# mir_eval

Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis

Panel on Objective Evaluation in Semantic Audio Analysis and Processing, 138th AES Convention, May 8th 2015

Lab ROSA

MARL
NYU Music and Audio
Research Laboratory

# Evaluation Needs

Everyone should use the same code for evaluating their algorithms.

# Evaluation Needs

Everyone should use the same code for evaluating their algorithms. In reality, researchers often write their own code. Why?

# Evaluation Needs

Everyone should use the same code for evaluating their algorithms. In reality, researchers often write their own code. Why?

- NEMA/MIREX codebase is powerful, but too many moving parts

# Evaluation Needs

Everyone should use the same code for evaluating their algorithms. In reality, researchers often write their own code. Why?

- NEMA/MIREX codebase is powerful, but too many moving parts
- Language preferences

# Evaluation Needs

Everyone should use the same code for evaluating their algorithms. In reality, researchers often write their own code. Why?

- NEMA/MIREX codebase is powerful, but too many moving parts
- Language preferences
- Ease of integration

# Evaluation Needs

Everyone should use the same code for evaluating their algorithms. In reality, researchers often write their own code. Why?

- NEMA/MIREX codebase is powerful, but too many moving parts
- Language preferences
- Ease of integration
- Want to understand the metrics

# Evaluation Needs

Everyone should use the same code for evaluating their algorithms. In reality, researchers often write their own code. Why?

- NEMA/MIREX codebase is powerful, but too many moving parts
- Language preferences
- Ease of integration
- Want to understand the metrics

Our solution: `mir_eval`

# `mir_eval`'s Goals

- **Standardized** - `mir_eval` should implement evaluation metrics as agreed upon by the community, rather than a single researcher.

# mir_eval's Goals

- **Standardized** - mir_eval should implement evaluation metrics as agreed upon by the community, rather than a single researcher.
- **Transparent** - The implementations in mir_eval should make it very clear why the metrics were implemented the way they were. Code should be readable and well-documented.

# mir_eval's Goals

- **Standardized** - mir_eval should implement evaluation metrics as agreed upon by the community, rather than a single researcher.
- **Transparent** - The implementations in mir_eval should make it very clear why the metrics were implemented the way they were. Code should be readable and well-documented.
- **Easy-to-use** - Using mir_eval should be easy whether you're familiar with Python or not, and should have minimal "start-up cost".

# Why Standardization Matters

Compared to NEMA/MIREX:

| Beat Detection | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F-measure | Cemgil | Goto | P-score | CMLc | CMLt | AMLc | AMLt | In. Gain |
| 0.703% | 0.035% | 0.054% | 0.877% | 0.161% | 0.143% | 0.137% | 0.139% | 9.174% |

| Structural Segmentation | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| NCE-Over | NCE-under | Pairwise F | Pairwise P | Pairwise R | Rand | F@.5 | P@.5 | R@.5 |
| 3.182% | 11.082% | 0.937% | 0.942% | 0.785% | 0.291% | 0.429% | 0.088% | 1.021% |

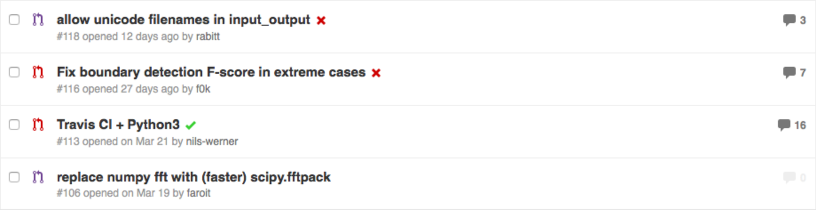
| Structural Segmentation (continued) | | | | | Onset Detection | | |
|---|---|---|---|---|---|---|---|
| F@3 | P@3 | R@3 | Ref-est dev. | Est-ref dev. | F-measure | Precision | Recall |
| 0.393% | 0.094% | 0.954% | 0.935% | 0.000% | 0.165% | 0.165% | 0.165% |

| Chord Estimation | | | | | Melody Extraction | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Root | Maj/min | Maj/min + Inv | 7ths | 7ths + Inv | Overall | Raw pitch | Chroma | Voicing R | Voicing FA |
| 0.007% | 0.163% | 1.005% | 0.483% | 0.899% | 0.070% | 0.087% | 0.114% | 0.000% | 10.095% |

Differences explained in ISMIR 2014 paper, “`mir_eval`: A Transparent Implementation of Common MIR Metrics”

# Community Development

Community involvement through issue tracking and pull requests:



□ allow unicode filenames in input_output ✗ 3
#118 opened 12 days ago by rabitt

□ Fix boundary detection F-score in extreme cases ✗ 7
#116 opened 27 days ago by f0k

□ Travis CI + Python3 ✓ 16
#113 opened on Mar 21 by nils-werner

□ replace numpy fft with (faster) scipy.fftpack 0
#106 opened on Mar 19 by faroit

`http://github.com/craffel/mir_eval`

# Using mir_eval

In Python:

```python
import mir_eval
# Load in beat annotations
reference_beats = mir_eval.io.load_events('ref_beats.txt')
estimated_beats = mir_eval.io.load_events('est_beats.txt')
# scores will be a dictionary where the key is the metric name
# and the value is the score achieved
scores = mir_eval.beat.evaluate(reference_beats, estimated_beats)
```

# Using mir_eval

In Python:

```
import mir_eval
# Load in beat annotations
reference_beats = mir_eval.io.load_events('ref_beats.txt')
estimated_beats = mir_eval.io.load_events('est_beats.txt')
# scores will be a dictionary where the key is the metric name
# and the value is the score achieved
scores = mir_eval.beat.evaluate(reference_beats, estimated_beats)
```

Using the evaluator scripts:

```
> ./beat_eval.py ref_beats.txt est_beats.txt -o scores.json
> cat scores.json
  {"F-measure": 0.6216216216216,
   "Cemgil": 0.36267669947376,
   "Cemgil Best Metric Level": ...
```

# Using mir_eval

In Python:

```python
import mir_eval
# Load in beat annotations
reference_beats = mir_eval.io.load_events(’ref_beats.txt’)
estimated_beats = mir_eval.io.load_events(’est_beats.txt’)
# scores will be a dictionary where the key is the metric name
# and the value is the score achieved
scores = mir_eval.beat.evaluate(reference_beats, estimated_beats)
```

Using the evaluator scripts:

```
> ./beat_eval.py ref_beats.txt est_beats.txt -o scores.json
> cat scores.json
  {"F-measure": 0.6216216216216,
   "Cemgil": 0.36267669947376,
   "Cemgil Best Metric Level": ...
```

Using our web API:

http://labrosa.ee.columbia.edu/mir_eval

# Where to find us

Code:

`http://github.com/craffel/mir_eval`

Documentation:

`http://craffel.github.io/mir_eval`

Web API:

`http://labrosa.ee.columbia.edu/mir_eval`

Paper:

C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "`mir_eval`: A Transparent Implementation of Common MIR Metrics", *Proceedings of the 15th International Conference on Music Information Retrieval*, 2014.