Humdrum File Format

craig@ccrma.stanford.edu

6 March 2018

Humdrum File Format

- Humdrum file format is analogous to XML: organizing structure for data without concern for content.

- Each data stream is a column of data (called a *spine*) started by an exclusive interpretation which is two stars followed by the data-type name. The end of the data is marked with star-minus (not star-underscore as it may seem in the Courier font).

- Temporal: organized strictly time-wise in the data: each succeeding row comes after the preceding (contrasts with all other polyphonic data formats except MIDI Type-0 files).

- Each element on a data line occurs simultaneously in time (a4, b4, c4 occur at the same time).

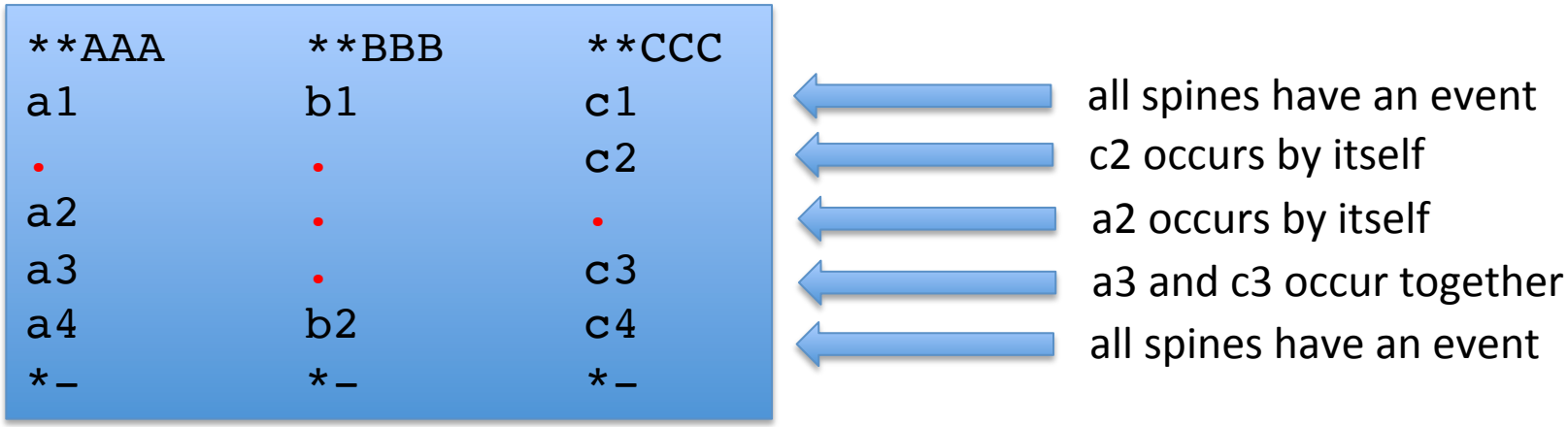
spine

**AAA	**BBB	**CCC	**exclusive interpretation (data type)**
a1	b1	c1	
a2	b2	c2	
a3	b3	c3	**data**
a4	b4	c4	
a5	b5	c5	
*-	*-	*-	**spine terminators (data end)**

Field Separator/Null Records

- Each field on a line is separated by exactly *one* tab character (may cause vertical alignment aberrations in a text editor if a field is wide).

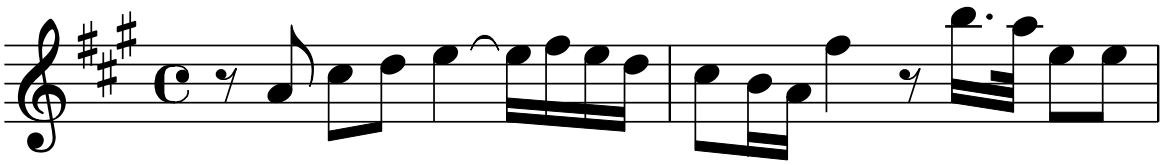
- If one spine has no event when others do, a **null token** (“.”) is used as a place holder to indicate that the previous item in the spine is still in effect.



Motivation for Humdrum format

- Reaction against the linear parsing needed in DARMS code

8-'A{"CD}4E+{6EFED}/{8C'6BA}"4F8-{6.B3A}{8EE}/



- Spreadsheet model for processing the data.

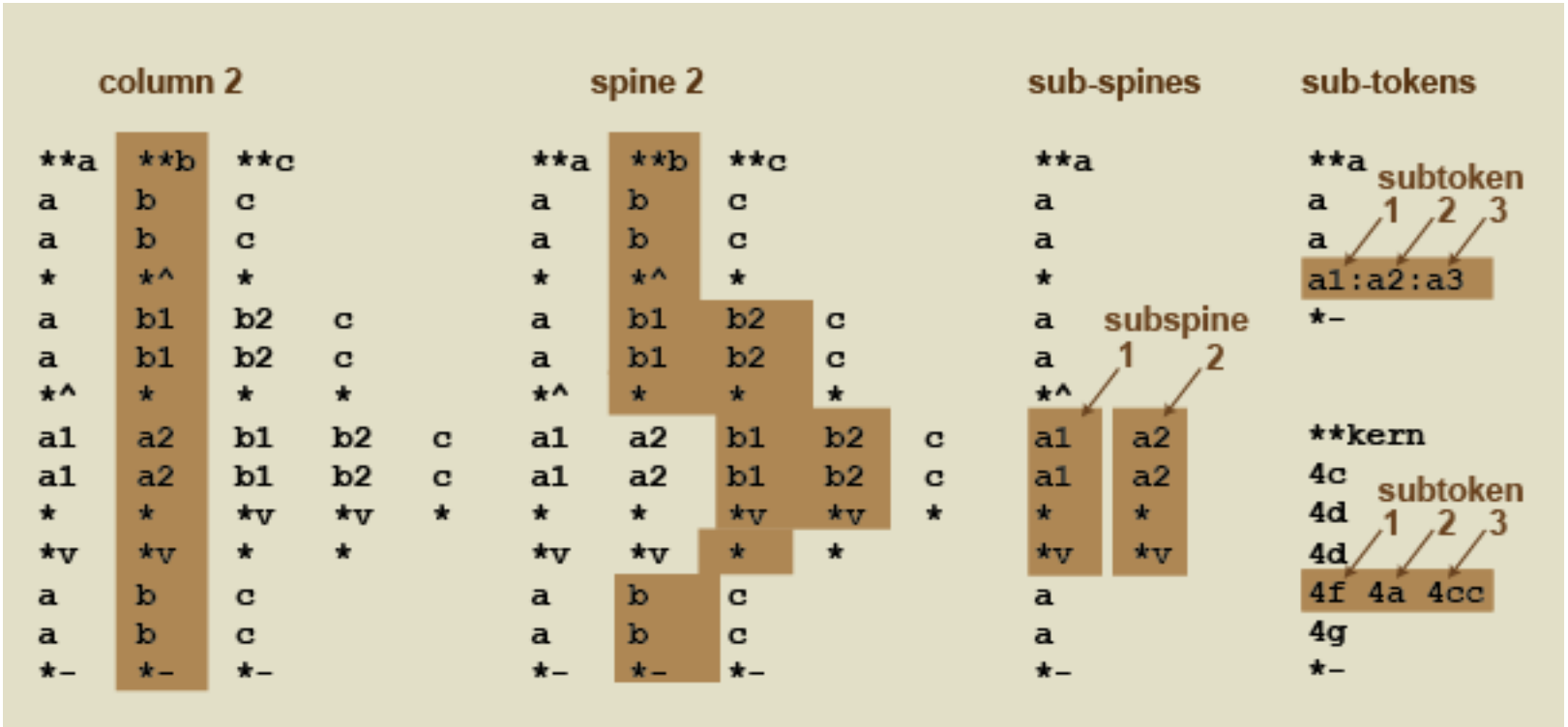
	A	B	C
1	**AAA	**BBB	**CCC
2	a1	b1	c1
3	.	.	c2
4	a2	.	.
5	a3	.	c3
6	a4	b2	c4
7	*-	*-	*-
8			

- Increased random access (e.g., no sticky settings for rhythm/octave like DARMS/Guido/Lilypond/ABC) for better **regular-expression** parsing.

```
**kern
*clefG2
*k[f#c#g#]
*M4/4
*met(c)
=1-
8r
8a
8cc#
8dd
[4ee
16ee]
16ff#
16ee
16dd
=2
*-
```

Spine Manipulators

**datatag	Exclusive interpretation
*^	Spine split (into two sub-spines)
*v	Spine merge (contiguous spines merge)
*x	Spines exchange column positions (rare)
*+	Add new spine
*-	Spine terminator (end of data for spine)
*	Null interpretation (null manipulator)



Record (line) Types

Comments (starting with !)

Reference Record (bibliographic record):

!!!key: value

Global Comment:

!!unstructured text

Local Comment:

!text

Interpretations (starting with *)

Spine Manipulators (fixed set):

**start, *^, *v, *x, *+, *-

Tandem Interpretations (data-type dependent):

*clefG2, *M4/4, *MM=120

Null interpretation (place-holder for empty interpretation)

*

Barline (starting with =)

technically a form of data in Humdrum (should be interpretation)

Data line (starting anything else)

Reference Records

!!!COM: Mozart, Wolfgang Amadeus
!!!CDT: 1756/01/27/-1791/12/05/
!!!CNT: German
!!!OTL: Piano Sonata No. 16 in B-flat major
!!!SCT: K 576
!!!OMV: Mvmt. 2
!!!OMD: Adagio
!!!ODT: 1789///

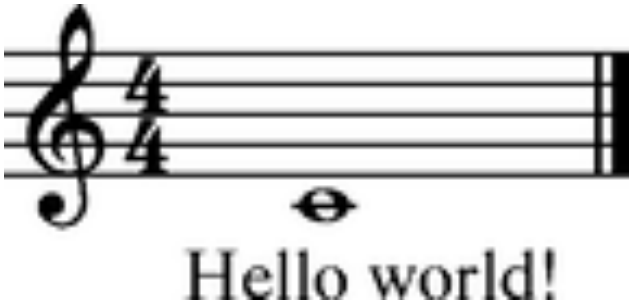
COM = Composer:	Mozart, Wolfgang Amadeus
CDT = Composer's dates:	27 Jan 1756 - 5 Dec 1791
CNT = Nationality:	German
OTL = Title:	Piano Sonata No. 16 in B-flat major
SCT = Scholarly cat. num.:	K 576
OMV = Movement number:	Mvmt. 2
OMD = Movement designation:	Adagio
ODT = Date of composition:	1789

http://www.humdrum.org/Humdrum/guide.append1.html

**kern

- **kern = exclusive interpretation (data type) which represents the “core” (in German) of musical data.
- Primary data format for storing music in Humdrum files.

```
**kern    **text
*M4/4     *
=-        =-
1c        Hello world!
==        ==
*-        *-
```



Tandem interpretations for **kern

*clefG2 = treble clef (G clef on second line from bottom of staff)
*clefGv2 = vocal tenor clef (G clef on second line, notes transposed down an octave)
*clefF4 = bass clef (F clef on fourth line from bottom of staff)
*clefC3 = alto clef
*MM120 = tempo marking (120 beats per *quarter* note)
*k[f#c#g#] = key signature of A major/F# minor
*A: = music is in A major
*f#: = music is in F# minor
*k[b-e-a-d-g-c-f-] = key signature for C-flat major
*C-: = music is in C-flat major
*M3/4 = ¾ meter
*met(c) = common time
*met(c|) = cut time
*met(O) = circle mensuration

http://www.musiccog.ohio-state.edu/Humdrum/guide02.html

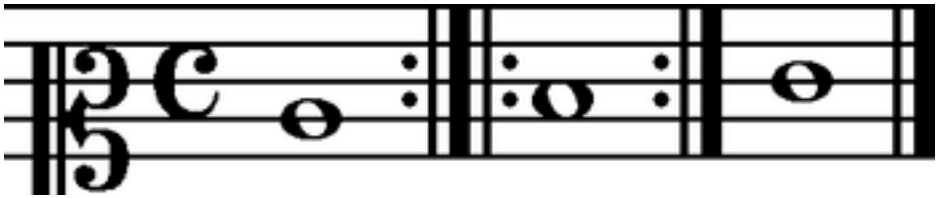
Pitch and Rhythm

c = middle c (C4)
cc = octave higher (C5)
ccc = two octaves higher (C6)
C = octave lower (C3)
CC = two octaves lower (C2)
B = minor second below c
b = major seventh above c
c# = c-sharp
c- = c-flat
c## = c-double sharp
c-- = c-double flat
r = rest

4 = quarter note
8 = eighth note
16 = 16th note
32 = 32nd note
64 = 64th note
2 = half note
1 = whole note
2. = dotted half note
4.. = double dotted quarter note
12 = triplet eighth note
20 = quintuplet sixteenth note
3%2 = triplet whole note (extension)

Barlines

=	plain barline
=23	barline #23
=23;	barline #23 with a fermata over it
=-	invisible barline
=:|!	left-pointing repeat barline (| = light line, !=heavy line).
=:|!|:	left-right repeat barline
==	double (final) barline
=’	partial mid barline
=`	partial top barline



```
**kern
*clefC2
*M4/4
*met(c)
=-
1c
=2:|!|:
1d
=3:|!
1e
==
*-
```

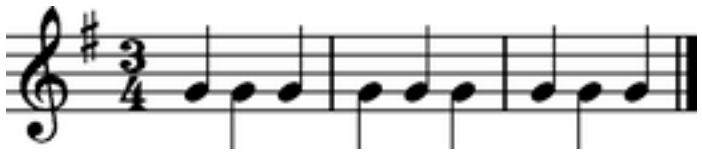
http://www.musiccog.ohio-state.edu/Humdrum/representations/barlines.rep.html

Stem directions

```
**kern
*M3/4
*k[f#]
=-
4g/
4g\
4g/
=2
4g\
4g/
4g\
=
4g/
4g\
4g/
==
*-
```

/ = stem up

\ = stem down



Beams and Ties/Slurs

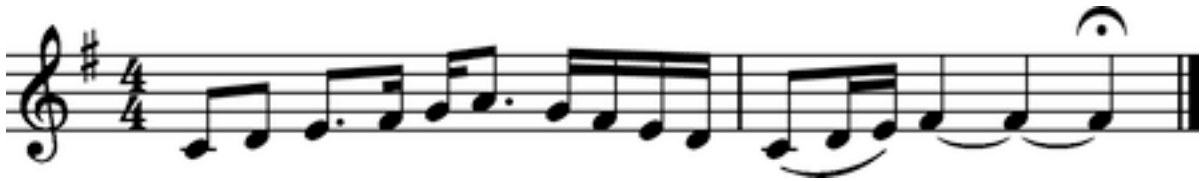
L = start of beam line
J = end of beam line
LL = start of two beam lines
JJ = end of two beam lines
K = right facing partial beam
k = left-facing partial beam

[= start of tie
_ = middle note in tie (continues in both directions from printed note.
] = end of tie

(= beginning of slur
) = end of slur

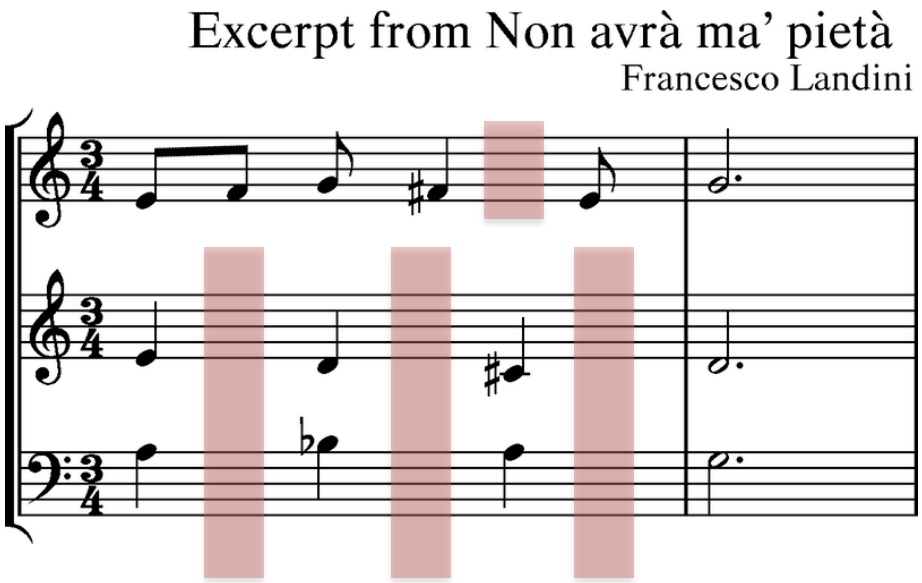
{ = beginning of phrase
} = end of phrase

```
**kern
*M4/4
*k[f#]
=-
{8c/L
8d/J
8.e/L
16f#/Jk
16g/LK
8.a/J
16g/LL
16f#/
16e/
16d/JJ}
=
(8c/L
16d/L
16e/JJ)
[4f#
4f#_
4f#];
==
*-
```



Multiple parts

```
!!!COM: Landini, Francesco
!!!OTL: Excerpt from Non avr&agrave; ma' piet&agrave;
**kern **kern **kern
*clefF4    *clefG2   *clefG2
*M3/4      *M3/4     *M3/4
=          =         =
4A         4e        8eL
.          .         8fJ
4B-        4d        8g
.          .         4f#
4A         4c#       .
.          .         8e
=          =         =
2.G        2.d       2.g
=          =         =
*-         *-        *-
!!muse2ps: z21jw1500
```



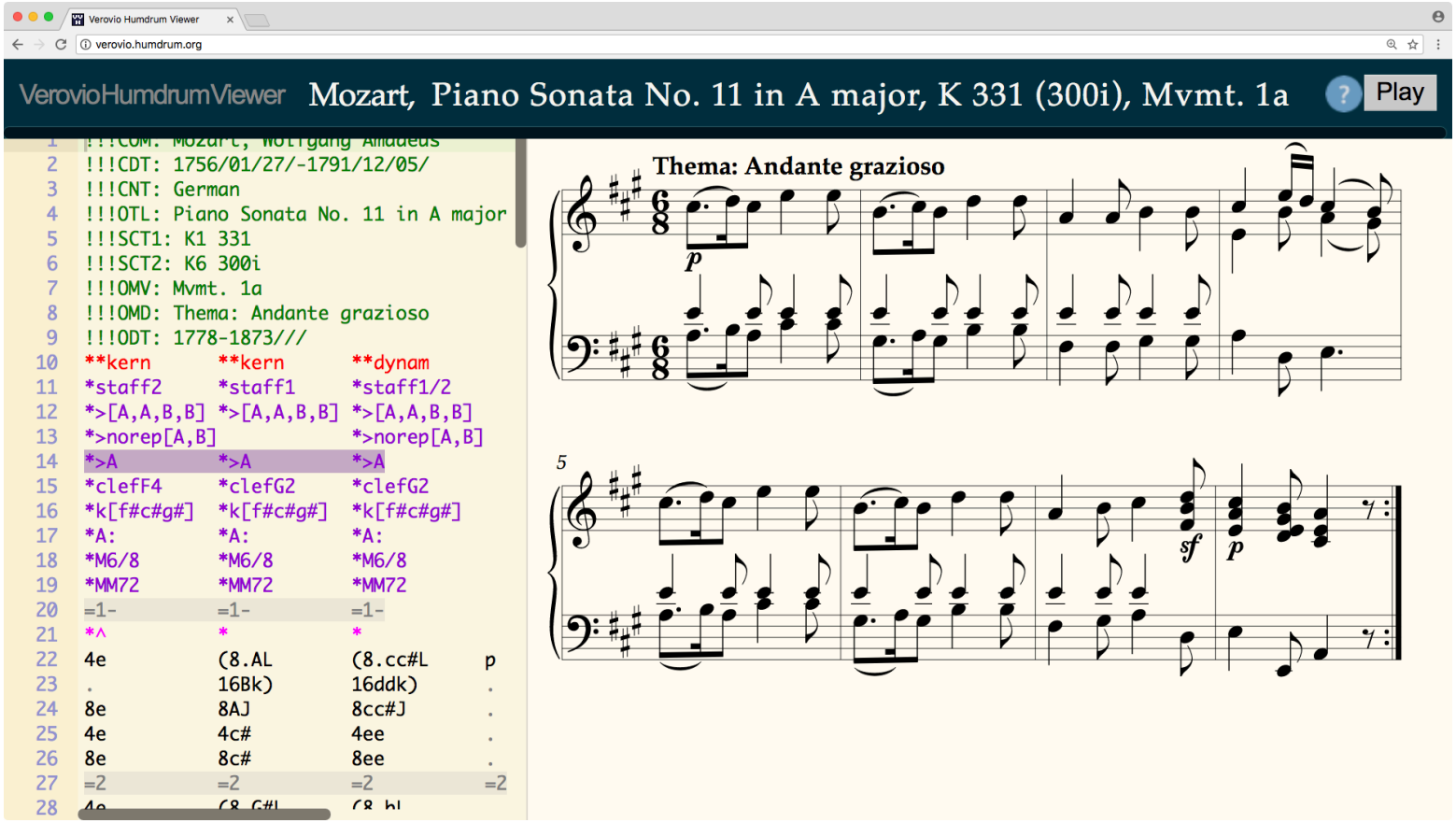
Null tokens

http://wiki.ccarh.org/wiki/Muse2ps#Options

Verovio Humdrum Viewer

http://bit.ly/mec2017-vhv

http://verovio.humdrum.org



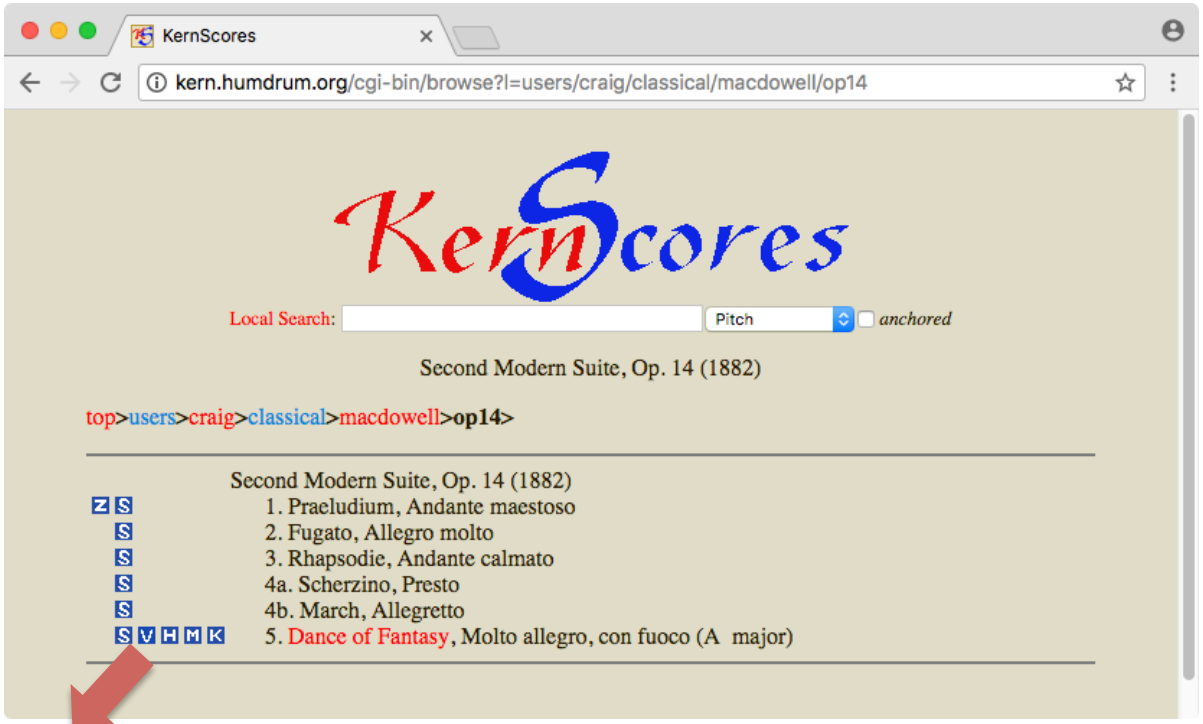
KernScores

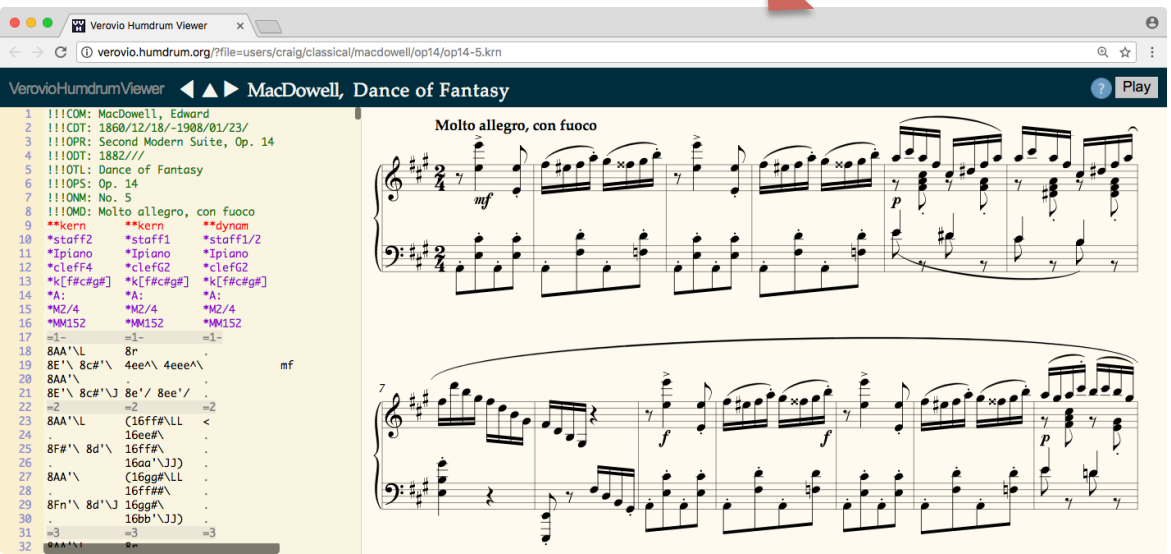
http://kern.ccarh.org

Database of Humdrum files containing **kern data.

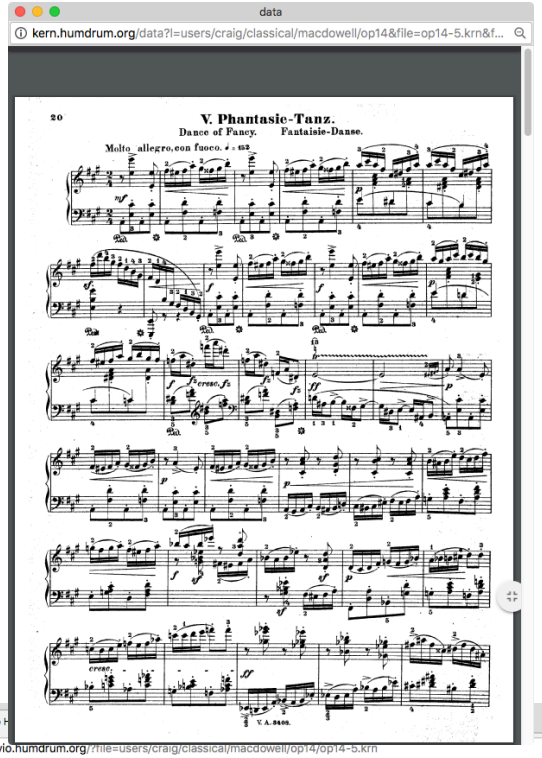
- http://kern.humdrum.org/help/tour
- http://kern.humdrum.org/cgi-bin/kseditor

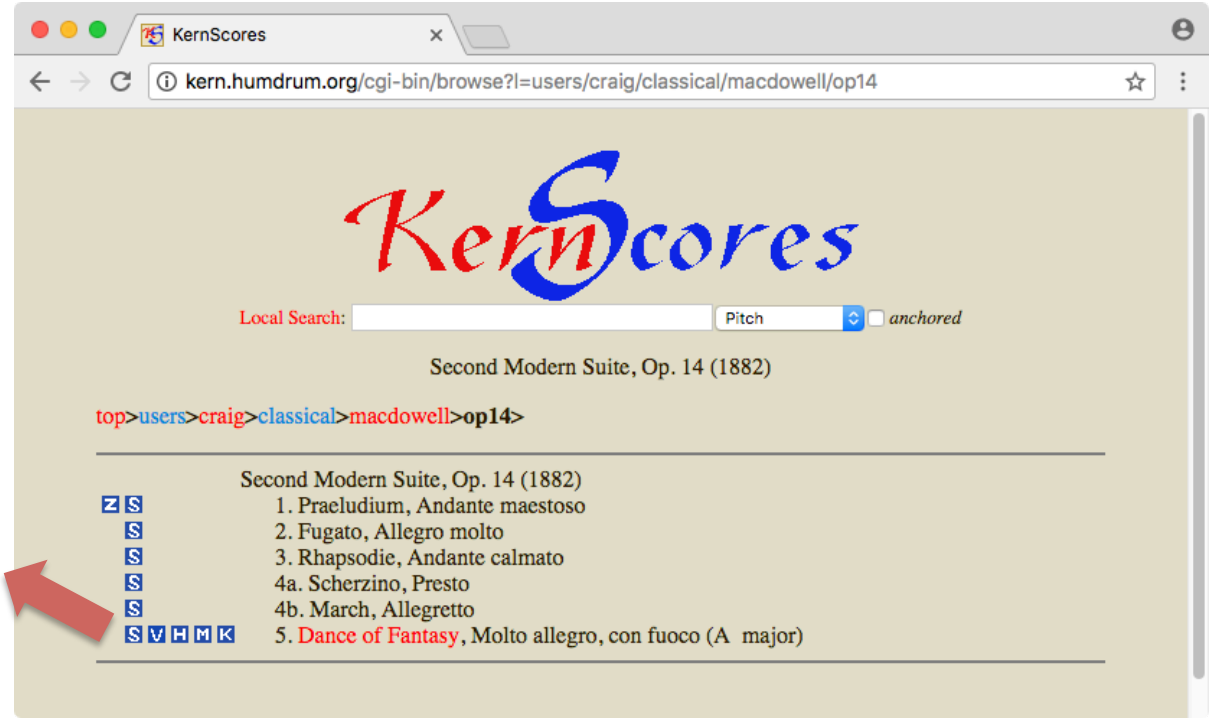
V button on KernScores

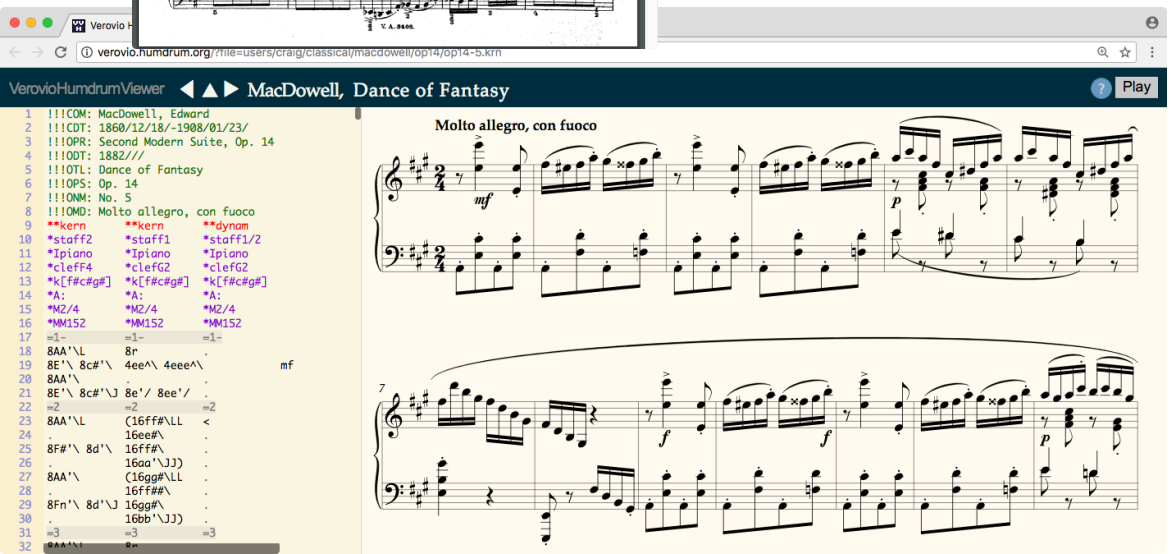




S button on KernScores



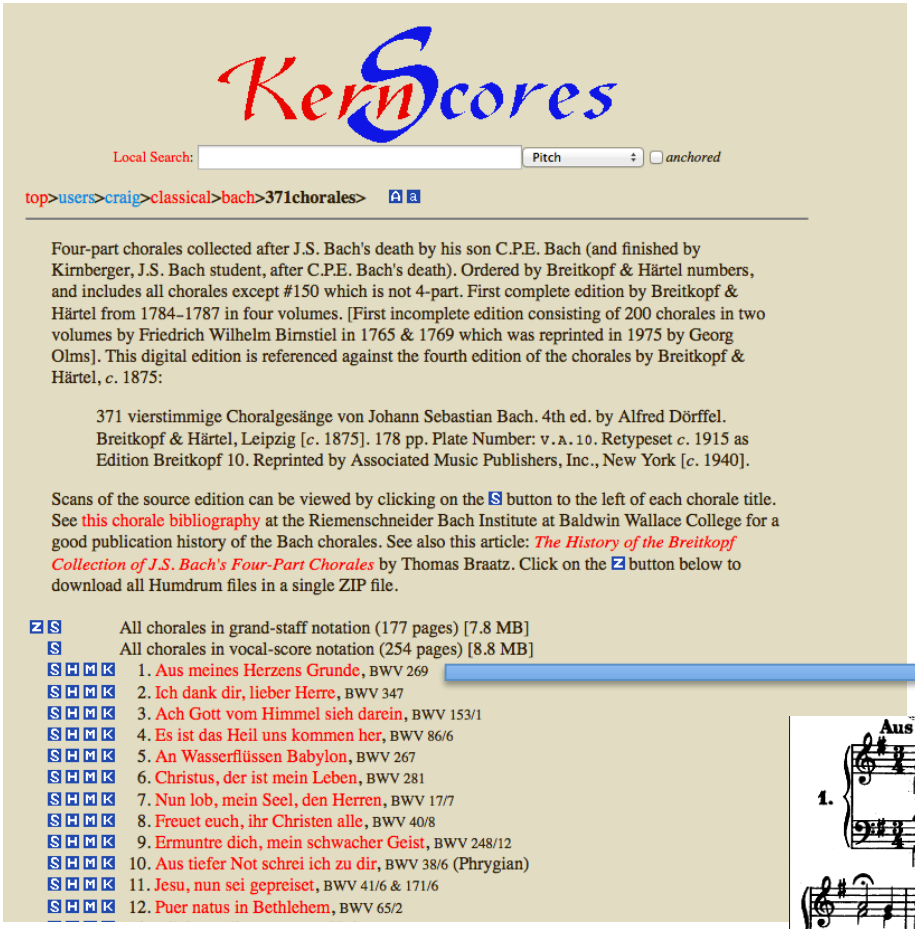


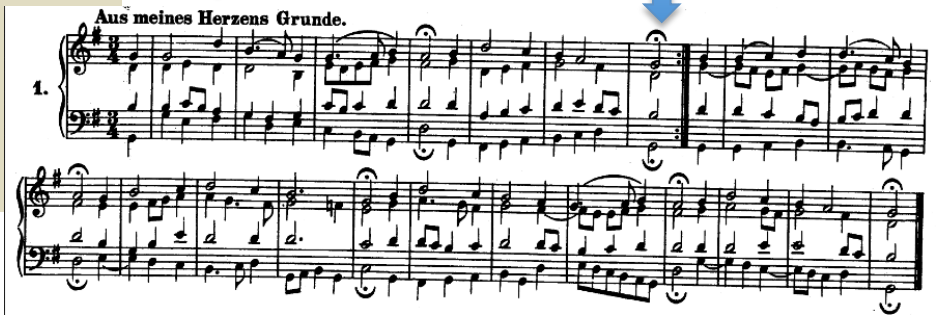


Or type **alt-p** in VHV
To view PDF of original scan.

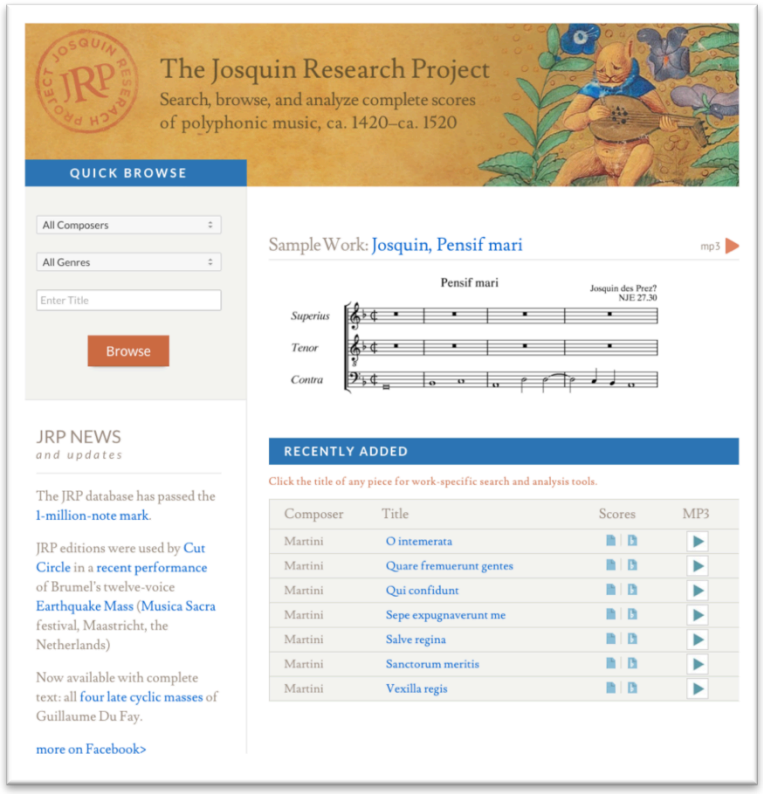
KernScore Browse

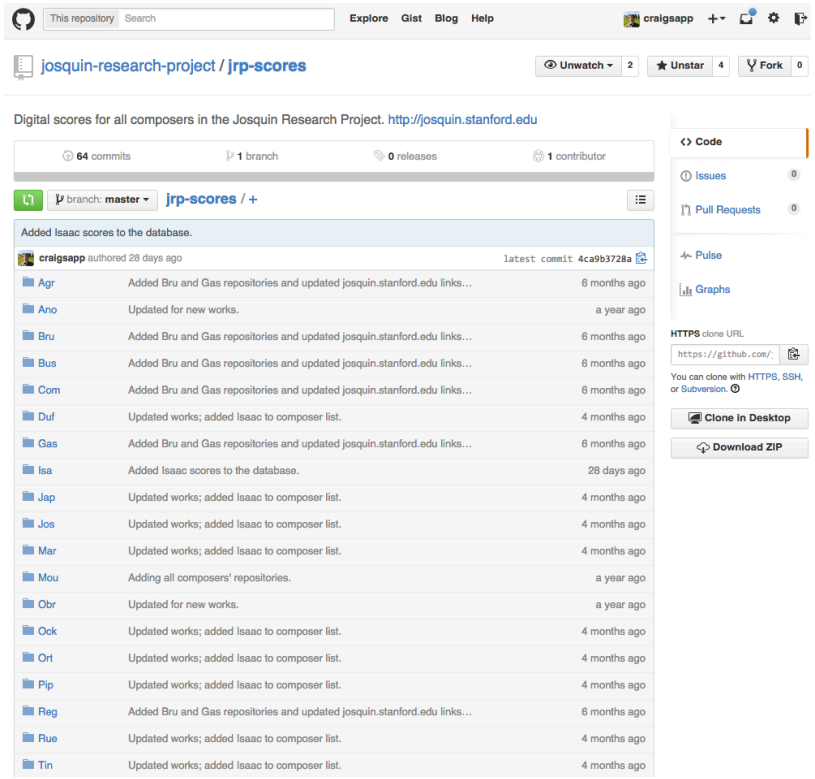
http://kern.ccarh.org/browse?l=371chorales





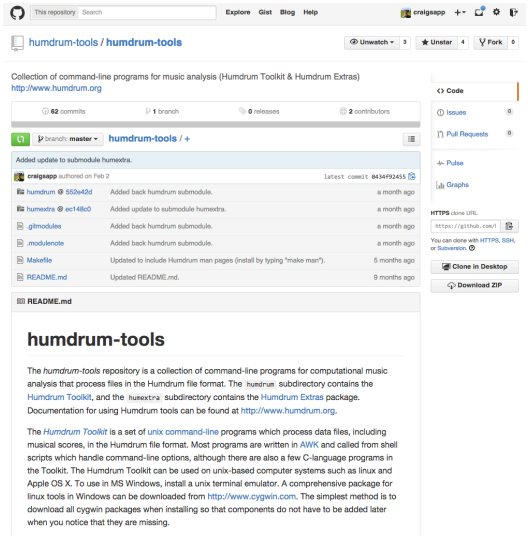
Josquin Research Project





https://github.com/josquin-research-project/jrp-scores

Github



- Install (unix) command-line utilities for procesing Humdrum data from Github

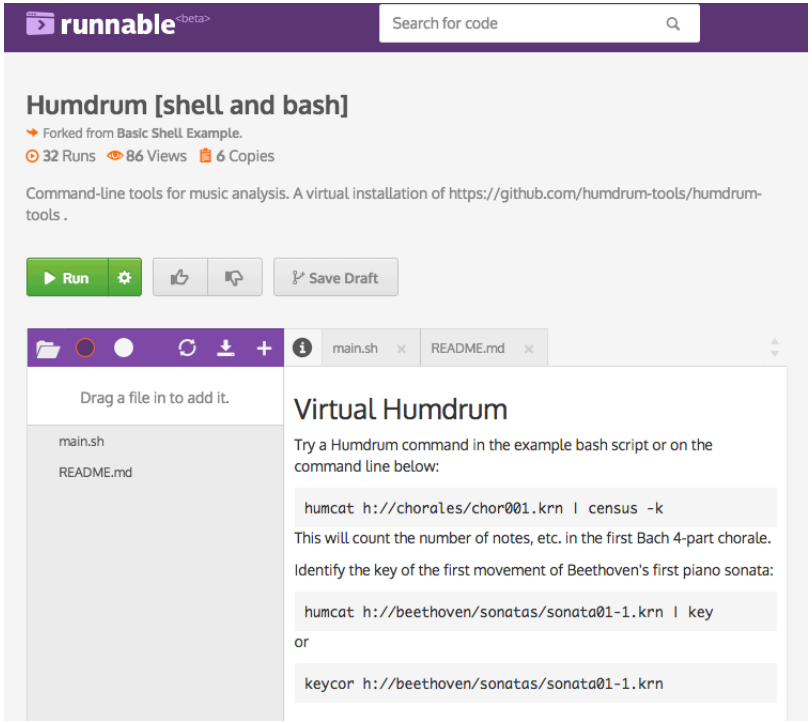
- Github repository for Humdrum tools:
  https://github.com/humdrum-tools/humdrum-tools

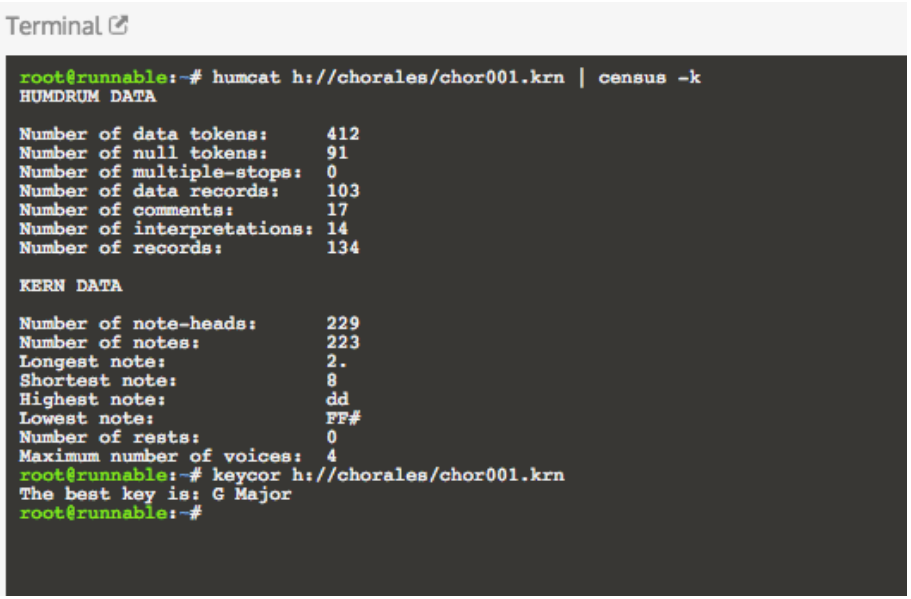
- Github repository for Humdrum data (taken from http://kern.humdrum.org website):
  https://github.com/humdrum-tools/humdrum-data

Virtual Humdrum

- Online bash shell for running Humdrum tool commands.
- Slow, but allows testing against your installation and easier than installing in Windows with cygwin

http://runnable.com/VD9ZirF3Zp1gkPZM/humdrum-for-shell-and-bash

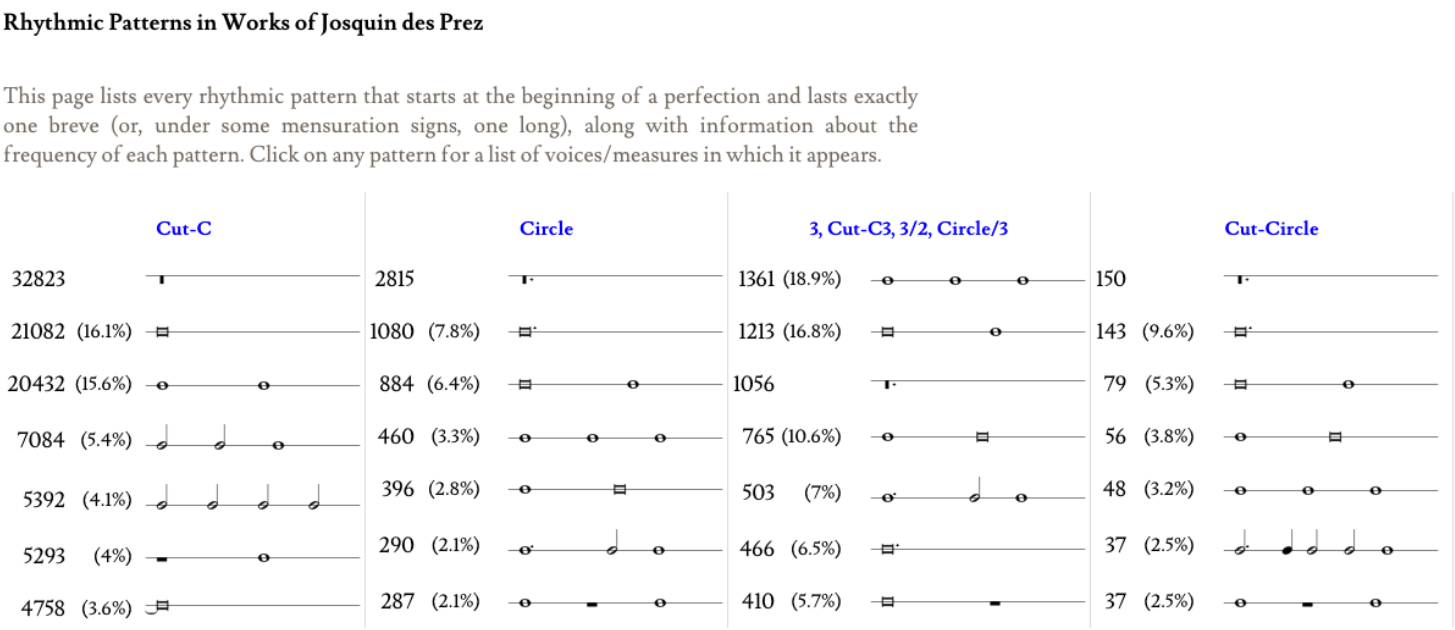




Example problem

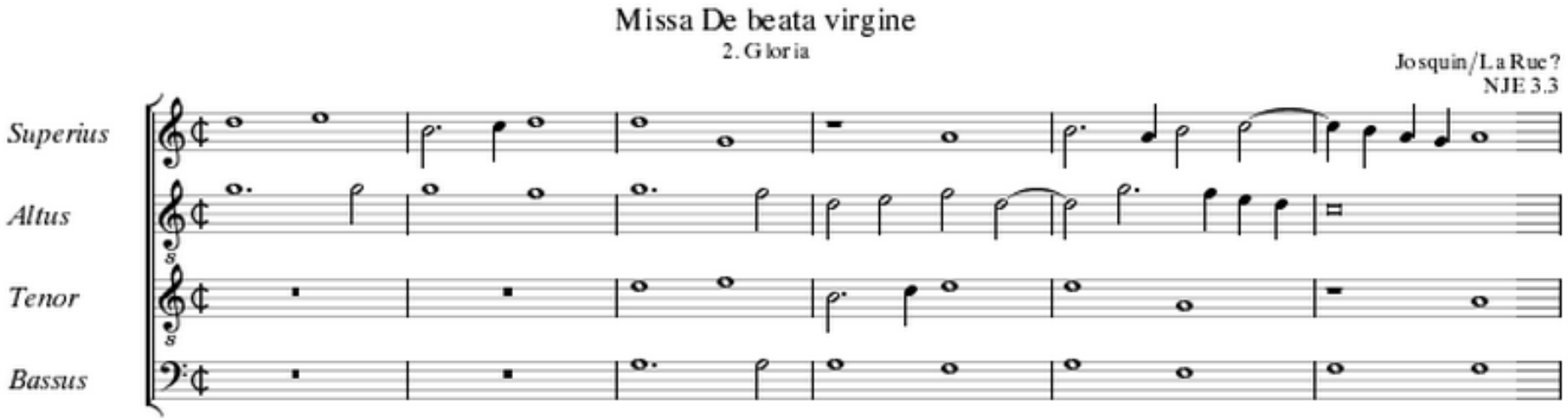
- Examine the trend in duple v. triple meters (mensurations) in early renaissance music

http://josquin.stanford.edu/analysis/rhythm



http://museinfo.sapp.org/examples/humdrum/menpat.cpp

Metric rhythm patterns




```
v01    menCutC ::    w_w
v01    menCutC ::    md_q_w
v01    menCutC ::    w_w
v01    menCutC ::    wr_w
v01    menCutC ::    md_q_m_m
v01    menCutC ::    tq_q_q_q_w

v02    menCutC ::    wd_m
v02    menCutC ::    w_w
v02    menCutC ::    wd_m
v02    menCutC ::    m_m_m_m
v02    menCutC ::    tm_md_q_q_q
v02    menCutC ::    b

v03    menCutC ::    br
v03    menCutC ::    br
v03    menCutC ::    w_w
v03    menCutC ::    md_q_w
v03    menCutC ::    w_w
v03    menCutC ::    wr_w

v04    menCutC ::    br
v04    menCutC ::    br
v04    menCutC ::    wd_m
v04    menCutC ::    w_w
v04    menCutC ::    w_w
v04    menCutC ::    w_w
```

menpat –v jrp://Jos0303b

Metric rhythm patterns (3)

```
menpat * | sed 's/:.*//' | sort | uniq -c | sort -nr
```

```
186206 menCutC
 17806 menCircle
  4868 men3
  3608 menC
  2241 menC2
  1878 menCircle2
  1866 menCutC3
  1688 menCutCircle
  1486 menCircleOver3
   336 menCircleDot
   320 menC3
   302 men3Over2
   274 menCutC2
   162 m(C|/3)
   140 m(C|/2)
   125 menCDot
   114 men2
   100 m(O|3)
    80 menCutCircle3Over2
    70 menReverseC
    60 m(O3)
```

menpat * → extract mensural patterns

sed 's/:.*//' → remove from ":" to end of line

sort → sort lines alphabetically

uniq -c → remove duplicates lines, counting number of duplicates

sort –nr → sort lines in reverse numeric order0

Metric rhythm patterns (4)

```
menpat Jos/* | grep -Pv "\s[^_]*r\s*$" | sed 's/:.*//' | sortcount -ph
```

```
**pcent **data
82.33    menCutC
8.05     menCircle
2.34     men3
1.56     menC
1.15     menC2
0.99     menCircle2
0.88     menCutC3
0.83     menCutCircle
0.7      menCircleOver3
0.18     menCircleDot
0.16
0.14     menCutC2
0.14     menC3
0.12     men3Over2
0.07     m(C|/2)
0.07     m(C|/3)
0.06     menCDot
0.05     men2
0.05     m(O|3)
0.04     menCutCircle3Over2
0.04     menReverseC
0.03     m(O3)
0
*-    *-
```

grep -Pv "\s[^_]*r\s*$" → remove measures with full-measure rest.

sortcount –ph → similar to sort | uniq –c | sort -nr

-p = show counts as percentages

-h = format data in Humdrum syntax

https://github.com/craigsapp/humextra/blob/master/scripts/perl/sortcount

Metric rhythm patterns (5)

```
menpat Ock/* | grep -Pv "\s[^_]*r\s*$" | sed 's/:.*//' | sortcount -ph
```

Ockeghem:

**pcent	**data
50.98	menCutC
32.3	menCircle
13.47	menC
1.06	menCDot
1.02	menCutCircle
0.5	men3
0.31	m(O3)
0.15	men2
0.07	m(C|.)
0.06	m(C.|)
0.06	menCircle2
0.02	menReverseC
0.01	menCircleDot
*-	*-

Josquin:

**pcent	**data
82.33	menCutC
8.05	menCircle
2.34	men3
1.56	menC
1.15	menC2
0.99	menCircle2
0.88	menCutC3
0.83	menCutCircle
0.7	menCircleOver3
0.18	menCircleDot
0.16	
0.14	menCutC2
0.14	menC3
0.12	men3Over2
0.07	m(C|/2)
0.07	m(C|/3)
0.06	menCDot
0.05	men2
0.05	m(O|3)
0.04	menCutCircle3Over2
0.04	menReverseC
0.03	m(O3)
0	
*-	*-

Metric rhythm patterns (6)

- Separate JRP data files by genre:

(mkdir songs; cd songs; humsplit h://jrp/Zso)
(mkdir songs; cd motets; humsplit h://jrp/Zmo)
(mkdir songs; cd masses; humsplit h://jrp/Zma)

```
menpat motets/*.krn | grep -Pv "\s[^_]*r\s*$" |  sed
's/:.*//' | sortcount -ph
```

**pcent	**data
86.73	menCutC
4.87	menCircle
2.55	men3
2.43	menC2
0.97	menCircleOver3
0.85	menC
0.4	menCircle2
0.2 6	menCutC3
0.16	menCutC2
0.15	menCircleDot
0.14	men3Over2
0.14	menCutCircle
0.07	menC3
0.01	menCDot
*-	*-

Motets:

**pcent	**data
86.73	menCutC
4.87	menCircle
2.55	men3
2.43	menC2
0.97	menCircleOver3
0.85	menC
0.4	menCircle2
0.26	menCutC3
0.16	menCutC2
0.15	menCircleDot
0.14	men3Over2
0.14	menCutCircle
0.07	menC3
0.01	menCDot
*-	*-

Masses:

**pcent	**data
52.87	menCutC
18.92	menCircle
9.23	menC
8.7	menC2
3.66	men3
2.2	menCutCircle
1.31	menCutC3
1.01	menCircle2
0.5	menC3
0.45	menCircleOver3
0.31	menCircleDot
0.29	menCDot
0.08	men2
0.08	m(C|/2)
0.08	m(C|/3)
0.07	menReverseC
0.06	m(O|3)
0.05	menCutC2
0.03	men3Over2
0.02	menCutCircle3Over2
0.01	m(C|.)
0.01	m(C.|)
*-	*-

Metric rhythm patterns (7)

```
menpat songs/Ock*.krn | grep -Pv "\s[^_]*r\s*$" |  sed 's/:.*//' | sortcount -ph
```

**pcent	**data
63.93	menCutC
19.02	menCircle
14.24	menC
2.81	m(O3)
*-	*-

**pcent	**data
93.84	menCutC
2.17	menC2
1.29	menCircle
1.25	menC
1.18	menCutC3
0.22	menCutC2
0.04	men3
*-	*-

```
menpat songs/Jos*.krn | grep -Pv "\s[^_]*r\s*$" |  sed 's/:.*//' | sortcount -ph
```