Source Code for Symbolic Music Alignment Tool

Contents

1 Package Content

1. Score-to-MIDI alignment by hidden Markov model (HMM) (`ScorePerfmMatcher`)
2. Error detection (`ErrorDetection`)
3. Realignment by merged-output HMM (`RealignmentMOHMM`)
4. MIDI to Spr converter (`midi2pianoroll`)
5. musicXML to Fmt3x converter (`MusicXMLToFmt3x`)
6. musicXML to hmm converter (`MusicXMLToHMM`)

The first three programs are the main algorithms for alignment, and the last three are data converters to prepare for the alignment process. The input files are a musicXML file for score information and a MIDI file (standard MIDI file; SMF) for the aligned performance signal. The alignment algorithms deal with three internal file formats, *Spr*, *Fmt3x*, and *hmm* formats. The Spr file describes the performance signal, and the Fmt3x and hmm files describe the score information. The output is described with the *match* file format. Please see Sec. 2.2 for details.

2 Compiling and Usage

2.1 Compile

To compile, process

```
$./compile.sh
```

or process lines written in this shell script file with adaptations to the environment.

2.2 File Formats

The package deals with the following file formats: *standard MIDI file (SMF)* format, *musicXML*, *Spr*, *Fmt3x*, *hmm*, and *match* format. A 'MIDI file' in this manual always means an SMF. For musicXML, see `https://www.musicxml.com`. The rest four formats are original and explained below.

The Spr file describes the note information in the MIDI file. The first lines look like the following:

```
..... (Some comment lines)
0 0.01 0.12 C4 19 80 0
1 0.02 0.15 C5 32 80 0
.....
```

From left to right, columns in each line indicate:
`ID (onset time) (offset time) (spelled pitch) (onset velocity)`
`(offset velocity) channel`

The fmt3x file describes the score information extracted from the musicXML file. The first lines go as

```
//TPQN: 24
..... (Some comment lines)
4 3 2 1 0 0 chord 12 3 C#4 G4 B4 N.. N.. N.. P1-3-45 P1-3-46 P1-3-47
7 3 1 0 0 0 chord 3 1 C#6 N.. P1-3-26
.....
```

The `TPQN` indicates the *tick per quarter note*, which defines the unit of score time. Each line in the main body indicates a score event. From left to right, columns indicate:
`(onset score time) (bar number) staff voice sub-voice order`
`(event type) duration (number of notes)`
followed by `(spelled pitch)`, `(note type)`, and `(note ID)` for each note. The note ID indicates the part, bar, and the order of notes in the bar, given in the musicXML file. For more information, see `Fmt3x_v170225.hpp` in the code folder.

The match file describe the result of alignment. The first lines go as

```
..... (Some comment lines)
0 0.841 1.041 F#3 60 80 0 0 0 P1-1-33 0 0
1 0.864 0.950 G6 83 80 0 0 0 P1-1-1 0 -
.....
//Missing 330 P1-4-42
.....
```

Each line in the first part describes a note in the aligned MIDI file as in the Spr file. From left to right, columns indicate:
`ID (onset time) (offset time) (spelled pitch) (onset velocity)`
`(offset velocity) channel (match status) (score time) (note ID)`
`(error index) (skip index)`

Error index is `0` for a correct note, `1` for a note with an erroneous pitch, and `3` for an extra note. For an extra note, the given note ID is `*`. Each line in the second part describes missing notes, notes in the score (fmt3x) file that do not appear in the alignment result. The number and the symbol indicate the score time and the note ID of the missing note.

2.3 Usage

For an input musicXML (`in1.xml`) and a MIDI file (`in2.mid`), please enter the following to run the alignment programs:

```
$./align.sh in1(.xml) in2(.mid)
```

The output files are the final alignment result (`in2_realigned_match.txt`), the alignment result by the HMM (`in2_pre_match.txt`), the result after applying the performance error detection to `in2_pre_match.txt` (`in2_err_match.txt`), and some internal files.