A notation-based query language for searching in symbolic music

Matan Gover and Ichiro Fujinaga

DLfM ’19
6th International Conference on Digital Libraries for Musicology
November 2019, The Hague, Netherlands
Motivation

- Symbolic music corpora are growing, thanks to:
  - Manual encoding (e.g., OpenScore [1])
  - Improvements in OMR [2]


Symbolic music search

Approaches:

• Exact matching (pitches/intervals/rhythm)

• Similarity-based ("fuzzy" matching)

• Pattern-based
Exact matching

Cum turba multa esset cum

Filii hominum*

Alleluia alleluia alleluia alleluia alleluia

Quid est quod me quaerebatis

http://cantus.uwaterloo.ca/melody (Lacoste 2012)
Easy to use

Limited expressivity
Similarity-based

https://www.musipedia.org (Prechelt and Typke 2001)
Imprecise by design
Useful for retrieval tasks
Pattern-based

- Precise matching
- Complex queries with wildcards
- Useful for musicological tasks:
  - Voice leading patterns (parallel fifths, cadences)
  - Motivic patterns (rhythmic and/or melodic)
Pattern-based

ThemeFinder

Pitch  C E - A -

Gross Contour  \ /

A-G, sharp=#, flat=-
e.g. C E - G F#

up=\, down=\, unison=--
e.g. //\-\- or uudsu

http://www.themefinder.org (Kornstädt 1998)
Pattern-based

Humdrum

• Text-based music encoding format

• Comes with powerful set of manipulation tools

• Can be searched using regular expressions (regexes)

http://www.humdrum.org (Huron 1994)
ditto -s ^ = inputfile | hint -l > temp1
deg inputfile > temp2
assemble temp1 temp2

**kern  **kern  **kern  **hint  **deg  **deg  **deg
*clefF4  *clefG2  *clefG2  *  *  *  *
=    =    =    =    =    =    =
4A    4e    8e    P5  P5    v2    v6    ^6
.     .     8f    P5  m6    .     .    ^7-
4B-   4d    8g    M3  M6    ^3-    v5    ^1
.     .    4f#    M3  A5    .     .    v7
4A    4c#    .    M3  M6    v2    v4+    .
.     .    8e    M3  P5    .     .    v6
=    =    =    =    =    =    =
2.G   2.d    2.g    P5  P8    v1    ^5    ^1
==   ==   ==   ==   ==   ==   ==
*    *    *    *    *    *    *
Regular expressions are extremely powerful

Steep learning curve for musicians and musicologists

Text manipulation
When I search for music

I want my query to look like music
Our approach

- Query primitives
  - Inspired by regular expressions
  - Extension of standard music notation

- Encoding – extension of MEI

- Execution engine

- Query interface

- Not “finished”
<beam>
  <note dur="8" dots="1"
    query:any-pitch="true" />
  <note dur="16"
    query:any-pitch="true" />
  <note dur="8"
    query:any-pitch="true" />
</beam>
<note> Matches a single note

Exact match

<note pname="e" oct="5" dur="4" />

Pitch-only

<note pname="e" oct="5"
query:any-duration="true" />

Rhythm-only

<note dur="4" query:any-pitch="true" />

Any accidental

<note pname="e" oct="5" dur="4"
query:any-accidental="true" />
<note> Matches a single note

Exact match

<note pname="e" oct="5" dur="4" />

Pitch-only

<note pname="e" oct="5"
query:any-duration="true" />

Rhythm-only

<note pname="e" oct="5"
query:any-accidental="true" />

Any accidental

<notepname="e" oct="5" dur="4"
query:any-accidental="true" />
Example: rhythm-only

Query:

<beam>
  <note dur="8" dots="1" query:any-pitch="true" />
  <note dur="16" query:any-pitch="true" />
  <note dur="8" query:any-pitch="true" />
</beam>

Results:
<query:or> Matches either of two patterns

Query:

Results:

Query encoding:

<note pname="e" oct="5" dur="4" />
<notepname="e" oct="5" dur="8" />
<query:or />
<note pname="b" oct="4" dur="4" />
<note pname="b" oct="4" dur="8" />
Nested or repeated patterns

Quantifiers

Nested groups
Example: group

Query:

Results:

Query encoding:

<note pname="c" oct="5" accid="s" query:any-duration="true" />
<query:group min-occurrences="1">
  <note pname="e" oct="5" query:any-duration="true" />
  <query:or />
  <note pname="d" oct="5" query:any-duration="true" />
</query:group>
<note pname="c" oct="5" accid="s" query:any-duration="true" />
Rendering

- In the browser using Verovio (Pugin et al. 2014)
- Extended MEI is transformed to standard MEI

```
<note query:any-duration="true" />
```

```
<note stem.visible="false" />
```
Execution engine

- Query primitives can be mapped directly to regexes
- Avoid re-inventing the wheel: Use existing regex engine
- Query transformed to regex and executed on Humdrum representation of score (**kern)**
<note pname="e" oct="5"
query:any-duration="true" />

**kern
(8.cc#L
16ddk)
8cc#J
4ee
8ee

\[(^|\t)(&?\{)?(&?\()?[?][0-9]+\).*ee(?! [a-gA-G#\-n]).*(^[!.*=].*)?\]*
Benchmark

• Tested on Essen Folksong Collection (Schaffrath 1995) (8,473 scores)

• Simple query took ~0.75 seconds to complete
  • Produced 4,738 matches in 2,369 scores

• For larger corpora, execution can be parallelized
Future work

• Transposition invariance, octave invariance, rests, chords

• Layers (scale degree, contour, harmony, lyrics…)

• Graphical query input! (Requires Verovio functionality)

• Server-side searching

• Polyphonic music – constraint-based?
Future work

- Transposition invariance, octave invariance, rests, chords
- Layers (scale degree, contour, harmony, lyrics…)
- Graphical query input! (Requires Verovio functionality)
- Server-side searching
- Polyphonic music – constraint-based
What motivates a musical query?

“The only common thread in music-query motivations, broadly defined, is – alas – human curiosity. Unless we lose that, designers of music-query software can expect to cater for an unending stream of “special” needs, as musical preferences continue to evolve and change.”

Try it!

https://www.matangover.com/musicquery
Thank you

https://www.matangover.com/musicquery