Audio Engineering Society

Convention Paper

Presented at the 134th Convention
2013 May 4–7 Rome, Italy

Evaluation of acoustic features for music emotion recognition

Chris Baume[1]

[1] *BBC Research and Development, London, UK*

Correspondence should be addressed to Chris Baume (chris.baume@bbc.co.uk)

ABSTRACT

Classification of music by mood is a growing area of research with interesting applications, including navigation of large music collections. Mood classifiers are usually based on acoustic features extracted from the music, but often they are used without knowing which ones are most effective. This paper describes how 63 acoustic features were evaluated using 2389 music tracks to determine their individual usefulness in mood classification, before using feature selection algorithms to find the optimum combination.

1. INTRODUCTION

Music has the uncanny ability to affect and reflect human emotion, often in a powerful way. Even the simplest of tunes can evoke feelings of happiness, sadness, anger, fear, or anything in between, but the mechanics behind this phenomenon are still not fully understood.

The digital music revolution has seen the size of music collections grow rapidly, and libraries containing millions of tracks are not uncommon. With such vast amounts of music available, there is more and more interest in developing new and innovative ways of searching and navigating music libraries. In broadcasting, music is often used as a tool to portray a certain mood, and so it would be useful to be able to search for music and refine searches in this way.

'Making Musical Mood Metadata' (M4) is a collaborative project between BBC R&D, the Queen Mary University of London (QMUL) Centre for Digital Music and I Like Music (ILM). ILM is a company that provides the BBC with an online music library service called 'Desktop Jukebox', which contains over a million tracks.

The project's goals are to improve navigation in large music libraries by using automated audio analysis to add features to the Desktop Jukebox. This includes finding similar-sounding tracks, generating visual thumbnails of what a track sounds like, and

navigating or filtering tracks by mood.

Searching for music by mood requires the music to be labelled with mood information. Doing this by hand is impractical on large collections, so the process must be automated. Such a system can be viewed as a black box with music going in one end and mood information coming out the other. This divides the problem into three parts – what to use as the input, how to represent the output and how to make the connection between the two.

This paper focusses on the first part by evaluating which inputs perform best when classifying mood.

2. RELATED WORK

Music mood classification, also known as music emotion recognition (MER), is a growing area of research with numerous papers at the ISMIR and CMMR conferences, and a book on the subject [1]. MIREX – the annual evaluation of music information retrieval algorithms – has included mood classifiers since 2007 [2].

Mood classifiers always use machine learning techniques [3, 4] to make connections between the music and the mood. There are many different methods that have been used to numerically represent mood. The most popular method is to categorise into a fixed number of basic emotions [5, 6, 7]. Although these models are easy to test and analyse, they cannot represent all possible emotions or combinations of emotions, so multi-dimensional regression is becoming more popular [8, 9, 10].

As classifiers cannot handle time-varying data, the music must first be processed to extract key bits of information known as 'features'. These can include statistics-based algorithms like FFT bins, zero-crossing rate and variance, but also musical features like tempo, mode and key. Finding a combination of features that can fully represent the mood of music is key to a successful classifier.

Large datasets of pre-extracted features exist, notably the Million Song Dataset [11] and Magnatagatune [12]. The former contains artist/title metadata for a million songs and the latter contains 25,863 low-quality audio clips with descriptive metadata. Both include features extracted using proprietary algorithms from The Echo Nest [13].

There are many open source software packages available which implement feature extraction algorithms. MIR Toolbox [14], MA Toolbox [15] and PsySound3 [16] are MATLAB-based, and Marsyas [17], CLAM [18], LibXtract [19], Aubio [20] and YAAFE [21] can be used as libraries with C++ or Python.

Researchers often use one or more of these packages to extract a range of features to use in their classifier. This can often be inefficient as some features represent mood better than others. To find and remove the poor performing ones, feature selection algorithms are sometimes used to select the most important features [22, 23, 24, 7].

In a recently-published paper, Song [6] used a 4-mood classifier to analyse the usefulness of different combinations of features from MIR Toolbox. This paper goes one step further to analyse the usefulness of 63 individual features to find which are most important in classifying mood.

3. METHODOLOGY

To analyse the usefulness of the features, each one was extracted from a set of music files and used to train a mood classifier. The classifier was then tested to produce a result for that feature. The process is described in detail below.

3.1. Dataset

As well as containing over a million commercial music tracks, the Desktop Jukebox houses over 300,000 tracks of production music. 128,024 of these were obtained for use in the M4 project. They are stored as 16-bit linear PCM files and take up 2.3TB of disk space.

Production music is so named because it is often used in television, radio and film productions for background and mood music. It differs from commercial music in that all of the rights are owned by one organisation who are paid each time the music is played. This business model means that they are keen to make their music easy to find and search, so every track is hand-labelled with lots of useful information.

The richness of the production music metadata and the breadth of musical styles it covers make it ripe for use as training data in a machine learning environment, provided that proper filtering is applied to avoid using ambiguous or incorrect metadata.

An analysis of the production music showed that the average track contains 40 keywords, of which 16 describe the genre, 12 describe the mood and 5 describe the instrumentation. By mapping the top 75 mood keywords and spacing each pair relative to their frequency of co-occurrence, the keywords formed a logical pattern with opposing moods appearing on different sides of the graph. The graph can be seen in a blog post at `http://bbc.in/XKeCEn`.

The four widest-spaced mood keywords were *terror, peace, joy* and *excitement*. These were chosen as the four mood categories used for classification. Tracks which contain any of these four words were set aside for the training/test dataset. Any tracks containing more than one of the keywords were removed. This resulted in a set of 2389 tracks, divided as follows. The baseline for results is $698/2389 = 29.22\%$.

terror	546	joy	698
peace	592	excitement	553

3.2. Processing chain

In addition to the feature extraction implementations listed in Section 2, QMUL developed the Vamp plugin architecture [25]. This allows audio feature extraction algorithms to be written as – or wrapped in – a standardised C++ plugin, much like a VST or LADSPA plugin. Using a standard open-source plugin architecture means that audio files can be processed without having to write programs to call various feature extraction libraries or having to pay for expensive software licences.

The functions of reading audio data, windowing the data, performing an FFT, running the plugin's algorithm and writing the output vector to a file are carried out by a host program. There are two reference host programs provided by QMUL - sonic annotator and sonic visualiser. The former is used on the command line to batch process files and save the results, while the latter is GUI-based and is used to visualise the resulting audio features.

A wide range of source code and binaries for Vamp plugins are publicly available, including wrappers for many parts of LibXtract and Marsyas. They can be downloaded from `www.vamp-plugins.org`.

Sonic annotator was used with custom Bash and Python scripts to process the dataset described in Section 3.1. The Vamp plugin and its parameters are specified by passing a 'transform file' to sonic annotator. The custom scripts are designed to, for a given transform, extract features from the dataset, find the mean and standard deviation of the output vectors, and write the results to a text file.

The Vamp plugins used in the implementation come from six packages, which together form a collection of 63 different features. Five of the plugin packages are from external sources, namely QMUL, LibXtract, Mazurka, NNLS/Chroma and Marsyas. All of them can be downloaded from the Vamp plugin website.

The final plugin package was developed by BBC R&D and is open source [26]. It includes implementations for intensity and intensity ratio [27], rhythmic features which take inspiration from [27] and [28], and spectral contrast [29]. It was found that for spectral contrast, the dimensionality reduction used in the original paper [29] and the contrast equations used in [30] were ineffective, and only the raw peak and valley features were necessary.

3.3. Machine learning

This section describes how classifiers were trained to predict mood based on individual features and tested to evaluate their performance. A separate classifier was trained for each feature in order to determine how good that feature is at classifying mood. The process for training/testing is described below and results are listed in Section 4.

The algorithm chosen for the machine learning process was the support vector machine (SVM), which has been shown to work well for mood and genre classification [2]. LibSVM [31] was chosen as its implementation is popular and efficient. It was set up to use the RBF kernel as this usually has the highest accuracy, however Song has since demonstrated [6] that a polynomial kernel can be faster and more accurate for mood classification.

Firstly, the music tracks were split randomly into training and test datasets. 20% – or 469 tracks – were set aside for testing purposes.

With SVMs, it is necessary to optimise the two parameters of the classifier – Gamma (γ), which controls the width of the gaussian function used and C

which is the cost value. The optimisation testing is done using 'cross-validation' on the training data, by splitting the data into k parts (known as k-fold). An SVM classifier is trained on all but one of the parts, and tested on the remaining one. This part is then swapped and the process is repeated until there have been k iterations. The mean accuracy is used as the final result. After many combinations of γ and C have been tested, the combination with the highest accuracy is chosen as the optimum.

The overall accuracy of the classifier is calculated by training an SVM classifier on all of the training data using the optimised γ and C values. The classifier is then tested against the test dataset that was put aside at the beginning.

4. RESULTS

The results were obtained by taking the 2389 audio files described in Section 3.1 and, for each feature, processing the music using the feature's algorithm, finding the mean and standard deviation of the output (if the output is a vector), and training/testing an SVM classifier as described in Section 3.3.

The results for each feature are listed below. 'Len' represents the length of the feature vector and 'Result' is the accuracy of the optimised classifier. Each feature has a superscript label referring to the Vamp plugin used to process the music – LibXtract (1), QMUL (2), BBC (3), Mazurka (4) and NNLS (5).

Feature	**Len**	**Result**
Bark coefficients[1]	52	71.22%
NNLS chromagram[5][32]	24	70.15%
MFCC[2](20 coefficients) [33]	40	70.15%
Intensity ratio[3](9 bands) [27]	18	69.72%
Spectral contrast[3][29]	28	69.66%
NNLS bass chromagram[5][32]	24	68.02%
MFCC[2](12 coefficients) [33]	24	66.31%
Intensity ratio[3](7 bands) [27]	14	65.67%
MFCC[2](7 coefficients) [33]	14	63.33%
Key strength[2][34]	50	62.05%
Chromagram[2]	24	60.77%
6D Tonal Content Space[2][35]	12	60.34%
Intensity ratio[3](5 bands) [27]	10	58.21%
Spectral centroid[2](log) [36]	2	50.32%
Average deviation[1][36]	2	48.61%
Spectral flatness[4][36]	2	48.40%
Spectral centroid[2](linear) [36]	2	48.19%
Variance[1][36]	2	47.55%
Spectral inharmonicity[1][36]	2	47.12%
Total loudness[1][36]	2	46.91%
NNLS harmonic change[5][37]	2	46.91%
Spectral standard deviation[1][36]	2	45.84%
Spectral rolloff[1][36]	2	45.42%
Intensity[3][27]	2	45.20%
Spectral slope[1][36]	2	45.20%
Spectral flux[3](L1 norm) [28]	2	44.99%
Non-zero count[1]	2	44.78%
Spectral irregularity[1](Krimphoff [38])	2	44.56%
Smooth power slope[4][39]	2	43.71%
Rhythm strength[3][27]	1	43.50%
Average onset frequency[3][27]	1	43.28%
Spectral smoothness[1][40]	2	43.28%
Zero-crossing rate[1][36]	2	42.64%
Consonance[5][32]	2	42.43%
Spectral sharpness[1][36]	2	42.22%
Spectral spread[1][36]	2	41.15%
Highest value[1]	2	40.72%
Spectral irregularity[1](Jensen [41])	2	40.72%
Third tristimulus[1][36]	2	40.09%
Spectral flux[3](L2 norm) [28]	2	39.87%
RMS energy[3](20ms window)	2	38.81%
Lowest value[1]	2	38.59%
Second tristimulus[1][36]	2	38.59%
Temporal centroid[3][36]	1	38.59%
RMS energy[3](10ms window)	2	36.67%
First tristimulus[1][36]	2	36.03%
Scaled smooth power slope[4][39]	2	35.82%
Peak-valley ratio[3][27]	1	34.54%
Log attack time[3][36]	1	34.54%
Mean correlation peak[3][27]	1	34.12%
Tonal content function[2][35]	2	33.05%
Low energy ratio[3][42] (< 0.2mean)	1	31.77%
Tempo[2](QMUL method) [43]	1	29.91%
Tempo[3](BBC method) [26]	1	29.85%
Odd/even ratio[1][36]	2	29.64%
Skewness[1][36]	2	29.42%
Spectral average deviation[1][36]	2	29.42%
Spectral flatness[1][36]	2	29.21%
Kurtosis[1]	1	29.21%
Spectral kurtosis[1][36]	1	29.21%
Spectral skewness[1][36]	1	29.21%
Low energy ratio[3][42] ($<$ mean)	1	29.00%
Spectral crest[1][36]	2	29.00%

Feature	Len	$\frac{(Result-Baseline)}{Length}$
Rhythm strength	1	14.29%
Average onset frequency	1	14.07%
Spectral centroid (log)	2	10.55%
Average deviation	2	9.70%
Spectral flatness	2	9.59%

Table 1: Top performing features when considering vector length.

4.1. Discussion

The peak result of 71% is unusually high for music emotion recognition. A very similar study [6] which classified 2904 tracks into four mood categories only achieved a maximum 54% accuracy with a vector length of 47. The high result is probably because the mood categories were chosen with prior knowledge of the dataset. It is also possible that the songs with those mood labels are similar in style/genre, or are from the same album, which would narrow the acoustic range and make the classification easier.

The intensity ratio and spectral contrast features performed much better than expected. Despite having shorter vector lengths and much simpler algorithms, they produced similar results to the chromagram and MFCCs.

Approximately a dozen features performed close to the baseline of 29.2%. With these, either there is no correlation between output and the mood, or there is a problem with the implementation. The low energy ratio and spectral crest features were both found to have some errors (`NaN` values) which would explain the result falling just below baseline.

Longer vectors performed, in general, better than shorter ones. This is predictable as they contain more information. However, the shorter vectors performed better for their size, as can be seen in Table 1.

Spectral and harmonic features performed better than temporal, energy and rhythmic features, indicating that timbre plays the most important role in determining the mood of music.

Despite this, Table 1 shows that rhythm and temporal features have better performance by vector length. Although they cannot represent mood well on their own, the information they represent will not overlap with the spectral or harmonic features. This means that they can be combined to boost the overall performance.

ReliefF (73.50%)	**Gain ratio (79.7%)**
Second tristimulus	Key strength
Consonance	Bark coefficients
NNLS chroma	MFCC (12 coeffs)
Key strength	MFCC (20 coeffs)
First tristimulus	Chromagram
NNLS harmonic change	Third tristimulus
Spectral flatness	Mean correlation peak
Spectral inharmonicity	NNLS chroma
Average onset frequency	Intensity ratio (9)
Intensity ratio (9)	Intensity ratio (7)
Chi squared (79.05%)	**SVM attribute (75%)**
Smooth power slope	Non-zero count
Spectral flux (L2 norm)	Smooth power slope
Bark coefficients	Bark coefficients
Spectral contrast	Spectral flatness
NNLS bass chroma	MFCC (20 coeffs)
Intensity ratio (9)	Spectral inharmonicity
NNLS chroma	Spectral smoothness
Chromagram	Intensity ratio (5)
MFCC (20 coeffs)	NNLS harmonic change
Key strength	Consonance

Table 2: Top-performing combinations of ten features, as chosen by feature selection algorithms. The results of an SVM classifier trained using those features is also displayed.

5. FEATURE SELECTION

A mood classification system would ideally use more than one set of features to achieve the highest accuracy, whilst maintaining a balance with the time it takes to calculate those features and the space it takes to store them. Evaluating all of the different combinations of features would require a large amount of processing time, but this can be significantly reduced by employing feature selection algorithms.

Weka [44] is a machine learning toolkit that incorporates a number of these algorithms. Each of the algorithms returns either an unranked subset of features, or places all the features into a single ranked order. The subsets that are returned generally contain at least one part of every feature vector, which

makes it impossible to compare features. However, by scoring each part of the feature vectors using a ranking-based algorithm and taking the maximum score for each feature, we can select the optimum combination.

The 1920 training items used in Section 3 were processed using the ReliefF, gain ratio, chi squared and SVM attribute evaluation algorithms to try and find a combination of features that produces the highest accuracy. The full results are published online [45], but the top combinations of ten features chosen by each algorithm are shown in Table 2. The results of training and testing an SVM classifier using those features are also shown.

Each of the selections contains a mixture of spectral (e.g. intensity ratio) and harmonic (e.g. chroma) features. The only feature to appear in all of the selections is intensity ratio, although Bark coefficients, NNLS chroma, key strength and MFCCs each appear in three out of four.

6. CONCLUSIONS

This paper has described how 63 audio descriptors and musical features were extracted from a set of 2389 production music tracks and evaluated to find out how well each of them performed as part of an SVM in classifying the music into four mood categories. The results found that:

- Overall, spectral and harmonic features perform better than rhythm, temporal and energy features (see Section 4).
- Rhythm and temporal features perform well considering their size (see Table 1), and as they don't overlap with spectral or harmonic features, they can be used together to boost the overall performance.
- The intensity ratio and spectral contrast features performed much better than expected for such simple algorithms, and with shorter vector lengths than similar spectral features.
- The highest result of 71% is unusually high and is most likely a result of optimising the mood model for the dataset. Real-world results will probably be lower, in the 55-65% region [24, 6, 46].
- A number of feature selection algorithms were tested (see Table 2), each resulting in a combination of spectral and harmonic features. By using ten features together, the overall accuracy increased to almost 80%.

The M4 project is continuing by upgrading the four-mood classifier used in this paper to a multi-dimensional regressor. It is hoped that this will allow a greater range and complexity of emotions to be represented. The final mood prediction system will be user tested and integrated into the Desktop Jukebox service. The results of this work will be published in a later paper.

7. ACKNOWLEDGEMENTS

This research is part-funded by the Technology Strategy Board, project reference #101040. Thanks go to Andy Hill from I Like Music and the music publishers for providing the production music, and to Erik Schmidt from Drexel and Alba Rosado from Pompeu Fabra for providing reference implementations.

8. REFERENCES

[1] Yi-Hsuan Yang and Homer H. Chen. *Music Emotion Recognition*. CRC Press, 2011.

[2] Xiao Hu, JS Downie, Cyril Laurier, and Mert Bay. The 2007 MIREX audio mood classification task: Lessons learned. In *International Society for Music Information Retrieval Conference*, pages 462–467, 2008.

[3] Mathieu Barthet and George Fazekas. Multidisciplinary Perspectives on Music Emotion Recognition: Implications for Content and Context-Based Models. In *International Symposium on Computer Music Modeling and Retrieval (CMMR)*, pages 19–22, 2012.

[4] Youngmoo E. Kim, Erik M. Schmidt, Raymond Migneco, Brandon G. Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A. Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *International Society for Music Information Retrieval Conference*, pages 255–266, 2010.

[5] Zhijun Zhao, Lingyun Xie, Jing Liu, and Wen Wu. The analysis of mood taxonomy comparision between chinese and western music. *International Conference on Signal Processing Systems*, pages V1–606–V1–610, July 2010.

[6] Yading Song, Simon Dixon, and Marcus Pearce. Evaluation of Musical Features for Emotion Classification. In *International Society for Music Information Retrieval Conference*, pages 523–528, 2012.

[7] Cyril Laurier, Mohamed Sordo, and Joan Serrà. Music mood representations from social tags. In *International Society for Music Information Retrieval Conference*, pages 381–386, 2009.

[8] Ei Ei Pe Myint and Moe Pwint. An Approach for Multi-Label Music Mood Classification. In *IEEE International Conference on Signal Processing Systems*, pages V1–290–V1–294. Ieee, July 2010.

[9] Yi-Hsuan Yang, Yu-Ching Lin, Ya-Fan Su, and Homer H. Chen. A Regression Approach to Music Emotion Recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2), 2008.

[10] Mark Mann and Trevor Cox. Music Mood Classification of Television Theme Tunes. In *International Society for Music Information Retrieval Conference*, pages 735–740, 2011.

[11] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *International Society for Music Information Retrieval Conference*, pages 591–596, 2011.

[12] Edith Law and Luis Von Ahn. Input-Agreement : A New Mechanism for Collecting Data Using Human Computation Games. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10, 2009.

[13] The Echo Nest. Developer API http://developer.echonest.com.

[14] Olivier Lartillot, Petri Toiviainen, and Tuomas Eerola. A Matlab Toolbox for Music Information Retrieval. In *Conference of the Gesellschaft für Klassifikation*, 2008.

[15] Elias Pampalk. A MATLAB Toolbox to Compute Music Similarity from Audio. In *International Society for Music Information Retrieval Conference*, 2004.

[16] Densil Cabrera, Sam Ferguson, and Emery Schubert. PsySound3: Software for Acoustical and Psychoacoustical Analysis of Sound Recordings. In *International Conference on Auditory Display*, pages 356–363, 2007.

[17] George Tzanetakis and Perry Cook. MARSYAS : A framework for audio analysis. *Organised Sound*, 4(3), 2000.

[18] Xavier Amatriain, Jordi Massaguer, David Garcia, and Ismael Mosquera. The CLAM Annotator : A Cross-platform Audio Descriptors Editing Tool. In *International Society for Music Information Retrieval Conference*, 2005.

[19] Jamie Bullock. LibXtract: A Lightweight Library for Audio Feature Extraction. In *International Computer Music Conference*, pages 3–6, 2007.

[20] Paul M Brossier. The Aubio Library at MIREX 2006. In *International Society for Music Information Retrieval Conference*, 2006.

[21] Benoit Mathieu, Slim Essid, Thomas Fillon, Jacques Prado, and Gael Richard. YAAFE: An Easy to Use and Efficient Audio Feature Extraction Software. In *International Society for Music Information Retrieval Conference*, 2010.

[22] Shyamala Doraisamy, Shahram Golzari, Noris Mohd. Norowi, Md. Nasir B. Sulaiman, and Nur Izura Udzir. A Study on Feature Selection and Classification Techniques for Automatic Genre Classification of Traditional Malay Music. In *International Society for Music Information Retrieval Conference*, pages 331–336, 2008.

[23] Tuomas Eerola and Olivier Lartillot. Prediction of multidimensional emotional ratings in music from audio using multivariate regression models. In *International Society for Music Information Retrieval Conference*, pages 621–626, 2009.

[24] Renato Panda and Rui Pedro Paiva. Music Emotion Classification: Dataset Acquisition and Comparative Analysis. In *International Conference on Digital Audio Effects (DAFx)*, pages 1–7, 2012.

[25] Chris Cannam. The Vamp audio analysis plugin system. *http://vamp-plugins.org*.

[26] Chris Baume and Yves Raimond. BBC Vamp plugin collection. *https://github.com/bbcrd/bbc-vamp-plugins*, 2013.

[27] Lie Lu, Dan Liu, and Hong-Jiang Zhang. Automatic Mood Detection and Tracking of Music Audio Signals. In *IEEE Transactions on Audio, Speech and Language Processing*, volume 14, pages 5–18, 2006.

[28] Simon Dixon. Onset Detection Revisited. In *International Conference on Digital Audio Effects (DAFx)*, pages 133–137, 2006.

[29] Dan-Ning Jiang, Lie Lu, and Hong-Jiang Zhang. Music type classification by spectral contrast feature. In *IEEE International Conference on Multimedia and Expo*, pages 113–116, 2002.

[30] Vincent Akkermans, Joan Serrà, and Perfecto Herrera. Shape-based spectral contrast descriptor. In *Sound and Music Computing Conference*, number July, pages 23–25, 2009.

[31] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1—-27:27, 2011.

[32] Matthias Mauch. *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*. PhD thesis, 2010.

[33] S Davis and P Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics Speech and Signal Processing*, 28(4):357–366, 1980.

[34] Katy C Noland. *Computational Tonality Estimation: Signal Processing and Hidden Markov Models*. PhD thesis, 2009.

[35] Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *ACM workshop on Audio and music computing multimedia (AMCMM)*, page 21, New York, New York, USA, 2006. ACM Press.

[36] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, 2004.

[37] Matthias Mauch and Simon Dixon. Approximate Note Transcription for the Improved Identification of Difficult Chords. In *International Society for Music Information Retrieval Conference*, number 1, 2010.

[38] J. Krimphoff, Stephen McAdams, and S. Winsberg. Caractérisation du timbre des sons complexes. *Journal de Physique IV*, 4:2–5, 1994.

[39] Craig Stuart Sapp. Mazurka project plugins (mazurka.org.uk/software/sv/plugin).

[40] Stephen McAdams. Perspectives on the Contribution of Timbre to Musical Structure. *Computer Music Journal*, 23(3):85–102, 1999.

[41] Kristoffer Jensen. *Timbre Models of Musical Sounds*. PhD thesis, 1999.

[42] George Tzanetakis and Perry Cook. Musical Genre Classification of Audio Signals. *IEEE Transactions on Audio, Speech and Language Processing*, 10(5):293–302, 2002.

[43] Mike Davies. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009–1020, 2007.

[44] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[45] Chris Baume. AES134 full results. *https://bbcarp.org.uk/m4/aes134*, 2013.

[46] JS Downie. MIREX 2012 Evaluation Results. In *International Society for Music Information Retrieval Conference*, 2012.