AES Show Spring 2021 • 150th Audio Engineering Convention • May 25 - 28, 2021

pyloudnorm

A simple yet flexible loudness meter in Python

Christian J. Steinmetz   Joshua D. Reiss

Centre for Digital Music, Queen Mary University of London
UKRI Centre for Doctoral Training in Artificial Intelligence and Music

Queen Mary
University of London

Easy installation with pip...

```
pip install pyloudnorm
```

and measure loudness in just a few lines of code

```python
import soundfile as sf
import pyloudnorm as pyln

data, rate = sf.read("test.wav") # load audio (with shape (samples, channels))
meter = pyln.Meter(rate) # create BS.1770 meter
loudness = meter.integrated_loudness(data) # measure loudness
```

Outline

- Loudness
- ITU-R BS.1770
- Modifications
- **pyloudnorm**
- Evaluation

What is **loudness**?

Humans perceive sound pressure on a nonlinear scale with respect to frequency and intensity.

Oftentimes we want to compare the relative loudness of two stimuli, but this can be challenging.

What is **loudness**?

Humans perceive sound pressure on a nonlinear scale with respect to frequency and intensity.

Oftentimes we want to compare the relative loudness of two stimuli, but this can be challenging.

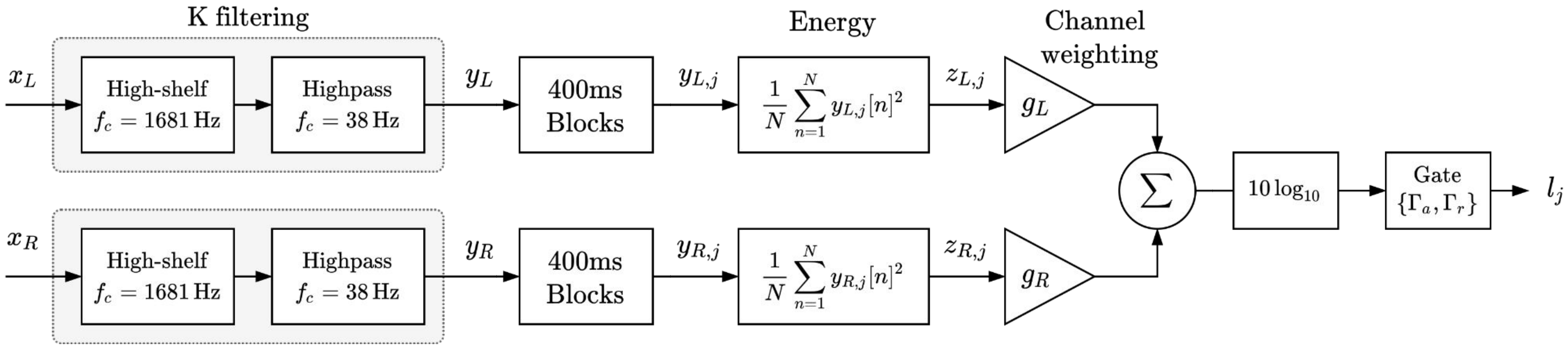
Models of loudness

There has been significant research on subjective loudness in psychoacoustics.
(Stevens, 1956; Zwicker & Scharf, 1965; Moore & Glasberg, 1996; Moore, 2014)

There has also been interest in methods for measuring the loudness of music such as Vickers' loudness and ReplayGain.
(Vickers, 2001) (Robinson, 2002)

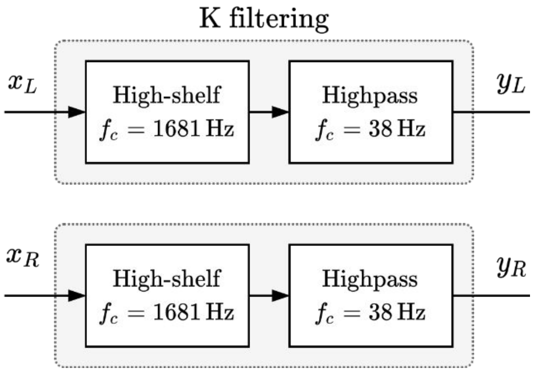
ITU-R BS.1770 recommendation attempts to standardize these methods with a simple algorithm, and has been adopted in EBU R 128, which dictates loudness for broadcast.

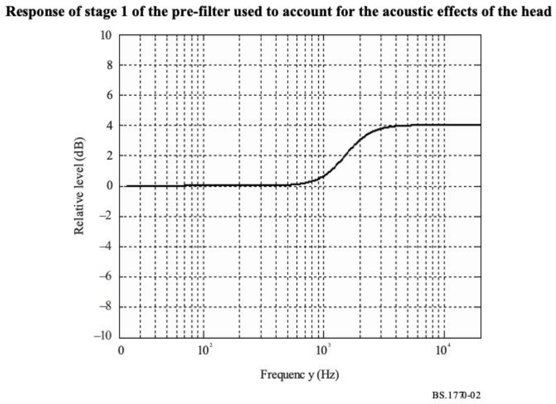
ITU-R BS. 1770
Integrated loudness

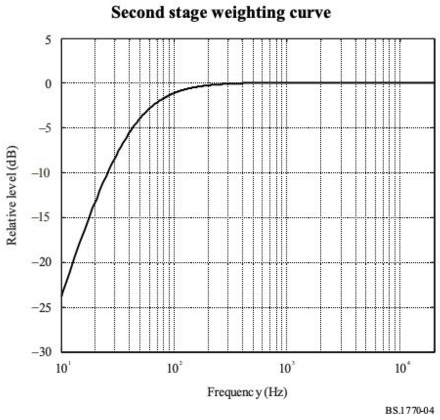


A simple algorithm for measuring loudness of electronically reproduced sounds (recordings, live broadcasts, etc.)

ITU-R BS. 1770
Integrated loudness

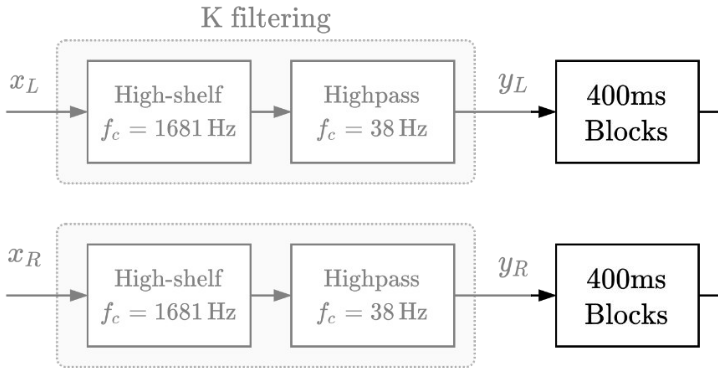






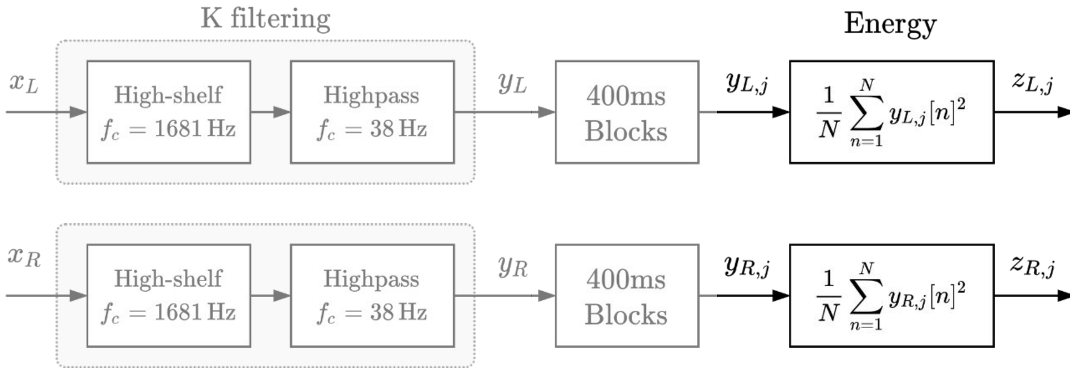
Two stage filtering processing to simulate human sensitivities.

ITU-R BS. 1770
Integrated loudness



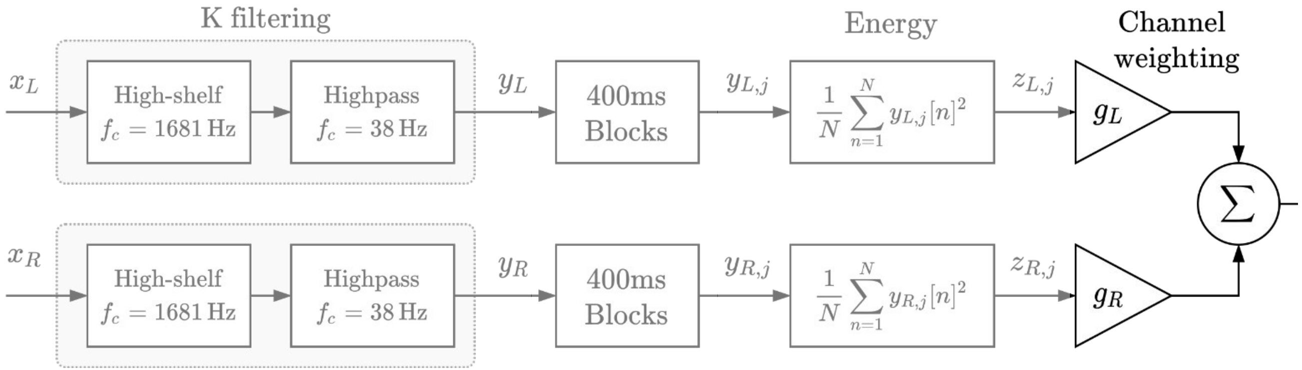
Split the filtered signal into overlapping blocks of 400ms.

ITU-R BS. 1770
Integrated loudness



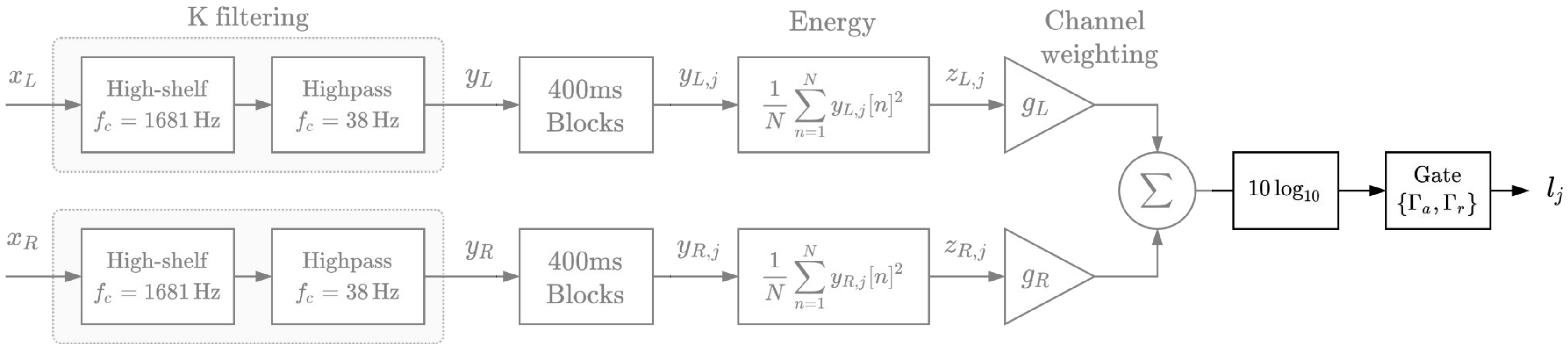
Measure the energy in each block.

ITU-R BS. 1770
Integrated loudness



Weight the channels appropriately. (Relevant for multichannel)

ITU-R BS. 1770
Integrated loudness



Apply log scaling and use a gate to remove blocks below the thresholds.

ITU-R BS. 1770
Integrated loudness

$$L_{KG} = -0.691 + 10\log_{10} \sum_i g_i \left(\frac{1}{|J_g|} \sum_{J_g} z_{i,j} \right),$$

this time where $J_g = \{j : l_j > \Gamma_a \text{ and } l_j > \Gamma_r\}$.

Compute integrated loudness by summing the energy of all blocks above the two thresholds.
(Refer the the original rec. for more details)

Applications

While designed for broadcast, there are now many other applications.

- Automatically normalizing stimuli for listening tests
  (Olive et al., 2013) (Jillings et al., 2015)
- Automatic loudness-based multitrack mixing
  (Ward et al., 2012) (Mansbridge et al., 2012) (Ward & Reiss, 2016) (Fenton, 2018)
- Pre-processing audio datasets in machine learning
  (Abdelnour et al., 2018) (Fischer et al., 2020) (Cosentino et al., 2020)
- Feature extraction and data augmentation
  (Lenain et al., 2020) (Salamon et al., 2017)

Modifications

It should be noted that while this algorithm has been shown to be effective for use on audio programmes that are typical of broadcast content, the algorithm is not, in general, suitable for use to estimate the subjective loudness of pure tones. ~ ITU-R BS. 1770

The recommendation makes clear that loudness measurements correlate well with perception only when the signal being measured is broadband in nature.

Modifications (cont.)

Cabrera et al., 2008

Cutoff frequency of the highpass filter to 149 Hz,
and the replacement of the high-shelf filter by a notch filter centered at 1 kHz

Pestana et al., 2013

Smaller gating block size of 280 ms and +10 dB gain on the high-shelf filter,
which was better optimized for measuring loudness of multitrack instrument sources

Fenton & Lee, 2017

Boosting the gain of the high-shelf filter by +5 dB and changing the cutoff of the highpass filter to 130 Hz, as well as a peaking filter with a center frequency of 500 Hz

De Man, 2018

The original recommendation only provided filter coefficients at 48 kHz, so they reverse-engineered the filter specification from the original recommendation.

pyloudnorm

Simple to install and run, but flexible to enable modifications.

```
import soundfile as sf
import pyloudnorm as pyln

data, rate = sf.read("test.wav") # load audio (with shape (samples, channels))
meter = pyln.Meter(rate) # create BS.1770 meter
loudness = meter.integrated_loudness(data) # measure loudness
```

Pyloudnorm

Utilizing proposed modifications

```
# block size
meter1 = pyln.Meter(rate)                          # 400ms block size
meter2 = pyln.Meter(rate, block_size=0.200)        # 200ms block size
```

```
# filter classes
meter3 = pyln.Meter(rate)                          # BS.1770 meter
meter4 = pyln.Meter(rate, filter_class="DeMan")    # fully compliant filters
meter5 = pyln.Meter(rate, filter_class="Fenton/Lee 1") # low complexity improvement by Fenton and Lee
meter6 = pyln.Meter(rate, filter_class="Fenton/Lee 2") # higher complexity improvement by Fenton and Lee
meter7 = pyln.Meter(rate, filter_class="Dash et al.")  # early modification option
```

Pyloudnorm

Enable future modifications

```
# create your own IIR filters
my_high_pass  = IIRfilter(0.0, 0.5, 20.0, rate, 'high_pass')
my_high_shelf = IIRfilter(2.0, 0.7, 1525.0, rate, 'high_shelf')

# create a meter initialized without filters
meter8 = pyln.Meter(rate, filter_class="custom")

# load your filters into the meter
meter8._filters = {'my_high_pass' : my_high_pass, 'my_high_shelf' : my_high_shelf}
```

Evaluation

Is **pyloudnorm** compliant
and how does it compare to
other loudness implementations?

Loudness implementations

Essentia (Bogdanov et al., 2013)	https://essentia.upf.edu
ffmepg	https://ffmpeg.org
libebur128	https://github.com/jiixyj/libebur128
loudness.py (De Man, 2018)	https://github.com/BrechtDeMan/loudness.py
Adobe Audition	https://www.adobe.com/products/audition
youlean	https://youlean.co/file-loudness-meter

Provided compliance material

File	Target	pyloudnorm Default	pyloudnorm De Man	loudness.py	ffmpeg	libebur128	Essentia	Audition	youlean
FrequencySweep	-18.0	-18.03	-17.99	-17.99	-18.00	-18.00	-18.18	-18.03	-18.02
25Hz_2ch	-23.0	-23.00	-22.99	-22.99	-23.10	-23.00	**-26.37**	-23.04	-23.02
100Hz_2ch	-23.0	-23.03	-22.99	-22.99	-23.10	-23.00	**-22.86**	-23.04	-23.02
500Hz_2ch	-23.0	-23.04	-22.99	-22.99	-23.10	-23.00	-22.99	-23.04	-23.02
1000Hz_2ch	-23.0	-23.03	-22.99	-22.99	-23.10	-23.00	-23.00	-23.04	-23.02
2000Hz_2ch	-23.0	-23.03	-22.99	-22.99	-23.10	-23.00	-23.00	-23.04	-23.02
10000Hz_2ch	-23.0	-23.04	-22.99	-22.99	-23.10	-23.00	-23.00	-23.04	-23.02
25Hz_2ch	-24.0	-24.00	-23.99	-23.99	-24.10	-24.00	**-27.21**	-24.04	-24.02
100Hz_2ch	-24.0	-24.03	-23.99	-23.99	-24.10	-24.00	-23.92	-24.04	-24.02
500Hz_2ch	-24.0	-24.04	-23.99	-23.99	-24.10	-24.00	-23.99	-24.04	-24.02
1000Hz_2ch	-24.0	-24.04	-23.99	-23.99	-24.10	-24.00	-24.00	-24.04	-24.02
2000Hz_2ch	-24.0	-24.04	-23.99	-23.99	-24.10	-24.00	-24.00	-24.04	-24.02
10000Hz_2ch	-24.0	-24.04	-23.99	-23.99	-24.10	-24.00	-24.00	-24.04	-24.02
RelGateTest	-10.0	-10.07	-10.03	-10.03	**-9.60**	-10.00	-10.03	-10.07	**-10.15**
AbsGateTest	-69.5	-69.49	-69.45	**-71.46**	-69.50	-69.50	-69.45	-69.49	-69.55

Table 1. Comparison of loudness algorithm implementations with provided compliance material (ITU-R BS.2217). Measurements that are not within the ± 0.1 dB LUFS tolerance for compliance are marked in **boldface**.

ITU-R BS.2217. Compliance material for recommendation ITU-R BS.1770. May 2011.

Potential Issues

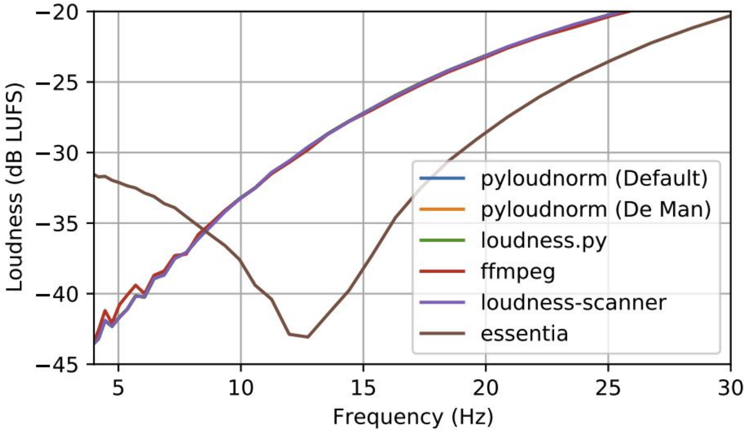


Figure 2. Measured loudness of -6 dB sinusoidal tones.

Essentia filters appear to deviate in very low frequencies.

Challenging compliance material

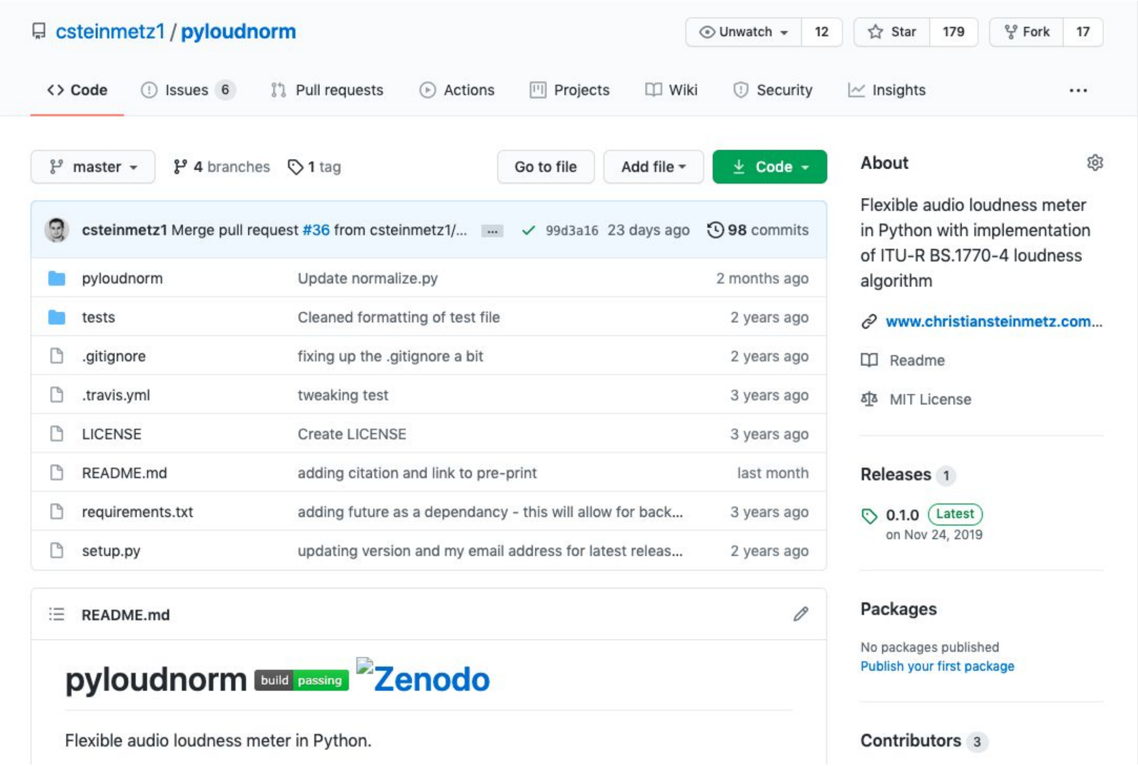
File	Mean	Implementation							
		pyloudnorm		loudness.py	ffmpeg	libebur128	Essentia	Audition	youlean
		Default	De Man						
sine_16Hz	-24.08	-23.44	-23.44	-23.36	-23.50	-23.40	**-28.48**	-23.48	-23.51
sine_1000Hz	-3.13	-3.05	-3.01	-3.01	-3.00	-3.00	-3.01	-3.05	**-3.88**
sine_1000Hz_pad	-4.18	-4.19	-4.15	-4.15	-4.20	-4.10	-4.15	-4.19	-4.32
sine_16000Hz	-19.77	-19.69	-19.64	-19.64	-19.70	-19.60	-19.64	-19.69	-20.52
sine_19000Hz	-19.78	-19.69	-19.64	-19.64	-19.80	-19.60	-19.64	-19.69	**-20.52**
multi-sines	-10.65	-10.67	-10.62	-10.62	-10.60	-10.60	-10.64	-10.67	-10.79
hf-noise	-9.34	-9.21	-9.16	-9.15	-9.60	-9.20	-9.16	-9.21	**-10.04**
chirp-150-190	-6.69	-6.55	-6.50	-6.52	-6.50	-6.50	-6.51	-6.55	**-7.88**
our_gating_test	-3.37	-3.37	-3.33	-3.33	-3.30	-3.30	-3.33	-3.37	-3.61
piano-D6	-25.12	-25.02	-24.98	-24.98	**-28.20**	-25.00	-24.98	-25.03	**-22.73**
soprano-E4	-29.74	-29.82	-29.77	-29.57	-29.60	-29.60	-29.78	-29.61	**-30.15**
vibraphone-C6	-17.29	-16.95	-16.90	-16.90	**-17.90**	-16.90	-19.60	-16.95	**-16.23**
violin-B3	-12.78	-12.82	-12.78	-12.69	-12.70	-12.70	-12.78	-12.74	-13.00

Table 2. Comparison of loudness algorithm implementations with alternative material.
Measurements that disagree with others significantly (≥ 0.5 dB LUFS) are marked in **boldface**.

Runtime

Implementation	RTF	Audio Loader
ffmpeg	26x	ffmpeg
Essentia	88x	Essentia
libebur128	114x	ffmpeg
loudness.py	421x	pysoundfile
pyln (Default)	338x	pysoundfile
pyln (De Man)	455x	pysoundfile

Table 3. Mean real-time factor.



https://github.com/csteinmetz1/pyloudnorm

https://github.com/csteinmetz1/pyloudnorm-eval

Summary

- Easy to install and use loudness package
- Fully compliant ITU-R BS. 1770 implementation
- Enables modifications and future improvements
- One of the fastest Python options available

AES Show Spring 2021 • 150th Audio Engineering Convention • May 25 - 28, 2021

pyloudnorm

A simple yet flexible loudness meter in Python

Christian J. Steinmetz   Joshua D. Reiss

Centre for Digital Music, Queen Mary University of London
UKRI Centre for Doctoral Training in Artificial Intelligence and Music

Queen Mary
University of London