**Core IO and DSP**

**Audio processing**

|  |  |
| --- | --- |
| [**load**](https://librosa.github.io/librosa/generated/librosa.core.load.html#librosa.core.load)(path[, sr, mono, offset, duration, …]) | Load an audio file as a floating point time series. |
| [**to\_mono**](https://librosa.github.io/librosa/generated/librosa.core.to_mono.html#librosa.core.to_mono)(y) | Force an audio signal down to mono. |
| [**resample**](https://librosa.github.io/librosa/generated/librosa.core.resample.html#librosa.core.resample)(y, orig\_sr, target\_sr[, res\_type, …]) | Resample a time series from orig\_sr to target\_sr |
| [**get\_duration**](https://librosa.github.io/librosa/generated/librosa.core.get_duration.html#librosa.core.get_duration)([y, sr, S, n\_fft, hop\_length, …]) | Compute the duration (in seconds) of an audio time series, feature matrix, or filename. |
| [**autocorrelate**](https://librosa.github.io/librosa/generated/librosa.core.autocorrelate.html#librosa.core.autocorrelate)(y[, max\_size, axis]) | Bounded auto-correlation |
| [**zero\_crossings**](https://librosa.github.io/librosa/generated/librosa.core.zero_crossings.html#librosa.core.zero_crossings)(y[, threshold, …]) | Find the zero-crossings of a signal *y*: indices *i* such that *sign(y[i]) != sign(y[j])*. |
| [**clicks**](https://librosa.github.io/librosa/generated/librosa.core.clicks.html#librosa.core.clicks)([times, frames, sr, hop\_length, …]) | Returns a signal with the signal **click** placed at each specified time |
| [**tone**](https://librosa.github.io/librosa/generated/librosa.core.tone.html#librosa.core.tone)(frequency[, sr, length, duration, phi]) | Returns a pure tone signal. |
| [**chirp**](https://librosa.github.io/librosa/generated/librosa.core.chirp.html#librosa.core.chirp)(fmin, fmax[, sr, length, duration, …]) | Returns a chirp signal that goes from frequency *fmin* to frequency *fmax* |

**Spectral representations**

|  |  |
| --- | --- |
| [**stft**](https://librosa.github.io/librosa/generated/librosa.core.stft.html#librosa.core.stft)(y[, n\_fft, hop\_length, win\_length, …]) | Short-time Fourier transform (STFT) |
| [**istft**](https://librosa.github.io/librosa/generated/librosa.core.istft.html#librosa.core.istft)(stft\_matrix[, hop\_length, win\_length, …]) | Inverse short-time Fourier transform (ISTFT). |
| [**ifgram**](https://librosa.github.io/librosa/generated/librosa.core.ifgram.html#librosa.core.ifgram)(y[, sr, n\_fft, hop\_length, …]) | Compute the instantaneous frequency (as a proportion of the sampling rate) obtained as the time-derivative of the phase of the complex spectrum as described by [[Ra44d590316d7-1]](https://librosa.github.io/librosa/generated/librosa.core.ifgram.html#ra44d590316d7-1). |
| [**cqt**](https://librosa.github.io/librosa/generated/librosa.core.cqt.html#librosa.core.cqt)(y[, sr, hop\_length, fmin, n\_bins, …]) | Compute the constant-Q transform of an audio signal. |
| [**icqt**](https://librosa.github.io/librosa/generated/librosa.core.icqt.html#librosa.core.icqt)(C[, sr, hop\_length, fmin, …]) | Compute the inverse constant-Q transform. |
| [**hybrid\_cqt**](https://librosa.github.io/librosa/generated/librosa.core.hybrid_cqt.html#librosa.core.hybrid_cqt)(y[, sr, hop\_length, fmin, …]) | Compute the hybrid constant-Q transform of an audio signal. |
| [**pseudo\_cqt**](https://librosa.github.io/librosa/generated/librosa.core.pseudo_cqt.html#librosa.core.pseudo_cqt)(y[, sr, hop\_length, fmin, …]) | Compute the pseudo constant-Q transform of an audio signal. |
| [**iirt**](https://librosa.github.io/librosa/generated/librosa.core.iirt.html#librosa.core.iirt)(y[, sr, win\_length, hop\_length, …]) | Time-frequency representation using IIR filters [[Rd4077732470d-1]](https://librosa.github.io/librosa/generated/librosa.core.iirt.html#rd4077732470d-1). |
| [**fmt**](https://librosa.github.io/librosa/generated/librosa.core.fmt.html#librosa.core.fmt)(y[, t\_min, n\_fmt, kind, beta, …]) | The fast Mellin transform (FMT) [[R6343f8d4cac9-1]](https://librosa.github.io/librosa/generated/librosa.core.fmt.html#r6343f8d4cac9-1) of a uniformly sampled signal y. |
| [**interp\_harmonics**](https://librosa.github.io/librosa/generated/librosa.core.interp_harmonics.html#librosa.core.interp_harmonics)(x, freqs, h\_range[, kind, …]) | Compute the energy at harmonics of time-frequency representation. |
| [**salience**](https://librosa.github.io/librosa/generated/librosa.core.salience.html#librosa.core.salience)(S, freqs, h\_range[, weights, …]) | Harmonic salience function. |
| [**phase\_vocoder**](https://librosa.github.io/librosa/generated/librosa.core.phase_vocoder.html#librosa.core.phase_vocoder)(D, rate[, hop\_length]) | Phase vocoder. |
| [**magphase**](https://librosa.github.io/librosa/generated/librosa.core.magphase.html#librosa.core.magphase)(D[, power]) | Separate a complex-valued spectrogram D into its magnitude (S) and phase (P) components, so that *D = S \* P*. |

**Magnitude scaling**

|  |  |
| --- | --- |
| [**amplitude\_to\_db**](https://librosa.github.io/librosa/generated/librosa.core.amplitude_to_db.html#librosa.core.amplitude_to_db)(S[, ref, amin, top\_db]) | Convert an amplitude spectrogram to dB-scaled spectrogram. |
| [**db\_to\_amplitude**](https://librosa.github.io/librosa/generated/librosa.core.db_to_amplitude.html#librosa.core.db_to_amplitude)(S\_db[, ref]) | Convert a dB-scaled spectrogram to an amplitude spectrogram. |
| [**power\_to\_db**](https://librosa.github.io/librosa/generated/librosa.core.power_to_db.html#librosa.core.power_to_db)(S[, ref, amin, top\_db]) | Convert a power spectrogram (amplitude squared) to decibel (dB) units |
| [**db\_to\_power**](https://librosa.github.io/librosa/generated/librosa.core.db_to_power.html#librosa.core.db_to_power)(S\_db[, ref]) | Convert a dB-scale spectrogram to a power spectrogram. |
| [**perceptual\_weighting**](https://librosa.github.io/librosa/generated/librosa.core.perceptual_weighting.html#librosa.core.perceptual_weighting)(S, frequencies, \*\*kwargs) | Perceptual weighting of a power spectrogram: |
| [**A\_weighting**](https://librosa.github.io/librosa/generated/librosa.core.A_weighting.html#librosa.core.A_weighting)(frequencies[, min\_db]) | Compute the A-weighting of a set of frequencies. |
| [**pcen**](https://librosa.github.io/librosa/generated/librosa.core.pcen.html#librosa.core.pcen)(S[, sr, hop\_length, gain, bias, power, …]) | Per-channel energy normalization (PCEN) [[Rb388d53f6b92-1]](https://librosa.github.io/librosa/generated/librosa.core.pcen.html#rb388d53f6b92-1) |

**Time and frequency conversion**

|  |  |
| --- | --- |
| [**frames\_to\_samples**](https://librosa.github.io/librosa/generated/librosa.core.frames_to_samples.html#librosa.core.frames_to_samples)(frames[, hop\_length, n\_fft]) | Converts frame indices to audio sample indices. |
| [**frames\_to\_time**](https://librosa.github.io/librosa/generated/librosa.core.frames_to_time.html#librosa.core.frames_to_time)(frames[, sr, hop\_length, n\_fft]) | Converts frame counts to time (seconds). |
| [**samples\_to\_frames**](https://librosa.github.io/librosa/generated/librosa.core.samples_to_frames.html#librosa.core.samples_to_frames)(samples[, hop\_length, n\_fft]) | Converts sample indices into STFT frames. |
| [**samples\_to\_time**](https://librosa.github.io/librosa/generated/librosa.core.samples_to_time.html#librosa.core.samples_to_time)(samples[, sr]) | Convert sample indices to time (in seconds). |
| [**time\_to\_frames**](https://librosa.github.io/librosa/generated/librosa.core.time_to_frames.html#librosa.core.time_to_frames)(times[, sr, hop\_length, n\_fft]) | Converts time stamps into STFT frames. |
| [**time\_to\_samples**](https://librosa.github.io/librosa/generated/librosa.core.time_to_samples.html#librosa.core.time_to_samples)(times[, sr]) | Convert timestamps (in seconds) to sample indices. |
| [**hz\_to\_note**](https://librosa.github.io/librosa/generated/librosa.core.hz_to_note.html#librosa.core.hz_to_note)(frequencies, \*\*kwargs) | Convert one or more frequencies (in Hz) to the nearest note names. |
| [**hz\_to\_midi**](https://librosa.github.io/librosa/generated/librosa.core.hz_to_midi.html#librosa.core.hz_to_midi)(frequencies) | Get MIDI note number(s) for given frequencies |
| [**midi\_to\_hz**](https://librosa.github.io/librosa/generated/librosa.core.midi_to_hz.html#librosa.core.midi_to_hz)(notes) | Get the frequency (Hz) of MIDI note(s) |
| [**midi\_to\_note**](https://librosa.github.io/librosa/generated/librosa.core.midi_to_note.html#librosa.core.midi_to_note)(midi[, octave, cents]) | Convert one or more MIDI numbers to note strings. |
| [**note\_to\_hz**](https://librosa.github.io/librosa/generated/librosa.core.note_to_hz.html#librosa.core.note_to_hz)(note, \*\*kwargs) | Convert one or more note names to frequency (Hz) |
| [**note\_to\_midi**](https://librosa.github.io/librosa/generated/librosa.core.note_to_midi.html#librosa.core.note_to_midi)(note[, round\_midi]) | Convert one or more spelled notes to MIDI number(s). |
| [**hz\_to\_mel**](https://librosa.github.io/librosa/generated/librosa.core.hz_to_mel.html#librosa.core.hz_to_mel)(frequencies[, htk]) | Convert Hz to Mels |
| [**hz\_to\_octs**](https://librosa.github.io/librosa/generated/librosa.core.hz_to_octs.html#librosa.core.hz_to_octs)(frequencies[, A440]) | Convert frequencies (Hz) to (fractional) octave numbers. |
| [**mel\_to\_hz**](https://librosa.github.io/librosa/generated/librosa.core.mel_to_hz.html#librosa.core.mel_to_hz)(mels[, htk]) | Convert mel bin numbers to frequencies |
| [**octs\_to\_hz**](https://librosa.github.io/librosa/generated/librosa.core.octs_to_hz.html#librosa.core.octs_to_hz)(octs[, A440]) | Convert octaves numbers to frequencies. |
| [**fft\_frequencies**](https://librosa.github.io/librosa/generated/librosa.core.fft_frequencies.html#librosa.core.fft_frequencies)([sr, n\_fft]) | Alternative implementation of *np.fft.fftfreq* |
| [**cqt\_frequencies**](https://librosa.github.io/librosa/generated/librosa.core.cqt_frequencies.html#librosa.core.cqt_frequencies)(n\_bins, fmin[, …]) | Compute the center frequencies of Constant-Q bins. |
| [**mel\_frequencies**](https://librosa.github.io/librosa/generated/librosa.core.mel_frequencies.html#librosa.core.mel_frequencies)([n\_mels, fmin, fmax, htk]) | Compute an array of acoustic frequencies tuned to the mel scale. |
| [**tempo\_frequencies**](https://librosa.github.io/librosa/generated/librosa.core.tempo_frequencies.html#librosa.core.tempo_frequencies)(n\_bins[, hop\_length, sr]) | Compute the frequencies (in beats-per-minute) corresponding to an onset auto-correlation or tempogram matrix. |
| [**samples\_like**](https://librosa.github.io/librosa/generated/librosa.core.samples_like.html#librosa.core.samples_like)(X[, hop\_length, n\_fft, axis]) | Return an array of sample indices to match the time axis from a feature matrix. |
| [**times\_like**](https://librosa.github.io/librosa/generated/librosa.core.times_like.html#librosa.core.times_like)(X[, sr, hop\_length, n\_fft, axis]) | Return an array of time values to match the time axis from a feature matrix. |

**Pitch and tuning**

|  |  |
| --- | --- |
| [**estimate\_tuning**](https://librosa.github.io/librosa/generated/librosa.core.estimate_tuning.html#librosa.core.estimate_tuning)([y, sr, S, n\_fft, …]) | Estimate the tuning of an audio time series or spectrogram input. |
| [**pitch\_tuning**](https://librosa.github.io/librosa/generated/librosa.core.pitch_tuning.html#librosa.core.pitch_tuning)(frequencies[, resolution, …]) | Given a collection of pitches, estimate its tuning offset (in fractions of a bin) relative to A440=440.0Hz. |
| [**piptrack**](https://librosa.github.io/librosa/generated/librosa.core.piptrack.html#librosa.core.piptrack)([y, sr, S, n\_fft, hop\_length, …]) | Pitch tracking on thresholded parabolically-interpolated STFT. |

|  |  |
| --- | --- |
| [**specshow**](https://librosa.github.io/librosa/generated/librosa.display.specshow.html#librosa.display.specshow)(data[, x\_coords, y\_coords, x\_axis, …]) | Display a spectrogram/chromagram/cqt/etc. |
| [**waveplot**](https://librosa.github.io/librosa/generated/librosa.display.waveplot.html#librosa.display.waveplot)(y[, sr, max\_points, x\_axis, …]) | Plot the amplitude envelope of a waveform. |
| [**cmap**](https://librosa.github.io/librosa/generated/librosa.display.cmap.html#librosa.display.cmap)(data[, robust, cmap\_seq, cmap\_bool, …]) | Get a default colormap from the given data. |
| [**TimeFormatter**](https://librosa.github.io/librosa/generated/librosa.display.TimeFormatter.html#librosa.display.TimeFormatter)([lag]) | A tick formatter for time axes. |
| [**NoteFormatter**](https://librosa.github.io/librosa/generated/librosa.display.NoteFormatter.html#librosa.display.NoteFormatter)([octave, major]) | Ticker formatter for Notes |
| [**LogHzFormatter**](https://librosa.github.io/librosa/generated/librosa.display.LogHzFormatter.html#librosa.display.LogHzFormatter)([major]) | Ticker formatter for logarithmic frequency |
| [**ChromaFormatter**](https://librosa.github.io/librosa/generated/librosa.display.ChromaFormatter.html#librosa.display.ChromaFormatter) | A formatter for chroma axes |
| [**TonnetzFormatter**](https://librosa.github.io/librosa/generated/librosa.display.TonnetzFormatter.html#librosa.display.TonnetzFormatter) | A formatter for tonnetz axes |

**Display**

|  |  |
| --- | --- |
| [**specshow**](https://librosa.github.io/librosa/generated/librosa.display.specshow.html#librosa.display.specshow)(data[, x\_coords, y\_coords, x\_axis, …]) | Display a spectrogram/chromagram/cqt/etc. |
| [**waveplot**](https://librosa.github.io/librosa/generated/librosa.display.waveplot.html#librosa.display.waveplot)(y[, sr, max\_points, x\_axis, …]) | Plot the amplitude envelope of a waveform. |
| [**cmap**](https://librosa.github.io/librosa/generated/librosa.display.cmap.html#librosa.display.cmap)(data[, robust, cmap\_seq, cmap\_bool, …]) | Get a default colormap from the given data. |
| [**TimeFormatter**](https://librosa.github.io/librosa/generated/librosa.display.TimeFormatter.html#librosa.display.TimeFormatter)([lag]) | A tick formatter for time axes. |
| [**NoteFormatter**](https://librosa.github.io/librosa/generated/librosa.display.NoteFormatter.html#librosa.display.NoteFormatter)([octave, major]) | Ticker formatter for Notes |
| [**LogHzFormatter**](https://librosa.github.io/librosa/generated/librosa.display.LogHzFormatter.html#librosa.display.LogHzFormatter)([major]) | Ticker formatter for logarithmic frequency |
| [**ChromaFormatter**](https://librosa.github.io/librosa/generated/librosa.display.ChromaFormatter.html#librosa.display.ChromaFormatter) | A formatter for chroma axes |
| [**TonnetzFormatter**](https://librosa.github.io/librosa/generated/librosa.display.TonnetzFormatter.html#librosa.display.TonnetzFormatter) | A formatter for tonnetz axes |

**Feature extraction**

**Spectral features**

|  |  |
| --- | --- |
| [**chroma\_stft**](https://librosa.github.io/librosa/generated/librosa.feature.chroma_stft.html#librosa.feature.chroma_stft)([y, sr, S, norm, n\_fft, …]) | Compute a chromagram from a waveform or power spectrogram. |
| [**chroma\_cqt**](https://librosa.github.io/librosa/generated/librosa.feature.chroma_cqt.html#librosa.feature.chroma_cqt)([y, sr, C, hop\_length, fmin, …]) | Constant-Q chromagram |
| [**chroma\_cens**](https://librosa.github.io/librosa/generated/librosa.feature.chroma_cens.html#librosa.feature.chroma_cens)([y, sr, C, hop\_length, fmin, …]) | Computes the chroma variant “Chroma Energy Normalized” (CENS), following [[R674badebce0d-1]](https://librosa.github.io/librosa/generated/librosa.feature.chroma_cens.html#r674badebce0d-1). |
| [**melspectrogram**](https://librosa.github.io/librosa/generated/librosa.feature.melspectrogram.html#librosa.feature.melspectrogram)([y, sr, S, n\_fft, …]) | Compute a mel-scaled spectrogram. |
| [**mfcc**](https://librosa.github.io/librosa/generated/librosa.feature.mfcc.html#librosa.feature.mfcc)([y, sr, S, n\_mfcc, dct\_type, norm]) | Mel-frequency cepstral coefficients (MFCCs) |
| [**rmse**](https://librosa.github.io/librosa/generated/librosa.feature.rmse.html#librosa.feature.rmse)([y, S, frame\_length, hop\_length, …]) | Compute root-mean-square (RMS) energy for each frame, either from the audio samples *y* or from a spectrogram *S*. |
| [**spectral\_centroid**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_centroid.html#librosa.feature.spectral_centroid)([y, sr, S, n\_fft, …]) | Compute the spectral centroid. |
| [**spectral\_bandwidth**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_bandwidth.html#librosa.feature.spectral_bandwidth)([y, sr, S, n\_fft, …]) | Compute p’th-order spectral bandwidth: |
| [**spectral\_contrast**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_contrast.html#librosa.feature.spectral_contrast)([y, sr, S, n\_fft, …]) | Compute spectral contrast [[R6ffcc01153df-1]](https://librosa.github.io/librosa/generated/librosa.feature.spectral_contrast.html#r6ffcc01153df-1) |
| [**spectral\_flatness**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_flatness.html#librosa.feature.spectral_flatness)([y, S, n\_fft, hop\_length, …]) | Compute spectral flatness |
| [**spectral\_rolloff**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_rolloff.html#librosa.feature.spectral_rolloff)([y, sr, S, n\_fft, …]) | Compute roll-off frequency. |
| [**poly\_features**](https://librosa.github.io/librosa/generated/librosa.feature.poly_features.html#librosa.feature.poly_features)([y, sr, S, n\_fft, hop\_length, …]) | Get coefficients of fitting an nth-order polynomial to the columns of a spectrogram. |
| [**tonnetz**](https://librosa.github.io/librosa/generated/librosa.feature.tonnetz.html#librosa.feature.tonnetz)([y, sr, chroma]) | Computes the tonal centroid features (tonnetz), following the method of [[Recf246e5a035-1]](https://librosa.github.io/librosa/generated/librosa.feature.tonnetz.html#recf246e5a035-1). |
| [**zero\_crossing\_rate**](https://librosa.github.io/librosa/generated/librosa.feature.zero_crossing_rate.html#librosa.feature.zero_crossing_rate)(y[, frame\_length, …]) | Compute the zero-crossing rate of an audio time series. |

**Rhythm features**

|  |  |
| --- | --- |
| [**tempogram**](https://librosa.github.io/librosa/generated/librosa.feature.tempogram.html#librosa.feature.tempogram)([y, sr, onset\_envelope, …]) | Compute the tempogram: local autocorrelation of the onset strength envelope. |

**Feature manipulation**

|  |  |
| --- | --- |
| [**delta**](https://librosa.github.io/librosa/generated/librosa.feature.delta.html#librosa.feature.delta)(data[, width, order, axis, trim, mode]) | Compute delta features: local estimate of the derivative of the input data along the selected axis. |
| [**stack\_memory**](https://librosa.github.io/librosa/generated/librosa.feature.stack_memory.html#librosa.feature.stack_memory)(data[, n\_steps, delay]) | Short-term history embedding: vertically concatenate a data vector or matrix with delayed copies of itself |

**Feature extraction**

**Spectral features**

|  |  |
| --- | --- |
| [**chroma\_stft**](https://librosa.github.io/librosa/generated/librosa.feature.chroma_stft.html#librosa.feature.chroma_stft)([y, sr, S, norm, n\_fft, …]) | Compute a chromagram from a waveform or power spectrogram. |
| [**chroma\_cqt**](https://librosa.github.io/librosa/generated/librosa.feature.chroma_cqt.html#librosa.feature.chroma_cqt)([y, sr, C, hop\_length, fmin, …]) | Constant-Q chromagram |
| [**chroma\_cens**](https://librosa.github.io/librosa/generated/librosa.feature.chroma_cens.html#librosa.feature.chroma_cens)([y, sr, C, hop\_length, fmin, …]) | Computes the chroma variant “Chroma Energy Normalized” (CENS), following [[R674badebce0d-1]](https://librosa.github.io/librosa/generated/librosa.feature.chroma_cens.html#r674badebce0d-1). |
| [**melspectrogram**](https://librosa.github.io/librosa/generated/librosa.feature.melspectrogram.html#librosa.feature.melspectrogram)([y, sr, S, n\_fft, …]) | Compute a mel-scaled spectrogram. |
| [**mfcc**](https://librosa.github.io/librosa/generated/librosa.feature.mfcc.html#librosa.feature.mfcc)([y, sr, S, n\_mfcc, dct\_type, norm]) | Mel-frequency cepstral coefficients (MFCCs) |
| [**rmse**](https://librosa.github.io/librosa/generated/librosa.feature.rmse.html#librosa.feature.rmse)([y, S, frame\_length, hop\_length, …]) | Compute root-mean-square (RMS) energy for each frame, either from the audio samples *y* or from a spectrogram *S*. |
| [**spectral\_centroid**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_centroid.html#librosa.feature.spectral_centroid)([y, sr, S, n\_fft, …]) | Compute the spectral centroid. |
| [**spectral\_bandwidth**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_bandwidth.html#librosa.feature.spectral_bandwidth)([y, sr, S, n\_fft, …]) | Compute p’th-order spectral bandwidth: |
| [**spectral\_contrast**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_contrast.html#librosa.feature.spectral_contrast)([y, sr, S, n\_fft, …]) | Compute spectral contrast [[R6ffcc01153df-1]](https://librosa.github.io/librosa/generated/librosa.feature.spectral_contrast.html#r6ffcc01153df-1) |
| [**spectral\_flatness**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_flatness.html#librosa.feature.spectral_flatness)([y, S, n\_fft, hop\_length, …]) | Compute spectral flatness |
| [**spectral\_rolloff**](https://librosa.github.io/librosa/generated/librosa.feature.spectral_rolloff.html#librosa.feature.spectral_rolloff)([y, sr, S, n\_fft, …]) | Compute roll-off frequency. |
| [**poly\_features**](https://librosa.github.io/librosa/generated/librosa.feature.poly_features.html#librosa.feature.poly_features)([y, sr, S, n\_fft, hop\_length, …]) | Get coefficients of fitting an nth-order polynomial to the columns of a spectrogram. |
| [**tonnetz**](https://librosa.github.io/librosa/generated/librosa.feature.tonnetz.html#librosa.feature.tonnetz)([y, sr, chroma]) | Computes the tonal centroid features (tonnetz), following the method of [[Recf246e5a035-1]](https://librosa.github.io/librosa/generated/librosa.feature.tonnetz.html#recf246e5a035-1). |
| [**zero\_crossing\_rate**](https://librosa.github.io/librosa/generated/librosa.feature.zero_crossing_rate.html#librosa.feature.zero_crossing_rate)(y[, frame\_length, …]) | Compute the zero-crossing rate of an audio time series. |

**Rhythm features**

|  |  |
| --- | --- |
| [**tempogram**](https://librosa.github.io/librosa/generated/librosa.feature.tempogram.html#librosa.feature.tempogram)([y, sr, onset\_envelope, …]) | Compute the tempogram: local autocorrelation of the onset strength envelope. |

**Feature manipulation**

|  |  |
| --- | --- |
| [**delta**](https://librosa.github.io/librosa/generated/librosa.feature.delta.html#librosa.feature.delta)(data[, width, order, axis, trim, mode]) | Compute delta features: local estimate of the derivative of the input data along the selected axis. |
| [**stack\_memory**](https://librosa.github.io/librosa/generated/librosa.feature.stack_memory.html#librosa.feature.stack_memory)(data[, n\_steps, delay]) | Short-term history embedding: vertically concatenate a data vector or matrix with delayed copies of itself |

**Beat and tempo**

|  |  |
| --- | --- |
| [**beat\_track**](https://librosa.github.io/librosa/generated/librosa.beat.beat_track.html#librosa.beat.beat_track)([y, sr, onset\_envelope, …]) | Dynamic programming beat tracker. |
| [**tempo**](https://librosa.github.io/librosa/generated/librosa.beat.tempo.html#librosa.beat.tempo)([y, sr, onset\_envelope, hop\_length, …]) | Estimate the tempo (beats per minute) |
| [decompose](https://librosa.github.io/librosa/generated/librosa.decompose.decompose.html#librosa.decompose.decompose)(S[, n\_components, transformer, …]) | Decompose a feature matrix. |
| [hpss](https://librosa.github.io/librosa/generated/librosa.decompose.hpss.html#librosa.decompose.hpss)(S[, kernel\_size, power, mask, margin]) | Median-filtering harmonic percussive source separation (HPSS). |
| [nn\_filter](https://librosa.github.io/librosa/generated/librosa.decompose.nn_filter.html#librosa.decompose.nn_filter)(S[, rec, aggregate, axis]) | Filtering by nearest-neighbors. |

**Effects**

**Harmonic-percussive source separation**

|  |  |
| --- | --- |
| [**hpss**](https://librosa.github.io/librosa/generated/librosa.effects.hpss.html#librosa.effects.hpss)(y, \*\*kwargs) | Decompose an audio time series into harmonic and percussive components. |
| [**harmonic**](https://librosa.github.io/librosa/generated/librosa.effects.harmonic.html#librosa.effects.harmonic)(y, \*\*kwargs) | Extract harmonic elements from an audio time-series. |
| [**percussive**](https://librosa.github.io/librosa/generated/librosa.effects.percussive.html#librosa.effects.percussive)(y, \*\*kwargs) | Extract percussive elements from an audio time-series. |

**Time and frequency**

|  |  |
| --- | --- |
| [**time\_stretch**](https://librosa.github.io/librosa/generated/librosa.effects.time_stretch.html#librosa.effects.time_stretch)(y, rate) | Time-stretch an audio series by a fixed rate. |
| [**pitch\_shift**](https://librosa.github.io/librosa/generated/librosa.effects.pitch_shift.html#librosa.effects.pitch_shift)(y, sr, n\_steps[, bins\_per\_octave]) | Pitch-shift the waveform by *n\_steps* half-steps. |

**Miscellaneous**

|  |  |
| --- | --- |
| [**remix**](https://librosa.github.io/librosa/generated/librosa.effects.remix.html#librosa.effects.remix)(y, intervals[, align\_zeros]) | Remix an audio signal by re-ordering time intervals. |
| [**trim**](https://librosa.github.io/librosa/generated/librosa.effects.trim.html#librosa.effects.trim)(y[, top\_db, ref, frame\_length, hop\_length]) | Trim leading and trailing silence from an audio signal. |
| [**split**](https://librosa.github.io/librosa/generated/librosa.effects.split.html#librosa.effects.split)(y[, top\_db, ref, frame\_length, hop\_length]) | Split an audio signal into non-silent intervals. |

**Output**

**Text output**

|  |  |
| --- | --- |
| [**annotation**](https://librosa.github.io/librosa/generated/librosa.output.annotation.html#librosa.output.annotation)(path, intervals[, annotations, …]) | Save annotations in a 3-column format: |
| [**times\_csv**](https://librosa.github.io/librosa/generated/librosa.output.times_csv.html#librosa.output.times_csv)(path, times[, annotations, …]) | Save time steps as in CSV format. |

**Audio output**

|  |  |
| --- | --- |
| [**write\_wav**](https://librosa.github.io/librosa/generated/librosa.output.write_wav.html#librosa.output.write_wav)(path, y, sr[, norm]) | Output a time series as a .wav file |

**Temporal segmentation**

**Recurrence and self-similarity**

|  |  |
| --- | --- |
| [**recurrence\_matrix**](https://librosa.github.io/librosa/generated/librosa.segment.recurrence_matrix.html#librosa.segment.recurrence_matrix)(data[, k, width, metric, …]) | Compute a recurrence matrix from a data matrix. |
| [**recurrence\_to\_lag**](https://librosa.github.io/librosa/generated/librosa.segment.recurrence_to_lag.html#librosa.segment.recurrence_to_lag)(rec[, pad, axis]) | Convert a recurrence matrix into a lag matrix. |
| [**lag\_to\_recurrence**](https://librosa.github.io/librosa/generated/librosa.segment.lag_to_recurrence.html#librosa.segment.lag_to_recurrence)(lag[, axis]) | Convert a lag matrix into a recurrence matrix. |
| [**timelag\_filter**](https://librosa.github.io/librosa/generated/librosa.segment.timelag_filter.html#librosa.segment.timelag_filter)(function[, pad, index]) | Filtering in the time-lag domain. |

**Temporal clustering**

|  |  |
| --- | --- |
| [**agglomerative**](https://librosa.github.io/librosa/generated/librosa.segment.agglomerative.html#librosa.segment.agglomerative)(data, k[, clusterer, axis]) | Bottom-up temporal segmentation. |
| [**subsegment**](https://librosa.github.io/librosa/generated/librosa.segment.subsegment.html#librosa.segment.subsegment)(data, frames[, n\_segments, axis]) | Sub-divide a segmentation by feature clustering. |

**Utilities**

**Array operations**

|  |  |
| --- | --- |
| [**frame**](https://librosa.github.io/librosa/generated/librosa.util.frame.html#librosa.util.frame)(y[, frame\_length, hop\_length]) | Slice a time series into overlapping frames. |
| [**pad\_center**](https://librosa.github.io/librosa/generated/librosa.util.pad_center.html#librosa.util.pad_center)(data, size[, axis]) | Wrapper for np.pad to automatically center an array prior to padding. |
| [**fix\_length**](https://librosa.github.io/librosa/generated/librosa.util.fix_length.html#librosa.util.fix_length)(data, size[, axis]) | Fix the length an array **data** to exactly *size*. |
| [**fix\_frames**](https://librosa.github.io/librosa/generated/librosa.util.fix_frames.html#librosa.util.fix_frames)(frames[, x\_min, x\_max, pad]) | Fix a list of frames to lie within [x\_min, x\_max] |
| [**index\_to\_slice**](https://librosa.github.io/librosa/generated/librosa.util.index_to_slice.html#librosa.util.index_to_slice)(idx[, idx\_min, idx\_max, …]) | Generate a slice array from an index array. |
| [**softmask**](https://librosa.github.io/librosa/generated/librosa.util.softmask.html#librosa.util.softmask)(X, X\_ref[, power, split\_zeros]) | Robustly compute a softmask operation. |
| [**sync**](https://librosa.github.io/librosa/generated/librosa.util.sync.html#librosa.util.sync)(data, idx[, aggregate, pad, axis]) | Synchronous aggregation of a multi-dimensional array between boundaries |
| [**softmask**](https://librosa.github.io/librosa/generated/librosa.util.softmask.html#librosa.util.softmask)(X, X\_ref[, power, split\_zeros]) | Robustly compute a softmask operation. |
| [**axis\_sort**](https://librosa.github.io/librosa/generated/librosa.util.axis_sort.html#librosa.util.axis_sort)(S[, axis, index, value]) | Sort an array along its rows or columns. |
| [**normalize**](https://librosa.github.io/librosa/generated/librosa.util.normalize.html#librosa.util.normalize)(S[, norm, axis, threshold, fill]) | Normalize an array along a chosen axis. |
| [**roll\_sparse**](https://librosa.github.io/librosa/generated/librosa.util.roll_sparse.html#librosa.util.roll_sparse)(x, shift[, axis]) | Sparse matrix roll |
| [**sparsify\_rows**](https://librosa.github.io/librosa/generated/librosa.util.sparsify_rows.html#librosa.util.sparsify_rows)(x[, quantile]) | Return a row-sparse matrix approximating the input *x*. |
| [**buf\_to\_float**](https://librosa.github.io/librosa/generated/librosa.util.buf_to_float.html#librosa.util.buf_to_float)(x[, n\_bytes, dtype]) | Convert an integer buffer to floating point values. |
| [**tiny**](https://librosa.github.io/librosa/generated/librosa.util.tiny.html#librosa.util.tiny)(x) | Compute the tiny-value corresponding to an input’s data type. |

**Matching**

|  |  |
| --- | --- |
| [**match\_intervals**](https://librosa.github.io/librosa/generated/librosa.util.match_intervals.html#librosa.util.match_intervals)(intervals\_from, intervals\_to) | Match one set of time intervals to another. |
| [**match\_events**](https://librosa.github.io/librosa/generated/librosa.util.match_events.html#librosa.util.match_events)(events\_from, events\_to[, left, …]) | Match one set of events to another. |

**Miscellaneous**

|  |  |
| --- | --- |
| [**localmax**](https://librosa.github.io/librosa/generated/librosa.util.localmax.html#librosa.util.localmax)(x[, axis]) | Find local maxima in an array *x*. |
| [**peak\_pick**](https://librosa.github.io/librosa/generated/librosa.util.peak_pick.html#librosa.util.peak_pick)(x, pre\_max, post\_max, pre\_avg, …) | Uses a flexible heuristic to pick peaks in a signal. |

**Input validation**

|  |  |
| --- | --- |
| [**valid\_audio**](https://librosa.github.io/librosa/generated/librosa.util.valid_audio.html#librosa.util.valid_audio)(y[, mono]) | Validate whether a variable contains valid, mono audio data. |
| [**valid\_int**](https://librosa.github.io/librosa/generated/librosa.util.valid_int.html#librosa.util.valid_int)(x[, cast]) | Ensure that an input value is integer-typed. |
| [**valid\_intervals**](https://librosa.github.io/librosa/generated/librosa.util.valid_intervals.html#librosa.util.valid_intervals)(intervals) | Ensure that an array is a valid representation of time intervals: |

**File operations**

|  |  |
| --- | --- |
| [**example\_audio\_file**](https://librosa.github.io/librosa/generated/librosa.util.example_audio_file.html#librosa.util.example_audio_file)() | Get the path to an included audio example file. |
| [**find\_files**](https://librosa.github.io/librosa/generated/librosa.util.find_files.html#librosa.util.find_files)(directory[, ext, recurse, …]) | Get a sorted list of (audio) files in a directory or directory sub-tree. |

**Filters**

**Filter bank construction**

|  |  |
| --- | --- |
| [**mel**](https://librosa.github.io/librosa/generated/librosa.filters.mel.html#librosa.filters.mel)(sr, n\_fft[, n\_mels, fmin, fmax, htk, norm]) | Create a Filterbank matrix to combine FFT bins into Mel-frequency bins |
| [**chroma**](https://librosa.github.io/librosa/generated/librosa.filters.chroma.html#librosa.filters.chroma)(sr, n\_fft[, n\_chroma, A440, ctroct, …]) | Create a Filterbank matrix to convert STFT to chroma |
| [**constant\_q**](https://librosa.github.io/librosa/generated/librosa.filters.constant_q.html#librosa.filters.constant_q)(sr[, fmin, n\_bins, …]) | Construct a constant-Q basis. |
| [**\_multirate\_fb**](https://librosa.github.io/librosa/generated/librosa.filters._multirate_fb.html#librosa.filters._multirate_fb)([center\_freqs, sample\_rates, …]) | Helper function to construct a multirate filterbank. |
| [**semitone\_filterbank**](https://librosa.github.io/librosa/generated/librosa.filters.semitone_filterbank.html#librosa.filters.semitone_filterbank)([center\_freqs, tuning, …]) | Constructs a multirate filterbank of infinite-impulse response (IIR) band-pass filters at user-defined center frequencies and sample rates. |

**Window functions**

|  |  |
| --- | --- |
| [**window\_bandwidth**](https://librosa.github.io/librosa/generated/librosa.filters.window_bandwidth.html#librosa.filters.window_bandwidth)(window[, n]) | Get the equivalent noise bandwidth of a window function. |
| [**get\_window**](https://librosa.github.io/librosa/generated/librosa.filters.get_window.html#librosa.filters.get_window)(window, Nx[, fftbins]) | Compute a window function. |

**Miscellaneous**

|  |  |
| --- | --- |
| [**constant\_q\_lengths**](https://librosa.github.io/librosa/generated/librosa.filters.constant_q_lengths.html#librosa.filters.constant_q_lengths)(sr, fmin[, n\_bins, …]) | Return length of each filter in a constant-Q basis. |
| [**cq\_to\_chroma**](https://librosa.github.io/librosa/generated/librosa.filters.cq_to_chroma.html#librosa.filters.cq_to_chroma)(n\_input[, bins\_per\_octave, …]) | Convert a Constant-Q basis to Chroma. |
| [**mr\_frequencies**](https://librosa.github.io/librosa/generated/librosa.filters.mr_frequencies.html#librosa.filters.mr_frequencies)(tuning) | Helper function for generating center frequency and sample rate pairs. |
| [**window\_sumsquare**](https://librosa.github.io/librosa/generated/librosa.filters.window_sumsquare.html#librosa.filters.window_sumsquare)(window, n\_frames[, …]) | Compute the sum-square envelope of a window function at a given hop length. |

**Deprecated**

|  |  |
| --- | --- |
| [**dct**](https://librosa.github.io/librosa/generated/librosa.filters.dct.html#librosa.filters.dct)(n\_filters, n\_input) | Discrete cosine transform (DCT type-II, normalized) basis. |