ACOUSTICBRAINZ: A COMMUNITY PLATFORM FOR GATHERING MUSIC INFORMATION OBTAINED FROM AUDIO

Alastair Porter[†‡], Dmitry Bogdanov[†], Robert Kaye[‡], Roman Tsukanov[‡], Xavier Serra[†]
[†]Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[‡]MetaBrainz Foundation
alastair.porter,dmitry.bogdanov,xavier.serra@upf.edu
rob,roman@metabrainz.org

ABSTRACT

We introduce the AcousticBrainz project, an open platform for gathering music information. At its core, AcousticBrainz is a database of music descriptors computed from audio recordings using a number of state-of-the-art Music Information Retrieval algorithms. Users run a supplied feature extractor on audio files and upload the analysis results to the AcousticBrainz server. All submissions include a MusicBrainz identifier allowing them to be linked to various sources of editorial information. The feature extractor is based on the open source Essentia audio analysis library. From the data submitted by the community, we run classifiers aimed at adding musically relevant semantic information. These classifiers can be developed by the community using tools available on the AcousticBrainz website. All data in AcousticBrainz is freely available and can be accessed through the website or API. For AcousticBrainz to be successful we need to have an active community that contributes to and uses this platform, and it is this community that will define the actual uses and applications of its data.

1. INTRODUCTION

One of the biggest bottlenecks in many Music Information Retrieval (MIR) tasks is the access to large amounts of music data, in particular to audio features extracted from commercial music recordings. Most approaches to tasks such as music classification, auto-tagging, music similarity and music recommendation, are based on using audio features obtained from well-established audio signal processing algorithms. This is a time consuming process that is beyond the possibilities of any individual researcher. It may not be possible for researchers to gather this much information, annotate it according to their needs, or compute the required features at the scale required for the task. For example, existing datasets for genre classification are of insufficient size with respect to both the number of instances per class and the ability of these instances to accurately represent the entire musical genre space [4]. A list of datasets commonly used in MIR is provided in [1]. Half of them have fewer than 10,000 instances, although in recent years there have been attempts to create larger datasets. Building such datasets would allow research at the scale of the requirements of commercial applications.

In general however, the creation of datasets may be difficult for researchers due to a number of reasons:

- Gathering and sharing datasets require legal considerations with regard to the distribution of copyrighted material [7].
- Collections which are hand-created may be biased in their contents and annotations, especially if they are created by only one person, or if they are created for the evaluation of a specific task or algorithm (such as the GZTAN dataset, commonly used to evaluate audio feature-based genre classification algorithms [10]).

One project in recent years to address some of these issues is the Million Song Dataset (MSD) [1]. At the time of its release, this was the largest dataset of music descriptors in the MIR community and has gained a lot of attention for its size and breadth of music content, as well as the simplicity of accessing its data. The MSD relies on the EchoNest API [1] to compute its descriptors, a commercial product which is closed to academic inspection. Some downsides of this approach include:

- Implementation details of the algorithms used to compute the descriptors are unknown and it is impossible to review the quality of their implementation.
- The dataset is fixed in time, and does not appear to have been updated with new features, or music released since it was created.

The MSD has been further expanded with features computed using open source algorithms, on audio samples from 7digital.com [9]. As this dataset reflects the MSD, it is also fixed in time, and features do not represent the whole recording, but only the sample.

[1] http://developer.echonest.com

Based on these considerations, we believe that there is still space for a large dynamic dataset consisting of music features calculated with open algorithms.

2. ACOUSTICBRAINZ

We are introducing a new platform, AcousticBrainz, [2] to assist with the gathering of musical data from the music enthusiast and research community, and to provide researchers with large datasets of recordings to work with. All of the source code in AcousticBrainz is open, [3] encouraging sharing of algorithms between contributors and providing the ability for people to improve on the work of others. All submitted and generated data is freely available under a Creative Commons CC-0 license (public domain).

The platform is split into three categories: feature extraction, data storage, and the creation of musical semantic information. A feature extractor, based on algorithms in the Essentia audio analysis library [2], can be downloaded by anyone who wishes to contribute data to the project. They run this extractor on their personal computer, giving audio files as input. The output of this extractor is a JSON file for each audio track containing descriptors (see Section 2.3.3). A submission tool provided with the extractor automatically uploads the JSON files to the AcousticBrainz server.

A database stores submissions and makes the data available via an API. AcousticBrainz only stores descriptors of audio, and never the actual audio itself. Submissions are identified by the MusicBrainz identifier (MBID) of the input audio file. These stable identifiers let us uniquely and unambiguously refer to a music recording, and can also let us obtain additional editorial information from MusicBrainz and from other services that also understand MBIDs.

To encourage experimentation with the data, the AcousticBrainz website lets anybody create, annotate, and share their own datasets consisting of recordings present in the database. A search interface lets users query for recordings based on editorial data from MusicBrainz or extracted features and add the results to the dataset. From these datasets users can build classifier models which can be used to estimate characteristics of any recording present in AcousticBrainz.

2.1 MusicBrainz

MusicBrainz [4] is a community-maintained open encyclopedia of music information. It contains editorial metadata for many musical concepts, including Artists (individuals, groups, and other people associated with musical events), Releases, Recordings, and Works. It also contains relationships between items, and to other external databases. Data is entered manually by a large community of volunteers (editors), who also vote on changes made by other editors to ensure its quality. It is used by a number of commercial companies. [5] Every item in the database is uniquely identified by an MBID and many companies and organizations rely on these IDs as identifiers for music-related concepts. MBIDs can be used to retrieve data from external services which understand them (e.g., Last.fm, WikiData), and are also a part of the Music Ontology.

2.2 Current submission statistics

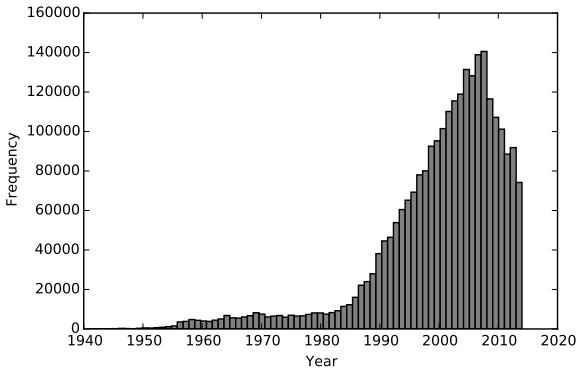


Figure 1: Release years of submissions from file metadata.

Format	Count
mp3	1,784,778
flac	777,826
vorbis	83,867
aac	64,733
alac	29,481
wmav2	4,019
other	1,320

Table 1: Number of submissions per audio codec.

At this time, [6] the AcousticBrainz database has audio features submitted for 1,671,701 unique recording MBIDs. We keep duplicate submissions from different sources, resulting in a total of 2,747,094 submissions. For these submissions we also have metadata information available from MusicBrainz, including 99,159 artists and 165,394 releases. The duplicates consist of analyzed features of various source audio files, with differing codecs, encoders, and bit rates. These duplicates let us see real-world examples of the effect of different codecs and encoding parameters on our descriptors. We have collected submission for 538,614 unique MBIDs (807,307 including duplicates) for audio files encoded using a lossless codec (FLAC and ALAC), which is in itself is a large database. The most common audio format for submitted files is MP3, with more submissions than for all other formats combined (Table 1). 94% of submitted files contain year metadata. We show a histogram of the year that submissions were released in Figure 1. The majority of tracks are from the

[2] http://acousticbrainz.org
[3] https://github.com/metabrainz/acousticbrainz-server
[4] https://musicbrainz.org
[5] https://metabrainz.org/customers
[6] July 7 2015

1990s and first decade of the 2000s. As tracks submitted to AcousticBrainz require a MBID this distribution may also be reflective of the content in the MusicBrainz database. The current size of the database (containing all JSON file submissions) is approximately 118GB, split between 102GB of low-level data (average file size 40kB), and 12GB of high-level (file size 4kB).

Tag	Count
Rock	195,837
Pop	103,486
Classical	90,231
Jazz	88,702
Soundtrack	79,056
Electronic	71,758
Metal	44,961
Other	42,706
Country	40,078
Alternative	35,900
Alternative Rock	35,525
Folk	32,108
Unknown	29,413
Punk	27,977
Hip-Hop	24,083
Blues	23,276
Indie	21,709
Classic Rock	18,417
Ambient	18,074
Industrial	17,816

(a) Genre as reported in file metadata.

Genre	%
Rock	41.15
Electronic	19.65
Pop	7.73
Jazz	6.80
Country	4.42
Folk	3.83
Rhythm & blues	3.61
Blues	2.86
Hip Hop	2.23
Classical	1.81
Asian	1.69
Caribbean & Latin	1.63
Ska	0.89
Avant-Garde	0.47
Easy Listening	0.45
Comedy	0.44
African	0.28
Other	0.09

(b) Percentages of broad genre categories.

Table 2: Genre statistics.

We find genre metadata present for 1,908,251 submissions. The top 20 genre annotations account for 52.9% of the tags used in this subset. We show the list of these genres and their counts in Table 2 (a). We also compute percentages over 691,431 recordings (41.4% of total recordings in AcousticBrainz) annotated by genre using Last.fm tags and shown in Table 2 (b). To find these broad genre labels we look up a recording by its MBID and if this fails, by the artist and track title. Last.fm tags are ranked by the most commonly applied tag. We match highly ranking tags to popular music genres found in beets, a tool for identifying, tagging, and renaming audio files, [7] . If a match occurs as a more specific subgenre, we report it as this subgenre's parent genre. While this process is lossy (we don't match tags which are misspelled) and subjective (not everyone agrees on genres or subgenres), we believe it nonetheless gives a good overview of the contents of the database.

2.3 Architecture

The architecture of AcousticBrainz is presented in Figure 2. The community uses the feature extractor and submission tools to send music features extracted from audio to the server. The server stores this data (which we call "low-level" data) in a database and makes it available to the rest of the community. The community can also provide classifier models (designed using the tools we provide), for inferring information from this data (which we call "high-level" data). The high-level data is computed on the server without needing to access audio files. The community can moderate the models and the good ones are used to compute high-level data for all AcousticBrainz submissions.

2.3.1 Feature extractor and submission tool

We have created a music feature extractor using the Essentia library. [8] We use this library for computing features because it has been successfully used in a number of similar audio analysis applications, such as Freesound, and other commercial systems. We describe the features computed by the extractor in more detail in Section 2.3.3. We distribute this extractor, written in C++, through our website [9] as a static binary for Windows, OSX, and Linux. We use a static binary because it lets us include the same version of all of our dependencies across all platforms.

The feature extractor runs at about 20× real time, that is, a file with length 3 minutes takes 9–10 seconds to run (on an Intel i5 3.30GHz machine).

We have two clients to help the community compute features on their audio files and submit them to the AcousticBrainz server—A command-line tool written in Python, and a graphical interface written in C++ with QT. These clients automatically search for all audio files in a directory, compute their features, and send JSON files containing the features to the server using its API.

The submission tool only submits data which have been previously tagged with MBIDs. Software exists to match audio files on disk to Releases on MusicBrainz based on track lengths, file names, existing tags, and audio fingerprinting. It is possible that audio files will be tagged incorrectly, either due to user error or incorrect fingerprint matching, however we believe this to account for only a small amount of data submitted.

Each JSON file contains metadata identifying the version of the feature extractor used, including information about the exact version the source code (git commit hash) and also an increasing version number which we will change as we make incompatible changes to features in the future.

2.3.2 Server

The features of submitted tracks are stored as JSON in a PostgreSQL database. The server interface is written in Python using the Flask web application framework. An API accepts requests from clients, filters exact duplicates (where the feature extractor outputs exactly the same content for two concurrent runs on the same file), and stores the results in the database. For privacy reasons, the server stores no identifying information about submitters.

Every 30 seconds the server starts a process to search for recent submissions. For these documents the server runs a feature extractor to obtain high-level descriptors for these files (Section 2.3.4). Once the high level computa-

[7] https://github.com/sampsyo/beets/blob/0c7823/beetsplug/lastgenre/genres-tree.yaml

[8] http://essentia.upf.edu

[9] http://acousticbrainz.org/download

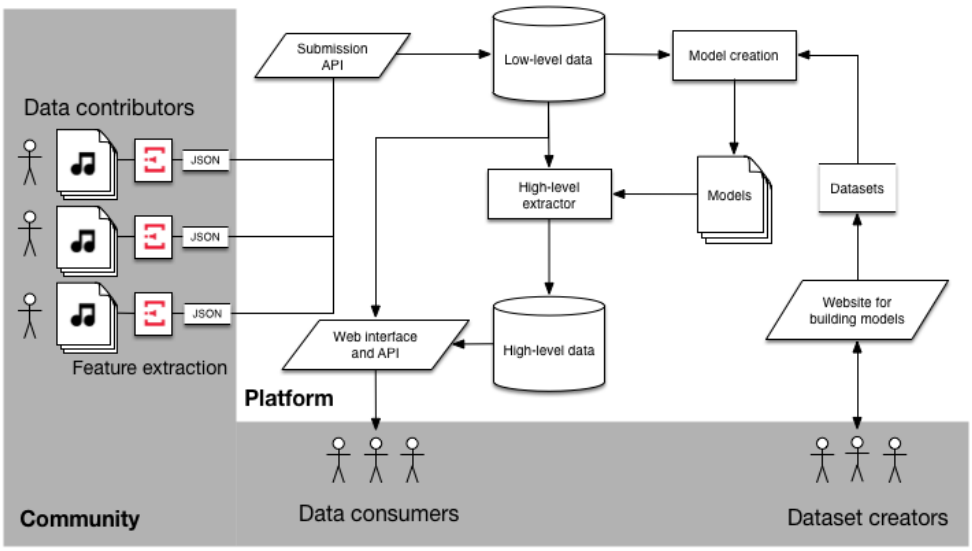


Figure 2: AcousticBrainz architecture.

tion process is complete, all of the data about the submitted track is made available to the community.

Once extracted features are made available we store all of the computed data and metadata collected from MusicBrainz in an ElasticSearch search server. This search system lets users perform queries such as finding all recordings with a particular attribute or attribute range (e.g., with a BPM between 110 and 120, or an estimated genre of jazz), or by filtering by some known metadata (such as all recordings by a particular artist).

All of the submitted and computed information is made available via the AcousticBrainz website and API. The website has a page for each submitted recording, outlining metadata, providing an overview of the low-level and high-level data, and linking to external sources, including a player to listen to the song if it is available on a public streaming service. The API gives access to the JSON documents that make up the low-level and high-level data, and access to the search interface. Documents are identified by their MBID. Groups of documents, for example all recordings in an album, can be downloaded by first getting the list of MBIDs from MusicBrainz.

2.3.3 *Low-level music data*

Our feature extractor computes spectral, time-domain, rhythm, and tonal descriptors. They include features characterizing overall loudness, dynamics, and spectral shape of the signal, rhythm descriptors (including beat positions and BPM value), and tonal information (including chroma features, keys and scales). All descriptors are analyzed on a signal resampled to 44.1kHz sampling rate, summed to mono and normalized using replay gain. Many of the descriptors are computed across frames and are therefore summarized by their statistical distribution (we currently do not provide per-frame information). More detailed information about the low-level data, including references to the employed MIR and audio analysis algorithms, is provided in the official documentation for Essentia, [10] or by reviewing the code. [11] An example of the output of the feature extractor can be seen on AcousticBrainz website. [12] We provide a list of music descriptors computed by the feature extractor and currently present in AcousticBrainz in Table 3.

2.3.4 *High-level music data*

Low-level data submitted by the users opens possibilities to apply data mining and machine learning techniques across the whole AcousticBrainz collection, or subsets, without needing access to audio files. In particular these techniques may allow us to infer semantic annotation of music in terms of concepts used by people when describing music (e.g., genres, styles, moods, uses of music, instrumentation, etc.) Currently, AcousticBrainz provides tools for creating datasets to represent these types of concepts and train classifier models (see Section 3). The training process is done automatically using SVM classifiers (C-SVC with polynomial or RBF kernels). A training script finds optimal data preprocessing and SVM parameterization given a ground-truth dataset of low-level data in a grid search using 5-fold cross-validation. The details on the considered parameters can be found in the classification project template in the source code. [13] After moderation the resulting high-level data can be computed from the low-level data in the AcousticBrainz database.

Our current high-level data includes estimations done by classifiers pre-trained using a number of annotated col-

[10] http://essentia.upf.edu/documentation/streaming_extractor_music.html
[11] https://github.com/MTG/essentia/tree/master/src/examples
[12] http://acousticbrainz.org/data
[13] https://github.com/MTG/gaia/tree/master/src/bindings/pygaia/scripts/classification

low-level.*	rhythm.*	tonal.*
average loudness, dynamic complexity, silence rate 20dB / 30dB / 60dB, spectral centroid / kurtosis / spread / skewness / rolloff / decrease, hfc, spectral strongpeak, zerocrossingrate, spectral rms, spectral flux, spectral energy, spectral energyband low / middle low / middle high / high, barkbands, melbands, erbbands, mfcc, gfcc, barkbands crest / flatness db / kurtosis / skewness / spread, melbands crest / flatness db / kurtosis / skewness / spread, erbbands crest / flatness db / kurtosis / skewness / spread, dissonance, spectral entropy, pitch salience, spectral complexity, spectral contrast coeffs / valleys	beats position, beats count, bpm, bpm histogram first peak bpm / spread / weight, bpm histogram second peak bpm / spread / weight, beats loudness, beats loudness band ratio, onset rate, danceability	tuning frequency, hpcp, thpcp, hpcp entropy, key key, key scale, key strength, chords strength, chords histogram, chords changes rate, chords number rate, chords key, chords scale, tuning diatonic strength, tuning equal tempered deviation, tuning nontempered energy ratio

Table 3: Descriptors extracted by Essentia's music extractor v1.0 currently present in AcousticBrainz. The descriptors are grouped according to the namespaces within the music extractor's output.

Name	Source	Type	Size
genre dortmund	Music Audio Benchmark Data Set [5]	Genre	1886 track excerpts, 46-490 per genre
genre rosamerica	In-house [4]	Genre	400 tracks, 50 per genre
genre tzanetakis	GTZAN Genre Collection [11]	Genre	1000 track excerpts, 100 per genre
genre electronic	In-house	Electronic music subgenres	250 track excerpts, 50 per genre
mood acoustic	In-house [8]	Sound (acoustic, non-acoustic)	321 full tracks + excerpts, 193/128 per class
mood electronic	In-house [8]	Sound (electronic, non-electronic)	332 full tracks + excerpts, 164/168 per class
timbre	In-house	Timbre colour (dark, bright)	3000 track excerpts, 1500 per class
tonal atonal	In-house	Tonality (tonal/atonal)	345 track excerpts, 200/145
danceability	In-house	Danceability	306 full tracks, 124/182 per class
ismir04 rhythm	ISMIR2004 Rhythm Classification Dataset [3]	Ballroom music dance styles	683 track excerpts, 60-110 per class
voice instrumental	In-house	Voice/instrumental music	1000 track excerpts, 500 per class
gender	In-house	Gender in vocal music (male/female)	3311 full tracks, 1508/1803 per class
mood happy	In-house [8]	Mood (happy, non-happy)	302 full tracks + excerpts, 139/163 per class
mood sad	In-house [8]	Mood (sad, non-sad)	230 full tracks + excerpts, 96/134 per class
mood aggressive	In-house [8]	Mood (aggressive, non-aggressive)	280 full tracks + excerpts, 133/147 per class
mood relaxed	In-house [8]	Mood (relaxed, non-relaxed)	446 full tracks + excerpts, 145/301 per class
mood party	In-house [8]	Mood (party, non-party)	349 full tracks + excerpts, 198/151 per class
moods mirex	MIREX Audio Mood Classification Dataset [6]	Mood (5 clusters)	269 track excerpts, 60-110 per class

Table 4: Music collections used for training high-level classifier models currently included in AcousticBrainz.

lections, some of which are commonly used in MIR (Table 4). These datasets pre-date the AcousticBrainz platform and so some of them are not yet open to inspection. We anticipate that the community can help to build better classifiers using the low-level data already submitted to AcousticBrainz.

The evaluation metrics obtained from training our current models [14] show promising results. However, the accuracy and reliability of our current high-level data is under doubt, as little research on the portability of such models to the large scale has been done within MIR. We see the design of new classifier models using AcousticBrainz data as an attractive challenge for MIR researchers and we anticipate AcousticBrainz to become a platform for building classifiers on larger collections created and annotated by the community using the tools we provide (see Section 3). The high-level data within AcousticBrainz will be constantly updated using the improved classifier models proposed by the community.

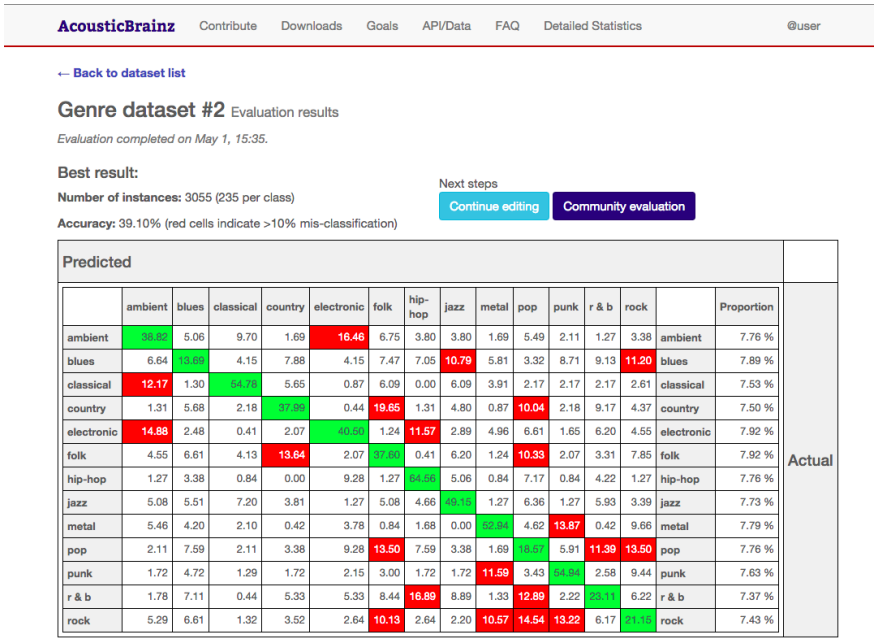
3. BUILDING ANNOTATED DATASETS

We have developed an interface which lets users create datasets, comprised of a name, a list of classes, and a list of instances for each class. These instances refer to recordings in the AcousticBrainz database, and so are referred to by MBID. MBIDs can be chosen manually, or added as the result of a search query for all recordings matching given criteria. To assist in the inspection of datasets, metadata of these recordings from MusicBrainz is also shown. Users can create these datasets individually or collaborate together to suggest classes, class boundaries, and content. We currently limit our interest to classification problems, though we see future value in allowing users to create other kinds of annotated datasets such as collections of singular types of data (e.g., music from a specific culture), user-defined lists of recordings, or sets of recordings with a freeform annotations including tags.

Once a dataset has been created, a user can choose to generate a model representing the dataset. This model is trained using the same training script used to generate our existing models (Section 2.3.4). We report to the user the accuracy of the model, giving them the chance to share the results with the wider community, or continue improving the model (Figure 3).

Once a model has been created and approved by the community we can choose to process all existing low-level data with this model in order to make these new estimations available for the community. We are able to compute high-level data at a rate of about 1000/minute using a sin-

[14] `http://acousticbrainz.org/data`



	ambient	blues	classical	country	electronic	folk	hip-hop	jazz	metal	pop	punk	r & b	rock		Proportion
ambient	38.82	5.06	9.70	1.69	16.46	6.75	3.80	3.80	1.69	5.49	2.11	1.27	3.38	ambient	7.76 %
blues	6.64	13.69	4.15	7.88	4.15	7.47	7.05	10.79	5.81	3.32	8.71	9.13	11.20	blues	7.89 %
classical	12.17	1.30	64.78	5.65	0.87	6.09	0.00	6.09	3.91	2.17	2.17	2.17	2.61	classical	7.53 %
country	1.31	5.68	2.18	37.99	0.44	19.65	1.31	4.80	0.87	10.04	2.18	9.17	4.37	country	7.50 %
electronic	14.88	2.48	0.41	2.07	40.50	1.24	11.57	2.89	4.96	6.61	1.65	6.20	4.55	electronic	7.92 %
folk	4.55	6.61	4.13	13.64	2.07	37.50	0.41	6.20	1.24	10.33	2.07	3.31	7.85	folk	7.92 %
hip-hop	1.27	3.38	0.84	0.00	9.28	1.27	64.56	5.06	0.84	7.17	0.84	4.22	1.27	hip-hop	7.76 %
jazz	5.08	5.51	7.20	3.81	1.27	5.08	4.66	49.15	1.27	6.36	1.27	5.93	3.39	jazz	7.73 %
metal	5.46	4.20	2.10	0.42	3.78	0.84	1.68	0.00	52.94	4.62	13.87	0.42	9.66	metal	7.79 %
pop	2.11	7.59	2.11	3.38	9.28	13.50	7.59	3.38	1.69	16.67	5.91	11.39	13.50	pop	7.76 %
punk	1.72	4.72	1.29	1.72	2.15	3.00	1.72	1.72	11.59	3.43	54.94	2.58	9.44	punk	7.63 %
r & b	1.78	7.11	0.44	5.33	5.33	8.44	16.89	8.89	1.33	12.89	2.22	23.11	6.22	r & b	7.37 %
rock	5.29	6.61	1.32	3.52	2.64	10.13	2.64	2.20	10.57	14.54	13.22	6.17	21.16	rock	7.43 %

Figure 3: Results of a classification. The user can choose to continue working on improving the classifier accuracy, or submit it to the community.

gle core, and so anticipate that recomputing the dataset at its current size will only take a few days. As the dataset grows the task can be parallelized over many machines.

4. CHALLENGES AND FUTURE WORK

To keep our low-level data at the level of the state of the art in MIR, we will continue to release updates to the feature extractor, and we also encourage participation in this process. Because we rely on the good will of the community to run this extractor on their audio collections we face a trade-off between the frequency of updates and their willingness to run the extractor. We anticipate that we could release an update once or twice a year, increasing the number and quality of the features. Our high-level data will also be under constant improvement. We hope that the system that we have developed will foster collaboration to build better annotations of musically useful concepts.

Other datasets, such as the MSD contain more detailed features than those which we compute for our low-level data. The continual testing and improvement and integration of new algorithms will allow us to close this gap of feature content. Since we rely on contributions by the community, we may be missing some popular music. Continuing to solicit requests will ensure we have as broad a coverage as possible. While soliciting audio features we have to ensure that incorrect submissions are not made, either maliciously or due to incorrect metadata. We are developing a technique to determine if two submissions are identical based on their features.

Updating the feature extractor and classifier models implies compatibility problems with our data. As our submitted data includes information about the version of the extractor used to compute it, we can determine if two pieces of data computed by different versions of the feature extractor are compatible. We are compiling a dedicated audio collection to perform tests with different extractor versions and estimate the differences in feature values. These tests can also help us to assess the robustness of music features present in low-level data, the identification of which is a challenging task [12]. To take advantage of as much data as possible, we will not discard old submissions from our database when a new extractors are released. High-level data will be updated with respect to low-level data when possible. Improvements to the collection creation interface on the AcousticBrainz website will let us build datasets to use with other machine learning techniques.

We expect that the data provided by AcousticBrainz will be useful to both the MIR community and others interested in this type of data. In exchange we need the AcousticBrainz community to help in expanding the dataset and improving its quality. The interest in our platform became apparent directly after its launch when we were able to obtain features for 500,000 files in less than 3 weeks, building up to over 2.7 million submissions. The continued support of providing features and collaborating on data collection projects will ensure the success of this project.

5. ACKNOWLEDGEMENTS

This research has been partially funded by the CompMusic (ERC 267583) and SIGMUS (TIN2012-36650) projects The authors would like to thank the entire MusicBrainz community and the members of the AcousticBrainz community who helped to test the feature extractor and contribute data to the project.

6. REFERENCES

[1] T. Bertin-Mahieux, D. PW Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 591–6, 2011.

[2] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J.R. Zapata, and X. Serra. Essentia: An audio analysis library for music information retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 493–498, 2013.

[3] P. Cano, E. Gómez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack. ISMIR 2004 audio description contest. Technical report, 2006. Available online: http://mtg.upf.edu/node/461.

[4] E. Guaus. *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers*. PhD thesis, Universitat Pompeu Fabra, 2009.

[5] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst. A benchmark dataset for audio classification and clustering. In *Proceedings of the International Conference on Music Information Retrieval*, pages 528–31, 2005.

[6] X. Hu and J. S. Downie. Exploring mood metadata: Relationships with genre, artist and usage metadata. In *Proceedings of the International Conference on Music Information Retrieval*, pages 67–72. Citeseer, 2007.

[7] D. Karydi, I. Karydis, and I. Deliyannis. Legal issues in using musical content from iTunes and YouTube for music information retrieval. In *International Conference on Information Law*, 2012.

[8] C. Laurier, O. Meyers, J. Serrà, M. Blech, and P. Herrera. Music Mood Annotator Design and Integration. In *International Workshop on Content-Based Multimedia Indexing*, 2009.

[9] A. Schindler, R. Mayer, and A. Rauber. Facilitating comprehensive benchmarking experiments on the million song dataset. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 469–74, 2012.

[10] B. L. Sturm. An analysis of the gtzan music genre dataset. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 7–12. ACM, 2012.

[11] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[12] J. Urbano, D. Bogdanov, P. Herrera, E. Gómez, and X. Serra. What is the effect of audio quality on the robustness of mfccs and chroma features? In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 573–8, 2014.