

Python in MIR

LibROSA + mir_eval



Laboratory for the Recognition and
Organization of Speech and Audio



the center for jazz studies
at columbia university

Brian McFee
HAMR.2014

Python in MIR... why?

- Better coding practices, reproducible research
- Critical mass of...
 - existing modules and API support
 - developer ecosystem
- Integration with modern tools
 - numpy, scipy
 - IPython (+notebook)
 - scikit-learn
 - theano

Python in MIR... why not before?

- Entrenched legacy MATLAB code
- (previously) Lack of DSP/MIR tools in Python
- Wrappers exist (Marsyas, YAAFE, ...)
 - but these can be unwieldy, difficult to modify

LibROSA

<https://github.com/bmcfee/librosa>

- 100% Python
 - Minimal dependencies
 - Thoroughly documented
 - Strict unit tests on core functions
 - Easy to read **and modify**
- Easy to use
- Easy to install via PyPI:
 - `$ pip install librosa`

librosa 0.1.0 (June 2013, **HAMR**)

- Basic audio processing (IO, STFT)
- Feature extraction
- Rhythm analysis
- Harmonic-percussive source separation

librosa 0.2.0 (December 2013)

- New modules:
 - `display`: visualization
 - `segment`: structural segmentation
 - `onset`: onset detection
- Additional features
 - `cqt`, `pseudo-cqt`, utility functions, code refactoring...
 - examples and demo code
 - The list goes on, see [CHANGELOG](#)
- Improved documentation

librosa 0.2.1 (January 2014, NEMISIG)

- Efficiency improvements and bug fixes
- Parameter optimization
- Improved annotation export
- Improved evaluation interoperability

Open evaluation

- What if I want to run my own MIREX?
- MIR evaluations are notoriously difficult
- MIREX evaluation code is complex, huge dependency chain

mir_eval

https://github.com/craffel/mir_eval

- A python implementation of MIR evaluations
 - beat tracking [Beat Evaluation Toolbox]
 - onset detection [Böck's onset evaluator]
 - segmentation
 - chord recognition [Harte/McVicar]
 - blind source separation [BSS_Eval]
- And helpers...
 - data processing, alignment, chord reduction, ...

mir_eval v0.0.1 coming soon

- Pure python
- Fully documented
- Minimal dependency chain
- Easy to use
- Unit tests for numerical equivalence to *MIREX*

DEMO