Deliverable D4.4

Evaluation report on the first prototype tool for the automatic semantic description of music samples

Grant agreement nr	688382
Project full title	Audio Commons: An Ecosystem for Creative Reuse of Audio Content
Project acronym	AudioCommons
Project duration	36 Months (February 2016 - January 2019)
Work package	WP4
Due date	31 July 2017 (M18)
Submission date	31 July 2017 (M18)
Report availability	Public (X), Confidential ( )
Deliverable type	Report (X), Demonstrator ( ), Other ( )
Task leader	MTG-UPF
Authors	Frederic Font, Dmitry Bogdanov
Document status	Draft ( ), Final (X)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Table of contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Executive Summary

As part of the Audio Commons Ecosystem, a number of tools are provided for the automatic analysis of audio content without the need for human intervention. These tools are designed for extracting i) musical audio properties for music pieces and music samples, and ii) non-musical audio properties for any kind of sounds. This deliverable addresses the evaluation of the first version of the Audio Commons tool for the automatic analysis of music samples.

For this evaluation we focus on those algorithms included in the annotation tool that provide estimates for specific musical properties. In particular, these algorithms include tempo estimation, key estimation and pitch estimation. We carry out evaluation tasks for each of these musical properties, and compare the algorithm included in the annotation tool with other algorithms described in the literature. To increase the validity of our results, for each of the evaluation tasks we use several datasets which contain content from different sources and different quality levels.

The evaluation results show that the algorithms included in the first version of the Audio Commons annotation tool for music samples tend to perform similar to the state of the art algorithms, although in some cases adopting the best scoring algorithm instead of the one currently included would definitely increase overall accuracies. Results also show that obtained accuracies are still below what would be desirable for a good user experience (e.g. when users search for Audio Commons content using any of these musical properties). At the end of this deliverable we propose some ideas about how can this be addresses and algorithms complemented to be able to offer a better user experience.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Background

This deliverable is part of the work package WP4 Semantic annotation of musical sound properties. In this work package, we first reviewed the state of the art in automatic extraction of music properties from audio signals (deliverable D4.1), and then released two tools for the automatic annotation of music samples and music pieces (D4.2 and D4.3, respectively). The present deliverable provides an evaluation of the tool described in D4.2. It is recommended to read the present deliverable only after reading deliverable D4.2. In the second half of the Audio Commons project, the tool for the automatic annotation of music samples will be updated (according to the feedback provided in the present deliverable), and evaluated again, yielding deliverables D4.7 and D4.10, respectively.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

1 Introduction

1.1 Main objectives and goals

To goal of this deliverable is to assess the effectivity of the automatic annotation algorithms for music samples[1] introduced in the previous deliverable D4.2. To this end, a number of evaluation tasks have been carried out that allow us to estimate the accuracy of the algorithms. The results of this evaluation will inform the next phases of development for the automatic annotation algorithms.

1.2 Methodology

To evaluate the algorithms we first grouped them into categories (following the grouping proposed in deliverable D4.1Report on the analysis and compilation of state-of-the-art methods for the automatic annotation of music pieces and music), and then ran an evaluation task for the algorithms that generate estimates. The following table shows how the algorithms are grouped, what the basis of their evaluation was and in which section of the present document it can be found. The evaluation tasks we carry out are based on the standards followed on existing sound and music computing literature. At the end of this document, a conclusion section provides a summary of the performance of all of the algorithms.

Name	Description	Group	Evaluation strategy
`ac:duration`	Duration of audio	File metadata	-
`ac:lossless`	Whether audio file is in lossless codec (1 or 0)	File metadata	-
`ac:codec`	Codec used for encoding the audio (e.g. pcm_s16le)	File metadata	-
`ac:filesize`	Size of the file	File metadata	-
`ac:bitrate`	Number of bits per second	File metadata	-
`ac:samplerate`	Number of samples per second	File metadata	-
`ac:channels`	Number of channels	File metadata	-
`ac:audio_md5`	MD5 checksum of raw undecoded audio payload. It can be used as a unique identifier of audio content.	File metadata	-
`ac:key`	Key and scale (e.g. A minor)	Harmony	Accuracy based on ground truth (Section 2)
`ac:tempo`	Tempo in BPM of the audio signal	Rhythm	Accuracy based on ground truth (Section 3)

[1] In this document this tool is also referred as the “Audio Commons extractor”.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 688382

`ac:note / ac:midi_note`	Played note name (e.g. C4) / Played note midi number (e.g. 60)	Pitch	Accuracy based on ground truth (Section 4)
`ac:loudness`	Loudness value	Dynamics	-
`ac:dynamic_range`	Dynamic range of audio recording	Dynamics	-
`ac:temporal_centroid`	Temporal centroid	Dynamics	-
`ac:log_attack_time`	Logarithm of the time it takes to reach maximum amplitude of audio signal (good for perceptual attack)	Dynamics	-

1.3 Terminology

AudioCommons: reference to the EC H2020 funded project AudioCommons, with grant agreement nr 688382.

Audio Commons Initiative: reference to the AudioCommons project core ideas beyond the lifetime and specific scope of the funded project. The term "Audio Commons Initiative" is used to imply i) our will to continue supporting the Audio Commons Ecosystem and its ideas after the lifetime of the funded project, and ii) our will to engage new stakeholders which are not officially part of the project consortium.

Audio Commons: generic reference to the Audio Commons core ideas, without distinguishing between the concept of the initiative and the actual funded project.

Audio Commons Ecosystem (ACE): set of interconnected tools, technologies, content, users and other actors involved in publishing and consuming Audio Commons content.

Audio Commons content (ACC): audio content released under Creative Commons licenses and enhanced with meaningful contextual information (e.g., annotations, license information) that enables its publication in the ACE.

Content creator: individual users, industries or other actors that create audio content and publish in the ACE through content providers.

Content provider: services that expose content created by content creators to the ACE.

Content user: individual users, industries or other actors that use the content exposed by content providers and created by content creators in their creative workflows.

Tool developer: individual users, industries or other actors that develop tools for consuming (and also potentially publishing) Audio Commons content.

Embeddable tools: tools for consuming Audio Commons content that can be embedded in existing production workflows of creative industries.

2 Evaluation of Tempo Estimation

In our evaluation context, the goal of the tempo estimation task is to provide a value in beats per minute (BPM) which matches the periodic rhythmic patterns of a given audio signal. Therefore we might refer to "tempo" or "BPM" interchangeably. Because the tool we are evaluating is aimed at the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

automatic description of sound samples (as opposed to music pieces), the concept of BPM is mostly relevant in the case of *music loops*. Music loops are typically used in music production environments as building blocks for a music composition. Musicians and producers can combine loops that can be played along together either by selecting loops with the same or compatible musical properties (e.g. tempo, key, genre) or by modifying existing ones. In either case, being able to browse music loops using such properties is crucial for the production process. As was shown in deliverable D2.1: Requirements Report and Use Cases, tempo is one of the most used musical properties for browsing content, therefore good tempo estimation algorithms are crucial for the meaningful annotation of Audio Commons content.

2.1 Datasets

To carry out the tempo estimation evaluation task, three datasets have been compiled which add up to more than 14k music loops of different production quality, music styles and availability. These are the datasets that we used for the tempo estimation task:

- **Freesound Loops 4000 (FSL4)**: This dataset contains user-contributed loops uploaded to Freesound. It has been built in-house by searching Freesound for sounds with the query terms *loop* and *bpm*, and then automatically parsing the returned sound filenames, tags and textual descriptions to identify tempo annotations made by users. For example, a sound containing the tag *120bpm* is considered to have a ground truth of 120 BPM.

- **Apple Loops (APPL)**: This dataset is composed of the audio loops bundled in Apple's Logic Pro[2] music production software. We parsed the metadata embedded in the audio files using source code available in a public repository[3], and extracted in this way tempo annotations for all the loops. This dataset contains professionally produced loops which are made available in Logic Pro and designed for making music by combining loops from the library. It is the most diverse and complete in terms of music genres, instrumentation, etc.

- **Mixcraft (MIXL)**: This dataset contains all the loops bundled with Acoustica's Mixcraft 7 music production software[4]. Tempo annotations are provided in its loop browser and can be easily exported into a machine-readable format. The dataset aggregates content from different loop sites and is professionally curated by Acoustica.

The following table includes some general statistics about the datasets:

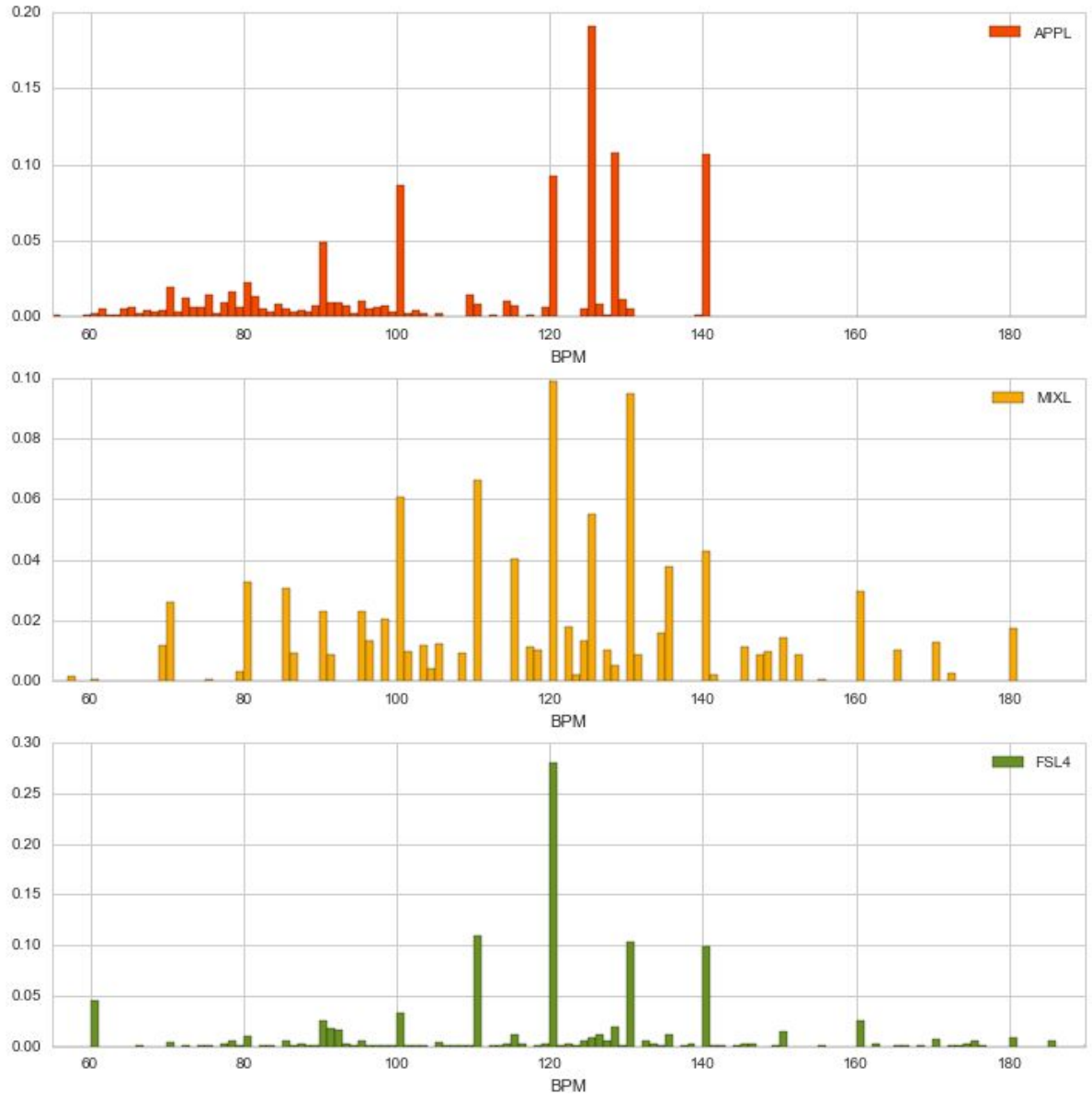
Dataset	N. sounds	Total duration	Mean duration	Duration range
APPL	4611	9h 43m	7.47s	1.32s - 40.05s
MIXL	5451	14h 11m	9.37s	0.32s - 110.77s
FSL4	3939	8h 22m	7.63s	0.15s - 30.0s

The plot below shows a histogram of the annotated BPM for each of the datasets. What this histogram tells us is that there are a number of typical BPM values (e.g. 100, 120, 140) which concentrate the majority of sounds in the datasets. Algorithms that are able to correctly estimate

[2] http://apple.com/logic-pro
[3] http://github.com/jhorology/apple-loops-meta-reader
[4] http://acoustica.com/mixcraft

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

BPM in these tempo ranges will perform well in the evaluation stage. Interestingly, the two datasets which come from professional companies (APPL and MIXL) seem to be slightly more evenly distributed than the one coming from Freesound. There are still prominent peaks representing common tempos in those datasets, but sounds are more widely distributed in other tempos as well.



2.2 Analysis algorithms

We evaluated the following algorithms for tempo estimation:

- **ACExtractorV1**: This is the algorithm included in the first version of the Audio Commons extractor. It uses the tempo estimation method by Degara et. al. [Degara12], which is a probabilistic approach for beat tracking based on inter-onset-interval times and a salience measure for individual beat estimates. This method builds from previous probabilistic beat

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

tracking methods such as Klapuri et. al [Klapuri06]. The final estimated tempo is given based on the mean of estimated beat intervals (see RhythmExtractor2013 algorithm[5]).

- **Percival14**: Percival and Tzanetakis [Percival14] describe a tempo estimation algorithm optimised for low computational complexity that combines several ideas from existing tempo estimation algorithms and simplifies their steps. The algorithm computes an onset strength function based on filtered spectral flux from which tempo lag candidates are estimated using autocorrelation. The most prominent tempo lag is selected and a simple decision tree algorithm is used to chose the octave of the final BPM output. We use a Python implementation of the algorithm provided by the authors in their original paper[6].

- **Böck15**: Böck et. al. [Böck15] propose a novel tempo estimation algorithm based on a recurrent neural network that learn an intermediate beat-level representation of the audio signal which is then feed to a bank of resonating comb filters to estimate the dominant tempo. This algorithm got the highest score in ISMIR 2015 Audio Tempo Estimation task. An implementation by the authors is included in the open-source Madmom audio signal processing library[7].

- **RekBox**: We also include an algorithm from a commercial DJ software, Rekordbox[8]. Details on how the algorithm works are not revealed, but a freely downloadable application is provided that can analyse a music collection and export the results in a machine-readable format.

2.3 Evaluation metrics

For testing the above algorithms against the three collected datasets we follow standard practice and adopt the methodology described by Gouyon et al. [Gouyon06]. This methodology includes the use of *Accuracy 1* and *Accuracy 2* evaluation metrics. In addition to these metrics, we add an extra measure that we call *Accuracy 1e*. Evaluation metrics are defined as follows:

- **Accuracy1**: percentage of instances whose estimated BPM is within 4% of the annotated ground truth.

- **Accuracy2**: percentage of instances whose estimated BPM is within a 4% of a multiple of ⅓, ½, 1, 2, or 3 times the ground truth BPM.

- **Accuracy1e**: percentage of instances whose estimated BPM is exactly the same as the ground truth after rounding the estimated BPM to the nearest integer.

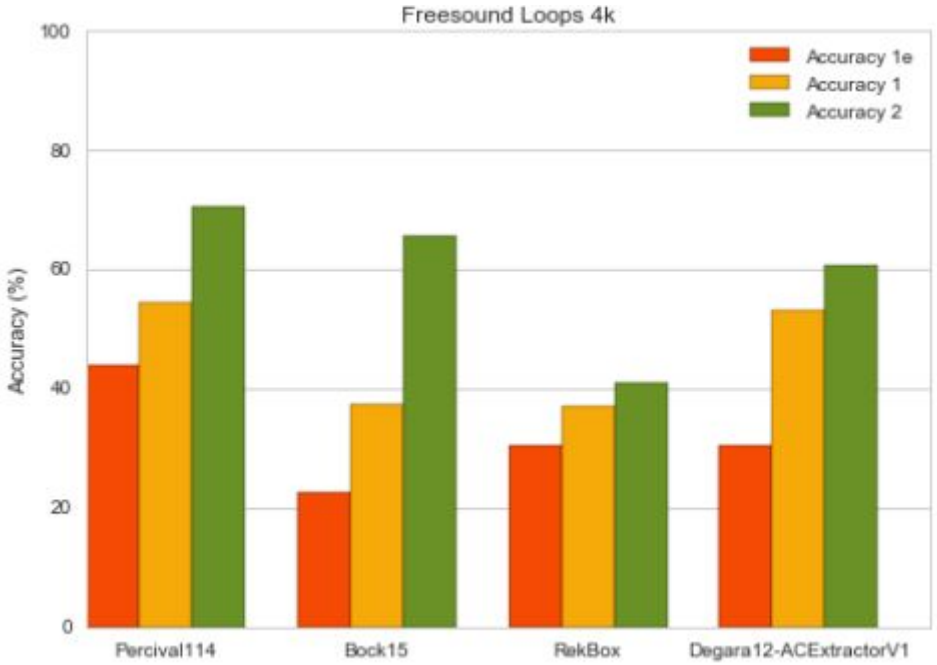
As it can be seen, Accuracy 1e is more strict than Accuracy 1. The reason why we added this extra accuracy measure is that the tool we are evaluating is for the annotation of music samples. Therefore, imagining a music creation context where loops can be queried in a database, it is of special relevance to get returned instances whose BPM exactly matches that specified as target.

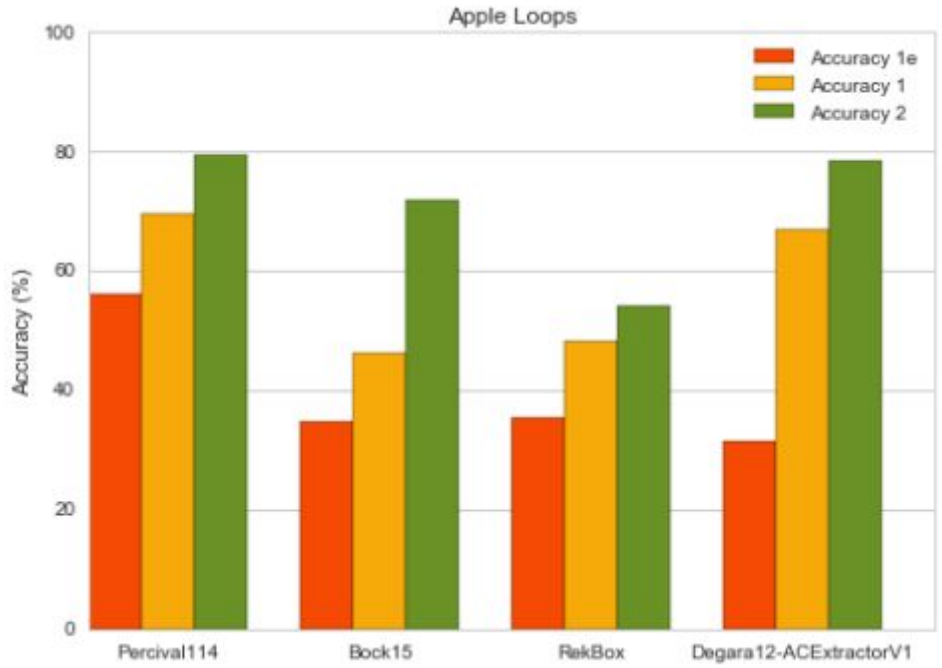
2.4 Results

The following figures show the results of each algorithm in each of the analysed datasets.

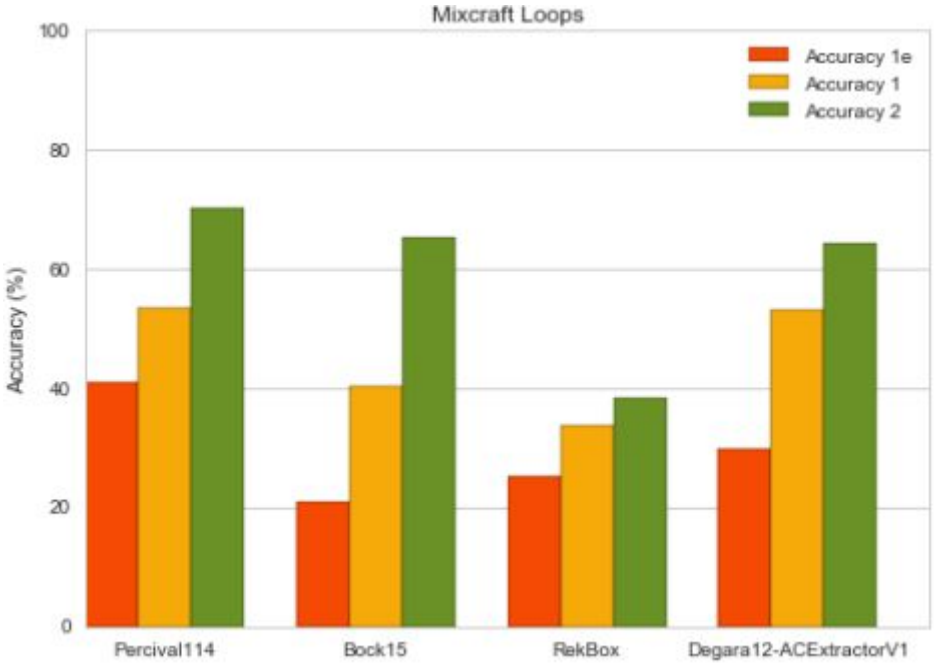
[5] http://essentia.upf.edu/documentation/algorithms_reference.html
[6] http://opihi.cs.uvic.ca/tempo
[7] http://github.com/CPJKU/madmom
[8] http://rekordbox.com

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382





This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382



The following table shows the average accuracy (in %) when taking into account all datasets at once. Marked in orange are the results of the algorithm included in the Audio Commons tool for the annotation of music samples:

Method	Accuracy 1e	Accuracy 1	Accuracy 2	Mean accuracy
Percival14	46.72	59.03	73.29	59.68
ACExtractorV1	30.46	57.58	67.77	51.94
Böck15	25.93	41.45	67.43	44.94
RekBox	30.01	39.34	44.33	37.89

Overall accuracy results show that Percival14 obtains the highest accuracy scores for all accuracy measures and all datasets. Considering the data from all datasets at once, mean accuracy values for Percival14 range from 47% (Accuracy 1e) to 73% (Accuracy 2), on average a 7% higher accuracy when compared to the second best-scoring method. With a few exceptions, pairwise accuracy differences between Percival14 and the second best-scored method in all datasets and accuracy measures are statistically significant using McNemar's test and a significance value of α=0.01 (i.e., $p \ll 0.01$).

We also observe that accuracies for the APPL dataset tend to be higher than for other datasets. This can be explained by the fact that APPL contains professionally created and curated loops, while the other datasets contain user contributed content, not necessarily created by professionals. As a conclusion, the algorithm included in the Audio Commons extractor performs relatively well compared to other algorithms, but there is room for improvement to reach the accuracies of the best performing algorithm in our comparison. In a future version of the Audio Commons extractor, we should consider including Percival14 method instead.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

3 Evaluation of Key Estimation

According to the aforementioned deliverable D2.1, musical key is another of the most important musical properties that are needed for browsing musical content in a context of music production. Similarly to the case of the evaluation of tempo estimation, here we carry out our evaluation aimed at key estimation for music loops. In our evaluation we assume that the music loops we analyze have a single music key (no significant modulations), and that the key is one of the 24 keys that can be formed with the combination of 12 note roots and two modes (major and minor). This approach is quite common in key estimation evaluation works [Raffel14].

3.1 Datasets

To carry out the key estimation evaluation task, three datasets have been compiled which add up to more than 6k music loops/song excerpts of different production quality, music styles and availability. These are the datasets that we used for the key estimation task:

- **Apple Loops (APPL)**: This dataset is composed of the audio loops bundled in Apple's Logic Pro[9] music production software. We parsed the metadata embedded in the audio files using source code available in a public repository[10], and extracted in this way key annotations for all the loops. This dataset contains professionally produced loops which are made available in Logic Pro and designed for making music by combining loops from the library. It is the most diverse and complete in terms of music genres, instrumentation, etc.

- **Mixcraft (MIXL)**: This dataset contains all the loops bundled with Acoustica's Mixcraft 7 music production software[11]. Key annotations are provided in its loop browser and can be easily exported into a machine-readable format. The dataset aggregates content from different loop sites and is professionally curated by Acoustica.

- **GiantSteps Key (GSKY)**: This dataset was compiled in the Giant Steps project [Knees15] and is available through a public Github repository[12]. The dataset contains short fragments of electronic dance music tracks. Music tracks and annotations are extracted from Beatport[13], a digital record store targeted at DJs and focusing on EDM genres. For the audio, only the previews provided by Beatport are used.

The following table includes some general statistics about the datasets:

Dataset	N. sounds	Total duration	Mean duration	Duration range
APPL	2743	6h 15m	8.21s	2.07s - 40.05s
MIXL	2935	9h 37m	11.8s	0.86s - 82.29s
GSKY	604	20h 4m	119.61s	44.7s - 120.01s

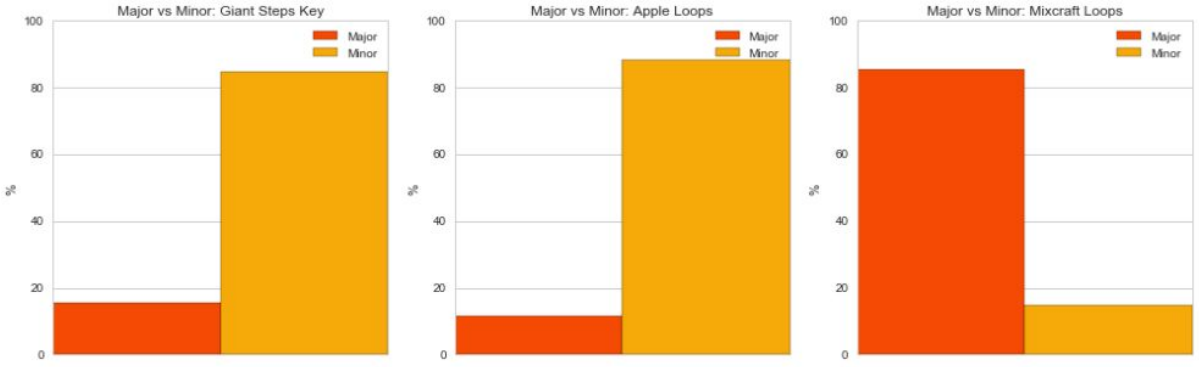
Note that the sizes of APPL and MIXL are smaller than those for the same datasets in the tempo estimation task. This is because not all of the loops in the datasets contain harmonic instruments

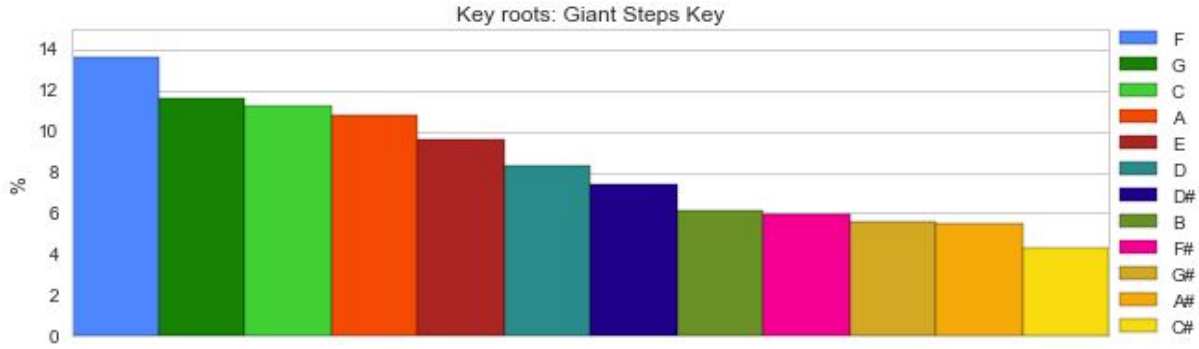
[9] http://apple.com/logic-pro
[10] http://github.com/jhorology/apple-loops-meta-reader
[11] http://acoustica.com/mixcraft
[12] https://github.com/GiantSteps/giantsteps-key-dataset
[13] http://beatport.com

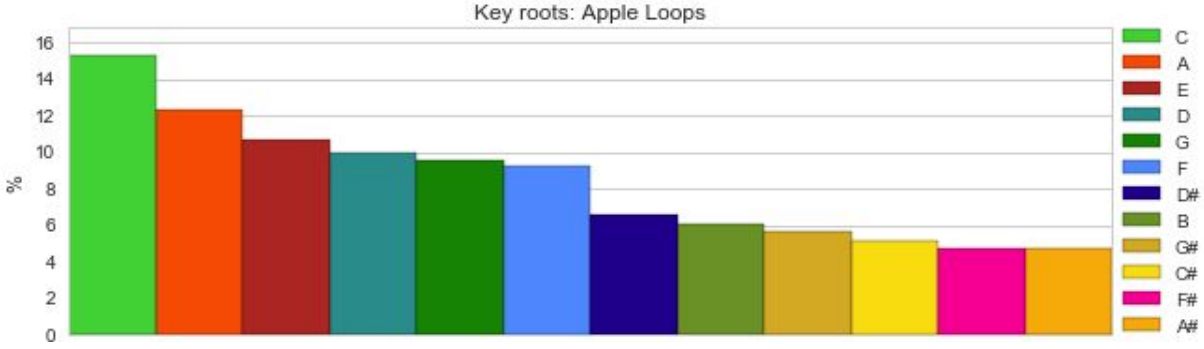
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

that bring a certain key to the loop. The datasets used here are those subsets for which key estimation is meaningful.

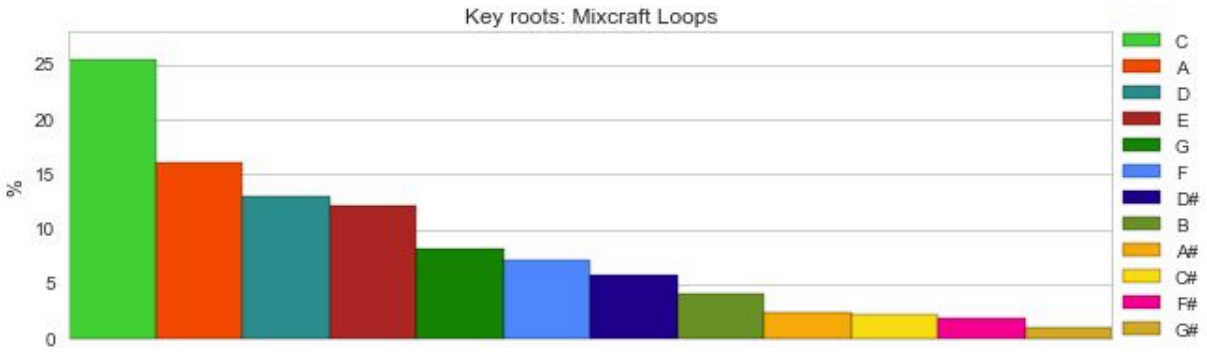
The plots below show the distribution of major/minor items in the each dataset and the most common roots. Surprisingly enough, both APPL and GSKY have a clear bias for minor tonalities, whereas the MIXL dataset features a clear bias for major tonalities. This may be due to the fact that the datasets have different genre coverage (for example, GSKY is focused on electronic dance music, for which minor keys are very common). Regarding key roots, all dataset feature a similar top 5. GSKY features a different ordering in the top 5 roots which can be due to the fact which is the only dataset made out of song excerpts and not loops which were designed for being used as building blocks for other compositions.







This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382



3.2 Analysis algorithms

We incorporated the following algorithms in our evaluation of key estimation:

- **ACExtractorV1**: This algorithm included in the first version of the Audio Commons extractor is a standard key estimation algorithm based on HPCP extraction and key profile matching [Gómez06]. Key profiles are extracted from a corpus of European Classical Music [Temperley99]. To provide a single key estimate, HPCP values for each frames are averaged and then compared with the profiles. We use the implementation provided in Essentia, in particular the one included in the FreesoundTonalExtractor[14].

- **QMULKey**: This algorithm uses the VAMP plugin for Key Detector [Noland07] released by Queen Mary University of London[15]. The Key Detector algorithm analyses a single channel of audio and continuously estimates the key of the music by comparing the degree to which a block-by-block chromagram correlates to the stored key profiles for each major and minor key. The key profiles are drawn from analysis of Book I of the Well Tempered Klavier by J S Bach, recorded at A=440 equal temperament. To provide the overall key we aggregate the result for each frame and take the most repeated (the most common) estimated key.

- **EDMKey#**: This algorithm is developed as an improvement to the existing key estimation method implemented in Essentia (same as in ACExtractorV1) by "introducing a system of multiple profiles and addressing difficult minor tracks as well as possibly amodal ones" [Faraldo17]. The algorithm also performs a HPCP cleaning stage to remove bins with low values which potentially only add noise. The original key profiles used in this algorithm are optimized for electronic dance music (EDM), which also makes it significantly different from the other two methods. In our evaluation we include 3 variations of the EDMKey algorithm which use profiles derived from electronic and popular music (EDMKey1 and EDMKey2) and from Temperley et al. [REF to temperley] (EDMKey3).

14 https://github.com/MTG/essentia/blob/master/src/essentia/utils/extractor_freesound/FreesoundTonalDescriptors.cpp

15 http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html#qm-keydetector

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

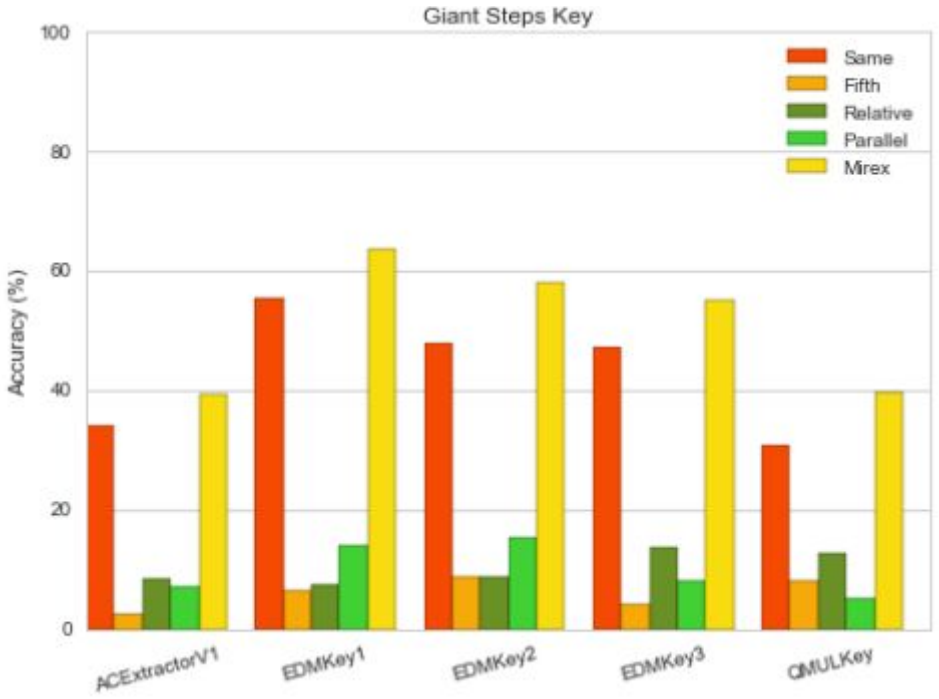
3.3 Evaluation metrics

We evaluate the key estimation task with a number of evaluation metrics which are common in the field. The evaluation metrics are defined as follows:

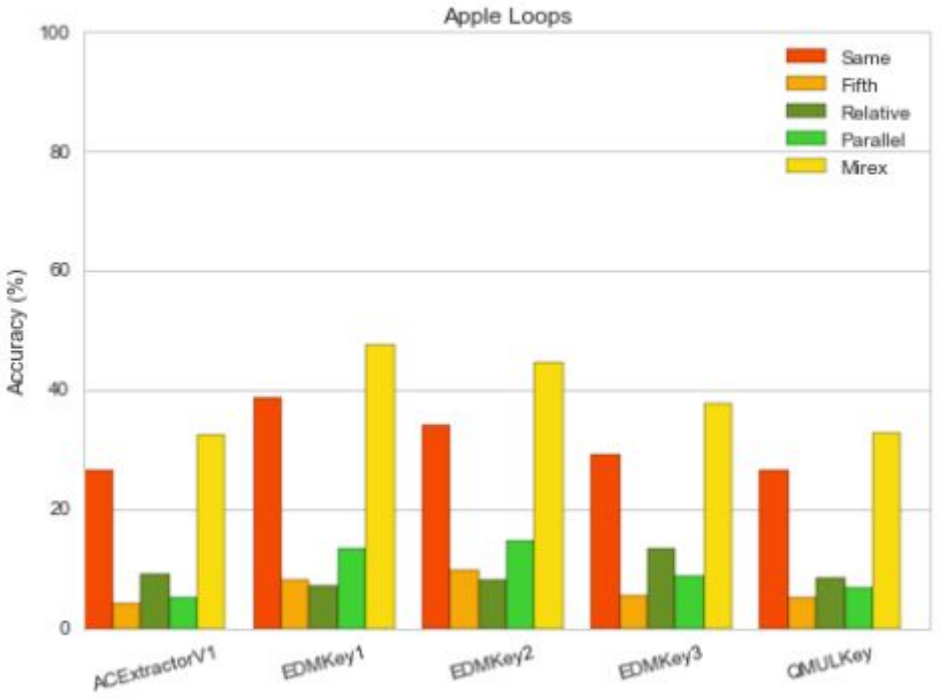
- **Same**: This metric reports the percentage of key estimations in which both the root and the mode are equal to the annotated ground truth. These are the "exact" matches.
- **Fifth**: This metric reports the percentage of key estimations which have a relation of a perfect fifth with the annotated ground truth.
- **Relative**: This metric reports the percentage of key estimations which have a relation of a relative minor (or relative major) with the annotated ground truth.
- **Parallel**: This metric reports the percentage of key estimations which have a relation of a parallel minor (or parallel major) with the annotated ground truth. That is to say, those in which the root is correctly estimated but the mode is wrong.
- **Mirex**: This is a combined evaluation metric which computes the ones described above and weights them into a single score. In particular, the "Mirex" score can be computed as *1 * Correct + 0.5 * Fifth + 0.3 * Relative + 0.1 * Parallel*.

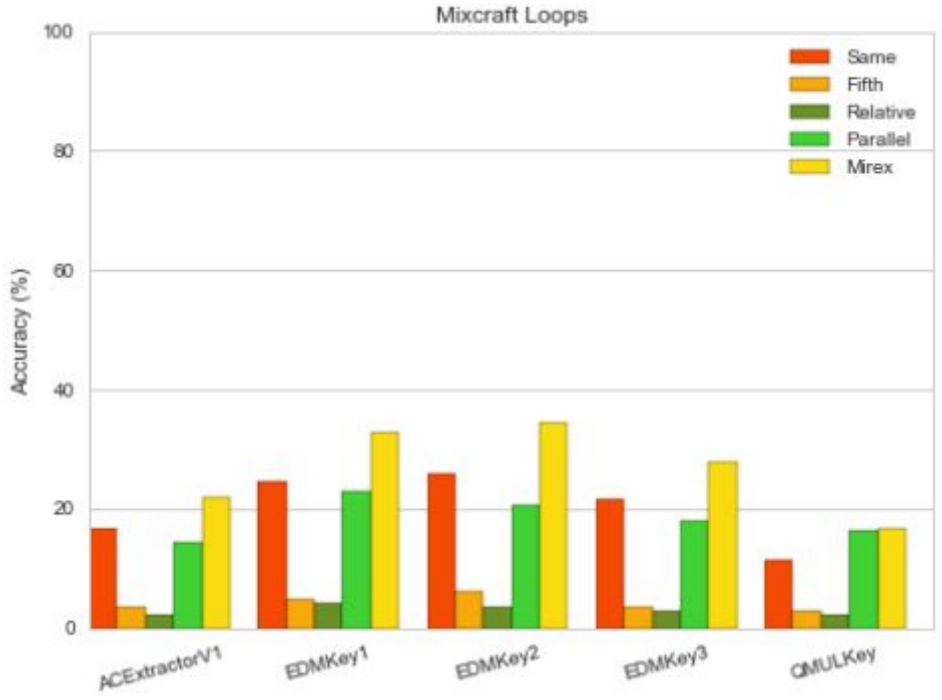
3.4 Results

The following figures show the results of each algorithm in each of the analysed datasets.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382





The following table shows the average accuracy (in %) when taking into account all datasets at once. Marked in orange are the results of the algorithm included in the Audio Commons tool for the annotation of music samples:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

Method	Same	Fifth	Relative	Parallel	Mirex
EDMKey1	33.60	6.59	5.83	17.89	42.23
EDMKey2	31.60	8.07	6.19	17.54	41.0
EDMKey3	27.33	4.44	8.56	12.99	34.72
ACExtractorV1	22.64	3.71	5.97	9.79	28.24
QMULKey	19.82	4.44	5.91	11.16	26.04

In general, we observe that the algorithm we use in the Audio Commons extractor is well below the state of the art and the highest score obtained by EDMKey1. This can be expected as EDMKey1 is an improvement over the algorithm included in ACExtractorV1. A possible reason why both ACExtractorV1 and QMULKey feature such low results when compared to the top rated algorithms is that the key profiles used are built from classical music. Even though the loop collections feature different music genres, most of them clearly belong to modern music and are closer to pop an electronic music than to classical music. In the context of music samples in Audio Commons it seems to make sense to use key profiles based on modern music. In future versions of the Audio Commons extractor tool we should consider incorporating the EDMKey1 algorithm.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

4 Evaluation of Pitch Estimation

In the previous evaluation tasks we focused on musical properties that make sense for music fragments long enough to introduce a notion of tempo and tonality. In this case we focus the evaluation in another relevant musical property for shorter music sounds. In particular we look into pitch estimation for recordings of single notes from pitched musical instruments. This is important as proper pitch annotation allows the reuse of these sounds in, for example, a music sampler. In our context of content reuse within a creative musical context, more than the pitch we are interested in estimating the musical note which is played in a single note recording. For this reason, we evaluate algorithms which are capable of outputting a MIDI note number (and a note name) along with the non-quantized pitch value (see below).

4.1 Datasets

To carry out the pitch estimation evaluation task, five datasets have been compiled which add up to more than 18k instrument notes from different instruments and recording conditions. These are the datasets that we used for the pitch estimation task:

- **Freesound, Carlos Vaquero dataset (CVAQ)**: This dataset contains single-note recordings from different western music instruments containing, among others, acoustic guitar, recorder, bassoon and several bowed instruments (played plucked and bowed with different techniques). This dataset was created in the Music Technology Group of Universitat Pompeu Fabra. All these sounds are hosted in Freesound and released under Creative Commons Licenses.

- **Freesound, Good-Sounds dataset (GSND)**: Similarly to CVAQ, this dataset contains single-note recordings from different western music instruments but only played with their most common playing technique. This dataset was created in the Music Technology Group of Universitat Pompeu Fabra. Again, all these sounds are hosted in Freesound and released under Creative Commons Licenses.

- **University of IOWA (IOWA)**: This is a classic dataset for pitch estimation which includes single notes from piano, cello, trumpet, marimba and xylophone, and use different playing techniques. This dataset has been developed at the Electronic Music Studios of University of Iowa[16].

- **Philharmonia Orchestra Sound Samples (PHIL)**: The Philharmonia Orchestra Sound Samples dataset is composed of single-note recordings of a wide range of orchestral instrument which were recorded and made available under a Creative Commons license by the London Philharmonia Orchestra[17]. For this particular dataset, we discarded recordings belonging to percussion instruments which are typically non-pitched.

- **NSynth test dataset (NSYT)**: NSynth dataset is a large-scale dataset of musical notes released by Google as part of the Magenta project [Engel17]. The dataset is released under a Creative Commons license and can be freely downloaded online[18]. The full version of NSynth contains more than 300k pitched musical notes generated using instrument plugins from commercial sample libraries. In our evaluation, we only use the subset of the dataset labeled as “test”, which is already big enough for our benchmarking purposes.

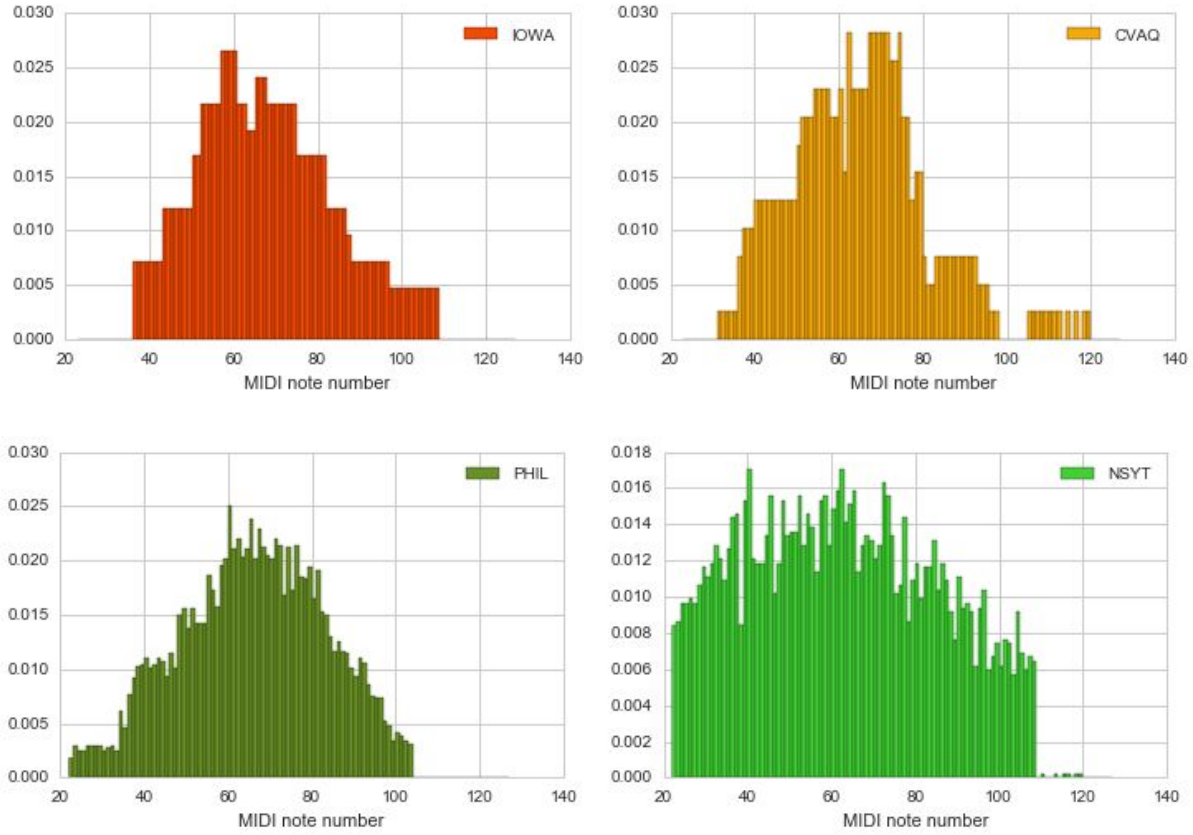
[16] http://theremin.music.uiowa.edu/mis.html
[17] http://www.philharmonia.co.uk/explore/sound_samples
[18] https://magenta.tensorflow.org/datasets/

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 688382

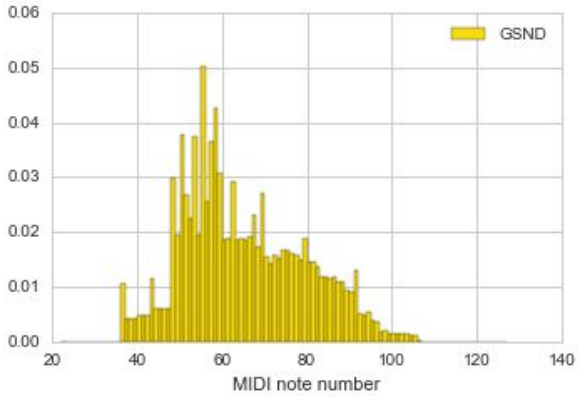
The following table includes some general statistics about the datasets:

Dataset	N. sounds	Total duration	Mean duration	Duration range
IOWA	415	0h 18m	2.68s	0.14s - 7.73s
CVAQ	391	0h 23m	3.62s	1.01s - 8.28s
PHIL	5462	2h 26m	1.61s	0.08s - 45.61s
NSYT	4096	4h 33m	4.0s	4.00s - 4.00s
GSND	8141	14h 24m	6.45s	0.48s - 33.65s

The plots below show a histogram of the MIDI note numbers featured in each of the datasets. We can see how NSYT shows the most uniform distribution. This is due to the fact that sounds in NSYT were generated through an automated process that played all possible notes of every instrument using software plugins and recorded the output. In this way it was possible to easily generate notes for the entire spectrum. The other datasets (which contain similar sets of western classical music instruments) feature a similar histogram. This suggests that NSYT dataset might include notes from instruments played in ranges which would not be played by such instruments very often.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382



4.2 Analysis algorithms

We incorporated the following algorithms in our evaluation of key estimation:

- **ESSPYin**: This is a classic implementation of the well-known YIN pitch estimation algorithm [Cheveigné02]. For this algorithm we use the implementation provided in Essentia[19]. Pitch YIN is based on autocorrelation of the signal and further peak detection to estimate the fundamental frequency. This algorithm outputs an estimated pitch for each frame of the audio. The final selected pitch is taken as the median of all these pitches. For our analysis, the output provided by the algorithm is translated into the (quantised) MIDI note number using the convention that A4 = MIDI 60 = 440hz.

- **QMULPYin:** This is an implementation of a variant of the YIN algorithm, probabilistic YIN [Mauch14], provided as a VAMP plugin from Queen Mary University of London. This algorithm outputs an estimated pitch for each frame of the audio. The final selected pitch is taken as the median of all these pitches. Again, the output provided by the algorithm is translated into the (quantised) MIDI note number using the convention that A4 = MIDI 60 = 440hz.

- **ACExtractorV1**: This is the algorithm that's included in the first version of the Audio Commons extractor. The Audio Commons extractor uses the algorithm pitch YIN FFT for pitch estimation, which is an optimisation proposed by Brossier [Brossier06] for reduced calculation time of the original YIN. This algorithm outputs an estimated pitch for each frame of the audio. Again, we use the implementation provided in Essentia[20], and the median value of the output provided by the algorithm for each frame is translated into the (quantised) MIDI note number using the convention that A4 = MIDI 60 = 440hz.

4.3 Evaluation metrics

To evaluate the effectiveness of the pitch estimation methods we use the following two accuracy metrics:

- **Exact**: This metric only considers an estimation to be correct when it's the same as the annotation in the ground truth. As mentioned above, our analysis algorithms output integer MIDI notes. These are the values that we use for comparison, therefore there is no tolerance

[19] http://essentia.upf.edu/documentation/reference/std_PitchYin.html
[20] http://essentia.upf.edu/documentation/reference/std_PitchYinFFT.html

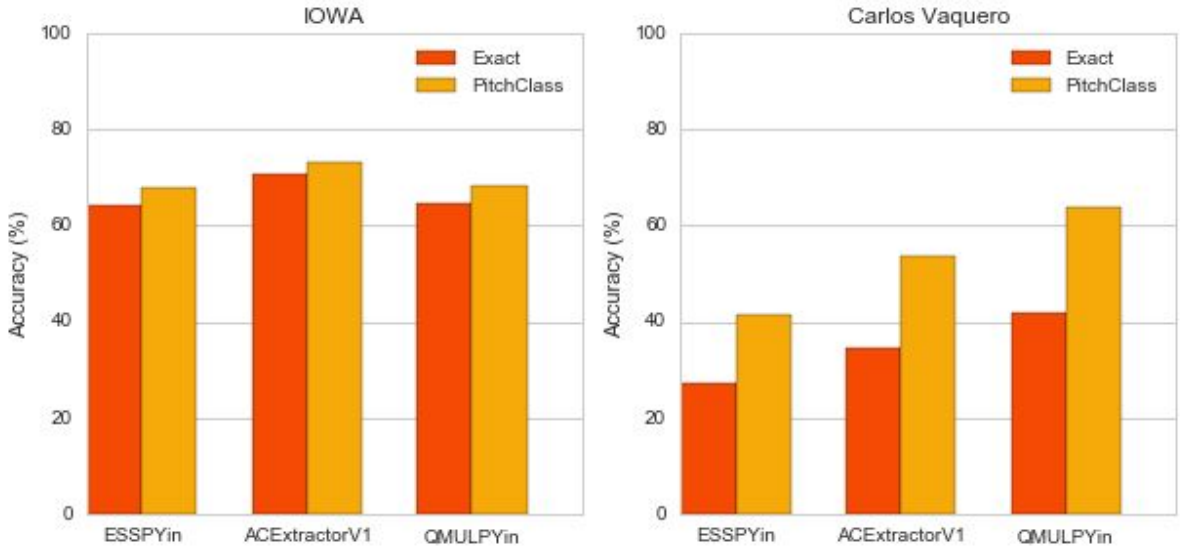
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

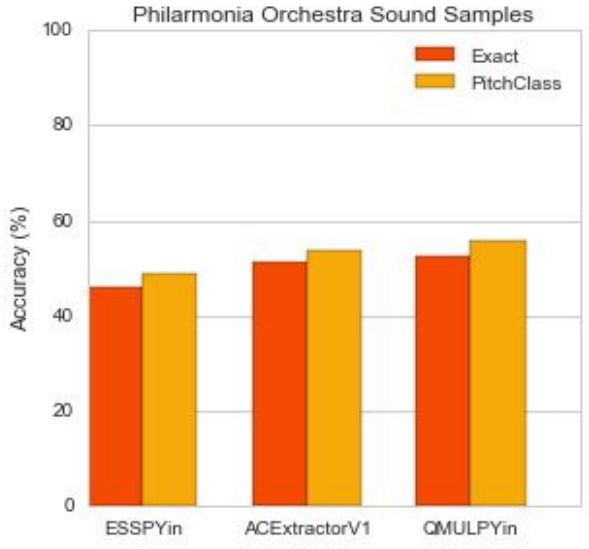
threshold. We assume all algorithm outputs and the dataset ground truth annotation use the convention that A4 = MIDI 60 = 440hz.

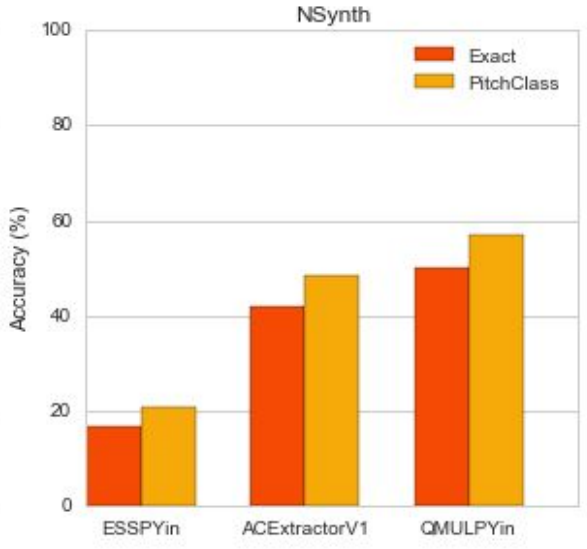
- **PitchClass**: This metric differs from the "Exact" one in that it also considers as correct those estimations which feature the same pitch class as the ground truth. This means that octave errors are not considered wrong in this metric.

4.4 Results

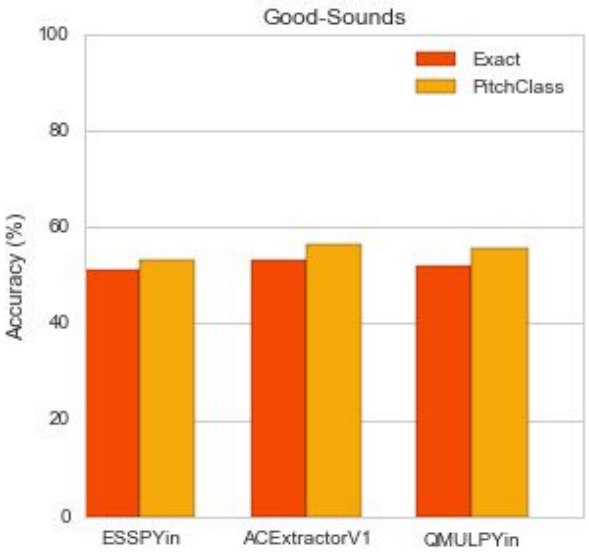
The following figures show the results of each algorithm in each of the analysed datasets.







This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382



The following table shows the average accuracy (in %) when taking into account all datasets at once. Marked in orange are the results of the algorithm included in the Audio Commons tool for the annotation of music samples:

Method	Exact	PitchClass	Mean
QMULPYin	51.86	56.36	54.11
ACExtractorV1	50.09	54.18	52.14
ESSPYin	41.80	44.87	43.34

It can be observed that ACExtractorV1 reports results similar to the best-scoring method, QMULPYin. Nevertheless, it is quite unexpected to observe such big differences among the results obtained using different datasets, and the inconsistencies in terms of which algorithm performs better. Interestingly the classic implementation of the YIN algorithm (ESSPYin) obtained really different results. This leads us to think that for future evaluations of the tool we should review these datasets to make sure that we are not including sounds which might bring significant noise to the evaluation like non-pitched notes or notes played with very unusual playing techniques which would make the pitch ambiguous.

5 Conclusion

In this deliverable we have described the evaluation tasks that we have carried out for the most important estimation algorithms included in the first version of the Audio Commons tool for the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

automatic annotation of music samples. The following table summarises the accuracies obtained for each of the tasks:

Task	Accuracy (for the Audio Commons tool)
Tempo estimation	30% - 67%
Key estimation	23% - 28%
Pitch estimation	50% - 54%

As it can be observed, the accuracies tend to be rather low and this is clearly an issue that should be improved in future versions of the tool. One way to improve such accuracies is by adopting the best scoring algorithms for each of the tasks (according to our evaluation). However, in some cases accuracies are already similar and the gain won't be large. Another way to improve the accuracies is to introduce the notion of *confidence measures*. The idea of a confidence measure is to estimate the probability of the key/tempo/pitch prediction to be correct. If this probability is not high enough, we can then mark this estimation as "unsure" and not consider the content when a user is making a query. By doing this we would be able to effectively increase the accuracy of our estimation task at the expense of reducing the number of audio resources which are made available through user queries. As part of research within the Audio Commons project, we started doing research in this line and publish a paper which deals with confidence measures for a tempo estimation task using music loop [Font16]. In this paper we show how using confidence measures accuracies can be increased by 20% at the expense of 40% of the dataset. We plan on incorporating such measures and study their impact in the other evaluation tasks as well.

Another important issue to mention here is that the current tool for the automatic annotation of music samples does not make any step to guess whether an input audio file does in fact correspond to a music sample or is any other kind of audio. One potential improvement for the next iteration of the tool is to have a method to differentiate between music samples and the other kinds of content, in this way we know when to apply this classifier. This is strongly related with the confidence measure ideas, as it would allow us to better identify for which kinds of content the estimation of our algorithms is relevant and for which kinds of content it is not.

In terms of descriptors, the current analysis tool does not include all the ones that have been proposed in deliverable D4.2 First prototype tool for the automatic semantic description of music samples. Future versions of the tool will incorporate new descriptors which will require new evaluation tasks. In particular, new versions of the tool will potentially include chord recognition and instrument identification algorithms. Those evaluation tasks will therefore be included in the corresponding evaluation report of the upgraded version of the tool. We are also planning on adding an evaluation task for the loudness algorithm (which is already included in the current version of the tool). As described in D4.2, the loudness algorithm we described was designed for music tracks and it would be desirable to assess its validity for the case of music samples. We leave this to future work.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

6 References

[Böck15] Böck, S., Krebs, F., & Widmer, G. (2015). Accurate Tempo Estimation Based on Recurrent Neural Networks and Resonating Comb Filters. In ISMIR (pp. 625-631).

[Brossier06] Brossier, P. M. (2006). Automatic annotation of musical audio for interactive applications (Doctoral dissertation).

[Cheveigné02] De Cheveigné, A., & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, 111(4), 1917-1930.

[Degara12] Degara, N., Rúa, E. A., Pena, A., Torres-Guijarro, S., Davies, M. E., & Plumbley, M. D. (2012). Reliability-informed beat tracking of musical signals. IEEE Transactions on Audio, Speech, and Language Processing, 20(1), 290-301.

[Engel17] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., & Norouzi, M. (2017). Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. arXiv preprint arXiv:1704.01279.

[Faraldo17] Faraldo, Á., Jordà, S., & Herrera, P. (2017, June). A Multi-Profile Method for Key Estimation in EDM. In Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio. Audio Engineering Society.

[Font16] Font, F., & Serra, X. (2016). Tempo Estimation for Music Loops and a Simple Confidence Measure. In ISMIR (pp. 269-275).

[Gómez06] Gómez, E. (2006). Tonal description of polyphonic audio for music content processing. INFORMS Journal on Computing, 18(3), 294-304.

[Gouyon06] Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., & Cano, P. (2006). An experimental comparison of audio tempo induction algorithms. IEEE Transactions on Audio, Speech, and Language Processing, 14(5), 1832-1844.

[Klapuri06] Klapuri, A. P., Eronen, A. J., & Astola, J. T. (2006). Analysis of the meter of acoustic musical signals. IEEE Transactions on Audio, Speech, and Language Processing, 14(1), 342-355.

[Knees15] Knees, P., Faraldo, A., Herrera, P., Vogl, R., Böck, S., Hörschläger, F., & Le Goff, M. (2015, October). Two Data Sets for Tempo Estimation and Key Detection in Electronic Dance Music Annotated from User Corrections. In ISMIR (pp. 364-370).

[Mauch14] Mauch, M., & Dixon, S. (2014, May). pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on (pp. 659-663). IEEE.

[Noland07] Noland, K., & Sandler, M. (2007, May). Signal processing parameters for tonality estimation. In Audio Engineering Society Convention 122. Audio Engineering Society.

[Percival14] Percival, G., & Tzanetakis, G. (2014). Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 22(12), 1765-1776.

[Raffel14] Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., & Ellis, D. P. W. (2014). mir_eval: A Transparent Implementation of Common MIR Metrics. In ISMIR.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382

[Temperley99] Temperley, D. (1999). What's Key for Key? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered. Music Perception: An Interdisciplinary Journal, 17(1), 65-100.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382