MIR PROGRAMMING WITH MININGSUITE

AIST2010 Lecture 4P
MININGSUITE WITH MATLAB

MIRtoolbox is widely used as a set of tools in MATLAB for MIR

- Development started in 2009, led by Olivier Lartillot
- “It has evolved quite significantly this last decade, downloaded tens of thousands of times, and cited in more than one thousand scientific papers.”
- “MIRtoolbox 2.0”: MiningSuite (by the same author)
  - Open-source
  - Supporting both audio and symbolic (MIDI) representations
MININGSUITE WITH MATLAB

Official website
- [http://olivierlar.github.io/miningsuite](http://olivierlar.github.io/miningsuite)
- Version 0.10 (first beta)
- May not be good enough for research use, but good enough for studies
  - Some features are not complete yet
- To be put into MATLAB path

Modules
- SigMinr: signal processing
- AudMinr: auditory modelling
- MusMinr: music analysis
- VidMinr: video analysis
- PhyMinr: physics and motion analysis
- SeqMinr: sequence processing
- PatMinr: pattern mining
Loading a waveform

- `x = sig.signal('test.wav')` % audio: wav/mp3/..., text: csv/txt/...
- A plot is made if output is not suppressed by `;
- Also a number of parameters for transformation of the audio, or extracting only partial data (channel, frequency band, time, etc.) from the audio
  - [https://github.com/olivierlar/miningsuite/wiki/sig.signal](https://github.com/olivierlar/miningsuite/wiki/sig.signal)

Data hidden in the return object

- `d1 = sig.getdata(x);` % `d` is the waveform samples
- `sr = get(x, 'Srate');` % sampling rate
Displaying the spectrum

- `y = sig.spectrum('test.wav')` % or `sig.spectrum(x), x → signal`
- `sig.spectrum('test.wav', 'Frame')` % spectrogram of 50ms, 50% overlap
- Frequency resolution can be specified
  - `https://github.com/olivierlor/miningsuite/wiki/sig.spectrum`
  - `https://github.com/olivierlor/miningsuite/wiki/sig.frame-and-'Frame'-option`

Data hidden in the return object

- `d2 = sig.getdata(y);` % `d` is the FFT points
- `ph = get(y, 'Phase').view;` % phase of every FFT point
  - Besides `.view`, there are also methods like `.max, .mean, .size`, etc., see `doc sig.data`
DETAILS OF THE SIGNAL

Amplitude related
- RMS amplitude: `sig.rms('test.wav')`
- Amplitude envelope: `sig.envelope('test.wav')`
- Autocorrelation: `sig.autocor(...)`
- Self-similarity: `sig.simatrix(...)`

Statistics from spectrogram
- % where `s=sig.spectrum(...)`
- Flux: `sig.flux(s)`
- Average: `sig.mean(s)`
- Standard deviation: `sig.std(s)`
- Median, histogram, centroid, spread, skewness, entropy, ...
Finding peaks

- \( p = \text{sig.peaks}(y) \) % all local maxima
- \( p = \text{sig.peaks}(y, 'Total',1) \) % only highest peak
- \( \text{get}(p, 'PeakPos').\text{view} \) % output positions
- \( \text{get}(p, 'PeakVal').\text{view} \) % output the value at peak

Segmentation based on events

- \( o = \text{sig.events('test.wav', method);} \); \( \text{sig.segment('test.wav', o)} \)
  - \textit{method} can be 'Envelope', 'SpectralFlux', etc.


**AUDIO ANALYSIS**

Redistributing frequencies
- % similar to sig.spectrum() with a different frequency axis
- aud.spectrum(..., 'Mel')
- aud.spectrum(..., 'Bark')

Spectral shape in Mel-scale
- aud.mfcc(..., 'Frame') % plot mel-spectrum and MFCCs

Pitch estimation
- pit = aud.pitch('test.wav'); pitmat=cell2mat(getdata(pit))
- pitamp = get(pit, 'Amplitude') % amp of each pitch component
TIME OF THE SOUNDS

Temporal shape
- `aud.attacktime()`
- `aud.attackslope()`
- `aud.attackleap()`
- `aud.decaytime()`
- `aud.decayslope()`
- `aud.decayleap()`

Tempo estimation (beat per minute)
- `aud.tempo()` % can be done in 'Autocor', 'Spectrum' or both
- `aud.tempo(..., 'Frame')` % tempo change in time
Available for both .wav/.mp3/... and .mid formats

- mus.score(...) % a plot of pitch vs. time
- mus.chromagram(...) % a plot of pitch distribution
- mus.chromagram(..., 'Frame') % a plot of chromagram with time
- mus.key(...) % a plot of possible key centre and guess of key
- mus.majorness(...) % is this major or minor?

Many more in development

- Try them: https://github.com/olivierlar/miningsuite/wiki/Guided-Tour:-Music-analysis
Among the three modules

Basic features
- Statistics and data getting methods are in the sig. module

Specialization
- For example, sig.spectrum() provides general mechanisms, while aud.spectrum() adds support of various auditory models and mus.spectrum() adds support of music-related operations

Refer to the overview of operators:
READ FURTHER

MiningSuite Guided Tour

▪️[https://github.com/olivierlar/miningsuite/wiki/Guided-Tour](https://github.com/olivierlar/miningsuite/wiki/Guided-Tour)

MIRtoolbox 1.7 User’s Manual