURE 0.1 – *Début du manuscrit de la fugue en Mi♭ majeur BWV 852 de Jean-Sébastien Bach (1722).*

Un énorme merci à tous mes collègues des équipes Bonsai et Algomus, ainsi qu'à toutes les personnes citées ou non dans ce document qui m'ont accompagné durant ces quinze années de recherche. Merci en particulier à ceux qui ont commenté des parties de ce mémoire, en tout premier lieu Marion et Mikaël, ainsi que Florence, Maude, Tatiana, Valérie et Yann.

à Marion, Mariette, Marcellin, Marguerite et Maximine,

Lille, le 29 janvier 2016

1 Prélude **5**

I Compter les globules blancs, Vidjil 9

2 Immunologie et oncologie **11**

2.1 Hématopoïèse et recombinaison V(D)J 12

2.2 Diagnostic et suivi des leucémies aiguës lymphoblastiques 13

2.3 Séquençage à haut-débit de répertoire immunologique (Rep-Seq) 16

3 Algorithmes pour l'analyse des recombinaisons V(D)J **19**

3.1 Défis de l'immunoinformatique et des études Rep-Seq 20

3.2 Méthodes classiques et haut-débit 20

3.3 Regrouper, puis compter et analyser : une nouvelle approche pour l'analyse haut-débit 22

3.4 Analyse multi-locus et recombinaisons incomplètes 25

4 Développement, évaluation et mise en production de Vidjil **27**

4.1 L'analyse haut-débit 28

4.2 L'application web 30

4.3 Le serveur et la base de données d'échantillons et de patients 31

4.4 Développement, intégration continue, mise en production 31

5 Vidjil : usages, analyse de données et résultats **33**

5.1 Utilisations du serveur de test `app.vidjil.org` 34

5.2 Diagnostic des leucémies aiguës lymphoblastiques (Lille) 35

5.3 Suivi des leucémies aiguës lymphoblastiques (Lille) 36

5.4 Estimation de la diversité du répertoire (Prague) 38

6 Perspectives **39**

6.1 Algorithmique des recombinaisons V(D)J 39

6.2 Développement, diffusion et transfert 40

II Analyser les partitions, Algomus **43**

7 Analyse musicale, analyse musicale computationnelle **45**

7.1 Pourquoi analyser des partitions musicales ? 46

7.2 Analyse et synthèse 48

7.3 L'analyse musicale computationnelle 48

7.4 Algomus 52

8 Algorithmes d'analyse musicale **55**

8.1 Analyse locale, analyse globale ? 56

8.2 Analyse locale 56

8.2.1 Analyse locale : motifs 56

8.2.2 Analyse locale : strates polyphoniques, texture 58

8.2.3 Analyse locale : harmonie et cadences 61

8.3 Analyse globale : vers l'analyse de formes musicales 63

9 Développement, visualisation et évaluation **69**

9.1 Quelques logiciels existants 70

9.2 Modélisation, développement et visualisation 70

9.3 Évaluation et analyses de références 73

10 Perspectives **77**

10.1 Algorithmique musicale 77

10.2 Développement et objectifs sociétaux 79

Coda **81**

11 Médiation scientifique et artistique **83**

11.1 Ateliers à la rencontre du public 84

11.2 Projets arts et science 87

12 Bilan **89**

13 Bibliographie **91**

13.1 Publications auxquelles j'ai contribué 91

13.2 Références 92

14 Figures et tables **99**

1 Prélude

Séquences de caractères. Qui a affirmé que *« Rien n'a de sens en biologie, si ce n'est à la lumière de l'évolution »* ? Tous les jours ou presque, nous demandons à notre mémoire, à notre dictionnaire ou, accessoirement, à un moteur de recherche de trouver des informations correspondant à une *requête*. Le plus souvent, on cherche parmi un ensemble de documents connus, ou *indexés*, ceux qui vont contenir cette requête. Dans le cas le plus général, la requête tout comme les documents sont des *séquences de caractères*, c'est-à-dire des suites de symboles pris dans un *alphabet*. Alphabets et séquences ne sont pas limités aux textes de nos langues usuelles :

Alphabet	Exemples de séquences (appelées encore *mots*)

{`a`, `b`}	mot vide ε
	`aaa`
	`abababaabbba`

26 lettres de l'alphabet latin	`maison`
	`ortografe`
	`kgjsnipachjoehqarwr`

caractères Unicode	`Le petit chat est mort.`
	`῾Εν ἀρχῇ ἦν ὁ λόγος`
	`MI8675drm-IB A,Bg!5061sekwaF-xQJ àZinJ+b`
	la bibliothèque d'Alexandrie

chiffres de 0 à 9	0328778554 la suite infinie des décimales de π
nucléotides de l'ADN {`A`, `C`, `G`, `T`}	`CAGCATCACGACTACGACATCAGTAT` le génome humain (3 milliards de nucléotides)
notes de musique {*Do, Ré, Mi, Fa, Sol, La, Si*}	*Sol Do Si Ré Do Sol Mi La Fa* l'intégrale des œuvres de J.-S. Bach

Distance d'édition entre séquences de caractères. Certaines séquences de caractères sont *similaires*, sans être égales. Un correcteur orthographique intégré à un traitement de texte doit identifier des mots « proches » de mots tels que `ortografe`. Lorsqu'on utilise un moteur de recherche, on souhaite trouver des documents pertinents, même s'ils ne correspondent pas exactement à la requête. Ces approximations ne se limitent pas aux textes – qui n'a jamais voulu retrouver une chanson ou une pièce de musique à partir d'une mélodie vaguement fredonnée ? Comment formaliser le fait que certaines séquences soient proches – ou non – d'autres séquences ? En 1965, alors qu'il travaillait sur les *codes correcteurs d'erreur*, Levenshtein propose ce qu'on appelle aujourd'hui la *distance de Levenshtein* [36] :

> *« We will say that a code K can correct s deletions, insertions and reversals if any binary word can be obtained from no more than one word in K by s or fewer deletions, insertions, or reversals. It can be shown that the function $r(x, y)$ defined on pairs of binary words as equal to the smallest number of deletions, insertions, and reversals that will transform x into y is a metric, and that a code K can correct s deletions,*

```
a) aaaaabb       aaaaabb       ababab
    x x  x 3     xx   xx 4     x  x x 3
   ababab        bbaaa         bbaaa

b) aaaaabb
      x    1
   aaababb
    x    x 2
   ababab
```

FIGURE 1.1 – *Distance d'édition et code correcteur d'erreurs [36]. a) On considère un ensemble de mots qui sont suffisamment différents deux à deux, comme* $K = \{\texttt{aaaaabb}, \texttt{ababab}, \texttt{bbaaa}\}$*, avec* $r(\texttt{aaaaabb}, \texttt{ababab}) = 3$*,* $r(\texttt{aaaaabb}, \texttt{bbaaa}) = 4$ *et* $r(\texttt{ababab}, \texttt{bbaaa}) = 3$*. b) Étant donné un autre mot, on peut alors déterminer le mot de l'ensemble le plus proche de ce mot : le mot* `aaababb` *est ainsi plus proche de* `aaaaabb` *que des autres mots de* K*.*

> *insertions, and reversals if and only if* $r(x, y) > 2s$ *for any two different words* x *and* y *in* K*. »*

Cette distance de Levensthein est donc le nombre minimal d'opérations pour transformer un mot en un autre en utilisant les *opérations d'édition* que sont le remplacement, la suppression ou l'insertion d'un caractère (Fig. 1.1). La distance de Levenshtein peut se calculer par programmation dynamique qui est un principe d'optimisation [33]. Plus généralement, on peut fixer des *scores* ou des *poids* qui favorisent ou pénalisent certaines opérations, et calculer des alignements *globaux* (une séquence contre une autre) ou *locaux* (une partie d'une séquence contre une partie d'une autre) [37, 47, 49].

Algorithmique du texte. Comparer, et plus généralement traiter, analyser ou indexer les séquences de caractères constitue le champ de recherches de *l'algorithmique du texte*. Les problèmes de *recherche de motifs* ont été abordés dès les années 1970 [40, 43] et ont par la suite suscité de nombreux travaux [161]. En parallèle, des études ont exploré les propriétés combinatoires et statistiques des mots [52].

Aujourd'hui l'algorithmique du texte est un domaine établi, avec une communauté se retrouvant lors d'événements tels que *Combinatorial Pattern Matching* (CPM, depuis 1992), *String Processing and Information Retrieval* (SPIRE, auparvant *South American Workshop on String Processing*, depuis 1993) et *Prague Stringology Club* (PSC, depuis 1996). Plusieurs ouvrages de référence présentent ce domaine [64, 65, 86], dont, en français, [71].

Les algorithmes traitant les séquences de caractères doivent être *efficaces* : si ce document d'une centaine de pages contient environ 160 000 caractères formant 30 000 mots, des recherches sur des séquences d'ADN, sur des documents ou sur des chansons peuvent concerner des millions, des milliards, voire des millions de millions de caractères et de mots, de nucléotides ou de notes. Dans les structures de données particulièrement efficaces, on peut mentionner les *tables de hachage*, les *arbres de suffixes*, et, conçues à partir des années 1990, les *tables de suffixes* et la *transformée de Burrows-Wheeler* [109].

Donner du sens à la comparaison de séquences. Une comparaison lettre à lettre n'est pas toujours la plus pertinente, car les mots sont construits de syllabes et de racines (Fig. 1.2). Idéalement, la comparaison entre séquences doit leur donner du sens, révéler leur organisation : grâce aux travaux menés depuis plusieurs décennies sur le langage naturel, certains moteurs de recherche peuvent aujourd'hui saisir partiellement le sens de phrases. Y a-t-il aussi un sens à comparer des séquences biologiques ou musicales ?

De génération en génération de cellules, les séquences d'ADN *mutent*, que ce soit des insertions, suppressions, et substitutions de nucléotides (dues à des « erreurs » de l'enzyme qui recopient l'ADN) ou, à plus grande échelle, des recombinaisons et réarrangements de gènes ou de portions de chromosomes. Ces transformations, bien qu'elles soient rares, sont moteur de l'évolution des cellules

```
a) immunologie          b) immun  ologie        c) [immuno][logie]
   xxxxx|||||| 5           xx||xxx|||||| 5            x       |
   musicologie               musicologie           [musico][logie]
```

FIGURE 1.2 – *Alignement de mots. Les alignements a) et b) sont les deux alignements optimaux suivant la distance d'édition, avec 5 opérations. Selon les poids donnés aux différentes opérations d'édition, on pourra considérer que l'alignement a) ou b) est le meilleur. L'alignement b) est celui qui propose le plus de correspondances – 8 lettres identiques, dont une syllabe* `mu` *conservée. Cependant, ce* `mu` *n'a pas vraiment beaucoup de sens ici. L'alignement c) fait lui correspondre des racines lexicales. S'appuyant sur l'étymologie, il éclaire un aspect de la signification des mots.*

et des espèces car elles apportent parfois un avantage sélectif. *« Rien n'a de sens en biologie, si ce n'est à la lumière de l'évolution »* était le titre d'un essai de 1973 du généticien Theodosius Dobjansky [38] : comparer des séquences d'ADN, c'est retracer certains mécanismes biologiques et donc expliquer quelques aspects de l'évolution. Dans ce document, je parlerai de mécanismes de recombinaison et de mutation très particuliers, les *recombinaisons immunologiques*, qui assurent la diversité de notre système immunitaire – système fonctionnant lui aussi grâce à une évolution et une pression de sélection.

Je parlerai aussi de musique... Un *motif musical* est « quelque chose qui se répète » pouvant être quelques notes, une phrase musicale, ou même un élément rythmique ou de texture. La musique est en effet organisée en répétitions et en contrastes. Là encore, ce n'est pas une comparaison note à note entre chansons qui donne du sens à la musique, mais une explication des fonctions de différents éléments musicaux. Une partition regroupe souvent quelques milliers de notes, dans une organisation séquentielle qui n'est plus uniquement linéaire, mais à deux dimensions, avec des hauteurs et du rythme.

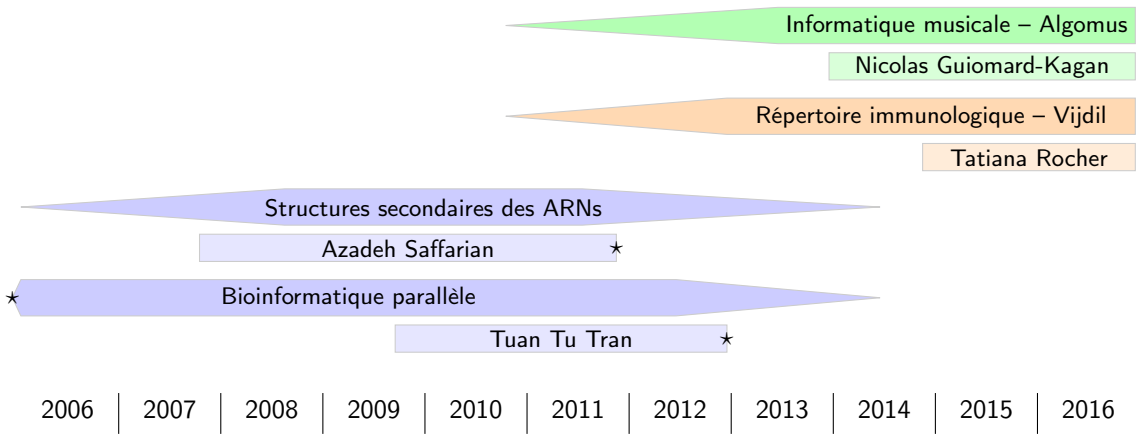


FIGURE 1.3 – *Thèmes de recherche et doctorants encadrés.*

Contenu de ce manuscrit. Les séquences de caractères ont rythmé mon parcours académique. En thèse, dans l'équipe Symbiose à Rennes, je travaillais avec Dominique Lavenier sur la comparaison de séquences protéiques avec des processeurs particuliers, les « processeurs reconfigurables FPGA ». En 2006, j'ai aussi commencé à étudier les structures secondaires d'ARN avec Nadia El-Mabrouk, à Montréal. J'ai été recruté fin 2006 dans le laboratoire LIFL, maintenant CRIStAL (UMR 9189, Université de Lille). En arrivant à Lille, j'ai rejoint Hélène Touzet et l'équipe de bioinformatique (alors Sequoia, et maintenant Bonsai), et mon projet était d'approfondir ces thématiques.

Dans les années 2006 – 2012, j'ai ainsi mené des recherches sur le *calcul parallèle pour la bioinformatique* tout en me rapprochant du logiciel, principalement autour du calcul sur cartes graphiques (GPGPU), notamment avec Jean-Stéphane Varré, Stéphane Janot et Jean-Frédéric Berthelot ainsi que par la thèse de Tuan Tu Tran. Là encore, les algorithmes étudiés concernaient la comparaison de séquences d'ADN. J'ai aussi travaillé avec Hélène Touzet sur les *algorithmes sur les multi-structures d'ARN*, autour de la thèse d'Azadeh Saffarian. Ces deux thématiques – bioinformatique parallèle et multi-structures d'ARN – ne seront pas mentionnées ici. J'ai préféré focaliser ce manuscrit sur les deux projets que je porte depuis les cinq dernières années :

— un projet de bioinformatique appliqué à l'hématologie et l'immunologie sur l'*analyse des populations de lymphocytes par leurs recombinaisons V(D)J* (« Compter les globules blancs », chapitres 2 à 6). Ce projet a débuté dans l'équipe Bonsai fin 2010 (au début, dans le cadre d'une soumission d'un projet avec du parallélisme GPGPU !), en constante collaboration avec nos collègues bioinformaticiens (plateforme de séquençage, Martin Figeac) et hématologues (laboratoire de Claude Preudhomme) de l'hôpital de Lille. Avec mon complice et collègue Mikaël Salson, nous avons réfléchi sur des algorithmes pour analyser un mécanisme très particulier – la recombinaison V(D)J.

  Le projet s'est depuis étendu, avec un travail d'implémentation et une application clinique : le logiciel Vidjil que nous développons est désormais capable d'analyser plus de recombinaisons immunologiques et s'est transformé en une plateforme contenant une application web, une base de données de patients et un serveur. Vidjil est utilisé régulièrement par plusieurs laboratoires en France et à l'étranger. Depuis janvier 2015, il est utilisé en routine à l'hôpital de Lille.

— un projet plus prospectif d'informatique musicale (« Analyser les partitions », chapitres 7 à 10). Un ordinateur est-il capable de comprendre la musique ? L'équipe émergente Algomus, que je dirige, est répartie entre les laboratoires MIS (Amiens, Univ. Picardie Jules Verne) et CRIStAL et développe des méthodes et des outils d'analyse musicale, c'est-à-dire d'annotation de partition musicale. Suite à des discussions informelles avec mes collègues Richard Groult et Florence Levé, du MIS, nous avons eu une première publication en 2011. Notre collaboration s'est amplifiée à partir de 2012, début de notre travail sur les fugues.

  Aujourd'hui, nous travaillons sur plusieurs problèmes de modélisation et de calcul sur partitions musicales – analyse de forme sonates, séparation de voix, analyse de textures, modélisation par grammaires – en lien avec des collègues musicologues. Le point commun de tous ces travaux est qu'ils cherchent à expliquer le plus possible le fonctionnement interne des partitions. Nous travaillons enfin sur la visualisation de partitions annotées.

Ce manuscrit se termine par la description de quelques actions de médiation scientifique et artistique ainsi que des réflexions sur ces deux domaines d'étude (chapitres 11 et 12). Séquences bioinformatiques, composées de A, C, G et T, séquences musicales composées de notes... Les deux domaines ont en commun des méthodes similaires en algorithmique du texte : représentation de séquences, techniques de comparaison, d'indexation, de filtrage.... Je ne ferai cependant pas de lien artificiel entre ces deux directions de recherches, les métiers et les finalités applicatives étant bien différentes. Je discuterai en particulier de l'évaluation de telles études interdisciplinaires, que ce soit au niveau des méthodes mathématiques et informatiques, ou des applications en immunologie et en musique.

Première partie

Compter les globules blancs, Vidjil

Donne-moi ton sang, je te dirai quels sont tes lymphocytes...

Les recombinaisons V(D)J sont au cœur de la diversité immunologique qui permet à notre corps de se défendre de manière adaptée à un grand nombre d'infections (chapitre 2). En collaboration avec la plateforme de séquençage (Martin Figeac, Shéhérazade Sebda) et le laboratoire d'hématologie du CHRU de Lille (Claude Preudhomme, Nathalie Grardel, Yann Ferret, Aurélie Caillault, Nicolas Duployez), nous travaillons depuis fin 2011 sur des méthodes bioinformatiques pour l'hématologie, dans un projet mêlant aspects algorithmiques et cliniques.

Avec Mikaël Salson, j'ai proposé un nouvel algorithme rapide et sûr pour analyser les recombinaisons V(D)J (chapitre 3). En particulier grâce à Marc Duez, ingénieur recruté dans l'équipe en 2013, nous avons développé Vidjil, une plateforme d'analyse des populations de lymphocytes (chapitre 4). Nous travaillons en collaboration avec des laboratoires d'hématologie en France et à l'étranger qui utilisent régulièrement notre logiciel à des buts cliniques – diagnostic des leucémies aiguës – ou de recherche (chapitre 5).

Les perspectives de ce travail (chapitre 6) sont à la fois algorithmiques – avec Mikaël Salson et Jean-Stéphane Varré, je co-encadre la thèse de Tatiana Rocher sur l'indexation des recombinaisons V(D)J – et appliquées – Ryan Herbert, ingénieur recruté pour deux ans par une ADT Inria, développe de nouvelles fonctionnalités pour la plateforme, dans le but de mieux répondre aux défis bioinformatiques soulevés par nos utilisateurs ou collaborateurs hématologues.

Vidjil est fortement soutenu par la Région Nord-Pas-de-Calais, la plateforme de bioinformatique Bilille, le SIRIC OncoLille et Inria, lors de projets que j'ai portés ou co-portés avec Mikaël Salson.

2 Immunologie et oncologie

Les lymphocytes, une partie des globules blancs, jouent un rôle clé dans l'immunité adaptative, en particulier grâce au mécanisme de recombinaison V(D)J (section 2.1). La recombinaison V(D)J est un marqueur utile au diagnostic et au suivi des leucémies aiguës (section 2.2). Les pages suivantes n'ont pas la prétention d'être un exposé complet d'immunologie ou d'oncologie, mais présentent quelques éléments pour comprendre le but du séquençage de répertoire immunologique (Rep-Seq, section 2.3). Compter les globules blancs, les regrouper par familles, c'est ainsi comprendre l'état du système immunitaire d'une personne à un moment donné.

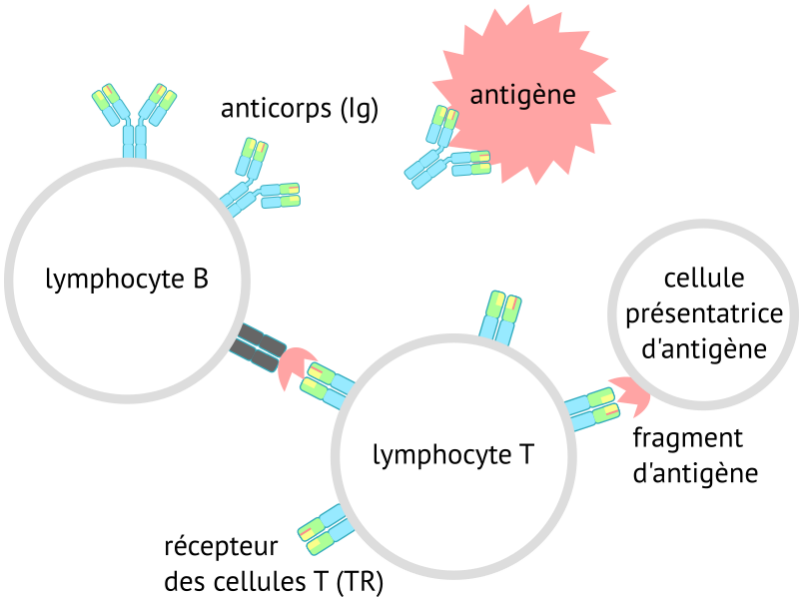


FIGURE 2.1 – *Les lymphocytes T et B coopèrent pour réaliser l'immunité adaptative. Les lymphocytes T produisent des récepteurs d'antigène (TR) reconnaissant un fragment d'antigène présenté par d'autres cellules. Les lymphocytes B produisent eux des anticorps, ou immunoglobulines (Ig), membranaires ou secrétées, qui reconnaissent directement l'antigène.*

Bien que ce chapitre s'adresse au lecteur non spécialiste, le choix a été fait ici de ne pas réexpliquer des généralités sur les cellules et l'ADN, généralités qui peuvent se trouver dans de nombreux ouvrages. Une partie de ce chapitre, rédigée avec Mikaël Salson, sera utilisée pour un article à destination du grand public.

2.1 Hématopoïèse et recombinaison V(D)J

L'immunité adaptative. *L'hématopoïèse* est le processus de production des cellules du sang. La famille des *lymphocytes* contribue à lutter contre les *antigènes* (composants bactériens, viraux, parasitaires ou issus de tout autre pathogène). Les lymphocytes B effectuent leur maturation dans la moelle osseuse, et produisent des *anticorps* ou immunoglobulines (Ig). Les lymphocytes T sont eux produits dans le thymus, un organisme situé entre les poumons. À leur surface, ils présentent des *récepteurs d'antigène* (TR) activés par des *épitopes*, c'est-à-dire des fragments d'antigènes, présentés par des lymphocytes B ou par d'autres cellules (Fig. 2.1).

Les anticorps et les récepteurs d'antigène sont multiples : le répertoire total des Ig et des TR est estimé à environ 10^{12} molécules [77]. Cette diversité permet une *réponse immunitaire adaptative* : lors d'une infection, certaines populations de lymphocytes B ou T vont reconnaître (imparfaitement) un certain nombre d'antigènes de l'agent pathogène. En 2 à 8 jours, ces populations se multiplient, en particulier celles qui sont les plus adaptées, menant à une spécialisation encore plus forte pour lutter contre ces antigènes. Elles sont ainsi sélectionnées pour la réponse immunitaire. Lorsque l'infection est résorbée, une partie de ces populations reste dans l'organisme. Ces *lymphocytes mémoire* contribuent à une réponse immunitaire plus rapide et efficace lors d'infections ultérieures par le même pathogène ou un agent très similaire, présentant des antigènes communs.

Les recombinaisons V(D)J. Le premier mécanisme expliquant la diversité des anticorps et des récepteurs d'antigène est la *recombinaison V(D)J* qui crée la région *CDR3* (*complementary determining region 3*) [53, 78]. La région CDR3 étant précisément celle en contact avec l'antigène, c'est cette variabilité qui explique la spécificité d'un Ig ou TR pour un fragment d'antigène donné (Fig. 2.2). Une recombinaison V(D)J met en œuvre des segments V, éventuellement D, et J provenant de gènes V, D, J qui peuvent avoir été tronqués ou mutés (Figs. 2.3 et 2.4). Entre ces segments, la *région de N-diversité* peut inclure des nucléotides aléatoires.

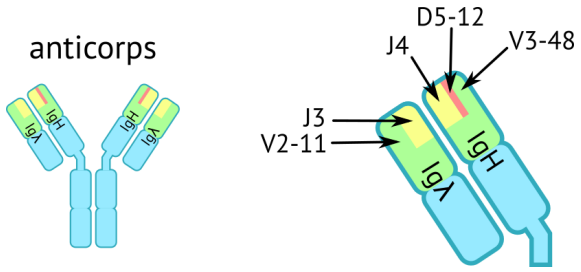


FIGURE 2.2 – *Cet anticorps est composé de deux chaînes lourdes identiques (IgH) et de deux chaînes légères identiques (Igλ). C'est précisément la zone au contact de l'antigène, le CDR3, qui est formée par les recombinaisons V(D)J. La chaîne lourde est ici recombinée à partir des gènes V3-48, D5-12 et J4 du locus IgH, sur le chromosome 14 (Fig. 2.4), tandis que la chaîne légère est recombinée à partir des gènes V2-11 et J3 du locus Igλ, sur le chromosome 22.*

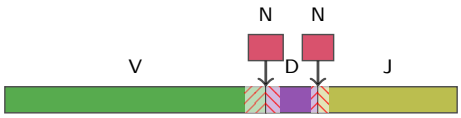


FIGURE 2.3 – *Recombinaison V(D)J. Les gènes V font typiquement entre 250 et 310 nucléotides (nt), les gènes D entre 10 et 35 nt, et les gènes J entre 40 et 70 nt.*

Les recombinaisons peuvent être soit VJ, soit VDJ. Ces recombinaisons peuvent se dérouler dans plusieurs *locus*, c'est-à-dire à plusieurs endroits du génome, simultanément ou successivement (Fig. 2.5). Au sein de chaque locus, les recombinaisons suivent un ordre bien déterminé.

```
                                      ++            **
IGHV3-48*01 ...TGTGTATTACTGTGCGAGAGA
recomb      ...TGTGTATTACTGTGCGAGAGAAAATAGTGGCTACGATTTGACTACTGGGGCCAGGG...
IGHD5-12*01                       gtggatATAGTGGCTACGATTac
                                  123456            4321
IGHJ4*02                                        actacTTTGACTACTGGGGCCAGGG...
                                                1234567
```

FIGURE 2.4 – *Exemple de recombinaison VDJ sur le locus IgH, provenant d'un patient suivi à Lille. La séquence* `recomb` *peut se décrire par le code* `IGHV3-48 0/AA/6 IGHD5-12 3//6 IGHJ4`*, qui s'explique de la manière suivante :*

Recombinaison V/D : `0/AA/6`**.** *Le gène V,* ***IGHV3-48****, a été pris entièrement. Notons que seule la fin de ce gène (qui fait 296 nucléotides) est representée ici. Le gène D,* ***IGHD5-12****, a lui perdu ses 6 premiers nucléotides (*`gtggat`*). Deux nucléotides* `AA`*, marqués par* `++`*, ont été rajoutés entre les gènes V et D.*

Recombinaison D/J : `2//7`**,** `3//6` **ou** `4//5`**.** *Dans le locus IgH, la recombinaison D/J est en fait la première, se faisant avant la recombinaison V/D. Le gène D,* ***IGHD5-12****, a été recombiné avec le gène J* ***IGHJ4*** *(seul le début du gène J, qui fait 48 nucléotides, est représenté ici). La simple vue de la séquence ne permet pas ici de déterminer quelle a été exactement la recombinaison : les nucléotides* ***TT****, marqués par* ********, *sont alignés aussi bien avec la fin du gène D qu'avec le début du gène J. Il y a donc plusieurs interprétations possibles sur le nombre de délétions à la fin du gène D et au début du gène J : la recombinaison D/J peut-être vue comme 2//7, 3//6, ou 4//5. Il n'y a probablement pas eu ici de nucléotides insérés.*

Par exemple, pour le locus IgH, la recombinaison D/J a lieu avant la recombinaison V/(D/J). Un précurseur de lymphocyte resté à un certain stade de développement possédera ainsi une recombinaison incomplète.

Diversité du répertoire immunologique Un lymphocyte B ou T a normalement besoin de deux (et uniquement deux) locus recombinés, suivant le type d'Ig ou de TR exprimé :

Lymphocytes B	Chaîne lourde IgH (VDJ) + chaîne légère Igκ ou Igλ (VD)
Lymphocytes T	Chaîne TRα (VD) + TRβ (VDJ) ou chaîne TRγ (VD) + TRδ (VDJ)

Le fait que les recombinaisons V(D)J aient lieu sur les deux chaînes utilisées par un lymphocyte augmente considérablement la diversité des anticorps (Tab. 2.6). Au cours de sa maturation, plusieurs choix aléatoires sont effectués pour assurer cette diversité. La recombinaison de certains locus bloque ainsi d'autres recombinaisons, que ce soit sur l'autre *allèle* (un gène similaire présent sur l'autre chromosome de même numéro) ou même sur d'autres locus. Par exemple, le locus TRδ est tout simplement éliminé lors d'une recombinaison TRα. Cependant, on peut trouver des recombinaisons non fonctionnelles ou multiples [155]. Une majorité des lymphocytes B dans les leucémies aiguës lymphoblastiques présentent ainsi des recombinaisons dans le locus TRγ [55]. Ces lymphocytes B ne produiront pas pour autant de récepteurs γ. Un même lymphocyte ou un précurseur de lymphocyte pourra ainsi avoir de zéro à près d'une dizaine de recombinaisons V(D)J différentes.

2.2 Diagnostic et suivi des leucémies aiguës lymphoblastiques

Leucémie aigüe lymphoblastique (LAL). Les cancers du sang regroupent les *leucémies*, aiguës ou chroniques, les lymphomes et les myélomes multiples. La *leucémie aiguë lymphoblastique* (LAL) est un cancer touchant principalement les enfants. Une population de cellules, appelée *lymphoblaste*, prolifère anormalement (Fig. 2.7). Ces cellules sont plus ou moins immatures – et

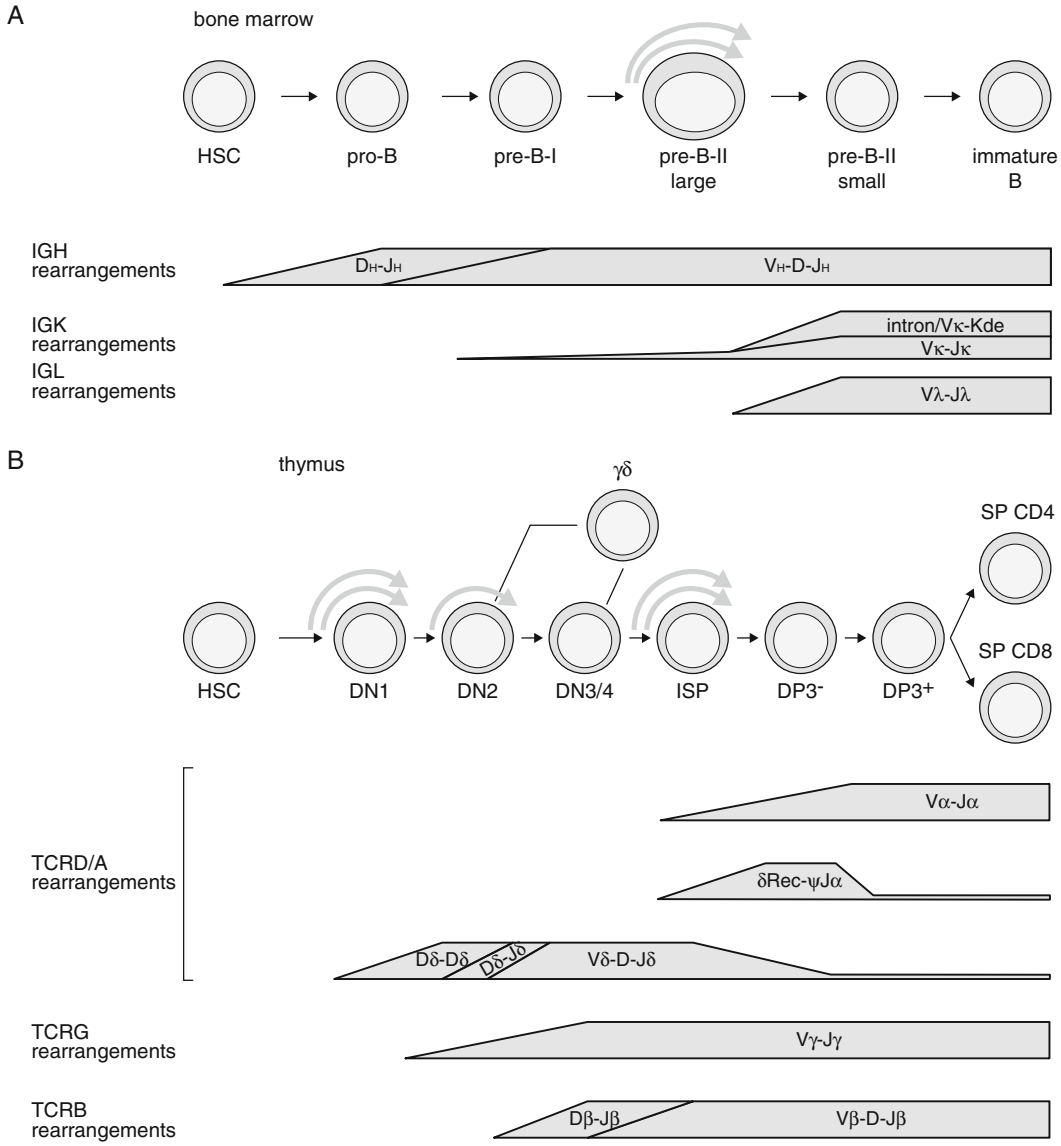


FIGURE 2.5 – *Développement lymphoïde et recombinaison des locus pour les lympohcytes B (haut) et T (bas). Figure extraite de [155]. Les recombinaisons se font au cours de la maturation de la cellule. Par exemple, le lymphocyte « pro-B » n'a intialement effectué que la recombinaison D-J sur le locus IgH, et aucune recombinaison sur les locus Igλ et Igκ.*

	Chaîne lourde	Chaîne légère κ ou λ	
locus	IgH (14q32.33)	Igκ (2p11.2)	Igλ (22q11.2)
répertoire	$\sim 40\,\mathrm{V} \times 23\,\mathrm{D} \times 6\,\mathrm{J}$	$\sim 30\,\mathrm{V} \times 5\,\mathrm{J}$	$\sim 30\,\mathrm{V} \times 5\,\mathrm{J}$
recombinaisons V(D)J	~ 6300	~ 150	~ 150
N-diversité, mutations somatiques	$\sim 6 \cdot 10^6$	$\sim 3 \cdot 10^5$	$\sim 3 \cdot 10^5$
	$2 \cdot 10^{12}$ anticorps différents		

TABLE 2.6 – *Diversité du répertoire immunologique des lymphocytes B [77]. Les chaînes proviennent de trois locus situés sur des chromosomes différents (14, 2 et 22). En plus de la recombinaison V(D)J avec la N-diversité, les séquences des lymphocytes B peuvent avoir des* mutations somatiques *qui améliorent encore la spécificité de l'anticorps pour un certain antigène.*

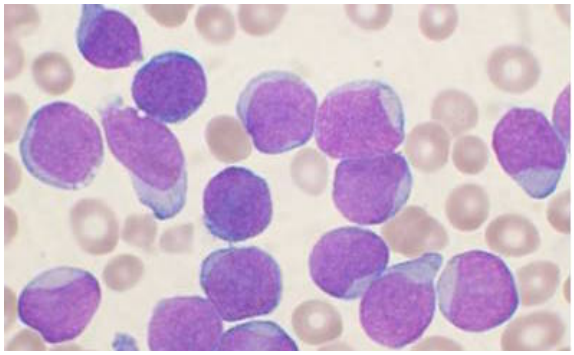


FIGURE 2.7 – *Leucémie aiguë lymphoblastique (LAL). Les lymphoblastes sont ici des précurseurs de lymphocyte B. (CC BY SA VashiDonsk, Wikipedia)*

certaines n'ont pas terminé leurs recombinaisons V(D)J, ayant par exemple réalisé uniquement la recombinaison D/J sur le locus IgH.

Les symptômes d'une leucémie aiguë sont un affaiblissement généralisé, provenant en particulier d'une *anémie* (diminution du taux d'hémoglobine dans le sang liée ici à une diminution des globules rouges). L'hématopoïèse défaillante provoque aussi une très grande vulnérabilité aux infections extérieures. En effet, lorsqu'un patient est diagnostiqué d'une LAL, la quasi totalité de ses globules blancs sont les lymphoblastes, et il n'y a quasiment plus de lymphocytes matures, ni de polynucléaires et monocytes (*leucopénie*). Enfin, des troubles de la coagulation sanguine peuvent apparaître en raison de la diminution de la production de plaquettes.

Prise en charge, diagnostic et suivi des LAL. Le pronostic des patients atteints de LAL s'est considérablement amélioré lors des dernières décades (survie à 5 ans de 41 % en 1975, 70 % en 2010). Les dernières années, peu de nouvelles molécules ont été découvertes. C'est surtout la *meilleure utilisation de l'arsenal thérapeutique existant* qui perfectionne les traitements en s'appuyant sur un diagnostic et un suivi plus précis.

Le suivi hématologique se fait par l'évaluation de la maladie résiduelle *(minimal residual disease, MRD)*, en quantifiant pour chaque échantillon de suivi la proportion du clone détecté au diagnostic. Cette évaluation se fait par une PCR quantitative (qPCR, estimation de la quantité de cellules ayant une certaine séquence d'ADN par amplification de cette séquence) [77] ou par cytométrie en flux (tri des cellules selon la présence de certains marqueurs) [154]. Dans plus de 90% des cas de LAL, on arrive à identifier une recombinaison V(D)J sur un des locus Ig ou TR des lymphoblastes : cette recombinaison est le marqueur suivi par qPCR. Ce suivi est standardisé par le consortium européen EuroMRD.

```
>00142:00601
AGCAGTGGGTAAGACAAGACAACA

>00313:01781
AGCAGTGGGTAAGACAAGACAACAAAGTGGTAGGCAAGAAAGAATTTCTTCAAACTCTCATCTTCAATCCCATTTACCATCAAGCTCC

>00193:00174
GTGTTGTTCCACTGCCAAAGAGAGTCGCAAAACGGTTGAAAAGGACACTGACTGGGAATTGAGAGCCCTGGGTTCATCATGCTAGCCCTGACAATAATT
TGTCATGTAATCTTGAACAAGCCCAGGTACTCTGGCTTTAATTGCCTTGTGAGTAAAATAGACGATGGTACTAAGTGATTATTTAAAATCCTTTCCAAT
TGTAGAAGTCCAAGGTTGTGTGAAATTTGGATATGTAGGTGAAAGACGCTGTGGGGCCTTCC

>00153:01261
GTGTTGTTCCACTGCCAAAGAGAGTCGCAAAACGGTTGAAAAGGACACTGACTGGGAATTGAGAGCCCTGGGTTCATCATGCTAGCCCTGACAATAATT
TGTCATGTAATCTTGAACAAGCCCAGGTACTCTGGCTTTAATTGCCTTGTGAGTAAAATAGACGATGGTACTAAGTGATTATTTAAAATCCTTTCCAAT
TGTAGAAGTCCAAGGTTGTGTGAAATTTGGATATGTAGGTGAAAGACGCTGTGGGGCCTTCC

>00670:02597
GGAAGGCCCCACAGCGTCTTTCACCTACATATCCAAATTTCACACAACCTTGGACTTCTACAATTGGAAAGGATTTTAAATAATCACTTAGTACCATCG
TCTATTTTACTCACAAGGCAATTAAAGCCAGAGCACCTGGGCTTGTTCAAGATTACATGACAAATTATTGTCAGGGCTAGCATGATGAACCCAGGGCTC
TCAATTCCCAGTCAGTGTCCTTTTCAACCGTTTTGCGACTCTCTTTGGCAGTGGAACAACAC

>00320:01219
GACGGCCCACAGCGTCTTTCACCTACATATCCAAATTTCACAGCAACCTTGGACTTCCTACAAGTTTGGAAAGGATTTTAAATATCACTTAGGTACCAT
CGTCTATTTTACTCACCAAGGCAATTAAA

>00254:00018
GTCGGTTGTTCCGACTGCCCAAAGGAAGTTTTCGTTATATTCGAATCCCCAGGTGGCTACAGTAAGTAGACGTTCCAGAGTCATTTTCAATAAGATTTC
CGCAGTTACA
```

FIGURE 2.8 – *Exemples de* reads *en sortie d'un séquenceur haut-débit lors d'une étude de Rep-Seq. Les séquenceurs haut-débit produisent aujourd'hui des milliers ou des millions de telles séquences, qui peuvent contenir jusqu'à plusieurs milliards de nucléotides au total. Le défi bioinformatique est d'analyser cet ensemble de séquences, en identifiant les recombinaisons V(D)J et les regroupant en clones (voir chapitre 3).*

La *stratification* des patients consiste à identifier le plus tôt possible le pronostic de chaque patient pour adapter au mieux son traitement. Par exemple, dans le protocole européen actuellement suivi (2010 EORTC 58081), les LAL de l'enfant sont réparties dans les groupes VHR (très haut risque), AR1 et AR2 (risque moyen) et SR (faible risque). Cette répartition dépend notamment de l'évaluation de la MRD au jour 35 après le diagnostic [73]. D'autres évaluations sont faites au jour 70, puis régulièrement, en fonction de l'état du patient. La MRD est aujourd'hui l'un des meilleurs critères de stratification [152] et permet, dans certains cas, de détecter précocement les rechutes.

2.3 Séquençage à haut-débit de répertoire immunologique (Rep-Seq)

Rep-Seq. La principale limite des techniques habituelles de MRD est qu'elles ne peuvent pas suivre des populations de plusieurs clones [93]. Les techniques ne sont pas capables d'identifier une rechute provenant d'un clone non identifié au diagnostic – soit parce qu'il était absent, soit en quantité trop faible.

De plus, on considère aujourd'hui que la leucémie est une maladie *hétérogène*. Ce n'est pas un clone mais une famille de clones, voire une population de clones qui est responsable de la maladie. Comprendre la leucémie (et, plus généralement, comprendre la réponse immunitaire) nécessite donc d'étudier, autant que possible, *l'ensemble de la population de lymphocytes* et donc l'ensemble de ses recombinaisons V(D)J.

Depuis une dizaine d'années, on dispose de *séquenceurs à haut-débit* (HTS ou NGS) qui ont révolutionné de nombreux champs de la biologie moléculaire. Un séquenceur permet d'obtenir la composition d'un très grand nombre de séquences d'ADN sous la forme de *reads* (ou traces

de lectures) [1] (Fig. 2.8). Dans notre cas, le principe du séquençage de répertoire immunologique (Repertoire Sequencing, *Rep-Seq*) est d'analyser tout un ensemble de lymphocytes, en séquençant des milliers voire des millions de recombinaisons immunologiques [149]. Les premières études sur ce sujet datent de 2009, que ce soit pour étudier le répertoire d'animaux [129] ou bien d'humains [124], en particulier sur des patients atteints de leucémie [120]. Depuis, de nombreuses études biologiques ont été publiées (revue dans [181], voir aussi à la section 3.1).

Stratégies pour le Rep-Seq. Séquencer un répertoire demande d'extraire les recombinaisons V(D)J de l'ADN des lymphocytes (Fig. 2.9). Pour cela, la plupart des études utilisent une ou plusieurs *réactions de PCR* (amplification sélective de l'ADN) avec des *amorces* consensus, c'est-à-dire capable de se fixer sur un ensemble de gènes V et J, comme les amorces développées au début des années 2000 par l'action BIOMED-2 [77]. On doit cependant utiliser au total 122 amorces réparties en 14 réactions de PCR (des « tubes », ou PCR multiplex) pour couvrir la majorité des locus. De nouvelles amorces, plus adaptées au NGS, sont en cours d'évaluation dans le groupe de travail EuroClonality-NGS (voir section 6.2).

Le principal défaut des stratégies de PCR est qu'elles consistent en une amplification du matériel génétique. Cette amplification peut induire un biais, rendant plus difficile la quantification précise (voir section 5.2), en particulier dès que plusieurs amorces sont mélangées. D'autres stratégies de séquençage sans amplification commencent à être développées, comme la *capture*, qui consiste à séquencer des fragments autour de séquences connues. Là encore, le choix des sondes de capture a une grande influence sur le résultat final. Enfin, il est possible de simplement rechercher les recombinaisons V(D)J au milieu d'autres données. On peut ainsi trouver des recombinaisons V(D)J dans des données de transcriptomique (séquençage des ARNs présents dans une cellule, *RNA-Seq*), même si ces recombinaisons seront peu nombreuses comparées à l'ensemble des ARNs produits par la cellule. Par contre, on ne trouvera pas dans les ARNs les recombinaisons non exprimées.

1. voir notre présentation grand public des séquenceurs à haut-débit [1].

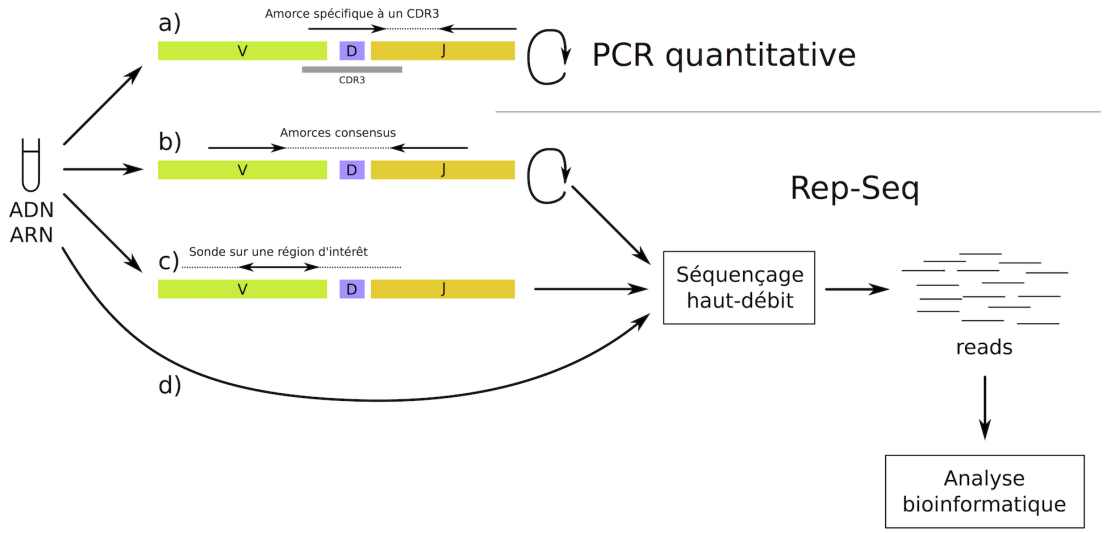


FIGURE 2.9 – *Stratégies pour le Rep-Seq (b, c et d). Aujourd'hui, l'approche principale pour étudier le répertoire immunologique est d'utiliser des amorces de PCR.*

a) La quantification traditionnelle en qPCR utilise des amorces spécifiques à un CDR3.

b) L'approche Rep-Seq utilise elle des amorces consensus entre plusieurs V ou plusieurs J, telles que les amorces BIOMED-2 [77]. Selon les amorces utilisées, la quasi-totalité des séquences récupérées sont des recombinaisons V(D)J pouvant être analysées par un outil bioinformatique de Rep-Seq.

c) La capture cible des régions d'intérêt (y compris des gènes V, D ou J), ce qui permet d'être plus flexible sur les séquences récupérées. Cependant, cette méthode récupère aussi les gènes non recombinés, ce qui ne fournit qu'une faible proportion de séquences recombinées.

d) Le RNA-Seq consiste à récupérer les ARN d'une population de cellules sans a priori. Là encore, une faible proportion de l'ARN total consiste en des séquences recombinées.

Avec de la PCR ou de la capture, on peut aussi se focaliser sur de l'ARN présentant certaines régions d'intérêt. En capture comme en RNA-seq, un seul séquençage permet d'analyser différents types d'événements pertinents (comme, pour l'oncologie, d'autres recombinaisons comme BCR-ABL, de l'épissage alternatif ou des mutations de la tumeur). Le but des outils bioinformatiques de Rep-Seq est alors de se focaliser sur la proportion (faible) des séquences recombinées. Comme ces approches peuvent être conduites sans amplification, elles ont, à terme, un potentiel de quantification avec moins de biais, mais elles demandent encore beaucoup de calibration.

3 Algorithmes pour l'analyse des recombinaisons V(D)J

Pouvons-nous compter les globules blancs qui luttent contre le virus de la grippe ? Un des défis de l'immunoinformatique est de pouvoir, à partir des séquences d'ADN, décrire la fonction d'un lymphocyte. La spécificité d'un lymphocyte reposant principalement sur son CDR3, l'analyse des recombinaisons V(D)J devrait, idéalement, prédire avec quel antigène le lymphocyte est capable d'interagir. Ainsi, une identification et un décompte d'un échantillon de lymphocytes donnerait un bilan complet des infections en cours et des immunisations existantes grâce aux infections passées ou aux vaccinations.

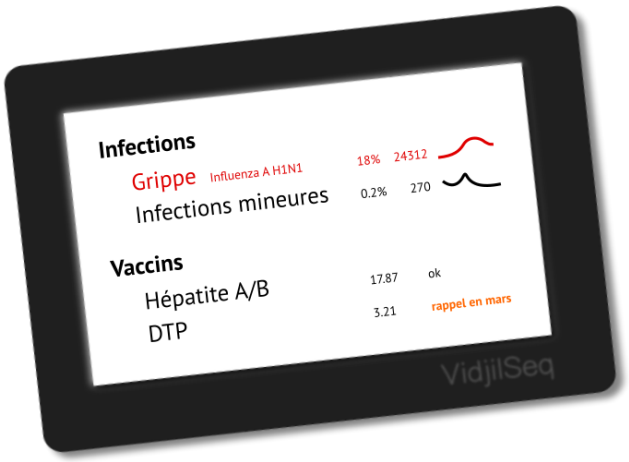


FIGURE 3.1 – *Avec les séquenceurs de troisième génération portables et bon marché, il sera peut-être possible de faire un diagnostic immédiat de son système immunitaire.*

Aujourd'hui, les programmes traitant des CDR3 ont un but bien plus modeste : ils cherchent à analyser les recombinaisons V(D)J (section 3.1). Les méthodes existantes d'analyse de séquences recombinées, classiques ou haut-débit, se basent sur cette analyse pour regrouper les reads en clones (section 3.2). Avec Mikaël Salson, j'ai proposé une vue radicalement nouvelle sur ce problème, menant à un algorithme rapide et fiable pour regrouper et analyser des populations de séquences recombinées (section 3.3). Notre méthode fonctionne également sur des recombinaisons incomplètes ou irrégulières (section 3.4).

Cet algorithme a été décrit dans notre publication de 2014 dans BMC Genomics [12]. Nous préparons une soumission d'un autre article algorithmique avec nos derniers résultats.

3.1 Défis de l'immunoinformatique et des études Rep-Seq

Que faire avec des séquences recombinées ? Modéliser directement le *lien entre recombinaisons V(D)J et antigène* est difficile : l'interaction 3D entre immunoglobuline et antigène demande des modèles chimiques précis et de grandes ressources de calcul. Il existe pour cela des logiciels de docking et des bases de données spécialisées (revues dans [165, 168, 172]).

Aujourd'hui, les approches Rep-Seq ne vont pas encore jusqu'à ce lien avec l'antigène, bien que certaines études utilisent des résultats de techniques de détection d'affinités entre anticorps et épitopes [177]. De plus, pour modéliser complètement l'immunoglobuline, on doit avoir les deux chaînes qui la constituent, avec une recombinaison V(D)J pour la chaîne lourde et une recombinaison VJ pour la chaîne légère. Lier chaînes lourdes et légères peut se faire par une technique nommée *bridge PCR* [159] ou bien par séquençage indépendant et lien statistique [176].

Sans aller jusqu'à ces modélisations complètes, de nombreuses études à haut-débit sur le répertoire sont dès à présent possibles. Étant donné une séquence recombinée, on peut tout d'abord souhaiter expliquer cette recombinaison et pour cela *identifier les régions V, D et J, avec les gènes de référence, et extraire le CDR3*. Étant donné les séquences d'un ensemble de lymphocytes, on cherchera alors à *regrouper les séquences en « clones »* pour décrire qualitativement et, si possible, quantitativement, la population globale (Fig. 3.2, haut). Comme un même lymphocyte ou lymphoblaste contient généralement plusieurs recombinaisons V(D)J (voir section 2.1), ne serait-ce que sur des locus différents, un « clone » d'un « même » lymphocyte ou lymphoblaste pourra correspondre à plusieurs « clones » de recombinaisons V(D)J.

Même si ce n'est qu'une étape dans la compréhension de la population des lymphocytes, cette détermination du répertoire V(D)J est au cœur des analyses bioinformatiques de données Rep-Seq. La connaissance de ce répertoire, l'estimation de sa diversité et la comparaison avec d'autres répertoires permettent déjà de révolutionner le diagnostic et le suivi des leucémies (voir section 2.2). De manière plus générale, en immunologie, les études Rep-Seq comparent des populations lymphocytaires correspondant à différentes situations biologiques : ces populations peuvent provenir de patients sains ou malades, de plusieurs prélèvements au cours du temps d'un même patient ou encore, à un instant donné, de prélèvements dans divers tissus [182, 167, 173]. Enfin, l'interaction immunoglobuline-antigène pourrait aussi être obtenue statistiquement : le séquençage de répertoire immunologique de plusieurs patients atteints de la même souche de grippe permettra probablement d'identifier des séquences communes de CDR3, que ce soit sur un ou plusieurs locus.

Spécificité bioinformatique de l'analyse des recombinaisons V(D)J. Les méthodes analysant des recombinaisons V(D)J doivent être spécifiques : en effet, *la plupart des outils bioinformatiques habituels sont difficiles à utiliser sur des données de Rep-Seq.* Les outils habituels d'alignement, de read mapping ou de correction d'erreurs n'ont pas été conçus pour analyser des séquences avec des recombinaisons. De plus, les variabilités technologiques (erreurs du séquençage ou de la PCR, séquences décalées) ou biologiques (zone de N-diversité, hypermutations) rendent difficile l'utilisation de ces algorithmes standards. *Lorsqu'on étudie des séquences recombinées, c'est précisément le détail de ce qui se passe dans le CDR3 qui est intéressant.*

3.2 Méthodes classiques et haut-débit

Méthodes classiques. Les premiers algorithmes spécifiques aux recombinaisons immunologiques sont apparus dans les années 1990. À Montpellier, le *international ImMunoGeneTics information system* (IMGT®), fondé par Marie-Paule Lefranc, a proposé de nombreux outils pour l'analyse approfondie de séquences avec des recombinaisons V(D)J [88, 110, 144, 148] [1]. Ce travail s'est accompagné d'un effort de standardisation, en particulier au sujet de la nomenclature des gènes, le

1. http://imgt.org/

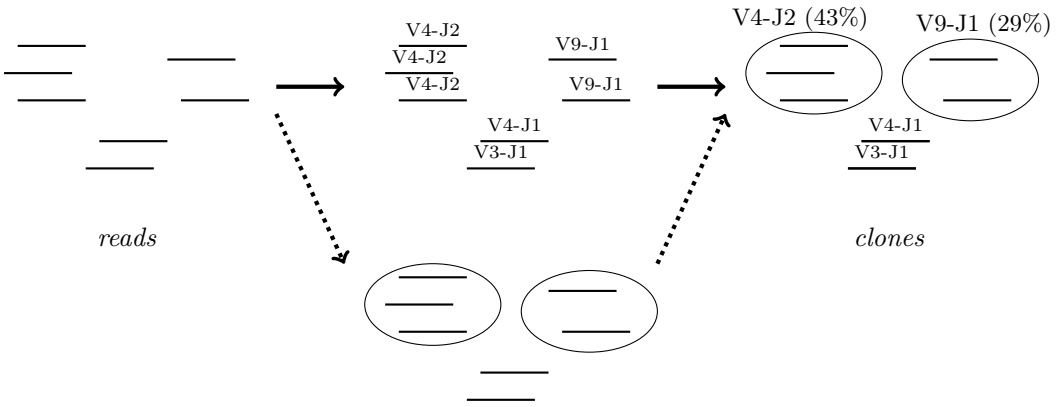


FIGURE 3.2 – *Actuellement, la plupart des méthodes bioinformatiques Rep-Seq ont pour but d'identifier et de quantifier le répertoire V(D)J en regroupant les reads en* clones. *(Haut.) Dans les méthodes classiques, chaque read est analysée pour trouver sa désignation V/D/J et/ou son CDR3, ce qui permet de regrouper les reads. (Bas, flèches pointillées.) L'algorithme que nous avons proposé regroupe les reads sans en faire une analyse détaillée. L'analyse détaillée est faite uniquement sur les clones.*

HUGO Gene Nomenclature Committee indiquant que les désignations des gènes Ig et TR suivent celles de la base de données IMGT/LIGM [144].

La plupart des logiciels d'analyse de séquences recombinées se concentrent sur la *désignation V(D)J*, identifiant les segments V, D et J recombinés les plus probables. Généralement, les logiciels calculent des *alignements* entre la séquence requête et les bases de données de gènes V(D)J (JoinSolver [87], IMGT/V-QUEST [110], IMGT/HighV-QUEST [148]), éventuellement avec certaines heuristiques ([141], IgBlast [163]), des modèles de Markov cachés (HMMs) (iHMMune-align [102], SoDA2 [140]), ou des techniques basées sur la maximisation de la probabilité (VDJSolver [100]). Une comparaison de certains de ces outils a été effectuée dans [137], mais une perspective serait de réaliser d'autres évaluations indépendantes et plus complètes.

Méthodes haut-débit : analyser, puis regrouper et compter. Le développement des méthodes Rep-Seq nécessite le développement d'algorithmes et de logiciels capables de traiter des millions de reads [149]. IMGT/HighV-QUEST [148] est une extension de IMGT/V-QUEST. Il permet d'avoir toute la puissance des outils IMGT, mais demande des calculs lourds sur chaque read. La version 1.4.1 du serveur IMGT est ainsi limitée à 500 000 reads, et analyser ces reads demande plusieurs heures de calcul sur des machines puissantes.

Dans les cinq dernières années, plusieurs logiciels ont été conçus spécifiquement pour le séquençage haut-débit, analysant beaucoup plus rapidement chaque read pour identifier le gène V, le gène J, extraire le CDR3, puis regrouper les reads en clones. Plusieurs de ces logiciels utilisent des calculs de similarité à base de k-mots (mots de longueur k), ce qui permet d'éviter ou d'accélérer le calcul des alignements complets entre les reads et les gènes de référence :

— **Decombinator** [162]. L'ensemble des gènes de référence sont analysés, pour identifier, pour chaque gène V et J, un « tag » de 20 nucléotides (nt) unique.

L'identification du gène V et du gène J dans la read se fait lorsque ce tag est exactement retrouvé, ou bien lorsqu'un demi-tag ($k = 10$ nt) est exactement retrouvé et que le tag correspondant s'aligne avec la read avec au plus une mutation. Tags et demi-tags sont recherchés en temps linéaire grâce à un automate d'Aho-Corasick [40].

Le CDR3 est ensuite localisé en cherchant une correspondance exacte de 3 nucléotides à la fin du V et au début du J. Cette localisation s'appuie sur la position du (demi-)tag identifié dans les segments V et J.

— **MiTCR** [157] et **MiXCR** [174]. L'identification des gènes V et J se fait en retrouvant des k-mots positionnés aléatoirement (avec généralement $k = 5$). L'alignement complet, par programmation dynamique avec k-band, ne se fait que lorsqu'il y a un chaînage de k-mots suffisant.

  Les clones ne sont créés qu'à partir des reads de haute qualité, et, dans une deuxième passe, les reads de basse qualité peuvent venir s'ajouter à un clone existant.

— **TCRklass** [170]. L'identification du V et du J se fait sur le nombre de 6-mers communs entre la read et les gènes de référence. Le V et le J identifiés sont ceux qui maximisent ces 6-mers communs (au moins 3) avec la read.

  La localisation du CDR3 se fait elle en reconnaissant des 3-mers *en acides aminés* (donc couvrant 9 nt). Le regroupement se fait ensuite sur les CDR3 ainsi que les gènes V et J utilisés, en deux passes selon la qualité des reads, comme dans MiTCR/MiXCR. TCRklass inclut aussi, en amont, un contrôle de qualité et un assemblage des reads paired-end, et, en aval, un traitement de certaines erreurs.

— **IMSEQ** [178]. Des k-mers sont extraits des répertoires V et J, avec leur position vis-à-vis du CDR3. L'identification du V et du J se fait par alignement complet (programmation dynamique, avec un algorithme à la k-band), mais cet alignement n'est calculé que lorsqu'il y a déjà un pré-alignement satisfaisant sur ces k-mots. Le regroupement des séquences se fait là aussi sur le CDR3. En aval, IMSEQ identifie si certains clones ne dérivent pas d'autres.

Toutes ces techniques débutent donc par l'identification d'un ou de plusieurs k-mots communs entre la read et les répertoires des gènes V et J. Plus la valeur de k est petite, meilleures sont les chances de détecter une similarité, mais plus grandes sont aussi les chances de détecter des faux positifs (comme des k-mots qui sont à la fois dans un gène V et un gène J) et de ralentir ainsi les étapes suivantes.

Decombinator et TCRklass ont une stratégie globalement linéaire – en tout cas, ces programmes ne font pas d'alignement du read avec l'ensemble des gènes V et J. Le fait qu'ils utilisent des k-mots permet de détecter des similarités approchées. Notons toutefois que Decombinator impose d'avoir au moins un demi-tag de 10 nt conservé (qui est à une position fixe sur le V) : un read avec deux mutations ou erreurs à la position des deux demi-tags ne sera donc pas analysé. Les reconnaissances de TCRklass, de MiTCR/MiXCR et d'IMSEQ sont elles plus flexibles.

3.3 Regrouper, puis compter et analyser : une nouvelle approche pour l'analyse haut-débit

Les méthodes exposées au précédent paragraphe, publiées entre 2013 et 2015, traitent les reads une par une pour obtenir les segments V et J, extraire le CDR3, puis regrouper les reads en clones (Fig. 3.2, haut). Bien que ce traitement soit optimisé, est-il vraiment indispensable d'analyser en détail chaque read ?

Avec Mikaël Salson, dès 2012, j'ai proposé d'aborder ce problème en changeant complètement de point de vue. Lorsqu'on a un grand nombre de reads (plusieurs milliers à plusieurs dizaines de millions), le but n'est pas de se focaliser sur chaque read, mais de fournir des informations pertinentes sur la population de lymphocytes et ses clones. Le problème principal devient : *Comment regrouper, le plus rapidement possible, les reads provenant d'un même clone, sans avoir identifié précisément les segments V et J ni le CDR3 ?* (Fig. 3.2, bas). Des outils génériques de regroupement de reads ne peuvent pas être employés, car de toutes petites différences peuvent conduire à des clones différents, en particulier lorsque ces différences sont dans la zone de N-diversité.

Nous avons proposé de localiser pour chaque read la zone de la jonction V(D)J, *très rapidement mais de manière approchée*, sans aucun alignement avec les répertoires V et J. Nous regroupons

les reads partageant une même *fenêtre* autour de cette jonction. L'analyse détaillée sur les clones, avec des alignements de séquence, se fait dans une seconde phase [12].

Phase 1 : Regroupement des reads suivant leur jonction V(D)J. La localisation approchée du CDR3 utilise une heuristique à base de k-mots (mots de longueur k, avec k dépendant du locus et valant généralement de 9 à 13). Tous les k-mots des gènes V et J connus sont indexés. Pour chaque read, on cherche ainsi une zone ayant à sa gauche une forte ressemblance avec des V et à sa droite une forte ressemblance avec des J.

1a. Découpe de la chaîne d'affectation en trois zones. On récupère l'ensemble des k-mots de chaque read, chacun ayant une « affectation » égale à `V`, `J`, ? (k-mer ambigu, présent à la fois dans les répertoires V et J) ou − (k-mer inconnu dans l'index). Une read de taille n donnera ainsi les $\ell = n - k + 1$ affectations $a = a_1 a_2 ... a_\ell$, comme par exemple $a =$ `VVV--VV--JJJ-`. On considère l'ensemble des positions $0, 1, ..., \ell$, et on note $a[i, j] = a_{i+1} a_{i+2} ... a_{j-1} a_j$ la zone entre les positions i et j.

La première heuristique, présentée à JOBIM 2013 [7] puis publiée dans [12], trouve dans la read deux positions $i \leq j$ telles que :

— la zone $a[0, i]$ contienne uniquement des k-mots `V` (ou ?, ou −),

— la zone $a[i, j]$ ne contienne ni `V`, ni `J`, et est la plus grande possible,

— la zone $a[j, \ell]$ contienne uniquement des k-mots `J` (ou ?, ou −).

Cette heuristique découpe ainsi la chaîne d'affectations $a =$ `VVV--VV--JJJ-` en trois zones `VVV--VV`, `--` et `JJJ-` ($i = 7$ et $j = 9$).

En 2014, nous avons amélioré cette heuristique pour tolérer un faible nombre de k-mots V au milieu du segment J, et réciproquement, comme dans une read avec une chaîne d'affectations `VVVJ-VV--JJJ-`. En notant $|s|_{\mathtt{V}}$ le nombre de caractères `V` dans une chaîne d'affectations s, on cherche les positions t telles que $\delta(t) = |a[0, t]|_{\mathtt{V}} - |a[0, t]|_{\mathtt{J}}$ soit maximum (beaucoup de `V` et peu de `J` à gauche). Cette condition est équivalente à maximiser $\delta'(t) = |a[t, \ell]|_{\mathtt{J}} - |a[t, \ell]|_{\mathtt{V}}$ (beaucoup de `J` et peu de `V` à droite), car, pour tout t, $\delta(t) - \delta'(t) = |a|_{\mathtt{V}} - |a|_{\mathtt{J}}$ est une constante. L'algorithme présenté à la figure 3.3 calcule, en temps linéaire, les deux positions $i \leq j$ qui sont la première et la dernière à maximiser δ. La chaîne d'affectation `VVVJ-VV--JJJ-` est par exemple découpée en trois zones `VVVJ-VV`, `--` et `JJJ-`, avec $\delta(7) = \delta(9) = 4$.

Comme avec tout traitement à base de k-mots, le choix de k est crucial : comment avoir un k aussi petit que possible (pour détecter des similarités approchées) tout en différenciant les zones V et J ? Avec la seconde heuristique, notre algorithme permet d'avoir des k plus petits et de tout de même détecter les zones V et J.

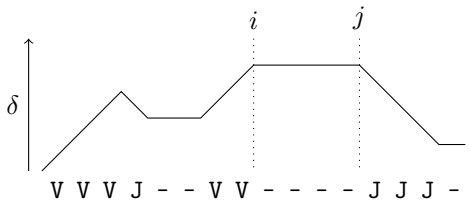
1b. Filtre et test statistique. Pour évaluer la pertinence de la découpe de la chaîne d'affectations, on vérifie tout d'abord que la zone V possède significativement plus de V que la zone J, c'est-à-dire $|a[0, i]|_{\mathtt{V}} \geq \tau \cdot |a[j, \ell]|_{\mathtt{V}}$ avec $\tau = 2$, ainsi que la condition symétrique pour les J. Ce test permet d'exclure les chaînes telles que `VVVV--JJJ--VVV-JJ` qui sont probablement chimériques et difficiles à interpréter.

On réalise enfin un simple test statistique pour savoir si la découpe est significative. La p-valeur de la zone V, c'est-à-dire la probabilité d'obtenir, par hasard, un nombre de V dans les affectations $a[0, i]$ égal à $|a[0, i]|_{\mathtt{V}}$ est estimée par $B(p, |a[0, i]|_{\mathtt{V}}, i)$, où $B(p, k, \ell) = \sum_{k \leq t \leq \ell} \binom{\ell}{t} p^t (1 - p)^{(\ell - t)}$ est la probabilité cumulée d'obtenir au moins k fois parmi ℓ un événement de probabilité p dans un schéma de Bernoulli – ce qui est très approximatif, les occurrences des k-mers n'étant pas indépendantes. La probabilité p d'obtenir, sur un k-mot, une affectation particulière `V` est définie en fonction du nombre de k-mots stockés dans l'index. La p-valeur de la recombinaison V-J est la somme des p-valeurs des deux zones V et J. Enfin, cette p-valeur est multipliée par le nombre de reads traités pour donner une e-valeur qui peut servir à un seuillage.

Entrée : une chaîne d'affectation $a = a_1 a_2 \dots a_\ell$
$\delta \leftarrow 0$
$\delta_{\max} \leftarrow 0$
$i \leftarrow 0$
$j \leftarrow 0$

Pour chaque t de 1 à ℓ
Invariant : $\delta = |a[0, t-1]|_{\mathtt{V}} - |a[0, t-1]|_{\mathtt{J}}$
si $a_t = \mathtt{V}$, alors $\delta \leftarrow \delta + 1$
si $a_t = \mathtt{J}$, alors $\delta \leftarrow \delta - 1$

si $\delta > \delta_{\max}$, alors $\delta_{\max} \leftarrow \delta$ et $i \leftarrow t$
si $\delta = \delta_{\max}$, alors $j \leftarrow t$
Fin Pour



Sortie : renvoyer i et j

FIGURE 3.3 – *Recherche en temps $O(\ell)$ du meilleur palier (i, j) dans lequel se trouve la jonction V-J. L'algorithme effectivement implémenté dans Vidjil (*`affectanalyser.cpp`*) calcule aussi, au sein du même parcours linéaire, les valeurs $|a[0,i]|_{\mathtt{V}}$, $|a[0,i]|_{\mathtt{J}}$, $|a[j,\ell]|_{\mathtt{V}}$ et $|a[j,\ell]|_{\mathtt{J}}$ qui servent à d'autres filtres et à l'estimation de e-valeur.*

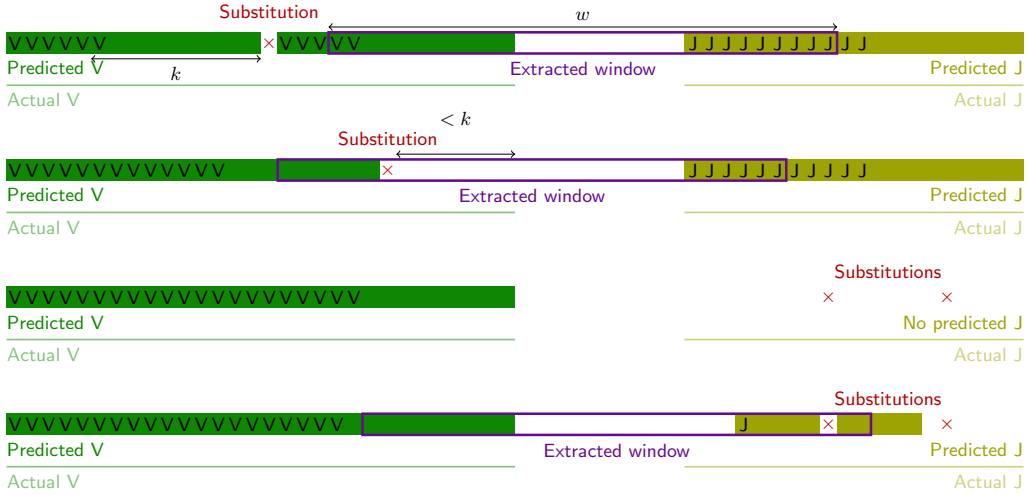


FIGURE 3.4 – *Localisation approximative de recombinaison VJ à base de k-mots [12]. Le but est de prédire une « fenêtre » à partir des k-mots V et J, centrée autant que possible sur la jonction réelle V-J. (Haut.) S'il n'y a pas de substitution à distance de k de la jonction réelle, la prédiction est parfaite. (Milieu haut.) Lorsqu'il y a une substitution proche de la jonction, la fenêtre prédite est au plus à k positions de la jonction réelle. (Milieu bas.) Quand il y a trop de substitutions, la prédiction échoue. (Bas.) En réalité, les k-mots peuvent être espacés, en suivant des modèles de graines avec des jokers [111]. Avec un joker, on peut ainsi détecter plus de recombinaisons, et limiter l'erreur sur la position de la fenêtre à $k/2$ positions lorsqu'il y a une substitution.*

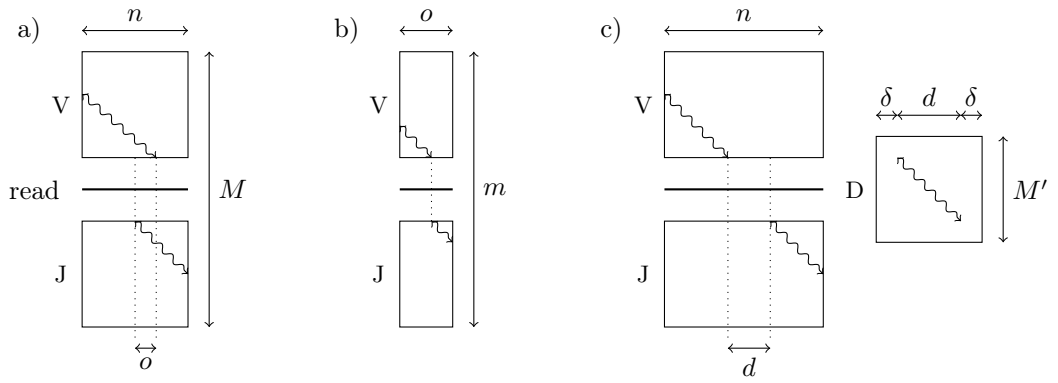


FIGURE 3.5 – *Détermination de la recombinaison VDJ d'une séquence par programmation dynamique (phase 2). Une présentation de la comparaison de séquences par programmation dynamique peut se trouver dans [86, chapitre 6], ou, en français, dans [71, chapitre 7]. a) Les meilleurs alignements de la read avec un gène V et un gène J sont recherchés en temps $O(Mn)$, où M est la taille totale des répertoires V et J et n la taille de la read. Dans cette première étape, les segments V et J sont traités de manière indépendante. b) Si les meilleurs alignements trouvés font se chevaucher le segment V et le segment J sur $o \leq n$ positions, le meilleur point de recombinaison est cherché par une autre programmation dynamique en temps $O(mo)$, où m est la somme des tailles des gènes V et J de référence considérés. c) Dans le cas d'une recombinaison VDJ, le meilleur D est recherché par un alignement local en temps $O(M'(d+2\delta))$, où M' est la taille totale du répertoire D et $d + 2\delta \leq n$ la taille de la zone de la read où le D est recherché. Les chevauchements potentiels entre V et D ou entre D et J sont traités de la même manière que précédemment. Au final, l'ensemble des étapes est en temps $O((M + M')n)$.*

1c. Extraction de la fenêtre et regroupement des reads. Si la découpe de la chaîne d'affectation a été estimée pertinente, on extrait une fenêtre, de taille $w = 50$ nt, centrée sur le k-mer débutant au milieu des positions i et j. Toutes les reads partageant exactement la même fenêtre sont alors regroupées dans un clone. Notons que la localisation par k-mots peut être approximative, à quelques nucléotides près (figure 3.4, voir aussi évaluation à la section 4.1). L'essentiel est que la fenêtre contienne suffisamment de matériel spécifique pour ne pas mener à des regroupements illusoires. Une localisation approximative mène à plusieurs clones qui seront regroupés, automatiquement ou manuellement, à la fin de l'algorithme.

Phase 2 : Analyse précise de chaque clone. Lorsque toutes les reads ont été regroupées en clones, une séquence consensus de chaque clone est extraite, là encore sans effectuer d'alignement. Dans chaque read, nous considérons les régions dont tous les k-mers sont présents avec une certaine proportion (par défaut 50%) dans toutes les reads du clone. La séquence consensus est alors la plus grande de ces régions. Elle inclut nécessairement la fenêtre de 50 nt.

La dénomination V(D)J se fait ensuite sur cette séquence, par programmation dynamique, en utilisant des méthodes similaires à celles des logiciels existants (Fig. 3.5). L'ensemble de l'analyse est ainsi très rapide, car, lors de la première phase, aucun alignement n'est réalisé entre les reads et les répertoires de gènes V(D)J. Une évaluation de cet algorithme sur des jeux de données de patients atteints de leucémie est présentée dans la section 4.1.

3.4 Analyse multi-locus et recombinaisons incomplètes

Locus et pseudo-locus. L'algorithme que nous avons proposé s'applique à l'ensemble des locus Ig et TR (voir Fig. 2.5). Les locus menant à des recombinaisons VJ (Igλ, Igκ, TRα, TRγ) s'analysent en sélectionnant les bons répertoires V et J pour construire les index de k-mers correspondants. Les recombinaisons VDJ (IgH, TRβ et TRδ) sont traitées par le même algorithme, en

étant considérées dans un premier temps comme des recombinaisons VJ. La détermination du D se fait uniquement lorsque le V et le J ont été déterminés (Fig. 3.5, c).

Certains lymphocytes possèdent aussi des *recombinaison incomplètes ou exceptionnelles* (comme les recombinaisons D/J en IgH, D2/D3 en TRδ, KDE/Intron, ou bien les recombinaisons mixtes TRα/TRδ). Il est toujours possible d'analyser ces recombinaisons tant qu'elles sont constituées d'une région « gauche » (5') et « droite » (3'). Nous appelons *pseudo-locus* ces régions (qui peuvent être parfois sur des locus différents, comme dans les translocations BCL1/2-IgH).

Recombinaisons exceptionnelles sur le locus TRδ. Une partie des recombinaisons exceptionnelles TRδ sont uniquement V/D ou D/J. Comme les gènes D sont très courts (8 à 37 nucléotides), il peut ne pas y avoir assez de k-mots pour les détecter, en particulier lorsque la recombinaison a impliqué des mutations ou délétions. Nous avons alors inclus des régions avoisinantes aux gènes de référence, qui se retrouvent dans ces recombinaisons exceptionnelles (Fig. 3.6). Le fait d'ajouter ces régions aux gènes de référence correspond donc à la réalité biologique. De plus, ces recombinaisons, pour pouvoir être séquencées, sont récupérées à l'aide d'amorces qui peuvent être présentes dans ces régions avoisinantes. Au final, les séquences représentées sur la figure 3.6 sont correctement analysées par Vidjil, ce qui n'est pas le cas avec les programmes usuels d'analyse (voir 4.1).

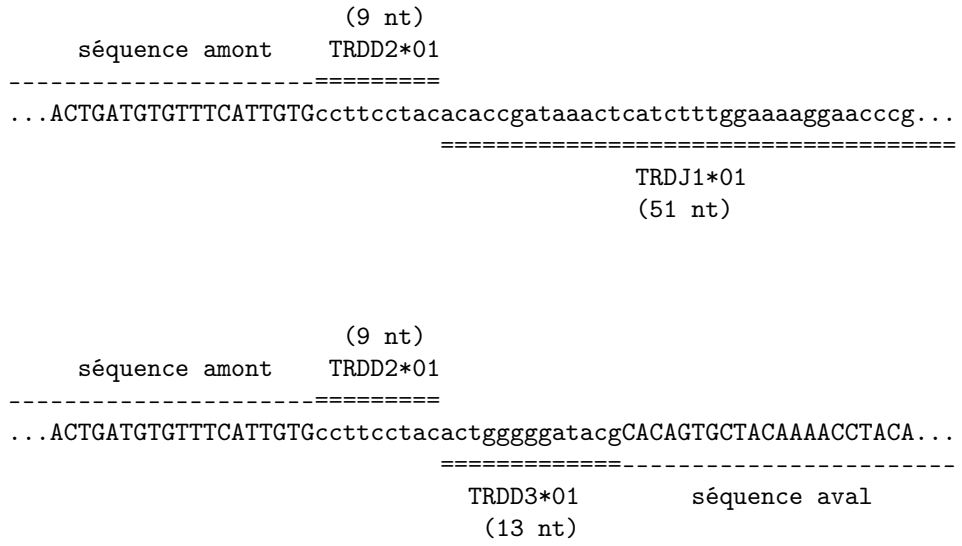


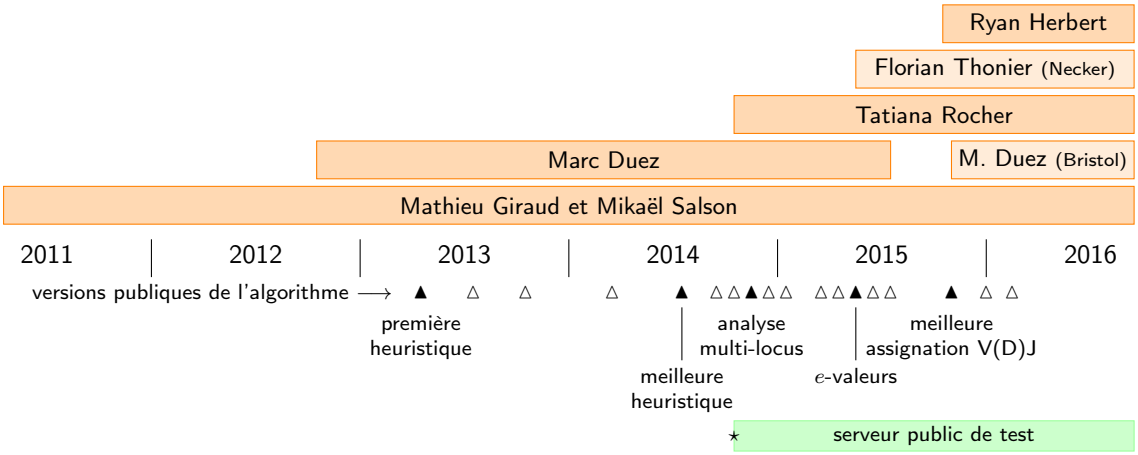
FIGURE 3.6 – *Recombinaisons exceptionnelles dans le locus TRδ. (Haut) Les recombinaisons D2-J conservent la séquence amont du gène D2, alors que cette séquence amont est supprimée dans les recombinaisons complètes VDJ. (Bas) Il existe même des recombinaisons D2-D3, qui incluent aussi la séquence aval du gène D3.*

Sélection du meilleur locus. L'analyse multi-locus contient actuellement 14 locus ou pseudo-locus (voir 4.1). Le locus retenu pour chaque read est celui qui propose une recombinaison avec la p-valeur la plus faible possible. La e-valeur est cette fois-ci calculée en multipliant la p-valeur par le nombre de reads ainsi que par le nombre de locus traités.

4 Développement, évaluation et mise en production de Vidjil

Les premiers contacts avec le laboratoire d'hématologie du CHRU de Lille (à l'époque avec Claude Preudhomme, Christophe Roumier, Nathalie Grardel et Aurélie Caillaut) ont eu lieu en décembre 2010 grâce à Martin Figeac, responsable de la plateforme de séquençage de Lille, lors d'une recherche de collaborateurs sur des thématiques cancer. Mikaël Salson et moi avons abordé ce projet par notre spécialité, *l'analyse de séquences*. Le premier run de séquençage a été effectué fin 2011. À partir de ce moment, nous avons travaillé sur l'analyse haut-débit, toujours en lien avec la plateforme et le laboratoire. Début 2012, à l'occasion d'un projet étudiant de David Chatel, nous commençons un nouveau programme en C++ qui deviendra par la suite Vidjil (section 4.1).

Nous avons progressivement ressenti le besoin de *permettre à des utilisateurs biologistes d'accéder à nos résultats*. À partir de 2013 et du recrutement de Marc Duez, nous avons développé une application web conviviale (`www.vidjil.org`, section 4.2) couplée à un serveur (section 4.3). Au début pensés comme une interface à Vidjil, l'application web et le serveur ont progressivement évolué pour être plus génériques et s'adapter au travail quotidien des hématologues ou immunologistes en situation clinique ou de recherche. À notre connaissance, Vidjil est la seule plateforme permettant aujourd'hui un travail autonome sur des données de Rep-Seq.



Désormais rejoints par Tatiana Rocher, Florian Thonier et Ryan Herbert, nous continuons le développement sur les trois composants – algorithme, application web et serveur. Antonin Carette et François Dubiez, étudiants en licence et en master, ont aussi participé ponctuellement à ces développements.

Nous soumettons à PLoS Computational Biology un court article décrivant l'ensemble de la plateforme [21].

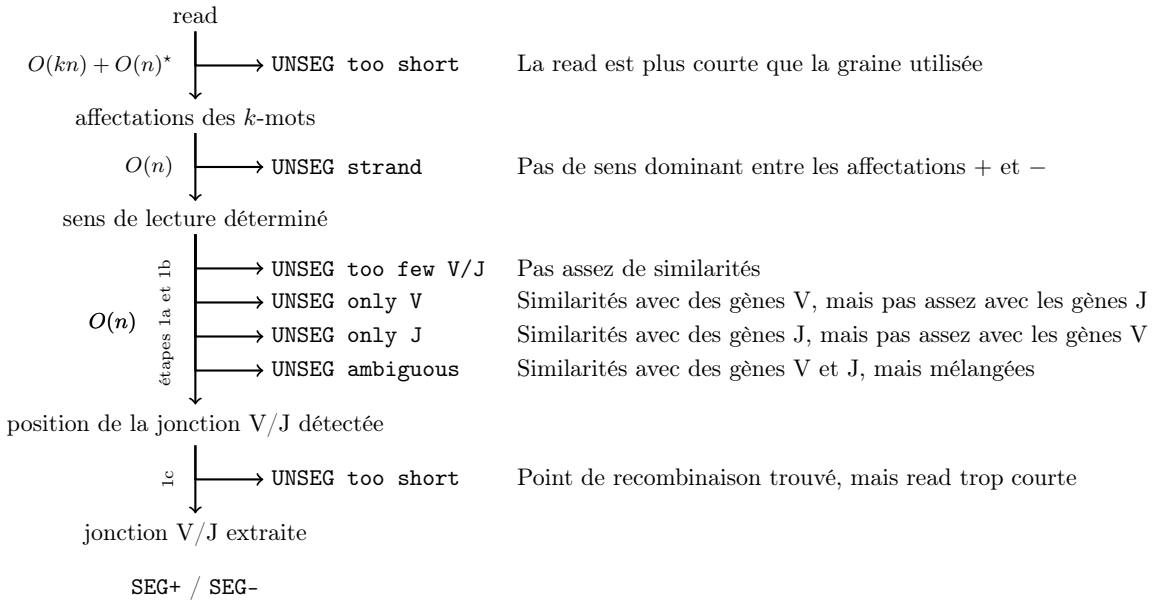


FIGURE 4.1 – *Traitement d'une read dans Vidjil (phase 1, voir pages 22 et suivantes) et raisons de non-segmentation. Les raisons* `too few V/J`, `only V`, `only J` *et* `ambiguous` *correspondent à des échecs des filtres et tests statistiques utilisés sur la découpe de la chaîne d'affectation (étape 1b, voir page 23). Toutes ces étapes sont réalisées en temps linéaire par rapport à la taille de la read (n) et ne comprennent aucun alignement complet avec les gènes V/J de référence. Cependant, la toute première étape demande d'extraire tous les k-mers ($O(k)$ opérations par position dans le cas de graines espacées) puis de faire $O(n)$ accès mémoire a priori non contigus dans l'index de tous les k-mers.*

4.1 L'analyse haut-débit

La méthode décrite au chapitre précédent est implémentée dans Vidjil en C++. Le programme prend en entrée un fichier de séquences (`.fasta`, `.fastq`, ou fichiers compressés `.gz`) et regroupe les séquences avec des recombinaisons V(D)J en clones. Pour cela, chaque read est traitée en temps linéaire pour extraire (ou non) une fenêtre centrée sur un CDR3 (voir pages 22 et suivantes). L'analyse détaillée, avec désignation des gènes V(D)J, n'est faite que dans une deuxième phase. On peut de plus limiter cette deuxième phase aux 10, 100 ou 1000 premiers clones et ainsi accélérer encore le traitement tout en obtenant rapidement les données pertinentes sur les clones dominants. Dans ce cas, on peut même s'intéresser à un clone connu qui ne serait pas majoritaire en spécifiant sa fenêtre.

Multi-locus. La première version de Vidjil analysait des recombinaisons complètes TRγ et IgH. Nous avons progressivement étendu cette analyse pour reconnaître l'ensemble des locus recombinés humains (voir Fig. 2.5), notamment en employant les techniques présentées à la p. 25. Il est aussi possible de configurer le programme pour rechercher d'autres recombinaisons. Les gènes de référence V/J et les autres régions génomiques sont récupérés depuis IMGT/GENE-DB [92] et par des requêtes à GenBank (`ncbi.nlm.nih.gov`).

Évaluation de l'efficacité et de la vitesse. Nous avons comparé Vidjil aux logiciels reconnus comme référence, à savoir IMGT/V-QUEST et IMGT/HighV-QUEST, mais aussi IgBlast. Sans recourir aux alignements, la phase 1 de Vidjil prédit une localisation de cette fenêtre suffisamment précise pour faire un bon regroupement : sur un jeu de données provenant d'un patient du CHRU de Lille, la localisation est à moins de 10 nt du centre prédit par IMGT dans 97% des cas, et à

	Distance	Vidjil – IgBlast	IgBlast – HighV-QUEST	Vidjil – HighV-QUEST
Diag	0 .. 4	26993 (94.4%)	22177 (87.6%)	21138 (90.0%)
	5 .. 9	903 (3.2%)	2646 (10.5%)	2140 (9.1%)
	10 .. 14	284 (1.0%)	211 (0.8%)	153 (0.7%)
	15 .. 19	158 (0.6%)	108 (0.4%)	23 (0.1%)
	≥ 20	0	0	347 (1.5%)

	Distance	Vidjil – IgBlast	IgBlast – HighV-QUEST	Vidjil – HighV-QUEST
Scale-10^{-5}	0 .. 4	25817 (96.1%)	21066 (88.4%)	20154 (88.1%)
	5 .. 9	855 (3.2%)	2450 (10.3%)	2328 (10.2%)
	10 .. 14	149 (0.6%)	289 (1.2%)	354 (1.5%)
	15 .. 19	25 (0.1%)	12 (0.1%)	29 (0.1%)
	≥ 20	0	53 (0.2%)	0

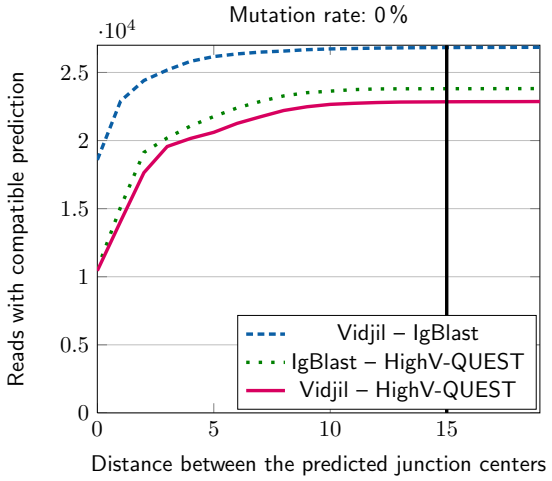


FIGURE 4.2 – *Comparaison de la localisation du centre de la fenêtre par la phase 1 de Vidjil avec la localisation du centre du CDR3 détecté par IgBlast et IMGT/HighV-QUEST, sur les 100 000 premières reads d'un échantillon de diagnostic de leucémie aiguë (Diag, haut) et sur une dilution (Scale-10^{-5}, bas et courbe) [12]. Dans plus de 99% des cas, la localisation prédite par Vidjil est à une distance inférieure à 15 nucléotides de la prédiction des autres programmes. Ce taux reste au-dessus de 99% même en ajoutant 6% de mutations supplémentaires (données non reproduites, voir [12]).*

moins de 15 nt dans 99% des cas (Fig. 4.2), ce qui fait que la fenêtre de 50 nt englobe largement une zone spécifique à chaque CDR3 [12].

La phase 2 de Vidjil analyse correctement plus de séquences que IMGT/V-QUEST et IgBlast en ce qui concerne la désignation V(D)J (Tab. 4.3) [19]. Notons cependant que ces outils, en particulier IMGT/V-QUEST, ont de nombreuses autres fonctionnalités qui ne rentrent pas actuellement dans les objectifs de Vidjil. À notre connaissance, Vidjil est aujourd'hui le seul logiciel publié à analyser certaines recombinaisons incomplètes (D2/D3 en TRδ, KDE/Intron en Igκ, recombinaisons mixtes TRδ/TRα, voir section 3.4). Cela dit, ces analyses sont abordables, même par une programmation dynamique : il est à prévoir que d'autres logiciels réussissent prochainement à traiter ces recombinaisons.

Enfin, les *temps d'analyse* sont compatibles avec un travail quotidien de recherche ou clinique. Sur un portable standard, la version 2015.07 de l'algorithme analyse 1 Gbp (un milliard de nucléotides) en moins de 5 minutes pour un locus. Les locus multiples et incomplets demandent pour l'instant plusieurs itérations de l'algorithme et sont jusqu'à 10 fois plus lents (voir section 6.1).

Ces temps restent très inférieurs aux temps demandés par les solutions faisant une analyse complète de chaque read. IMGT/HighV-QUEST (version 3.2.31, lancée à partir du serveur web

Concordance entre les trois logiciels		
Même désignation		58 (46 %)
Différences négligeables		21 (17 %)
Différences significatives		46 (37 %)
IMGT/V-QUEST et IgBlast		
Désignation correcte (par au moins un des deux logiciels)		77 (62 %)
Pas de désignation ou mauvaise désignation		48 (38 %)
TRδD3	29	
TRδD3-TRαJ29	1	
TRαJ29	1	
KDE (locus Igκ)	17	
Vidjil		
Désignation correcte		113 (90 %)
Pas de désignation ou mauvaise désignation		12 (10 %)
Mauvaise détection du gène central	2	
TRδJ2 au lieu de TRαJ29	2	
Mauvaise détection de la jonction	7	
Pas de désignation	1	

TABLE 4.3 – *Évaluation des désignations V(D)J faites par IMGT/V-QUEST (version 3.3.2), IgBlast (version 1.4.0) et la phase 2 de Vidjil (versions 2015.04 et 2015.05), sur les séquences de 125 clones identifiés au diagnostic et à la rechute chez 34 patients atteints de leucémie aiguë (janvier-mars 2015) [19]. Les 46 séquences dont la désignation était discordante entre au moins deux des logiciels ont été analysées manuellement pour déterminer la bonne solution et classifier les erreurs des logiciels. Cette évaluation, conduite par Yann Ferret, montre que Vidjil trouve une désignation correcte à 90% des clones. De plus, certaines des erreurs de Vidjil ont été corrigées dans les versions ultérieures.*

IMGT sur des machines puissantes) met ainsi plus d'une heure pour analyser 100 000 reads. Nous effectuerons dans les prochains mois une comparaison plus complète incluant les outils présentés page 21, certains n'ayant été publiés qu'en 2015.

4.2 L'application web

L'application web de Vidjil, ou *browser*, permet d'explorer des populations de lymphocytes. Conçue en 2013 pour afficher les résultats de l'algorithme, elle a aujourd'hui évolué vers un environnement de travail complet pour l'hématologue, en clinique ou en recherche. Marc Duez a conçu l'essentiel de cette application, en Javascript avec jQuery et d3.js, et depuis tous les développeurs sur Vidjil y contribuent, en particulier Ryan Herbert, ingénieur recruté pour deux ans grâce au support d'une ADT Inria.

En entrée, l'application web prend des données analysées par l'algorithme de Vidjil (ou par d'autres pipelines) sous forme d'un fichier Json. Ce fichier contient diverses informations sur les clones principaux, avec en particulier leur abondance et leur assignation V(D)J. L'application web est composée de différentes *vues* : une liste de clones, une représentation des clones en grille ou en histogramme, une liste de séquences, et, lorsqu'il y a plusieurs points de suivi, un graphe au cours du temps. Sur la grille, chaque clone est représenté par une bulle. La collision des bulles est gérées par une méthode utilisant un quad-tree [39]. Les axes de la grille, représentant par défaut les gènes V et J, sont configurables pour réaliser différentes statistiques sur la population. Un clic sur un clone n'importe où dans l'application sélectionne le clone dans toutes les vues, en particulier en affichant sa séquence et l'alignant éventuellement contre d'autres séquences.

D'autres logiciels font aussi de la visualisation de résultats d'analyse Rep-Seq, tels que VdjViz [184] ou ARReST/Interrogate [183]. L'originalité de l'application web de Vidjil est de permettre d'explorer en détail certains clones et de se rapprocher le plus possible de la pratique clinique hé-

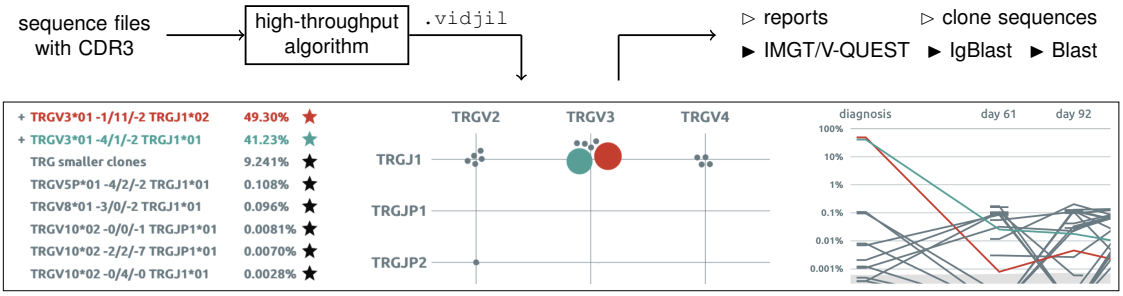


FIGURE 4.4 – *Architecture de la plateforme Vidjil. Les clones détectés par l'algorithme sont stockés dans un fichier json* `.vidjil`. *L'application web charge ce fichier et affiche les clones en liste (gauche), sur une grille (milieu), et sur un graphe lorsqu'il y a plusieurs points (droite). Les données peuvent être exportées ou envoyées vers d'autres logiciels d'analyse. Lorsque l'application web de Vidjil est couplée à un serveur avec une base de données patients, l'utilisateur charge directement ses séquences depuis l'application web pour lancer l'algorithme. Il peut ensuite sauvegarder ses annotations dans la base de données [21].*

matologique, tout en étant facilement lié au programme d'analyse. L'ensemble de l'application web est pensée pour être en interaction avec l'utilisateur qui peut annoter, étiqueter ou corriger certains clones et les transmettre à d'autres programmes d'analyse : IMGT/V-QUEST [110], IgBlast [163] et Blast [59]. L'utilisateur peut aussi éditer manuellement la désignation V(D)J pré-calculée. Une opération particulière est la *fusion* de clones similaires décidée par l'utilisateur qui souhaite regrouper des séquences avec quelques différences qui pourraient venir d'imprécisions technologiques (PCR, séquençage) ou d'hypermutations biologiques. Pour aider cette décision, l'application web propose un outil d'alignement multiple et une représentation 2D de l'ensemble des clones reposant sur une distance d'alignement (algorithme tSNE [116]). L'application web permet enfin de normaliser des séries de données et de générer des rapports pour les dossiers patients.

4.3 Le serveur et la base de données d'échantillons et de patients

Les utilisateurs peuvent directement entrer leurs séquences dans l'application web. En arrière-plan, un serveur avec une base de données d'échantillons et de patients, développé en web2py, fait le lien avec l'algorithme et gère une file d'attente. Après authentification, les utilisateurs peuvent ainsi créer des fiches de patients, lancer Vidjil (ou prochainement d'autres programmes) et enregistrer leurs modifications. Ce serveur, développé à partir de début 2014 par Marc Duez, rend la plateforme autonome et a permis d'attirer de nombreux utilisateurs (voir section 5.1).

Via le serveur, il est possible de visualiser en même temps les échantillons du même patient, que ce soit en suivi de MRD ou pour l'étude de réponse immunologique, mais aussi d'afficher des échantillons de patients différents ou provenant de différents programmes ou paramétrages. Les fichiers de résultats sont privés mais peuvent être partagés avec d'autres utilisateurs ou rendus publics, éventuellement après anonymisation des données personnelles.

4.4 Développement, intégration continue, mise en production

Vidjil est un logiciel stable. En particulier grâce à Jean-Frédéric Berthelot, nous avons mis progressivement en place des bonnes pratiques de développement logiciel pour accompagner la mise en production du logiciel, son ouverture à plusieurs utilisateurs, et l'agrandissement de notre équipe (6 développeurs, plus de 4 000 commits lors des derniers 24 mois). Nous utilisons en interne

```
>TRGV5*01 4/AG/5 TRGJP2*01  [TRG]
TTGATACTACGAAATCTAATTGAAAATGATTCTGGGGTCTATTACTGTGCCACCTGGGAagAGTGATTGGATCAAGACGTTTGCAAA
AGGGACTAGGCTCATAGTAACTTCGCCTGGTAA

>TRBV7-2*02 0//3 TRBJ2-3*01  [TRB]
CCAAGGCAACAGTGCACCAGACAAATCAGGGCTGCCCAGTGATCGCTTCTCTGCAGAGAGGACTGGGGAATCCGTCTCCACTCTGAC
GATCCAGCGCACACAGCAGGAGGACTCGGCCGTGTATCTCTGTGCCAGCAGCTTTAGCACAGATACGCAGTATTTTGGCCCAGGCAC
CCGGCTGACAGTGCTCGGTAAGCGGG

>TRDD2*01 0/4/3 TRDD3 0/2/3 TRDJ1*01  [TRD+] TODO
AgcgggtggtgatggcaaagtgccaaggaaagggaaaaaggaagaagagggtttttatactgatgTGTTTCATTGTGCCTTCCTACG
TGAGGGGGATACGCCCCGATAAACTCATCTTTGGAAAAGGAACCCGTGTGACTGTGGAAC

# The D/J junction can be seen as 2//7, 3//6, or 4//5
>IGHV3-48*01 0/AA/6 IGHD5-12*01 3//6 IGHJ4*02  [IGH]
TGTGAAGGGCCGATTCACCATCTCCAGAGACAATGCCAAGAACTCACTGTATCTGCAAATGAACAGCCTGAGAGCCGAGGACACGGC
TGTGTATTACTGTGCGAGAGAaaATAGTGGCTACGAttTGACTACTGGGGCCAGGGAACCCTGGTCACCGTCTCCTCAGTT

# or TRDV2*03
>TRDV2*01 0/C/1 TRDD3*01 4/CCGCCT/0 TRAJ29*01  [TRA+D]
ATACCGAGAAAAGGACATCTATGGCCCTGGTTTCAAAGACAATTTCCAAGGTGACATTGATATTGCAAAGAACCTGGCTGTACTTAA
GATACTTGCACCATCAGAGAGAGATGAAGGGTCTTACTACTGTGCCTGTGACACCCCTGGGGGAccgcctGGAATTCAGGAAACACA
CCTCTTGTCTTTGGAAAGGGCACAAGACTTTCTGTGATTGCAAGTAAGTGTTTCTAGC
```

FIGURE 4.5 – *Quelques séquences manuellement annotées (*`should-vdj.fa`*). Ces tests incluent des séquences relativement simples à analyser, d'autres pour lesquelles Vidjil ne donne pas encore de réponse satisfaisante (cas marqué* `TODO`*, ici un double D), ainsi que des séquences ambiguës (recombinaison* `IGHV3-48*01 0/AA/6 IGHD5-12*01 3//6 IGHJ4*02`*, voir Fig. 2.4). Une syntaxe pour spécifier formellement les ambiguïtés sera prochainement définie.*

un gestionnaire de tâches pour suivre et hiérarchiser les évolutions à faire au logiciel (depuis début 2014, environ 900 tâches créées dont 500 réalisées).

Nous faisons particulièrement attention à la qualité du code et à la documentation utilisateur et développeur. Dans une démarche d'intégration continue (Jenkins) et de releases régulières, nous avons ajouté systématiquement des tests à Vidjil. Plus de 1 200 tests, unitaires et fonctionnels, visent les trois composants, algorithme, application web et serveur.

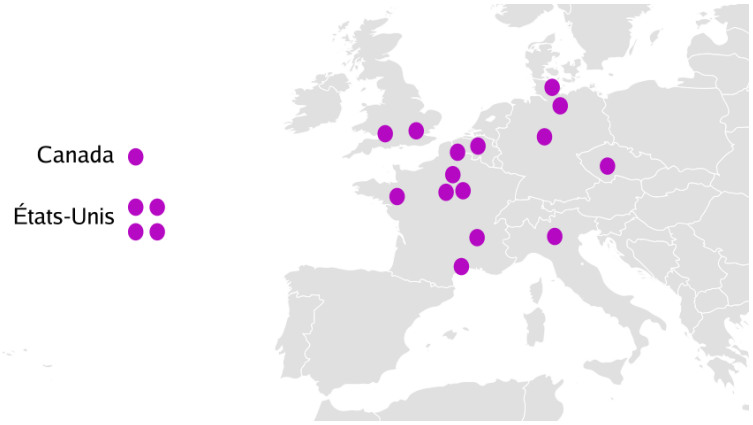
En particulier, pour les tests fonctionnels de l'algorithme, nous avons rassemblé une collection d'une centaine de séquences manuellement annotées (`should-vdj.fa`) pour tester la dénomination V(D)J de la phase 2, mais aussi la détermination du locus au cours de la phase 1 (Fig. 4.5). Les séquences sont de difficultés différentes. Ces désignations ont été vérifiées à la main, éventuellement avec d'autres outils bioinformatiques, et ont été fournies par Yann Ferret et Aurélie Caillault (CHRU Lille), Florian Thonier (Inserm, Paris Necker) et d'autres développeurs de Vidjil.

Déploiement et hébergement du serveur. Nous avons ouvert à l'automne 2014 un serveur de test (`app.vidjil.org`), et le serveur est en cours d'installation à l'hôpital de Lille et dans d'autres endroits.

Vu l'efficacité de l'algorithme, installer le serveur ne nécessite pas de grosses ressources, mais surtout de l'espace disque pour stocker les fichiers originaux de séquences. Notre serveur de test actuel utilise deux processeurs que l'on trouve même sur des portables grand-public (Intel(R) Core(TM) i5-2400 CPU avec 16 Go de RAM).

5 Vidjil : usages, analyse de données et résultats

Vidjil a tout d'abord été conçu pour nos partenaires de l'hôpital de Lille, où il est maintenant utilisé en situation de routine au moment du diagnostic des leucémies aiguës (section 5.2) et testé au cours du suivi (section 5.3). Au cours de l'année 2014, nous avons commencé à avoir des liens avec d'autres laboratoires. Nous avons ouvert un serveur public de test en octobre 2014, et Vidjil est depuis lors utilisé de manière régulière par des laboratoires en France et dans le monde (section 5.1). Vidjil a été en particulier utilisé par le laboratoire de Prague pour une étude sur la diversité du répertoire (section 5.4).



Quelques utilisateurs réguliers de Vidjil au 1er janvier 2016.

La première publication avec nos collègues lillois dans BMC Genomics comprend à la fois de la méthodologie bioinformatique et l'application en hématologie [12]. Les publications ou soumissions suivantes, que ce soit celles de Lille [13, 19, 22] ou des autres groupes (dans lesquelles nous ne sommes pas forcément co-auteurs) [18, 179, 175], portent avant tout sur des thématiques hématologiques ou immunologiques. Le fait que Vidjil soit utilisé directement par des biologistes dans leur pratique clinique ou de recherche est, pour nous, un indicateur clé de son succès.

5.1 Utilisations du serveur de test `app.vidjil.org`

En 14 mois, plus de 40 laboratoires provenant de 11 pays différents ont testé Vidjil et ont ainsi soumis 1600 jobs totalisant environ 5 milliards de séquences (moyenne de 3 millions de reads par job). 95% de ces travaux soumis ont été traités en moins de 10 minutes.

Stratégies de séquençage. Les données proviennent majoritairement des séquenceurs Illumina Mi-Seq et Ion Torrent. Une partie des données sont des données « paired-end », pour lesquelles chaque fragment d'ADN est séquencé par les deux extrémités, engendrant deux reads. Ces données peuvent être pré-traitées par des logiciels tels que PEAR [171] ou pRESTO [169]. Les données proviennent d'approches de séquençage à haut-débit basées sur la PCR (comme les amorces BIOMED2), séquençage d'ARN (RNA-Seq) ou bien sur la capture (voir section 2.3).

Des PCR avec des amorces spécifiques aux gènes V(D)J conduisent généralement à plus de 95% de séquences analysées. Au contraire, la capture avec de nombreuses sondes et le séquençage de l'ARN total donnent généralement un ensemble de séquences avec seulement une faible partie de recombinaisons V(D)J (moins de 0,1%).

Quelques utilisations de Vidjil.

— Plusieurs laboratoires travaillent sur les leucémies aiguës, en fort lien avec la clinique. Les travaux des laboratoires d'hématologie de Lille et de Prague sont détaillés dans les prochaines pages. À Bristol, Marc Duez met désormais en place d'un pipeline de diagnostic et de suivi des leucémies aiguës à l'échelle du Royaume-Uni (voir page 41). D'autres laboratoires testent Vidjil pour l'étude des leucémies (Rennes, Montpellier, Bergame).

— À Londres (UCL), des tests sont en cours depuis début 2015 sur des stratégies de capture (M. Hubank, J. Bartram) pour déceler des recombinaisons particulières.

— À Paris, au laboratoire d'hématologie de Necker (E. Macintyre), plusieurs projets sont menés : étude sur la normalité des CDR3, étude de populations de lymphocytes T chez des patients sains en fonction de leur localisation dans le thymus. Pour cela, le laboratoire finance le contrat de F. Thonier, ingénieur qui travaille aussi partiellement sur Vidjil (voir page 41). D'autres projets sur la capture sont aussi prévus.

— Plusieurs utilisateurs mènent des projets de RNA-Seq et se servent de Vidjil pour analyser ou filtrer les recombinaisons immunologiques au milieu d'autres données, tels que l'hôpital de Lyon (S. Huet), l'Institut Gustave Roussy (Villejuif), ou l'université McGill (Montréal, Canada).

— Vidjil est aussi utilisé pour des données d'autres organismes, comme à Göttingen (Allemagne) dans des études sur le répertoire de la souris et du rat [179, 175].

Nous avons d'autres utilisateurs réguliers (voir carte page précédente) dont nous ne connaissons pas l'objet des recherches. Enfin, certains laboratoires se servent directement de Vidjil en ligne de commande. En septembre 2015, nous avons envoyé un sondage aux utilisateurs connus (`www.vidjil.org/survey`), en particulier pour définir les prochaines priorités. Nous avons obtenu des réponses de 17 laboratoires provenant de 8 pays. Une partie de ces utilisateurs se retrouveront à l'occasion du workshop que nous organisons en mars 2016 (voir section 6.2) pour échanger sur les protocoles, les utilisations de Vidjil, et discuter d'évolutions futures.

5.2 Diagnostic des leucémies aiguës lymphoblastiques (Lille)

Notre collaboration avec le laboratoire d'hématologie du CHRU de Lille (Claude Preudhomme, Nathalie Grardel, Aurélie Caillault, Yann Ferret, Nicolas Duployez) et la plateforme de séquençage (Martin Figeac, Céline Villenet, Shéhérazade Sebda) a débuté fin 2010 (voir historique page 27). Depuis 2012, le laboratoire et la plateforme ont fait plusieurs tests [13] combinant le NGS avec une utilisation de Vidjil, pour arriver à un protocole utilisant l'Ion Torrent sur 3 jours et demi (Tab. 5.1). C'est pour répondre à leur besoins d'utilisation hospitalière que nous avons progressivement développé et déployé l'application web couplée à la base de données de patients. La phase d'analyse bioinformatique avec Vidjil est ainsi réalisée aujourd'hui en autonomie par les hématologues.

Jour 1 (8h)
Préparation administrative
Préparation et réalisation des 5 PCRs
Jour 2 (7-8h)
Pool des 5 PCRs, barcodage, purifications
Préparation et lancement de l'OT2
Jour 3 (4h)
Fin de l'OT2, préparation et lancement du run PGM
Jour 4 (2h)
Récupération des données, nettoyage du PGM
Transfert des données et analyse Vidjil

TABLE 5.1 – *Résumé du protocole mis en place à Lille par A. Caillaut (ingénieur biologiste) et Y. Ferret (interne en biologie médicale, pharmacien) pour l'analyse des échantillons de diagnostic et de rechute avec le séquenceur IonTorrent PGM [19].*

Diagnostic en routine. Depuis le 1er janvier 2015, ce protocole est testé en routine par Aurélie Caillault, Yann Ferret et Shéhérazade Sebda, directement dans le laboratoire d'hématologie. Ce protocole concerne pour l'instant les échantillons de diagnostic (et de rechute) qui vont donner lieu à un suivi MRD : l'objectif principal est de détecter le plus de marqueurs possibles pour le suivi futur de la maladie. Il n'a pas encore remplacé la méthode traditionnelle, mais il est fait en parallèle dans le but de, un jour, la remplacer. Sur l'ensemble de l'année 2015, les échantillons de 215 nouveaux patients atteints d'une leucémie aiguë (LAL) ont été pris en charge par le laboratoire d'hématologie de Lille – le laboratoire analysant les échantillons de patients de Nord-Pas-de-Calais–Picardie, mais aussi ceux de Rhône-Alpes et de Provence-Alpes-Côte-d'Azur. Tous ces patients ne rentrent pas dans le cadre du suivi MRD. Au final, 129 patients (112 pédiatriques, 17 adultes) ont été suivis par Rep-Seq avec analyse Vidjil.

Résultats. Une étude détaillée des échantillons analysés au premier trimestre (janvier – mars 2015, 34 patients, 34 diagnostics + 2 rechutes), menée dans le cadre du M2 de Yann Ferret, est sous presse dans *British Journal of Haematology* [19]. Les visualisations des analyses de ces échantillons sont consultables publiquement (`www.vidjil.org/bjh-2016`). À court terme, les bénéfices obtenus sont moins d'échec de séquençage (14 % en Rep-Seq contre 34 % par la méthode traditionnelle). Sans le NGS, trois patients n'auraient pas pu bénéficier d'un suivi de MRD faute d'identification de clone majoritaire. De plus, on détecte d'emblée plusieurs marqueurs permettant la MRD (les protocoles européens recommandent de suivre au moins 2 marqueurs par patient). Même si d'autres méthodes et logiciels Rep-Seq auraient pu analyser ces données, l'utilisation de Vidjil a eu un certain nombre d'avantages décisifs :

— L'analyse multi-locus de Vidjil est précise et complète. Les désignations V(D)J proposées par Vidjil ont été vérifiées et comparées aux résultats d'IgBlast et IMGT/V-QUEST. Celles de Vidjil sont en général plus conformes à l'analyse manuelle des séquences, et permettent de

détecter plus de recombinaisons, principalement parce que nous analysons aussi des recombinaisons incomplètes ou exceptionnelles (Tab. 4.3, au chapitre précédent) ;

— L'ensemble de la plateforme Vidjil, via l'application web couplée à la base de données de patients, a permis une utilisation simple et efficace par les hématologues. Les échantillons des 129 patients ont ainsi été analysés en autonomie par les hématologues, y compris pour la partie bioinformatique (téléchargement des données, lancement de Vidjil, post-analyse et annotations manuelles, sauvegarde, impression des rapports).

Coûts et perspectives. Ce protocole exige une expertise technique et humaine différente des techniques traditionnelles. Cependant, à moyens équivalents, l'analyse de la réception de l'échantillon à la validation biologique est plus courte que par le protocole classique. Les coûts de fonctionnement pour ce protocole, incluant le séquençage comme les autres étapes, sont estimés aujourd'hui entre 70 € et 120 € par échantillon, en fonction du nombre d'échantillons séquencés simultanément (20 à 8). Le coût dominant reste l'acquisition des séquenceurs.

Les amorces de suivi sont désormais réalisées à partir des clones identifiés par NGS. D'autres hôpitaux (Rennes, Bruxelles) testent maintenant ce protocole. En 2016, l'hôpital de Lille poursuit des analyses Rep-Seq et Vijdil pour les patients atteints de leucémie aiguë (LAL) et cherche à étendre le diagnostic aux leucémies chroniques (LLC, locus IgH avec mesure des hypermutations).

5.3 Suivi des leucémies aiguës lymphoblastiques (Lille)

À terme, le but est de réaliser aussi le suivi des LAL en Rep-Seq/Vidjil. Nos résultats préliminaires portent sur le suivi de 11 patients sur les locus TRγ et/ou IgH et concernent des échantillons prélevés entre 2011 et 2014. Nos collègues du CHRU, en particulier Nathalie Grardel, ont pu suivre les évolutions de la maladie, y compris sur plusieurs clones, certains émergeant après le diagnostic (Fig. 5.2) :

— Le patient 0013, initialement à risque standard (AR1 du protocole EORTC-58081, voir section 2.2), a été réévalué à haut risque (VHR) après une insuffisance de réponse aux premiers traitements (induction). L'analyse Rep-Seq montre que les deux clones principaux au diagnostic ont une évolution non parallèle.

— Le patient 0064 a eu deux rechutes aux jours 615 (après diagnostic) et 837. Les clones principaux détectés en TRγ (`TRGV3*01 -0/6/-16 TRGJP1*01`) et en IgH (`IGHV6-1*01 -27/19/-4 IGHJ4*02`) ont des évolutions parallèles : ces séquences proviennent probablement de la même population de cellules. En TRγ, la deuxième rechute, au jour 837, comporte un autre clone à environ 20% des reads (`TRGV2*02 -0/2/-0 TRGJP1*01`). Ce clone était présent depuis le diagnostic et avait eu une augmentation dès le jour 403. Malheureusement ce jeune patient est décédé 4 ans après le diagnostic.

— Le patient 0010 a eu une rechute (jour 413) puis une allogreffe de moelle osseuse au jour 486. Les deux derniers points avant la greffe (jours 445 et 475) ont été relevés après administration des traitements préparatoires à la greffe (NECTAR, Nelarabine, Etoposide et Cyclophosphamide). Les techniques conventionnelles ne détectaient pas le clone du diagnostic dans les prélèvements des jours 81, 445 et 475, tandis que l'étude Rep-Seq les détecte à un très faible niveau (4 à 11 reads). Deux ans après la greffe, ce patient n'a pas rechuté.

Ces résultats sont détaillés dans un article en cours de préparation [22] et sont pour l'instant descriptifs, obtenus a posteriori. Ils devraient être approfondis avant de devenir utilisables sur des études prospectives. Est-ce possible de détecter, avant rechute, un nouveau clone émergeant pour mieux adapter le traitement ? Ces questions sont complexes et soulèvent des défis cliniques, biologiques et algorithmiques.

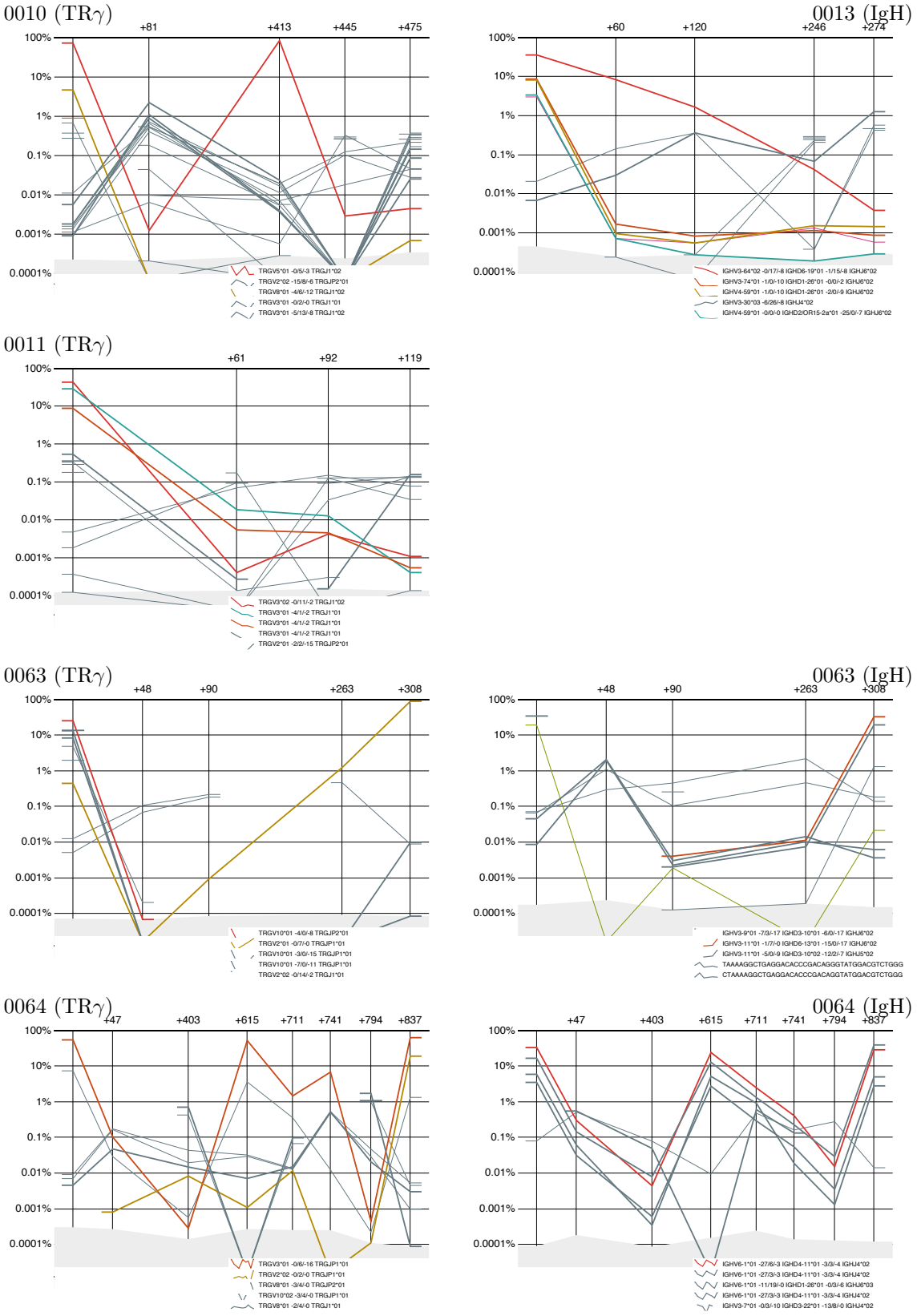


FIGURE 5.2 – *(Haut.) Suivi des patients 0010, 0011 en TRγ (gauche), et du patient 0013 en IgH (droite). (Bas.) Suivi des patients 0063 et 0064, simultanément en TRγ (gauche) et IgH (droite). Ces courbes ont été obtenues après inspection manuelle des résultats de Vidjil, éventuellement avec regroupement additionnel [22].*

Dans tous les cas, la quantification est ici fondamentale, et nécessite d'être étalonnée. À Lille, les étalons témoins sont réalisés par pool d'échantillons de patients malades dont le clone majoritaire est connu, mais qui demanderaient à être mieux calibrés. D'autres laboratoires utilisent des *plasmides* (séquences artificielles). Le groupe de travail EuroClonality-NGS travaille actuellement sur l'évaluation des biais de séquençage pour la quantification et devrait publier et diffuser de tels étalons (voir section 6.2).

5.4 Estimation de la diversité du répertoire (Prague)

Le laboratoire d'hématologie de Prague (Univerzita Karlova), dans une étude rétrospective sur 210 échantillons de 76 patients atteints de leucémie aiguë (LAL), a utilisé Vidjil pour quantifier la diversité du répertoire [18]. Lors d'un diagnostic, le répertoire est très peu divers, même en dehors du clone principal. Même en cas de réussite des traitements, la reconstruction du répertoire prend du temps, au minimum plusieurs semaines.

En collaboration avec Michaela Kotrova, nous avons ainsi quantifié la diversité par la valeur $\rho_{c/r}$, le rapport entre le nombre de clones rendus par Vidjil et le nombre de reads analysées. Dans une situation sans erreurs de PCR ou de séquençage, cette valeur vaut entre 1 (chaque read provient d'un clone différent) et quasiment 0 ($1/n$, un seul clone rassemblant toutes les reads). Les erreurs de PCR ou de séquençage font baisser artificiellement cette valeur, et, dans ce cas, la taille de la fenêtre a aussi une influence. On peut supposer ces erreurs constantes pour un protocole donné avec un séquenceur donné. Notons que $\rho_{c/r}$ permet de différencier des situations très diverses à même taux de MRD. Par exemple, le clone majoritaire peut être à 1%, mais un $\rho_{c/r}$ faible signifiera que les 99% restants sont tout de même concentrés sur quelques clones, alors qu'un $\rho_{c/r}$ élevé, idéalement proche de 1, indiquera qu'un répertoire diversifié s'est reconstruit derrière le clone à 1%.

Le but est de stratifier les patients le mieux possible, afin de leur proposer un traitement adapté à leur situation (voir section 2.2). La valeur $\rho_{c/r}$ au jour 35 après le diagnostic apparaît ainsi comme un très bon moyen de stratification (Fig. 5.3) [18]. Ces résultats demandent à être confirmés, mais témoignent déjà d'un changement radical de pensée : de telles mesures sur l'ensemble de la population étaient complètement impossibles avec les techniques habituelles. Le Rep-Seq ne permet donc pas seulement de mieux mesurer certaines valeurs connues, mais aussi de proposer de nouvelles métriques. Trouver des métriques encore plus pertinentes pour décrire, de manière agrégée, la population de lymphocytes est une piste de recherche intéressante (voir section 6.1).

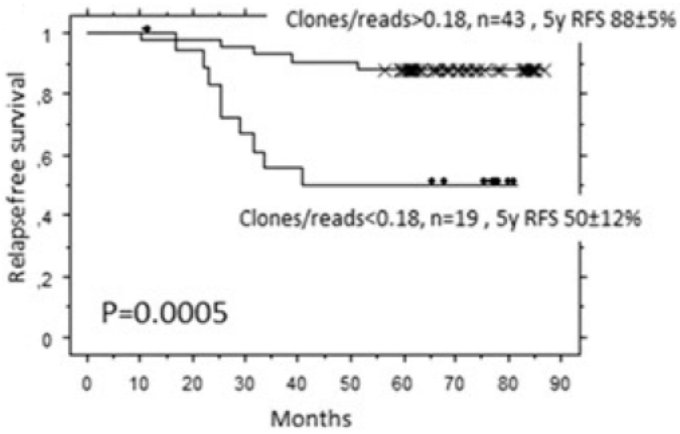


FIGURE 5.3 – *Stratification de patients atteints de LAL en fonction de la diversité au jour 35 après le diagnostic évaluée par $\rho_{c/r}$ [18]. Les patients ayant un $\rho_{c/r}$ supérieur à 0.18 ont un répertoire immunologique mieux reconstitué et une meilleure évolution de leur maladie (88 ± 5 % de survie à 5 ans sans rechute).*

6 Perspectives

Vidjil est un logiciel en constante évolution. Avec Mikaël Salson et Tatiana Rocher, je continue le travail algorithmique. Nous avons toujours des défis théoriques à résoudre qui amélioreraient l'efficacité et la sensibilité de notre programme (section 6.1). Nous désirons aussi répondre de mieux en mieux aux besoins de nos utilisateurs. Nous souhaitons faire fructifier les compétences en hématologie et immunologie acquises depuis 4 ans et développer notre communauté d'utilisateurs et nos collaborations. Cela passe par la poursuite de nos efforts de développement et de diffusion, en particulier par le travail de Ryan Herbert, Marc Duez et Florian Thonier (section 6.2).

6.1 Algorithmique des recombinaisons V(D)J

Comparaison des méthodes et des logiciels. De nombreuses méthodes et logiciels pour les études Rep-Seq ont été publiés en 2014 et 2015 (voir section 3.2). Pour l'instant, nous nous sommes d'abord comparés aux outils d'IMGT, vus comme la référence. Nous comptons réaliser une évaluation plus complète de ces méthodes et logiciels, en essayant de distinguer les principes algorithmiques et les implémentations, que cela soit sur la qualité des résultats ou sur les temps d'exécution.

Analyse haut-débit plus complète. Nous souhaitons pouvoir analyser plus finement certains aspects des recombinaisons. C'est d'abord le cas des CDR3. Si nous ne prétendons pas faire une analyse aussi complète que les outils d'IMGT (auquel nous nous lions, voir ci-dessous), quelques informations clés (longueur et fonctionnalité) pourraient tout de même être analysées par Vidjil/C++. Nous souhaitons aussi proposer des statistiques plus détaillées sur le répertoire, et proposer de nouvelles mesures de diversité, au delà du $\rho_{c/r}$ déjà utilisé par Prague (section 5.4). Nous avons déjà implémenté le calcul d'indices de diversité (Shannon, Simpson) utilisés dans d'autres logiciels de Rep-Seq. Nous cherchons à définir et calculer d'autres métriques globales décrivant la richesse du répertoire immunologique, par exemple concernant des statistiques sur les nucléotides insérés ou supprimés ou le taux de mutations hypersomatiques, utile au suivi des leucémies chroniques (LLC).

Certaines de ces informations plus fines peuvent être calculées dans la phase 2, sur chaque clone. Cependant, nous aimerions aussi pouvoir remonter certaines de ces analyses dans la phase 1, pour avoir l'information au niveau de toutes les reads – et pas seulement pour les 100 ou 1000 premiers clones analysés. En particulier, les nouvelles mesures de diversité telles que $\rho_{c/r}$ devraient pouvoir s'appuyer sur une analyse aussi complète que possible des « petits » clones. Notre défi est donc *d'étendre les analyses sur chaque read, tout en conservant l'efficacité de Vidjil,* c'est-à-dire en maintenant un traitement globalement linéaire sur chaque séquence.

Analyse optimisée de données multi-locus. Mikaël Salson et moi travaillons sur deux points liés au cœur de l'heuristique :

— *Analyse simultanée de plusieurs locus.* Le traitement de p locus se fait en temps $O(pkn)$, en itérant l'extraction de fenêtre pour chaque locus. Cependant, dans les $O(kn)$ opérations de l'heuristique, la première étape pour obtenir l'affection des k-mots est un goulot d'étranglement car elle demande $O(kn)$ opérations et surtout $O(n)$ accès mémoire a priori *non contigus* (Fig. 4.1). Nous sommes en train de transformer cette étape pour la rendre en temps $O(n)$ pour tous les locus par l'utilisation d'un automate dérivé de l'automate d'Aho-Corasick [40]. À cette occasion, nous souhaitons pouvoir intégrer des k-mots avec différentes valeurs de k dans le même index, pour reconnaître au mieux les zones V et J, ce qui permettra une meilleure sensibilité sur certains locus.

— *Optimisation semi-automatique des paramètres de graines.* Les k-mots sont extraits suivant des graines espacées [111]. Nous avions fait « manuellement » le choix de k et des graines pour les locus TRγ et IgH dans notre première étude [12]. Maintenant que nous traitons de nombreuses recombinaisons (14 locus ou pseudo-locus dans la version 2016.02), nous souhaitons mettre en place une méthode semi-automatique pour optimiser ces paramètres, en prenant en compte la distance entre gènes de référence et éventuellement en spécifiant certains k-mots interdits car trop ambigus.

Indexation, compression et mesure de populations avec des recombinaisons V(D)J. Vidjil se contente d'un traitement globalement linéaire sur chaque read. Peut-on aller encore plus loin et considérer directement l'ensemble des reads, et faire des requêtes en temps presque constant sur une structure qui rassemblerait toutes les reads ?

Tatiana Rocher, dans sa thèse débutée fin 2014 que je co-encadre avec Mikaël Salson et Jean-Stéphane Varré, essaie de proposer une *structure d'indexation spécifique aux recombinaisons V(D)J.* Serait-il possible d'indexer les reads, ou au moins les séquences consensus des clones, pour permettre de répondre rapidement à des requêtes statistiques (comme les gènes V, D, J utilisés, ou les métriques globales telles que $\rho_{c/r}$) ou de comparaisons (entre plusieurs échantillons d'un même patient voire entre patients différents) ? Une telle indexation permettrait de répondre à des questions hématologiques et immunologiques, en particulier sur la comparaison et l'évolution de répertoires.

Tatiana Rocher s'inspire de structures et méthodes existantes : LZ-77 et LZ-78 [44, 45], arbres et tables de suffixes, transformée de Burrows-Wheeler [109]. Ses travaux intéressent aussi le consortium EuroClonality-NGS (voir ci-dessous) qui souhaite disposer d'une *base de données des clones* : comment stocker l'ensemble, ou au moins, les 1000 clones les plus abondants de tous les patients traités dans un grand nombre de centres, et pouvoir faire des requêtes statistiques ou comparatives ?

Plus généralement, Mikaël Salson, dans son projet de recherche, s'intéresse au lien entre compression et indexation. *Comment compresser au mieux des données de séquençage tout en les indexant ?* Mikaël est en contact avec les équipes Genome-Scale algorithmics (université de Helsinki) et Reinert lab (université libre de Berlin). Il souhaite visiter début 2017 ces équipes pour collaborer sur la détection de recombinaisons inconnues et sur celle de familles de clones. À terme, nous souhaiterions pouvoir retracer l'*évolution des populations lymphocytaires* au cours du temps, et proposer des algorithmes comme des métriques globales permettant d'estimer cette évolution.

6.2 Développement, diffusion et transfert

Développement collaboratif de Vidjil et financements. Depuis 2012, l'Université Lille 1, la Région Nord-Pas-de-Calais et le SIRIC OncoLille contribuent au développement de Vidjil. Désormais, Vidjil est aussi soutenu par une ADT Inria pour 2015 – 2017. Mikaël Salson, Tatiana Rocher et moi continuons à développer les aspects algorithmiques, et contribuons partiellement aux autres composants. En octobre 2015, Ryan Herbert a été recruté sur le soutien de l'ADT pour travailler à plein temps sur l'application web et le serveur, avec les priorités suivantes :

— *Amélioration de la pratique médicale.* Répondre aux attentes d'une utilisation clinique (rapports professionnels, traçabilité des actions, statistiques), évoluer vers une standardisation.

— *Plateforme.* Positionner Vidjil comme le centre d'un écosystème de logiciels et de méthodes d'analyse de clonalité, en particulier en intégrant des logiciels de pré- ou post- processing et en proposant aussi le choix d'autres algorithmes que Vidjil/C++. Cette tâche est en lien avec l'évaluation des différentes solutions Rep-Seq.

— *Diffusion.* Faciliter la diffusion de Vidjil en améliorant l'installation, l'administration et l'utilisation en environnement de production.

Deux de nos partenaires contribuent eux aussi au développement de Vidjil :

— Depuis mai 2015, le laboratoire d'hématologie de l'hôpital Necker, à Paris, dirigé par Elizabeth Macintyre, a recruté un bioinformaticien, Florian Thonier. Son projet est d'identifier statistiquement quelques paramètres montrant la *normalité* d'un CDR3, en utilisant des échantillons de patients sans maladie hématologique connue, ainsi que d'étudier des populations de lymphocytes à divers endroits du thymus. Florian Thonier contribue aussi à l'application web de Vidjil.

— Après avoir passé deux ans et demi au sein de Bonsai, Marc Duez travaille depuis novembre 2015 à l'Université de Bristol (School of Social and Community Medicine et MRC Integrative Epidemiology Unit, John Moppett). Son travail est de mettre en place un pipeline à l'échelle du Royaume-Uni pour le diagnostic et le suivi des leucémies aiguës. En lien avec des laboratoires d'hématologie anglais, il travaille en particulier sur la qualité des reads, leur regroupement, de nouvelles fonctionnalités de visualisation ainsi que sur le serveur et la base de données. Depuis Bristol, Marc continue ainsi à contribuer à Vidjil. Il fait aussi le lien avec Ryan Herbert, nouvel ingénieur dans l'équipe.

Diffusion et fédération d'une communauté d'utilisateurs de Vidjil et du Rep-Seq. Nous accompagnons déjà la diffusion de Vidjil, notamment par un support aux utilisateurs. Depuis un an, nous avons ainsi échangé avec une quinzaine de nos 40 utilisateurs, parfois pour du simple support, mais souvent pour retravailler certaines analyses de données. Certains utilisateurs peuvent devenir de véritables collaborateurs avec lesquels nous co-publions (comme pour le laboratoire de Prague, voir section 5.4). Nous avons pour l'instant surtout travaillé avec des hématologues (diagnostic et suivi de leucémies) et nous cherchons désormais des collaborations en immunologie (étude de la réponse immunitaire mais aussi développement de vaccins).

Pour mieux fédérer cette communauté naissante, nous organisons le 14-15 mars 2016 une rencontre réunissant utilisateurs et développeurs de Vidjil (`www.vidjil.org/workshop-2016`). Ce sera un lieu pour échanger sur les techniques et les résultats, bioinformatiques comme immunologiques. Nous étudions aussi la possibilité d'organiser en 2016 ou 2017 une rencontre plus large entre bioinformaticiens et hématologistes ou immunologues travaillant sur le Rep-Seq lors d'un événement satellite à une grande conférence de bioinformatique.

EuroClonality-NGS. Le consortium ESLHO (European Scientific foundation for Laboratory Hemato Oncology), et plus particulièrement ses divisions EuroClonality (`www.euroclonality.org`) et EuroMRD (`www.euromrd.org`), standardise au niveau européen les diagnostics de clonalité, notamment pour les leucémies. Ses protocoles sont suivis par l'ensemble des hôpitaux européens impliqués dans le suivi des leucémies, et c'est ce consortium qui a été à l'origine des amorces BIOMED-2 [77] (voir section 2.3). Tous ces hôpitaux (dont celui de Lille) participent régulièrement à des contrôles de qualité en aveugle.

Nous allons depuis 2013 aux conférences publiques du consortium. Depuis 2014, nous avons été invités à faire partie du groupe de travail « EuroClonality-NGS », qui a pour but de définir

les nouveaux protocoles ayant recours au séquençage à haut débit. Ces protocoles seront mis en œuvre dans les hôpitaux d'ici quelques années. Ce groupe de travail, débuté en 2013, est composé d'hématologues. Il comprend aussi une équipe de bioinformatique, dirigée par Nikos Darzentas, en République Tchèque, qui développe le logiciel officiel du consortium, ARReST/Interrogate [183]. L'équipe IMGT y est aussi présente.

Nous avons eu 4 rencontres depuis 2014 de ce groupe de travail, où nous présentons régulièrement nos avancées sur Vidjil. Si l'utilisation officielle de Vidjil n'est pas à l'ordre du jour du consortium, ce groupe de travail permet des échanges fructueux pour tous les partenaires. Certaines des idées algorithmiques ou de visualisations proposées dans Vidjil ont ainsi été reprises dans le logiciel du consortium. Des échanges de données seraient aussi possibles, les focus des programmes n'étant pas les mêmes. Enfin, participer au consortium nous permet d'être au contact de nombreux hématologues et de mieux comprendre leurs besoins. Une partie de nos utilisateurs provient ainsi de ce consortium.

Transfert et financement. Dès à présent, le diagnostic de patients atteints de leucémie aigüe peut être facilité par le NGS et Vidjil (voir section 5.2). À moyen et long terme, le suivi devrait être possible, et même amélioré par des nouvelles mesures rendues possibles par la connaissance détaillée du répertoire. Notre premier objectif de transfert est ainsi que, à la fin de l'ADT Inria en 2017, *1 000 patients soient suivis de manière régulière.* Fin 2015, si environ 1 300 jobs ont été soumis sur le serveur, il n'y a que le laboratoire de Lille qui a franchi le pas en systématisant ces analyses, en routine (129 patients en 2015, voir section 5.2).

Le transfert de Vidjil est donc d'abord clinique. Faut-il envisager un autre transfert, avec une valorisation plus économique de ce travail ?

— Nous tenons à ce que Vidjil reste open-source. Un modèle fermé aujourd'hui ralentirait notre croissance. Nous avons pu gagner la confiance de nos utilisateurs, qui sont en grande majorité des laboratoires publics. Un des facteurs d'adhésion de nos utilisateurs est justement cette ouverture et cette liberté académique (plusieurs entreprises proposent des services Rep-Seq groupés, biologie et informatique, sur lesquels les utilisateurs ont peu de prise).

— De l'autre côté, nous avons aussi quelques utilisateurs privés, et certains laboratoires seraient prêts à contribuer au développement de Vidjil. Nous serions intéressés par trouver un moyen de valoriser ces développements, et aussi de réussir à financer du support aux utilisateurs, que nous faisons actuellement mais qui, sur le long terme, devrait être transféré à une autre structure.

Ces deux points ne sont pas si contradictoires. Une possibilité serait de créer une start-up fournissant du *service* (hébergement, analyse, support) autour de Vidjil qui resterait open-source, et une autre serait de proposer un appui et une facturation via une plateforme existante, comme la plateforme Bilille (M. Pupin puis G. Marot). Enfin, les derniers recrutements de Florian Thonier à Necker et Marc Duez à Bristol montrent une troisième voie : nous pouvons chercher à faire financer le développement de Vidjil par nos partenaires et collaborateurs. Dans tous les cas, nous devons réfléchir à un modèle de propriété intellectuelle, peut-être via un accord de consortium.

Deuxième partie

Analyser les partitions, Algomus

Est-ce qu'un ordinateur peut comprendre la musique ?

La musique est complexe, faite de mélodies, de rythmes et d'harmonies structurées dans le temps. La *partition musicale* formalise un ensemble de sons et est l'un des moyens principaux pour transmettre, échanger et préserver les oeuvres musicales en Occident. Analyser des partitions, c'est apporter un éclairage sur leur construction (chapitre 7). Aujourd'hui, les humanités numériques lient les méthodes informatiques au patrimoine culturel et à la recherche en sciences humaines et sociales. Comment les ordinateurs peuvent aider à modéliser les partitions, et idéalement à comprendre la musique ?

J'ai été initié à l'analyse musicale par Maxime Joos, professeur au conservatoire de Lille. Avec Richard Groult et Florence Levé, du laboratoire MIS (Université de Picardie Jules Verne, Amiens), j'ai eu une première publication en 2011 concernant l'analyse du rythme [2]. Cette collaboration s'est accentuée en 2012 avec le début de notre travail sur les fugues [3, 5]. Désormais rejoints par Emmanuel Leguy, Nicolas Guiomard-Kagan, Pierre Allegraud et Sławek Staworko, nous formons l'équipe émergente Algomus, que je dirige.

Dans le champ des humanités numériques, le projet d'Algomus est de mener des recherches en analyse musicale computationelle, c'est-à-dire d'inventer des *méthodes numériques d'analyse de partitions*, en combinant expertise musicologique et méthodes d'algorithmique du texte, de fouille de données et d'apprentissage. L'analyse repose sur un ensemble d'*éléments locaux d'analyse*, dont les motifs, les accords et les enchaînements d'accords ou la texture. L'analyse combine ces éléments pour comprendre la *structure haut-niveau* de la musique (chapitre 8). Enfin, Algomus travaille aussi sur la modélisation et la visualisation de partitions analysées, que ce soit à destination des musiciens, des apprenants, des mélomanes ou du grand public (chapitre 9), et réalise des projets combinant sciences et arts ainsi que des actions de médiation autour de la musique et de l'informatique (qui seront évoqués à la fin du document, au chapitre 11).

Cette partie, plus prospective, présente quelques méthodes existantes et les premiers résultats obtenus par l'équipe. Pour les prochaines années, les objectifs d'Algomus sont à la fois méthodologiques, fondamentaux, en recherche en informatique musicale, et objectifs sociétaux, appliqués et culturels (chapitre 10).

Algomus est soutenu par Sciences et Cultures du Visuel (iCAVS/IrDIVE) et Pictanovo, lors de projets que j'ai portés, ainsi que par un projet de la région Picardie porté par F. Levé.

7 Analyse musicale
Analyse musicale computationnelle

Que comprenons-nous dans une partition musicale ? Qu'elle soit formalisée ou inconsciente, l'analyse musicale est une activité importante du mélomane, de l'auditeur, de l'interprète, du théoricien comme du compositeur (section 7.1). L'analyse concerne à la fois des éléments locaux et globaux, dans un double mouvement d'analyse et de synthèse (section 7.2).

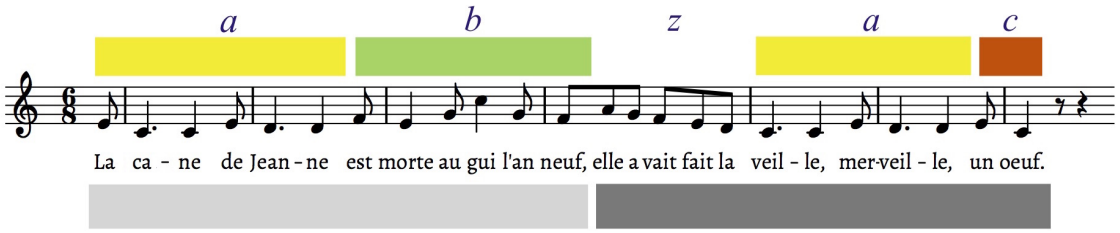


FIGURE 7.1 – *Motifs dans le thème de la* Cane de Jeanne, *de Georges Brassens. Le découpage du bas suit le texte, en deux propositions. Le découpage du haut est une lecture possible des motifs de ce thème. L'ensemble du thème peut se modéliser comme « abac », la partie « ab » étant un* antécédent, *une phrase ouverte, et sa reprise « ac » un* conséquent, *une phrase fermée se terminant par le motif « c » qui joue ici le rôle conclusif d'une* cadence. *Le motif « b » peut être vu comme une extension du début du motif « a ». Les notes « z », bien qu'elles se rattachent au motif « b » qui les précède, se comprennent aussi comme une ligne mélodique se dirigeant vers le retour du motif « a ».*

Du côté informatique, l'analyse musicale computationnelle (CMA) fait partie de la fouille de données musicales (music information retrieval, MIR), un domaine établi depuis une vingtaine d'années. Plusieurs équipes se focalisent ainsi sur l'analyse de partitions, tentant d'expliquer, commenter et générer des données musicales symboliques (section 7.3).

L'équipe émergente Algomus est une collaboration entre les laboratoires CRIStAL (UMR 9189 CNRS, Université de Lille) et MIS (Université de Picardie Jules Verne, Amiens). Je dirige cette équipe naissante qui a pour but de parvenir à des méthodes informatiques et des visualisations facilitant la compréhension globale d'une partition (section 7.4).

Une partie de ce chapitre provient d'un article de revue écrit avec Marc Rigaudière pour Techniques et Sciences Informatique [11]. Le projet d'Algomus (section 7.4) est une réflexion commune à toute l'équipe, réflexion qui sera complétée par la présentation de nos objectifs au chapitre 10.

7.1 Pourquoi analyser des partitions musicales ?

La partition musicale, formalisation de la musique. La musique est tout d'abord une tradition orale, interprétée, transmise et écoutée. Cette tradition se matérialise aujourd'hui par l'échange de fichiers son et leur étude. La musique repose aussi sur une formalisation des sons sous la forme de notes, chacune ayant une hauteur et une durée. La **notation musicale** est une tradition écrite, retranscrivant sous forme de *symboles* un ensemble de notes et d'indications d'interprétation (Fig. 0.1). Elle permet des constructions complexes de mélodies, d'harmonies et de structures. Durant des siècles, la partition a été le moyen principal pour transmettre, échanger et préserver des oeuvres musicales en Occident. Des théoriciens ont formalisé et étudié la musique. La notation musicale, parfois actualisée, est aussi pertinente pour comprendre, analyser et enseigner la musique d'aujourd'hui.

Qui fait de l'analyse musicale ? Analyser, c'est détailler, expliquer, comparer, comprendre. Chacun d'entre nous, musicien ou non, fait de l'analyse plus ou moins inconsciemment à l'écoute d'une musique. Lorsqu'on écoute une chanson ou une œuvre classique, l'analyse peut tout d'abord être *locale* : « le chanteur chante haut », « c'est plus rapide à cet instant », « il y a un climax ici ». L'analyse peut aussi être plus *globale*, faisant référence à la structure de la pièce : « le refrain revient », « c'est une transition instrumentale ». L'analyse peut enfin être *comparative*, s'appuyant sur la connaissance du répertoire d'un artiste, d'un compositeur, d'un style, d'une époque. Pour le mélomane non musicien, analyser de la musique se fait d'abord en suivant son intuition, en utilisant sa culture musicale, même sans utiliser des notions théoriques.

Pour le théoricien de la musique, l'analyse se fait en formalisant, à partir de l'écoute ou de la lecture de la partition, un ensemble d'informations partielles, locales, que nous nommons *éléments d'analyse* dans [11] : tonalité et harmonie, mélodies et thèmes, rythme et métrique, dynamique, instrumentation, texture... À plus grande échelle, l'analyse rassemble ces éléments dans une structure globale, souvent appelée *forme*. Analyser une partition, c'est apporter un éclairage sur ces points de vue locaux et globaux et renouveler son écoute sur une pièce de musique [54, 56, 121]. L'analyse est une activité essentielle de l'interprète (Fig. 7.2), de l'auditeur tout comme du théoricien de la musique. Celui qui joue, entend ou étudie une pièce le fait avec sa propre compréhension de la musique qu'on peut chercher à formaliser [160].

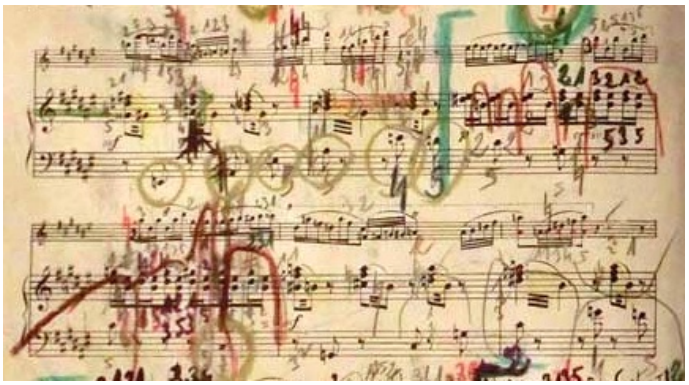


FIGURE 7.2 – *Partition de l'Oiseau de Feu de Stravinksy (1910, transcription pour piano du compositeur) annotée par la pianiste Lydia Jardon. L'interprétation demande des choix techniques (doigtés, répartition des mains) qui s'appuient sur une compréhension approfondie de l'œuvre, en particulier sur une analyse des plans sonores de la partition orchestrale.*

L'analyse peut même *précéder* l'interprétation. Glenn Gould écrivait ainsi [41] :

> *« Deux ou trois semaines avant de jouer cette sonate pour la première fois, je commençai à étudier la partition, et à une semaine du concert, je me mis au piano (cela semble suicidaire, mais c'est pourtant ainsi que j'ai toujours procédé). (...) Ma technique est de passer le plus de temps possible loin de mon piano, ce qui pose certaines difficultés : on a souvent envie de savoir comment cela sonne. Mais un certain idéal analytique (. . .), un certain* achèvement *analytique, quoiqu'il en soit, est théoriquement possible tant que vous n'êtes pas au piano. »*

Enfin, l'analyse a un lien particulier avec la composition. Est-ce que l'analyse est censée retrouver la manière dont le compositeur a construit la pièce ? D'un côté, l'analyse est souvent *anachronique*. Les théories musicales des formes musicales (comme la fugue ou la forme sonate) ont été principalement conçues au cours du XIX^{e} siècle, soit un ou deux siècles après l'émergence de la forme elle-même. L'analyse comparative d'une pièce dans l'œuvre d'un compositeur ou de son époque n'est souvent possible qu'a posteriori. De l'autre, les compositeurs sont influencés par les cadres formels de l'enseignement de la composition (et maintenant de l'analyse) et, plus généralement, par les pratiques de leur temps. Si certains compositeurs préfèrent ne pas détailler leurs méthodes, d'autres, comme Messiaen dans sa *Technique de mon langage musical* [32], se livrent à une auto-analyse de leurs œuvres. Mais, même réalisée par le compositeur, l'analyse n'est qu'un point de vue sur une partition aboutie, et ne reflète pas entièrement le processus compositionnel.

Analyse et sémiologie. Plus généralement, peut-on donner du *sens* à une partition qui semble n'être qu'un ensemble de symboles utilisés par le compositeur pour transmettre sa musique ? C'est tout l'enjeu de la *tripartition* de la sémiologie musicale proposée par Jean-Jacques Nattiez [57, 66] à la suite des travaux de Jean Molino :

> *« Dans la théorie de Molino (...) :*
> *a) une forme symbolique n'est pas l'intermédiaire d'un processus de « communication » qui transmettrait à une audience des significations produites intentionnellement par un auteur,*
> *b) mais le résultat d'un processus complexe de création (le processus poïétique) qui concerne tout autant la forme que le contenu de l'œuvre,*
> *c) et le point de départ d'un processus complexe de perception (processus esthésique) qui reconstruit le message ;*
> *d) les processus poïétiques et esthésiques, enfin, ne coïncident pas nécessairement. »*

Entre ces processus de composition (« processus poïétique ») et de réception (« processus esthésique »), la partition (« forme symbolique ») est alors un « niveau neutre », *qu'il est possible d'étudier de manière autonome* (« analyse immanente ou matérielle »). Certaines analyses musicales dépassent ce niveau neutre (« analyses poïétiques » ou « analyses esthésiques »), allant même jusqu'à l'« analyse de la communication musicale » qui concerne l'ensemble de la communication, du compositeur au récepteur [66].

L'analyse musicale, pratique et discipline. L'analyse comme pratique existe depuis longtemps. Tous les traités de la musique, depuis ceux de Boèce de ou Rameau, ont une composante analytique, expliquant certains aspects de la musique. Au XIX^{e} siècle, l'institutionnalisation de l'enseignement de la musique et de la composition (en particulier au conservatoire de Paris, fondé en 1795) donne lieu à la rédaction de traités détaillant précisément certaines techniques d'écriture. Ainsi, des traités sur la forme sonate [25, 27] ou la fugue [28] sont avant tout pédagogiques, normatifs, mais contiennent aussi une part d'analyse.

L'analyse comme discipline musicologique autonome date elle du milieu du XX^{e} siècle. La classe d'Olivier Messiaen au Conservatoire de Paris, initialement classe d'harmonie, devient en 1947 une

classe d'analyse où O. Messiaen décrypte les partitions du répertoire. Les traités d'analyse peuvent désormais être détachés de contraintes de composition, comme, pour la forme sonate, les ouvrages de Rosen [46] ou d'Hepokoski et Darcy [98]. Aujourd'hui, l'analyse musicale est enseignée dans les parcours de formation musicale initiale et supérieure. C'est une discipline universitaire établie, avec ses journaux *(Music Analysis, Analyse Musicale)*, ses sociétés savantes *(Society for Music Analysis, Society for Music Theory, Société Française d'Analyse Musicale)* et ses congrès *(EuroMAC)*.

7.2 Analyse et synthèse

Tout comme celle du mélomane, l'analyse pratiquée par le théoricien de la musique combine des points de vue locaux et globaux. Dans la revue que nous avons publiée dans *Techniques et Sciences Informatique*, Marc Rigaudière écrit [11] :

> *« On peut analyser une œuvre pour dégager des grandes lignes esthétiques, pour caractériser un style, pour y rechercher une signification (herméneutique), pour observer des éléments systémiques (fonctionnement tonal par exemple), pour y repérer des principes de composition, pour alimenter l'histoire des formes, pour tester la validité d'une méthode, etc. Dans toutes les manifestations de l'analyse, pourtant, un concept est central : celui de forme musicale. Il n'est guère d'analyse qui puisse l'évacuer, dès lors qu'on le prend au sens large, la forme étant alors comprise comme une façon unique d'assembler des éléments constitutifs, aboutissant à une œuvre unique. »*

Si de nombreuses analyses concernent ou s'appuient sur la *forme musicale*, le débat même sur la nature des formes musicales est toujours d'actualité. L'ouvrage « Musical form, Forms, Formenlehre » édité en 2009 par Pieter Bergé [122] fait ainsi dialoguer trois approches contemporaines de la forme musicale : la « théorie des fonctions formelles » de W. Caplin (décomposant la forme en sections et expliquant leur rôles propres et relatifs), la « forme dialogique » de J. Hepokoski (considérant la forme dans son contexte de composition historique), et l'« analyse multivalente » de J. Webster (décrivant la forme selon plusieurs éléments d'analyse).

Tous ces points de vues, complémentaires ou parfois opposés, ont en commun de s'appuyer sur des éléments locaux d'analyse. Comme l'indique Marc Rigaudière [11] :

> *« Quelle que soit la façon de le formaliser, les études modernes de la forme musicale enregistrent toutes l'idée que la forme résulte d'une interaction entre différents éléments. (...) Il n'y a pas une somme de lectures linéaires et stratifiées dont chacune serait limitée à un seul élément, mais une lecture synthétique (souvent guidée par l'écoute ou l'exécution de la pièce) susceptible d'une part de se focaliser tour à tour sur l'un ou l'autre aspect du texte musical et, d'autre part, de rassembler le fruit de cette lecture discontinue en une image globale. C'est bien un point essentiel de l'analyse musicale : ce qu'on nomme analyse est en fait un double mouvement d'analyse suivi d'une synthèse. »*

Cette double approche d'analyse et de synthèse se retrouvera au cœur de nos préoccupations : si la plupart des méthodologies d'analyse informatique se concentrent sur des éléments locaux, Algomus souhaite proposer des analyses haut-niveau, sémantiques, de la structuration d'une pièce (voir ci-dessous, section 7.4, puis section 8.3).

7.3 L'analyse musicale computationnelle

Quel peut être l'apport des méthodes numériques à la musique, et plus particulièrement à l'analyse musicale ?

Sciences, musique et calcul. La science et la musique ont toujours eu des liens forts, tout d'abord pour les aspects d'acoustique et de fréquences des notes. Le Quadrivium fut défini par Boèce au VI^e^ siècle et regroupe les arts libéraux des « sciences des nombres », à savoir l'arithmétique, *la musique*, la géométrie et l'astronomie. L'harmonie des sphères, liant proportions célestes et musique, est une des théorie pythagoriciennes étudiée par Boèce [23]. Mille ans plus tard, notamment à partir des ouvrages de Zarlino [24], les questions de tempérament, c'est-à-dire de division fréquentielle de la gamme, vont être un sujet majeur de débat musical et scientifique du XVI^e^ au XIX^e^ siècle.

À côté des préoccupations acoustiques, le côté plus symbolique et calculatoire de la musique a aussi une longue histoire. Des thématiques de génération musicale ont été développées dans des jeux musicaux aléatoires (avec plusieurs partitions attribuées à Mozart à la fin du XVIII^e^ siècle). En 1843, Ada Lovelace, percevant l'universalité de la machine proposée par Babbage, imaginait déjà que la musique puisse se formaliser au point que l'ordinateur devienne compositeur [26] :

> *« It might act upon other things besides* number*, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine... Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent. »*

Aujourd'hui, il y a de nombreuses interactions entre science, musique, et nouvelles technologies. À Paris, l'Ircam (Institut de Recherche et Coordination Acoustique/Musique), créé par Pierre Boulez en 1969, est un lieu unique d'interaction entre musiciens et chercheurs, lieu d'expérimentation et de création, notamment pour les musiques post-sérielles, spectrales et contemporaines.

Recherche d'informations musicales (MIR). Comment réaliser la prédiction d'Ada Lovelace en modélisant numériquement *« the fundamental relations of pitched sounds in the science of harmony and of musical composition »* ? Aujourd'hui, la communauté « recherche d'informations musicales » (*music information retrieval*, MIR) traite de nombreuses questions concernant à la fois des données audios, des données symboliques (partitions), mais aussi des données d'image ainsi que des métadonnées. Ce champ s'est développé les dernières années, en partie en lien avec des intérêts commerciaux, comme par exemple, ceux liés aux questions de recherche ou de recommandation par le contenu. La communauté publie en particulier dans la conférence majeure du domaine, ISMIR (à l'origine « International Symposium for Music Information Retrieval »), créée en 2000, ainsi que dans les journaux *Computer Music Journal* (depuis 1977) et *Journal of New Music Research* (1972).

Recherche audio, recherche symbolique. Dans les groupes de recherche en MIR, le traitement du signal est dominant. ISMIR réunit chaque année environ 300 participants, dont environ un quart travaillent sur des données symboliques de partitions. Ces études symboliques sont indispensables pour la communauté, car les approches qui ne considèrent que le traitement du signal ne permettent pas de rentrer en détail dans la construction des pièces musicales. J. Stephen Downie, un des leaders du domaine MIR, indiquait ainsi en 2007 [113] :

> *« Many MIR researchers come from a signal processing research discipline. (...) Dealing with music in its audio form requires less music-specific knowledge than dealing with its symbolic forms (i.e., one needs to be able to read and understand music to work with symbolic music representations in a non-trivial manner.) »*

Traiter des données symboliques – principalement des hauteurs et des durées de notes – demande ainsi une expertise musicologique différente des compétences utilisées en traitement du signal.

Encodages de la partition. La partition est elle-même une représentation codée de l'œuvre musicale impliquant certaines conventions. L'édition musicale est loin d'être uniforme, les conventions typographiques variant selon les époques. Il existe plusieurs formats informatiques pour encoder ces partitions, formats n'ayant pas tous le même but ni la même précision.

Les fichiers MIDI, initialement destinés à la communication entre synthétiseurs ou autres instruments électroniques, représentent la hauteur (mesurée en demi-tons) et la durée des notes. Ils peuvent indiquer les différentes voix sur différents *canaux*. De plus, certains fichiers sont correctement *quantifiés*, c'est-à-dire avec des durées qui peuvent être facilement ramenées à des valeurs entières ou rationnelles. D'autres éléments des partitions (expression, dynamique...) peuvent aussi être représentés.

Les fichiers **kern, MusicXML ou MEI sont eux plus proches de la partition (Fig. 7.3). Ils détaillent précisément les hauteurs *diatoniques*. La cinquième note de la figure 7.3, avec une hauteur MIDI de 68, est ainsi un $La^{\flat}$ *(La bémol)*, qui est différente d'un $Sol^{\sharp}$ *(Sol dièse)* qui a pourtant la même hauteur MIDI. Ces deux notes correspondent au même son, mais, dans un contexte tonal, elles ont des *fonctions* différentes, de la même manière que, en français, le « a » et le « à » se prononcent similairement mais ne sont pas interchangeables.

De plus, les fichiers **kern, MusicXML ou MEI spécifient les durées suivant la notation musicale traditionnelle et permettent aussi de coder d'autres éléments de notation. Le format **kern [74], destiné à l'analyse musicologique, permet de plus d'encoder dans certains canaux des résultats d'analyse. Le format MusicXML a lui été conçu comme format d'échange entre éditeurs de partitions. Il permet de décrire fidèlement une représentation visuelle de la partition, y compris certains choix typographiques. Plus récent, le format MEI, produit par le groupe « Music Encoding Initiave », vise à permettre l'échange de représentations musicales, indépendamment de la typographie.

Buts de l'analyse musicale computationnelle. Un ordinateur est-il capable de comprendre une partition comme un théoricien de la musique ? Pour cela, ce que le théoricien lit dans la partition doit être formalisé. Certaines notions s'y prêtent facilement, au moins pour les cas simples : une phrase a généralement un début et une fin, une section peut s'inscrire clairement dans une tonalité, et l'œuvre est généralement structurée par des marqueurs formels comme des cadences (voir section 8.2.3). D'autres notions sont par nature plus difficiles à formaliser (tension, texture, structure...). Toute approche algorithmique doit donc traduire des concepts subjectifs en modélisations précises. À la suite de plusieurs travaux comme [90, 126, 130, 147], nous avons identifié deux grands objectifs des recherches de l'analyse musicale computationnelle [11] :

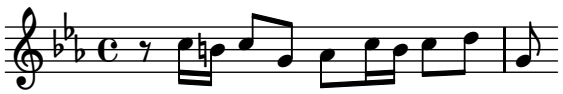
— produire des résultats musicologiques de manière automatisée ou semi-automatisée, éventuellement en bénéficiant de gains en temps de calcul ou en résolution (musicologie systématique) ;

— mener une réflexion sur nos disciplines, en questionnant, du point de vue du musicologue, la formalisation du procédé analytique, et, du point de vue de l'informaticien, la pertinence des modélisations et des algorithmes.

Le premier point, le plus technique, cherche à affiner *les résultats de l'analyse*, en essayant d'analyser des partitions comportant de vrais défis musicologiques. Le second point, sur la *démarche de l'analyse*, est tout aussi important. Célestin Deliège confiait [91, p. 21] :

> *« L'analyse écrite se justifie principalement en fonction de l'explication d'une méthode ou d'une démonstration d'un phénomène spécifique. (...) Sauf dans le cas d'une œuvre nouvelle, quand je lis une analyse qui n'a d'autre but qu'elle-même, je bâille d'ennui. »*

Ces propos s'appliquent aussi à l'analyse musicale par ordinateur : l'intérêt de ces recherches réside autant dans les aspects méthodologiques que dans les résultats produits.



pitch p	72	71	72	67	68	72	71	72	74	67
interval Δp		−1	1	−5	1	4	−1	1	2	−7
onset o	2	3	4	6	8	10	11	12	14	16
length l	1	1	2	2	2	1	1	2	2	2

MIDI

```
2, 2880, Note_on_c,  60, 64
2, 2910, Note_off_c, 60, 64
2, 2910, Note_on_c,  59, 64
2, 2940, Note_off_c, 59, 64
2, 2940, Note_on_c,  60, 64
2, 3000, Note_off_c, 60, 64
2, 3000, Note_on_c,  55, 64
2, 3060, Note_off_c, 55, 64
2, 3060, Note_on_c,  56, 64
2, 3120, Note_off_c, 56, 64

...

3,    0, Note_on_c,  72, 64
3,   30, Note_off_c, 72, 64
3,   30, Note_on_c,  71, 64
3,   60, Note_off_c, 71, 64
3,   60, Note_on_c,  72, 64
3,  120, Note_off_c, 72, 64
3,  120, Note_on_c,  67, 64
3,  180, Note_off_c, 67, 64
3,  180, Note_on_c,  68, 64
3,  240, Note_off_c, 68, 64

...

4,  960, Note_on_c,  79, 64
4,  990, Note_off_c, 79, 64
4,  990, Note_on_c,  78, 64
4, 1020, Note_off_c, 78, 64
4, 1020, Note_on_c,  79, 64
4, 1080, Note_off_c, 79, 64
4, 1080, Note_on_c,  72, 64
4, 1140, Note_off_c, 72, 64
4, 1140, Note_on_c,  75, 64
4, 1200, Note_off_c, 75, 64
```

**kern

```
*clefF4 *clefG2 *clefG2
*k[b-e-a-]      *k[b-e-a-]
        *k[b-e-a-]
*M4/4   *M4/4   *M4/4
*c:     *c:     *c:
*MM72   *MM72   *MM72
=1      =1      =1
1r      8r      1r
.       16cc    .
.       16bn    .
.       8cc     .
.       8g      .
.       8a-     .
.       16cc    .
.       16b     .
.       8cc     .
.       8dd     .
=2      =2      =2
1r      8g      1r
.       16cc    .
.       16bn    .
.       8cc     .
.       8dd     .
.       16f     .
.       16g     .
.       4a-     .
.       16g     .
.       16f     .
=3      =3      =3
1r      16e-    8r
.       16cc    .
.       16bn    16gg
.       16an    16ff#
.       16g     8gg
.       16fn    .
.       16e-    8cc
.       16d     .
.       8c      8ee-
.       8ee-    16gg
```

MEI

```
<scoreDef>
 <staffGrp>
  <staffDef n="1" lines="5" meter.unit="4"
            clef.shape="G" clef.line="2"
            key.sig="3f" meter.count="4" />
  <staffDef n="2" lines="5" meter.unit="4"
            clef.shape="G" clef.line="2"
            key.sig="3f" meter.count="4" />
  <staffDef n="3" lines="5" meter.unit="4"
            clef.shape="F" clef.line="4"
            key.sig="3f" meter.count="4" />
 </staffGrp>
</scoreDef>
<measure n="1">
  <staff n="1">
    <layer n="1">
      <rest dur="1" />
    </layer>
  </staff>
  <staff n="2">
    <layer n="1">
      <rest dur="8" />
      <note dur="16" oct="5" pname="c" />
      <note dur="16" oct="4" pname="b">
        <accid accid="n" />
      </note>
      <note dur="8" oct="5" pname="c" />
      <note dur="8" oct="4" pname="g" />
      <note dur="8" oct="4" pname="a" />
      ...
    </layer>
  </staff>
  <staff n="3">
    <layer n="1">
      <rest dur="1" />
    </layer>
  </staff>
</measure>
```

FIGURE 7.3 – *Encodages informatiques de la partition. (Haut.) Une séquence monophonique de notes (début de la fugue en Do mineur BWV 847 de Jean-Sébastien Bach, voir Fig. 8.11). Une note a deux dimensions : hauteur et durée. Elle peut être décrite par un triplet (p, o, ℓ), où p est la hauteur (pitch) de la note, o son instant de début (onset), et ℓ sa durée. Ici, les onsets et durées sont comptés en double-croches, et donc une croche vaut 2. Les hauteurs sont comptées suivant le standard MIDI (Do au milieu du clavier = 60).*

*(Bas.) Représentation du début de la fugue en Do mineur en MIDI, **kern et MEI, prises à partir d'un fichier disponible sur* `http://kern.humdrum.org`.

((Gauche.)) Les messages MIDI (ici décodés par midicsv, le format étant binaire) ont chacun une piste et un offset. Chaque note se transmet en deux messages (`Note_on` *et* `Note_off`*). Le La$^\flat$ est représenté par le pitch 68, comme un Sol$^\sharp$. La polyphonie résulte de la simple superposition des voix.*

*((Milieu.)) Le format **kern, destiné à l'analyse musicologique, est un encodage texte en deux dimensions : la polyphonie se lit horizontalement tandis que le temps s'écoule verticalement. Toutes les altérations sont explicites (*`a-` *pour La$^\flat$). Des pistes d'analyse peuvent être rajoutées.*

((Droite.)) Le format MEI utilise sur un schéma XML. Les altérations sont ici implicites : comme il y a trois bémols à la clé (`key.sig="3f"`*), tous les La, représentés uniquement par* `pname="a"`*, sont des La$^\flat$. Une altération accidentelle, comme pour le Si bécarre, s'indique par* `accid='n'`.

7.4 Algomus

L'équipe émergente Algomus est une collaboration entre les laboratoires **CRIStAL** (UMR 9189 CNRS, Université de Lille) et **MIS** (Université de Picardie Jules Verne, Amiens). Un ordinateur peut-il comprendre la musique ? Dans le champ des **humanités numériques**, Algomus mène des recherches en analyse musicale computationnelle (CMA, Computational Music Analysis), dans le champ plus général de la recherche d'informations musicales (MIR, Music Information Retrieval). Notre but est de faire des algorithmes analysant des données musicales présentées sous forme de partitions. Les racines méthodologiques d'Algomus sont dans l'**algorithmique du texte**, plus précisément la comparaison de séquences musicales, que ce soit par des modèles de distance, géométriques ou statistiques.

Les données que nous manipulons sont ainsi des partitions musicales, symboliques. Ces données doivent être décodées, organisées, comprises : le cœur de métier d'Algomus est d'apporter de la sémantique aux données musicales, de proposer des descripteurs haut-niveau, des arbres, des grammaires... Pour cela, nous développons des méthodes couplant **analyse** et **synthèse** : *fouille de données* et *apprentissage* sur des informations musicales pour analyser des motifs, des accords et des enchaînements d'accords, de la texture ainsi que d'autres notions musicales, informations que nous synthétisons ensuite pour étudier la structure haut-niveau de la musique (chapitre 8). Nos données d'entrées sont à la fois des *corpus de partitions* (corpus étant de plus en plus disponibles avec des projets de numérisation et d'encodage symbolique), mais aussi des *données d'annotation musicale*, réalisées en collaboration avec des musicologues (section 9.3). Une partie de notre travail est ainsi de créer des jeux d'analyse de référence, jeux sur lesquels les algorithmes de fouille de données musicales peuvent être appris ou évalués.

Enfin, la *modélisation* et la *visualisation* de l'analyse musicale sont primordiales pour faire se comprendre deux mondes : celui des informaticiens et celui des musiciens. Nos représentations des analyses musicales servent également de supports pédagogiques. Elles s'adressent aussi bien au musicologue, au mélomane averti qu'à un plus large public (chapitre 9). Nous sommes ainsi résolument dans une *démarche pluridisciplinaire entre informatique et sciences humaines et sociales.* Nous menons des collaborations avec des musicologues, des professeurs de musique et des artistes, et réalisons des projets combinant sciences et arts ainsi que des actions de médiation autour de la musique et de l'informatique (chapitre 11).

Une approche résolument symbolique. Comme évoqué ci-dessus, le domaine MIR est dominé par l'audio. Certains groupes font simultanément de l'audio et du symbolique, tandis qu'Algomus est focalisé sur le symbolique. Rappelons les propos de J. Stephen Downie : *(« one needs to be able to read and understand music to work with symbolic music representations in a non-trivial manner. »)*. La recherche d'Algomus repose précisément sur expertise musicologique et vise à « comprendre » le plus possible la musique.

Un focus sur l'analyse tonale. Le domaine MIR s'intéresse à tout type de musiques, populaires, savantes, de diverses traditions... et de nombreuses musiques ne sont pas notées sur partitions ! Chez Algomus, nous nous concentrons sur l'analyse d'œuvres de *musique occidentale tonale* (*Western tonal music*, appelée encore *common practice*, qui peut s'étendre en *extended common practice* [146]).

Une *tonalité*, telle que *Do majeur*, est un ensemble de notes ordonnées dans des *gammes*. Les notes décrites dans une partition musicale construisent des mélodies (dimension horizontale, temporelle) mais aussi des accords (dimension verticale, harmonique). Les enchaînements de ces accords, appelés *progressions harmoniques*, structurent l'accompagnement d'un morceau et créent un plan tonal qui alterne entre moments de tension et de relâchement.

La musique tonale couvre une très grande partie de ce que nous entendons : dans la musique dite classique, les périodes baroque, classique, romantique, post-romantique, ainsi que la plupart des musiques folkloriques, de variétés, jazz, pop et rock. À l'inverse, certaines musiques des XX^{e} et XXI^{e} siècles (musique sérielle, spectrale, électronique, concrète) ne sont pas tonales, et demandent d'autres outils (en particulier de traitement du signal) pour être analysées. Même si la musique tonale est écrite, jouée, entendue et théorisée depuis plus de quatre siècles, elle garde de nombreux défis de modélisation musicologique comme informatique.

Positionnement par rapport à la communauté. Un acteur majeur de la musique à Paris et dans le monde est l'Ircam, qui combine création et recherche. Si une grande partie de ses travaux est tournée vers l'audio, plusieurs chercheurs y travaillent, au moins en partie, sur les partitions, en particulier dans l'équipe RepMus (G. Assayag). Nous y avons des liens particuliers avec J. Bresson* et F. Jacquemard*. En France, plusieurs autres groupes travaillent sur les partitions, dont M. Desainte-Catherine (LaBRI, Bordeaux), D. Fober (GRAME, Lyon)*, F. Pachet (Sony CSL, Paris). Avec une partie de ces acteurs, nous participons au réseau MusICAL et nous envisageons de nouveaux projets (voir page 79).

À l'international, les groupes ou les chercheurs de référence travaillant sur les partitions sont en particulier E. Cambouroupoulos (Univ. Thessalonique, Grèce)*, E. Chew (Queen Mary Univ. of London, UK), T. Collins (Leicester, UK), I. Fujinaga (McGill Univ., Canada), M. Goto (AIST, Japon), D. Meredith (Univ. Aalborg, Danemark)*, J. Pablo Bello (New York Univ, US), A. Volk (Univ. Utrecht, Pays-Bas)*, G. Wiggins (Goldsmiths' College, UK). Nous connaissons la plupart de ces groupes, nous retrouvons à diverses occasions (principalement la conférence ISMIR, mais aussi des rencontres en réseau) et collaborons avec certains d'entre eux (*).

L'originalité d'Algomus est son focus sur des questions musicologiques de « haut-niveau », en particulier sur la forme dans le répertoire classique. Ce focus se voit à la fois dans nos compétences internes et externes (collaborations avec des musicologues, avec lesquels nous co-publions) et se ressent sur nos outils méthodologiques (combinaison d'approches explicites et statistiques).

8 Algorithmes d'analyse musicale

L'analyse musicale informatique propose des algorithmes pour expliquer les partitions encodées symboliquement. L'analyse peut porter sur des éléments *locaux* ou *globaux* (section 8.1). En ce qui concerne les éléments locaux, de nombreux travaux ont concerné la recherche et l'extraction de *motifs mélodiques* en comparant des séquences de notes (section 8.2.1). Nous avons travaillé sur ces sujets, ainsi que sur la séparation de voix et la texture (section 8.2.2) et les cadences (section 8.2.3).

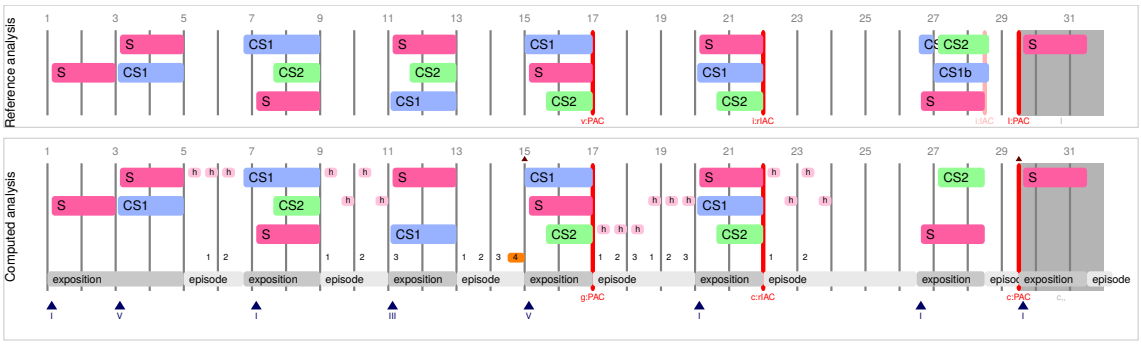


FIGURE 8.1 – *L'analyse de la fugue en Do mineur BWV 847 de Jean-Sébastien Bach révèle des occurrences de motifs (S/CS1/CS2 et motifs incomplets h), des marches harmoniques (1/2/3), des degrés (chiffres romains), des cadences (traits verticaux épais, PAC et IAC) et finalement une structure (exposition/épisode) [15]. L'analyse calculée, en bas, est très proche de l'analyse de référence, en haut.*

Certains travaux, moins nombreux, tentent un mouvement de synthèse pour réaliser une analyse de structure, globale. Avec Richard Groult et Florence Levé, j'ai souhaité relever le défi de l'analyse de formes musicales (section 8.3), en particulier en ce qui concerne les fugues et les formes sonates. Le but est alors de proposer une interprétation analytique de l'ensemble de la pièce en s'appuyant sur les éléments locaux d'analyse et en leur donnant du sens.

Ce chapitre n'a pas pour but de rentrer dans le détail des algorithmes présentés. Il reprend en partie notre revue [11] et notre chapitre [20], et résume quelques contributions techniques d'Algomus – en particulier notre contribution la plus aboutie, celle sur l'analyse des fugues [15]. Enfin, plusieurs exemples musicaux sont utilisés. Je ne présenterai pas ici d'introduction à la musique, mais j'espère que le lecteur non musicien profitera tout de même des figures pour comprendre nos objectifs d'analyse.

8.1 Analyse locale, analyse globale ?

Nous regroupons sous le terme d'*analyse locale* ce qui peut s'appréhender dans un contexte restreint (d'une à quelques mesures), tandis que l'*analyse globale* nécessite d'avoir une vue complète de la partition.

La séparation entre ces concepts est en partie artificielle : de nombreuses méthodes locales – telle que les découvertes de motifs évoquées ci-dessous – s'appuient sur l'ensemble de la partition. En retour, la connaissance d'aspects globaux peut guider une nouvelle analyse de motifs, plus précise : ainsi, dans les fugues (voir ci-dessous, page 63), des occurrences approchées des sujets et contre-sujets ne seront pas traitées de la même manière dans les parties d'exposition et dans les divertissements.

Cependant, les analyses de motif sur l'ensemble d'une partition considèrent généralement la partition comme un ensemble, mais *ni ordonné ni structuré*. Ainsi, ces méthodes se concentrent sur un contexte local et ne font appel au reste de la partition que pour une évaluation statistique – elles donneraient le même résultat si la partition globale était « mélangée ». Le propre des méthodes d'*analyse globale* est d'essayer d'expliciter la structure haut-niveau de la partition, dans une démarche qui tente d'être ordonnée et structurée. Mais cette analyse doit sans cesse utiliser des aspects locaux : les deux points de vue, local et global, sont tous deux indispensables pour obtenir une analyse d'ensemble de la partition.

8.2 Analyse locale

8.2.1 Analyse locale : motifs

Un *motif* est une unité minimale utilisée dans la composition, qui acquiert un statut particulier par son retour exact ou transformé. Sa prégnance, c'est-à-dire sa faculté de marquer la mémoire, lui est conférée par des facteurs agissant le plus souvent en combinaison : intervalles, contour mélodique, rythme, harmonie. Une analyse de motifs complète devrait à la fois détecter des motifs pertinents, estimer leurs bornes, détailler leurs occurrences et décrire leur transformation [11]. De nombreuses personnes ont travaillé sur l'extraction et la recherche de motifs – on pourra se référer à notre revue [11] pour quelques pointeurs. La suite de cette section présente des travaux en lien direct avec nos objectifs, ainsi que quelques-uns de nos résultats.

Occurrences et similarités approchées. Comment comparer plusieurs segments de la partition pour identifier les différentes occurrences d'un motif ? La similarité entre deux séquences de notes monodiques (une seule voix) peut être estimée par l'algorithme de programmation dynamique de Mongeau-Sankoff [60] et ses extensions. Ces algorithmes, inspirés des algorithmes classiques de comparaison de séquences biologiques de Needleman-Wunsch [37] et Smith-Waterman [47], calculent une score ou une distance *d'édition* (voir page 5), c'est-à-dire qu'ils évaluent les opérations nécessaires pour transformer un motif en une de ses variantes (Figs. 8.2 et 8.3).

Les opérations d'édition habituelles sont l'identité, les remplacements, les insertions et suppressions, les consolidations et les fragmentations. Cependant, insérer ou supprimer des notes, ou même substituer un rythme, détruit généralement la perception des temps et de la mesure. Les opérations de fragmentation et de consolidation sont plus appropriées pour transformer un motif en un autre. De plus, l'algorithme de Mongeau-Sankoff utilise le contexte tonal (voir page 52) : le score mesurant la similarité entre deux notes dépend des gammes respectives des deux séquences.

Ce sont par ces algorithmes, très similaires à la bioinformatique, que j'ai découvert le domaine MIR. En 2011, nous avons eu une première publication qui comparait des rythmes avec une adaptation de l'algorithme de Mongeau-Sankoff [2]. Notre algorithme de détection des sujets et

$$S(i,j) = \max \begin{cases} 0 & \text{(début d'une correspondance locale)} \\ S(i-1,j-1) + \delta(x_i, y_j) & \text{(identité ou remplacement d'une note } x_i \text{ en } y_j) \\ S(i,j-1) + \delta(\epsilon, y_j) & \text{(insertion d'une note } y_j) \\ S(i-1,j) + \delta(x_i, \epsilon) & \text{(suppression d'une note } x_i) \\ \\ \max_\ell S(i-1, j-\ell) + \delta(x_i, \{y_{j-\ell+1} ... y_j\}) & \text{(fragmentation de } x_i \text{ en } \ell \text{ notes)} \\ \max_k S(i-k, j-1) + \delta(\{x_{i-k+1} ... x_i\}, y_j) & \text{(consolidation de } k \text{ notes en } y_j) \end{cases}$$

FIGURE 8.2 – *Algorithme de Mongeau-Sankoff et variantes [60]. L'algorithme calcule, par programmation dynamique, la similarité entre deux séquences de notes $x = x_1 \dots x_m$ et $y = y_1 \dots y_n$. La fonction δ est la fonction de score pour chaque type de mutation. La première ligne, en initialisant à 0 l'ensemble des valeurs $S(i,j)$, permet de calculer un alignement local : la valeur $S(i,j)$ est le meilleur score pour un alignement local entre un suffixe de $x_1 \dots x_i$ et un suffixe de $y_1 \dots y_j$. Si la première ligne n'est utilisée que pour la valeur $S(0,0)$, l'algorithme calcule alors le score du meilleur alignement global entre les séquences. Enfin, si elle n'est utilisée que pour les valeurs $S(0,j)$, l'algorithme calcule les scores des meilleurs alignements semi-globaux, c'est-à-dire le score de l'alignement de la séquence $x_1 \dots x_m$ (vue comme un motif) à l'intérieur de la séquence $y_1 \dots y_n$ (vue comme un texte, typiquement une voix complète dans la partition).*

Au lieu de compararer directement les notes ($\delta(x_i, y_j)$), on a souvent intérêt à comparer les intervalles de notes ($\delta(^\Delta x_i, ^\Delta y_j)$, où $^\Delta x_i$ est construit à partir de x_{i-1} et x_i). Dans notre étude sur les fugues, nous avons en plus utilisé une deuxième table $S_f(m,j) = S(m-1, j-1) + \delta_f(^\Delta x_m, ^\Delta y_j)$, où les valeurs $S_f(m,j)$ sont des scores « finalisés » mettant en jeu des scores différents (δ_f) pour la dernière note x_m du motif [3].

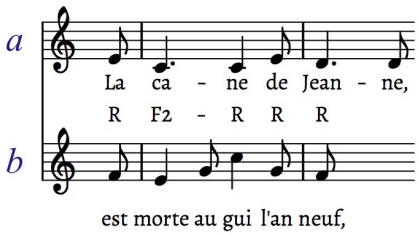


FIGURE 8.3 – *Alignement entre motifs « a » et « b » du thème de* la Cane de Jeanne*, de Georges Brassens (voir Fig. 7.1), obtenu avec une adaptation de l'algorithme de Mongeau-Sankoff. Le motif « b » s'obtient à partir du motif « a » par plusieurs remplacements de notes (R) ainsi que par une fragmentation d'une note (noire pointée) en deux notes (noire et croche) (F2).*

contre-sujets dans les fugues (voir ci-dessous) repose aussi sur une adaptation de l'algorithme de Mongeau-Sankoff [3]. Enfin, concernant l'analyse de thèmes et variations, j'ai proposé, en collaboration avec Emilios Cambouropoulos puis lors du stage de Ken Deguernel, une généralisation des opérations de fragmentation [6], en considérant une variation comme une fragmentation étendue par rapport à une réduction du thème (Fig. 8.4).

Plusieurs auteurs ont proposé d'autres méthodes de recherche approximatives [84, 67], y compris pour des séquences polyphoniques [119]. Certaines méthodes de similarité musicale ne sont pas basées sur la distance d'édition [101, 96], en particulier les méthodes géométriques [108, 81].

Nous pensons que de meilleures opérations d'édition devraient voir le jour, permettant de décrire de manière plus musicale les *transformations* successives d'un même motif en prenant en compte le contexte tonal. On pourrait par exemple modéliser directement des transformations de k notes en ℓ notes, qui donneraient une contribution $\max_{k,\ell} S(i-k, j-\ell) + \delta(\{x_{i-k+1} ... x_i\}, \{y_{j-\ell+1} ... y_j\})$ au score calculé dans la figure 8.2.

Motifs répétés et occurrences. Comment inférer un motif à partir d'une partition ? On peut sélectionner les motifs donnant l'ensemble d'occurrences le plus satisfaisant [69]. Les répétitions des différentes occurrences ont une grande importance sur la perception du début d'un motif [95],

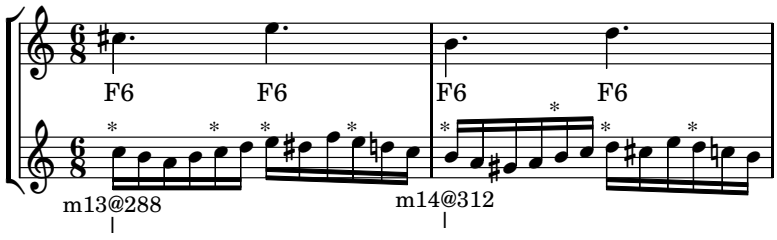


FIGURE 8.4 – *Correspondance entre le thème et la variation mineure de Andante grazioso de la sonate pour piano numéro 11 de Mozart (K 331). La texture de la variation (en bas) peut se voir comme une série de fragmentations en six notes d'une réduction manuelle du thème (en haut). De plus, la même opération de transformation est appliquée sur les deux mesures. Cela permet d'effectuer un test simple de parallélisme, en regardant les notes identiques à la réduction, marquées ici par des étoiles : ces notes sont placées à des positions similaires dans les deux mesures [6].*

comme dans ce que nous avons fait sur la fin des sujets (voir page 63). Certaines techniques de détection de motifs combinent ces aspects locaux et globaux [79].

O. Lartillot a développé une approche originale d'extraction de motifs [75, 105]. La description de ces motifs peut prendre des valeurs dans plusieurs dimensions. Ainsi, un motif peut prendre la forme *ré noire, croche, croche, sol noire*, la hauteur des deux croches n'étant pas spécifiée. Cette approche permet de construire un ensemble de motifs incluant un grand nombre de paramètres (hauteurs relatives ou absolues, diatoniques ou chromatiques, durées, métrique mais aussi potentiellement accentuation, fonctions harmoniques...). Pour être significatif, un motif « fermé » doit se répéter au moins deux fois, et être le plus spécifique possible vis-à-vis de ces occurrences (Fig. 8.5). J'ai débuté en 2014 une collaboration avec O. Lartillot, dans le but de mieux comprendre les possibilités de son approche et de les comparer avec des techniques d'extraction de motifs plus classiques.

8.2.2 Analyse locale : strates polyphoniques, texture

Motifs et strates polyphoniques, séparation de voix. Comment comprendre une partition polyphonique (à plusieurs voix) ? La polyphonie peut parfois se *séparer en voix monophoniques* – et, pour nos analyses les fugues ou les formes sonates (section 8.3), nous partons de fichiers où les différentes voix sont séparées. Sur ce type de polyphonie, Conklin et Bergeron ont proposé d'extraire des *motifs contrapuntiques élémentaires*, c'est-à-dire des motifs présentant localement l'agencement de voix mélodiques distinctes. Ces motifs tiennent compte de la conduite des voix tout comme de relations de consonance et de dissonance [131] (Fig. 8.6).

Les algorithmes de séparation de voix, parmi lesquels [89] et [143], tentent de retrouver ces voix à partir de la polyphonie (Fig. 8.8). De manière générale, cette séparation de voix n'est pas toujours possible ou souhaitable : de nombreuses textures pour piano ou pour d'autres instruments polyphoniques font intervenir des accords avec un nombre de notes variables. Inversement, le jeu d'un instrument monophonique peut faire apparaître plusieurs strates. Pour regrouper horizontalement (au cours du temps) et verticalement (dans l'espace des hauteurs) des *strates* appartenant à un même tissu musical, une étude a proposé d'utiliser un clustering en k plus proches voisins à partir de critères de groupement (Fig. 8.7) [117].

Nicolas Guiomard-Kagan, dans la première partie de sa thèse que je co-encadre avec V. Villain, F. Levé et R. Groult, a travaillé sur ces sujets, pour aboutir à une comparaison unifiée entre les algorithmes de séparation de voix et ceux de regroupement de strates [17]. Pour cela, il utilise et généralise un ensemble de métriques, que ce soit sur les notes, les paires de notes [89] ou en utilisant une mesure d'information mutuelle [114]. Il travaille désormais sur l'amélioration de ces algorithmes.

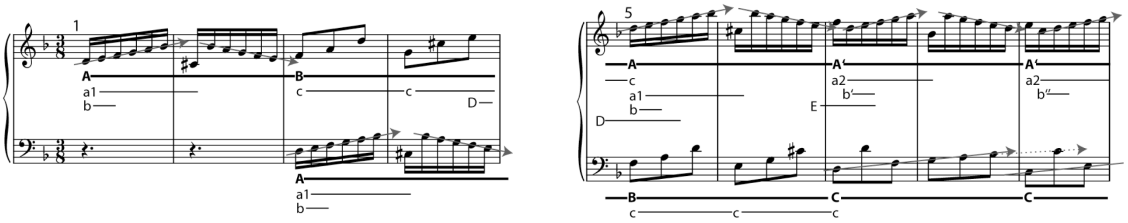


FIGURE 8.5 – *Analyse motivique de l'invention à deux voix BWV 775 de J.-S. Bach [105]. Chacun des motifs est identifié par une lettre, et les motifs sont reliés entre eux par des relations de spécificité. Par exemple, le motif b, « ré mi fa » en doubles-croches débutant sur un temps, se transforme en b' (même motif, mais débutant à contre-temps) puis b'' (« do ré mi », transposition diatonique).*

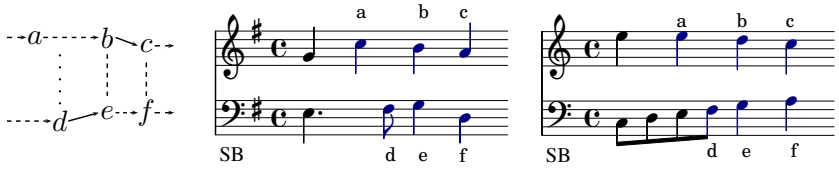


FIGURE 8.6 – *Détection d'un « motif contrapuntique » élémentaire dans un corpus de 185 chorals de J.-S. Bach, et exemple de deux de ses occurrences [131]. Ce motif est « distinctif » du couple SB (soprane/basse), ayant cinq fois plus d'occurrence dans le couple SB que dans les autres ensembles de voix. Le motif est constitué des relation de consonnance (b/e et c/f : unissons, tierces et sixtes majeures et mineures, quinte juste) et de dissonance (a/d : autres intervalles), ainsi que de relations de simultanéité ou de succession temporelle.*

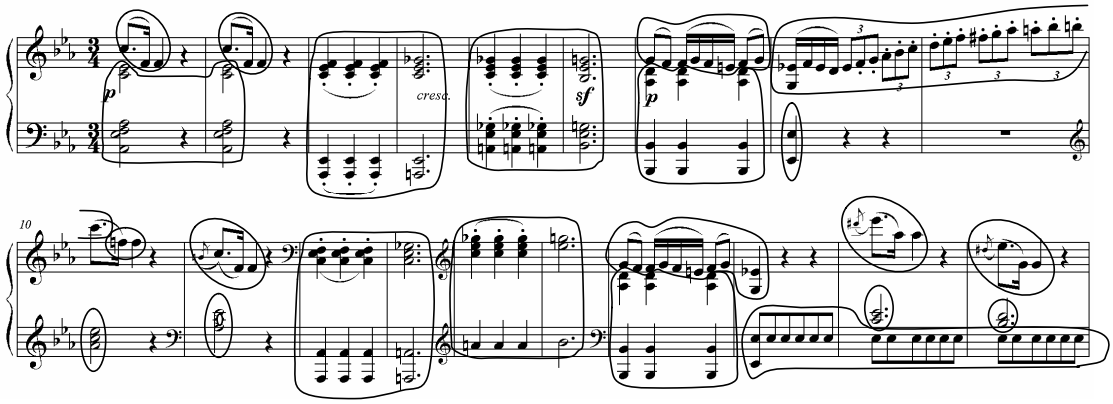


FIGURE 8.7 – *Détection de strates polyphoniques dans l'introduction de la sonate op. 31 no 3 de Beethoven [117]. Cette analyse automatique reprend certains critères de groupement de Deutsch [48, chapitre 6]. Les groupes favorisent des notes synchrones, et, pour des notes successives, celles étant à hauteurs proches. Dans les deux premières mesures, cette détection réussit à séparer la mélodie (haut de la main droite) de l'accompagnement (accords en blanches, répartis sur les deux mains). Dans les deux toutes dernières mesures, l'algorithme propose, avec raison, trois strates, une de mélodie et deux d'accompagnement*

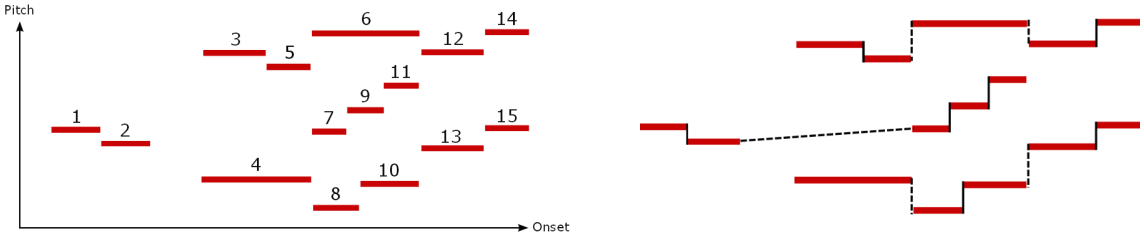


FIGURE 8.8 – *Séparation de voix [17]. À partir d'une polyphonie (gauche), le but des algorithmes de séparation de voix est d'extraire plusieurs voix monophoniques (droite). Trois voix sont ici extraites :* $\{3, 5, 6, 12, 14\}$, $\{1, 2, 7, 9, 11\}$ *et* $\{4, 8, 10, 12, 13\}$. *La séparation complète n'est pas toujours réalisable ni souhaitable : la séparation peut aussi produire des* segments *isolés, comme par exemple* $\{1, 2\}$ *et* $\{7, 9, 11\}$.

Texture. La *texture* est le rendu de la surface musicale obtenu par une combinaison de notes, possiblement jouées par plusieurs instruments. On parle souvent de texture pour décrire les différents *timbres* d'instruments ou de groupe d'instruments (texture de cordes, de cuivres...). La texture s'applique aussi sur des partitions destinées à un seul instrument polyphonique (piano, orgue, guitare...) ou à un ensemble d'instruments de timbres similaires (quatuor de cordes, ensemble de flûtes à bec...). La texture qualifie alors différentes *strates* de notes telles que mélodies ou accompagnements.

On différencie habituellement les textures *homophoniques* (voix rythmiquement similaires) et les textures *polyphoniques* (plusieurs strates, le plus souvent une mélodie avec un accompagnement, mais aussi contrepoint). On peut aussi avoir une combinaison de ces notions, comme par exemple des voix rythmiquement similaires deux à deux (Fig. 8.9). La notion de texture recouvre aussi des aspects de densité (nombre de voix, degré d'intrication des voix par le rythme et les registres). Au contraire de l'harmonie ou de la mélodie, la texture a été rarement modélisée par les musicologues. Nordgren propose de classifier des textures orchestrales selon la répartition verticale des accords, en tenant compte de la présence éventuelle de vides et du registre de ceux-ci, le nombre de doublures ou le nombre d'instruments présents [34].

Du côté de l'informatique musicale, David Huron modélise la texture en considérant deux paramètres, la synchronisation des notes et la présence de mouvements parallèles [58]. Il parvient ainsi à reconnaître quatre types de texture : monophonie, homophonie, polyphonie et hétérophonie, et à classifier certains styles de musique.

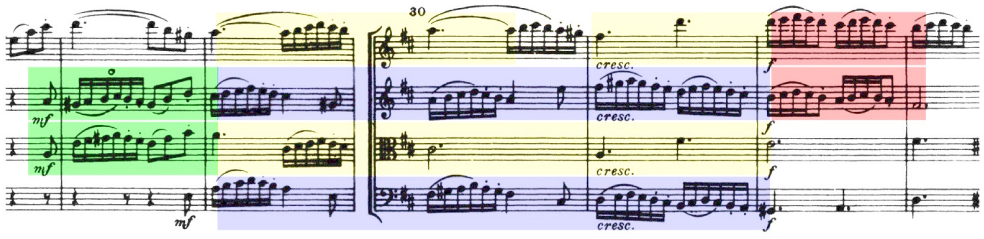


FIGURE 8.9 – *Textures et mouvements parallèles [10]. Extrait d'un quatuor à cordes de Haydn (op. 33 no. 6, m. 28-33). L'analyse de référence contient quatre mouvements parallèles indiqués ici par quatre couleurs différentes. Chacun de ces mouvements parallèles fait jouer les voix deux par deux.*

Avec Florence Levé, Marc Rigaudière, Florent Mercier et Donatien Thorez, j'ai proposé de formaliser la décomposition de la texture en strates [10]. À l'intérieur de ces strates, les voix peuvent avoir plusieurs types de relations (du moins au plus contraignant, h, homorythmie, p, mouvement parallèle, o, octave, u, unisson). Nous avons conçu un algorithme de détection de passages homorythmiques (éventuellement avec des contraintes supplémentaires $p/o/u$) à l'intérieur d'une

polyphonie non séparée en voix. Grâce à une simple programmation dynamique, cet algorithme est en temps $O(n^2)$, où n est le nombre de notes. En évaluant cet algorithme par rapport à une analyse manuelle de dix mouvements de quatuor à cordes (Mozart, Haydn), nous retrouvons plus de la moitié des mouvements parallèles indiquées avec une précision de plus de 80% [10].

Nos travaux sur la texture sont plus particulièrement portés par Florence Levé, en collaboration avec Marc Rigaudière (Sorbonne). En 2015, nous avons débuté une nouvelle collaboration avec Florence Doé de Maindreville (Reims), spécialiste des quatuors à cordes du XIX^{e} siècle. Nous cherchons à mieux modéliser les différentes textures et à trouver des moyens algorithmiques de les détecter.

8.2.3 Analyse locale : harmonie et cadences

L'analyse *harmonique*, qui s'intéresse aux accords de notes, a suscité de nombreux travaux d'informatique musicale. Les algorithmes de détection de tonalité locale [42, 50, 70] estiment la tonalité la plus probable pour chaque segment de la pièce en considérant les hauteurs de notes. La détection d'accord peut notamment se réaliser suite à une segmentation harmonique de la partition [76]. Des travaux plus récents ont pris aussi en compte la distribution des intervalles [106] ou la combinaison de méthodes [118].

Détecter les tonalités locales est un premier pas vers une analyse *fonctionnelle*, précisant les degrés et fonctions de chaque accord tout comme les progressions harmoniques (enchaînements d'accords) [104, 127]. Les *cadences* sont des progressions harmoniques particulières qui marquent les fins de phrases musicales et les transitions entre les parties d'une pièce, et contribuent à l'impression que la musique « se termine », est « suspendue », ou « part dans d'autres directions ». Elles sont le principal moyen d'articuler le but tonal d'une pièce (voir, ci-dessous, la forme sonate 8.3). Si la détection de tonalité locale réussit généralement, les bornes précises de ces tonalités sont souvent mal estimées, les notes modulantes et les cadences n'étant pas précisément identifiées.

Nous avons proposé dans [15] une modélisation des cadences parfaites (mouvement de basse, présence des accords de dominante et de tonique) (Fig. 8.10). Cette technique simple parvient à détecter 57% des cadences parfaites, sans quasiment aucun faux positif, dans un corpus de 36 fugues. L'analyse des cadences et des progressions harmoniques est désormais l'un des principaux objectifs de l'équipe (voir section 10.1).

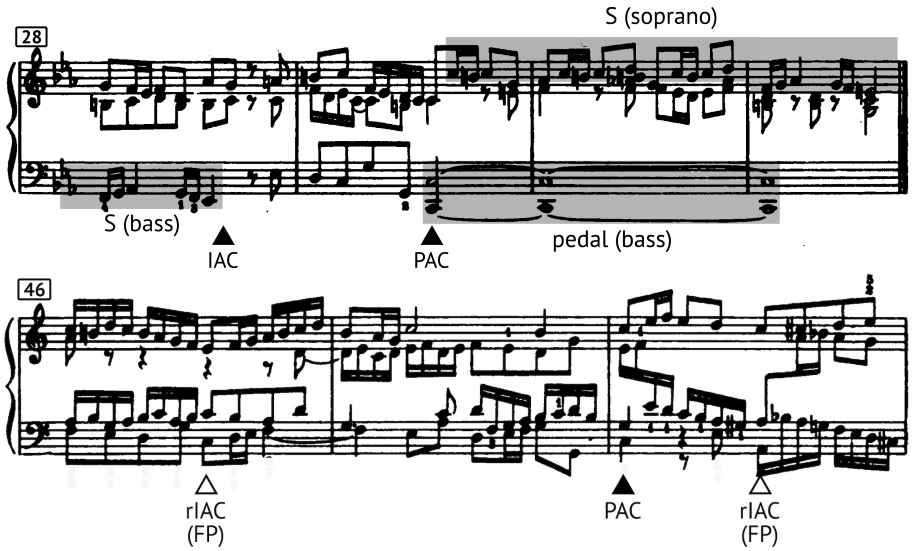


FIGURE 8.10 – *Détection de cadences [15]. (Haut) La fin de la fugue en Do mineur BWV 847 comporte une cadence parfaite (PAC), correctement détectée. La cadence imparfaite (IAC) n'est pas pour l'instant recherchée. (Bas) Dans la fugue en La mineur BWV 865, la cadence parfaite (PAC) est bien détectée, mais l'algorithme détecte ici deux faux positifs (cadences parfaites avec voix supérieure autre que la tonique, rIAC). Ces faux positifs viennent en particulier du fait que la détection se limite au contexte harmonique et n'inclut pas encore des données phraséologiques (détection de fins de phrases).*

8.3 Analyse globale : vers l'analyse de formes musicales

Idéalement, l'analyse d'une oeuvre inconnue se fait sans a priori – si ce n'est une connaissance implicite de l'ensemble du répertoire et de ses formes existantes. Le but d'une analyse globale est alors de réaliser directement une analyse à grande échelle. La forme convenant aux données d'entrée, pièce ou ensemble de pièces, sera éventuellement inférée.

Une telle inférence de structure pose de grandes difficultés combinatoires, mais certains travaux prometteurs se sont attaqués à ces questions. Alan Marsden [138] tente de s'approcher automatiquement d'une analyse schenkérienne [31], par des réductions successives d'une partition monophonique. Au Japon, une équipe essaie d'automatiser directement les modèles de la *Generative Theory of Tonal Music* (GTTM) [51] en implémentant des règles de groupement préférentielles [103]. Leur modèle se limite pour l'instant à l'analyse de thèmes. Algorithmiser ce type de modèles demande de faire face à des choix analytiques et à une explosion combinatoire du nombre de dérivations possibles.

Les différentes techniques développées ou utilisées par Algomus ont pour but d'aller vers une *analyse globale* de la partition, en s'inscrivant dans une *analyse de forme* d'un corpus identifié. Une de nos originalités est que nous ne cherchons pas à faire une analyse globale à partir des notes. De la même manière qu'une analyse syntaxique d'une phrase ne part pas des lettres ou des phonèmes, mais plutôt des mots, *nos analyses ne partent pas des notes mais d'éléments d'analyse locale, calculés ou manuels.* Cette simplification du « bas-niveau » nous permet de viser des objectifs plus synthétiques pour le « haut-niveau ». Je présente ici nos résultats de notre pipeline d'analyse de fugues, ainsi que quelques résultats préliminaires sur les formes sonates ainsi que sur la modélisation d'inventions avec des grammaires.

Fugues. Une *fugue* est une pièce de musique polyphonique *contrapuntique*, c'est-à-dire où chaque voix joue une ligne mélodique : l'harmonie d'ensemble résulte de la combinaison des voix.

Le début de la fugue de J.-S. Bach en *Do* mineur BWV 847 est représenté sur la figure 8.11. Les fugues sont généralement composées de deux à cinq voix, et cette fugue a trois voix. La fugue est construite sur un thème appelé *sujet* (S) qui revient tout au long de la pièce. La figure montre ainsi les trois premières *entrées* (ou *occurrences*) du sujet. Le sujet est *exposé* dans une voix (l'alto), en commençant par un *Do*, jusqu'à ce que la seconde voix entre (la soprano, mesure 3). Le sujet est alors exposé à cette seconde voix, mais il est désormais transposé, débutant par un *Sol*. Durant ce temps, la première voix continue avec le *premier contre-sujet* (CS1).

La fugue alterne entre d'autres entrées « thématiques » du sujet et des contre-sujets (huit occurrences de S, six de CS1 et cinq du *deuxième contre-sujet*, CS2) et des développements sur ces motifs appelés *épisodes* ou *divertissements* (E, les deux premiers épisodes sont indiqués sur la Fig. 8.11).

Les épisodes peuvent contenir des *cadences* qui concluent des moments de tension. Ils ont fréquemment des *marches harmoniques*, c'est-à-dire des passages où un motif à plusieurs voix est répété de manière consécutive, en commençant sur des notes différentes. La fin de la fugue est souvent marquée par un *stretto* qui est une succession d'entrées incomplètes de S dans toutes les voix. La cadence finale est souvent suivie d'une *pédale de basse* (une note tenue durant plusieurs harmonies, qui peut accompagner une dernière exposition d'un sujet, voir le haut de la figure 8.10). La figure 8.1, au début du chapitre, montre la structure de toute cette fugue en *Do* mineur.

Analyse automatisée de fugues. Weng et Chen ont essayé d'identifier certaines formes (fugue, rondo), mais sans analyse détaillée [94]. J'ai commencé à travaillé sur ce sujet en 2011 avec Richard Groult et Florence Levé. Nous avons proposé un pipeline complet d'annotation des fugues [15], testé sur les 24 fugues du premier livre du *Clavier bien tempéré* de Bach et les 12 premières fugues de l'opus 87 de Shostakovitch. Le pipeline utilise les étapes suivantes :

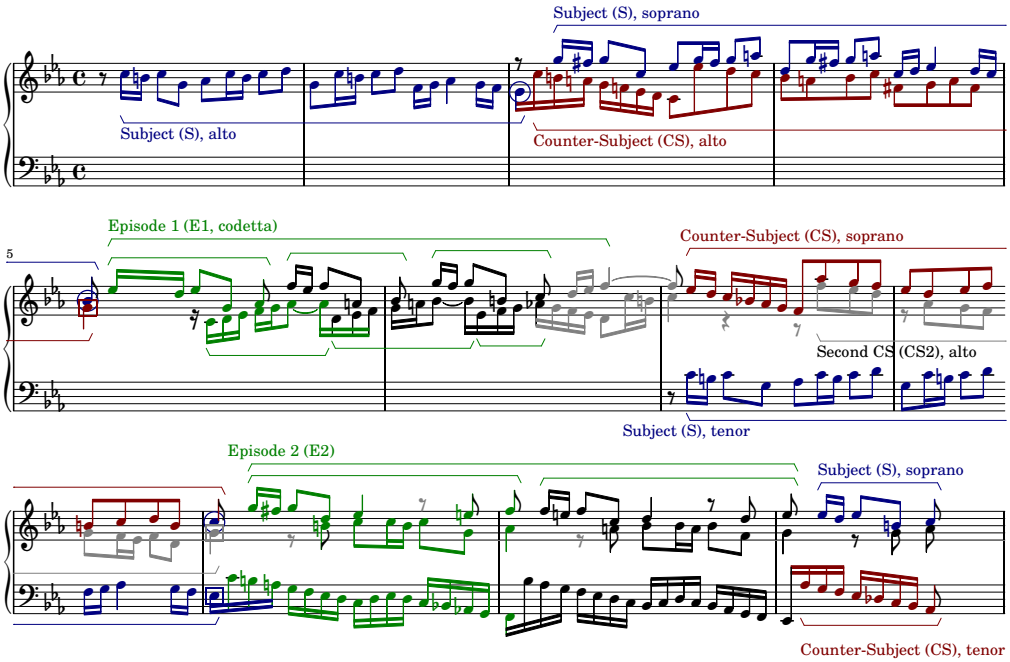


FIGURE 8.11 – *Début de la fugue en Do mineur de J.-S. Bach (fugue 2 du premier livre du* Clavier bien tempéré*, BWV 847), avec indication des sujets (S), contre-sujets (CS1 and CS2), et des deux premiers épisodes (E1 and E2). Les notes terminant les sujets sont entourées, et les notes terminant les contre-sujets sont encadrées. Dans les épisodes, les crochets montrent les motifs récurrents (trois occurrences pour E1, deux pour E2).*

— *choix de la note de fin des sujets* (succès dans 64 % des cas) *et des contre-sujets*, par comparaison des motifs potentiels avec l'ensemble des occurrences dans la partition [1] ;

— *détection des occurrences des sujets et contre-sujets* (avec plus de 80 % de sensibilité et de précision), par comparaison approchée sur les hauteurs diatoniques, et avec une recherche exacte des durées, excepté sur la première note et la dernière note [3]. Sur la figure 8.1, les sujets et les deux contre-sujets sont parfaitement retrouvés, excepté une occurrence d'un contre-sujet répartie entre plusieurs voix (mesures 26 à 28) ;

— *recherche de marches harmoniques*, détectées à partir de trois occurrences successives d'un motif diatonique sur toutes les voix. Les marches harmoniques sont un des marqueurs des épisodes (plus de 40 % des épisodes des fugues de Bach sont composés de telles marches harmoniques) [5] ;

— *détection des cadences* (fins de phrases musicales) *et des pédales* (notes maintenues alors que plusieurs accord s'enchaînent) [15] ;

Tous ces éléments sont locaux, et les trois premiers points sont obtenus par des adaptations de l'algorithme de Mongeau-Sankoff présenté page 56. Cependant, *nous inscrivons ces éléments locaux dans une analyse globale*. En effet, les fugues ont été souvent prises comme exemples pour la recherche de motifs, car elles présentent de nombreuses occurrences des sujets et contre-sujets. En étendant les algorithmes, on arrive à localiser des occurrences approximatives de ces motifs... à peu prêt partout dans la fugue ! Même s'il est juste de dire que « toute la fugue est construite sur S et CS », l'intérêt d'une analyse est de décortiquer cette construction, de lui donner du sens, en particulier sur la structure.

1. La figure 9.5, au chapitre suivant, montre les résultats de l'algorithme (notes encadrées) sur des sujets dont la fin est ambiguë.

Notre approche est différente et nous ne considérons pas la fugue juste comme un prétexte pour faire des recherches de motifs. Ce que nous considérons comme occurrence s'inscrit dans une analyse globale : les occurrences incomplètes (comme « h » sur la figure 8.1, pour tête du sujet, *subject head*) sont précisément dans les épisodes, tandis que les passages d'exposition sont caractérisés par des occurrences complètes. Nos analyses de référence (voir page 74) font ce type de distinction.

Enfin, nous utilisons un modèle de Markov pour *rassembler une partie des éléments locaux dans une segmentation* (Fig. 8.12) [15], L'idée est à la fois d'intégrer une certaine flexibilité dans la reconnaissance de structure et de pouvoir réaliser cette reconnaissance même avec une analyse locale parfois ambiguë ou erronée. Nous souhaitons compléter ce modèle très simple en prédisant plus d'états tout comme en prenant en compte plus d'éléments locaux. Pour cela, nous avons commencé un travail sur l'apprentissage des poids du modèle, qui étaient jusqu'à maintenant fixés manuellement (voir page 78).

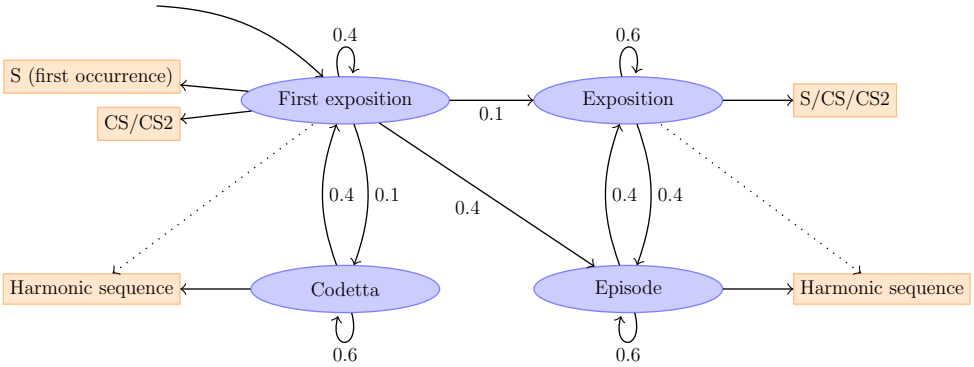


FIGURE 8.12 – *Modèle de Markov caché utilisé pour prédire la structure (exposition/épisode) à partir des éléments locaux d'analyse [15]. Les probabilités de transitions ont été choisies manuellement pour favoriser une certaine stabilité à chaque état, les différentes sections durant habituellement au moins deux mesures. Les états « first exposition » et « codetta » servent uniquement durant l'exposition initiale. Les probabilités d'émission n'ont pas été représentées sur ce schéma. Les émissions avec des flèches pointillées sont rares (marches harmoniques durant les expositions).*

Le modèle actuel permet déjà de prédire une première segmentation avec une alternance des sections d'exposition et d'épisodes. Une évaluation subjective du résultat de la segmentation couplée à l'ensemble des analyses locales sur les 36 fugues a jugé 17 analyses comme « bonnes », 12 « correctes » et 7 « mauvaises ». En utilisant les outils décrits au chapitre suivant, nous proposons une visualisation interactive de l'analyse sur ces 36 fugues (`http://algomus.fr/fugues`).

Formes sonates. Nous avons présenté en 2014 un travail préliminaire sur la *forme sonate* effectué lors des stages de Laurent David et Corentin Louboutin [9]. Cette forme est bâtie sur deux zones thématiques (principale, P, et secondaire, S) mais surtout sur un *plan tonal* sur l'ensemble de la pièce (Fig. 8.13). L'analyse d'une forme sonate demande donc elle aussi de combiner des aspects locaux avec des aspects globaux (plan tonal, structure d'ensemble). Le premier défi est dans la *détection de la structure générale* : couple exposition/réexposition, avec des marqueurs précis lorsqu'ils existent, formant la structure tonale à grande échelle.

Après une identification de fins de phrases, nous avons tenté une première approche de détection des zones P et S : recherche de la zone P du départ à une fin de phrase, sans transposition, puis recherche de la zone S sous la contrainte d'une transposition depuis la dominante. Cela permet de retrouver le couple exposition/réexposition, mais la zone S prédite peut être détectée en amont de la zone réelle S, la fin de la transition entre P et S étant déjà transposée à la dominante dans l'exposition (Fig. 8.14). Un objectif serait d'identifier précisément (lorsque c'est possible) la fin du premier thème et le début du second, en s'appuyant sur une demi-cadence particulière, la *césure médiane*, lorsqu'elle existe [98].

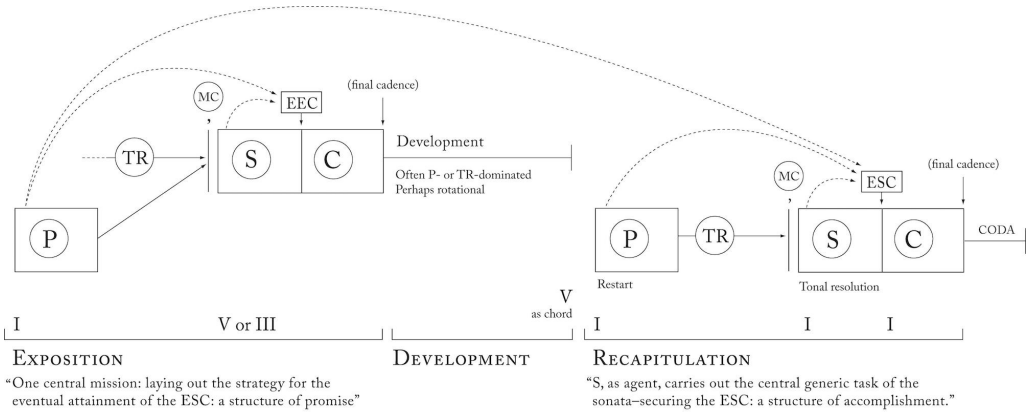


Figure 8.13 – *Structure d'ensemble d'une forme sonate, schéma et notations tirés du livre de Hepokoski et Darcy [98, page 17]. La forme sonate est construite sur un thème (ou une zone thématique) principal (P), un thème secondaire (S) et une conclusion (C). Entre les deux thèmes, la transition (TR) se termine par la césure médiane (*medial caesura*) (MC). À la fin du thème secondaire se trouve une cadence parfaite, dénommée EEC (*Essential Expositional Closure*) dans l'exposition et ESC (*Essential Structural Closure*) dans la réexposition (*recapitulation*).*

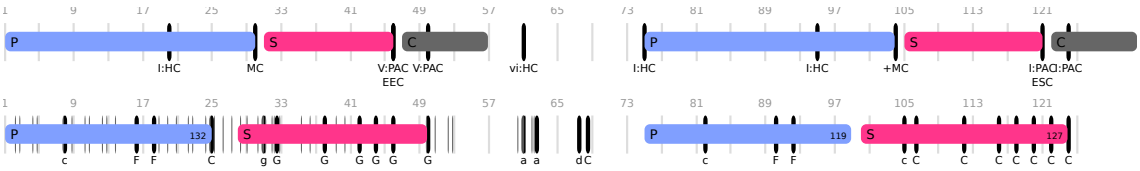
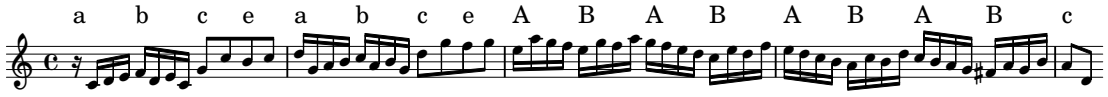


Figure 8.14 – *Recherche du couple exposition/réexposition dans le premier mouvement du quatuor à cordes K. 157 n° 4 de W. A. Mozart [9]. Le schéma du haut montre une analyse de référence avec cadences et marqueurs structurels en utilisant les notations de [98]. Le schéma du bas représente ce qui est trouvé par le programme. Les traits fins verticaux dans l'exposition montrent des fins de phrases potentielles, et les traits gras des cadences potentielles.*

Formalisation par grammaires paramétriques. Dans l'élan des modélisations du langage naturel, la modélisation de la musique par grammaires a été proposée depuis les années 1980, notamment dans les travaux précurseurs de Marcel Mesnage et d'André Riotte [99]. Certaines modélisations par grammaires ou automates peuvent aller jusqu'à la composition automatique [61, 83]. La musique est différente des langages naturels : il n'y a généralement pas de grammaire aussi bien formalisée, le sens fait généralement débat, et enfin les éléments de base, les notes, sont dans un espace bi-dimensionnel. Les solutions existantes jusqu'à présent réussissent parfois à modéliser une pièce mais sont peu flexibles et ne traduisent pas vraiment le discours musical.

Avec Sławek Staworko, j'ai proposé d'utiliser des *grammaires paramétrées*, ce qui permet de réutiliser du matériau musical identique ou similaire dans différents contextes [16]. Cette étude contient un premier algorithme identifiant une dérivation d'une telle grammaire dans une séquence musicale réencodée sur des motifs élémentaires, et une application sur des inventions de Bach (Fig. 8.15). Les inventions sont des pièces pour clavier jouant en imitation de manière obstinée sur un petit nombre de motifs de base. Dans cet exemple, l'invention à deux voix est construite sur les motifs des premiers temps (`ab` et `ce`). La paramétrisation nous a permis d'utiliser la même grammaire pour dériver partiellement la structure de trois des inventions. Pour cela, pour gagner en généralité :

— nous avons choisi de ne pas partir des notes mais d'éléments d'analyse locaux (ici calculés manuellement) représentant la surface musicale. Partir de la partition ajouterait une difficulté




```
abce|abce|ABAB|ABAB|c?AB|BBzz||--ab|--ab|--AB|--AB|eece|cees| ...
--ab|--ab|eece|cees|abce|ec?z||abce|abce|ABec|ABec|ABAB|ABAB| ...

... |abAA|BbAz||ABss|abss|ABss|abss|abab|abcz|AB??|----||
... |c?AB|BBcz||--AB|ssab|ssAB|ssab|eece|ceAB|ceaz|----||
```

$$
\begin{cases}
S_0() & \to P(x,y,z,w) + P(x,y,z,w) + P(z,w,x,y) \\
P(x,y,\{z,w\}) & \to T(x,y) + D(z,w) + I(w) \leq 2 \\
P(x,y,\{z,w\}) & \to T(x,y) + I(w) \leq 2 \\
P(x,y,\{z,w\}) & \to T(x,y) + D(z,w) \leq 2 \\
T(x,y) & \to (x/- + y/- + -/x + -/y) * [1;2] \mid (-/x + -/y + x/- + y/-) * [1;2] \\
D(x,y) & \to (x/- + y/-) * [3;4] \mid (-/x + -/y) * [3;4] \\
I(x) & \to (x/-) * [3;4] \mid (-/x) * [3;4]
\end{cases}
$$

FIGURE 8.15 – *Modélisation par grammaire paramétrique de la structure musicale [16].*

(Haut) Motifs élémentaires, d'une durée d'une noire, utilisés pour modéliser l'invention de Bach en Do majeur BWV 772. a/A : double-croches, montantes/descendantes, b/B : double-croches en tierces, descendantes/montantes, c : croches, grand intervalle, e : croches, petit intervalle.

(Milieu) Réduction manuelle de la pièce en utilisant ces motifs élémentaires.

(Bas) Grammaire paramétrique modélisant la pièce (S_0) comme trois parties (P) contenant chacune une section thématique (T) suivie d'une section de développement (D) et/ou une section d'intensification (I). Chacune des trois sections utilise différement le matériel musical passé en paramètres*, x et y.*

supplémentaire. Cette modélisation est *avec perte*, nous ne sommes pas capable de restituer toute la pièce ;

— la modélisation n'est jamais exacte ni régulière : l'algorithme permet d'identifier un alignement entre une dérivation de la grammaire et la pièce jusqu'à une certaine *distance*. Pour limiter l'explosion combinatoire, nous avons introduit des bornes sur chaque production (« ≤ 2 » sur la figure 8.15), limitant la distance autorisée entre la dérivation de la production et le segment de pièce correspondant.

Ce travail ne reste qu'une première approche : la modélisation reste difficile. Nous poursuivrons ces pistes en tentant, d'un côté, de relâcher les contraintes pesant sur les éléments locaux, et, de l'autre, de favoriser l'émergence de structures à grande échelle.

9 Développement, visualisation et évaluation

Comment utiliser et évaluer les algorithmes décrits dans le chapitre précédent ? Plusieurs logiciels existants ont des fonctionnalités d'analyse, de visualisation ou d'interaction avec des partitions (section 9.1).

Chez Algomus, nous ne cherchons pour l'instant à développer un code de production. Fin 2011, lorsque nous débutions notre travail sur les fugues, j'avais commencé à écrire un code prototype. Ce code a été repris, approfondi et étendu par toute l'équipe pour nos différentes études. Notre code de développement s'appuie sur la bibliothèque Python `music21`. Emmanuel Leguy, Guillaume Bagan, Richard Groult et Nicolas Guiomard-Kagan ont particulièrement contribué à ces développements.

Notre souhait est d'arriver à une plateforme logicielle de *prototypage d'analyse musicale*, permettant d'expérimenter différentes techniques, de les tester, de les partager et de les confronter. Pour encoder les éléments analytiques locaux ou globaux discutés dans le chapitre 8, nous avons ainsi formalisé le concept de *schéma d'analyse* vu comme ensemble de *lignes d'analyse* groupant des *étiquettes*. Nous avons réalisé plusieurs visualisations de ces schémas (section 9.2).

Enfin, la recherche et le développement de nos algorithmes d'analyse musicale se sont accompagnés d'un travail d'évaluation par la réalisation de *corpus avec des analyses de références*, que ce soit au sein de l'équipe ou en lien avec des collègues musicologues, ainsi que par une réflexion sur cette évaluation (section 9.3).

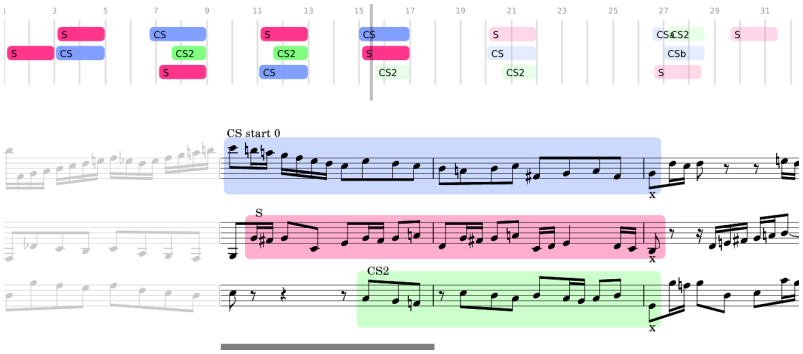


FIGURE 9.1 – *Extrait d'une vidéo produite par ly2video de la fugue en Do mineur BWV 847, reprenant les annotations de référence que nous avons encodées [14]. La vidéo complète est disponible à l'adresse* `algomus.fr/video`.

Ce chapitre reprend plusieurs parties d'un article présenté aux Journées d'Informatique Musicale 2015 [14] ainsi que des discussions publiées dans le chapitre [20].

9.1 Quelques logiciels existants

Plusieurs *logiciels* sont destinés totalement ou partiellement à l'analyse, la visualisation et l'édition de partitions.

Analyse musicale. On peut tout d'abord évoquer les travaux pionniers autour du Morphoscope [63], boîte à outils au service d'une analyse manuelle. D'autres suites logicielles sont spécialisées dans l'analyse, comme Humdrum [74] (collection de scripts appelés en ligne de commande), MIDI Toolbox [85] (bibliothèque Matlab), ou, plus récemment, music21 [133] (bibliothèque Python). De plus, certaines fonctionnalités analytiques sont présentes dans des logiciels qui sont aussi destinés à la composition, comme Rubato (Mazzola) ou OpenMusic [142, 151]. Des outils comme iAnalyse sont spécialisés dans l'édition et la visualisation [132]. Enfin, certaines plateformes d'analyse sont destinées à étudier de la musique électro-acoustique, comme l'Acousmographe [135], EAnalysis [153] et TIAALS [158].

Visualisation de partitions. De nombreux logiciels de gravure musicale existent. Si l'on se concentre sur les logiciels prenant en entrée une notation symbolique et produisant des partitions, on peut tout d'abord citer le logiciel libre Lilypond [186], développé depuis 20 ans et se caractérisant par une très grande qualité typographique ainsi que par une excellente flexibilité. VexFlow [189] est une bibliothèque Javascript pour afficher du contenu musical. Guido [68] propose une bibliothèque C/C++ portable permettant la mise en page et le rendu de partitions. Elle est aussi accessible via un service web REST pour générer en ligne l'affichage de partitions.

Édition et visualisation. INScore [185, 136] est un éditeur de partition augmentée. Il est capable d'animer une partition et de lui ajouter des éléments graphiques (curseur, annotations...). L'animation est considérée comme une « partition interactive » car elle peut être modifiée en temps réel en interagissant soit via un panneau de contrôle (modification de tempo par exemple) soit directement en interprétant une pièce musicale. INScore est interfaçable en python et en pure data via OSC (Open Sound Control).

Le logiciel iAnalyse d'aide à l'analyse musicale est destiné à la présentation, à l'annotation et à l'analyse musicale [112]. Il permet de synchroniser puis d'annoter manuellement des fichiers images de partitions, et de produire à partir de ces annotations une vidéo de l'analyse. Les annotations sont ici des symboles graphiques et ne sont pas liées de manière logicielle aux notes de la musique sous-jacente. Enfin, le projet « Écoutes signées » a exploré de nombreuses voies de représentation, de manipulation et de visualisation de partitions avec leurs annotations, en créant des « maquettes » transmettant une « manière d'écouter » particulière [134, 160].

9.2 Modélisation, développement et visualisation

Algomus développe principalement en Python. Notre code a été tout d'abord développé dans le cadre de nos recherches – en particulier pour l'analyse des fugues. Nous avons ensuite essayé d'améliorer sa qualité et réutilisabilité, en commençant par les éléments qui nous semblaient le plus intéressant pour d'autres. Les paragraphes suivant décrivent ces développements en ce qui concerne la *modélisation* et la *visualisation* d'analyses. Le code correspondant directement aux tâches d'analyse musicale est lui toujours un code de prototypage, utilisé en interne. Une de nos perspectives est de fiabiliser et de distribuer une partie de ce code (voir section 10.2).

Modélisation d'analyse : Étiquettes, lignes et schémas. Traditionnellement, l'analyste *annote les partitions* avec plusieurs symboles graphiques (Fig. 7.2). On peut ainsi voir cette analyse comme un ensemble de *calques* qui regrouperaient chacun un ensemble de symboles concernant une facette particulière de l'analyse.

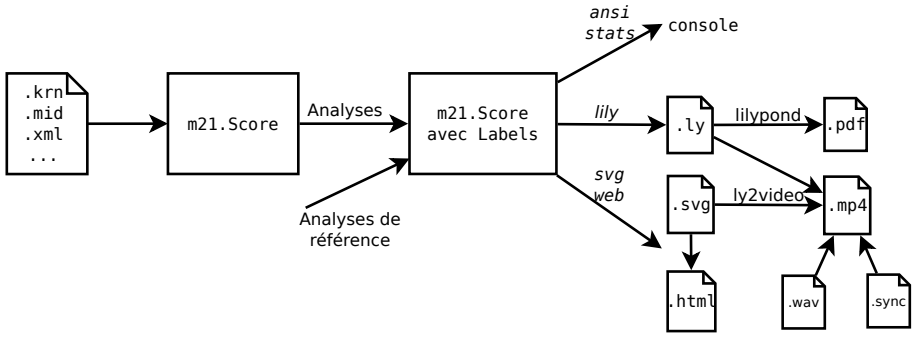


FIGURE 9.2 – *Modélisation et visualisation de schémas d'analyse avec music21. Nous avons étendu music21 en proposant un « schéma d'analyse », qui est une partition music21 (*`Score`*) annotée avec des étiquettes (*`Label`*). La partition peut avoir été créée à partir d'un fichier d'entrée (*`.krn`*, *`.mid`*, *`.xml`* ...), puis analysée avec les fonctionnalités de music21 ou de n'importe quel autre programme. On peut aussi considérer des analyses manuelles, par exemple pour des analyses de référence. Nous avons développé plusieurs visualisations de ces schémas : sortie texte pour la console, partition* `.pdf`*, schéma* `.svg`*, page web* `.html` *et vidéo* `.mp4` *[14].*

Comment modéliser cette annotation de partition ? Destiné principalement à l'annotation de fichiers son, JAMS (JSON Annotated Music Specification) [166] est une proposition récente de formats d'annotation interopérables. Côté symbolique, les formats `.krn` / Humdrum [74] et MEI ont aussi des possibilités d'annotation. Nous avons proposé dans [14] de modéliser une analyse par les éléments suivants :

— l'*étiquette* (`Label`) est l'élément analytique de base, correspondant à une annotation graphique sur une partition. Cette étiquette peut avoir une durée (motif, section, pédale) ou non (point d'arrivée d'une cadence, autre événement) ainsi qu'un type et d'autres informations ;

— plusieurs étiquettes peuvent se regrouper dans une même *ligne* d'analyse. Une ligne représente un objectif particulier d'analyse, et peut s'imaginer comme un ensemble de symboles graphiques qui seraient sur un même calque. Une ligne peut être attachée à une voix de la partition (e.g étiquettes pour la voix soprano) ou être indépendante (l'ensemble des marches harmoniques, ou bien une structuration de la partition). Sur nos représentations des schémas (figures 8.1 et 9.4), les lignes sont visualisées comme alignement des étiquettes ;

— enfin, un *schéma* est un ensemble de lignes d'analyse concernant une même pièce. Un schéma peut s'imaginer comme une superposition des calques des différentes lignes d'analyse. Le schéma peut représenter une analyse de référence ou bien une analyse produite par ordinateur.

En collaboration avec Guillaume Bagan, Nicolas Guiomard-Kagan, Richard Groult et Emmanuel Leguy, j'ai étendu la bibliothèque Python music21 [133] en proposant un nouveau module `music21.schema` avec ces principes. Les schémas servent à représenter facilement certains éléments d'une analyse mais aussi à comparer des analyses entre elles, par exemple lors de l'évaluation d'algorithmes d'analyse musicale. Les schémas sont créés par un analyste (voir section 9.3) ou calculés de manière algorithmique par l'implémentation de méthodes telles que celles présentées au chapitre précédent.

Visualisations des analyses. Principalement grâce au travail d'Emmanuel Leguy, nous avons proposé plusieurs méthodes de visualisation pour les schémas représentés comme des objets `music21` (Fig. 9.2) :

— sur partition Lilypond (Fig. 9.3), notamment grâce au projet étudiant de Jonathan Collet,

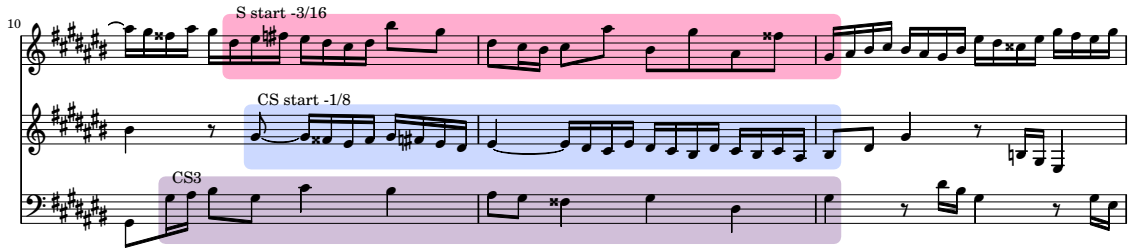


FIGURE 9.3 – *Extrait de la partition analysée de la fugue en Do♯ majeur BWV 848 de J.-S. Bach, compilée par Lilypond. Les étiquettes sont affichées au moyen des macros de* `frameEngraver.ly` *[188].*

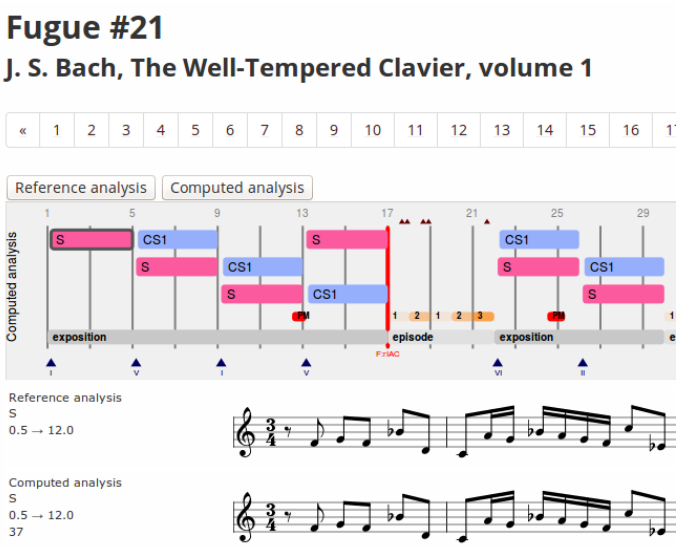


FIGURE 9.4 – *Visualisation interactive de la fugue en Si♭ majeur BWV 866 de J.-S. Bach (`www.algomus.fr/fugues`). Cliquer sur les étiquettes permet d'afficher et d'entendre les extraits de partition.*

— via une page web interactive avec des extraits musicaux produits par les bibliothèques javascript music21j et Vexflow (Fig. 9.4),

— ou enfin sur une vidéo produite par ly2video [187]. Dans ce logiciel, développé par Jiří "FireTight" Szabó, Adam Spiers et Emmanuel Leguy, une vidéo, générée à partir d'un fichier Lilypond, permet de suivre la partition tout en l'écoutant. Le son provient d'un flux MIDI ou d'un enregistrement réel, et ly2video est capable de gérer la synchronisation entre les espaces graphique (de la partition), temporel (de l'audio) et musical (symbolique). Nous avons ajouté la possibilité de visualiser d'autres images, synchronisées avec la partition. Grâce à ces fonctionnalités, nous avons produit des vidéos d'analyses manuelles ou calculées (Fig. 9.1), et, en collaboration avec l'artiste Zviane, une première animation sur les fonctions harmoniques (voir section 11.2).

À terme, nous souhaitons proposer une visualisation interactive de ces partitions annotées (voir section 10.2).

Distribution et valorisation. Le code est développé en licence libre (GPLv3+). Lorsque c'est possible, nous essayons de faire intégrer ce code aux projets open-source existants. C'est déjà le cas pour notre contribution à ly2video [1], et nous cherchons à le faire pour le module `music21.schema`. Une partie de ce code est déjà librement téléchargeable [2].

1. `github.com/aspiers/ly2video`
2. `git.algomus.fr`

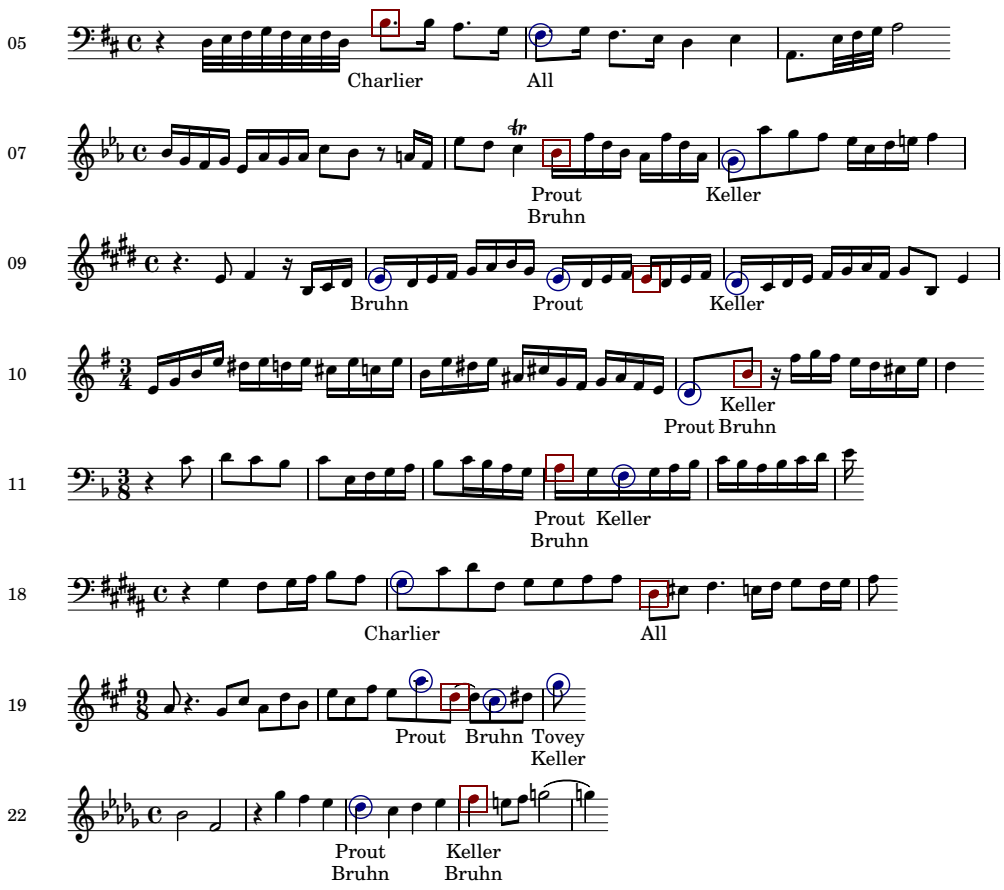


FIGURE 9.5 – *Les huit sujets des fugues du premier livre du Clavier bien tempéré de Bach où au moins deux sources ont une définition différente du sujet [15]. Les notes entourées montrent les fins possibles de sujet, et la note encadrée est le résultat de notre algorithme (voir page 63). Par exemple, sur la fugure en Mi♭ majeur BWV 852, Prout [29] et Bruhn [62] entendent un sujet qui se termine sur le Si♭ (tonique), avec un mouvement cadentiel renforcé par le trille. Keller [35] propose lui de terminer le sujet après les doubles-croches, ce qui permet d'être synchronisé avec le début du contre-sujet. Enfin, Charlier [123] a une approche motivique, résultant dans des sujets généralement beaucoup plus courts.*

9.3 Évaluation et analyses de références

Comment estimer la qualité d'un algorithme d'analyse musicale ? Idéalement, on souhaite disposer d'analyses de référence, que ce soit pour les éléments locaux et globaux, et mesurer ainsi la performance des méthodes proposées.

Ambiguïté de l'analyse musicale. Cependant, il n'existe pas une seule analyse correcte d'une pièce donnée – différentes approches musicologiques peuvent donner différentes analyses tout aussi pertinentes. Même des notions a priori simples et fondamentales sont souvent débattues, que ce soit dans des éléments locaux ou globaux, comme dans les deux exemples suivants.

Identification des sujets de fugues. Le sujet est le motif principal d'une fugue autour duquel toute la pièce est construite (section 8.3). Son identification précise est ainsi un des premiers éléments d'analyse, mais même cette identification peut faire débat. Dans le *Clavier bien tempéré* de Bach, pour 8 des 24 fugues du premier livre, au moins deux sources musicologiques proposent ainsi une interprétation différente du sujet (Fig. 9.5). En effet, si le début du sujet est un événement cognitif

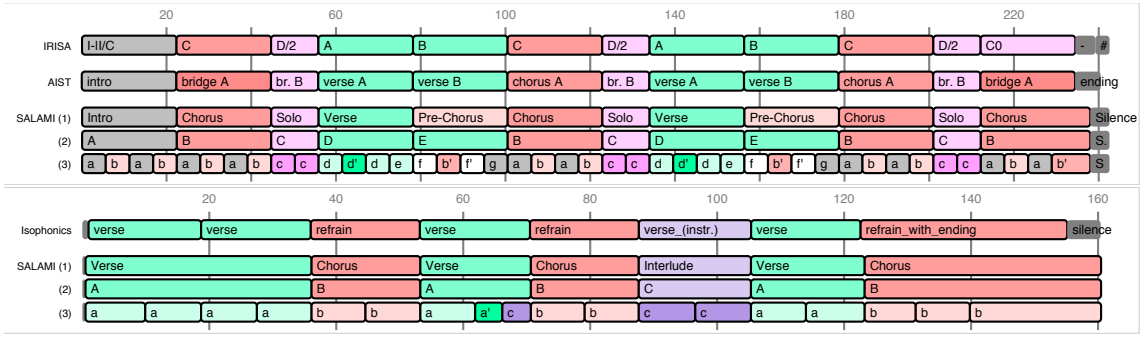


FIGURE 9.6 – *Comparaison d'annotations de référence [20]. (Haut) Annotations proposées par IRISA, AIST et SALAMI sur la chanson « Spice of Life », chantée par Hisayoshi Kazato (piste 4 de RWC-MDB-P-2001). Les trois annotations sont globalement concordantes. Cependant, les sections nommées* `B, Chorus` *par SALAMI sont nommées* `C, C, C,` *and* `C0` *par IRISA et* `bridge A, chorus A, chorus A,` *and* `bridge A` *par AIST. Les différents niveaux de l'annotation SALAMI indiquent des fonctions (1), des sections (2) ainsi qu'une segmentation à bas niveau (3). (Bas) Annotations proposées par Isophonics et SALAMI sur la chanson des Beatles « Yellow Submarine ». Les deux premiers segments nommés* `verse` *par Isophonics correspondent à un seul segment* `A` *(*`aaaa`*) par SALAMI.*

significatif, c'est, dans certains cas, moins le cas de la fin du sujet. De plus, a-t-on réellement besoin d'identifier toujours des motifs aux bornes claires ? Considérant que l'ambiguïté fait partie de la musique, Tovey écrit ainsi, à propos de la fugue en *Mi* majeur numéro 9 (BWV 854) *« It is not worthwhile settling where the subject ends and where the countersubject begins. »* [30].

Segmentation de musique pop. Nous avons passé en revue dans [20] des analyses de référence de *segmentation* de pièces, présentes dans les bases de données SALAMI (Structural Analysis of Large Amounts of Music Information) [145], Semiotic Annotations [164, 150], AIST Annotations [97] et Isophonics [139]. Ces segmentations concernent surtout de la musique populaire, et sont réalisées sur des fichiers audio – mais avec une sémantique ajoutée sur les différentes sections. Si les bords des segmentations coïncident généralement (correspondant à la détection de temps forts dans les fichiers audio), des différences manifestes de sémantique se voient sur la même chanson (Fig. 9.6).

Analyses de référence. Malgré ces difficultés, sur de nombreux points, il peut y avoir un consensus qui peut servir de vérité de terrain (« ground truth ») pour évaluer les algorithmes. Ainsi toutes nos sources sont d'accord sur la définition du sujet pour 16 des 24 fugues du premier livre du *Clavier bien tempéré*. Ces points ne font pas toute l'analyse – ce ne sont que des éléments techniques qui participent à une analyse. Mais il est déjà intéressant de pouvoir disposer de ces analyses de référence – ou mieux, de *plusieurs* analyses – sous forme informatique.

Les schémas d'analyse peuvent ainsi être donnés par un format texte encodant des analyses de référence (Fig. 9.7), format ensuite traduit dans nos objets `music21.schema`. La syntaxe de ce format a été conçue pour faciliter une saisie musicologique. Les repères temporels peuvent ainsi être entrés en mesure, temps, ou bien en fraction de mesures, ou en se servant de repères logiques. À ce jour, nous avons dans ces analyses de référence plus de 1 800 annotations encodées par 9 personnes (Tab. 9.8). Ces annotations recouvrent des indications de motifs, de structures, de cadences et de texture. Ces fichiers sont progressivement mis a disposition de la communauté sous licence libre (`www.algomus.fr/datasets`).

Évaluations quantitatives et qualitatives. En supposant qu'une analyse de référence est disponible, comment mesurer l'efficacité d'un algorithme ? Comme il n'y a pas qu'une seule analyse correcte, on ne pourra évaluer que ce qui est attendu par rapport à tel ou tel modèle ou présupposé. Les étudiants en analyse musicale sont bien eux évalués, à la fois sur des points formels ou techniques et aussi sur leurs commentaires esthétiques. Les algorithmes doivent aussi l'être, même si cette évaluation soulève quelques difficultés.

```
== S [length 4 start +1/8]
     S  1, 13, 37
     A  5, 22, 41
     T  9, 26

== S-inc [base S length 2]
     A  35

== cadences
     *  17 (V:rIAC)
     *  48 (I:PAC)

== CS1 [length 3 start -1/4]
       S  6, 23, 42
       A  10, 27, 38
       T  14

== CS1-inc [base CS1 length 1]
       S  36

== CS2 [length 3 start -3/16]
       S  10, 27
       A  14
       T  23, 38, 42

== CS2-inc [base CS2 length 1]
       T  36
```

FIGURE 9.7 – *Nos annotations de référence sur la fugue en Si♭ majeur BWV 866 de J.-S. Bach, réalisées à partir de plusieurs sources dont [62] et [29]. Ce jeu d'annotations de référence encode l'ensemble des éléments présentés en section 8.3 : motifs (sujets (S), contre-sujets (CS1 and CS2)), dont des entrées incomplètes (S-inc, CS1-inc, CS2-inc). La position de chaque occurrence est indiquée par le numéro de mesure (5 est le premier temps de la mesure 5), mais le début effectif du motif peut être après cette position logique (une croche après pour S (`+1/8`), une noire avant pour CS1 (`-1/4`), trois double-croches avant pour CS2 (`-1/16`). Les annotations donnent aussi les cadences parfaites (PAC et rIAC) aux tons de la tonique (I) et de la dominante (V). La visualisation de ces annotations (`www.algomus.fr/fugues`) est analogue à celle de la figure 8.1.*

Si la plupart de nos articles contiennent des expériences et des évaluations, nous avons discuté plus en détail des méthodologies d'évaluation dans notre revue sur la forme et la segmentation [20] ainsi que dans notre étude sur la séparation de voix [17] :

— L'évaluation peut d'abord être un ensemble de *mesures quantitatives*. À côté des mesures habituelles en classification telles que la sensibilité et la spécificité, d'autres mesures peuvent être utilisées, comme par exemple des mesures d'entropie pour la segmentation [115], mesures que nous avons étendues à l'évaluation d'algorithmes de séparation de voix [17]. Selon l'application, il faut déterminer sur quels éléments portent ces mesures. Quelques difficultés proviennent de la nature même des données musicales, se déployant dans le temps : certains éléments d'analyse (formalisés par des `Label` dans nos schémas) sont ponctuels, d'autres ont une durée. Décompte-t-on le nombre de motifs détectés, ou bien la proportion, dans le temps, de la partition correctement analysée ? Autorise-t-on une détection approximative dans le temps ? Se focalise-t-on sur les frontières des motifs ?

— L'évaluation peut aussi être plus subjective, par un ou plusieurs experts, sur la *qualité* du résultat global de l'algorithme (est-ce que mon ordinateur est un bon analyste ?). Ce type d'évaluation est particulièrement nécessaire lorsque l'objectif de la méthode est une analyse globalisante, calculant un ensemble d'éléments structurés comme dans ce que nous essayons de faire sur les fugues ou les formes sonates. Par exemple, notre analyse des fugues inclut un jugement expert sur chaque résultat (« bon », « correct », « mauvais »), qui, même s'il repose sur des conseils, laisse une certaine latitude d'interprétation.

Fugues – ISMIR 2012 [3], CMMR 2012 [5], Computer Music Journal 2015 [15]
36 fugues (Bach, Shostakovitch)
1020 annotations : thèmes S/CS/CS2, pédales, cadences
Variations – CMMR 2013 [6]
4 thèmes et variations (Mozart, Beethoven)
35 annotations : occurrences variées
Texture – ISMIR 2014 [10] + en cours
10 mouvements (Haydn, Mozart, Schubert)
691 annotations : mélodies, accompagnements, homorythmies
Formes sonates – JIM 2014 [9] + en cours
11 mouvements (Haydn, Mozart, Schubert)
67 annotations : structure P(T)S(C)/Dev/P(T)S(C)

TABLE 9.8 – *Analyses de références produites par Algomus et ses collaborateurs. Ces analyses, dont une partie est disponible à partir de* `www.algomus.fr/datasets`*, ont été en particulier effectuées et encodées par E. Cambouropoulos, K. Deguernel, F. Doé de Maindreville, M. Giraud, R. Groult, E. Leguy, F. Levé, F. Mercier et M. Rigaudière.*

10 Perspectives

Le projet de recherche d'Algomus, que je porte avec toute l'équipe, est de calculer des éléments locaux d'analyse et les synthétiser dans des analyses « complètes » de pièces musicales, se rapprochant le plus possible d'analyses musicologiques (section 10.1).

L'analyse musicale par ordinateur, qu'elle soit automatique ou semi-automatique, nécessite des collaborations et, à terme, des chercheurs mieux formés aux deux domaines [90, 126]. Ces recherches stimulent à la fois l'informatique musicale, soulevant des questions de modélisation et d'algorithmique, que cela soit dans la production d'éléments d'analyse isolés ou dans celle de synthèses, et posent aussi de nouvelles questions à la musicologie. Nous continuerons ainsi à travailler avec des théoriciens de la musique pour tenter de modéliser et d'algorithmiser certains concepts musicaux.

Cette recherche s'accompagne d'un effort de développement et de médiation pour donner aux musiciens et au grand public des outils permettant de visualiser, éditer ou discuter l'analyse musicale (section 10.2).

10.1 Algorithmique musicale

Analyse locale. Pratiquement l'ensemble des éléments locaux évoqués au chapitre 8 mériteraient d'être retravaillés. La notion même de *motif musical*, pourtant étudiée depuis de nombreuses années (section 8.2.1), gagnerait à prendre en compte le contexte musical. En effet, les diverses répétitions et transformations d'un motif peuvent mieux s'interpréter vis-à-vis de leur contexte. Par exemple, dans une fugue, on s'attend à trouver le sujet complet dans les expositions et incomplet dans les épisodes (occurrences marquées « h » dans nos schémas tel que celui de la figure 8.1). Nous aurions aussi intérêt à étendre des techniques que nous avons développées pour l'analyse de fugue (comme la détection de marches harmoniques) à d'autres répertoires. Dans les prochaines années, nous souhaitons nous focaliser sur les points suivants :

— *Cadences et progressions harmoniques.* Les cadences sont des progressions harmoniques particulières qui marquent les fins de phrases musicales et les transitions entre les parties d'une pièce. Des études musicologiques récentes se consacrent aux cadences [180], mais on trouve peu de travaux algorithmiques tentant de modéliser ces progressions [72, 128, 156]. À l'heure actuelle, seuls quelques algorithmes permettent de reconnaître des cadences simples de type cadence parfaite ou imparfaite, comme ce que nous avons fait dans les fugues, en se basant sur le contenu des accords et les mouvements de basse (voir page 63). Les outils des musicologues pour traiter des progressions plus complexes nécessitent une grande part de traitement manuel [125].

Je travaille sur ce sujet avec Florence Levé et Richard Groult. Notre approche sera de mêler règles explicites et règles apprises sur des corpus. Ce sujet est le cœur d'une nouvelle collaboration avec le conservatoire d'Amiens, pour laquelle nous avons rédigé une soumission commune d'un projet région Picardie « Cadences » porté par F. Levé. Nous travaillons

aussi sur la visualisation et la représentation de ces progressions, en particulier avec Zviane, dessinatrice de BD et compositrice québécquoise (voir section 11.2).

— *Polyphonie et texture.* Dans sa thèse, Nicolas Guiomard-Kagan travaille sur des données polyphoniques [17] et cherche à réaliser une analyse de fugues sur des données non séparées par voix. Nous souhaitons aussi approfondir nos travaux sur la texture [10], en identifiant des textures remarquables dans une partition. Ce sujet est plus particulièrement porté par Florence Levé, en collaboration avec deux musicologues, Florence Doé de Maindreville et Marc Rigaudière. Un travail a commencé en 2015 pour, d'un côté, mieux modéliser la texture dans des quatuors à cordes classiques, et de l'autre, rechercher de nouveaux algorithmes pour identifier certaines textures, comme par exemple l'imitation. Nous envisageons aussi un travail avec Emilios Cambouropoulos sur une approche complémentaire : peut-on d'abord segmenter une polyphonie, en utilisant les algorithmes tels que [117] (Fig. 8.7) puis en classifiant les segments selon leur texture ?

Analyse globale. Que cela soit dans l'analyse de formes connues ou dans celle utilisant des principes schenkériens, nous ne sommes qu'au début des recherches dans ce domaine de l'informatique musicale. Nous rejoignons l'affirmation de Anja Volk et de ses collègues [147] :

> *« The integration of isolated components of music into a holistic model of musical structure through computational modelling is a challenge for future research. »*

Analyser correctement une forme musicale, même simple, reste un défi de modélisation et de calcul. En 2011, nous avons commencé à travailler sur les fugues parce que leur structure est relativement codifiée. Nous avons depuis réussi à proposer un pipeline d'analyse quasi-automatique qui fonctionne dans une majorité de cas (page 63). Dans les prochaines années, une partie du travail sera d'améliorer cette analyse des fugues. Mais nous souhaitons avant tout aller vers *l'étude de formes plus complexes*. Le projet central d'Algomus pour les prochaines années est ainsi l'étude des *formes sonates* (voir page 65). Bien que ces formes s'inscrivent dans une théorie générale [98], elles ont de nombreuses variations. Nous poursuivrons et approfondirons le travail débuté en 2014 qui a permis, pour l'instant, de localiser de manière approximative le couple exposition/réexposition [9].

La stratégie pour analyser de telles formes est toujours de combiner expertise et apprentissage, et de *s'appuyer sur les éléments locaux – motifs, cadences et progressions, textures – pour faire une analyse globale*. Nous développerons ainsi les modèles grammaticaux et les modèles statistiques comme le modèle de Markov proposé en partie 8.3, comme nous avons commencé à le faire avec le stage de Matthieu Caron. Plusieurs modèles probabilistes pour analyser ou générer la musique ont déjà été proposés [80, 82, 107]. De notre côté, les éléments de base de ces modèles ne seront pas les notes mais les éléments d'analyse locale. Ce travail est celui de toute l'équipe, mais en particulier celui de Richard Groult et Pierre Allegraud, en collaboration avec des collègues musicologues (Marc Rigaudière).

Structures de données et algorithmes adaptés aux données musicales. Les données de partitions musicales ont intrinsèquement plusieurs dimensions : hauteur et temps. Afin de réussir nos objectifs d'analyse, nous développerons de **nouvelles représentations**, séquentielles, géométriques, arborescentes ou grammaticales, pour traiter efficacement et convenablement les données musicales et obtenir des complexités en temps permettant une application sur des partitions réelles voire sur des corpus entiers. Ce travail se fera en particulier sur la recherche de textures (voir page 60) et sur la reconnaissance et l'apprentissage de grammaires paramétrées (voir page 66). Plus généralement, nous essaierons de trouver de nouvelles **structures d'indexation** et des méthodes de **fouille de données** capables de répondre efficacement à nos principaux défis de recherche de motifs, d'accords et de texture.

10.2 Développement et objectifs sociétaux

— Nous souhaitons proposer une meilleure interaction avec les schémas d'analyse en développant une **application web interactive** pour visualiser la partition annotée. Le but n'est pas de réaliser un éditeur de partitions (tâche très complexe, demandant tout un travail en interface comme en typographie musicale), mais bien un *éditeur d'analyses*. La navigation dans la structure de la pièce sera facilitée et le focus sur certains éléments de l'analyse pourra être fait par l'auditeur. Nous cherchons ainsi à implémenter une **édition collaborative** d'analyse musicale pour confronter ou comparer ses propres annotations à d'autres, manuelles ou automatiques. Une telle visualisation interactive renforcera nos liens avec musicologues, artistes et grand public. Ce point est porté par Emmanuel Leguy au cours de l'année 2016.

— À plus long terme, nous aimerions progressivement fiabiliser et distribuer une partie du code développé par Algomus pour arriver à une bibliothèque d'outils utilisables par d'autres. Ces modules pourraient soit être directement appelés par l'application web (avec une architecture client/serveur), soit être téléchargeables pour être réutilisés dans le cadre de `music21`. Nous mènerons aussi une réflexion sur les formats d'analyse (tels que ceux proposés par MEI, voir page 50) pour permettre une interopérabilité entre outils. Nous envisageons d'ailleurs de collaborer avec des collègues de plusieurs laboratoires (IRCAM, IreMus, CNAM, GRAME) sur un projet de recherche autour d'une *bibliothèque de partitions enrichie d'outils d'analyse*.

Enfin, nous continuerons à mener des actions de médiation et de nouveaux projets artistiques (voir le chapitre suivant) :

— Nous approfondirons le lien avec certains artistes (V. Béland, Zviane) autour de **projets artistiques impliquant l'analyse musicale informatique**. L'aspect génératif de nos modèles pourra aussi être développé. Nous pourrons entamer de nouvelles collaborations artistiques, du moment qu'elles concernent au moins partiellement le traitement des partitions musicales. Nous chercherons aussi à **renforcer nos liens avec les conservatoires**, dans un objectif à la fois de médiation et de participation des professeurs et étudiants à nos recherches. Nous travaillerons particulièrement avec le conservatoire d'Amiens, avec lequel le MIS signera bientôt une convention.

— Nous préparerons des actions pour **partager nos recherches** avec le grand public, en expliquant à la fois le côté informatique (modélisation, algorithmes) et le côté musical. Nous travaillerons sur la pédagogie du répertoire dit « classique » (baroque/classique/romantique), et nous démontrerons aussi nos résultats sur des répertoires plus actuels. Nous développerons en particulier l'atelier des *pierres musicales* (chapitre suivant, page 85).

Coda

11 Médiation scientifique et artistique

« La médiation scientifique concerne toutes les actions à destination de publics sortant du cercle professionnel habituel des chercheurs : enfants et jeunes, curieux de la science, grand public, décideurs politiques et partenaires sociaux-économiques. La médiation fait intervenir le chercheur, source du contenu scientifique. Pour démultiplier notre message scientifique, certaines activités de médiation impliquent aussi d'autres personnes : professionnels de la communication et des médias (nationaux ou de proximité), enseignants et autres médiateurs. »

Ces propos sont issus d'un groupe de travail national que j'ai animé en 2010 (voir ci-dessous, page 87), s'inscrivant notamment à la suite de la démarche de collègues impliqués dans la médiation (Thierry Viéville). Je décris ici les ateliers participatifs en bioinformatique et en musique auxquels j'ai contribué (section 11.1). Algomus est aussi engagé dans la médiation artistique (section 11.2).

FIGURE 11.1 – *Actions de médiation scientifique. Habillage des « puzzles du génome », design par Marc Peyret Imagineur / Inria (2007). Assemblage d'ADN, au Palais de la Découverte, avec Mikaël Salson (2010). Structures secondaires d'ARN, avec Maude Pupin (SciencesOPark, Lille, 2007). « Compter les globules blancs », alignement collectif d'ADN au Plateau Inria (Euratechnologies) avec des élèves suivant la spécialité ISN (2015).*

Pierre Audin, Jean-Frédéric Berthelot, Samuel Blanquart, Ségolène Caboche, Rayan Chikhi, Marc Duez, Yoann Dufresne, Arnaud Fontaine, Mathieu Giraud, Benjamin Grenier-Boley, Isabelle Guigon, Sylvain Guillemot, Robin Jamet, Stéphane Janot, Jesper Jansson, Gregory Kucherov, Alan Lahure, Dominique Lavenier, Maxime Labat, Aude Liefooghe, Alban Mancheron, Antoine de Monte, Laurent Noé, Louise Ott, Aïda Ouangraoua, Pierre Peterlongo, Grégory Ranchy, Élodie Retout, Guillaume Reuiller, Tatiana Rocher, Azadeh Saffarian, Mikaël Salson, Hélène Touzet, Anne-Sophie Valin, Jean-Stéphane Varré, Christophe Vroland...

TABLE 11.2 – *De 2004 à 2015, plus de 40 bioinformaticiens et médiateurs scientifiques ont animé à une occasion ou une autre l'atelier les « puzzles du génome » et ont participé à son évolution. Merci pour leur enthousiasme, et mes excuses à celles et ceux que j'aurais pu oublier.*

11.1 Ateliers à la rencontre du public

Ateliers de bioinformatique. La bioinformatique est une excellente opportunité pour transmettre des notions théoriques d'informatique à tous les publics. Avec certains supports, on peut introduire et faire comprendre des problèmes d'algorithmique du texte et de complexité, tout en se rattachant à des enjeux sociétaux sur le vivant et la médecine.

— *Les puzzles du génome.* (http://cristal.univ-lille.fr/~giraud/puzzles) Cet atelier généraliste sur la bioinformatique a tout d'abord été conçu en 2005 durant ma thèse à Rennes (ancienne équipe Symbiose, Jacques Nicolas). J'ai continué d'étendre cet atelier en arrivant à Lille, avec plusieurs collègues de Bonsai (alors Sequoia). L'atelier a évolué et s'est progressivement professionnalisé, notamment grâce au soutien et aux conseils du service de communication d'Inria Lille (Marie-Agnès Énard). En 2007, à l'occasion de SciencesOPark, un événement pour la fête de la science organisé sur le nouveau parc de la Haute-Borne, Marc Peyret, designer, a réalisé un nouvel habillage graphique et nous a aidé à réfléchir à l'ensemble de la scénographie. En 2010, une présentation durant deux mois au Palais de la Découverte a été l'occasion de faire encore évoluer l'atelier, en collaboration avec les équipes du Palais (Sylvain Lefavrais) et les médiateurs titualaires ou stagiaires du Palais (Pierre Audin, Guillaume Reuiller, Maxime Labat).

  Ces puzzles concernent actuellement trois jeux : l'assemblage de fragments d'ADN, la recherche de structures secondaires d'ARN, et la reconstruction d'arbres phylogénétiques entre des séquences de différentes espèces (Fig. 11.3). Chacun des jeux est décliné en trois « niveaux » pour introduire progressivement des concepts algorithmiques ou biologiques. Le premier niveau est, d'un point de vue combinatoire, très facile, pour pouvoir introduire les objets biologiques. Les niveaux suivants ajoutent des difficultés combinatoires ou biologiques.

  Bien que le support n'ait plus évolué depuis 2011, ces puzzles sont toujours fréquemment utilisés par l'équipe Bonsai lors de présentations pour des événements de médiation ou lors de séances avec des lycées ou des étudiants organisées par ou avec des collègues scientifiques ou communiquants (Marion Blasquez, Caterina Calgaro, David Coupier, Alice Decarpigny, Marie-Agnès Énard, Éric Wegrzynowski, Hélène Xypas). L'atelier peut être animé par une seule personne. Nous préférons cependant être deux ou trois, pour, après une présentation introductive, maximiser les échanges avec des sous-groupes du public sur les différents jeux. En dix ans, ces jeux ont été présentés à plus de 3 000 personnes, grâce aux interventions de nombreux doctorants, chercheurs, médiateurs et ingénieurs à Rennes, Lille et Paris (Tab. 11.2). Mikaël Salson et Maude Pupin sont particulièrement actifs en médiation et se servent des puzzles, mais plusieurs collègues de Bonsai nous rejoignent à chaque événement.

— *Compter les globules blancs.* En 2013, Mikaël Salson, Marc Duez et moi avons fait un exposé interne au centre Inria sur notre projet Vidjil. Nous avons depuis développé quelques idées d'interaction avec le public dans un atelier autonome sur l'immunologie et la bioinformatique. Cet atelier est beaucoup plus spécifique, expliquant le contexte de Vidjil ainsi que quelques

FIGURE 11.3 – *Les puzzles du génome : assemblage d'ADN, structure d'ARN.*

éléments d'algorithmique. Nous parlons ainsi de recombinaison V(D)J et d'alignement de séquences (figure 11.1, en bas à droite). Même le cœur de notre algorithme, la reconnaissance par k-mots, peut être expérimentée de manière ludique par un groupe de participants. Cet atelier a pour l'instant touché 200 lycéens lors de diverses présentations.

Ces deux ateliers se présentent à l'occasion de « stands » ouverts lors d'événements (fête de la science, rencontre avec des étudiants) ou en situation plus cadrée, dans une classe ou une conférence (entre 1h à 1h30 avec introduction, jeux et conclusion, voire entre 2h30 et 3h en enchaînant les deux ateliers).

Les ateliers s'adressent à tous publics. Nous sommes particulièrement habitués à interagir avec des *classes de lycée*. Les puzzles font directement écho à certaines notions abordées en SVT (cellule et ADN, hérédité, phylogénie), en mathématiques (puissances et logarithmes, dénombrement), et évidemment aux notions algorithmiques et de programmation de la nouvelle spécialité ISN. J'aime aussi évoquer des notions plus avancées (distances, distance d'édition, k-mots, complexité asymptotique, recherche dans un dictionnaire, indexation) qui éveillent l'intérêt du public.

Les pierres musicales. (`http://www.algomus.fr/pierres`) Nous avons entamé chez Algomus en 2013 une réflexion pour créer un atelier de médiation. Les concepts d'analyse musicale sont peu connus du grand public – et même d'une partie des musiciens ! De plus, nous travaillons non pas sur des fichiers son, mais sur un objet relativement conceptuel, la *partition*. Comment parler de nos recherches, alors que la plupart de nos visiteurs ne sont pas lecteurs de partitions ?

Les recherches en perception de la musique ont montré que la plupart des personnes, même non lectrices de musique, savent reconnaître et classifier des extraits musicaux suivant certaines notions théoriques. Nous avons donc souhaité créer un atelier *qui soit d'abord à destination des non-musiciens* – même si des musiciens peuvent bénéficier de leurs connaissances pour accéder à un autre niveau de compréhension – mais de tout de même *garder notre focus sur la partition*. Le choix a été de réaliser un dispositif interactif où le participant puisse *visualiser, manipuler et écouter des notes ou des partitions*.

Le développement des « pierres musicales » est un projet d'équipe, porté par Florence Levé, Richard Groult, Emmanuel Leguy et moi, avec des contributions de David Durand, Marion Giraud, Nicolas Guiomard-Kagan, Ophélie Hérouart, Nathan Marécaux et Hervé Midaine. Un premier prototype, appelé « tapis musicaux », a été réalisé en 2014, à l'occasion d'un stage de Nathan Marécaux, étudiant L3 électronicien. Un tapis de quatre emplacements identifie des pièces RFID et permet de jouer du son. Le circuit avec les antennes RFID est relié à un Arduino contenu dans un PCDuino. Le premier jeu réalisé est une *dictée musicale* (reconstruire une mélodie à partir de fragments ou de notes).

L'atelier en lui-même porte sur un contenu musical plus qu'informatique, mais permet d'introduire la *modélisation* de la musique : nous pouvons littéralement *prendre* des notes ou des extraits

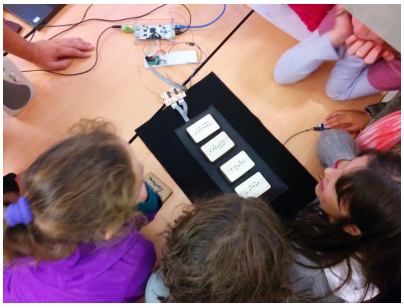


FIGURE 11.4 – *Prototype des « tapis musicaux », lors de la fête de la science 2014 à Amiens.*

de partitions, les écouter, les comparer, les classifier, et expliquer à la fois des notions d'analyse et des notions calculatoires d'informatique musicale.

Les tapis ont pour l'instant été manipulés par 500 personnes, principalement des scolaires. Nous souhaitons à la fois évoluer en qualité (nouveaux jeux, professionalisation du design, de l'électronique, de la production) et en quantité (pour toucher 5 000 personnes sur les 4 ans à venir). En 2015-2016, nous travaillons ainsi à un second prototype, avec deux objectifs :

— *Technique et design.* Hervé Midaine (ingénieur électronicien de l'UPJV, à Amiens) conçoit un circuit électronique plus robuste, permettant de gérer jusqu'à 4×4 antennes RFID. Ophélie Hérouart, de l'École Supérieure d'Art et de Design d'Amiens, effectue un stage sur le design de l'ensemble de l'atelier, en collaboration avec le FabLab d'Amiens. Elle propose une nouvelle conception du support et des pièces qui seront désormais des *pierres musicales* ainsi qu'un habillage graphique.

— *Conception de jeux d'analyse musicale.* Toute l'équipe Algomus souhaite concevoir des jeux plus proches de nos thématiques de recherche. Dès 2016, nous prévoyons de développer des jeux de classification sur la similarité de mélodies ainsi que sur les progressions harmoniques. À plus long terme, nous souhaitons réaliser des jeux sur la structuration à grande échelle, en lien avec nos thématiques d'analyse globale.

Le prochain prototype sera présenté en avril 2016, à l'occasion des journées Connexions à Amiens. Nous souhations obtenir pour 2017 un matériel relativement autonome, utilisable en conservatoire, en école de musique, mais aussi en milieu scolaire, et nous prévoyons des présentations à Lille, à Amiens et dans la région Nord-Pas-de-Calais–Picardie.

Réflexions communes. Le but de ces ateliers est de susciter des vocations pour l'informatique et pour les sciences en général... et tout simplement parler de science, et de faire, comme disait Gilles Kahn, que des « étincelles s'allument dans les yeux ». C'est aussi l'occasion échanger sur des sujets sociétaux importants comme celui de la médecine personalisée, y compris des sujets d'éthique sur la protection de la vie privée.

Manipulation physique. J'ai souhaité que l'interactivité de ces ateliers repose sur une *manipulation physique*. Pour chacun de ces ateliers, nous aurions pu développer une application web et mobile réalisant peu ou prou les mêmes fonctionnalités. Mais le fait que les objets biologiques ou musicaux soient représentés en bois ou en résine provoque une certaine expérience sensorielle : toucher, déplacer, classifier ou trier des objets physiques facilite l'intériorisation d'aspects calculatoires d'analyse et de manipulation de données. La manipulation physique favorise ainsi la participation active du public, directement pour les puzzles du génome ou les pierres musicales (les jeux sont dupliqués afin que, dans une classe, tous ou presque tous manipulent à tour de rôle), ou en groupe pour l'atelier « compter les globules blancs » (distribution de séquences à tous, alignement collectif).

Qualitatif et quantitatif. Pour être efficace tout en proposant une activité de qualité, il est essentiel d'être *quantitatif*, de viser la démultiplication de nos actions [8]. Nous avons créé un contenu réutilisable, que ce soit pour les puzzles du génomes ou les pierres musicales. Toucher 100 personnes de plus ne demande maintenant qu'une demi-journée à un, deux ou trois intervenants, alors que la conception ou l'évolution d'un atelier demande des dizaines et même des centaines d'heures de travail. Ce n'est pas moi qui ai directement rencontré les quelques milliers de personnes touchées par les puzzles du génome : la participation des équipes Symbiose, Bonsai et Algomus et du Palais de la Découverte a été indispensable.

Le *qualitatif* vient bien sûr de la conception de l'atelier, mais surtout de la *rencontre directe* entre le participant et le médiateur, la doctorante, l'ingénieur ou la chercheuse. Si le jeu est cadré, le discours est beaucoup plus libre : les ateliers sont l'occasion de rencontrer l'intervenant et d'échanger sur son parcours et ses thématiques de recherche ou de développement.

Transmettre son domaine, transmettre ses recherches. Nous acceptons que la médiation soit *d'abord sur le domaine*, avant d'être sur ses propres recherches. Les puzzles du génome portent en partie sur de l'alignement d'ADN, ce qui ouvre la porte à des discussions qui couvrent une grande partie des recherches de Bonsai. Les pierres musicales introduisent les notes, les mélodies, et prochainement les accords, et facilitent ainsi tout discours musical ou d'informatique musicale.

Évaluation. En 2011 – 2012, alors que j'étais membre élu de la Commission d'Évaluation (CE) Inria, j'ai coordonné un groupe de travail sur l'évaluation de la médiation scientifique. Comment améliorer l'identification de ces activités afin d'*inciter les collègues* qui le souhaitent à s'y investir ? Notre principale réalisation est le document *« Éléments pour une auto-appréciation des activités de médiation scientifique »* [4] [1]. Ce court document propose quelques clés de lecture d'activités de diffusion scientifique – non pas dans le but de normaliser, mais bien pour aider les collègues à décrire et à valoriser leurs activités de médiation.

Perspectives. Dans l'avenir, je souhaite continuer à développer ce type de projets collaboratifs, en particulier celui sur les pierres musicales. Outre les questions de pédagogie, ces projets soulèvent aussi de nouvelles questions de recherche sur la modélisation et sur la perception. Enfin, je suis prêt à intervenir comme conseil et soutien auprès de collègues qui voudraient créer leurs ateliers sur leurs domaines et idées.

11.2 Projets arts et science

Algomus souhaite partager l'analyse musicale avec le plus grand nombre, ce qui passe déjà par les visualisations des schémas d'analyses présentées en section 9.2. Depuis 2015, nous collaborons aussi à deux projets combinant sciences et arts. Le but principal n'est pas ici de faire passer un message musical ou scientifique, mais de contribuer à une proposition artistique. Ces projets naissants, lieux de dialogues pluridisciplinaires, se poursuivront les prochaines années.

Zviane. Zviane est dessinatrice de BD et musicienne québécoise. Zviane a suivi des études de composition musicale et d'analyse, notamment avec Luce Baudet, de l'Université de Montréal. La musique occupe une place importante dans sa vie et dans ses bandes dessinées, comme dans ses albums « Les deuxièmes » et « Ping-Pong » (éditions Pow-Pow). Zviane est particulièrement sensible à l'analyse harmonique et à son rôle dans la construction de la musique. En 2010, elle avait proposé une première représentation des fonctions harmoniques sous formes de *bonhommes*.

1. http://www.inria.fr/institut/organisation/instances/commission-d-evaluation

Après une invitation de Zviane à notre séminaire, début 2015, nous avons continué notre collaboration, notamment au cours d'une visite que j'ai effectué à Montréal. Zviane et moi avons réalisé une animation des fonctions harmoniques dans le début d'une sonate pour piano de Mozart (Fig. 11.5). Cette réalisation a demandé à la fois un aspect technique (synchronisation des vidéos par rapport au temps musical, en s'appuyant sur le travail fait par Emmanuel Leguy dans ly2video) et des réflexions sur la modélisation et l'encodage des fonctions harmoniques, en lien avec notre projet de recherche sur la détection des progressions harmoniques.

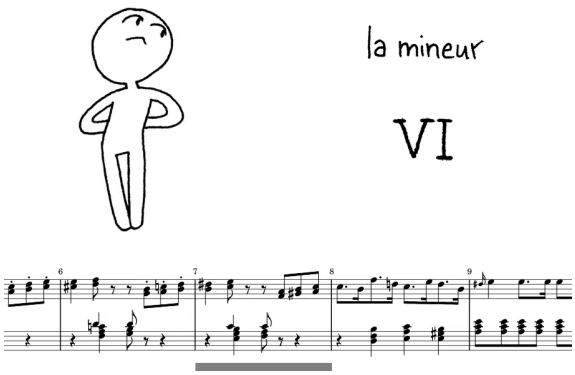


FIGURE 11.5 – *Mozart et les fonctions harmoniques, http://youtube.com/watch?v=QvE00Rfe8wQ*

J'étais heureux de faire rencontrer un univers graphique original avec nos thématiques de recherche. Nous souhaitons poursuivre cette collaboration pour étendre ce film et réaliser de nouvelles animations illustrant de manière artistique et ludique d'autres éléments d'analyse musicale... À terme, nous aimerions produire automatiquement ou semi-automatiquement ces vidéos, en couplant nos outils d'analyse et des outils comme ly2video.

Projet réalisée avec le soutien de l'axe « Arts, Sciences et Technologies » (Laurent Grisoni) du projet « Sciences et Cultures du Visuel ».

Véronique Béland. Nous débutons une collaboration avec Véronique Béland, artiste plasticienne, sur « As we are blind », qui est une « installation interactive pour aura et piano mécanique » dont les premières présentations sont prévues fin 2016. V. Béland travaille sur « l'effort pour construire des images claires à partir d'impressions confuses » :

> *« Ma recherche s'est graduellement précisée vers un désir de pointer des processus a priori invisibles ou inaudibles, comme un besoin d'ausculter différents types de silences ou de vides pour en relever le contenu. Par diverses astuces de traduction ou de transcodage, je cherche à créer des points de contact entre le visible et l'invisible, entre l'audible et l'inaudible. »*

Dans cette installation, le spectateur posera sa main sur un capteur mesurant quelques paramètres (conductance, température, sudation et rythme cardiaque), paramètres qui seront transformés en « aura » visuelle ainsi qu'en une musique jouée par un piano mécanique. Richard Groult, Emmanuel Leguy et Sławek Staworko et moi allons ainsi travailler sur la *génération* de musique à partir de l'analyse d'un matériau fourni par le composieur Quentin Denival. Ce projet contiendra une partie technique (vacations à venir en 2016 de Guillaume Libersart, intégration d'outils d'apprentissage et lien avec notre code) et une partie recherche (analyse haut-niveau puis génération sur le matériel musical).

Projet soutenu par un « Bonus recherche Expériences Interactives » Pictanovo pour 2016-17.

12 Bilan

J'ai déjà exposé, dans les chapitres 6 et 10, les perspectives de chacun des deux projets. Travailler sur deux projets aussi différents fait nécessairement se poser la question du focus d'ensemble. Début 2016, je me sens bien plus focalisé que je n'ai pu l'être il y a quelques années, les deux projets étant chacun bien défini et ayant des perspectives passionnantes. J'ai envie de continuer à m'investir scientifiquement dans ces deux projets, même si évidemment leurs situations sont différentes.

L'activité autour de Vidjil, déjà bien mature, a vocation à être un jour pérennisée et donc au moins en partie transférée. Comment continuer les travaux théoriques, et accompagner le développement et l'extension de Vidjil, au-delà de Bonsai et de nos premiers collègues lillois ? Pour l'informatique musicale, l'objectif est à la fois de découvrir de nouveaux algorithmes d'analyse musicale et de mener des projets à la croisée de l'art et de la science. Comment rendre cette activité plus mature et renforcer l'équipe naissante Algomus ? Aux questions scientifiques s'ajoutent des défis de financement et de structuration, avec de nouvelles opportunités rendues possible par la création de notre université, l'Université de Lille, ainsi que notre nouvelle région Nord-Pas-de-Calais–Picardie.

Hématologie et musique, séquences d'ADN ou séquences de notes... dans mon parcours académique, le lien entre ces deux domaines s'est fait sur la conception et l'utilisation d'algorithmique du texte. Dès les années 1990, certaines publications MIR de référence ont été faites par des bioinformaticiens [60]. Chez Algomus, nous continuons, en partie, à nous inspirer de techniques bioinformatiques pour proposer de nouveaux algorithmes d'analyse. Les objets et les finalités étant différents, je ne souhaite cependant pas imaginer de ponts artificiels entre les deux domaines, mais plutôt réfléchir maintenant aux *démarches communes* à ces études transdisciplinaires.

Recherche appliquée. Au fond, derrière des objectifs techniques, quantifiables – compter les globules blancs, analyser les partitions – dans ces deux projets, nous cherchons une meilleure *compréhension* du monde qui nous entoure, que ce soit une réalité biologique (comprendre le système immunitaire) ou artistique et sociale (comprendre la musique). Nous essayons de proposer des modélisations du système immunitaire ou de la structure musicale pour répondre à des questions de recherche propre à chaque domaine. Que ce soit chez Bonsai ou chez Algomus, nous publions une partie de nos travaux dans des revues et conférences du domaine d'application, ce qui demande en particulier des *évaluations* propres à chaque métier.

Au final, l'impact sociétal *appliqué* de mes travaux sera ainsi que notre plateforme Vidjil soit évaluée et utilisée par des hématologues et immunologistes, et que les travaux d'Algomus servent à mieux illustrer, communiquer et interpréter la musique. Dans cet objectif, l'expert du numérique doit aujourd'hui comprendre finement les objets du domaine d'application et proposer des méthodes qui sont idéalement transparentes pour l'expert du domaine... Par nos méthodes, nos développements logiciels et notre travail sur les données, pouvons-nous finalement répondre à des questions appliquées en immunologie tout comme en musique ?

Recherche théorique. J'apprécie toujours de revenir aux structures de données et algorithmes, d'optimiser leur conception comme leur implémentation, et de communiquer ces résultats à la communauté d'algorithmique du texte, même si je n'y consacre aujourd'hui qu'une petite partie de mon temps. Des questions extrêmement appliquées que nous avons sur Vidjil (comparaison et évolution de répertoires) demandent ainsi des avancées théoriques sur l'indexation et la comparaison de séquences recombinées (voir en particulier la thèse de Tatiana Rocher, p. 40). Pour vraiment pouvoir travailler sur les objets théoriques, nous avons besoin de définir très précisément le problème informatique, parfois en simplifiant encore la modélisation.

Cet *impact sociétal fondamental* est tout aussi essentiel. En bioinformatique, depuis le début de ma thèse en 2002, j'ai ainsi mené des travaux fondamentaux sur la comparaison de séquences, le parallélisme et les structures d'ARN, qui même s'il n'ont pas eu d'impact visible, ont approfondi mon expertise, participé à la formation et à la recherche de nouveaux docteurs et préparé le terrain à Vidjil. Du côté d'Algomus, nous sommes pour l'instant plus dans une démarche théorique de modélisation et d'algorithmisation de la musique.

Collectivement, c'est bien la présence de travaux plus fondamentaux réalisés par l'ensemble de la communauté de recherche qui permet que, à moyen et long terme, certains travaux aient un impact sociétal appliqué.

Un travail d'équipe. Les recherches pluridisciplinaires théoriques comme appliquées demandent un effort impossible à faire seul, ne serait-ce que vu l'éventail des connaissances et techniques mobilisées. Ces cinq dernières années, ces projets Vidjil et Algomus ont été avant tout des projets collaboratifs. Cette dimension collaborative de la recherche me plaît énormément, et je suis heureux et fier d'avoir pu travailler et faire travailler des informaticiens avec des hématologues et des musiciens. Au quotidien, ce travail se déroule au sein de deux équipes formidables : Bonsai et Algomus. Leurs membres me surprennent tous les jours par l'apport de leurs compétences complémentaires et leurs questions inattendues.

Enfin, même si j'ai déjà passé un certain temps sur les bancs du conservatoire ou de la fac de médecine, je ne suis pas hématologue ou immunologiste, ni théoricien de la musique... Les collaborations avec des collègues non informaticiens sont essentielles. Depuis son début, fin 2011, le projet Vidjil est nourri par les besoins et les questions des hématologues biologistes. Algomus est plus universitaire. Nous avons commencé à travailler avec des musicologues, et, plus récemment, des artistes. Nous espérons pouvoir, dans les prochaines années, approfondir ce lien entre théorie analytique et algorithmes d'informatique musicale et aspects théorique, pédagogiques, et artistiques. C'est ainsi que je vois la direction de recherche : créer, faire vivre et progresser des projets scientifiques trans-disciplinaires, en s'appuyant sur les compétences internes et externes à nos laboratoires.

13 Bibliographie

13.1 Publications auxquelles j'ai contribué

[1] Mathieu Giraud et Mikaël Salson. Les séquenceurs à haut-débit. https://interstices.info/jcms/int_63223/les-sequenceurs-a-haut-debit. 2011 (→ 17).

[2] Florence Levé, Richard Groult, Guillaume Arnaud, Cyril Séguin, Rémi Gaymay et Mathieu Giraud. Rhythm extraction from polyphonic symbolic music. In : *International Society for Music Information Retrieval Conference (ISMIR 2011)*. 2011, 375–380 (→ 43, 56).

[3] Mathieu Giraud, Richard Groult et Florence Levé. Subject and counter-subject detection for analysis of the Well-Tempered Clavier fugues. In : *International Symposium on Computer Music Modeling and Retrieval (CMMR 2012)*. 2012, 661–673 (→ 43, 57, 64, 76).

[4] Hélène Barucq, Olivier Beaumont, Gérard Berry et al. Eléments pour une auto-appréciation des activités de médiation scientifique. https://www.inria.fr/content/download/37522/741584/version/2/file/2012-mediation-fr.pdf. 2012 (→ 87).

[5] Mathieu Giraud, Richard Groult et Florence Levé. Detecting Episodes with Harmonic Sequences for Fugue Analysis. In : *International Society for Music Information Retrieval Conference (ISMIR 2012)*. 2012 (→ 43, 64, 76).

[6] Mathieu Giraud, Ken Déguernel et Emilios Cambouropoulos. Fragmentations with pitch, rhythm and parallelism constraints for variation matching. In : *International Symposium on Computer Music Multidisciplinary Research (CMMR)*. T. LNCS 8905. 2013, 298–312 (→ 57, 58, 76).

[7] Mathieu Giraud, Mikaël Salson, Marc Duez et al. Suivi de la leucémie résiduelle par séquençage haut-débit. In : *Journées Ouvertes Biologie, Informatique et Mathématiques (JOBIM 2013)*. 2013 (→ 23).

[8] Antoine Rousseau et al. Médiation Scientifique : une facette de nos métiers de la recherche. https://hal.inria.fr/hal-00804915. 2013 (→ 87).

[9] Laurent David, Mathieu Giraud, Richard Groult, Corentin Louboutin et Florence Levé. Vers une analyse automatique des formes sonates. In : *Journées d'Informatique Musicale (JIM 2014)*. 2014 (→ 65, 66, 76, 78).

[10] Mathieu Giraud, Florence Levé, Florent Mercier, Marc Rigaudière et Donatien Thorez. Modeling texture in symbolic data. In : *International Society for Music Information Retrieval Conference (ISMIR 2014)*. 2014 (→ 60, 61, 76, 78).

[11] Mathieu Giraud et Marc Rigaudière. Algorithmes pour l'analyse de la musique tonale. In : *Revue Technique et Science Informatiques* **33**.7-8 (2014), 567–586 (→ 45, 46, 48, 50, 55, 56).

[12] Mathieu Giraud, Mikaël Salson, Marc Duez et al. Fast multiclonal clusterization of V(D)J recombinations from high-throughput sequencing. In : *BMC Genomics* **15**.1 (2014), 409 (→ 19, 23, 24, 29, 33, 40).

[13] Nathalie Grardel, Mikaël Salson, Aurélie Caillault et al. Multiclonal Diagnosis and MRD Follow-up in ALL with HTS Coupled with a Bioinformatic Analysis. In : *Blood* **124**.21 (2014), 1083–1083 (→ 33, 35).

[14] Guillaume Bagan, Mathieu Giraud, Richard Groult et Emmanuel Leguy. Modélisation et visualisation de schémas d'analyse musicale avec music21. In : *Journées d'Informatique Musicale (JIM 2015)*. 2015 (→ 69, 71).

[15] Mathieu Giraud, Richard Groult, Emmanuel Leguy et Florence Levé. Computational Fugue Analysis. In : *Computer Music Journal* **39**.2 (2015) (→ 55, 61–65, 73, 76).

[16] Mathieu Giraud et Sławek Staworko. Modeling Musical Structure with Parametric Grammars. In : *Mathematics and Computation in Music (MCM 2015)*. 2015, 85–96 (→ 66, 67).

[17] Nicolas Guiomard-Kagan, Mathieu Giraud, Richard Groult et Florence Levé. Comparing Voice and Stream Segmentation Algorithms. In : *International Society for Music Information Retrieval Conference (ISMIR 2015)*. 2015, 493–499 (→ 58, 60, 75, 78).

[18] Michaela Kotrova, Katerina Muzikova, Ester Mejstrikova et al. The Predictive Strength of Next Generation Sequencing MRD Detection for Relapse Compared with Current Methods in Childhood ALL. In : *Blood* **126**.8 (2015), 1045–1047 (→ 33, 38).

[19] Yann Ferret, Aurëlie Caillault, Shéhérazade Sebda et al. Multi-loci Diagnosis of Acute Lymphoblastic Leukemia with High-Throughput Sequencing and Bioinformatics Analysis. In : *British Journal of Haematology* (2016) (→ 29, 30, 33, 35).

[20] Mathieu Giraud, Richard Groult et Florence Levé. Computational Analysis of Musical Form. In : *Computational Music Analysis*. Sous la dir. de David Meredith. 2016, 113–136 (→ 55, 69, 74, 75).

[21] Marc Duez, Mathieu Giraud, Ryan Herbert, Tatiana Rocher, Mikaël Salson et Florian Thonier. Vidjil : High-throughput analysis of immune repertoire. In : (en préparation) (→ 27, 31).

[22] Mikaël Salson, Mathieu Giraud, Aurélie Caillault et al. TRG and IgH Follow-Up of Acute Lymphoblastic Leukemia by High-Throughput Sequencing and Bioinformatics Analysis. In : (en préparation) (→ 33, 36, 37).

13.2 Références

[23] Boèce. De institutione musica. VI^{e} siècle (→ 49).

[24] Zarlino. Istitutioni harmoniche. 1558 (→ 49).

[25] Adolph Bernhard Marx. Die Lehre von der musikalischen Komposition (volumes 2 et 3). Breitkopf & Härtel, Leipzig, 1838, 1845 (→ 47).

[26] Ada Lovelace. A Sketch of the Analytical Engine, with Notes by the Translator. In : *Scientific Memoirs* **3** (1843), 666–731 (→ 49).

[27] Carl Czerny. School of Practical Composition. R. Cocks, London, 1848 (→ 47).

[28] André Gedalge. Traité de la fugue. Enoch, Paris, 1901 (→ 47).

[29] Ebenezer Prout. Analysis of J. S. Bach's forty-eight fugues (Das Wohltemperierte Clavier). E. Ashdown, 1910 (→ 73, 75).

[30] Donald Tovey, éd. Forty-Eight Preludes and Fugues, J.-S. Bach. Associated Board of the Royal Schools of Music, 1924 (→ 74).

[31] Heinrich Schenker. Der freie Satz. Universal Edition, 1935 (→ 63).

[32] Olivier Messiaen. Technique de mon language musical. 1944 (→ 47).

[33] Richard Bellman. The theory of dynamic programming. Rapp. tech. P-550. RAND Corporation, 1954 (→ 6).

[34] Quentin R. Nordgren. A Measure of Textural Patterns and Strengths. In : *journal of Music Theory* **4**.1 (1960), 19–31 (→ 60).

[35] Hermann Keller. Das Wohltemperierte Klavier von Johann Sebastiann Bach. Bärenreiter, 1965 (→ 73).

[36] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In : *Soviet Physics Doklady* **10**.8 (1966). original russe publié dans les *Comptes rendus de l'Académie des sciences de l'URSS*, 163(4) : 845–8, 1965, 707–710 (→ 5, 6).

[37] S. B. Needleman et C. D. Wunsch. A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins. In : *Journal of Molecular Biology* **48** (1970), 443–453 (→ 6, 56).

[38] Theodosius Dobzhansky. Nothing in biology makes sense except in the light of evolution. In : *American Biology Teacher* 35 (1973), 125–29 (→ 7).

[39] R. A. Finkel et J. L. Bentley. Quad trees : a data structure for retrieval on composite keys. In : *Acta Informatica* **4**.1 (1974), 1–9 (→ 30).

[40] Alfred V. Aho et Margaret J. Corasick. Efficient string matching : An aid to bibliographic search. In : *Communications of the ACM* **18**.6 (1975), 333–340 (→ 6, 21, 40).

[41] Jonathan Cott. Conversations with Glenn Gould. University of Chicago Press, 1977, traduction Jacques Drillon (1983) (→ 47).

[42] Steven R. Holtzman. A program for key determination. In : *Interface* **6** (1977), 29–56 (→ 61).

[43] Donald Knuth, James H. Morris et Vaughan Pratt. Fast pattern matching in strings. In : *SIAM Journal on Computing* **6**.2 (1977), 323–350 (→ 6).

[44] Jacob Ziv et Abraham Lempel. A universal algorithm for sequential data compression. In : *IEEE Transactions on Information Theory* **23**.3 (1977), 337–343 (→ 40).

[45] Jacob Ziv et Abraham Lempel. Compression of individual sequences via variable-rate coding. In : *IEEE Transactions on Information Theory* **24**.5 (1978), 530–536 (→ 40).

[46] Charles Rosen. Sonata Forms. W. W. Norton, 1980 (→ 48).

[47] T. F. Smith et M. S. Waterman. Identification of common molecular subsequences. In : *Jouranl of Molecular Biolology* **147** (1981), 195–197 (→ 6, 56).

[48] Diana Deutsch, éd. The psychology of music. Academic Press, 1982 (→ 59).

[49] O. Gotoh. An Improved Algorithm for Matching Biological Sequences. In : *Journal of Molecular Biology* **162**.3 (1982), 705–708 (→ 6).

[50] Carol L. Krumhansl et Edward journal Kessler. Tracing the Dynamic Changes in Perceived Tonal Organisation in a Spatial Representation of Musical Keys. In : *Psychological Review* **89**.2 (1982), 334–368 (→ 61).

[51] Fred Lerdhal et Ray Jackendoff. A Generative Theory of Tonal Music. MIT Press, 1983, 1996 (→ 63).

[52] M. Lothaire. Combinatorics on words. Addison-Wesley, 1983 (→ 6).

[53] Susumu Tonegawa. Somatic generation of antibody diversity. In : *Nature* **302**.5909 (1983), 575–581 (→ 12).

[54] Ian Bent et William Drabkin. Analysis. 1987 (→ 46).

[55] Zhu Chen, Denis Le Paslier, Jean Dausset et al. Human T cell gamma genes are frequently rearranged in B-lineage acute lymphoblastic leukemias but not in chronic B cell proliferations. In : *The Journal of experimental medicine* **165**.4 (1987), 1000–1015 (→ 13).

[56] Nicholas Cook. A guide to musical analysis. Oxford University Press, Dent, 1987 (→ 46).

[57] Jean-Jacques Nattiez. Musicologie générale et sémiologie. Christian Bourgeois, 1987 (→ 47).

[58] David Huron. Characterizing musical textures. In : *International Computer Music Conference (ICMC)*. 1989, 131–134 (→ 60).

[59] S. Karlin et S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. In : *Proceedings of the National Academy of Sciences* **87**.6 (1990), 2264–2268 (→ 31).

[60] Marcel Mongeau et David Sankoff. Comparison of Musical Sequences. In : *Computers and the Humanities* **24**.3 (1990), 161–175 (→ 56, 57, 89).

[61] David Cope. Computers and Musical Style. Madison, 1991 (→ 66).

[62] Siglind Bruhn. J. S. Bach's Well-Tempered Clavier. In-depth Analysis and Interpretation. Mainer International, 1993 (→ 73, 75).

[63] Marcel Mesnage. Morphoscope, a computer system for music analysis. In : *Journal of New Music Research* **22**.2 (1993), 119–131 (→ 70).

[64] Maxime Crochemore et Wojciech Rytter. Text Algorithms. Oxford University Press, 1994 (→ 6).

[65] Dan Gusfield. Algorithms on Strings, Trees, and Sequences. Cambridge University Press, 1997 (→ 6).

[66] Jean-Jacques Nattiez. De la sémiologie générale à la sémiologie musicale. L'exemple de la Cathédrale engloutie de Debussy. In : *Protée* (1997), 7–20 (→ 47).

[67] T. Crawford, C. Iliopoulos et R. Raman. String matching techniques for musical similarity and melodic recognition. In : *Computing in Musicology* **11** (1998), 71–100 (→ 57).

[68] H. H. Hoos, K. A. Hamel, K. Renz et J. Kilian. The GUIDO Music Notation Format. – A Novel Approach for Adequately Representing Score-level Music. In : *Int. Computer Music Conference (ICMC 1998)*. 1998, 451–454 (→ 70).

[69] J. L. Hsu, C. C. Liu et A. Chen. Efficient repeating pattern finding in music databases. In : *International Conference on Information and Knowledge Management (CIKM 1998)*. 1998, 281–288 (→ 57).

[70] David Temperley. What's key for key ? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered. In : *Music Perception* **17**.1 (1999), 65–100 (→ 61).

[71] Maxime Crochemore, Christophe Hancart et Thierry Lecroq. Algorithmique du texte. Vuibert, 2001 (→ 6, 25).

[72] David Temperley. The Cognition of Basic Musical Structures. The MIT Press, 2001 (→ 77).

[73] J. J. M. van Dongen, T. Szczepański et H. J. Adriaansen. Immunobiology of leukemia. In : *Leukemia*. Sous la dir. de Greaves M Henderson ES Lister TA. 7th. Philadelphia : WB Saunders, 2002, 85–130 (→ 16).

[74] David Huron. Music Information Processing Using the Humdrum Toolkit : Concepts, Examples, and Lessons. In : *Computer Music Journal* **26**.2 (2002), 11–26 (→ 50, 70, 71).

[75] Olivier Lartillot. Kanthume : un projet d'analyse analogique suivant un modèle cognitif d'induction. In : *2ème colloque international d'épistémologie musicale*. 2002 (→ 58).

[76] Bryan Pardo et William P. Birmingham. Algorithms for Chordal Analysis. In : *Comput Music journal* **26**.2 (2002), 27–49 (→ 61).

[77] J. J. M. van Dongen, A. W. Langerak, M. Brüggemann et al. Design and standardization of PCR primers and protocols for detection of clonal immunoglobulin and T-cell receptor gene recombinations in suspect lymphoproliferations : report of the BIOMED-2 Concerted Action BMH4-CT98-3936. In : *Leukemia* **17**.12 (2003), 2257–317 (→ 12, 15, 17, 18, 41).

[78] Eleonora Market et F. Nina Papavasiliou. V(D)J recombination and the evolution of the adaptive immune system. In : *PLoS Biology* **1**.1 (2003), E16 (→ 12).

[79] Colin Meek et William P Birmingham. Automatic Thematic Extractor. In : *journal of Intelligent Information Systems* **21**.1 (2003), 9–33 (→ 58).

[80] Francois Pachet. The continuator : Musical interaction with style. In : *Journal of New Music Research* **32**.3 (2003), 333–341 (→ 78).

[81] Esko Ukkonen, Kjell Lemström et Veli Mäkinen. Geometric algorithms for transposition invariant content based music retrieval. In : *International Conference on Music Information Retrieval (ISMIR 2003)*. 2003, 193–199 (→ 57).

[82] Gérard Assayag et Shlomo Dubnov. Using Factor Oracles for Machine Improvisation. In : *Soft Comput.* **8**.9 (2004), 604–610 (→ 78).

[83] Marc Chemillier. Grammaires, automates et musique. In : *Informatique musicale*. Hermès, Lavoisier, 2004, 195–230 (→ 66).

[84] Raphaël Clifford et Costas S. Iliopoulos. Approximate string matching for music analysis. In : *Soft. Comput.* **8**.9 (2004), 597–603 (→ 57).

[85] Tuomas Eerola et Petri Toiviainen. MIR in Matlab : the MIDI toolbox. In : *International Conference on Music Information Retrieval (ISMIR 2004)*. 2004 (→ 70).

[86] Neil C. Jones et Pavel A. Pevzner. An introduction to bioinformatics algorithms. MIT Press, 2004 (→ 6, 25).

[87] M. M. Souto-Carneiro, N. S. Longo, D. E. Russ, H. Sun et P. E. Lipsky. Characterization of the human Ig heavy chain antigen binding complementarity determining region 3 using a newly developed software algorithm, JoinSolver. In : *The Journal of Immunology* **172**.11 (2004), 6790 (→ 21).

[88] Mehdi Yousfi Monod, Véronique Giudicelli, Denys Chaume et Marie-Paule Lefranc. IMGT/JunctionAnalysis : the first tool for the analysis of the immunoglobulin and T cell receptor complex V-J and V-D-J JUNCTIONs. In : *Bioinformatics* **20 Suppl 1** (2004), i379–85 (→ 20).

[89] Elaine Chew et Xiaodan Wu. Separating voices in polyphonic music : A contig mapping approach. In : *International Symposium on Computer Music Modeling and Retrieval (CMMR 2005)*. 2005, 1–20 (→ 58).

[90] Nicholas Cook. Towards the compleat musicologist. In : *International Conference on Music Information Retrieval (ISMIR 2005)*. 2005 (→ 50, 77).

[91] Célestin Deliège. Sources et ressources d'analyses musicales. Mardaga, 2005 (→ 50).

[92] Véronique Giudicelli, Denys Chaume et Marie-Paule Lefranc. IMGT/GENE-DB : a comprehensive database for human and mouse immunoglobulin and T cell receptor genes. In : *Nucleic Acids Research* **33**.S1 (2005), D256–D261 (→ 28).

[93] Gunter Kerst, Hermann Kreyenberg, Carmen Roth et al. Concurrent detection of minimal residual disease (MRD) in childhood acute lymphoblastic leukaemia by flow cytometry and real-time PCR. In : *British Journal of Haematology* **128**.6 (2005), 774–782 (→ 16).

[94] Pei-Hsuan Weng et Arbee L. P. Chen. Automatic Musical Form Analysis. In : *International Conf. on Digital Archive Technologies (ICDAT 2005)*. 2005 (→ 63).

[95] Emilios Cambouropoulos. Musical parallelism and melodic segmentation. In : *Music Perception* **23**.3 (2006), 249–268 (→ 57).

[96] Darrell Conklin et Christina Anagnostopoulou. Segmental pattern discovery in music. In : *INFORMS journal on Computing* **18**.3 (2006) (→ 57).

[97] Masataka Goto. AIST Annotation for the RWC Music Database. In : *International Conference on Music Information Retrieval (ISMIR 2006)*. 2006, 359–360 (→ 74).

[98] James Hepokoski et Warren Darcy. Elements of Sonata Theory : Norms, Types, and Deformations in the Late-Eighteenth-Century Sonata. Oxford University Press, 2006 (→ 48, 65, 66, 78).

[99] Marcel Mesnage et André Riotte. Formalismes et modèles musicaux (volumes 1 et 2). Musique/Sciences, Delatour, 2006 (→ 66).

[100] L. Ohm-Laursen, M. Nielsen, S. R. Larsen et T. Barington. No evidence for the use of DIR, D-D fusions, chromosome 15 open reading frames or VH replacement in the peripheral repertoire was found on application of an improved algorithm, JointML, to 6329 human immunoglobulin H rearrangements. In : *Immunology* **2**.119 (2006), 265–77 (→ 21).

[101] Sven Ahlbäck. Melodic Similarity as a Determinant of Melody Structure. In : *Musicae Scientiae* **Discussion Forum 4A** (2007), 235–280 (→ 57).

[102] Bruno A. Gaëta, Harald R. Malming, Katherine J. L. Jackson, Michael E. Bain, Patrick Wilson et Andrew M. Collins. iHMMunealign : hidden Markov model-based alignment and identification of germline genes in rearranged immunoglobulin gene sequences. In : *Bioinformatics* **23**.13 (2007), 1580–1587 (→ 21).

[103] Masatoshi Hamanaka, Keiji Hirata et Satoshi Tojo. FATTA : Full Automatic Time-Span Tree Analyzer. In : *International Computer Music Conference (ICMC 2007)*. 2007 (→ 63).

[104] Plácido R. Illescas, David Rizo et José M. Iñesta. Harmonic, melodic, and functional automatic analysis. In : *International Computer Music Conference (ICMC 2007)*. 2007, 165–168 (→ 61).

[105] Olivier Lartillot. Motivic Pattern Extraction in Symbolic Domain. In : *Intelligent Music Information Systems : Tools and Methodologies*. IGI Global, 2007 (→ 58, 59).

[106] Soren T. Madsen et Gerhard Widmer. Key-Finding with Interval Profiles. In : *International Computer Music Conference (ICMC 2007)*. 2007 (→ 61).

[107] David Temperley. Music and probability. The MIT Press, 2007 (→ 78).

[108] Rainer Typke. "Music retrieval based on melodic similarity." Thèse de doct. Univ. Utrecht, 2007 (→ 57).

[109] Donald Adjeroh, Timothy Bell et Amar Mukherjee. The Burrows-Wheeler Transform : : Data Compression, Suffix Arrays, and Pattern Matching. Springer, 2008 (→ 6, 40).

[110] Xavier Brochet, Marie-Paule Lefranc et Véronique Giudicelli. IMGT/V-QUEST : the highly customized and integrated system for IG and TR standardized V-J and V-D-J sequence analysis. In : *Nucleic Acids Research* **36**.Web Server issue (2008), W503–8 (→ 20, 21, 31).

[111] Daniel G. Brown. Bioinformatics Algorithms : Techniques and Applications. In : 2008. Chap. A survey of seeding for sequence alignment, 126–152 (→ 24, 40).

[112] Pierre Couprie. iAnalyse : un logiciel d'aide à l'analyse musicale. In : *Journées d'Informatique Musicale (JIM 2008)*. 2008, 115–121 (→ 70).

[113] J. S. Downie. The music information retrieval evaluation exchange (2005-2007) : A window into music information retrieval research. In : *Acoustical Science and Technology* **29**.4 (2008), 247–255 (→ 49).

[114] Hanna M Lukashevich. Towards Quantitative Measures of Evaluating Song Segmentation. In : *International Conference on Music Information Retrieval (ISMIR 2008)*. 2008, 375–380 (→ 58).

[115] Hanna M Lukashevich. Towards Quantitative Measures of Evaluating Song Segmentation. In : *International Conference on Music Information Retrieval (ISMIR 2008)*. 2008, 375–380 (→ 75).

[116] L.J.P. van der Maaten et G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. In : *Journal of Machine Learning Research* **9** (2008), 2579–2605 (→ 31).

[117] Dimitrios Rafailidis, Alexandros Nanopoulos, Yannis Manolopoulos et Emilios Cambouropoulos. Detection of Stream Segments in Symbolic Musical Data. In : *International Conference on Music Information Retrieval (ISMIR 2008)*. 2008 (→ 58, 59, 78).

[118] Matthias Robine, Thomas Rocher et Pierre Hanna. Improvements of Key-Finding Methods. In : *International Computer Music Conference (ICMC 2008)*. 2008 (→ 61).

[119] Julien Allali, Pascal Ferraro, Pierre Hanna, Costas Illiopoulos et Matthias Robine. Toward a General Framework for Polyphonic Comparison. In : *Fundamenta Informaticae* **97** (2009), 331–346 (→ 57).

[120] Scott D. Boyd, Eleanor L. Marshall, Jason D. Merker et al. Measurement and clinical monitoring of human lymphocyte clonality by massively parallel VDJ pyrosequencing. In : *Science Translational Medicine* **1**.12 (2009), 12ra23 (→ 17).

[121] Rémy Campos et Nicolas Donin. L'analyse musicale : une pratique et son histoire. Droz, 2009 (→ 46).

[122] William E. Caplin, James Hepokoski et James Webster. Musical Form, Forms & Formenlehre – Three Methodological Reflections. Leuven University Press, 2009 (→ 48).

[123] Claude Charlier. Pour une lecture alternative du Clavier bien tempéré. Jacquart, 2009 (→ 73).

[124] J. Douglas Freeman, René L. Warren, John R. Webb, Brad H. Nelson et Robert A. Holt. Profiling the T-cell receptor beta-chain repertoire by massively parallel sequencing. In : *Genome Research* **19**.10 (2009), 1817–24 (→ 17).

[125] Christophe Guillotel-Nothmann. Telos : logiciel d'analyse harmonique et contrapuntique. `http://www.guillotel-nothmann.com/publications/TelosDoc121108.pdf`. 2009 (→ 77).

[126] Alan Marsden. "What was the question ?" : music analysis and the computer. In : *Modern Methods for Musicology*. Crawford, Gibson (eds.), Farnham : Ashgate, 2009, 137–147 (→ 50, 77).

[127] Alexandre Passos, Marcos Sampaio, Pedro Kröger et Givaldo de Cidra. Functional Harmonic Analysis and Computational Musicology in Rameau. In : *Brazilian Symposium on Computer Music (SBCM 2009)*. 2009, 207–210 (→ 61).

[128] Thomas Rocher, Matthias Robine, Pierre Hanna et Robert Strandh. Dynamic Chord Analysis for Symbolic Music. In : *International Computer Music Conference (ICMC 2009)*. 2009 (→ 77).

[129] Joshua A Weinstein, Ning Jiang, Richard A White 3rd, Daniel S Fisher et Stephen R Quake. High-throughput sequencing of the zebrafish antibody repertoire. In : *Science* **324**.5928 (2009), 807–10 (→ 17).

[130] Christina Anagnostopoulou et Chantal Buteau. Can computational music analysis be both musical and computational ? In : *journal of Mathematics and Music* **4**.2 (2010), 75–83 (→ 50).

[131] Darrell Conklin et Mathieu Bergeron. Discovery of Contrapuntal Patterns. In : *International Society for Music Information Retrieval Conference (ISMIR 2010)*. 2010, 201–206 (→ 58, 59).

[132] Pierre Couprie. Utilisations avancées du logiciel iAnalyse pour l'analyse musicaler. In : *Journées d'Informatique Musicale (JIM 2010)*. 2010 (→ 70).

[133] Michael Scott Cuthbert et Christopher Ariza. music21 : A Toolkit for Computer-Aided Musicology and Symbolic Music Data. In : *International Society for Music Information Retrieval Conference (ISMIR 2010)*. 2010 (→ 70, 71).

[134] Nicolas Donin, Samuel Goldszmidt et Jacques Theureau. Instrumenter les opérations d'écoute analytique ? Un bilan du projet "Écoutes signées" (2003-2006). In : *Journées d'Informatique Musicale (JIM 2010)*. 2010, 165–174 (→ 70).

[135] Emmanuel Favreau, Yann Geslin et Adrien Lefèvre. L'acousmographe 3. In : *Journées d'Informatique Musicale (JIM 2010)*. 2010 (→ 70).

[136] D. Fober, C. Daudin, Y. Orlarey et S. Letz. Partitions musicales augmentées. In : *Journées d'Informatique Musicale (JIM 2010)*. 2010 (→ 70).

[137] Katherine J. L. Jackson, Scott Boyd, Bruno A. Gaëta et Andrew M. Collins. Benchmarking the performance of human antibody gene alignment utilities using a 454 sequence dataset. In : *Bioinformatics* **26**.24 (2010), 3129–30 (→ 21).

[138] Alan Marsden. Schenkerian analysis by computer. In : *Journal of New Music Research* **39**.3 (2010), 269–289 (→ 63).

[139] M. Mauch, C. Cannam, M. Davies et al. OMRAS2 Metadata Project 2009. In : *International Society for Music Information Retrieval Conference (ISMIR 2010)*. 2010 (→ 74).

[140] S. Munshaw et T.B. Kepler. SoDA2 : a Hidden Markov Model approach for identification of immunoglobulin rearrangements. In : *Bioinformatics* **26**.7 (2010), 867–872 (→ 21).

[141] Ramy Arnaout, William Lee, Patrick Cahill et al. High-Resolution Description of Antibody Heavy-Chain Repertoires in Humans. In : *PLoS ONE* **6**.8 (2011), e22365 (→ 21).

[142] Jean Bresson, Carlos Agon et Gérard Assayag. OpenMusic : visual programming environment for music composition, analysis and research. In : *International Conference on Multimedia*. 2011, 743–746 (→ 70).

[143] Asako Ishigaki, Masaki Matsubara et Hiroaki Saito. Prioritized contig combining to segragate voices in polyphonic music. In : *Sound and Music Computing Conference (SMC 2011)*. T. 119. 2011 (→ 58).

[144] Marie-Paule Lefranc. IMGT, the International ImMunoGeneTics Information System. In : *Cold Spring Harbor Protocols* **2011**.6 (2011), pdb–top112 (→ 20, 21).

[145] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure et J. S. Downie. Design and creation of a large-scale database of structural annotations. In : *International Society for Music Information Retrieval Conference (ISMIR 2011)*. 2011 (→ 74).

[146] Dmitri Tymoczko. A geometry of music. Oxford University Press, 2011 (→ 52).

[147] A. Volk, F. Wiering et P. van Kranenburg. Unfolding the potential of computational musicology. In : *International Conference on Informatics and Semiotics in Organisations (ICIS 2011)*. 2011, 137–144 (→ 50, 78).

[148] Eltaf Alamyar, Véronique Giudicelli, Shuo Li, Patrice Duroux et Marie-Paule Lefranc. IMGT/HighV-QUEST : the IMGT® web portal for immunoglobulin (IG) or antibody and T cell receptor (TR) analysis from NGS high throughput and deep sequencing. In : *Immunome Research* **8**.1 (2012) (→ 20, 21).

[149] Jennifer Benichou, Rotem Ben-Hamo, Yoram Louzoun et Sol Efroni. Rep-Seq : uncovering the immunological repertoire through next-generation sequencing. In : *Immunology* **135**.3 (2012), 183–91 (→ 17, 21).

[150] Frédéric Bimbot, Emmanuel Deruty, Gabriel Sargent et Emmanuel Vincent. Semiotic Structure Labeling of Music Pieces : Concepts, Methods and Annotation Conventions. In : *International Society for Music Information Retrieval Conference (ISMIR 2012)*. 2012, 235–240 (→ 74).

[151] Jean Bresson et C. Pérez-Sancho. New Framework for Score Segmentation and Analysis in OpenMusic. In : *Sound and Music Computing Conference (SMC 2012)*. 2012, 506–513 (→ 70).

[152] Monika Brüggemann, Thorsten Raff et Michael Kneba. Has MRD monitoring superseded other prognostic factors in adult ALL ? In : *Blood* **120**.23 (2012), 4470–4481 (→ 16).

[153] Pierre Couprie. EAnalysis : aide à l'analyse de la musique électroacoustique. In : *Journées d'Informatique Musicale (JIM 2012)*. 2012, 183–189 (→ 70).

[154] T. Kalina, J. Flores-Montero, V. H. J. van der Velden et al. EuroFlow standardization of flow cytometer instrument settings and immunophenotyping protocols. In : *Leukemia* **26**.9 (2012), 1986–2010 (→ 15).

[155] Anton W. Langerak et Jacques J. M. van Dongen. Multiple clonal Ig/TCR products : implications for interpretation of clonality findings. In : *Journal of Hematopathology* **5**.1-2 (2012), 35–43 (→ 13, 14).

[156] Louis Bigo, Jean-Louis Giavitto, Moreno Andreatta, Olivier Michel et Antoine Spicher. Computation and Visualization of Musical Structures in Chord-Based Simplicial Complexes. In : *Mathematics and Computation in Music (MCM 2013)*. 2013, 38–51 (→ 77).

[157] Dmitriy A Bolotin, Mikhail Shugay, Ilgar Z Mamedov et al. MiTCR : software for T-cell receptor sequencing data analysis. In : *Nature Methods* **10** (2013), 813–814 (→ 22).

[158] Michael Clarke, Frédéric Dufeu et Peter Manning. TIAALS : A new generic set of tools for the interactive aural analysis of electroacoustic music. In : *Electroacoustic Music Studies Network (EMS 2013)*. 2013 (→ 70).

[159] Brandon J. DeKosky, Gregory C. Ippolito, Ryan P. Deschner et al. High-throughput sequencing of the paired human immunoglobulin heavy and light chain repertoire. In : *Nature Biotechnology* **31**.2 (2013), 166–169 (→ 20).

[160] Nicolas Donin. Manières d'écouter des sons. Quelques aspects du projet Écoutes signées (Ircam). In : *DEMéter* (2013) (→ 46, 70).

[161] Simone Faro et Thierry Lecroq. The Exact Online String Matching Problem : A Review of the Most Recent Results. In : *ACM Computing Surveys* **45**.2 (2013), 1–42 (→ 6).

[162] Niclas Thomas, James Heather, Wilfred Ndifon, John Shawe-Taylor et Benjamin Chain. Decombinator : a tool for fast, efficient gene assignment in T-cell receptor sequences using a finite state machine. In : *Bioinformatics* **29**.5 (2013), 542–550 (→ 21).

[163] Jian Ye, Ning Ma, Thomas L. Madden et James M. Ostell. IgBLAST : an immunoglobulin variable domain sequence analysis tool. In : *Nucleic Acids Research* **41** (2013). doi :10.1093/nar/gkt382, W34–W40 (→ 21, 31).

[164] Frédéric Bimbot, Gabriel Sargent, Emmanuel Deruty, Corentin Guichaoua et Emmanuel Vincent. Semiotic Description of Music Structure : an Introduction to the Quaero/Metiss Structural Annotations. In : *International Conference on Semantic Audio (AES 2014)*. 2014, P1–1 (→ 74).

[165] Rajat K. De et Namrata Tomar, éds. Immunoinformatics. Springer, 2014 (→ 20).

[166] Eric J. Humphrey, Justin Salamon, Oriol Nieto, Jon Forsyth, Rachel M. Bittner et Juan P. Bello. JAMS : A JSON Annotated Music Specification for Reproducible MIR Research. In : *International Society for Music Information Retrieval Conference (ISMIR 2014)*. 2014 (→ 71).

[167] Barbera van Schaik, Paul Klarenbeek, Marieke Doorenspleet et al. Discovery of Invariant T Cells by Next-Generation Sequencing of the Human TCR α-Chain Repertoire. In : *The Journal of Immunology* **193**.10 (2014), 5338–5344 (→ 20).

[168] Namrata Tomar et RajatK. De. Immunoinformatics : A Brief Review. In : *Immunoinformatics*. Sous la dir. de Namrata Tomar et Rajat K. De. T. 1184. Methods in Molecular Biology. 2014, 23–55 (→ 20).

[169] Jason A. Vander Heiden, Gur Yaari, Mohamed Uduman et al. pRESTO : a toolkit for processing high-throughput sequencing raw reads of lymphocyte receptor repertoires. In : *Bioinformatics* **30**.13 (2014), 1930–1932 (→ 34).

[170] Xi Yang, Di Liu, Na Lv et al. TCRklass : A New K-String-Based Algorithm for Human and Mouse TCR Repertoire Characterization. In : *Journal of Immunology* **194**.1 (2014) (→ 22).

[171] Jiajie Zhang, Kassian Kobert, Tomáš Flouri et Alexandros Stamatakis. PEAR : a fast and accurate Illumina Paired-End reAd mergeR. In : *Bioinformatics* **30**.5 (2014), 614–620 (→ 34).

[172] Linus Backert et Oliver Kohlbacher. Immunoinformatics and epitope prediction in the age of genomic medicine. In : *Genome Medicine* **7**.1 (2015), 119 (→ 20).

[173] Simone Becattini, Daniela Latorre, Federico Mele et al. Functional heterogeneity of human memory CD4+ T cell clones primed by pathogens or vaccines. In : *Science* **347**.6220 (2015), 400–406 (→ 20).

[174] Dmitriy A. Bolotin, Stanislav Poslavsky, Igor Mitrophanov et al. MiXCR : software for comprehensive adaptive immunity profiling. en. In : *Nature Methods* **12**.5 (2015), 380–381 (→ 22).

[175] Henrike J. Fischer, Jens van den Brandt, Thomas Lingner et al. Modulation of CNS autoimmune responses by CD8+ T cells coincides with their oligoclonal expansion. In : *Journal of Neuroimmunology* (2015), 476231 (→ 33, 34).

[176] Bryan Howie, Anna M. Sherwood, Ashley D. Berkebile et al. High-throughput pairing of T cell receptor α and β sequences. In : *Science Translational Medicine* **7**.301 (2015), 301ra131–301ra131 (→ 20).

[177] Sally A. Hunsucker, Colleen S. McGary, Benjamin G. Vincent et al. Peptide/MHC Tetramer–Based Sorting of CD8+ T Cells to a Leukemia Antigen Yields Clonotypes Drawn Nonspecifically from an Underlying Restricted Repertoire. In : *Cancer immunology research* **3** (2015), 228–235 (→ 20).

[178] Leon Kuchenbecker, Mikalai Nienen, Jochen Hecht et al. IMSEQ – a fast and error aware approach to immunogenetic sequence analysis. en. In : *Bioinformatics* **31**.18 (2015), btv309 (→ 22).

[179] Ralf A. Linker, De-Hyung Lee, Anne-Christine Flach et al. Thymocyte-derived BDNF influences T-cell maturation at the DN3/DN4 transition stage. In : *European Journal of Immunology* **45**.5 (2015), 1326–1338 (→ 33, 34).

[180] Markus Neuwirth et Pieter Bergé, éds. What Is a Cadence ? : Theoretical and Analytical Perspectives on Cadences in the Classical Repertoire. Leuven University Press, 2015 (→ 77).

[181] William H. Robinson. Sequencing the functional antibody repertoire – diagnostic and therapeutic discovery. In : *Nature Reviews Rheumatology* **11**.3 (2015), 171–182 (→ 17).

[182] Qian Zhang, Qingzhu Jia, Tianxing Deng, Bo Song et Longkun Li. Heterogeneous expansion of CD4+ tumor-infiltrating T-lymphocytes in clear cell renal cell carcinomas. In : *Biochemical and biophysical research communications* **458**.1 (2015), 70–76 (→ 20).

[183] Nikos Darzentas et al. ARReST – Antigen Receptors Research Tool. `http://tools.bat.infspire.org/arrest` (→ 30, 42).

[184] Bagaev Dmitry et al. VDJviz. `http://vdjviz.milaboratory.com` (→ 30).

[185] INScore – an Interactive Augmented Music Score. `http://inscore.sourceforge.net/` (→ 70).

[186] LilyPond – Music notation for everyone. `http://www.lilypond.org/` (→ 70).

[187] ly2video. `https://github.com/aspiers/ly2video` (→ 72).

[188] David Nalesnik. FrameEngraver (→ 72).

[189] VexFlow – HTML5 Music Engraving. `http://www.vexflow.com/` (→ 70).

14 Figures et tables

0.1 Début du manuscrit de la fugue en $Mi\flat$ majeur BWV 852 de J.-S. Bach 2

1.1 Distance d'édition et code correcteur d'erreurs 6
1.2 Alignement de mots 7
1.3 Thèmes de recherche et doctorants encadrés. 7

2.1 L'immunité adaptative, coopération entre lymphocytes T, B, et d'autres cellules . 11
2.2 Structure d'un anticorps, chaînes lourdes et légères 12
2.3 Recombinaison V(D)J 12
2.4 Exemple de recombinaison VDJ sur le locus IgH 13
2.5 Développement lymphoïde et recombinaison des locus pour les lympohcytes B et T 14
2.6 Diversité du répertoire immunologique des lymphocytes B 15
2.7 Leucémie aiguë lymphoblastique (LAL) 15
2.8 Exemples de *reads* en sortie d'un séquenceur haut-débit lors d'une étude de Rep-Seq 16
2.9 Stratégies pour le Rep-Seq 18

3.1 Un diagnostic immédiat du système immunitaire ? 19
3.2 Méthodes bioinformatiques pour le Rep-Seq 21
3.3 Recherche en temps $O(\ell)$ du meilleur palier (i, j) dans lequel se trouve la jonction V-J 24
3.4 Localisation approximative de recombinaison VJ à base de k-mots 24
3.5 Détermination de la recombinaison VDJ d'une séquence par programmation dynamique (phase 2) 25
3.6 Recombinaisons exceptionnelles dans le locus TRδ 26

4.1 Traitement d'une read dans Vidjil (phase 1) et raisons de non-segmentation 28
4.2 Comparaison de la localisation du centre de la fenêtre par la phase 1 de Vidjil avec les résultats de IgBlast et IMGT/HighV-QUEST 29
4.3 Évaluation des désignations V(D)J faites par IMGT/V-QUEST, IgBlast et la phase 2 de Vidjil 30
4.4 Architecture de la plateforme Vidjil 31
4.5 Quelques séquences manuellement annotées (`should-vdj.fa`) 32

5.1 Résumé du protocole mis en place à Lille pour l'analyse des échantillons de diagnostic et de rechute des LAL 35
5.2 Suivi de leucémie aiguë 37

5.3 Stratification de patients atteints de LAL réalisée par la mesure de diversité $\rho_{c/r}$. 38

7.1 Motifs dans le thème de *la Cane de Jeanne*, de Georges Brassens 45
7.2 Partition de l'Oiseau de Feu de Stravinsky annotée par la pianiste Lydia Jardon . 46
7.3 Encodages informatiques d'une partition musicale : MIDI, **kern, MEI 51

8.1 Schéma de l'analyse de la fugue en *Do* mineur BWV 847 de Jean-Sébastien Bach . 55
8.2 Algorithme de Mongeau-Sankoff et variantes 57
8.3 Alignement entre motifs « *a* » et « *b* » du thème de *la Cane de Jeanne*, de Georges Brassens 57
8.4 Correspondance entre le thème et la variation mineure de Andante grazioso de la sonate pour piano numéro 11 de Mozart (K 331) 58
8.5 Analyse motivique de l'invention à deux voix BWV 775 de J.-S. Bach 59
8.6 Détection d'un « motif contrapuntique » élémentaire dans un corpus de 185 chorals de J.-S. Bach 59
8.7 Détection de strates polyphoniques dans l'introduction de la sonate op. 31 no 3 de Beethoven 59
8.8 Séparation de voix 60
8.9 Textures et mouvements parallèles 60
8.10 Détection de cadences 62
8.11 Début de la fugue en *Do* mineur BWV 847 de J.-S. Bach 64
8.12 Modèle de Markov caché utilisé pour prédire la structure de fugues 65
8.13 Structure d'ensemble d'une forme sonate 66
8.14 Recherche du couple exposition/réexposition dans le premier mouvement du quatuor à cordes K. 157 n° 4 de W. A. Mozart 66
8.15 Modélisation par grammaire paramétrique de la structure musicale 67

9.1 Extrait d'une vidéo produite par ly2video de la fugue en *Do* mineur BWV 847 . . 69
9.2 Modélisation et visualisation de schémas d'analyse avec music21 71
9.3 Extrait de la partition analysée de la fugue en *Do*♯ majeur BWV 848 de J.-S. Bach 72
9.4 Visualisation interactive de la fugue en *Si*♭ majeur BWV 866 de J.-S. Bach 72
9.5 Les huit sujets des fugues du premier livre du *Clavier bien tempéré* de Bach où au moins deux sources ont une définition différente du sujet 73
9.6 Comparaison d'annotations de référence 74
9.7 Annotations de référence sur la fugue en *Si*♭ majeur BWV 866 de J.-S. Bach . . . 75
9.8 Analyses de références produites par Algomus et ses collaborateurs 76

11.1 Actions de médiation scientifique 83
11.2 Bioinformaticiens et médiateurs ayant animé l'atelier des « puzzles du génome » . 84
11.3 Les puzzles du génome : assemblage d'ADN, structure d'ARN 85
11.4 Prototype des « tapis musicaux », lors de la fête de la science 2014 à Amiens. . . . 86
11.5 Mozart et les fonctions harmoniques 88

Comparer, et plus généralement traiter, analyser ou indexer les séquences de caractères constitue le champ de recherche de *l'algorithmique du texte*. Chercheur CNRS en informatique depuis fin 2006 dans le laboratoire LIFL, maintenant CRIStAL (UMR 9189, Université de Lille), Mathieu Giraud a eu son parcours académique rythmé par les comparaisons de séquences. Ce manuscrit d'habilitation décrit les deux projets dans lesquels il s'est investi les cinq dernières années.

Au sein de l'équipe Bonsai, commune avec le centre Inria Lille, Mathieu mène avec Mikaël Salson un projet de bioinformatique appliqué à l'hématologie et l'immunologie sur l'analyse des populations de lymphocytes par leurs recombinaisons V(D)J *(« Compter les globules blancs »)*. Débuté par une collaboration avec des collègues bioinformaticiens et hématologues de l'hôpital de Lille, ce projet combine algorithmique pour l'immunologie et l'hématologie, développement logiciel et applications fondamentales et cliniques. Le logiciel Vidjil conçu par Mathieu et ses collègues est utilisé régulièrement par plusieurs laboratoires en France et à l'étranger, dont, en situation de routine, à l'hôpital de Lille.

Mathieu dirige aussi un projet d'informatique musicale *(« Analyser les partitions »)*. Les humanités numériques lient les méthodes informatiques au patrimoine culturel et à la recherche en sciences humaines et sociales. Est-ce qu'un ordinateur peut *comprendre* la musique ? L'équipe émergente Algomus, répartie entre les laboratoires MIS (Amiens, Univ. Picardie Jules Verne) et CRIStAL, rassemble expertise musicologique et compétences algorithmiques pour proposer des méthodes analysant les partitions musicales – comparaison de motifs et d'accords, détection de textures, analyse de formes. Algomus mène des collaborations pluridisciplinaires avec des musicologues, des professeurs de musique et des artistes et réalise des projets combinant science et art.

Ce manuscrit se conclut par la description d'actions de médiation scientifique et artistique.

Research in *text algorithmics* focuses on comparing, analyzing, handling or indexing sequences of characters. Working since 2016 as a CNRS researcher in computer science in the LIFL laboratory, now CRIStAL (UMR 9189, Université de Lille), Mathieu Giraud plays with sequence comparisons. This habilitation thesis focuses on two projects he worked on the last five years.

In the Bonsai team, joint with Inria, Mathieu leads with Mikaël Salson a project on bioinformatics applied to hematology and immunology on the analysis of lymphocytes through their V(D)J recombinations *(« Counting white blood cells »)*. This project, started by a collaboration with bioinformaticians and hematologists in the Lille hospital, combines research in algorithmics for immunology and hematology, software engineering and fundamental and clinical applications. The Vidjil software developed by Mathieu and his colleagues is now used by several labs in France and worldwide, sometimes in a routine hospital practice as in the Lille hospital.

Mathieu also leads a projet on computer music *(« Analyzing music scores »)*. Digital humanities link computational methods to cultural heritage and humanities research. Can computers *understand* music ? The Algomus emergent team, shared between the CRIStAL and the MIS (Université de Picardie Jules Verne, Amiens) labs, combines musicological knowledge and computer science methods to propose algorithms analyzing music scores – with focus on patterns, chords and chord progressions, music texture and high-level structure of music. Algomus collaborates with music theorists, music teachers and artists, and contributes to science and art projects.

This thesis concludes by the description of some popular science events.