Functional Harmonic Analysis and Computational Musicology in Rameau

Alexandre Tachard Passos[1], Marcos Sampaio[2], Pedro Kröger[2], Givaldo de Cidra[2]

[1]Instituto de Computação (IC)
Universidade Estadual de Campinas (UNICAMP)
Campinas, Brazil

[2]Genos—Computer Music Research Group
School of Music (EMUS)
Federal University of Bahia (UFBA)
Salvador, Brazil

Abstract. *In this paper we present the infrastructure for computational musicology and functional harmonic analysis in Rameau, a framework for experimentation with musicological ideas in software. Rameau supports out of the box chord labeling, key finding, tonal function detection, cadence detection, voice crossing identification, parallel fifths and octaves recognition, seventh note resolution analysis, and can be easily extended to support many other features. It can also generate textual reports, graphical visualization, and typeset scores with the results of these analyses. Rameau is fully open source and implemented in Common Lisp.*

1. Introduction

In the past few years we have been developing Rameau, an open-source system for automatic harmonic analysis and computational musicology [Kröger et al. 2008a]. With Rameau one can process Lilypond files to generate typeset harmonically analysed scores, identify non-chordal sonorities and label chords, perform some basic musicological tasks (such as cadence and voice crossing detection). Rameau is written in Common Lisp and its source code can be found in `http://genos.mus.br/rameau/`, together with the data set of Bach chorales from the Riemenschneider [Riemenschneider 1941] edition we use to benchmark and test the system. In this paper we present Rameau's infrastructure for computational musicology and Rameau's implementation of a functional harmonic analysis framework.

The organization of this paper is as follows; section 2 gives an overview of how Rameau works, section 3 presents the main musicological features implemented in the system. Section 4 describes the functional harmonic analysis structure of Rameau. Section 5 describes superficially each algorithm currently implemented in Rameau. Section 6 contains some concluding remarks and directions for future work.

2. Overview of Rameau

There are two ways of accessing Rameau: a comprehensive command-line interface and a simple web server to perform functional harmonic analysis. When starting the command-line interface the user can choose which Lilypond files are to be processed and what to do with them. The possible operations are called *commands*, and some of the available commands are:

- **help**: provides help and describes the other commands individually;
- **octaves**: identifies parallel octaves in the specified files;
- **crossings**: detects voice crossings in the specified bach chorales;
- **cadences**: detects the cadences in the specified files;
- **analysis**: identifies non-chordal sonorities and name the chords in the specified files, with many different algorithms;
- **functional**: performs functional harmomnic analysis of the given files;
- **document**: generates html documentation for the Rameau source code.

There are altogether 26 different commands, and a new useful command can be implemented in five lines of common lisp code.

Musicological commands can output their data in many formats, such as a typeset score of the interesting section of the song, a list of interesting matches, a histogram of interesting matches and a cloud plot of the extracted data. The harmonic analysis commands (**functional** and **analysis**) can generate tabular results and annotated typeset scores.

3. Computational musicology

The commands **octaves** and **fifths** show how many consecutive octaves and fifths are in a piece and where they are. We found that all consecutive octaves in Bach Chorales are in the form unison–octave or octave–unison, but no consecutive octaves in all Chorales are parallel, although a few fifths are (in chorales 4, 46, 71, and 266). More information can be found in [Kröger et al. 2008b].

The command **chords** lists the frequency of each type of chord in a set of chorales. The command **crossings** finds passages where are voice crossings. We found that in 57% of the Bach chorales there is some kind of voice crossing, although most of the crossings happen in a short period of time (no more than two beats). There are a few interesting cases. For instance, the alto is the lowest voice for a brief period of time in chorale #35 and there is a crossing of the soprano and alto and tenor and bass at the same time in chorale # 290.

There are also commands to find the vocal range used in a composition (**kostka-amb**), to find melodic jumps in a voice (**jumps**), to collect statistics on seventh notes' resolution (**resolve-seventh**), to collect data on how many chord progressions found in the chorales are strong, weak, superstrong and neutral, according to Schoenberg's theory of harmony [Schoenberg 1983] (**schoenberg**), to detect the final cadences in the analyzed pieces (**cadences**), and to count the notes found in major and minor modes (**count-major-notes** and **count-minor-notes**).

4. The functional harmonic analysis structure of Rameau

Roman numeral functional analysis consists, roughly, of two activities: key finding—determining what is the tonal center of the piece and its parts—and roman numeral function detection—determining the tonal function of each segment of the piece. In Rameau we chose to merge these two conceptual concerns into one, and the internal representation chosen reflects that, by stating that the analysis of a song is a list of the analyses of every distinct sonority in the song, and the analysis for each sonority is a local key, mode, and tonal function.

One of the goals of the Rameau project to understand and compare previously published and new techniques for automatic harmonic analysis. To properly compare harmonic analysis algorithms we have a corpus of 371 Bach chorales from the Riemenschneider edition [Riemenschneider 1941], 20 of them annotated by experts with an acceptable harmonic analysis. Rameau includes facilities to train machine learning algorithms on these analyzed chorales (commands named **algorithms** and **funalg** for chord-labeling and functional analysis algorithms, respectively) and a bayesian framework for estimating the correctness of the expert annotations and using this estimate to derive confidence intervals for the accuracies of the algorithms (command named **information-theory**). The analysis results are automatically typeset with the aid of the Lilypond music typesetting program [Nienhuys and Nieuwenhuizen 2008].

5. Functional harmonic analysis algorithms

Using the infrastructure described in section 4 Rameau currently has implementations of four different roman numeral functional analysis; a hidden Markov model, a k-nearest neighbors classifier, a neural network-based harmonic analyzer similar to the one described by Tsui [Tsui 2002], and a trivial extension of Pardo & Birmingham's chord labeling algorithm [Pardo and Birmingham 1999].

5.1. Hidden Markov Model

A hidden Markov model is any probabilistic function of a Markov chain. Using a hidden Markov model to perform roman numeral function analysis consists of modeling the notes in a tonal piece as a probabilistic function of the underlying harmonies, and finding these harmonies, given the notes, using traditional hidden Markov model algorithms. Our approach closely follows that of Raphael and Stoddard [Raphael and Stoddard 2003], and the differences are noted in [Passos 2008].

5.2. K-Nearest Neighbors

A tool used in machine learning for many non-trivial tasks is the k-nearest neighbors classifier [Mitchell 1997]. It works by first representing the instances to be classified in some metric space. Then, to classify an instance x, the knn algorithm chooses, from the training data set, the set s of the k closest examples to x and outputs the most common class in s. The spatial representation we chose for Rameau is a pitch frequency array a, in which, if $f*n$ on the n pitches in a given sonority are encoded as having number p, then $a[p] = f$. When considering surrounding context, we concatenate these arrays and, to avoid adding too much noise in the distance function we weight them down in proportion to the square of the distance between the contextual sonority and the sonority being analyzed. More details can be found in [Passos 2008].

5.3. Pardo & Birmingham's

Pardo & Birmingham [Pardo and Birmingham 1999] describes an algorithm for chord labeling that has some predefined chord templates and chooses among them the one that most closely matches the notes sounding in a given sonority. To extend the original algorithm to perform roman numeral functional analysis we simply created the key for the whole piece using the root and mode of the first chord found, and thus computed the roman numeral function for all other chords as if they were in that key. While simplistic, this approach performs almost competitively with the hidden Markov model.

6. Conclusions and future work

In this paper we presented the current status of Rameau, a framework for automatic harmonic analysis and computational musicology. The framework is mature, and has implementations of many useful musicological functions. The architecture is still too tied to 4-voice part writing, command-line operation, and the Lilypond format, but we are working to correct this in future releases.

While still preliminary, the current implementation of functional harmonic analysis in Rameau is promising, and already produces useful results. Rameau has implementations of a hidden Markov model, a K-nearest neighbors, Pardo & Birmingham's, and neural networks functional harmonic analysis algorithms. Rameau is open source, written in Common Lisp and its source code (together with our data sets and instructions on how to compile, install and run it) is available at `http://genos.mus.br/rameau/`.

References

[Kröger et al. 2008a] Kröger, P., Passos, A., Sampaio, M., and de Cidra, G. (2008a). Rameau: A system for automatic harmonic analysis. In *Proceedings of the 2008 International Computer Music Conference*, pages 273–281.

[Kröger et al. 2008b] Kröger, P., Passos, A., Sampaio, M., de Cidra, G., de Souza Ourives, N., dos Anjos, E. S., and dos Santos, W. S. (2008b). Musicologia computacional aplicada à análise dos corais de bach. In *Anais do XVIII Congresso da ANPPOM*.

[Mitchell 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

[Nienhuys and Nieuwenhuizen 2008] Nienhuys, H.-W. and Nieuwenhuizen, J. (2008). Lilypond. Available at `www.lilypond.org`.

[Pardo and Birmingham 1999] Pardo, B. and Birmingham, W. P. (1999). Automated partitioning of tonal music. Technical report, Electrical Engineering and Computer Science Department, University of Michigan.

[Passos 2008] Passos, A. (2008). Modelos de markov ocultos para análise harmônica funcional de música tonal. Monografia de Graduação defendida na Universidade Federal da Bahia.

[Raphael and Stoddard 2003] Raphael, C. and Stoddard, J. (2003). Harmonic analysis with probabilistic graphical models. In Hoos, H. H. and Bainbridge, D., editors, *Proceedings of ISMIR 2003*, Baltimore, Maryland, USA. The Johns Hopkins University.

[Riemenschneider 1941] Riemenschneider, A., editor (1941). *371 Harmonized Chorales and 69 Chorale Melodies with figured bass*. G. Schirmer, New York.

[Schoenberg 1983] Schoenberg, A. (1983). *Theory of Harmony*. University of California Press.

[Tsui 2002] Tsui, W. S. V. (2002). Harmonic analysis using neural networks. Master's thesis, University of Toronto.