

Humdrum Analysis Tools I

craig@ccrma.stanford.edu

7 March 2013

Unix shell

- Humdrum Toolkit designed for use in a unix shell (terminal)

http://en.wikipedia.org/wiki/Unix_shell

linux: should be easy to find a terminal

OS X: /Applications/Utility/Terminal.app

Windows: not a unix system, so install unix emulator:

- * <http://www.cygwin.com>
- * or install your favorite Linux OS in VirtualBox (<http://www.virtualbox.org>)

- Some learning resources for terminal:
 - Humdrum User's Guide (see Documentation slide)
 - <http://www.ee.surrey.ac.uk/Teaching/Unix>
 - http://barelystable.com/tutorials/terminal/Terminal_Tutorial/Introduction.html
 - http://macdevcenter.com/pub/a/mac/2001/12/14/terminal_one.html
 - <http://www.youtube.com/watch?v=PYW7UG5VRgw>

Installing Humdrum tools

- Humdrum Toolkit:
<http://wiki.humdrum.org/wiki/downloads>
- Humdrum Extras:
<http://extras.humdrum.org/download>

Humdrum documentation

- Links to scans and HTML files for the User and Reference Guides are listed on <http://humdrum.ccarh.org>
- Main webpage for the Humdrum Toolkit:
<http://www.humdrum.org/Humdrum>
- Humdrum Wiki:
<http://wiki.humdrum.org>
- Humdrum Extras documentation:
<http://extras.humdrum.org/man>
- Humdrum Users' Group (**HUG):
<https://groups.google.com/forum/?fromgroups#!forum/starstarhug>

humdrum

- humdrum is a command which validates the Humdrum file format structure of a file.

humdrum file.krn

```
*kern  
1c  
*_
```

humdrum: ERROR 17: First exclusive interpretation record contains a non-exclusive interpretation in line 1, file file.krn.

```
*kern  
1c  
==
```

humdrum: ERROR 14: All spines have not been properly terminated in line 4, file file.krn.

```
! comment  
**kern  
1c  
*_
```

humdrum: ERROR 9: Local comment precedes first exclusive interpretation record in line 1, file file.krn.

proof

- proof is a command which validates the content of ****kern** data (more specific than humdrum command which validates structure)

```

**kern
*clefG2
*k[f#]
*M4/4
*MM120
=1-
1g
=2
2f#
=3
[1a
==
*_

```

proof file.krn

proof: Warning: Possible change of meter in measure 2, line 10
 proof: Warning: Possible change of meter in measure 3, line 12
 proof: Error: Incorrect tie specification in spine 1, line 11 "1a"

- -w option suppresses warnings and only lists errors.

proof -w file.krn

proof: Error: Incorrect tie specification in spine 1, line 11 "1a"

census

- census is a command which gives basic counting statistics on a Humdrum file
- -k option can be added to include **kern data counting.

```
**kern
*M3/4
=1-
2c
[4d
=2
4d]
4e
4r
==
*_
```

census file.krn

HUMDRUM DATA

Number of data tokens:	8
Number of null tokens:	0
Number of multiple-stops:	0
Number of data records:	8
Number of comments:	0
Number of interpretations:	3
Number of records:	11

KERN DATA

Number of note-heads:	4
Number of notes:	3
Longest note:	2
Shortest note:	4
Highest note:	e
Lowest note:	c
Number of rests:	1
Maximum number of voices:	1
Number of single barlines:	2
Number of double barlines:	1

humcat

- The humcat command can be used to download data from KernScores.
- All Humdrum Extras have built-in downloading capability (except when compiled natively for Windows) from the web, KernScores, and the Josquin Research Project.

humcat h://371chorales/chor001.krn | less

```

!!!COM: Bach, Johann Sebastian
!!!CDT: 1685/02/21/-1750/07/28/
!!!OTL@@DE:      Aus meines Herzens Grunde
!!!OTL@EN:      From the Depths of My Heart
!!!SCT: BWV 269
!!!PC#: 1
!!!AGN: chorale
**kern          **kern          **kern          **kern
*ICvox          *ICvox          *ICvox          *ICvox
*Ibass          *Itenor         *Ialto         *Isoprns
*I"Bass        *I"Tenor        *I"Alto        *I"Soprano
*>[A,A,B]      *>[A,A,B]      *>[A,A,B]      *>[A,A,B]
*>norep[A,B]   *>norep[A,B]   *>norep[A,B]   *>norep[A,B]
*>A            *>A            *>A            *>A
*clefF4        *clefGv2        *clefG2        *clefG2
*k[f#]         *k[f#]         *k[f#]         *k[f#]
*G:            *G:            *G:            *G:
*M3/4          *M3/4          *M3/4          *M3/4
*MM100         *MM100         *MM100         *MM100
4GG            4B            4d            4g
=1             =1             =1             =1
4G             4B            4d            2g
4E             8cL          4e            .
.              8BJ          .              .
4F#            4A            4d            4dd

```


humcat (2)

- The humcat command can be used to download and pipe data to standard Humdrum Toolkit commands:

```
humcat h://371chorales/chor001.krn | census -k
```

HUMDRUM DATA

Number of data tokens:	412
Number of null tokens:	91
Number of multiple-stops:	0
Number of data records:	103
Number of comments:	16
Number of interpretations:	14
Number of records:	133

KERN DATA

Number of note-heads:	229
Number of notes:	223
Longest note:	2.
Shortest note:	8
Highest note:	dd
Lowest note:	FF#
Number of rests:	0
Maximum number of voices:	4
Number of single barlines:	22
Number of double barlines:	1

humcat (3)

- humcat can also be used to stream multiple files from KernScores:

```
humcat -s h://371chorales | census -k
```

HUMDRUM DATA

Number of data tokens:	143760
Number of null tokens:	34224
Number of multiple-stops:	0
Number of data records:	35940
Number of comments:	5996
Number of interpretations:	4770
Number of records:	46706

KERN DATA

Number of note-heads:	86109
Number of notes:	84666
Longest note:	0
Shortest note:	32
Highest note:	aa
Lowest note:	CC
Number of rests:	783
Maximum number of voices:	4
Number of single barlines:	5291
Number of double barlines:	370

- For lots of data, better to download files for local use:

```
humcat -s h://371chorales | humsplit  
census -k chor*.krn
```

thru/thrux

- thru can be used to generate performance sequence from score sequence

```
**kern
*>[A,A,B]
*clefG2
*M2/4
*k[ ]
=1-
*>A
2c
=2: | !
*>B
2d
==
*_
```

thru file.krn

```
**kern
*thru
*clefG2
*M2/4
*k[ ]
=1-
*>A
2c
=2: | !
*>A
2c
=2: | !
*>B
2d
==
*_
```

thru/thrux (2)

- alternate thru sequences
- “norep” is a convention for the score without repeating sections

```

**kern
*>[A,A1,A,A2,B]
*>norep[A,A2,B]
*k[ ]
=1-
*>A
2c
=2
*>A1
2d
=3: | !
*>A2
2e
=4
*>B
2d
==
*_

```

thru -v norep file.krn

```

**kern
*thru
*k[ ]
=1-
*>A
2c
=2
*>A2
2e
=4
*>B
2d
==
*_

```

thru and census

census with repeated sections

thru *.krn | census -k

HUMDRUM DATA

Number of data tokens:	168723
Number of null tokens:	39889
Number of multiple-stops:	0
Number of data records:	42189
Number of comments:	5627
Number of interpretations:	6027
Number of records:	53843

KERN DATA

Number of note-heads:	101015
Number of notes:	99393
Longest note:	0
Shortest note:	32
Highest note:	aa
Lowest note:	CC
Number of rests:	889
Maximum number of voices:	4
Number of single barlines:	6365
Number of double barlines:	372

census without repeats

thru -v norep *.krn | census -k

HUMDRUM DATA

Number of data tokens:	143755
Number of null tokens:	34224
Number of multiple-stops:	0
Number of data records:	35944
Number of comments:	5626
Number of interpretations:	4784
Number of records:	46354

KERN DATA

Number of note-heads:	86104
Number of notes:	84661
Longest note:	0
Shortest note:	32
Highest note:	aa
Lowest note:	CC
Number of rests:	783
Maximum number of voices:	4
Number of single barlines:	5293
Number of double barlines:	371

for-loops in bash shell

```
for i in *.krn
do
    echo $i `census -k $i | grep "Number of notes"`
done
```

```
chor001.krn Number of notes: 223
chor002.krn Number of notes: 229
chor003.krn Number of notes: 196
chor004.krn Number of notes: 185
chor005.krn Number of notes: 330
chor006.krn Number of notes: 120
chor007.krn Number of notes: 346
chor008.krn Number of notes: 358
chor009.krn Number of notes: 238
...
```

sort

```
for i in *.krn
do
    echo $i `census -k $i | grep "Number of notes"`
done | sort -nrk5 | less
```

```
chor205.krn Number of notes: 957
chor132.krn Number of notes: 790
chor133.krn Number of notes: 600
chor197.krn Number of notes: 554
chor241.krn Number of notes: 517
chor215.krn Number of notes: 475
chor259.krn Number of notes: 471
chor091.krn Number of notes: 467
chor252.krn Number of notes: 458
chor214.krn Number of notes: 443
chor116.krn Number of notes: 429
chor069.krn Number of notes: 429
chor011.krn Number of notes: 414
chor277.krn Number of notes: 405
chor220.krn Number of notes: 380
```

...

sort options being used:

- n == sort numerically
- r == reverse order (largest first)
- k 5 == sort by 5th field on line

transpose

transpose -b 6

transpose -d 1 -c 2

transpose -k f#

```
**kern
*k[ ]
*C:
c
d
e
f
g
a
b
cc
*_
```

```
**kern
*Trd1c2
*k[ f#c# ]
*D:
d
e
f#
g
a
b
cc#
dd
*_
```

```
**kern
*Trd3c6
*k[ f#c#g#d#a#e# ]
*F#:
f#
g#
a#
b
cc#
dd#
ee#
ff#
*_
```


key/keycor

**kern	key file.krn
c	keycor file.krn
d	
e	transpose -b 23 file.krn key
f	
g	transpose -b 18 file.krn key
a	
b	keycor h://wtc/wtc1p01.krn
cc	keycor h://wtc/wtc1p02.krn
*_	keycor h://wtc/wtc1p03.krn
	keycor h://wtc/wtc1p04.krn

hum2mid

- Convert Humdrum file into a MIDI file.

```
hum2mid h://essen/asia/china/shanxi/shanx276.krn -o shanx276.mid
```

Some options:

-O == create a Type-0 MIDI file

--autopan == array parts in stereo field

--plus == store pitch spellings using MIDI+ method

--temperament == see <http://kern.ccarh.org/browse?l=temperament>

<http://extras.humdrum.org/man/hum2mid/>

myank

- “Measure Yank”: extracts selected measures from full score:

myank -m1-2 h://371chorales/chor032.krn

**kern	**kern	**kern	**kern
*clefF4	*clefGv2	*clefG2	*clefG2
*k[f#c#g#]	*k[f#c#g#]	*k[f#c#g#]	*k[f#c#g#]
*A:	*A:	*A:	*A:
*M4/4	*M4/4	*M4/4	*M4/4
*met(c)	*met(c)	*met(c)	*met(c)
*MM100	*MM100	*MM100	*MM100
=1-	=1-	=1-	=1-
8AL	4c#	4a	4ee
8BJ	.	.	.
8c#L	4c#	4a	4ee
8AJ	.	.	.
8DL	4d	4a	4ff#
8EJ	.	.	.
8F#L	4d	4a	4ff#
8DJ	.	.	.
=2	=2	=2	=2
2A;	2c#;	2a;	2ee;
4r	4ry	4ry	4r
4A	4f#	4a	4cc#
=3	=3	=3	=3
4G#	4e	4b	4dd
4A	4e	4a	4cc#
8EL	4e	4g#	4b
8DJ	.	.	.
8C#L	4e	4a	8cc#L
8AAJ	.	.	8ddJ
=	=	=	=
*_	*_	*_	*_

humplay

- Humplay will perform Humdrum files in real-time as it displays the file on screen.
- See <http://extra.humdrum.org/man/humplay> for real-time commands.

Try:

humplay h://beethoven/sonatas/sonata08-2.krn

humplay h://musedata/haydn/sym/sym101-1.krn

humplay h://wtc/wtc2/p18.krn

humplay h://371chorales/chor268.krn

extract/extractx

- Use the extract program to pull out a particular part from a full score:

```
humcat h://371chorales/chor043.krn | extract -f2
extractx -s2 h://371chorales/chor043.krn
extractx -g alto h://371chorales/chor043.krn
```

```
**kern
*ICvox
*Ialto
*I"Alto
*>[A,A,B]
*>norep[A,B]
*>A
*clefG2
*k[f#c#g#d#]
*E:
*M4/4
*met(c)
*MM100
=1-
2r
.
.
4r
8g#L
8g#J
=2
4a
8aL
8g#J
8g#L
16f#L
16eJJ
8f#L
8f#J
.
=3
```

- Use extractx if input data contains spine splits/joines (*^, *v).

prange

- Generate pitch histograms of input data (like census -k but more detail)

```
extractx -g alto h://371chorales/chor043.krn | prange
```

**keyno	**kern	**count
59	B	3
61	c#	1
63	e-	5
64	e	10
66	f#	19
68	g#	13
69	a	3
70	b-	1

*- *- *-

!!tessitura: 11 semitones
!!mean: 65.6 (f#)
!!median: 66 (f#)

Metric Position

- The beat command can be used to identify the beat number in measure for each data line.
- -a option used to append analysis data to input score.

beat -a h://371chorales/chor200.krn

**kern	**kern	**kern	**kern	**beat
=1-	=1-	=1-	=1-	=1-
4c	4e	4g	4cc	1
4B	4d	[4g	4g	2
4A	4c	8gL]	4a	3
.	.	8f#J	.	3.5
8GL	4d	4g	4b	4
8FJ	.	.	.	4.5
=2	=2	=2	=2	=2
4E	8cL	4.g	2cc	1
.	8B-J	.	.	1.5
4F	4A	.	.	2
.	.	[8fnX	.	2.5
4C;	4G;	8fL]	4cc;	3
.	.	8e;J	.	3.5
8cL	4c	[4e	4g	4
8B-J	.	.	.	4.5
=3	=3	=3	=3	=3

rid/ridx

- Rid program removed various categories of Humdrum file structure:
 - G == remove global comments (and reference records)
 - L == remove local comments
 - I == remove interpretations (and spine manipulators)
 - M = remove measure lines (ridx only)
 - d == remove null token data lines

```
beat h://371chorales/chor200.krn | ridx -GLIMd
```

```
1  
2  
3  
3.5  
4  
4.5  
1  
1.5  
2  
2.5  
3  
3.5  
4  
...
```


uniq -c

```
beat h://371chorales/chor200.krn | ridx -GLIMd | sort -nr | uniq -c
```

15	1
10	1.5
1	1.75
11	2
9	2.5
15	3
11	3.5
1	3.75
13	4
12	4.5
2	4.75

- 15 times an event on beat 1 (could be tied note or rest)
- 10 times an event on beat 1.5
- Type “man uniq” to see the manual page for the uniq command.

Sonority analysis

```
humcat -s h://371chorales | humsplit
for i in chor*.krn
do
    beat -a $i | tntype -a | extract -f 4,5
done | rid -GLld | grep -v = | sort -n | uniq -c | less
```

6	1	1-1								
	1	2-1		2	1.5	2-2		5	1.5	4-18A
44	1	2-3		24	1.5	2-3		5	1.5	4-18B
45	1	2-4		41	1.5	2-4		15	1.5	4-19A
	3	2-5		49	1.5	2-5		10	1.5	4-19B
99	1	3-10		185	1.5	3-10		195	1.5	4-20
1182	1	3-11A		288	1.5	3-11A		8	1.5	4-21
2380	1	3-11B		350	1.5	3-11B		148	1.5	4-22A
	16	3-12		14	1.5	3-12		72	1.5	4-22B
	1	3-2A		3	1.5	3-2B		142	1.5	4-23
	3	3-2B		26	1.5	3-4A		7	1.5	4-24
27	1	3-4A		19	1.5	3-4B		203	1.5	4-26
	4	3-6		4	1.5	3-5A		86	1.5	4-27A
82	1	3-7A		1	1.5	3-5B		245	1.5	4-27B
	1	3-7B		3	1.5	3-6		28	1.5	4-28
	16	3-8A		130	1.5	3-7A		6	1.5	4-29A
239	1	3-9		85	1.5	3-7B		20	1.5	4-29B
	7	4-11A		28	1.5	3-8A		1	1.5	4-7
	1	4-11B		19	1.5	3-8B		1	1.5	4-8
	1	4-13A		146	1.5	3-9				
103	1	4-14A		17	1.5	4-10				
	1	4-19A		18	1.5	4-11A				
	2	4-19B		22	1.5	4-11B				
45	1	4-20		1	1.5	4-12A				
79	1	4-22A		3	1.5	4-12B				
	13	4-23		7	1.5	4-13A				
242	1	4-26		5	1.5	4-13B				
136	1	4-27A		91	1.5	4-14A				
198	1	4-27B		85	1.5	4-14B				
	32	4-28		5	1.5	4-16A				
	1	4-29A		8	1.5	4-16B				
				1	1.5	4-17				