Tampere University

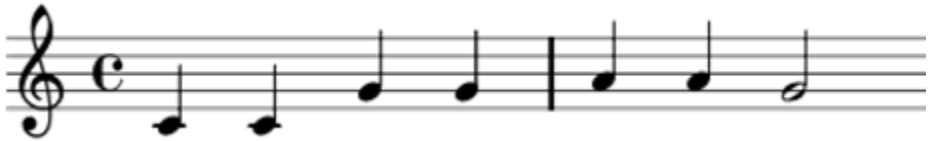
Music signals processing

SGN 14007

Lecture 11

Annamaria Mesaros

Music

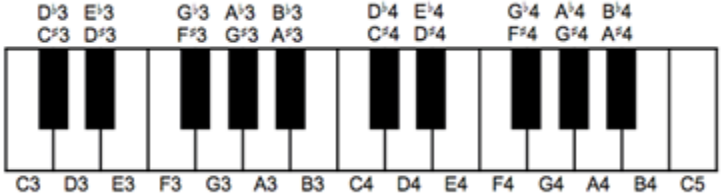


- Music is one of the the richest and most carefully constructed audio signals
- Compared to speech:
  - Music has several overlapping and simultaneously active sources
  - Prominent structures are organized in both time and frequency
- Music waveform carries more information than most other signals
- Music has its own written notation (musical score, see Fig.)
  - Describes musical work with formal language based of musical symbols and letters
  - Musician can create a performance by following such instructions
  - Similar to different languages, one needs to study this language to master it.

In this course, we don’t assume prior knowledge of music.

Basic concepts required by the automatic processing methods are explained.

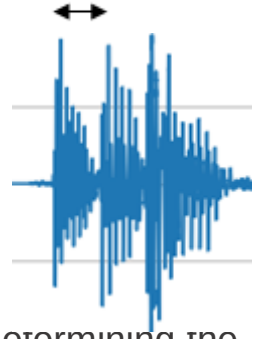
Music terms



- Western music uses log-scale with 12 notes in each octave range
  - Notes are one semitone apart
  - Called twelve-tone equal-tempered scale
- Nominal F0 of note *n* is $440*2^{(n/12)}$ (Hz)
  - 440 Hz is an agreed anchor point
  - Index *n* varies from -48 to 39 on a standard piano keyboard
- Notes in each octave are lettered:
  - C,C#,D,D#,E,F,G,G#,A,A#,B
  - The pound symbol (#), read as “sharp”, increases pitch of a note by one semitone
- The octave range is indicated with a number following the letter.
  - By convention: A4 = 440 Hz
  - E.g. A4# = 466.16 Hz, A5#=932,3 Hz

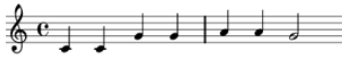
https://pages.mtu.edu/~suits/notefreqs.html

Music terms



- Inter-onset interval (IOI)
  - Time interval between beginning of two sounds
  - Defines the rhythmic characteristic of a melody
  - Although duration plays a role too, the IOIs are more crucial in determining the perceived rhythm
  - Rhythmically important instruments (e.g. drums) produce exponentially decaying waveshapes without a unique definition of duration.

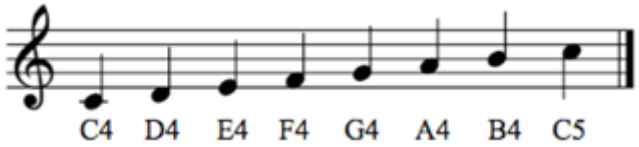
- Melody
  - Series of pitched sounds with musically meaningful pitch and IOI relationships
  - e.g. sequence of written notes



- Chord
  - A combination of three or more simultaneous notes

Musical score

- Instruments such as the violin, or a singer, can produce notes with arbitrary pitch values
- These are quantized into notes



Symbolic representation of music:

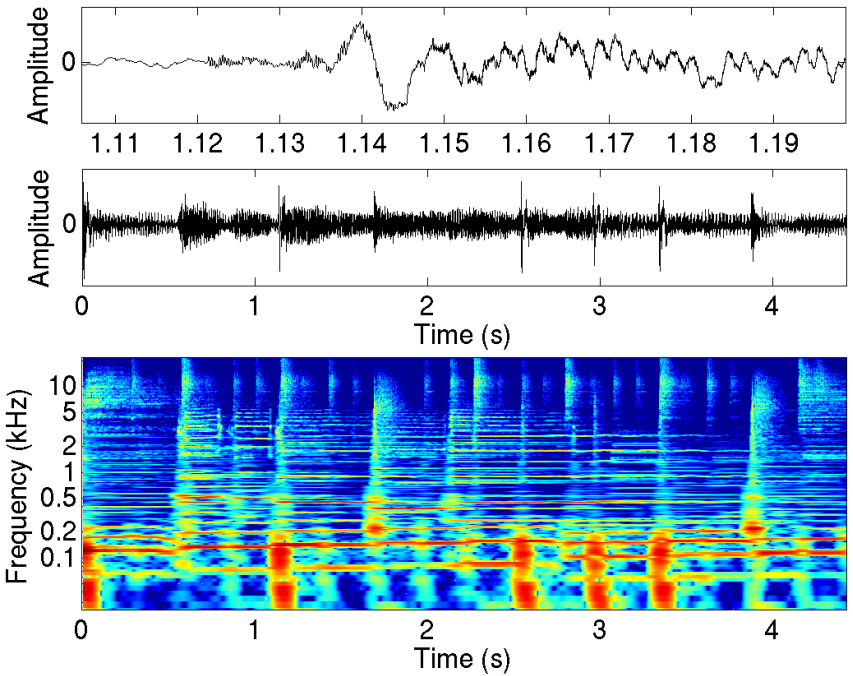
- Figure: staff (5 lines, 4 spaces) denote pitch
  - Pitch raises upwards
- Time goes from left to right
- The clef at the start of the staff indicates the position of a particular note on the staff.
  - Figure: G-clef, indicates second line is pitch G4
- Also timing of notes is quantized. (Whole note, quarter note, one-eigth, etc.)
  - Denoted with different shapes of note symbols (or additional symbols)
- Timbral information is quantized by naming the instrument

Constant Q transform

The part about constant Q transform is not required for exam!

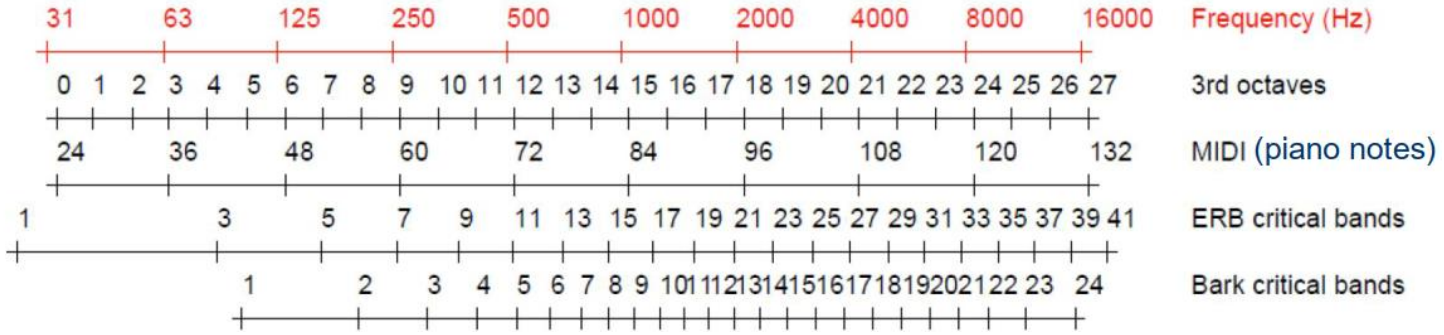
STFT Spectrogram

- Spectrum estimated in short consecutive frames
  - Windowing to avoid spectral blurring
- Transient-like sounds are difficult to represent and process in the frequency domain
  - Results in time blurring



Constant Q transform

- Time-frequency representation where the frequency bins are uniformly distributed in log-frequency and their Q-factors (ratios of center frequencies to bandwidths) are all equal
- In effect, that means that the frequency resolution is better for low frequencies and the time resolution is better for high frequencies
- Musically and perceptually motivated
  - frequency resolution of the inner ear is approx. constant Q above 500 Hz
  - in music (equal temperament), note frequencies obey Fk = 440Hz 2k/12



Constant Q transform

- Mathematical definition: $X_{CQT}(k,n) = \sum_{m=0}^{N-1} x(m) g_k(m-n) \mathrm{e}^{-j 2\pi m k f_k / f_s}$

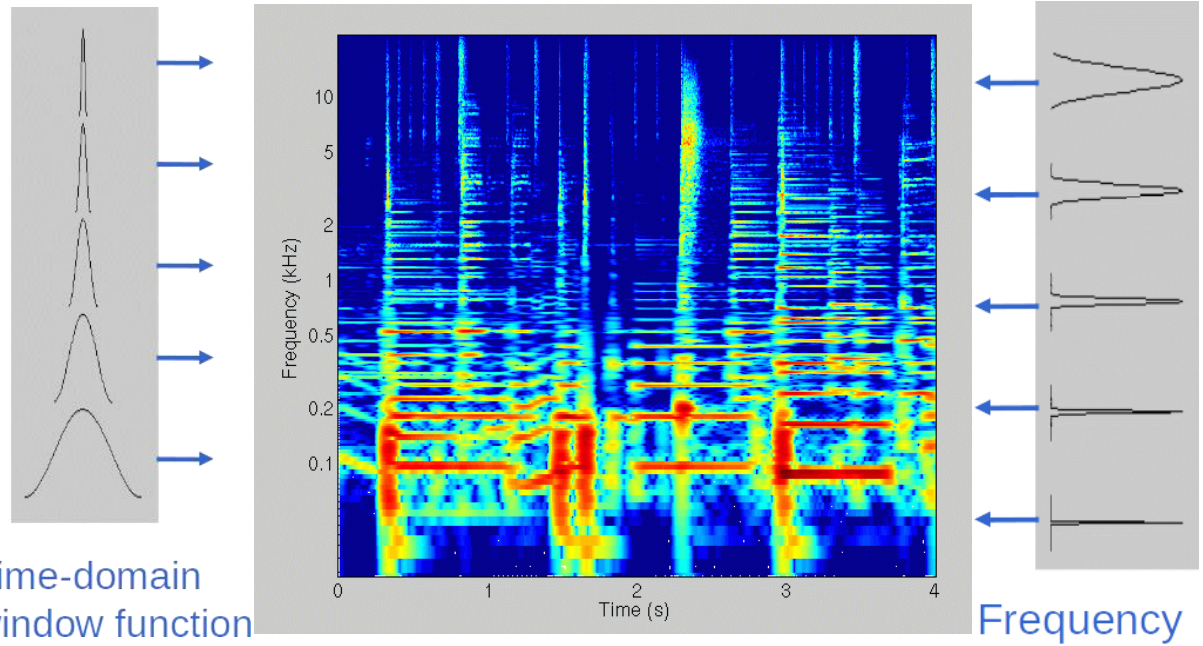
  where N is length of input signal, $g_k(m)$ is zero-centered window function that picks one time frame of the signal at point n and f_s sample rate

- Compare CQT with short-time Fourier transform (STFT): $X_{STFT}(k,n) = \sum_{m=0}^{N-1} x(m) h(m-n) e^{-j 2\pi m k f_k / f_s}$

  where now the window function h(m) is the same for all frequency bins

- In CQT, to achieve constant Q-factors, the support of the window (time length of significant non-zero values) is inversely proportional to f_k
- In CQT, the center frequencies are geometrically spaced: $f_k = f_0 2^{k/B}$ where B determines the number of bins per octave and f_0 is lowest bin
- In DFT, the center frequencies are linearly spaced: $f_k = k f_{resolution}$

Constant Q transform illustrated



Time-domain window function

Frequency resolution

Computing CQT

Efficient computation achieved by:

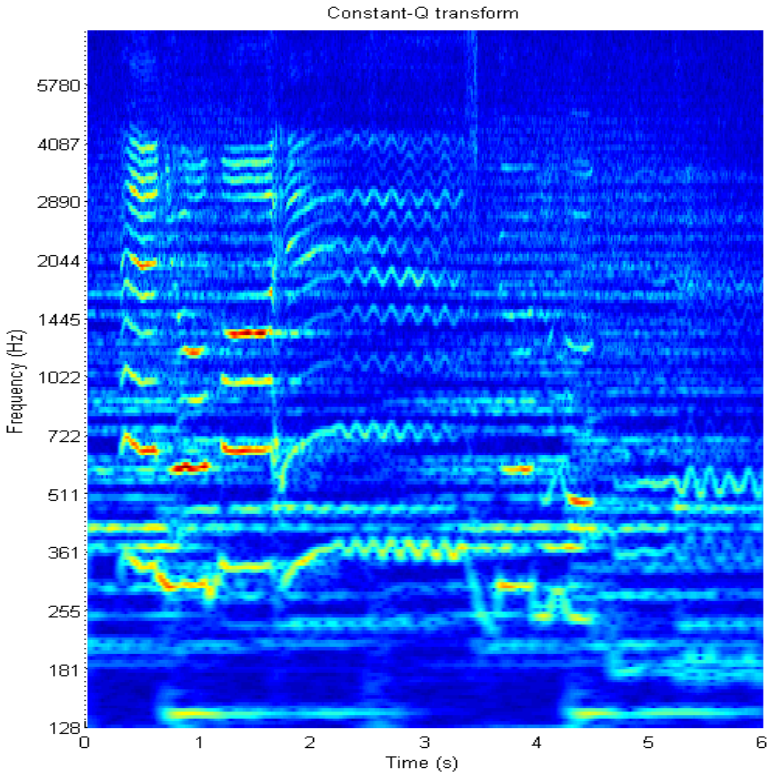
1. FFT of the entire input signal
2. apply one CQT frequency-bin wide bandpass on the (huge) spectrum
3. move subband around zero
4. inverse-FFT transform the narrowband spectrum to get CQT coefficients over time for that bin

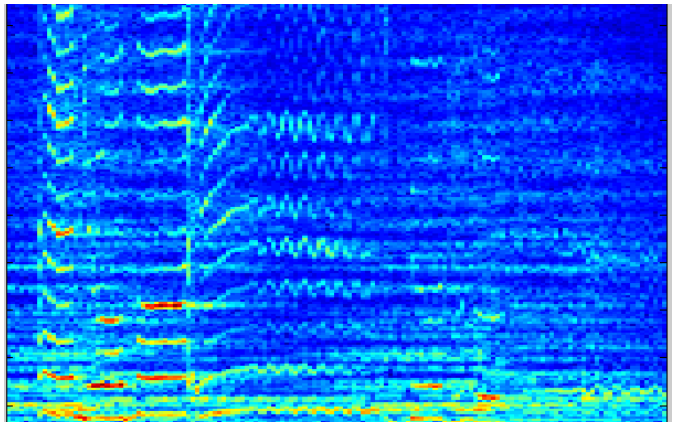
Schörkhuber, C., Klapuri, N. Holighaus, and M. Döfler, "A Matlab Toolbox for Efficient Perfect Reconstruction Time-Frequency Transforms with Log-Frequency Resolution," AES 53rd International Conference on Semantic Audio, London, UK
https://www.cs.tut.fi/sgn/arg/CQT/

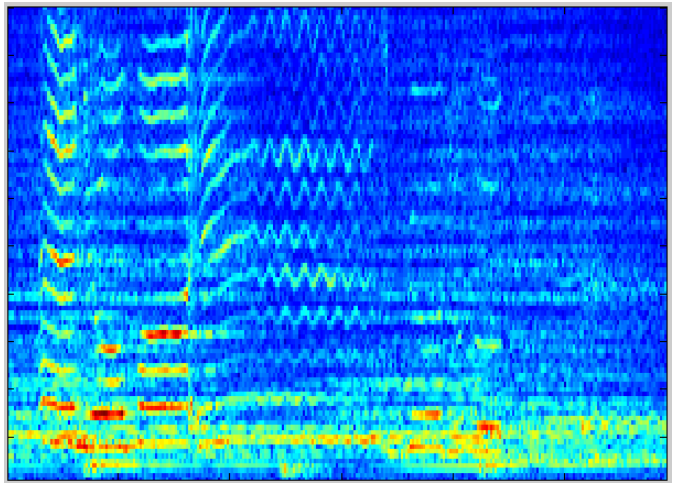
librosa.core.cqt

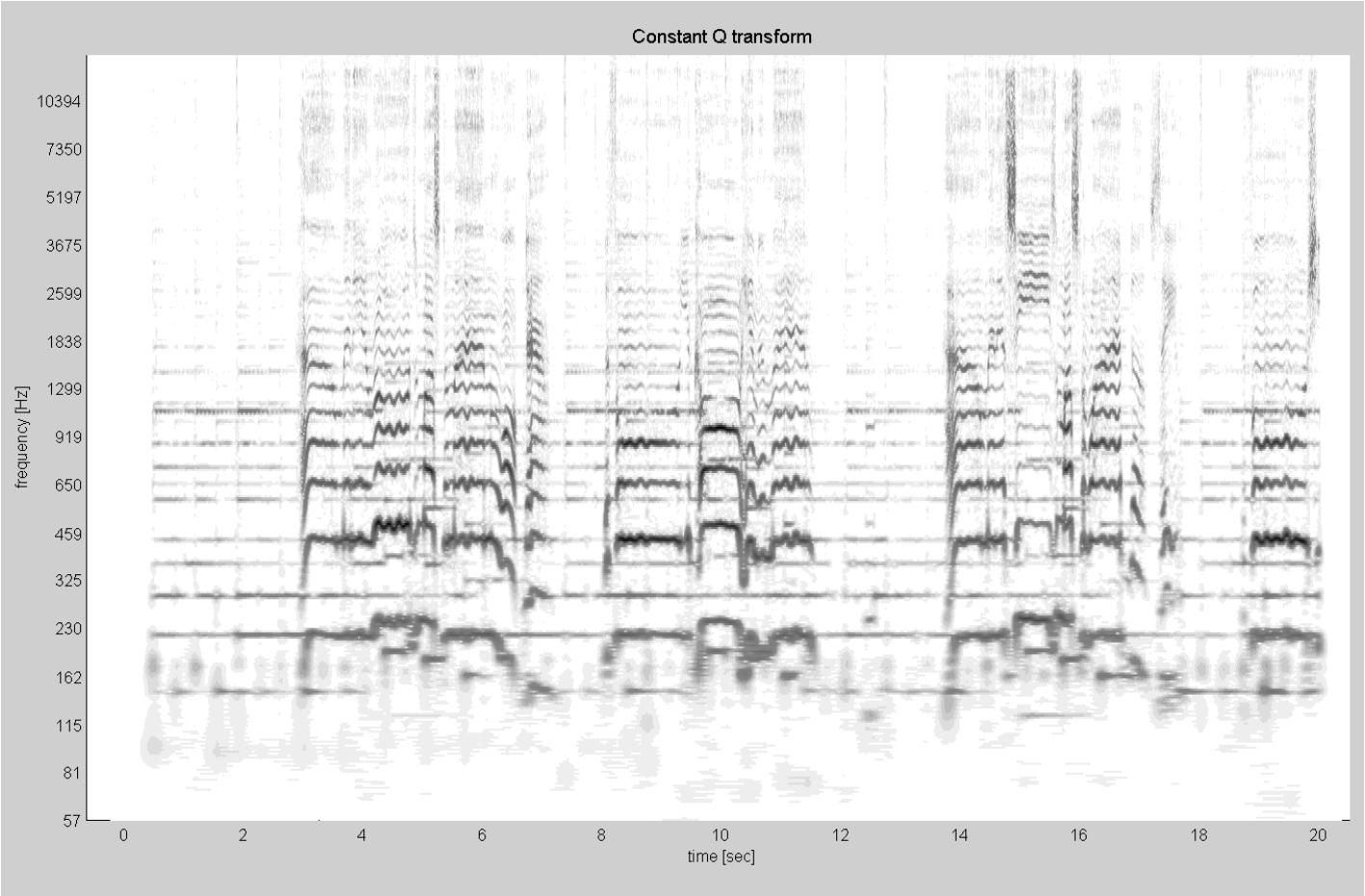
Tampere University

CQT vs STFT



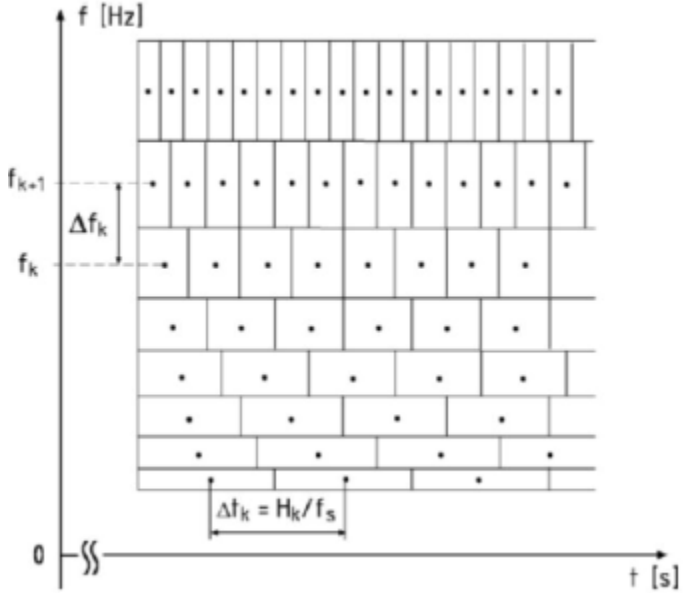





Constant Q transform
frequency [Hz]
10394
7350
5197
3675
2599
1838
1299
919
650
459
325
230
162
115
81
57
0
2
4
6
8
10
12
14
16
18
20
time [sec]

Drawbacks of CQT

- CQT is computationally more intensive than DFT spectrogram
- CQT produces a data structure that is more difficult to handle than the time-frequency matrix (spectrogram)



Example application: Pitch shifting

Pitch shifting is a natural operation in CQT domain:

1. CQT
2. translate CQT coefficients or in frequency
3. retain phase coherence
4. inverse CQT

Examples:

Original	- 6 semitones	+6 semitones
		
		
		

Transients at high freqs. are retained due to short frame

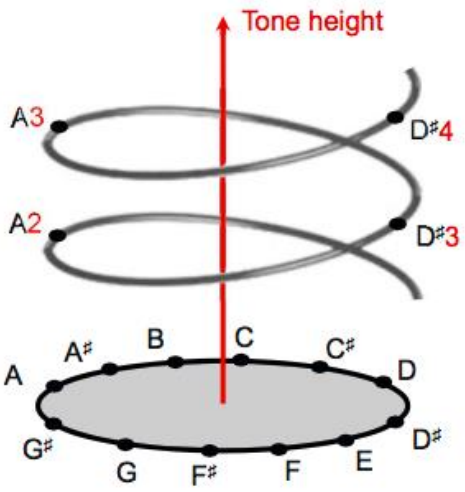
Thinking break (2 min)

Chroma

Pitch class profile

- Two notes with F0 in a ratio of any power of two are perceived very similar
  - Such notes can be grouped under the same **pitch class**
  - Also leads to notion of an **octave**: interval between one musical note and another with half or twice the fundamental frequency.

- In a **pitch class profile**, the entire spectrum is folded onto a single octave (e.g. 12 bins representing each semitone)
- **Chroma representation** gives a description of the different notes present in the audio *without distinguishing the octave in which they occur.*
- Figure: linear pitch space is represented by a helix wrapped around a cylinder.
  - The projection onto horizontal plane is the chromatic circle.



Chromagram

- The main idea of chroma features is to aggregate all spectral information that relates to a given pitch class into a single coefficient
- Basic algorithm using FFT

```
for each fft_bin do
     chroma_bin = mod(round(12*log2(freq(fft_bin)/freq(C4)) ),12)
     chroma_energy[chroma_bin] += fft_energy[fft_bin]
end for
     freq(fft_bin): returns the frequency of fft_bin in Hz.
     freq(C4): 261.5 Hz
```

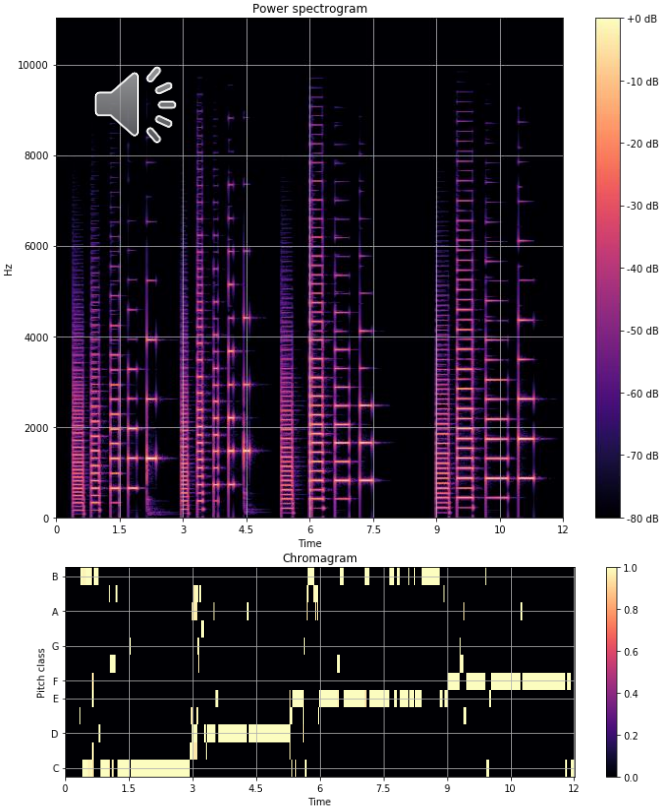
e.g.

C4	12*log2(261.5/261.5)=0
C#4/Db4	12*log2(277.18/261.5)=1
D4	12*log2(293.66/261.5)=2

- Often also `chroma_energy` is normalized to retain only pitch info and remove signal level information
- The frequency resolution of lower bins and non-tonal energy cause issues, which refined algorithms mitigate
- Tools: Librosa offers tools to compute chromagram

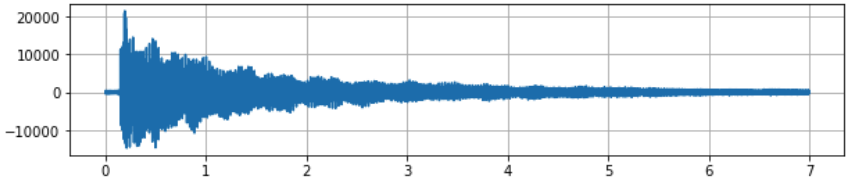
Example: Piano notes in different octaves

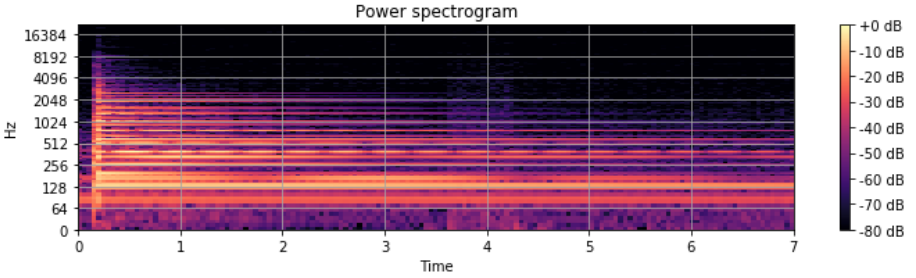
- Notes C0-C3, D0-D3,E0-E3,F0-F3 are played in a sequence
- Top figure: power spectrogram of the signal.
  - Note the octave raise.
- Chromagram shows the same pitch class value for the different octave notes
  - Bottom figure evaluated from data using Python/librosa.

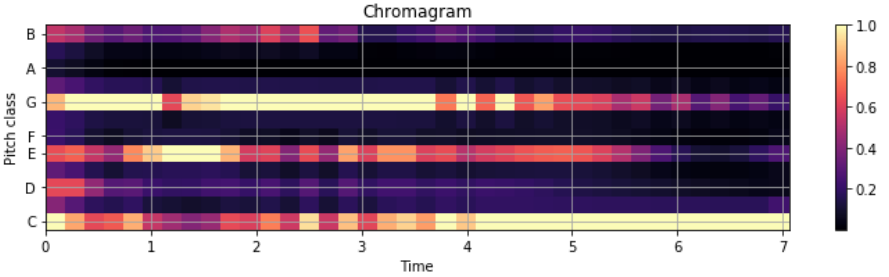


Example: Chromagram of a chord

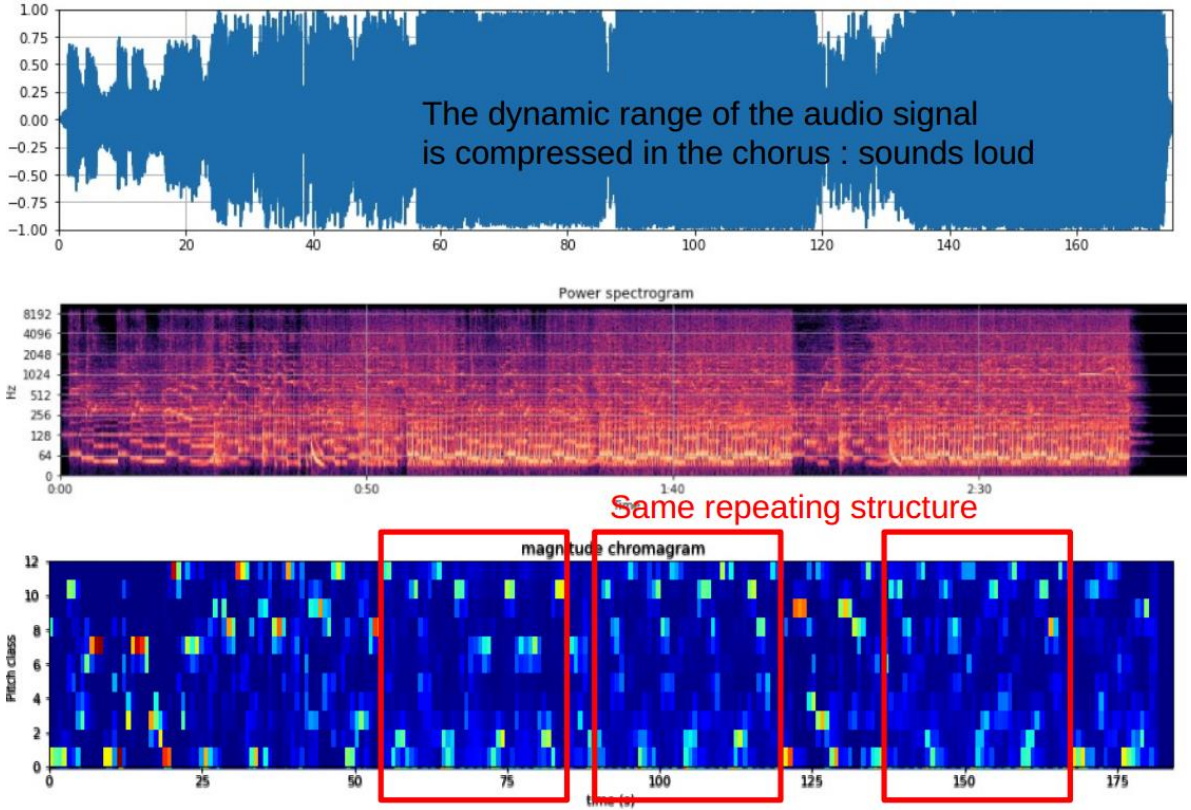
- C-major (G,E,C), visible in chromagram







Example: Finnish Eurovision 2018 candidate song Monsters by Saara Aalto



Music analysis

Structure detection

- Self-similarity of temporal segments can be examined using a similarity matrix, with cells representing similarity of two feature vectors $\boldsymbol{x}_n$, $\boldsymbol{x}_m$ from time instantants *n,m*
  - e.g. cosine similarity:

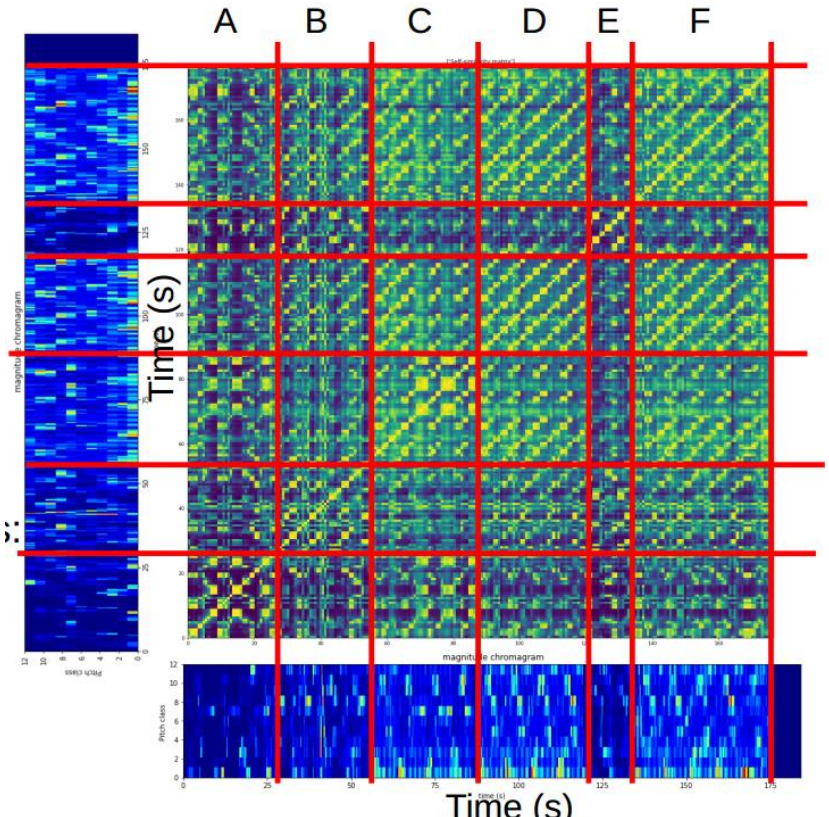
$$s(\boldsymbol{x}_n, \boldsymbol{x}_m) = \frac{\langle \boldsymbol{x}_n, \boldsymbol{x}_m \rangle}{\|\boldsymbol{x}_n\| \cdot \|\boldsymbol{x}_m\|}$$

  - Idea: similar vectors $\boldsymbol{x}_n$, $\boldsymbol{x}_m$ should lead to high similarity value.

- Recurring patterns become visible as large values
  - Homogeneity results in a block structure (features capture musical sequences that stay somewhat constant over the part)
  - Two repeating harmonic progressions or melodies highly similar to each other create diagonal lines in the self-similarity matrix.
- Some post-processing is typically performed to enhance these structures for automatic extraction.

Structure with self-similarity matrix

- Similarity measure: cosine.
- Features: chroma.
- Each cell is the cosine-distance between the chroma vectors in x,y coordinate time positions of same signal.
- Repeated patterns are similar (high cosine similarity) and show as diagonal strips
- D & F are most similar:
  - chorus (song+accompaniment)
- C is similar to D and F, visible as elevated value block structure, less diagonal stripes:
  - It is the chorus without singer
- B has diagonal stripes as D and F.
  - Chorus with singing and only some instruments present)



Self-similarity matrix

- Self-similarity matrix be used to analyze the structure of a song

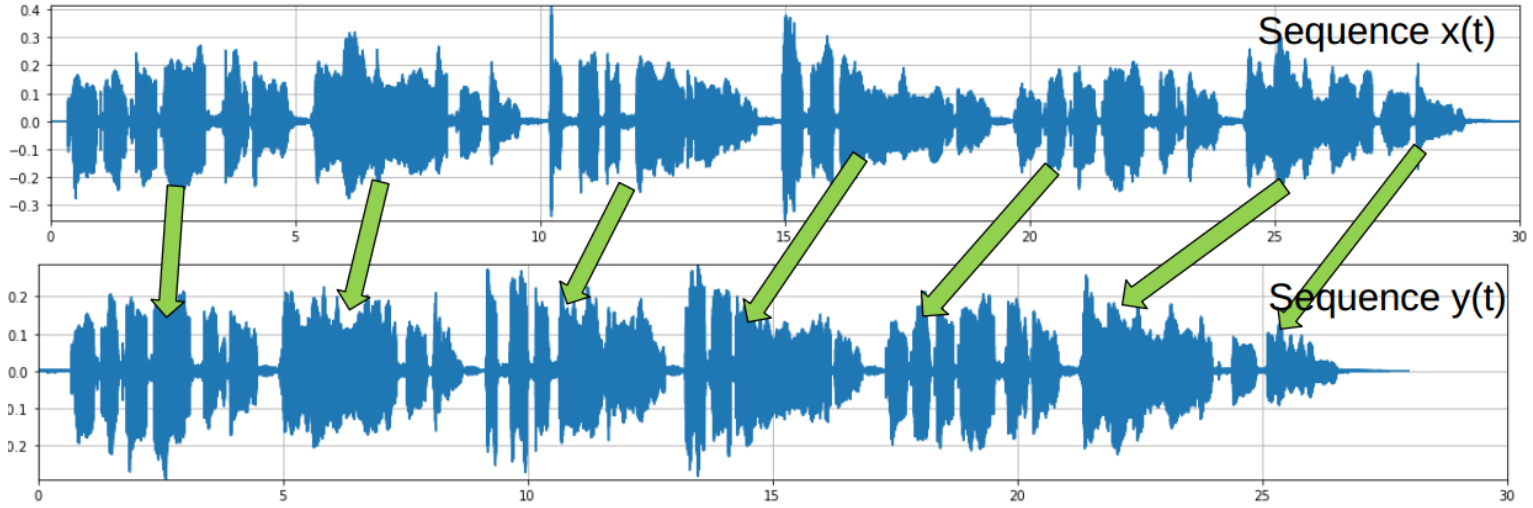
- Applications
  - Automatic segmentation
    - Here the lines between song parts were drawn manually, but this can be automatically done
  - Audio thumbnailing:
    - Extracting most representative part of a song

- Tools: Music Structure Analysis Framework

  https://github.com/urinieto/msaf (contains automatic segmentation)

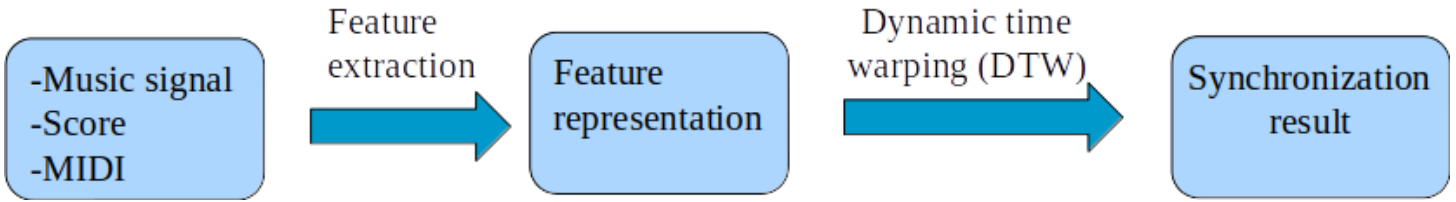
Music synchronization

- Synchronization here refers to a procedure to determine the corresponding position in a piece of music, given another (slightly differently interpreted) piece of music
- Figure: Two different tempo performances of a same song



Music synchronization

- Procedure for music synchronization



- Uses
  - Allows synchronization of score of music with audio track
  - Jump to audio position, based on selected score position
  - Seamless switching between different versions of same music
  - Measure differences in tempo between same musical piece versions
    - To analyze the performance

Music synchronization

- Need right types of features that
  - Retain relevant information, while suppressing irrelevant details
  - Capture similarity between notes and chords
  - Allow variation between instrumentation (timbre), tempo, rhythm
  - Suitable features include chroma vectors

- Dynamic time warping (DTW)
  - Used to align the two feature representation sequences (of feature vectors) of two different versions of the same piece of music
  - DTW Goal: find an optimal alignment between two sequences
  - DTW is used in data mining, information retrieval, and bioinformatics

Dynamic time warping

- Input: two sequences $\boldsymbol{X}= [\boldsymbol{x}_1, .. \boldsymbol{x}_N]$ and $\boldsymbol{Y}=[\boldsymbol{y}_1,..., \boldsymbol{y}_M]$
- Define a real-valued cost matrix $C(n,m) \equiv c(\boldsymbol{x}_n, \boldsymbol{y}_m)$ $n \in [1,N]$ $m \in [1,M]$

  $c(\boldsymbol{x}_n, \boldsymbol{y}_m)$ is a cost function:
  - Cosine distance $c(\boldsymbol{x}_n, \boldsymbol{y}_m) = 1 - \frac{\langle \boldsymbol{x}_n, \boldsymbol{y}_m \rangle}{\|\boldsymbol{x}_n\| \cdot \|\boldsymbol{y}_m\|}$
  - Euclidean distance $c(\boldsymbol{x}_n, \boldsymbol{y}_m) = \|\boldsymbol{x}_n - \boldsymbol{y}_m\|$
  - Generally: if $\boldsymbol{x}_n$ is similar to $\boldsymbol{y}_n$ then the cost should be low
  - A warping path P = $(\boldsymbol{p}_1, .. \boldsymbol{p}_L)$ of length L,
    - $\boldsymbol{p}_l=(n_l, m_l)$ denotes correspondence between the two sequences
    - Starts from $\boldsymbol{p}_1$=(1,1) and ends in $\boldsymbol{p}_1$=(N,M)
    - Steps are monotonically increasing (no going back in time)

      $n_1 \leq n_2 \leq \cdots \leq n_L$, and $m_1 \leq m_2 \leq \cdots \leq m_L$

Dynamic time warping

- Step-size condition

$$\boldsymbol{p}_{l+1} - \boldsymbol{p}_l \in \{(1,0), (0,1), (1,1)\}$$

- Total cost of path:

$$c(X,Y) = \sum_{l=1}^{L} C(n_l, m_l)$$

Optimal warping path P* has minimal total cost of all path

Algorithm: DTW

Input: Cost matrix $\mathbf{C}$ of size $N \times M$
Output: Accumulated cost matrix $\mathbf{D}$
Optimal warping path P^*

Procedure: Initialize $(N \times M)$ matrix $\mathbf{D}$ by $\mathbf{D}(n,1) = \sum_{k=1}^{n} \mathbf{C}(k,1)$ for $n \in [1:N]$ and $\mathbf{D}(1,m) = \sum_{k=1}^{m} \mathbf{C}(1,k)$ for $m \in [1:M]$. Then compute in a nested loop for $n = 2,\ldots,N$ and $m = 2,\ldots,M$:

$$\mathbf{D}(n,m) = \mathbf{C}(n,m) + \min\{\mathbf{D}(n-1,m-1), \mathbf{D}(n-1,m), \mathbf{D}(n,m-1)\}.$$

Set $\ell = 1$ and $q_\ell = (N,M)$. Then repeat the following steps until $q_\ell = (1,1)$:

Increase ℓ by one and let $(n,m) = q_{\ell-1}$.
If $n = 1$, then $q_\ell = (1,m-1)$,
else if $m = 1$, then $q_\ell = (n-1,m)$,
else $q_\ell = \text{argmin}\{\mathbf{D}(n-1,m-1), \mathbf{D}(n-1,m), \mathbf{D}(n,m-1)\}$.
(If 'argmin' is not unique, take lexicographically smallest cell.)

Set $L = \ell$ and return $P^* = (q_L, q_{L-1}, \ldots, q_1)$ as well as $\mathbf{D}$.

Müller,2015

Example

- X=[1,1,0,2,2], N=5, and Y=[1,0,2], M=3
- Cost function $c(n,m)=|x_n-y_m|$, i.e. absolute distance (L1)

2	1	2	0
2	1	2	0
0	1	0	2
1	0	1	1
1	0	1	1
	1	0	2

2	2	3	0
2	1	2	0
0	1	0	2
1	0	1	2
1	0	1	2
	1	0	2

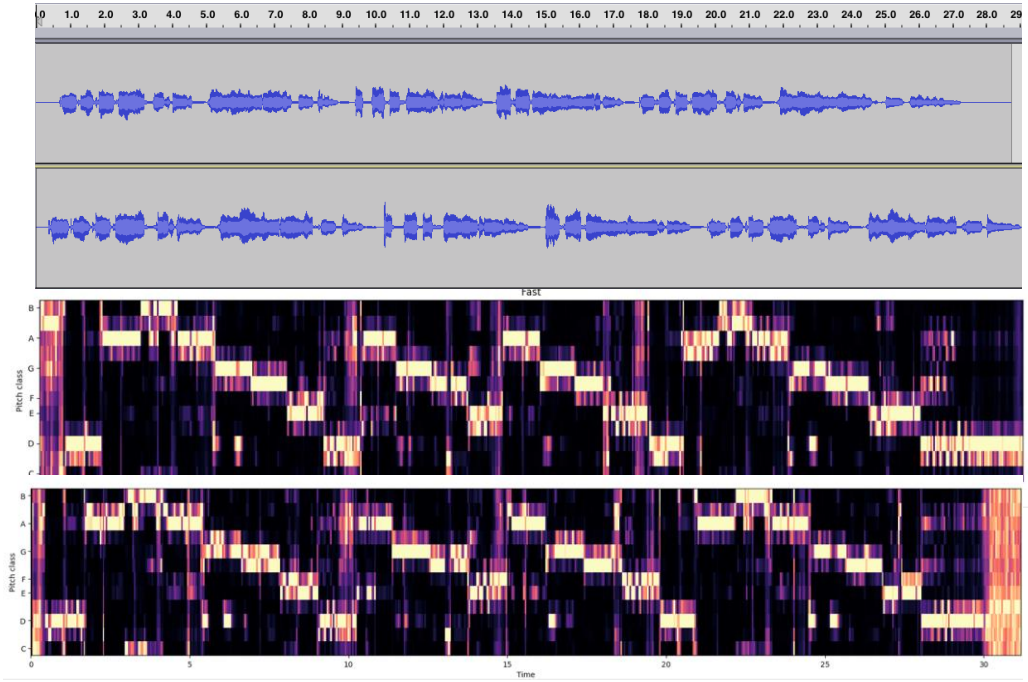
Optimal path:

P* = {(3,5),(3,4),(2,3),(1,2),(1,1)}, L=5

Cost of path: 0

(sum the optimal path cell values)

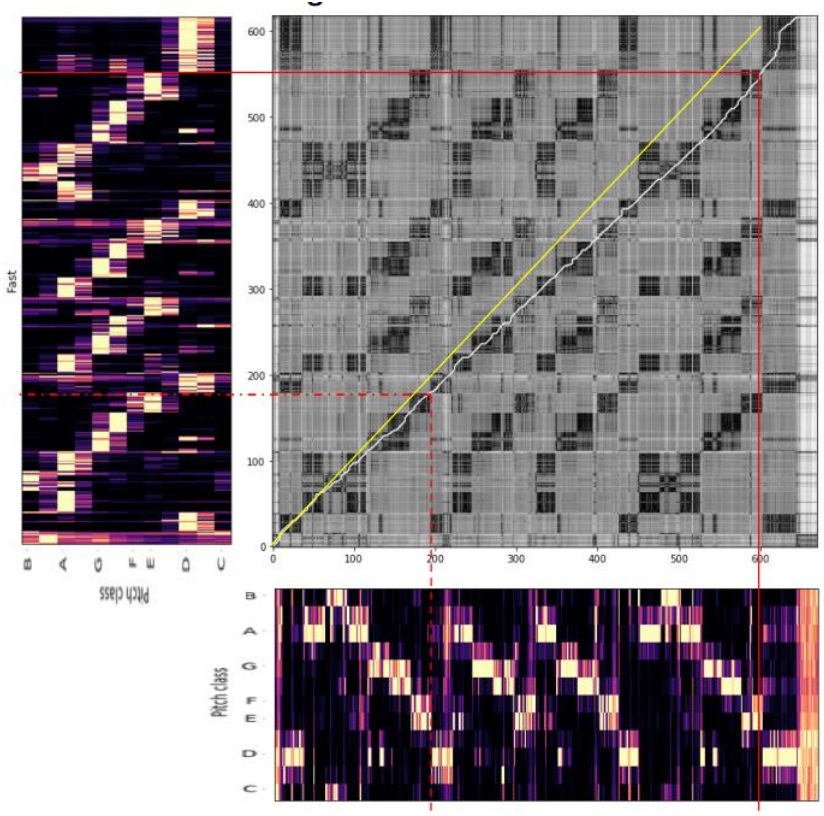
Example: Music synchronization



- A song is sung twice, upper one is 7% faster
- Bottom figure: Chroma features (using librosa)

Cost matrix and optimal path

- Distance: Euclidean distance between chromagram vectors at each time instant
- Red-lines: Two synchronized notes
- Yellow line: Linear time
- White line: Optimal path
- Optimal path shows the corresponding time instant for both songs.



DTW summary

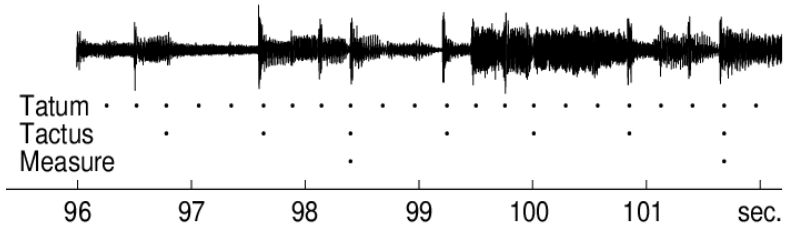
- Gives the synchronization of feature representations of two signals.
- Cost functions
  - Cosine distance: not sensitive to absolute level (since the vectors are normalized by definition)
  - Euclidean distance. Is sensitive to level differences if features are not normalized.
  - Feature normalization: by normalizing the feature vector of each time-value, the feature is not sensitive to level anymore.
- Tools: dtw available in librosa

Break

Tempo, beat, meter

Temporal structure

- Music is organized into temporal units, called beats
  - Rhythm refers to pattern of strong and weak beats
  - Musical meter refers to the rhythm aspect of music
- Most noticeable metrical level is the tactus
  - Sometimes called “the foot tapping rate” or just “beat”
  - Beat is the backbone of a piece of music, which makes the automatic detection of beat an important sub-task in music processing.
  - Duration of a beat is given by tempo, indicated in beats per minute (BPM).



- Above the level of chords and notes, are phrases and sections
  - Coherent sequences of chords/notes that often repeat with no or little variation, e.g. in pop songs: intro-verse-chorus-verse-chorus…

Beat tracking

- Tempo and beat are fundamental aspects of music
- Beat position approximations:
  - Often occurs at note onset position
  - Intervals are approximately constant

Typical onset detection flow

- Onset detection: detect note start time
  - For single notes, simply looking at the signal energy is a start
  - However, simultaneous occurrence of events makes this approach not so robust, therefore time-frequency domain is more suitable
  - Often, temporally simultaneous events do not overlap in frequency.

Beat tracking

The spectral based novelty function for beat tracking:

1. Take STFT *X(n,k)* of signal *x(t)*

   n=time frame, *k*=frequency index, *t*=sample time index

1. Apply log-compression to magnitude spectrum:

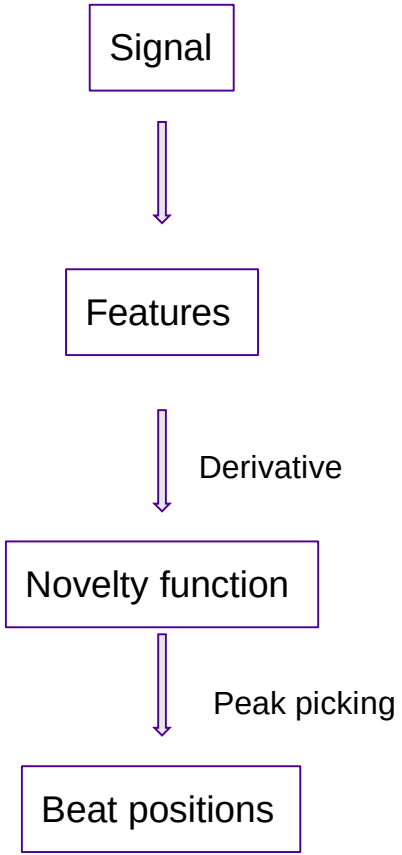
$$Y(n,k) = \log(1 + \gamma|X(n,k)|)$$

   - where γ = 1 weak components become visible
   - γ >> 1: may amplify also noise-like weak components

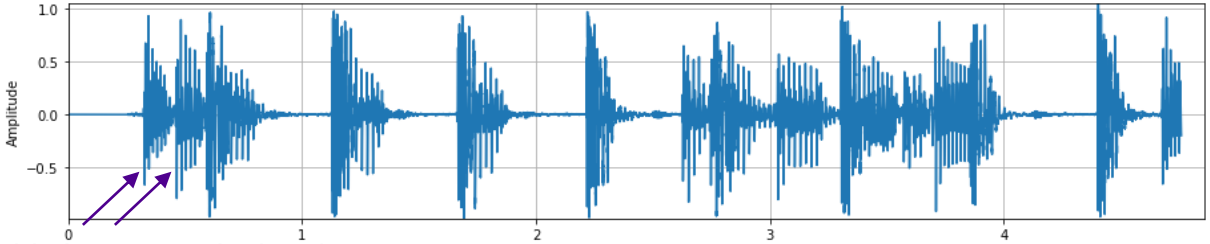
1. Calculate the spectral novelty function:

$$\Delta(n) = \sum_{k=0}^{K-1}|Y(n+1,k) - Y(n,k)|$$

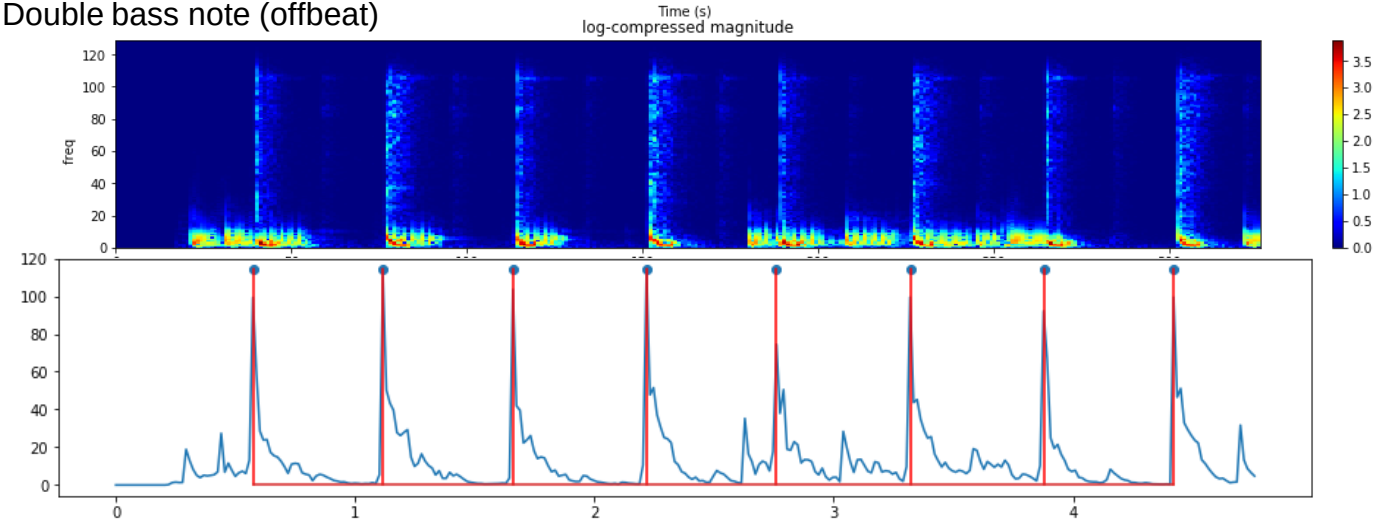
1. Pick peak locations of Δ(n) and convert STFT time frame index n into seconds.



Example: Queen Another one bites the dust

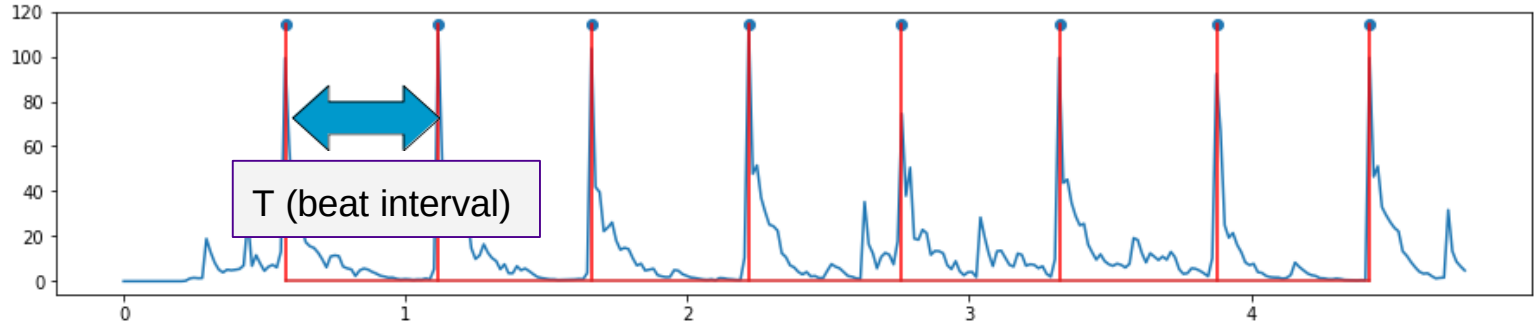


Double bass note (offbeat)



Tempo analysis

- Tempo = how many times per minute the beat repeats?



- Tempo = 60 / T [BPM],
  - where T is “beat interval time”, BPM is beats per minute
- Tempo is extracted by analyzing the periodic behavior of the novelty function.
  - Tempo can change during song, so analysis is done in segments of 4-12 s
  - Practically we can limit tempo between [30,600] BPM

Summary

- Basic understanding of music signals
  - Notes, chords, the score, melody, temporal structure

- Music signal processing: Tools for extracting information from audio
  - (CQT, a musically and perceptually motivated transform
  - Chromagram – a musically meaningful representation
  - Dynamic time warping – music synchronization
  - Beat, tempo, structure analysis