MIREX 2014:
VAMP PLUGINS FROM THE CENTRE FOR DIGITAL MUSIC

Chris Cannam, Emmanouil Benetos, Matthias Mauch, Matthew E. P. Davies, Simon Dixon, Christian Landone, Katy Noland, and Dan Stowell
Queen Mary, University of London
chris.cannam@eecs.qmul.ac.uk

ABSTRACT

In this submission we offer for evaluation several audio feature extraction plugins in Vamp format.

Some of these plugins represent efficient implementations based on modern work, while others are no longer state-of-the-art and were developed a few years ago. The methods implemented in this set of plugins are described in the literature and are referenced throughout this paper. All of the plugins are written in C++ and have been published under open source licences, in most cases the GPL.

A number of these plugins were also submitted to the 2013 edition of MIREX: those are unchanged here and may offer a useful baseline for comparison across years. One plugin, that submitted for the Multiple Fundamental Frequency Estimation and Tracking task, is new this year. Three plugins submitted last year have been omitted from this year's submission.

1. INTRODUCTION

The Vamp plugin format [1] was developed at the Centre for Digital Music (C4DM) at Queen Mary, University of London, during 2005-2006 in response to a desire to publish work in a form that would be immediately useful to people outside this research field. The Vamp plugin format was published with an open source SDK, alongside the Sonic Visualiser [3] audio analysis application which provided a useful host for Vamp plugins.

In subsequent years the Vamp format has become a moderately popular means of distributing methods from the Centre and other research groups. Some dozens of Vamp plugins are now available from groups such as the Music Technology Group at UPF in Barcelona, the Sound and Music Computing group at INESC in Porto, the BBC, and others, as well as from the Centre for Digital Music.

The plugins submitted for this evaluation are provided as a set of dynamic library files. Those with names starting "QM" are all provided in a single library file, the QM Vamp Plugins set, made available in binary form for Windows, OS/X, and Linux from the Centre for Digital Music's download page. [2] These plugins come from a number of authors who are credited in this abstract and in the plugins' accompanying documentation.

In addition to the QM Vamp Plugins set, this submission contains a number of separate plugins: the Chordino plugin from Matthias Mauch; the BeatRoot Vamp Plugin from Simon Dixon; OnsetsDS from Dan Stowell; and the Silvet note transcription plugin from Emmanouil Benetos and Chris Cannam.

The plugins are all provided as 64-bit Linux shared objects depending on GNU libc 2.15 or newer and GNU libstdc++ 3.4.15 or newer. Sonic Annotator v1.0 is also required [3] in order to run the task scripts.

For an overview of this submission across all of the tasks and plugins it covers, please see the relevant repository at the SoundSoftware site. [4]

2. SUBMISSIONS BY MIREX TASK

2.1 Audio Beat Tracking

2.1.1 QM Tempo and Beat Tracker

The QM Tempo and Beat Tracker [4] Vamp plugin analyses a single channel of audio and estimates the positions of metrical beats within the music.

This plugin uses the complex-domain onset detection method from [7] with a hybrid of the two-state beat tracking model proposed in [4] and a dynamic programming method based on [8].

To identify the tempo, the onset detection function is partitioned into 6-second frames with a 1.5-second increment. The autocorrelation function of each 6-second onset detection function is found and this is then passed through a perceptually weighted comb filterbank [4]. The successive comb filterbank output signals are grouped together into a matrix of observations of periodicity through time. The best path of periodicity through these observations is found using the Viterbi algorithm, where the transition matrix is defined as a diagonal Gaussian.

Given the estimates of periodicity, the beat locations are recovered by applying the dynamic programming algo-

[1] http://vamp-plugins.org/

[2] http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html

[3] http://code.soundsoftware.ac.uk/projects/sonic-annotator/

[4] http://code.soundsoftware.ac.uk/projects/mirex2013/

This document is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 License.
http://creativecommons.org/licenses/by-nc-sa/3.0/
© 2010 The Authors.

rithm [8]. This process involves the calculation of a recursive cumulative score function and backtrace signal. The cumulative score indicates the likelihood of a beat existing at each sample of the onset detection function input, and the backtrace gives the location of the best previous beat given this point in time. Once the cumulative score and backtrace have been calculated for the whole input signal, the best path through beat locations is found by recursively sampling the backtrace signal from the end of the input signal back to the beginning.

The QM Tempo and Beat Tracker plugin was written by Matthew Davies and Christian Landone.

2.1.2 BeatRoot

The BeatRoot Vamp plugin [5] is an open source Vamp plugin library that implements the BeatRoot beat-tracking method of Simon Dixon [5]. The BeatRoot algorithm has been submitted to MIREX evaluation in earlier years [6]; this plugin consists of the most recent BeatRoot code release, converted from Java to C++ and modified for plugin format.

The BeatRoot plugin was written by Simon Dixon and Chris Cannam.

2.2 Audio Key Detection

2.2.1 QM Key Detector

The QM Key Detector Vamp plugin continuously estimates the key of the music by comparing the degree to which a block-by-block chromagram correlates to stored key profiles for each major and minor key.

This plugin uses the correlation method described in [10] and [9], but using different tone profiles. The key profiles used in this implementation are drawn from analysis of Book I of the Well Tempered Klavier by J S Bach, recorded at A=440 equal temperament, as described in [13].

The QM Key Detector plugin was written by Katy Noland and Christian Landone.

2.3 Audio Chord Estimation

2.3.1 Chordino

The Chordino plugin [6] was developed following Mauch's 2010 work on chord extraction, submitted to MIREX in that year [12]. While that submission used a C++ chroma implementation with a MATLAB dynamic Bayesian network as a chord extraction front-end [11], Chordino is an entirely C++ implementation that was developed specifically to be made freely available as an open-source plugin for general use.

The method for the Chordino plugin has two parts:

NNLS Chroma — NNLS Chroma analyses a single channel of audio using frame-wise spectral input from the Vamp host. The spectrum is transformed to a log-frequency spectrum (constant-Q) with three bins per semitone. On this representation, two processing steps are performed: tuning, after which each centre bin (i.e. bin 2, 5, 8,) corresponds to a semitone, even if the tuning of the piece deviates from 440 Hz standard pitch; and running standardisation: subtraction of the running mean, division by the running standard deviation. This has a spectral whitening effect.

The processed log-frequency spectrum is then used as an input for NNLS approximate transcription using a dictionary of harmonic notes with geometrically decaying harmonics magnitudes. The output of the NNLS approximate transcription is semitone-spaced. To get the chroma, this semitone spectrum is multiplied (element-wise) with the desired profile (chroma or bass chroma) and then mapped to 12 bins.

Chord transcription — A fixed dictionary of chord profiles is used to calculate frame-wise chord similarities. A standard HMM/Viterbi approach is used to smooth these to provide a chord transcription.

Chordino was written by Matthias Mauch.

2.4 Audio Onset Detection

2.4.1 QM Note Onset Detector

The QM Note Onset Detector Vamp plugin estimates the onset times of notes within the music. It calculates an onset likelihood function for each spectral frame, and picks peaks in a smoothed version of this function.

Several onset detection functions are available in this plugin; this submission uses the complex-domain method described in [7].

The QM Note Onset Detector plugin was written by Chris Duxbury, Juan Pablo Bello and Christian Landone.

2.4.2 OnsetsDS

OnsetsDS [7] is an onset detector plugin wrapping Dan Stowell's OnsetsDS library [8], described in [14].

OnsetsDS was designed to provide an FFT-based onset detection that works very efficiently in real-time, with a fast reaction time. It is not tailored for non-real-time use or for any particular type of signal.

The OnsetsDS plugin was written by Dan Stowell and Chris Cannam.

2.5 Multiple Fundamental Frequency Estimation and Tracking

2.5.1 Silvet

Silvet (for Shift-Invariant Latent Variable Transcription) [9] is a Vamp plugin for automatic music transcription, using a method based on that of [2]. It produces a note transcription as output, and so is here submitted only for the note tracking evaluation in this task, and not for framewise evaluation.

Silvet uses a probablistic latent-variable estimation method to decompose a Constant-Q time-frequency matrix into note activations using a set of spectral templates learned from

[5] http://code.soundsoftware.ac.uk/projects/beatroot-vamp/
[6] http://isophonics.net/nnls-chroma
[7] http://code.soundsoftware.ac.uk/projects/vamp-onsetsds-plugin/
[8] http://onsetsds.sourceforge.net/
[9] http://code.soundsoftware.ac.uk/projects/silvet/

recordings of solo instruments. The method is thought to perform quite well for clear recordings that contain only instruments with a good correspondence to the known templates. Silvet does not contain any vocal templates, or templates for typical rock or electronic instruments.

The method implemented in Silvet is very similar to that submitted to MIREX in 2012 as the BD1, BD2 and BD3 submissions in the Multiple F0 Tracking task of that year [1]. In common with that submission, and unlike the paper cited at [2], Silvet uses a simple thresholding method instead of an HMM for note identification. However, Silvet follows [2] rather than [1] in including a 5-bin-per-semitone pitch shifting parameter.

The Silvet plugin was written by Chris Cannam and Emmanouil Benetos.

3. REFERENCES

[1] Emmanouil Benetos and Simon Dixon. Multiple-F0 estimation and note tracking for MIREX 2012 using a shift-invariant latent variable model. In *Music Information Retrieval Evaluation Exchange (MIREX)*, 2012.

[2] Emmanouil Benetos and Simon Dixon. A shift-invariant latent variable model for automatic music transcription. *Computer Music Journal*, 36(4):81–94, 2012.

[3] Chris Cannam, Christian Landone, and Mark Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the ACM Multimedia 2010 International Conference*, 2010.

[4] Matthew E. P. Davies and Mark D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009–1020, 2007.

[5] Simon Dixon. An interactive beat tracking and visualisation system. In *Proceedings of the 2001 International Computer Music Conference (ICMC'2001)*, 2001.

[6] Simon Dixon. MIREX 2006 audio beat tracking evaluation: Beatroot. In *Music Information Retrieval Evaluation Exchange (MIREX)*, 2006.

[7] Chris Duxbury, Juan Pablo Bello, Mike Davies, and Mark Sandler. Complex domain onset detection for musical signals. In *Proceedings of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*, 2003.

[8] D. P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 37(1):51–60, 2007.

[9] Emilia Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing, Special Cluster on Computation in Music*, 18, 2006.

[10] C. L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, 1990.

[11] Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[12] Matthias Mauch and Simon Dixon. MIREX 2010: Chord detection using a dynamic bayesian network. In *Music Information Retrieval Evaluation Exchange (MIREX)*. 2010.

[13] Katy Noland and Mark Sandler. Signal processing parameters for tonality estimation. In *Audio Engineering Society Convention 122*, May 2007.

[14] Dan Stowell and Mark D. Plumbley. Adaptive whitening for improved real-time audio onset detection. In *Proceedings of the International Computer Music Conference (ICMC'07)*, 2007.