



Deliverable D4.12

Release of the 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	31st January 2019 (M36)
Submission date	31st January 2019 (M36)
Report availability	Public (X), Confidential ()
Deliverable type	Report (), Demonstrator (), Other (X)
Task leader	MTG-UPF
Authors	Frederic Font
Document status	Draft (), Final (X)





Table of contents

Table of contents	2
Executive Summary	3
1 Description of the annotation tool	4
1.1 Goals	4
1.2 Implementation	6
1.3 Usage instructions	7
1.4 Improvements of the final release	9
1.4.1 Improvements in “single eventness” detection	9
1.4.2 Improvements in pitch estimation	10
1.4.3 Improvements in key estimation	13
1.5 Web app demonstrator	14
2 Conclusion	15





Executive Summary

As part of the Audio Commons Ecosystem, a number of tools have been developed 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. Two prototypes of each of these tools have been released and evaluated during the course of the project.

The current deliverable represents the final release of the tool for the automatic extraction of musical audio properties for music samples. The tool consists of an Essentia audio extractor which extracts a number of audio properties and a Python script which post-processes some of these properties and renames them to the commonly agreed descriptor names of the Audio Commons Ontology. The current version of the tool also integrates the final version of the timbral descriptors developed in Work Package 5 and a linked data compatible output format (JSON-LD). The whole ensemble is provided as a Docker container which makes it very easy to be installed and executed on any platform.

The version of the tool demonstrated in this deliverable includes significant improvements with respect to the previous version demonstrated in deliverable D4.7 and evaluated in deliverable D4.10. These include accuracy improvements for pitch, key, and single event-ness estimation and a number of processing fixes to prevent the tool from failing to analyze certain sounds. Complementarity to the improvements in the annotation tool, as part of this deliverable we also released a simple web application that demonstrates the usefulness and accuracy of some of the descriptors for music samples.





1 Description of the annotation tool

1.1 Goals

The Audio Commons Ecosystem provides tools for the automatic annotation of different kinds of audio content. These tools are developed and evaluated in in different tasks across work packages 4 and 5. In particular, Audio Commons focuses on:

- Musical properties for music samples
- Musical properties for music pieces
- Non-musical audio properties (timbral models)

The differentiation between music samples and music pieces has been done to tailor some descriptors to the particular context in which they're used. We define music pieces as audio recordings typically corresponding to *complete* songs, while we define music samples as individual or simpler music elements such as single notes, percussive hits, chords, melodies or loops. Note that loops (or short music fragments) can also be classified under music pieces in case they're complex enough.

As part of the development of the Audio Commons Ontology, a sound schema has been developed in which a number of musical and non-musical properties are specified. The following table shows the musical properties that are considered for both music pieces and music samples. Marked in green are the descriptors for music samples that have been added to the tool demonstrated in this deliverable, also including the non-musical audio properties (also relevant for music samples) from Work Package 5. Highlighted in red are the descriptors that were initially planned to be added to the extractor but that have been excluded because are not relevant enough for music samples (`ac:bitdepth`, `ac:attack`, `ac:decay`, `ac:sustain`, `ac:release`, `ac:chord`, `ac:mood`, `ac:genre`), can be easily guessed from the file name (`ac:format`), or because they have already been implemented in the music analysis tools (D4.3, D4.8, D4.13).

Name	Description	Relevant for...
<code>ac:duration</code>	Duration of audio	Music pieces + Music samples
<code>ac:format</code>	Audio format (e.g. wav, mp3)	Music pieces + Music samples
<code>ac:lossless</code>	Whether audio file is in lossless codec (1 or 0)	Music pieces + Music samples
<code>ac:codec</code>	Codec used for encoding the audio (e.g. pcm_s16le)	Music pieces + Music samples
<code>ac:filesize</code>	Size of the file	Music pieces + Music samples
<code>ac:bitrate</code>	Number of bits per second	Music pieces + Music samples
<code>ac:bitdepth</code>	Number of bits per sample	Music pieces + Music samples





ac:samplerate	Number of samples per second	Music pieces + Music samples
ac:channels	Number of channels	Music pieces + Music samples
ac:audio_md5	MD5 checksum of raw undecoded audio payload. It can be used as a unique identifier of audio content.	Music pieces + Music samples
ac:genres	Music genre(s) of the musical content	Music pieces
ac:instruments	Musical instruments present in the recording	Music pieces + Music samples
ac:mood	Mood conveyed by the musical content	Music pieces
ac:tonality	Tonic and scale (e.g. A minor)	Music pieces + Music samples
ac:tonality_confidence	Confidence of the tonality estimation	Music pieces + Music samples
ac:tempo	Tempo in BPM of the audio signal	Music pieces + Music samples
ac:tempo_confidence	Confidence of the tempo estimation	Music pieces + Music samples
ac:loop	Whether audio file is <i>loopable</i> (Boolean)	Music samples
ac:chord	Played chord (e.g. G#m)	Music samples
ac:note_name	Played note name (e.g. C4)	Music samples
ac:note_midi	Played note midi number (e.g. 60)	Music samples
ac:note_frequency	Played note frequency (e.g. 440Hz)	Music samples
ac:note_confidence	Confidence of the note estimation	Music samples
ac:loudness	Loudness value	Music pieces + Music samples
ac:dynamic_range	Dynamic range of audio recording	Music pieces + Music samples
ac:temporal_centroid	Temporal centroid	Music samples
ac:attack	Attack length	Music samples
ac:decay	Decay length	Music samples
ac:sustain	Sustain amount	Music samples
ac:release	Release length	Music samples





ac:log_attack_time	Logarithm of the time it takes to reach maximum amplitude of audio signal (good for perceptual attack)	Music samples
ac:single_event	Whether the audio file contains one single <i>audio event</i> or more than one (Boolean). This computation is based on the loudness of the signal and does not do any frequency analysis.	Music samples
ac:brightness	Brightness of the analyzed audio in a scale from [0-100]. A <i>bright</i> sound is one that is clear/vibrant and/or contains significant high-pitched elements.	Music samples
ac:hardness	Hardness of the analyzed audio in a scale from [0-100]. A <i>hard</i> sound is one that conveys the sense of having been made (i) by something solid, firm or rigid; or (ii) with a great deal of force.	Music samples
ac:depth	Depth of the analyzed audio in a scale from [0-100]. A <i>deep</i> sound is one that conveys the sense of having been made far down below the surface of its source.	Music samples
ac:roughness	Roughness of the analyzed audio in an undetermined scale. A <i>rough</i> sound is one that has an uneven or irregular sonic texture.	Music samples

1.2 Implementation

The annotation tool in particular can be found in the following public Github repository: <https://github.com/AudioCommons/ac-audio-extractor>. It consists of a “dockerized”¹ Python script which uses the Essentia Python bindings to compute audio descriptors as well as computes some other audio descriptors (the timbral models) in pure Python code. Then it reformats the output to match the property names defined in the Audio Commons sound schema and to generate both standard JSON and JSON-LD (linked data compatible) output analysis files. Essentia (<http://essentia.upf.edu>) is an audio analysis library developed and maintained at the Music Technology Group in Universitat Pompeu Fabra, coordinators of the Audio Commons Project. More information about the Essentia framework is provided in deliverable [D4.3 First prototype tool for the automatic semantic description of music pieces](#).

The tool provides a command line utility to analyze a single audio file and write the output analysis into a JSON file (see next section for instructions). It does not manage the analysis of collections of audio files. This is expected to be done by another script that calls the command line utility and is run in the host machine. That other script is out of the scope of this deliverable.

¹ Docker (<http://docker.com>) is a multi-platform software that allows the deployment of software into containers. Containers work like lightweight virtual machines which include all the dependencies needed for the software to run. Therefore, “dockerized” programs are easy to run in different platforms as the Docker layer handles dependencies and other configuration parameters.





1.3 Usage instructions

In order to use the analysis tool, Docker will need to be installed in the host system. If docker is installed the steps are quite simple:

1. Clone repository <https://github.com/AudioCommons/ac-audio-extractor>.
2. cd to repository folder and run `docker build -t mtgupf/ac-audio-extractor:v3 .`
3. Place the audio files that need to be analyzed in the same repository folder and run `docker run -it --rm -v `pwd`:/essentia mtgupf/ac-audio-extractor:v3 -i filename.wav -o analysis_output.json`

This will analyze the file `filename.wav` and place the results in `analysis_output.json`. In order to analyze files from other directories, one can either create a symbolic link in the same directory as the repository, or mount a different directory in the Docker container by replacing ``pwd`` with the path to that directory.

Running the above commands will produce an output like the following:

```
{
  "duration": 9.241541862487793,
  "lossless": 1.0,
  "codec": "pcm_s16le",
  "bitrate": 705600.0,
  "samplerate": 44100.0,
  "channels": 1.0,
  "audio_md5": "2722ac23a142ce727e0642b0a63c7347",
  "loudness": -28.64586639404297,
  "dynamic_range": 3.432065963745117,
  "temporal_centroid": 0.5782503485679626,
  "log_attack_time": 0.6950863599777222,
  "filesize": 815294,
  "single_event": false,
  "tonality": "G# major",
  "tonality_confidence": 0.5119080543518066,
  "loop": false,
  "tempo": 84,
  "tempo_confidence": 0.42026047706604003,
  "note_midi": 74,
  "note_name": "D5",
  "note_frequency": 608.390625,
  "note_confidence": 0.0,
  "brightness": 60.313207479409286,
  "depth": 16.728879931862544,
  "hardness": 82.90738501480826,
  "roughness": 6.646583836789146
}
```

Alternatively, using the JSON-LD output format it would look like:

```
{
  "@context": {
    "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
    "ac": "https://w3id.org/ac-ontology/aco#",
  }
}
```





```

    "afo": "https://w3id.org/afo/onto/1.1#",
    "afv": "https://w3id.org/afo/vocab/1.1#",
    "ebucore": "http://www.ebu.ch/metadata/ontologies/ebucore/ebucore#",
    "nfo": "http://www.semanticdesktop.org/ontologies/2007/03/22/nfo#"
  },
  "@type": "ac:AudioFile",
  "ebucore:bitrate": 705600.0,
  "ebucore:filesize": 529278,
  "ebucore:hasCodec": {
    "@type": "ebucore:AudioCodec",
    "ebucore:codecId": "pcm_s16le"
  },
  "nfo:compressionType": "nfo:losslessCompressionType",
  "ac:audioMd5": "8da67c9c2acbd13998c9002aa0f60466",
  "ac:availableItemOf": {
    "@type": "ac:AudioClip"
  },
  "ac:signalAudioFeature": [
    {
      "@type": "afv:Loop",
      "afo:value": true
    },
    {
      "@type": "afv:Tempo",
      "afo:confidence": 1.0,
      "afo:value": 120
    },
    {
      "@type": "afv:Key",
      "afo:confidence": 0.2868785858154297,
      "afo:value": "G# minor"
    },
    {
      "@type": "afv:TemporalCentroid",
      "afo:value": 0.5078766345977783
    },
    {
      "@type": "afv:MIDINote",
      "afo:confidence": 0.0,
      "afo:value": 74
    },
    {
      "@type": "afv:Pitch",
      "afo:confidence": 0.0,
      "afo:value": 592.681884765625
    },
    {
      "@type": "afv:Loudness",
      "afo:value": -28.207069396972656
    },
    {
      "@type": "afv>Note",
      "afo:confidence": 0.0,
      "afo:value": "D5"
    }
  ],

```





```

    {
      "@type": "afv:LogAttackTime",
      "afo:value": 0.30115795135498047
    }
  ],
  "ac:signalChannels": 1,
  "ac:signalDuration": 6.0,
  "ac:singalSamplerate": 44100.0
}

```

A blog post in the Audio Commons website describes the tool and provides instructions on how to used. It can be accessed here:

<https://www.audiocommons.org/2018/07/15/audio-commons-audio-extractor.html>

1.4 Improvements of the final release

The present release of the tool (ACExtractorV3) incorporates a number of improvements with respect to the previous version (ACExtractorV2). Because there is no other deliverable that will cover evaluation of this release of the tool, we summarize here the latest improvements and updated evaluation results. These subsections here assume that the reader is familiar with the contents, structure and methodology of deliverable D4.10.

1.4.1 Improvements in "single eventness" detection

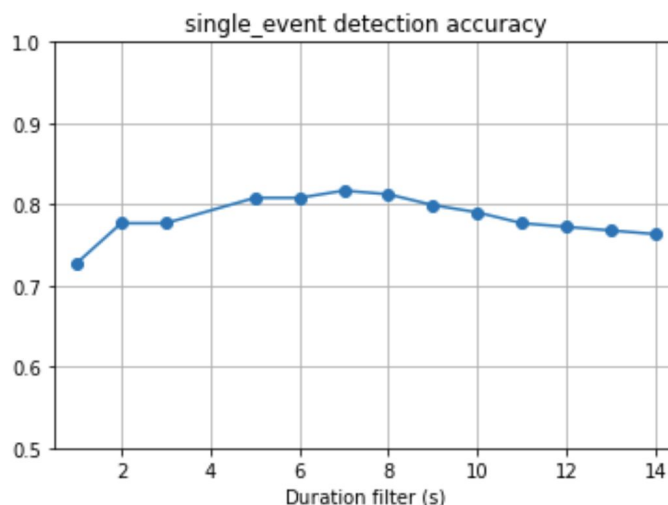
We improved the performance of the single event detector by optimizing the default parameters (optimized to the ground truth we collected in D4.10) and by adding an extra heuristic to determine whether an audio signal should be considered as being a single event or not. The change for the optimized parameters can be seen in [this code commit](#). It adjusts the minimum energy that a region needs to have to be considered a region, the silence threshold and the minimum separation in milliseconds from one region to the next (see deliverable D4.10). The extra heuristic we added is based on sound duration. Sound longer than a threshold are simply considered not to be single events. The table below shows updated evaluation results for the current version of the single eventness descriptor:

	ACExtractorV2 (Deliverable 4.10)	ACExtractorV2 + parameter optimization	ACExtractorV2 + parameter optimization + duration heuristic (ACExtractorV3)
True Positives	0.246	0.241	0.219
True Negatives	0.397	0.446	0.598
False Positives	0.268	0.219	0.067
False Negatives	0.089	0.094	0.116
Accuracy	0.643	0.686	0.817





The figure below shows the accuracy values obtained for different duration filters:



The experimental results show that optimum filter duration is found at 7 seconds. As seen in the table above, using this filter drastically reduces the number of false positives and increases the number of true negatives, which ends up with a higher overall accuracy. Overall, in ACExtractorV3 the single event descriptor shows an accuracy increase of 17 percentage points with respect to the version included in ACExtractorV2.

1.4.2 Improvements in pitch estimation

In the evaluation carried out in D4.10 we observed that pitch estimation was not performing as good as expected, and pitch confidence measures seemed to be behaving strangely. After careful investigation of the algorithms we realized that a bug in the code was causing pitch frequency values returned by the algorithms to be, in some cases, wrongly translated to MIDI note numbers. This was causing lower accuracy numbers and unpredictable behaviour in confidence measure. After fixing this error we re-ran the analysis now obtaining much better results. The table below shows the updated general accuracy values:

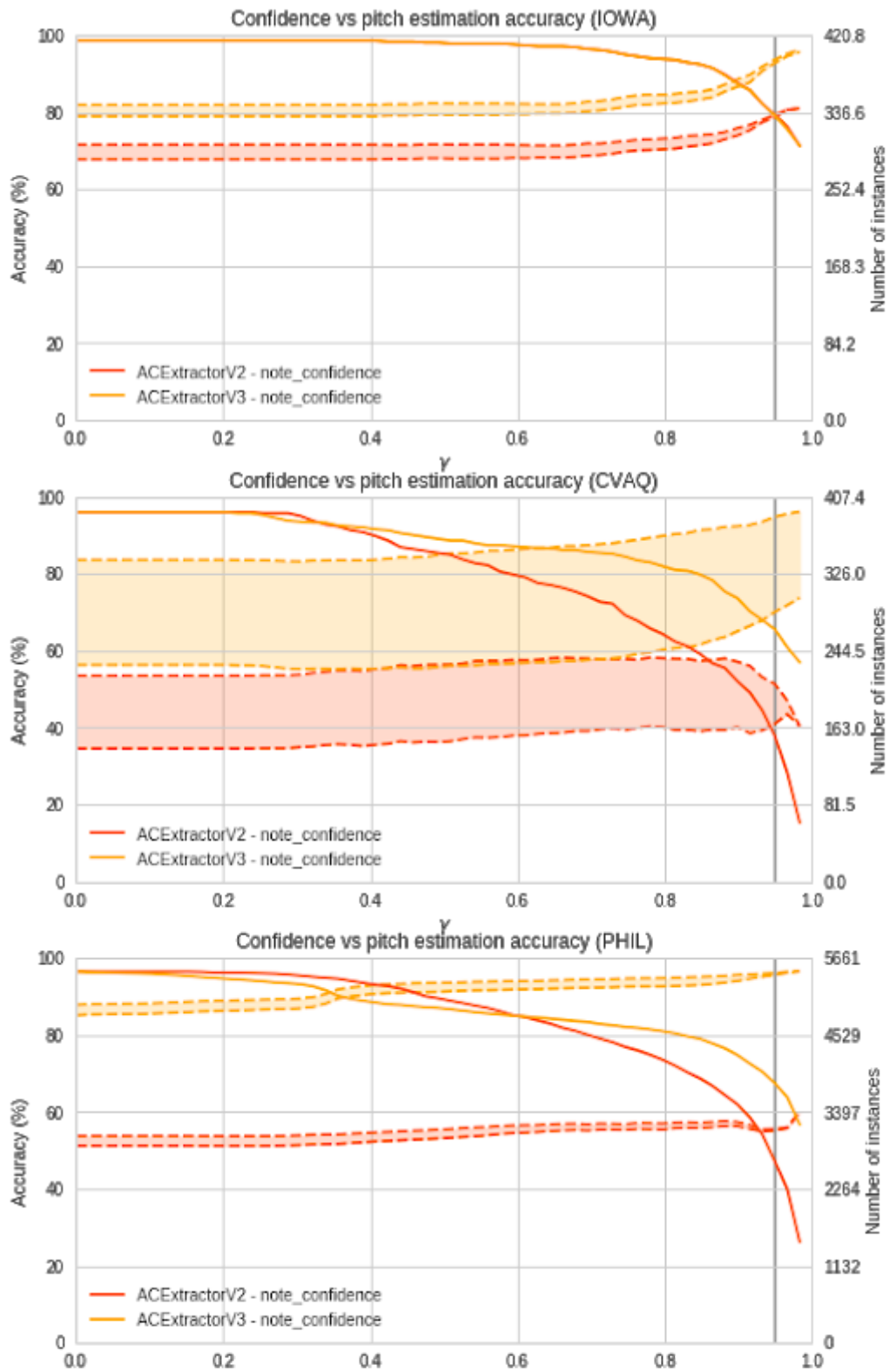
Method	Exact	PitchClass	Mean
QMULPYin	67.70	77.35	72.52
ACExtractorV3	67.60	76.08	71.84
ESSPYin	52.51	61.29	56.90
ACExtractorV2	50.24	57.04	53.64
ESSPYinFFT	46.15	53.04	49.59

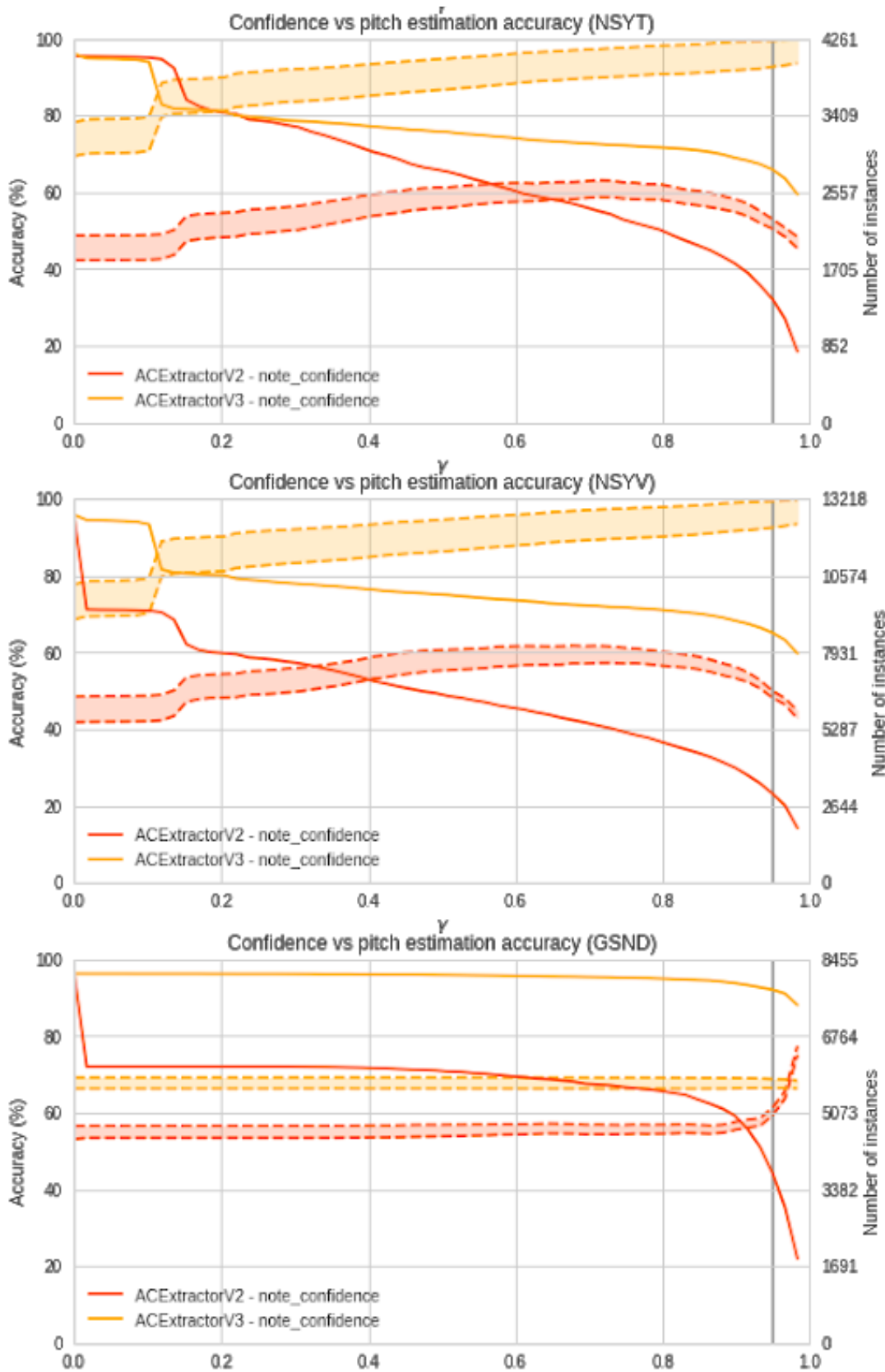




As it can be observed, ACEXtractor3 performs much better than previous version (with an increase of 18 percentage points) and at the state of the art level. The figures below show updated accuracy vs confidence threshold per dataset along with the values obtained in previous evaluation. Remember that the lower bound of the intervals drawn correspond to the accuracy computed with the most strict accuracy measure (exact match) while the upper bound shows the less strict accuracy measure (pitch class).









Now we can see how, as would be expected, accuracy values for ACExtractorV3 increase as the confidence threshold gets higher. Therefore, pitch estimation confidence values are now correctly filtering estimations. The table below shows accuracy ranges for a confidence threshold of 95%.

	IOWA	CVAQ	PHIL	NSYT	NSYV	GSND
V2	A=86.36 - 86.36% N=48%	A=40.52 - 50.98% N=39%	A=55.03 - 55.44% N=49%	A=50.66 - 53.16% N=33%	A=50.26 - 52.04% N=24%	A=59.52 - 61.59% N=46%
V3	A=92.79 - 93.69% N=80%	A=70.45 - 95.08% N=68%	A=95.67 - 96.14% N=70%	A=92.80 - 99.54% N=69%	A=92.60 - 99.48% N=68%	A=66.37 - 68.76% N=96%

Note the significant improvement of ACExtractorV3 compared to ACExtractorV2 both in terms of accuracy (A) and the number of remaining items in the dataset (N).

1.4.3 Improvements in key estimation

For ACExtractorV3 tool we also updated the implementation of the key estimation algorithm which resulted in an increase of key estimation accuracy, making it closer to the results obtained with EDMKey# algorithms (see D4.10). The table below shows the updated accuracy results after re-running the evaluation and including ACExtractorV3.

Method	Same	Fifth	Relative	Parallel	Mirex
EDMKey1	47.13	7.33	8.56	13.41	56.05
ACExtractorV3	47.40	6.22	11.35	10.17	55.94
EDMKey2	42.88	8.69	9.36	14.58	52.95
ACExtractorV2	40.17	5.42	9.55	8.31	47.41
EDMKey3	39.02	4.97	13.10	9.36	47.31
QMULKey	28.67	4.68	8.51	7.08	34.98


As it can be observed ACExtractorV3 shows an increase of 9 percentage points with respect to ACExtractorV2, and performs almost exactly the same as EDMKey1 (as expected).





1.5 Web app demonstrator

In addition to the improvements in the annotation tools, for this deliverable we also set up a very simple web app that allows filtering of Freesound content using Audio Commons descriptors for music samples. Playing with the web app can give an idea about the accuracy and usefulness of the AC descriptors for music samples. The web app can be accessed here: http://www.audiocommons.org/ac-audio-extractor/web_demonstrator/.



AUDIO COMMONS AUDIO EXTRACTOR DEMONSTRATOR

This website allows you to search in **Freesound** and filter using the Audio Commons descriptors for **music samples** developed during the Audio Commons project. The Audio Commons Audio Extractor is available in this [source code repository](#).

Query

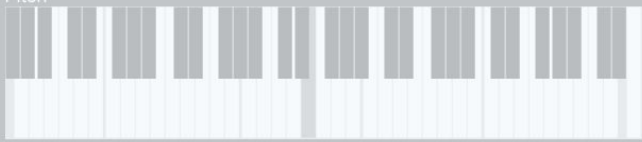
Single event

Loop

Tonality: A major

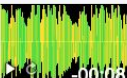
Tempo: 120

Pitch



Your query returned 13 results! These are the first ones:

[GADLY_Guitar.mp3](#) by [GrowingUp](#)

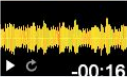


```

{
  "ac_tempo_confidence": 0.9723355770111084,
  "ac_tonality_confidence": 0.9340853691101074,
  "ac_loop": true,
  "ac_note_confidence": 0.5237458944320679,
  "ac_log_attack_time": 0.3153228759765625,
  "ac_dynamic_range": 0.943939208984375,
  "ac_note_frequency": 162.92864990234375,
  "ac_note_name": "D#3",
  "ac_tonality": "A major",
  "ac_loudness": -11.391944885253906,
  "ac_tempo": 120,
  "ac_note_midi": 51,
  "ac_temporal_centroid": 0.4959513247013092,
  "ac_single_event": false
}

```

[FAWN_Guitar.mp3](#) by [GrowingUp](#)





2 Conclusion

In this deliverable we described the final version of the automatic tool for annotating musical properties of music samples. We present the tool as a command line utility that can be given an input audio file and extract relevant musical (and non-musical) properties. The version of the tool presented in this deliverable incorporates significant improvements with respect to the previous version presented in D4.7. Also as part of this deliverable we developed an online tool that demonstrates the usefulness and accuracy of the extracted audio properties by allowing to search content in Freesound and filter results by music properties.

With the final updates in the algorithms for extracting music properties the annotation tool has reached a satisfactory level of accuracy, specially when content is filtered using confidence measures. In this way the tool fulfils the expectations set at the beginning of the Audio Commons project.

Future potential improvements to this tool would be the addition of new descriptors, the improvement of the performance of existing ones, and the integration with music descriptors released in D4.13. Because the music pieces descriptor tool has been designed as an online service rather than a stand-alone tool, it's integration with the tool presented here would require further development. Overall, the tool for the annotation of music samples developed during task T4.2 generates useful metadata that would allow an unannotated collection of music samples to be automatically annotated and made available in the Audio Commons Ecosystem.

