

# Étude d'algorithmes pour la détection de la tonalité de fichiers musicaux et implémentation en Clojure

Antoine Passemiers

Université Libre de Bruxelles  
apassemi@ulb.ac.be

## Résumé

Le projet consiste en la discussion de différents algorithmes relatifs à la détection automatisée de tonalité de fichiers musicaux, l'implémentation de ceux-ci en Python, ainsi qu'une réflexion sur la comptabilité de ces derniers avec le paradigme fonctionnel. Le choix de l'algorithme a concevoir selon l'approche fonctionnelle sera basé sur des critères de rapidité d'exécution et de précision de la détection. L'algorithme final sera alors implémenté en Clojure.

## Introduction

La tonalité d'une oeuvre musicale se caractérise par l'ensemble des sons formant une même gamme diatonique. A la différence de la gamme, où les sons se succèdent de façon contigüe, la tonalité (ou ton) regroupe des sons qui peuvent être disjoints et/ou superposés (Danhauser, 1929). En conséquence, nous nous intéressons à l'analyse de mélodies polyphoniques, où plusieurs notes peuvent être jouées en même temps.

En particulier, nous allons nous pencher sur deux catégories d'algorithmes : ceux basés sur des modèles cognitifs, et ceux incluant des notions d'apprentissage automatique. Les premiers tentent d'intégrer au mieux les connaissances de la théorie musicale et reposent sur la façon dont les personnes reconnaissent les différentes tonalités, alors que les autres utilisent l'inférence statistique pour déterminer celles-ci.

L'approche cognitive utilisée pour la détection de la tonalité repose en partie sur la solution proposée par Ibrahim Sha'ath lors de la conception du logiciel KeyFinder (Sha'ath, 2011). La précision de la détection est évaluée à l'aide d'une base de données, constituée de 250 fichiers musicaux au format wav, dont les tonalités sont connues et inscrites dans un fichier csv. Ces fichiers font partie de ceux utilisés par Ibrahim Sha'ath dans le cadre de sa recherche.

Pour ce qui est de la partie apprentissage automatique, les méthodes présentées seront principalement en lien avec les modèles de Markov cachés. Leur évaluation se fera en divisant la base de données en un jeu de

données d'apprentissage et un jeu de données de validation (respectivement 60 % et 40 % du jeu de données d'origine). Ce mini-mémoire se voulant concis et centré sur les objectifs décrits, le lecteur est supposé déjà disposer de connaissances suffisantes en apprentissage automatique, en théorie musicale et en traitement logiciel de signaux.

TODO : (Pauws, 2004) (Takeuchi, 1994)

## Considérations théoriques

**Modèle cognitif** Le signal audio est premièrement extrait du fichier wav, puis la moyenne entre les deux canaux est effectuée si le fichier a été enregistré en stéréo. En effet il n'est pas nécessaire de prendre en compte le panoramique puisque celui-ci n'a que peu d'influence sur la mélodie dans le domaine spectral. Étant donné que les notes jouées sont uniquement caractérisées par leur fréquence fondamentale, il n'est pas nécessaire de considérer l'entière du spectre du fichier musical. De fait, la fréquence d'échantillonnage est abaissée à un dixième de la fréquence standard (4410 Hz), mais ce sous-échantillonnage est susceptible de provoquer des phénomènes d'aliasing. Contrairement à la solution de Sha'ath, qui gère les problèmes d'aliasing sonore par l'application d'un filtre passe-bas, une approche plus simpliste et plus rapide se limiterait à l'application d'une moyenne mobile sur une fenêtre glissante de taille arbitraire. L'avantage de la méthode est de bénéficier d'effets passe-bas sans devoir effectuer de calculs dans le domaine fréquentiel. Pour ce qui est de la taille de la fenêtre temporelle, il s'agit d'un paramètre fixé durant l'évaluation de l'algorithme.

TODO - suite

**Modèles de Markov Cachés** TODO : (Peeters, 2006)

**Modèles de Markov Cachés de type entrée-sortie (IO-HMM)** TODO : (Bengio and Frasconi, 1996)

## Résultats

Méthode	ACC	REL	PAR	OOF	TOT
COGN	—	—	—	—	—
EAA	—	—	—	—	—
HMM	—	—	—	—	—
IO-HMM	—	—	—	—	—

Table 1: Évaluation des méthodes présentées selon différents indices : ACC (accuracy), REL (relative keys), PAR (parallel keys) et OOF (out-of-a-fifth keys). Le tableau reprend les scores du modèle cognitif (COGN), du modèle d'autocorrélation, des modèles de Markov cachés (HMM), et du modèle de Markov caché d'entrée-sortie (IO-HMM).

## Implémentation en Clojure

TODO : (Kumar, 2015)

## References

- Bengio, Y. and Frasconi, P. (1996). Input-output hmm's for sequence processing.
- Danhauser, A. (1929). *Théorie de la Musique*.
- Kumar, S. (2015). Clojure high performance programming.
- Pauws, S. (2004). Musical key extraction from audio. *Proceedings of the 9th International Conference on Digital Audio Effects, DAFx-06, Montreal, Canada*, pages 96–99.
- Peeters, J. (2006). Musical key estimation of audio signal based on hidden markov modeling of chroma vectors. *Proceedings of the 9th International Conference on Digital Audio Effects, DAFx-06, Montreal, Canada*.
- Sha'ath, I. (2011). Estimation of key in digital music recordings. pages 1–64.
- Takeuchi, A. H. (1994). Maximum key-profile correlation (mkc) as a measure of tonal structure in music. *Perception & Psychophysics*, pages 335–346.