

INTRODUCTION – EARSKETCH AND THE CAI ANALYSIS MODULE

- EarSketch is an online coding and music education platform where students learn to create music in a Digital Audio Workstation through coding in a Code Editor.
- This paper presents the initial stages of the CAI (Co-creative Artificial Intelligence) System, a creativity-support tool for EarSketch.
- CAI is designed to support student learning through co-creative musical output – assisting students in learning EarSketch’s computing and music pedagogy by acting as a collaborative partner.
- The CAI system is in progress and will include multimodal analysis tools, an interactive dialogue system, and suggestion generation system. This paper discusses the analysis tools, which we call the Analysis Module.

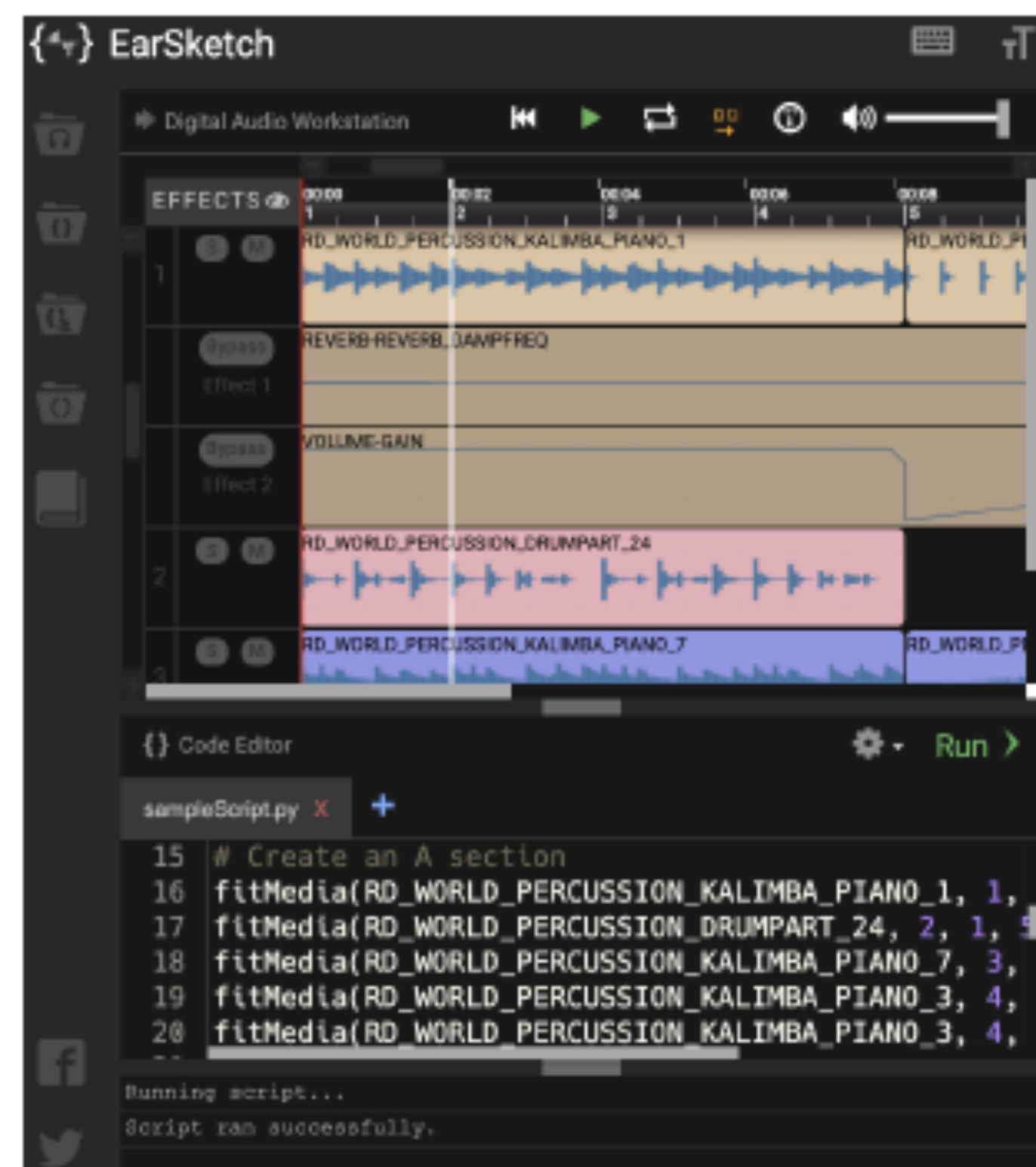


Figure 1. A Screenshot of the EarSketch web-based application, containing the Digital Audio Workstation (top), and Code Editor (bottom).

SYMBOLIC MUSIC ANALYSIS

- Projects in EarSketch take the form of lines of code that place sound and effect variables on a track listing, to form a sample-based composition.
- The CAI Analysis Module represents these projects to a measure-by-measure view of sounds, effects, and how effect parameters change over time.

```
#Setup
from earsketch import *
init()
setTempo(120)

# Create an A section
fitMedia(RD_WORLD_PERCUSSION_KALIMBA_PIANO_1,1,1,5) # main
fitMedia(RD_WORLD_PERCUSSION_DRUMPART_24,2,1,5) # drums
fitMedia(RD_WORLD_PERCUSSION_KALIMBA_PIANO_7,3,1,5) # baseline
fitMedia(RD_WORLD_PERCUSSION_KALIMBA_PIANO_3,4,1,2) # backing
fitMedia(RD_WORLD_PERCUSSION_KALIMBA_PIANO_3,4,3,4) # backing

# Create a 4 measure B section between measures 5 and 9
fitMedia(RD_WORLD_PERCUSSION_DRUMPART_3,1,5,9) # sparse drums
fitMedia(RD_WORLD_PERCUSSION_SEEDSRATTLE_1,3,5,9) # rattling
fitMedia(RD_WORLD_PERCUSSION_KALIMBA_PIANO_3,4,5,6) # backing

# Then back to section A at measure 9
fitMedia(RD_WORLD_PERCUSSION_KALIMBA_PIANO_1,1,9,13) # main
fitMedia(RD_WORLD_PERCUSSION_DRUMPART_24,2,9,13) # drums
fitMedia(RD_WORLD_PERCUSSION_KALIMBA_PIANO_7,3,9,13) # baseline
fitMedia(RD_WORLD_PERCUSSION_KALIMBA_PIANO_3,4,9,10) # backing
fitMedia(RD_WORLD_PERCUSSION_KALIMBA_PIANO_3,4,11,12) # backing

# Effects
setEffect(1,REVERB)
setEffect(1,VOLUME,GAIN,0,4.9,-12,5)
setEffect(1,VOLUME,GAIN,-60,5,0,2)

#Finish
finish()
```

Figure 2. Code in the Code Editor of an example project created in EarSketch.

```
0: ["RD_WORLD_PERCUSSION_KALIMBA_PIANO_1",
  "VOLUME_GAIN 1",
  "REVERB_REVERB_DAMPFREQ 8000",
  "RD_WORLD_PERCUSSION_DRUMPART_24",
  "RD_WORLD_PERCUSSION_KALIMBA_PIANO_7",
  "RD_WORLD_PERCUSSION_KALIMBA_PIANO_3"]

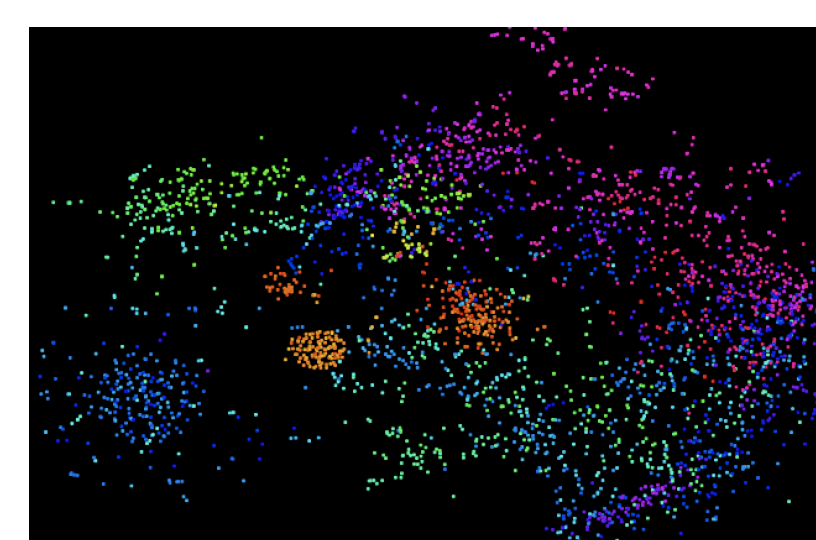
4: ["RD_WORLD_PERCUSSION_DRUMPART_3",
  "VOLUME_GAIN 0",
  "REVERB_REVERB_DAMPFREQ 8000",
  "RD_WORLD_PERCUSSION_SEEDSRATTLE_1",
  "RD_WORLD_PERCUSSION_KALIMBA_PIANO_3"]

6: ["RD_WORLD_PERCUSSION_DRUMPART_3",
  "VOLUME_GAIN 0.4995",
  "REVERB_REVERB_DAMPFREQ 8000",
  "RD_WORLD_PERCUSSION_SEEDSRATTLE_1"]

8: ["RD_WORLD_PERCUSSION_KALIMBA_PIANO_1",
  "VOLUME_GAIN 0.999",
  "REVERB_REVERB_DAMPFREQ 8000",
  "RD_WORLD_PERCUSSION_DRUMPART_24",
  "RD_WORLD_PERCUSSION_KALIMBA_PIANO_7",
  "RD_WORLD_PERCUSSION_KALIMBA_PIANO_3"]
```

Figure 3. Timeline representation for measures 1, 5, 7, and 9 of the sample EarSketch project (see Figure 2).

AUDIO FEATURE ANALYSIS



Co-usage Data & Audio Fingerprinting

Form & Structure Analysis

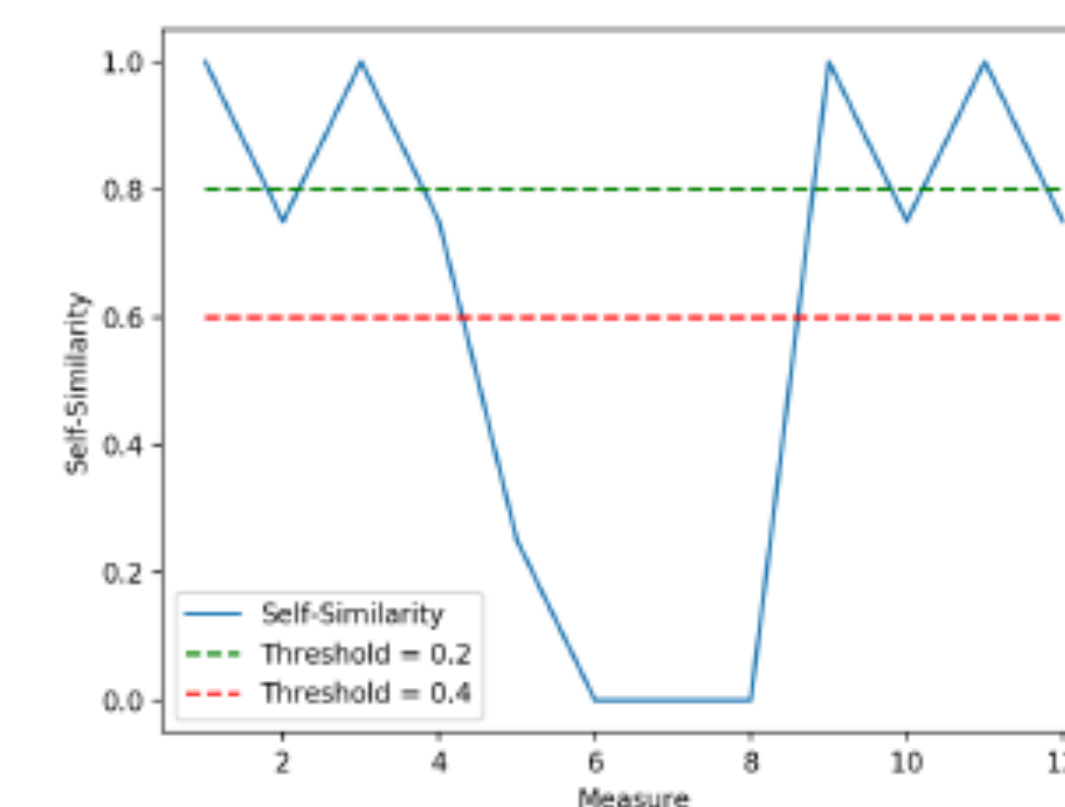


Figure 4. Self-similarity of instrumentation for the example EarSketch project (see Figure 2), indicating an A-B-A form. Dotted lines represent similarity thresholds of 0.4 and 0.2 that, when crossed, mark section beginnings.

- The symbolic music representation is then combined with recorded features and metadata (including statistical usage data) of the sounds in the EarSketch sound library.
- The Analysis Module can use this data to form a large-scale understanding of a user project, such as by using differences in instrumentation to determine form (sections and subsections).
- It can also use this data to assess the overall genre of a project and to perform audio recommendations for a certain section.



Genre & Instrument Metadata

INSIGHTS AND FUTURE WORK

- The module was tested using a collection of over 100 user and researcher-generated scripts, to test its ability for expected output and unorthodox test cases. It was integrated into the EarSketch webpage in Spring 2020, maintaining anonymous error reporting analytics for continuous performance improvements.
- This analysis system is part of the CAI agent, which is currently in development. It will use this analysis data for generating dialogue and suggestions for EarSketch users in a chat menu interface. We aim to increase student engagement and creativity through their interactions with CAI.

CONTACT INFORMATION

Jason Smith, Georgia Tech Center for Music Technology
jsmith775@gatech.edu

EarSketch can be found at:
<http://earsketch.gatech.edu/>

CODE ANALYSIS

- The EarSketch curriculum contains information about various coding concepts which students are encouraged to implement in their projects.

| Category | Concepts |
|--------------|--|
| Value Types | String, Integer, Float, Boolean |
| Data Storage | List, Variable |
| Operations | String Operation, List Operation, Comparison, Boolean Logic, Mathematical Operator |
| Procedure | For Loop, Conditional Statement, User-defined Function, Console Input |

Table 1. Concepts in the analysis module taxonomy.

| Level | String | User-Defined Function |
|-------|---|---|
| 0 | Does Not Use | Does Not Use |
| 1 | Uses | Uses |
| 2 | Uses Originally | Uses Originally |
| 3 | Uses Originally for Purpose | Uses and Calls Originally |
| 4 | Uses Originally and Indexes or Iterates for Purpose | Uses and Calls Originally with Return OR Arguments |
| 5 | N/A | Uses and Calls Originally with Return AND Arguments |

Table 2. Knowledge levels for two concepts: "String" and "User-Defined Function."

| Concept | 0 | 1 | 2 | 3 | 4 | 5 |
|---------------|---|---|---|---|---|---|
| String | | | | | | |
| Integer | | | | | | |
| Float | | | | | | |
| Boolean | | | | | | |
| List | | | | | | |
| Variable | | | | | | |
| String Op | | | | | | |
| List Op | | | | | | |
| Comparison | | | | | | |
| Boolean Logic | | | | | | |
| Math Op | | | | | | |
| For Loop | | | | | | |
| Conditional | | | | | | |
| User Function | | | | | | |
| Console Input | | | | | | |

Table 3. Visualization of code analysis output for a sample project.

```
def sectionA():
    fitMedia(RD_RNB_PIANO_1,1,1,16)
    fitMedia(Y25_DRUMPAD_1,2,1,16)

def sectionA(start, end):
    fitMedia(RD_RNB_PIANO_1,1,start,end)
    fitMedia(Y25_DRUMPAD_1,2,start,end)

sectionA()
sectionA(1,16)

User-defined Function: Level 4
User-defined Function: Level 5
```