Version: 0.1.0
Date: 2025-04-07

TONEARCLIB: A Semantic Music Engine for Cross-Domain Emotional Interpretation and Structural Analysis

Abstract

Despite significant advances in digital signal processing, machine learning, and music theory modeling, no existing system bridges the gap between technical musical structure and semantic emotional meaning in a universally accessible format. Most audio tools treat music as either signal (for engineers) or experience (for listeners), lacking an intermediate semantic layer that makes music interpretable by artificial intelligence systems or reusable across creative and analytical domains.

TONEARCLIB addresses this gap by providing a modular, extensible Python library that performs symbolic and emotional interpretation of audio and MIDI data. It transforms raw musical input into structured emotional profiles, motif mappings, temporal arcs, and mood tags—creating a narrative representation of music that can be understood and utilized by both humans and LLM-based systems. The framework bridges composition, analysis, and reactive sound design by exposing a semantic layer between measurable audio features and emotionally resonant output.

---

1. Introduction

1.1 Problem Space

Music, unlike static text or visual data, evolves over time and conveys meaning through structure, tone, and progression. While digital audio analysis can extract features like RMS, spectral centroid, MFCCs, and tempo, these values have little semantic weight without contextual interpretation. Human composers instinctively craft emotional arcs, but modern AI and algorithmic systems cannot replicate this process meaningfully without symbolic representation.

1.2 Project Vision

TONEARCLIB serves as a translation engine that interprets musical compositions as emotional narratives. It enables intelligent systems to understand, respond to, and even generate music in a manner that is

emotionally aware and structurally informed. Whether used in games, therapy, scoring tools, or AI augmentation, it provides the missing layer of meaning.

---

2. System Architecture

2.1 Input Modalities

- **Waveform Audio (WAV/FLAC)**
- **MIDI Files**
- **Optional JSON Metadata for narrative tagging**

2.2 Feature Extraction Pipeline

- RMS and dynamic range calculation
- Spectral centroid, rolloff, flatness, and entropy
- Beat and onset tracking
- Chroma and tonnetz analysis for key/mode estimation
- KMeans-based temporal segmentation

2.3 Symbolic Interpretation Layer

- Mood classification (e.g., "Dark / Still / Tonal")
- Emotional arc mapping (intro, climax, decay)
- Motif tracking via chroma recurrence
- Temporal centroid calculation for narrative pacing

2.4 Output Schema

- `track_profile.json` with structured metadata:
  - Key, tempo, mode
  - Dominant mood and energy profile
  - Section markers with emotional tags
  - Peak and decay moments
  - Motif identifiers for cross-reference

---

3. Core Applications

3.1 Narrative Sound Engine (Games & TTRPGs)

TONEARCLIB allows music playback to dynamically shift based on in-game emotional tone, faction hostility, or event type. It connects map tags, quest states, and AI logs to soundtrack selection.

3.2 Generative Composition Feedback

In a generative loop, TONEARCLIB can analyze music output from AI or human sources and suggest revisions:

> "Your track builds emotionally but lacks motif recall. Consider reintroducing the intro figure in the final section."

3.3 Music Therapy & Mood Profiling

Analyzing patient-created playlists or compositions to: - Track mood cycles - Identify sonic preferences - Detect depressive/synthetic patterns via spectral signatures

3.4 AI-Language Model Augmentation

TONEARCLIB serves as an auditory context interpreter, enabling LLMs to: - Comment on mood alignment between narrative and soundtrack - Reason about musical callbacks in storytelling - Identify emotional mismatches in scene scoring

3.5 Educational Analysis

Students can visualize: - How emotional pacing changes over time - Where motifs recur structurally - Why certain instrumentation implies certain moods

---

4. Technical Specifications

4.1 Language and Dependencies

- Python 3.10+
- Libraries: `librosa`, `numpy`, `pandas`, `scikit-learn`, `matplotlib`, `pydantic`

4.2 Data Format Standards

- Inputs: `.wav`, `.mid`, `.json`
- Outputs: `track_profile.json`, `emotional_summary.txt`, optional `.pdf` report

4.3 Modular Design

The system is organized into five distinct modules, each with a specific role:

- **`features.py`** – Extracts all raw and derived musical features (RMS, chroma, spectral metrics, etc.)

- **`models.py`** – Defines Pydantic-based data classes for track profiles, motifs, and segment schema
- **`moodmap.py`** – Converts extracted features into symbolic mood labels and emotion arcs
- **`reporting.py`** – Generates visual and textual output summaries (JSON, TXT, PNG, etc.)
- **`analyzer.py`** – Core orchestration logic for emotional inference and structural evaluation

---

5. Future Development

5.1 Real-Time Analysis Hooks

Support for streamed or DAW-synced input (via plugin or pipe).

5.2 Visual Emotion Arc Editor

A GUI tool for tagging emotional intent manually and comparing to detected structure.

5.3 Multi-track Cross-Referencing

Detect and track motif reuse across entire albums or campaigns.

5.4 Integration Plugins

DAWs, game engines (Unity, Godot), or middleware (FMOD, Wwise) could consume TONEARCLIB output to make music reactive in real-time.

---

6. Conclusion

TONEARCLIB transforms music from an opaque sonic object into a symbolically rich, emotionally interpretable system. It empowers machines to understand music more like humans do, and enables humans to design, tag, and analyze music with structural and emotional clarity. By bridging the divide between data and meaning, TONEARCLIB becomes not just a tool for analysis, but a universal language layer between music, machine, and meaning.

---

7. Licensing & Distribution

- Custom non-commercial license (see LICENSE file)
- Free for personal, educational, and research use

- Commercial use requires explicit permission from the author
- Designed for integration in both commercial and research settings
- 100% offline-capable

Appendix A: Example Output

```
json { "track_title": "Reliquary", "tempo": 64, "key": "D minor",
"mode": "Aeolian", "mood": "Dark / Still / Tonal", "structure": [
{"section": "Intro", "emotion": "mourning", "start_sec": 0},
{"section": "Climax", "emotion": "reverence", "start_sec": 82},
{"section": "Outro", "emotion": "silence", "start_sec": 140} ],
"motif_profile": ["pad_low", "cello_ritual"],
"narrative_summary": "A slowly unfolding, grief-stricken piece
resolving into still reverence." }
```