Harmony II

craig@ccrma.stanford.edu

26/28 April 2016

keycor

keycor h://chopin/preludes -s -l --errors

```
prelude28-01.krn:   C Major
prelude28-02.krn:   A Minor
prelude28-03.krn:   G Major
prelude28-04.krn:   E Minor
prelude28-05.krn:   D Major
prelude28-06.krn:   B Minor
prelude28-07.krn:   A Major
prelude28-08.krn:   F# Minor
prelude28-09.krn:   E Major
prelude28-10.krn:   C# Minor
prelude28-11.krn:   B Major
prelude28-12.krn:   G# Minor
prelude28-13.krn:   F# Major
prelude28-14.krn:   E- Minor
prelude28-15.krn:   G# Minor X:D- major
prelude28-16.krn:   B- Minor
prelude28-17.krn:   A- Major
prelude28-18.krn:   F Minor
prelude28-19.krn:   E- Major
prelude28-20.krn:   C Minor
prelude28-21.krn:   B- Minor X:B- major
prelude28-22.krn:   G Minor
prelude28-23.krn:   F Major
prelude28-24.krn:   D Minor
```

-s == simple weights
--aa == Aarden-Essen weights
--bb == Bellman-Budge weights
--kk == Krumhansl-Kessler weights
--kp == Kostka-Payne weights

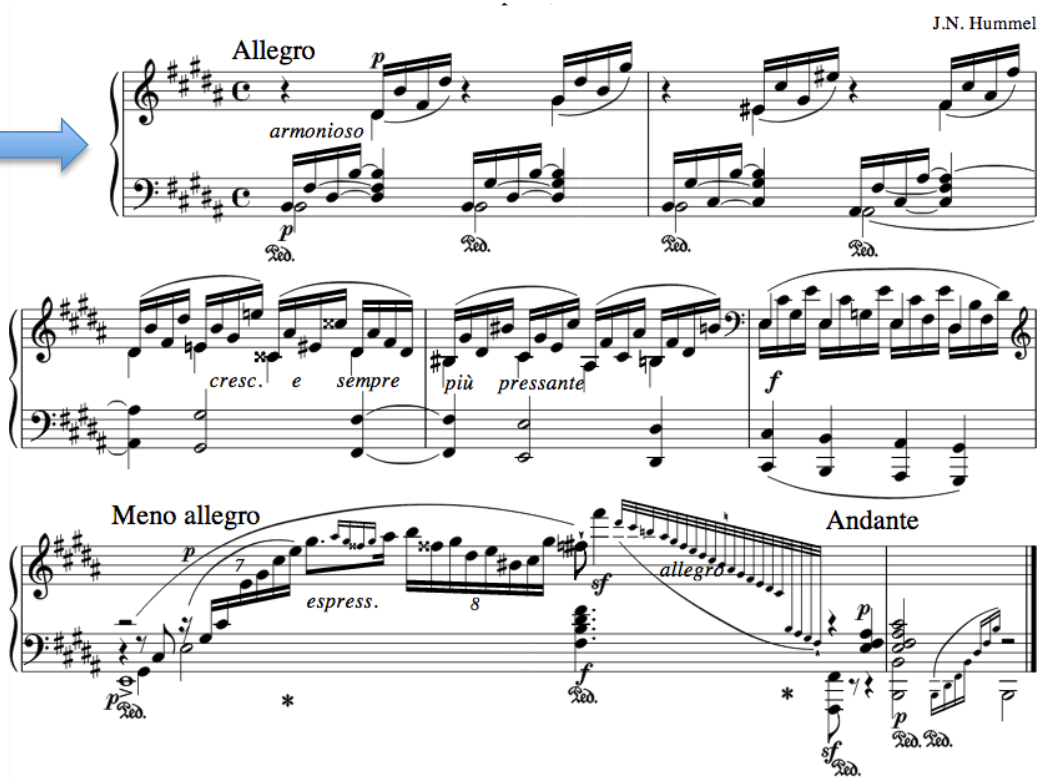
-l == show filenames
--errors == mark errors

http://extras.humdrum.org/man/keycor

Hummel preludes, Op. 67

keycor h://hummel/preludes --bb -l --errors

prelude67-01.krn: C Major
prelude67-02.krn: D Minor X:A minor
prelude67-03.krn: G Major
prelude67-04.krn: E Minor
prelude67-05.krn: D Major
prelude67-06.krn: B Minor
prelude67-07.krn: A Major
prelude67-08.krn: F# Minor
prelude67-09.krn: E Major
prelude67-10.krn: C# Minor
prelude67-11.krn: B Major
prelude67-12.krn: G# Minor
prelude67-13.krn: F# Major
prelude67-14.krn: E- Minor
prelude67-15.krn: D- Major
prelude67-16.krn: B- Minor
prelude67-17.krn: A- Major
prelude67-18.krn: F Minor
prelude67-19.krn: E- Major
prelude67-20.krn: C Minor
prelude67-21.krn: B- Major
prelude67-22.krn: G Minor
prelude67-23.krn: F Major
prelude67-24.krn: D Minor



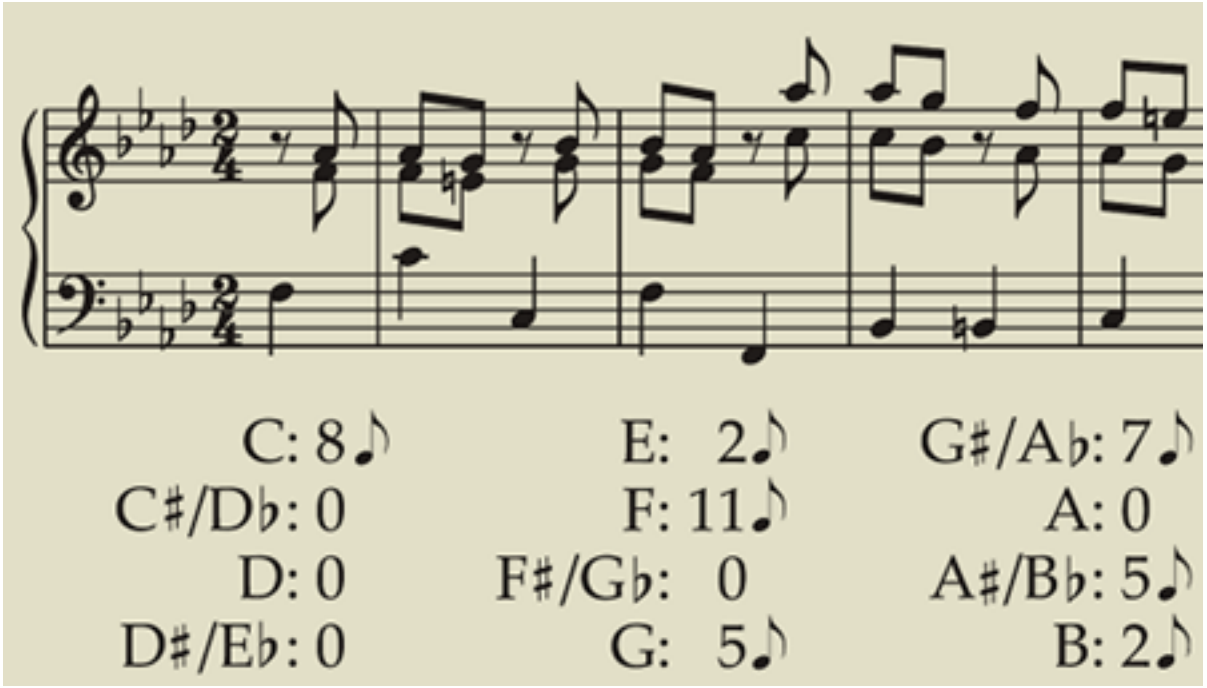
Krumhansl-Schmuckler key-finding algorithm

- The best key assignment is the key which generates the highest Pearson correlation between a pitch histogram extracted from the music compared to major/minor key prototypes

Pearson correlation (normalized correlation):

$$R(x,y) = \frac{\sum (x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum (x_n - \bar{x})^2 \sum (y_n - \bar{y})^2}}$$

Pitch-class histogram



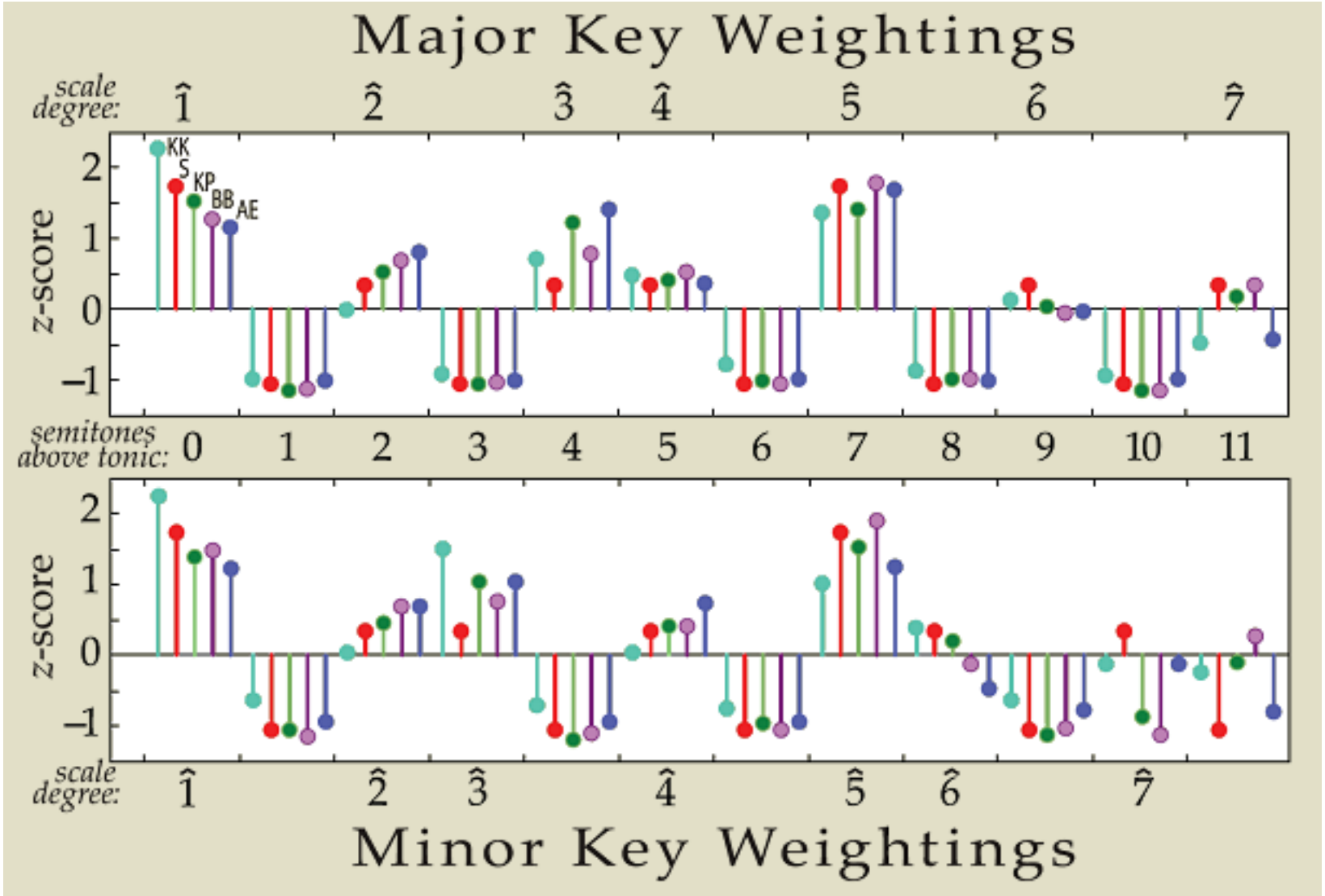
extracted histogram as a 12-dimensional vector: (8, 0, 0, 0, 2, 11, 0, 5, 7, 0, 5, 2)

Prototype weights

scale degree	*semi-tones above tonic*	KK	AE	KP	BB	LC
		Major Key Weights				
1	0	6.35	17.77	0.748	16.80	2
	1	2.23	0.15	0.060	0.86	0
2	2	3.48	14.93	0.488	12.95	1
	3	2.33	0.16	0.082	1.41	0
3	4	4.38	19.80	0.670	13.49	1
4	5	4.09	11.36	0.460	11.93	1
	6	2.52	0.29	0.096	1.25	0
5	7	5.19	22.06	0.715	20.28	2
	8	2.39	0.15	0.104	1.80	0
6	9	3.66	8.15	0.366	8.04	1
	10	2.29	0.23	0.057	0.62	0
7	11	2.88	4.95	0.400	10.57	1
		Minor Key Weights				
1	0	6.33	18.26	0.712	18.16	2
	1	2.68	0.74	0.084	0.69	0
2	2	3.52	14.05	0.474	12.99	1
3	3	5.38	16.86	0.618	13.34	1
	4	2.60	0.70	0.049	1.07	0
4	5	3.53	14.44	0.460	11.15	1
	6	2.54	0.70	0.105	1.38	0
5	7	4.75	18.62	0.747	21.07	2
6	8	3.98	4.57	0.404	7.49	1
	9	2.69	1.93	0.067	1.53	0
7	10	3.34	7.38	0.133	0.92	1
	11	3.17	1.76	0.330	10.21	0

Prototype comparisons

Unnormalized major-key simple weights: (2, 0, 1, 0, 1, 1, 0, 2, 0, 1, 0, 1)



KS algorithm (2)

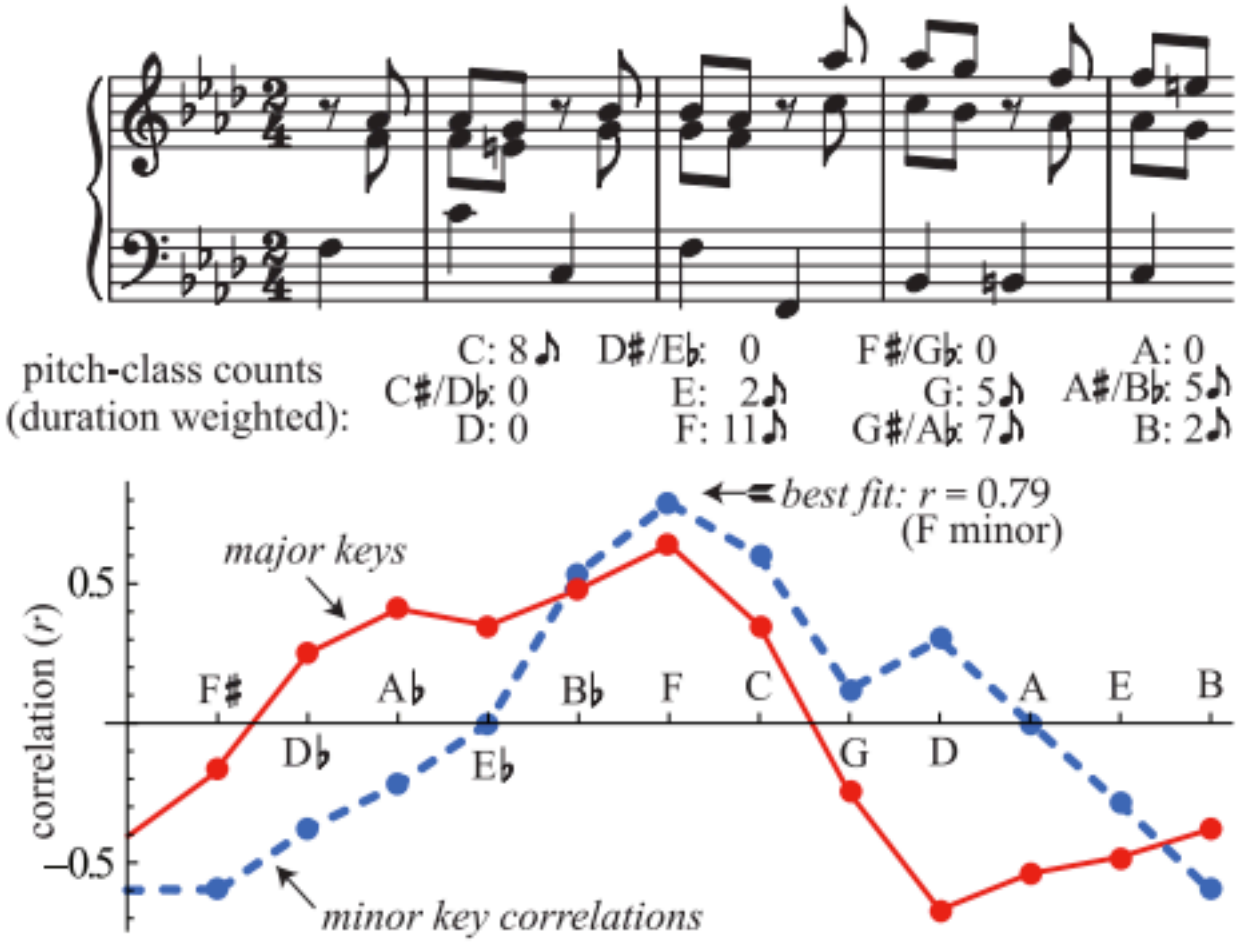
$$R(x,y) = \frac{\sum (x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum (x_n - \bar{x})^2 \sum (y_n - \bar{y})^2}}$$

$$\text{key}_k = \arg\max_k R(x, y_k)$$

$$\text{key}_k = \arg\max_k \frac{1}{N} \sum_n z_{x,n} \; z_{y_k,n}$$

dot product of normalized vectors

Key correlation spectrum



pitch-class histogram: $x = (8, 0, 0, 0, 2, 11, 0, 5, 7, 0, 5, 2)$

major key prototype: $y_M = (2, 0, 1, 0, 1, 1, 0, 2, 0, 1, 0, 1)$

minor key prototype: $y_m = (2, 0, 1, 1, 0, 1, 0, 2, 1, 0, 1, 0)$

Key correlation spectrum (2)

keycor -a -s h://wtc/wtc2p12.krn

The best key is: F Minor

Major[0] = -0.0469801	Minor[0] = 0.629898
Major[1] = 0.664563	Minor[1] = -0.138204
Major[2] = -0.853396	Minor[2] = -0.194762
Major[3] = 0.631723	Minor[3] = 0.356228
Major[4] = -0.550534	Minor[4] = -0.729332
Major[5] = 0.372648	Minor[5] = 0.934585
Major[6] = 0.115398	Minor[6] = -0.556007
Major[7] = -0.590672	Minor[7] = 0.0935041
Major[8] = 0.808697	Minor[8] = 0.126345
Major[9] = -0.700141	Minor[9] = -0.430119
Major[10] = 0.476643	Minor[10] = 0.774032
Major[11] = -0.327949	Minor[11] = -0.866167

-a == show all key weights [0] = C, [1] == C#/D-flat, etc.

Evaluate all key-cycle pieces

- Download all key-cycle works:

  humcat -s h://wtc h://chopin/preludes h://hummel/preludes > keycycles.krns

- wtc: 48 prelues, 48 fugues chopin: 24 prelues, hummel: 24 preludes: 144 pieces
- Evaluate various key prototypes:

  keycor –l --errors --bellman keycycles.krns | grep X: | wc –l
  10

  keycor –l --errors --aarden keycycles.krns | grep X: | wc –l
  8

  keycor –l --errors --kk keycycles.krns | grep X: | wc –l
  13

  keycor –l --errors --simple keycycles.krns | grep X: | wc –l
  7

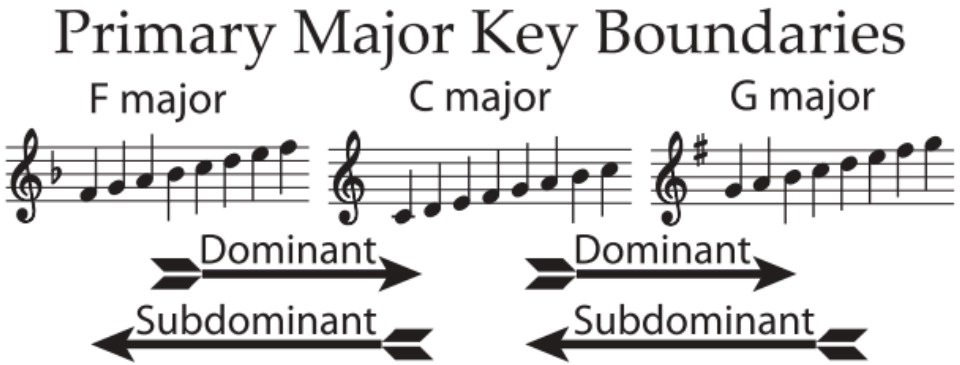
  keycor –l --errors --kp keycycles.krns | grep X: | wc –l
  9

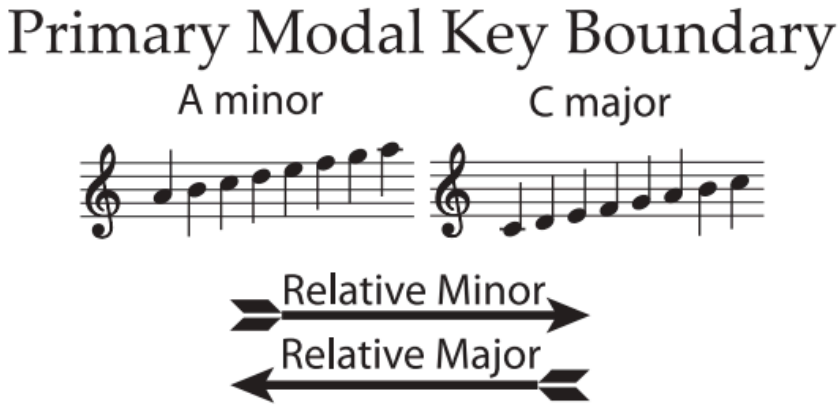
Types of identification errors

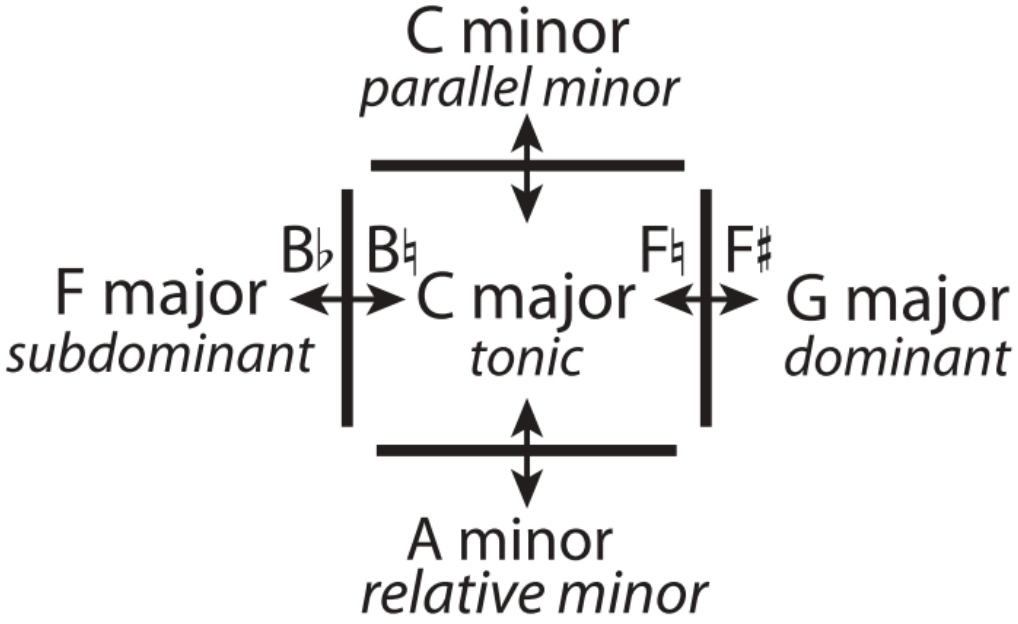
	J.S. Bach, The Well-tempered Clavier																				Chopin preludes op. 28					
	Book I										Book II															
	preludes					fugues					preludes					fugues										
key	AE	BB	KK	KP	S	AE	BB	KK	KP	S	AE	BB	KK	KP	S	AE	BB	KK	KP	S	AE	BB	KK	KP	S	*key*
C maj																										C maj
c min												rel		rel												c min
C♯maj								dom														para	dom	para	dom	C♯maj
c♯min																										c♯min
D maj																										D maj
d min																										d min
E♭maj																										E♭maj
e♭min																										e♭min
E maj																					para	para	para	para		E maj
e min							rel		rel																	e min
F maj	rel	rel	rel	rel	rel					rel	rel												dom			F maj
f min												rel														f min
F♯maj																										F♯maj
f♯min																										f♯min
G maj												dom					dom									G maj
g min																										g min
A♭maj																										A♭maj
g♯min																										g♯min
A maj																rel		dom								A maj
a min							rel		rel													D		D		a min
B♭maj			dom																		e♭	e♭			para	para	B♭maj
b♭min																										b♭min
B maj												dom				rel										B maj
b min																		f♯		rel						b min
errors	1	1	2	1	1	0	2	1	2	1	1	2	2	1	0	2	0	3	1	0	2	4	4	3	2	

cumulative errors: AE: 6/120 (5.0%) BB: 9/120 (7.5%) KK: 12/120 (10.0%) KP: 8/120 (6.7%) S: 4/120 (3.3%)

Key confusions







Evaluate Beethoven piano sonatas

(102 movements)

```
keycor -l --errors -s h://beethoven/sonatas | grep X: | wc -l
3

keycor -l --errors --aa h://beethoven/sonatas | grep X: | wc –l
6

keycor -l --errors --kp h://beethoven/sonatas | grep X: | wc –l
5

keycor -l --errors --bb h://beethoven/sonatas | grep X: | wc –l
5

keycor -l --errors --kk h://beethoven/sonatas | grep X: | wc –l
29
```

Evaluate Mozart piano sonatas

(69 movements)

keycor -l --errors -s h://mozart/sonatas | grep X: | wc –l
2

keycor -l --errors --aa h://mozart/sonatas | grep X: | wc –l
1

keycor -l --errors --kp h://mozart/sonatas | grep X: | wc –l
1

keycor -l --errors --bb h://mozart/sonatas | grep X: | wc –l
2

keycor -l --errors --kk h://mozart/sonatas | grep X: | wc –l
10

Evaluate Scarlatti piano sonatas

(59 sonatas)

keycor -l --errors -s h://scarlatti/sonatas | grep X: | wc –l
11

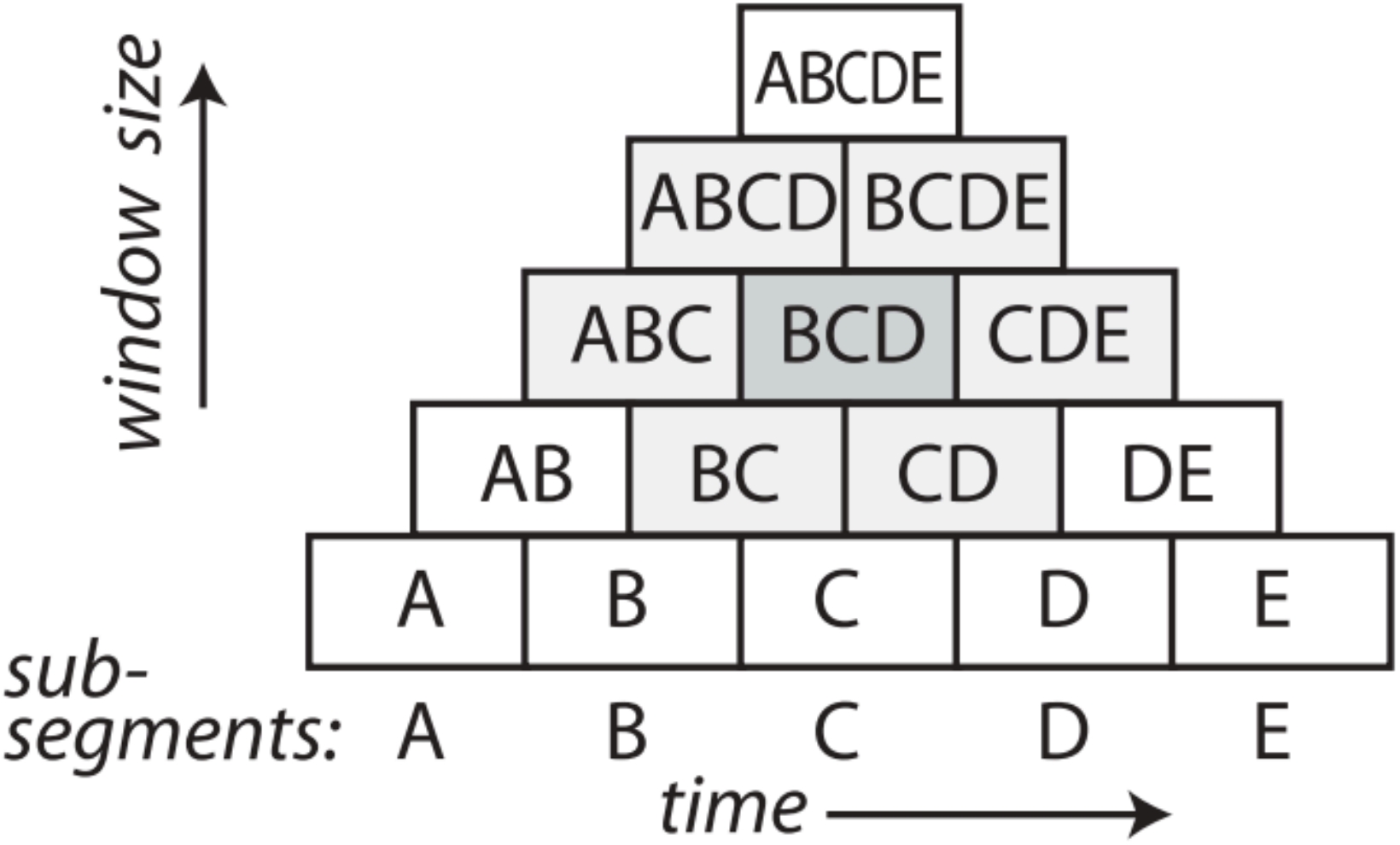
keycor -l --errors --aa h://scarlatti/sonatas | grep X: | wc –l
6

keycor -l --errors --kp h://scarlatti/sonatas | grep X: | wc –l
4

keycor -l --errors --bb h://scarlatti/sonatas | grep X: | wc –l
6

keycor -l --errors --kk h://scarlatti/sonatas | grep X: | wc –l
26

Keyscape plotting domain

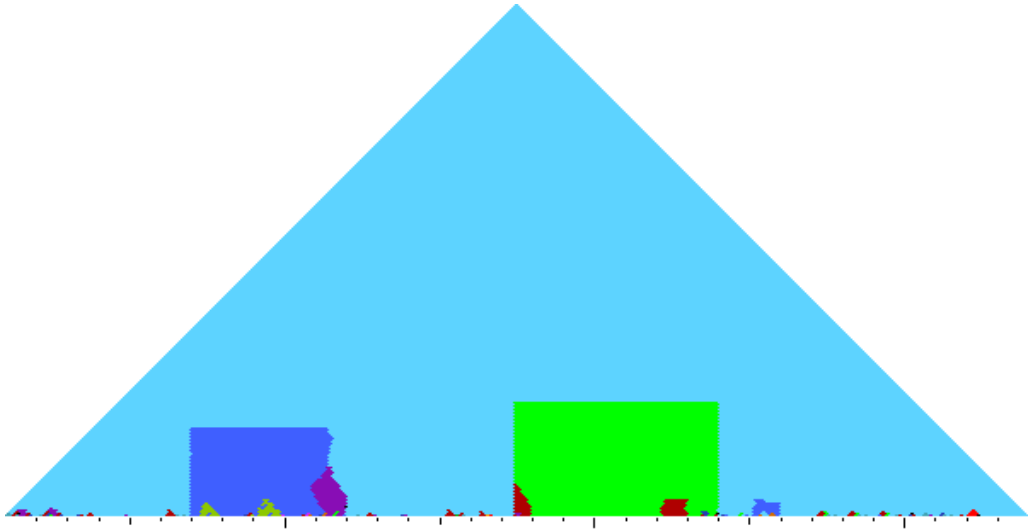


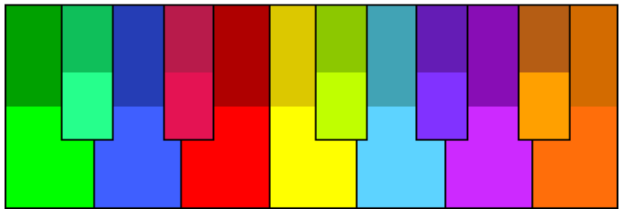
mkeyscape

http://extras.humdrum.org/man/mkeyscape

mkeyscape == “MIDI-aware keyscape program” (can read MIDI files and Humdrum files)

mkeyscape h://chopin/preludes/prelude28-03.krn -n --trim -l --aa | convert - z.png

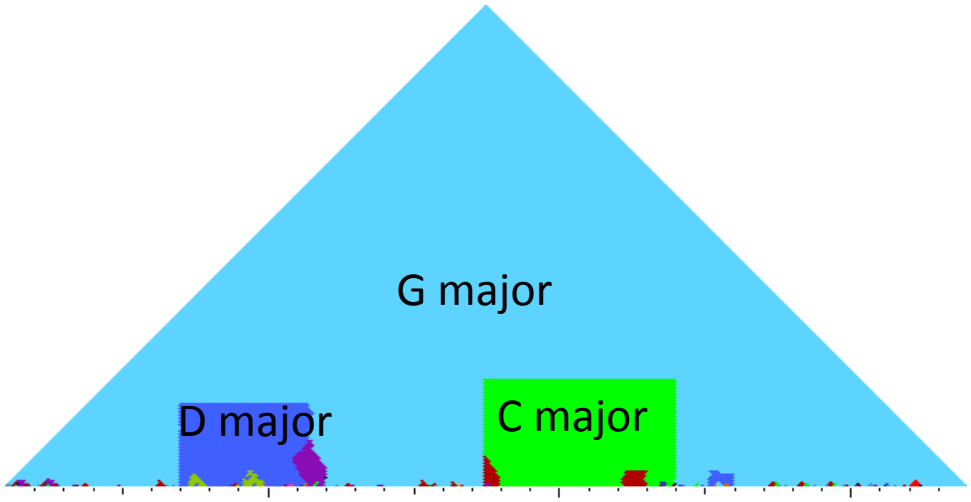




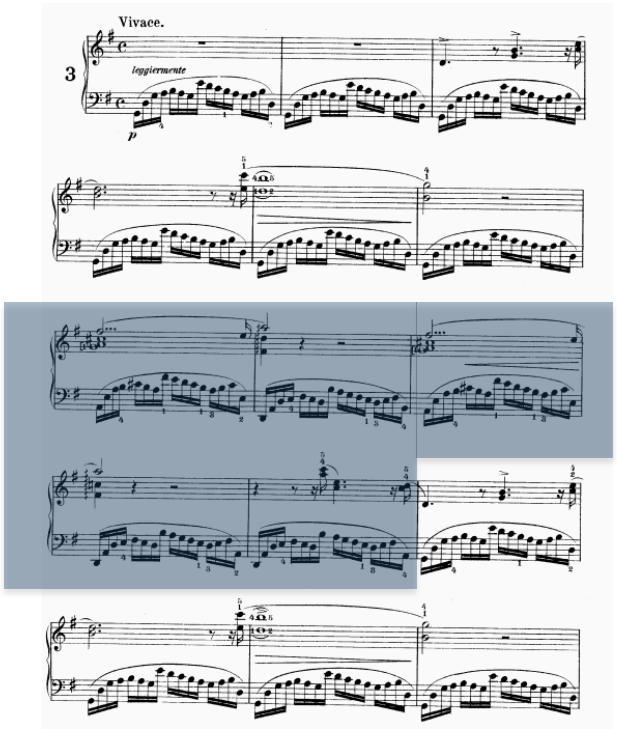
-n == show measure ticks
--trim == post clean-up
-l == show color mapping legend
--aa == Use Aarden-Essen weights

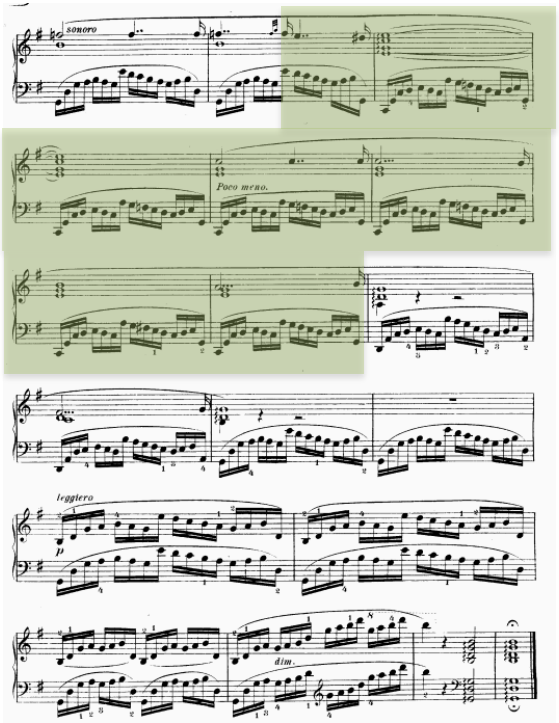
convert == ImageMagick image processing program (mkeyscape output is type PPM)

Chopin Prelude #3, Op. 28

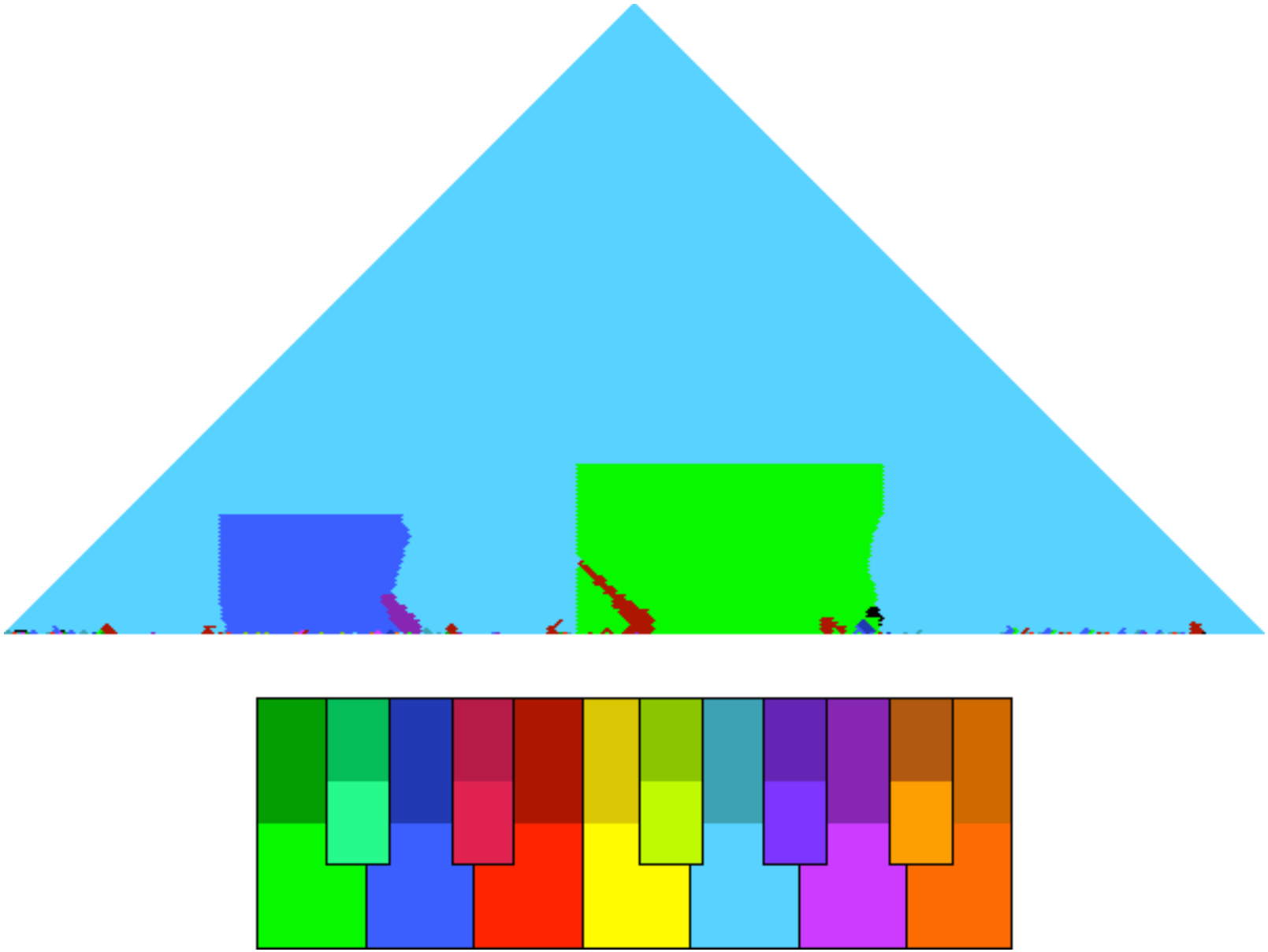


D major





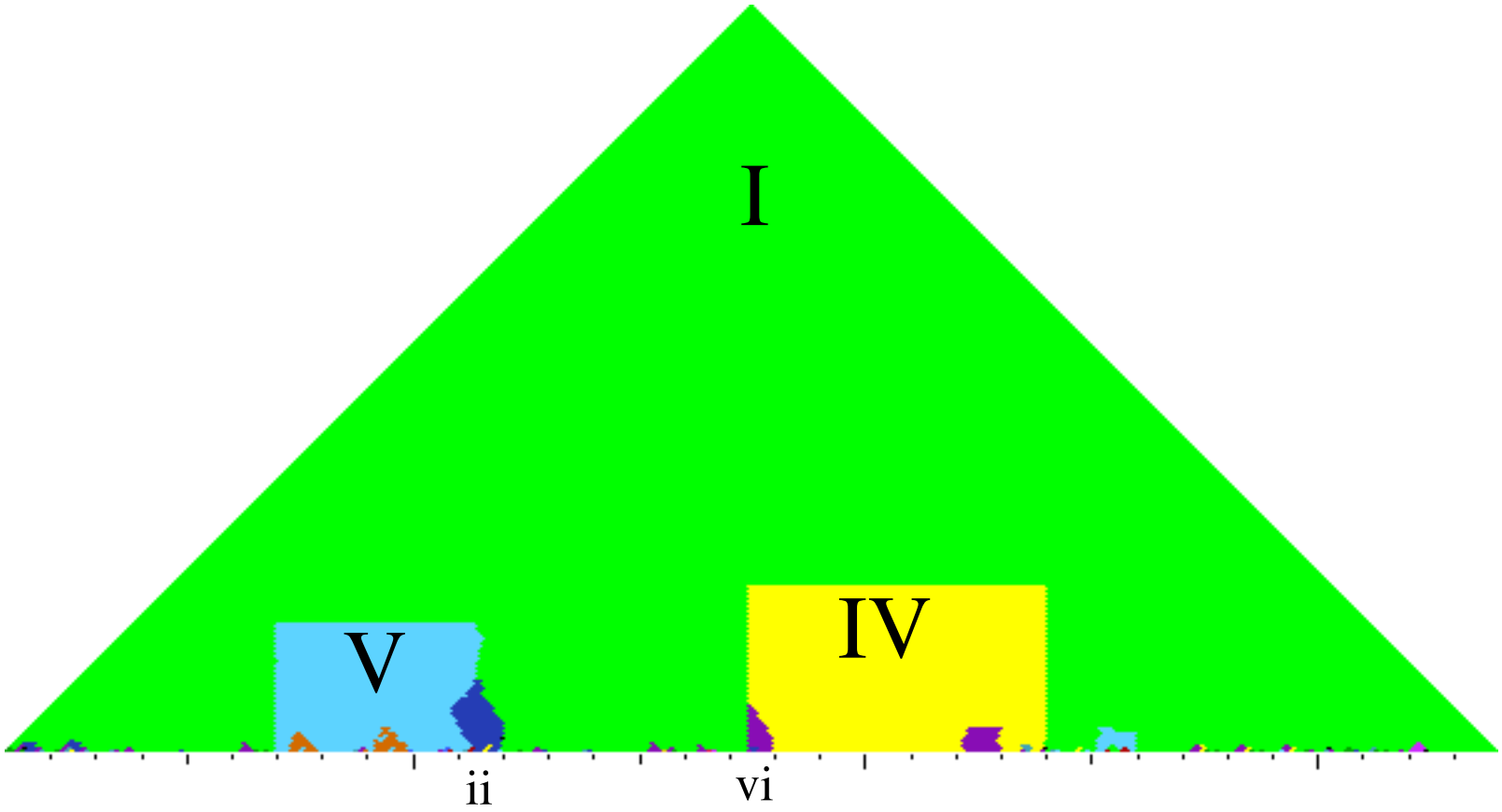
C major



mkeyscape --trim -l --bb h://chopin/preludes/prelude28-03.krn | convert - z.png

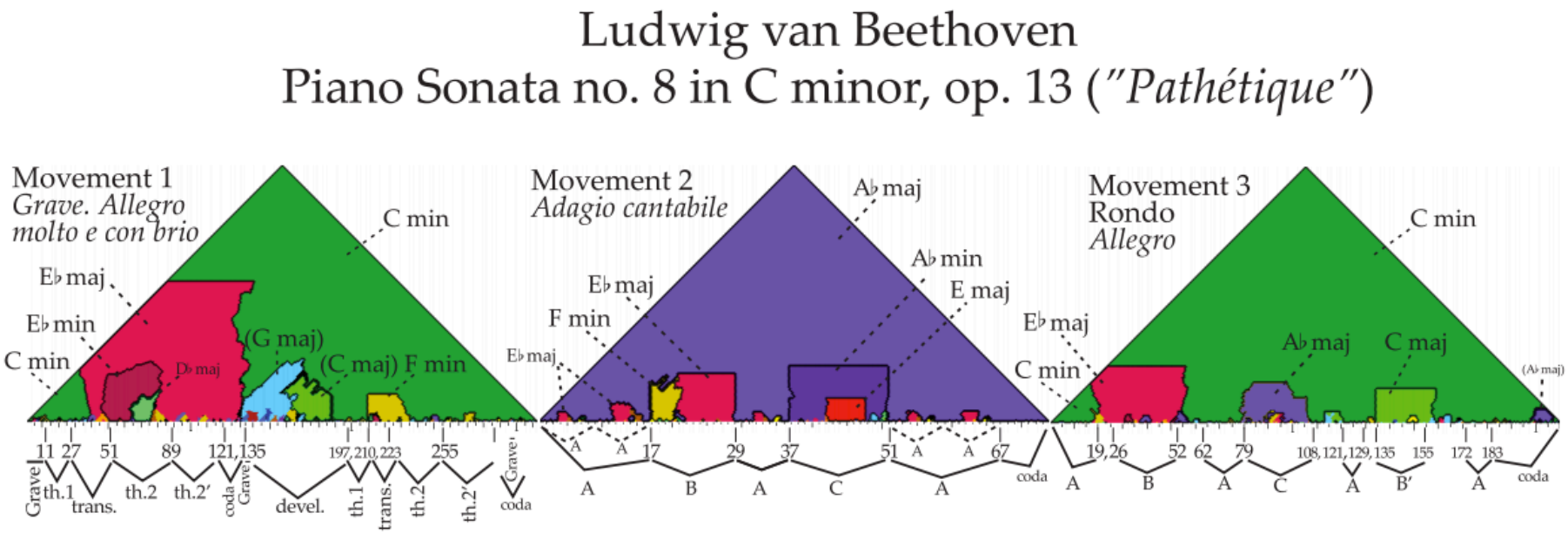
Functional coloring

transpose -k C h://chopin/preludes/prelude28-03.krn | mkeyscape -n --trim --aa | convert - z.png



http://extras.humdrum.org/man/transpose

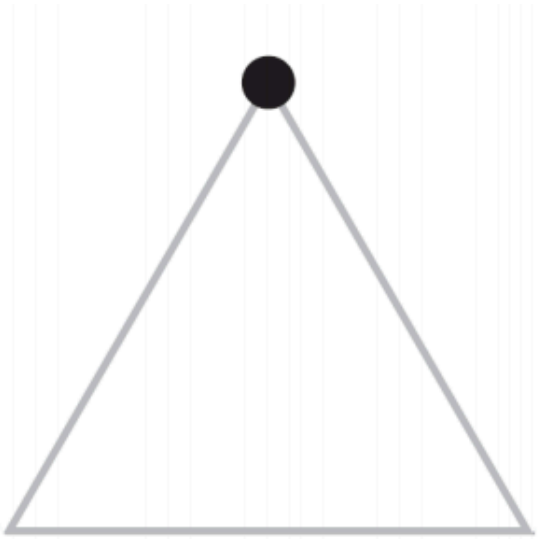
Sonata movements



http://extras.humdrum.org/man/mkeyscape/beet-sonata

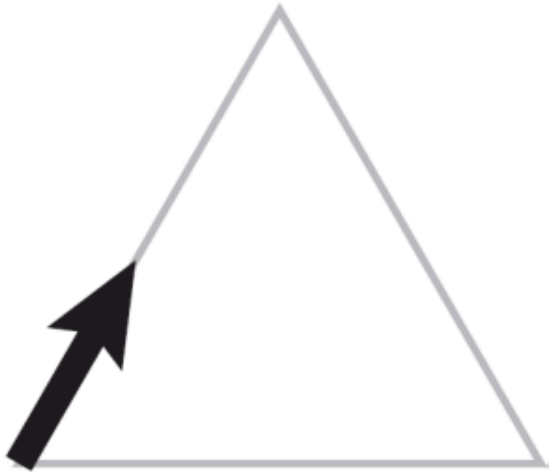
http://extras.humdrum.org/man/mkeyscape/moz-sonata

Key evaluation methods



Top-Level Test

high-data behavior



Leading-Edge Test

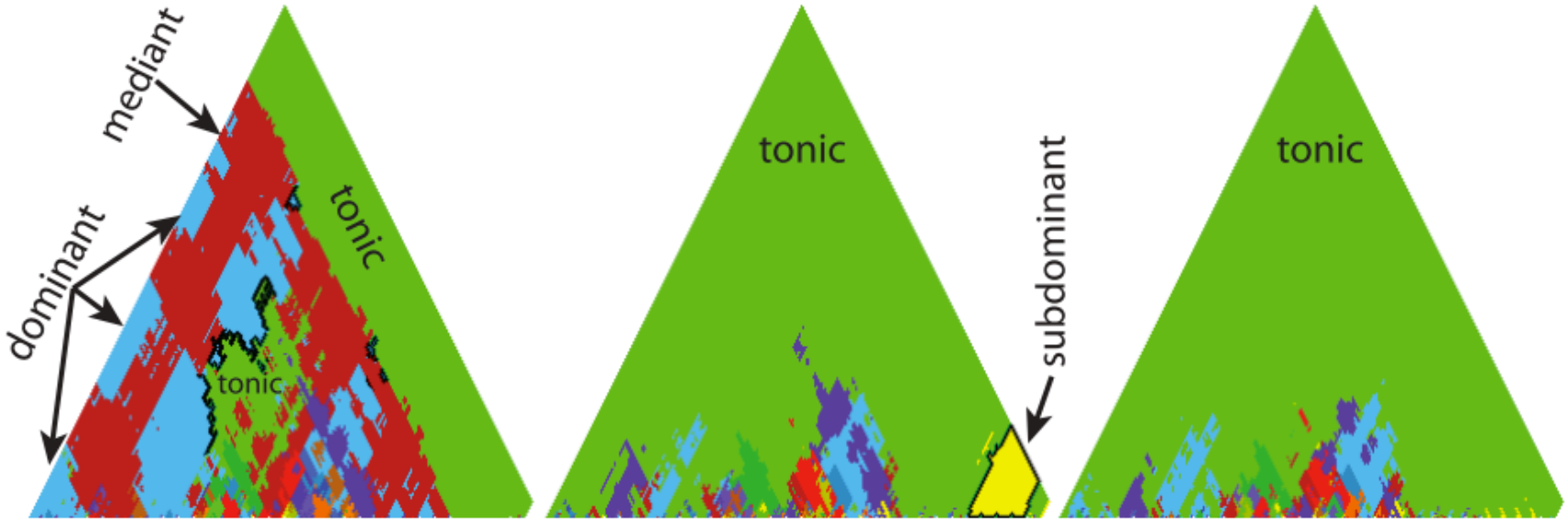
low-data behavior

(& trailing edge)

Tonic-Area Test

intermediate behavior

Keyscapes by weight



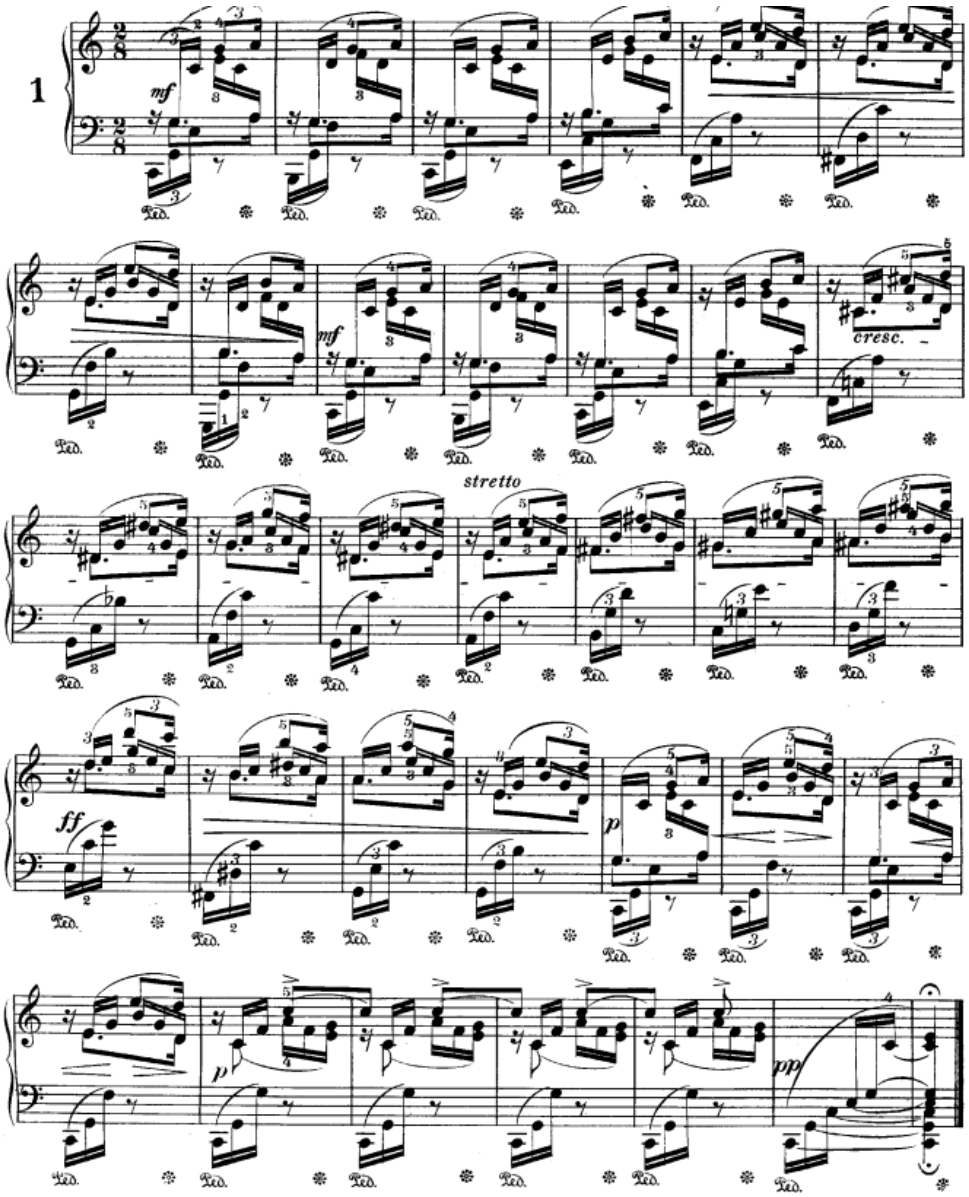
Krumhansl-Kessler weights

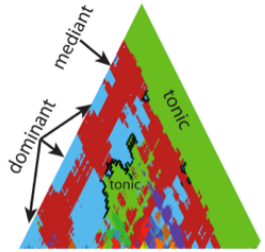
Aarden-Essen weights

Kostka-Payne weights

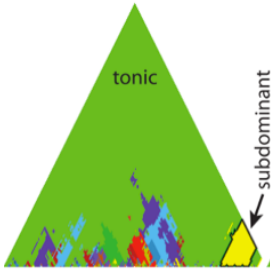
Chopin Prelude No. 1, Op. 28 (C major)

Chopin prelude in C major 28/1

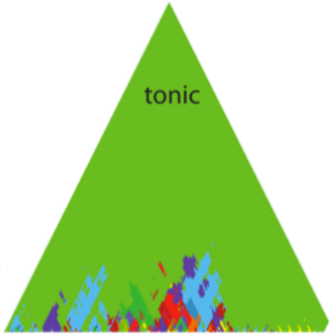




Krumhansl-Kessler weights



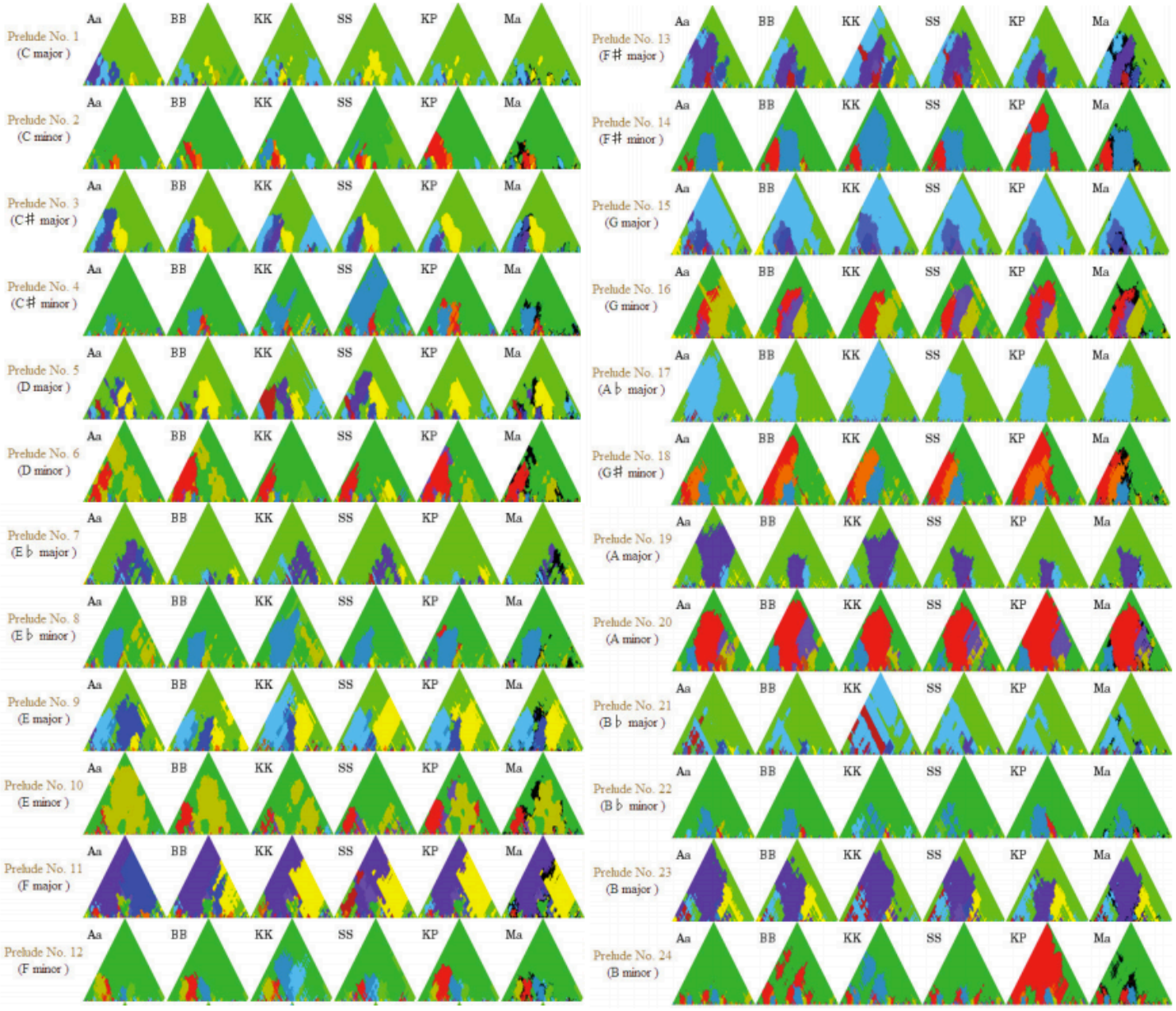
Aarden-Essen weights



Kostka-Payne weights

Keyscapes of WTC Book I preludes

http://extras.humdrum.org/man/mkeyscape/wtc



Tonic-area evaluation

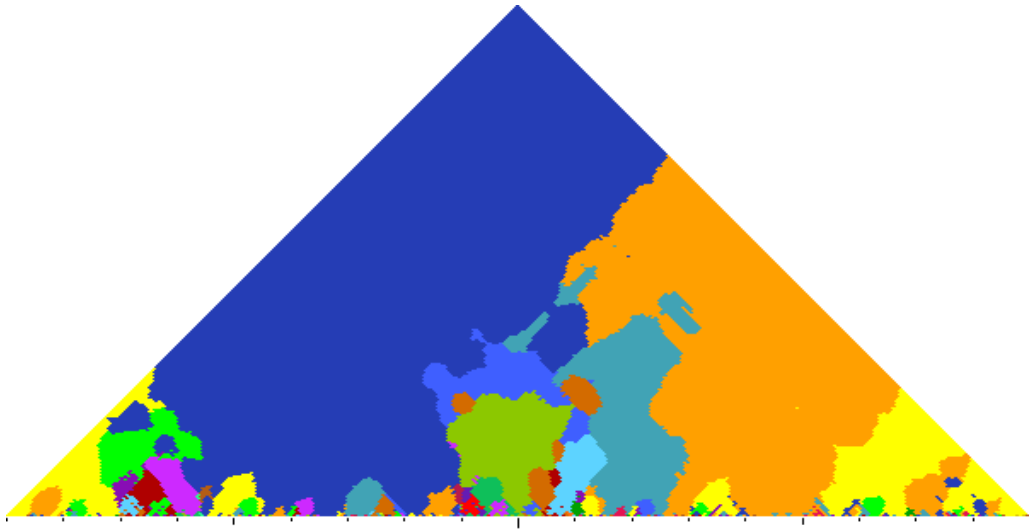
	J.S. Bach, The Well-tempered Clavier																						Chopin preludes op. 28					
	Book I												Book II															
	preludes						fugues						preludes						fugues									
key	AE	BB	KK	KP	S	M	AE	BB	KK	KP	S	M	AE	BB	KK	KP	S	M	AE	BB	KK	KP	S	M	AE	BB	KK	KP	S	M
C maj	74	74	70	82	74	80	43	59	32	80	43	46	20	23	26	26	24	22	85	88	77	92	83	87	84	83	37	87	85	86
c min	87	85	76	74	65	81	88	81	82	36	83	83	14	11	18	10	16	13	90	91	90	72	90	90	83	86	80	83	77	82
C♯maj	63	67	50	68	68	67	65	73	31	75	67	69	79	82	34	81	72	79	92	91	85	94	88	92	39	20	2	24	29	23
c♯min	86	81	64	73	37	75	89	78	88	37	88	87	65	47	55	29	60	56	84	51	57	10	61	57	80	85	39	81	72	80
D maj	64	63	50	68	54	62	60	77	74	76	79	75	48	62	34	61	53	54	30	76	37	83	70	67	54	77	41	78	72	71
d min	47	51	75	56	70	62	84	86	71	79	73	82	75	76	56	59	77	73	90	90	89	77	89	89	55	55	32	45	43	46
E♭maj	64	86	67	88	76	76	46	71	50	88	52	56	65	86	49	84	78	78	90	91	46	89	85	89	72	73	60	77	74	73
e♭min	56	72	33	64	71	65	78	75	80	26	83	76	61	43	37	27	57	47	38	25	30	9	65	31	59	60	59	52	42	55
E maj	38	51	28	39	41	39	53	74	46	89	62	65	52	68	45	65	66	62	70	83	37	92	57	72	11	13	8	14	41	12
e min	34	46	61	48	71	49	26	11	26	7	13	13	60	51	34	26	59	51	73	46	62	19	59	57	48	65	49	67	81	62
F maj	3	7	6	7	8	6	24	44	12	67	26	31	10	33	12	59	25	24	67	73	67	79	74	73	82	84	38	77	65	77
f min	79	79	53	70	71	74	77	61	56	19	70	62	26	28	36	24	33	29	75	66	51	25	54	57	68	71	48	69	67	68
F♯maj	41	57	27	59	43	45	34	63	35	69	50	50	40	65	28	72	50	51	40	79	56	79	70	69	85	86	75	89	84	87
f♯min	73	62	46	45	64	60	90	87	82	36	84	85	70	64	63	46	76	67	82	42	76	3	80	74	44	65	19	55	32	46
G maj	18	23	16	21	25	21	32	52	32	63	53	51	62	74	27	70	65	67	43	69	8	77	56	55	61	53	46	63	58	58
g min	46	56	51	51	40	48	65	52	71	23	68	62	78	76	70	64	73	74	56	23	59	10	71	49	31	46	37	47	30	38
A♭maj	45	65	35	57	64	56	44	78	47	81	71	70	32	57	30	55	43	44	78	81	77	78	77	79	23	24	34	33	51	32
g♯min	59	46	46	38	54	50	71	64	50	10	72	65	40	38	25	26	32	33	93	88	85	22	87	89	56	58	54	52	55	56
A maj	40	75	48	78	74	71	53	86	63	89	78	79	50	59	67	69	60	62	19	65	11	71	37	38	86	87	61	85	84	86
a min	34	22	38	18	29	27	69	27	57	4	66	55	73	72	56	56	62	67	77	67	79	22	75	75	16	24	17	20	16	21
B♭maj	74	76	20	69	58	69	29	54	28	72	37	39	58	73	41	75	58	66	58	81	63	79	70	72	22	24	27	31	30	28
b♭min	81	84	78	78	79	82	37	32	70	24	65	40	52	33	51	23	50	45	79	38	73	9	61	64	67	72	60	49	69	66
B maj	25	40	31	52	38	37	70	84	54	84	77	79	45	65	20	61	50	52	25	82	59	88	66	72	94	94	68	93	90	92
b min	84	69	66	25	83	74	78	53	68	7	71	67	41	28	35	21	51	35	53	15	23	5	36	21	68	83	77	74	66	73

Tonic-area evaluation (2)

= best result = better than simple weights

	weight set:	AE	BB	KK	KP	SS	Majority
Major keys:	mean	0.70	0.90	0.58	**0.96**	0.83	0.82
	median	0.73	0.96	0.57	**1.00**	0.88	0.88
	sd	0.22	0.14	0.23	**0.11**	0.15	0.16
Minor keys:	mean	**0.91**	0.81	0.81	0.56	0.87	0.84
	median	**0.99**	0.88	0.84	0.61	0.93	0.87
	sd	0.14	0.21	0.17	0.30	0.14	**0.13**
Major and Minor combined:	mean	0.81	**0.86**	0.69	0.76	0.85	0.83
	median	0.91	**0.93**	0.72	0.90	0.90	0.87
	sd	0.21	0.18	0.23	0.30	0.14	**0.14**

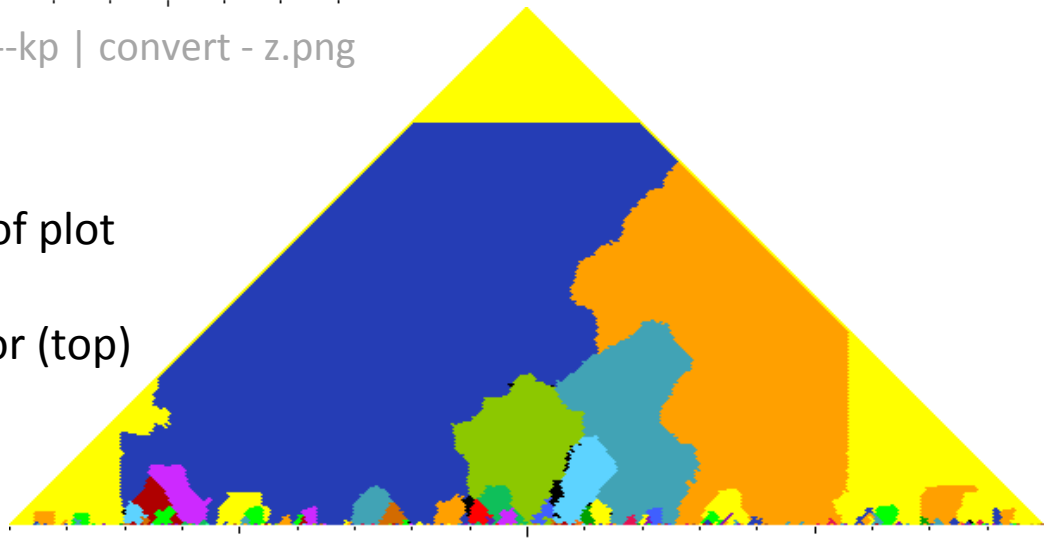
WTC1p11 (F major prelude)



- mostly secondary keys
- 3-8% of keyscape is tonic (yellow)
- blue = D minor (relative minor)

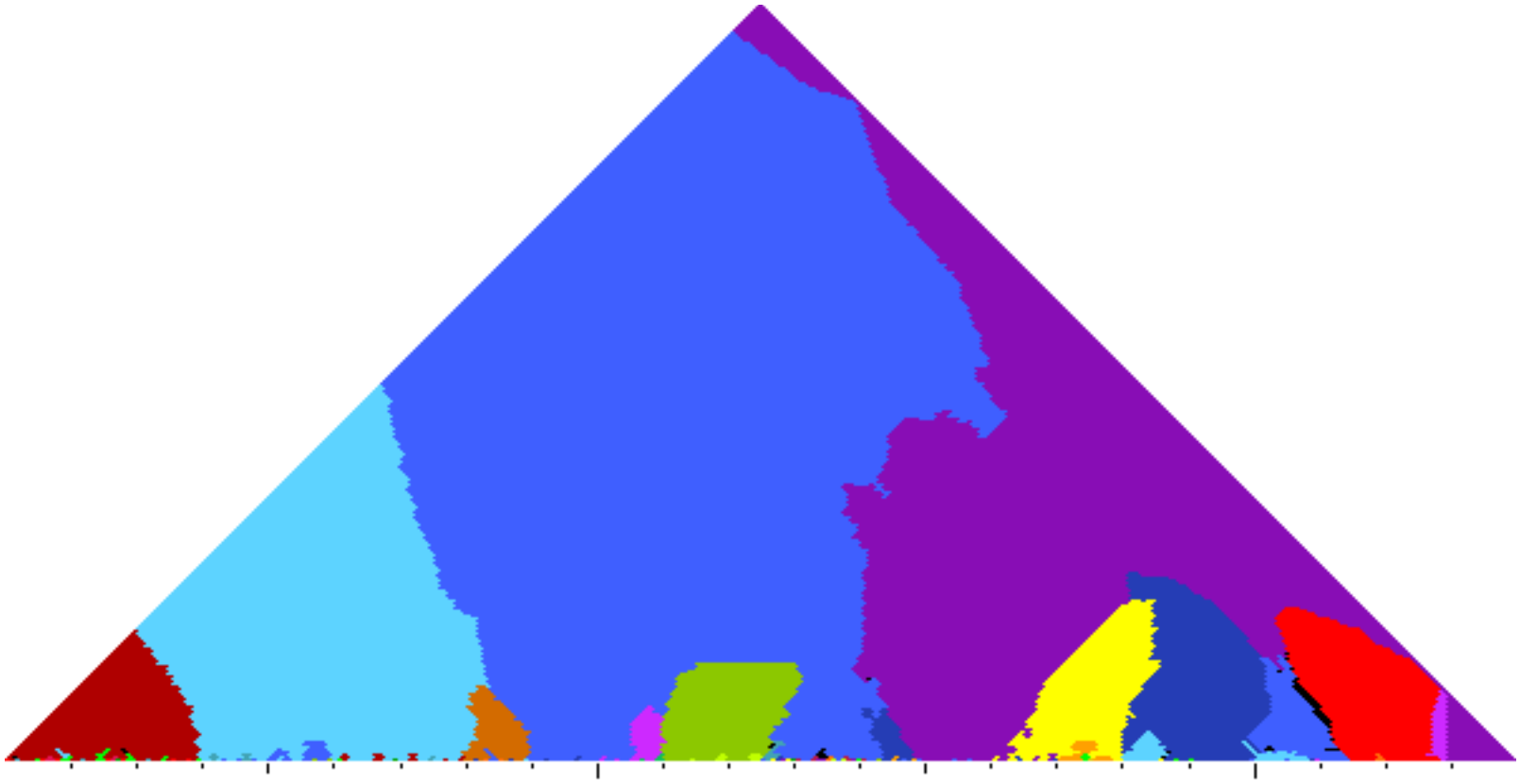
mkeyscape h://wtc/wtc1p11.krn –n --kp | convert - z.png

- post processing:
- regions anchored to bottom of plot
- tonal biasing
- key assignment is now F major (top)



mkeyscape h://wtc/wtc1p11.krn -n --kp --trim | convert - z.png

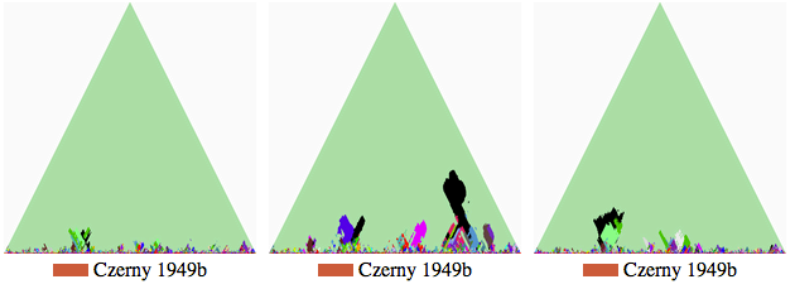
Chopin prelude in A minor 28/2



mkeyscape h://chopin/preludes/prelude28-02.krn -n --ss --trim | convert - z.png

Other Scape plotting applications

performance analysis



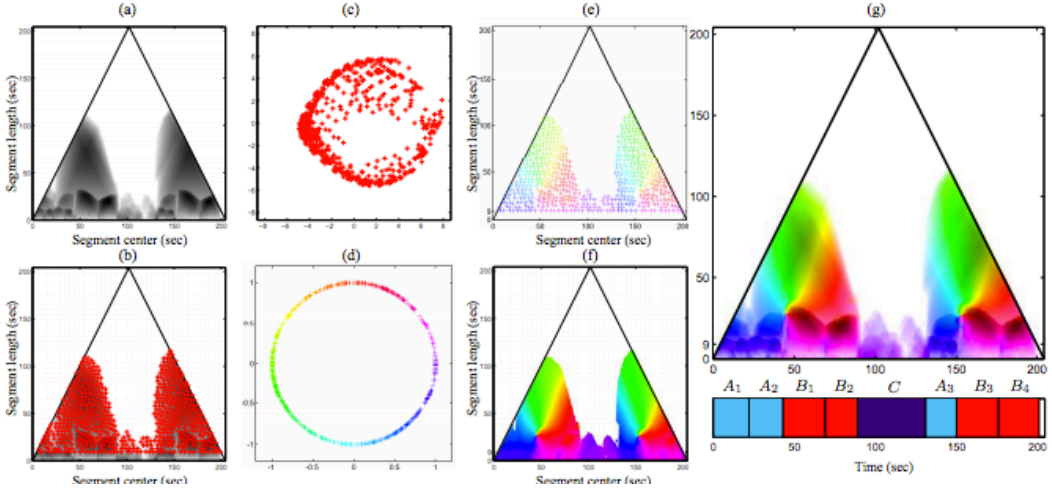
Czerny1949b	
96.7%	Czerny 1949
0.97	Czerny 1949
0.70	Kissin 1993
0.68	Magin 1975
0.63	Osinska 1989
0.63	Poblocka 1999
0.63	Average
0.62	Kilenyi 1937
0.62	Ezaki 2006
0.61	Fou 1978
0.57	Clidat 1994
0.57	Fliere 1977
0.57	Luisada 1990
0.57	Shebanova 2002

Czerny1949b	
89.3%	Czerny 1949
3.3%	Average
1.0%	Osinska 1989
0.96	Czerny 1949
0.65	Osinska 1989
0.62	Average
0.60	Szpilman 1948
0.58	Fliere 1977
0.55	Kilenyi 1937

Czerny1949b	
94.5%	Czerny 1949
1.3%	Average
0.97	Czerny 1949
0.73	Kissin 1993
0.73	Kiepura 1999
0.70	Fou 1978
0.68	Magin 1975
0.57	Cortot 1951
0.57	Horowitz 1971

http://ismir2008.ismir.net/papers/ISMIR2008_240.pdf

form analysis



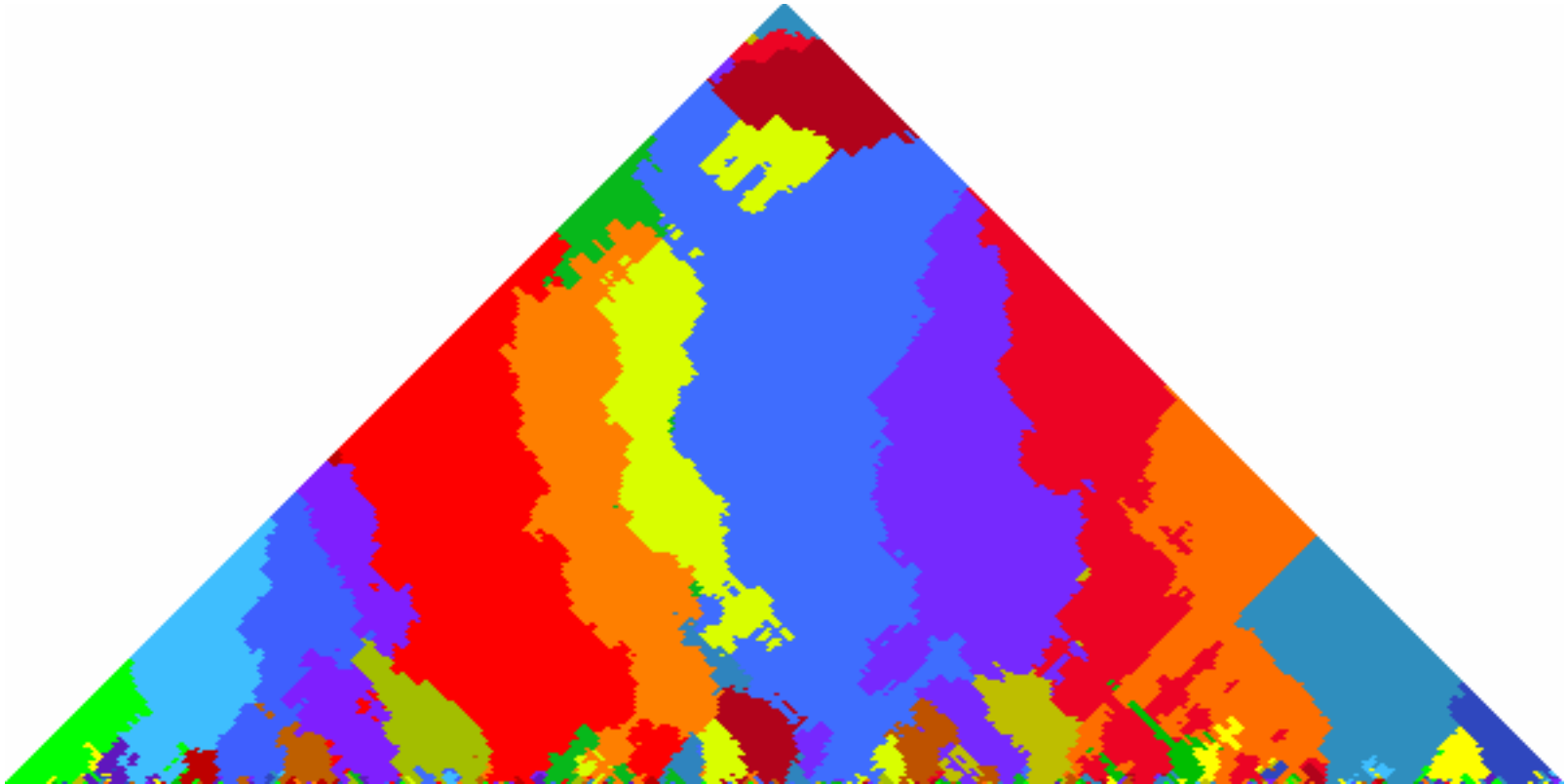
http://ismir2012.ismir.net/event/papers/097-ismir-2012.pdf

More Info

http://purl.stanford.edu/br237mp4161

Lots of keyscapes (raw plots; not post-processed):

https://ccrma.stanford.edu/~craig/keyscape/class



Other Harmony Tools

1. Intervals

hint: harmonic intervals

2. Chords

tntype: sonority types

3. Key

key/**keycor**: musical key by correlation

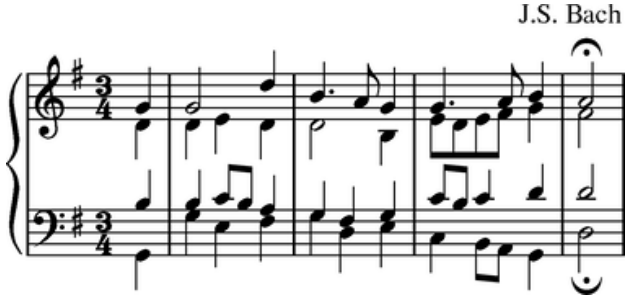
hint

```
humcat h://371chorales/chor001.krn > chor001.krn
hint chor001.krn > hint.txt
assemble chor001.krn hint.txt > chor001.hint
```

(Save a local copy of chorale 1)

(Save output from hint)

(Merge hint data with original score)




```
!!!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
**kern       **kern       **kern       **kern       **hint
*Ibass       *Itenor      *Ialto       *Isoprn      *
*clefF4      *clefGv2     *clefG2      *clefG2      *
*k[f#]       *k[f#]       *k[f#]       *k[f#]       *k[f#]
*G:          *G:          *G:          *G:          *G:
*M3/4        *M3/4        *M3/4        *M3/4        *M3/4
4GG          4B           4d           4g           M10 m3 P4
=1           =1           =1           =1           =1
4G           4B           4d           2g           M3 m3 P4
4E           8cL          4e           .            m6 M3
.            8BJ          .            .            -
4F#          4A           4d           4dd          m3 P4 P8
=2           =2           =2           =2           =2
4G           4G           2d           4.b          P1 P5 M6
4D           4F#          .            .            M3
.            .            .            8a           -
4E           4G           4B           4g           m3 M3 m6
```

```
satb2gs file.krn | autostem | hum2muse \
  | muse2ps =z21v120,120c120T^^ \
  | pstopnm -dpi=300 | convert - -trim \
    -resize '33%' file.png
```

Humdrum program documentation

hint –h gives one-page summary of *hint* command (same for all Humdrum Toolkit programs)
tntype --options list options for serialize command (same for all Humdrum Extras programs)

Humdrum Toolkit man pages:
http://www.humdrum.org/Humdrum/commands/hint.html

Humdrum Extras man pages:
http://extras.humdrum.org/man/tntype

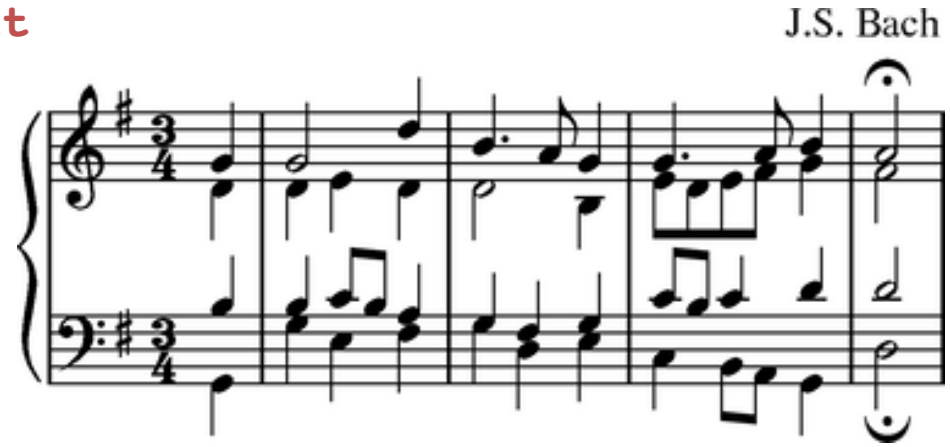
Chapter 15 in the Humdrum Users' Guide (Harmonic Intervals):
http://www.humdrum.org/Humdrum/guide15.html

List of various Humdrum resources:
http://humdrum.ccarh.org

hint -a

- -a option shows intervals for all note permutations, not just “stacked intervals”

hint –a chor001.krn > hinta.txt



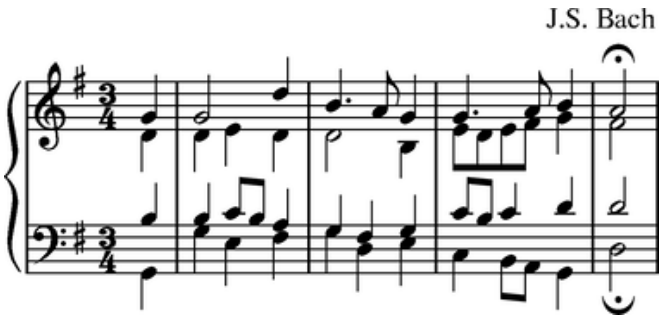

```
!!!COM: Bach, Johann Sebastian
!!!CDT: 1685/02/21/-1750/07/28/
!!!OTL@@DE:      Aus meines Herzens Grunde
!!!OTL@EN:       From the Depths of My Heart
!!!SCT: BWV 269
**kern   **kern   **kern   **kern   **hint
*Ibass   *Itenor  *Ialto   *Isoprn  *
*clefF4  *clefG2  *clefG2  *clefG2  *
*k[f#]   *k[f#]   *k[f#]   *k[f#]   *k[f#]
*G:      *G:      *G:      *G:      *G:
*M3/4    *M3/4    *M3/4    *M3/4    *M3/4
4GG      4B       4d       4g       M10 P12 P15 m3 m6 P4
=1       =1       =1       =1       =1
4G       4B       4d       2g       M3 P5 P8 m3 m6 P4
4E       8cL      4e       .        m6 P8 M3
.        8BJ      .        .        -
4F#      4A       4d       4dd      m3 m6 m13 P4 P11 P8
=2       =2       =2       =2       =2
4G       4G       2d       4.b      P1 P5 M10 P5 M10 M6
4D       4F#      .        .        M3
.        .        .        8a       -
4E       4G       4B       4g       m3 P5 m10 M3 P8 m6
```

(GG,B), (GG,d), (GG, g), (B,d), (B, g), (d, g)

hint -c

- Collapse the interval to a single octave. Such as: P12 → P8+P4 → P4

hint –ac chor001.krn > hinta.txt
hint –a -c chor001.krn > hintac.txt



**kern	**kern	**kern	**kern	**hint
*Ibass	*Itenor	*Ialto	*Isoprn	*
*clefF4	*clefGv2	*clefG2	*clefG2	*
*k[f#]	*k[f#]	*k[f#]	*k[f#]	*k[f#]
*G:	*G:	*G:	*G:	*G:
*M3/4	*M3/4	*M3/4	*M3/4	*M3/4
4GG	4B	4d	4g	M3 P5 P1 m3 m6 P4
=1	=1	=1	=1	=1
4G	4B	4d	2g	M3 P5 P1 m3 m6 P4
4E	8cL	4e	.	m6 P1 M3
.	8BJ	.	.	-
4F#	4A	4d	4dd	m3 m6 m6 P4 P4 P1
=2	=2	=2	=2	=2
4G	4G	2d	4.b	P1 P5 M3 P5 M3 M6
4D	4F#	.	.	M3
.	.	.	8a	-
4E	4G	4B	4g	m3 P5 m3 M3 P1 m6
=3	=3	=3	=3	=3
4C	8cL	8eL	4.g	P1 M3 P5 M3 P5 m3
.	8BJ	8d	.	m3
8BBL	4c	8e	.	m2 P4 M3
8AAJ	.	8f#J	8a	M6 P1 m3
4GG	4d	4g	4b	P5 P1 M3 P4 M6 M3

M10 → M3
P12 → P5
P15 → P1
m3 → m3
m6 → m6
P4 → P4

Most common harmonic interval

hint –ac chor001.krn | serialize –c | ridx –H | sort | uniq –c | sort -nr

1. hint –ac chor001.krn: do harmonic interval analysis (Humdrum Toolkit)
2. serialize –c: force intervals one to a line (Humdrum Extras)
3. ridx –H: remove Humdrum file structure from data (Humdrum Extras)
4. sort: sort lines alphabetically (Unix)
5. uniq –c: output lines without repetitions, counting occurrences (Unix)
6. sort –nr: sort numerically, in reverse order (largest count first) (Unix)

```
**hint
*k[f#]
*G:
*M3/4
M3 P5 P1 m3 m6 P4
=1
M3 P5 P1 m3 m6 P4
m6 P1 M3
-
m3 m6 m6 P4 P4 P1
=2
P1 P5 M3 P5 M3 M6
M3
-
m3 P5 m3 M3 P1 m6
```

```
**hint
*k[f#]
*G:
*M3/4
M3
P5
P1
m3
m6
P4
=1
M3
P5
P1
m3
m6
P4
```

```
M3
P5
P1
m3
m6
P4
M3
P5
P1
m3
m6
P4
m6
P1
M3
m3
m6
m6
```

```
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
```

```
21 –
2  A4
4  M2
55 M3
23 M6
43 P1
30 P4
44 P5
 3 d5
 2 m2
42 m3
24 m6
 6 m7
```

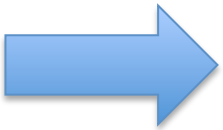
```
55 M3
44 P5
43 P1
42 m3
30 P4
24 m6
23 M6
21 –
 6 m7
 4 M2
 3 d5
 2 m2
 2 A4
```

ditto

- *hint* ignores null tokens, resulting in harmonic intervals between note attacks only
- To also include intervals to sustained intervals from previous attacks, use *ditto*
- This fills in the null token with the continuing data token it represents

ditto chor001.krn

```
**kern  **kern  **kern  **kern
*Ibass  *Itenor *Ialto  *Isoprn
*clefF4 *clefG2 *clefG2 *clefG2
*k[f#]  *k[f#]  *k[f#]  *k[f#]
*G:     *G:     *G:     *G:
*M3/4   *M3/4   *M3/4   *M3/4
4GG     4B      4d      4g
=1      =1      =1      =1
4G      4B      4d      2g
4E      8cL     4e      .
.       8BJ     .       .
4F#     4A      4d      4dd
=2      =2      =2      =2
4G      4G      2d      4.b
4D      4F#     .       .
.       .       .       8a
4E      4G      4B      4g
```



```
**kern  **kern  **kern  **kern
*Ibass  *Itenor *Ialto  *Isoprn
*clefF4 *clefG2 *clefG2 *clefG2
*k[f#]  *k[f#]  *k[f#]  *k[f#]
*G:     *G:     *G:     *G:
*M3/4   *M3/4   *M3/4   *M3/4
4GG     4B      4d      4g
=1      =1      =1      =1
4G      4B      4d      2g
4E      8cL     4e      2g
4E      8BJ     4e      2g
4F#     4A      4d      4dd
=2      =2      =2      =2
4G      4G      2d      4.b
4D      4F#     2d      4.b
4D      4F#     2d      8a
4E      4G      4B      4g
```

Most common harmonic interval including sustained notes

hint –ac chor001.krn | serialize –c | ridx –H | sort | uniq –c | sort -nr

ditto chor001.krn | hint –ac | serialize –c | ridx –H | sort | uniq –c | sort -nr

attacks

```
55 M3
44 P5
43 P1
42 m3
30 P4
24 m6
23 M6
21 -
 6 m7
 4 M2
 3 d5
 2 m2
 2 A4
```

+sustains

```
76 P5
75 M3
74 m3
59 P1
56 P4
43 M6
41 m6
18 m7
16 M2
 9 A4
 7 d5
 4 m2
 2 M7
```

All Bach chorales

download and save all 370 chorales locally:

humcat –s h://371chorales > 371chorales.krns (one file containing all chorales)
humcat –s h://371chorales | humsplit (separate each chorale into a separate file)

hint –ac 371chorales.krns | serialize –c | ridx –H | sort | uniq –c | sort -nr

ditto 371chorales.krns | hint –ac | serialize –c | ridx –H | sort | uniq –c | sort -nr

Attacked minor thirds more common than attacked 5ths.

```
18053 m3
17545 P5
16147 M3
15263 P1
10812 P4
10035 M6
 9935 m6
 2537 M2
 1542 m7
 1534 A4
 1103 d5
  397 m2
  248 M7
  175 d7
   64 A2
   62 d4
   43 A5
    2 A6
    1 d1
```

```
29352 P5
28149 m3
23975 M3
22049 P1
20642 P4
17349 M6
15016 m6
 7015 M2
 6498 m7
 3721 A4
 3131 d5
 1555 m2
 1183 M7
  299 d7
  182 r
  176 d4
  146 A2
  111 A5
    3 d1
    3 A6
    1 d3
```

Sustained 5ths more common sustained minor thirds.

Beethoven string quartets

download and save all quartets locally:

humcat –s h://beethoven/quartets > beethoven-quartets.krns

or humcat –s h://beethoven/quartests | humsplit

hint –ac beethoven-quartets.krns | serialize –c | ridx –H | sort | uniq –c | sort -nr

ditto beethoven-quartets.krns | hint –ac | serialize –c | ridx –H | sort | uniq –c | sort -nr

```
60156 P1
 40163 m3          269 A6
 31479 M3          104 A1
 26374 P5           80 d6
 24285 M6           67 d3
 21775 P4           55 d1
 18348 m6           39 A3
 10697 m7           27 AA4
  9520 M2           23 d2
  7491 A4           18 A7
  6559 d5           10 dd5
  2237 M7            7 dd1
  1659 A2            7 AA2
  1419 m2            5 dd7
  1395 d7            2 dd4
   529 A5            2 AA5
   431 d4
```

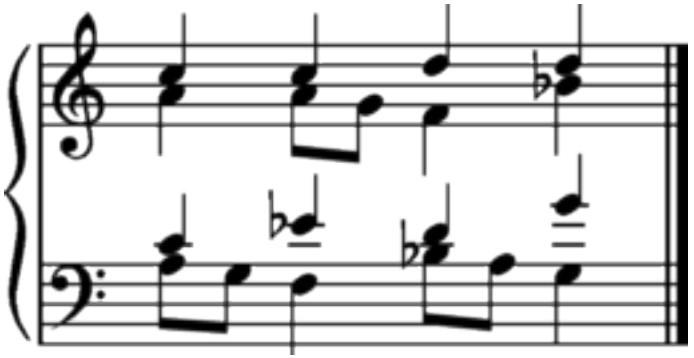
```
102597 P1
 69409 m3          403 A1
 57966 M3          223 d1
 55746 P5          147 d6
 46270 M6          121 A3
 45913 P4          113 d3
 35295 m6           72 d2
 23213 m7           40 A7
 22159 M2           37 AA4
 14037 A4           34 dd5
 12924 d5           15 dd1
  6139 M7           13 dd7
  4529 m2           13 AA2
  3020 A2            3 dd4
  2764 d7            3 AA5
  1453 A5            1 AA6
  1155 d4            1 AA3
   533 A6            1 AA1
```

tntype

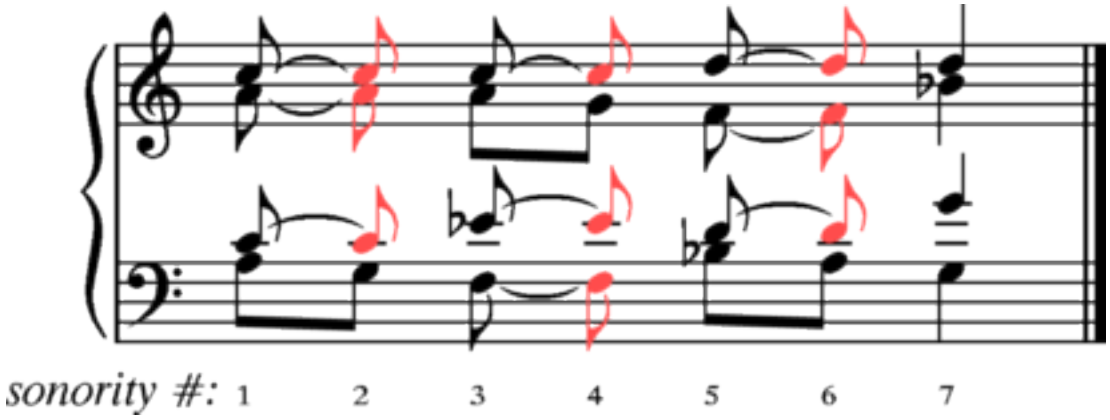
Tool for generalized description of sonority types (pitch-class sets sounding together)

documentation: http://extras.humdrum.org/man/tntype

similar to Humdrum Toolkit command pcset http://www.humdrum.org/Humdrum/commands/pcset.html

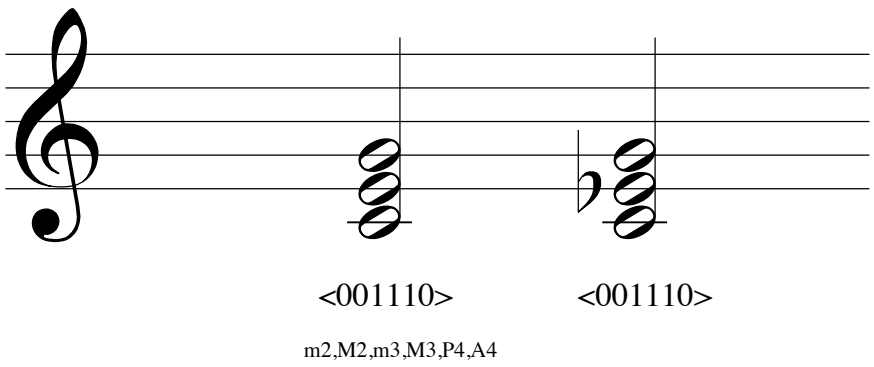


seven successive “sonorities” present in the above music:



Interval vectors

- Similar data generated by *hint*, but more compact

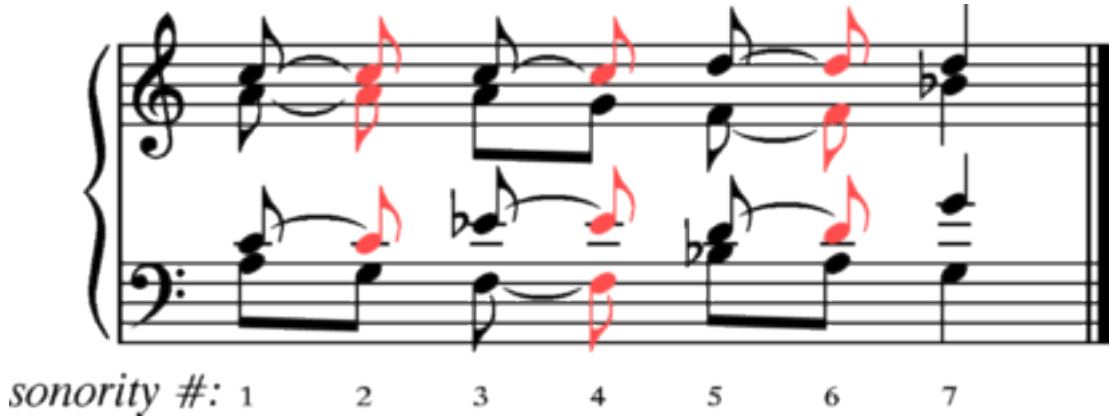


- Major and minor triads have the same interval content: m3, M3, P5
- <001110> is can be represented by the Forte number enumeration 3-11.
  - 3 = three pitch classes in set
  - 11 = 11th most compact organization of 3 pitch classes

    3-1 = c,c#,d 3-2 = c,c#,d# 3-3 = c,c#,e 3-4 = c,c#,f 3-5 = c,c#,f# etc.

tntype -d

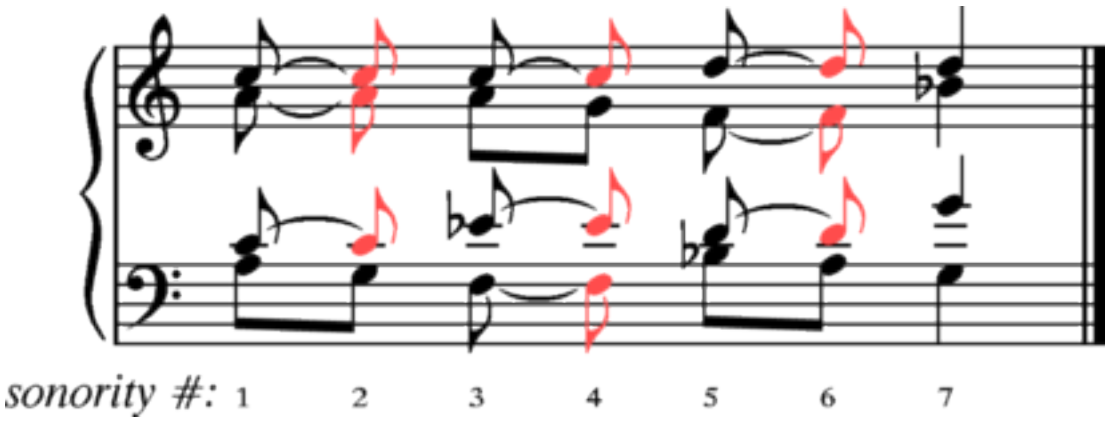
Generate a **dpc (diatonic pitch-class) spine listing unique pitch classes in sonorities



tntype –ad file.krn

```
**kern   **kern   **kern   **kern   **num   **dpc
=1-      =1-      =1-      =1-      =1-     =1-
8AL      4c       4a       4cc      1       A c
8GJ      .        .        .        2       (a) (c) G
4F       4e-      8aL      4cc      3       F a c e-
.        .        8gnXJ    .        4       (c) (e-) (F) g
8B-L     4d       4f       4dd      5       B- d f
8AJ      .        .        .        6       (d) (f) A
4G       4g       4b-      4dd      7       G b- d
==       ==       ==       ==       ==      ==
*-       *-       *-       *-       *-      *-
```

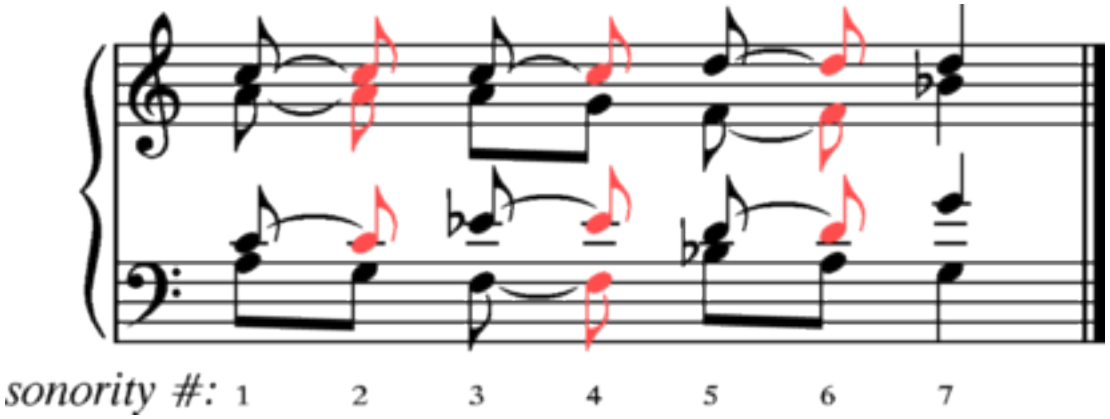
Normal form



tntype –na file.krn

**kern	**kern	**kern	**kern	**num	**nf
=1-	=1-	=1-	=1-	=1-	=1-
8AL	4c	4a	4cc	1	[90]
8GJ	.	.	.	2	[790]
4F	4e-	8aL	4cc	3	[9035]
.	.	8gnXJ	.	4	[0357]
8B-L	4d	4f	4dd	5	[A25]
8AJ	.	.	.	6	[259]
4G	4g	4b-	4dd	7	[7A2]
==	==	==	==	==	==
*-	*-	*-	*-	*-	*-

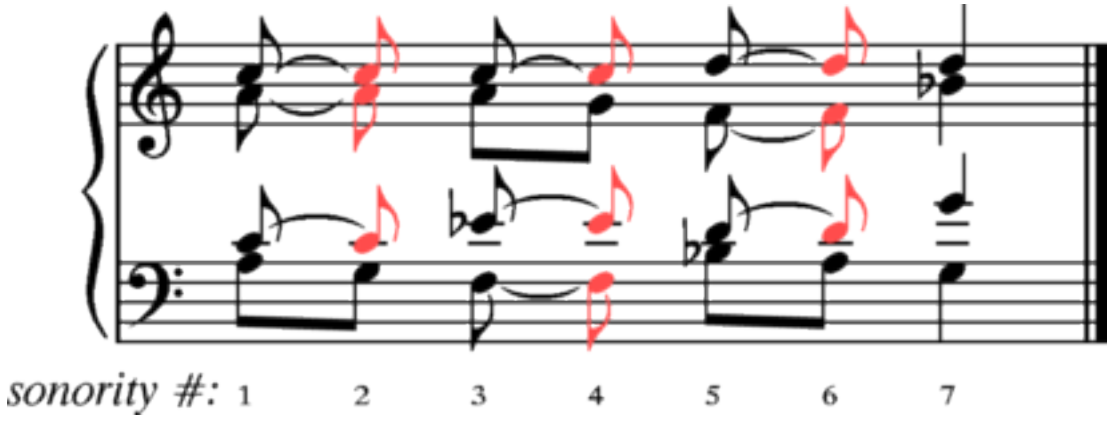
Transposed normal form



tntype –af file.krn

**kern	**kern	**kern	**kern	**num	**tnf
=1-	=1-	=1-	=1-	=1-	=1-
8AL	4c	4a	4cc	1	{03}
8GJ	.	.	.	2	{025}
4F	4e-	8aL	4cc	3	{0368}
.	.	8gnXJ	.	4	{0357}
8B-L	4d	4f	4dd	5	{047}
8AJ	.	.	.	6	{037}
4G	4g	4b-	4dd	7	{037}
==	==	==	==	==	==
*-	*-	*-	*-	*-	*-

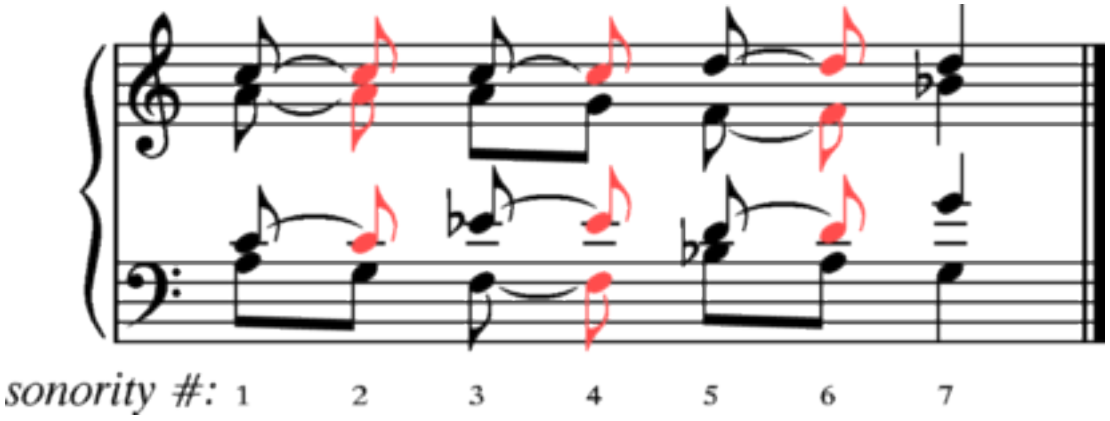
Preserving transposition



tntype –an input | tntype –aft

**kern	**kern	**kern	**kern	**num	**nf	**tnf
=1-	=1-	=1-	=1-	=1-	=1-	=1-
8AL	4c	4a	4cc	1	[90]	{03}T9
8GJ	.	.	.	2	[790]	{025}T7
4F	4e-	8aL	4cc	3	[9035]	{0368}T9
.	.	8gnXJ	.	4	[0357]	{0357}T0
8B-L	4d	4f	4dd	5	[A25]	{047}T10
8AJ	.	.	.	6	[259]	{037}T2
4G	4g	4b-	4dd	7	[7A2]	{037}T7
==	==	==	==	==	==	==
*-	*-	*-	*-	*-	*-	*-

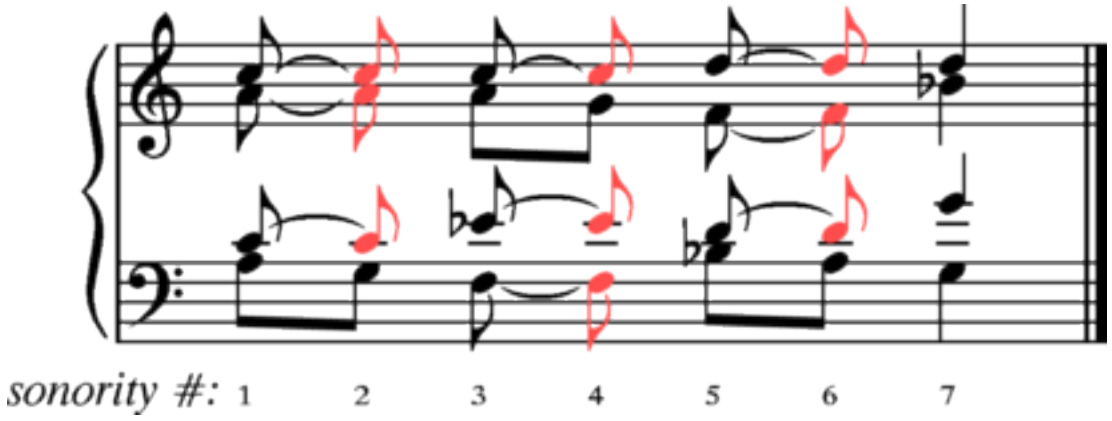
Forte pc-set enumerations



Tntype –an | tntype –aft | tntype -aF

**kern	**kern	**kern	**kern	**num	**nf	**tnf	**forte
=1-	=1-	=1-	=1-	=1-	=1-	=1-	=1-
8AL	4c	4a	4cc	1	[90]	{03}T9	2-3
8GJ	.	.	.	2	[790]	{025}T7	3-7
4F	4e-	8aL	4cc	3	[9035]	{0368}T9	4-27
.	.	8gnXJ	.	4	[0357]	{0357}T0	4-22
8B-L	4d	4f	4dd	5	[A25]	{047}T10	3-11
8AJ	.	.	.	6	[259]	{037}T2	3-11
4G	4g	4b-	4dd	7	[7A2]	{037}T7	3-11
==	==	==	==	==	==	==	==
*-	*-	*-	*-	*-	*-	*-	*-

Forte numbers, without inversions



tntype –an | tntype –aft | tntype –aF --Tn

**kern	**kern	**kern	**kern	**num	**nf	**tnf	**Tn
=1-	=1-	=1-	=1-	=1-	=1-	=1-	=1-
8AL	4c	4a	4cc	1	[90]	{03}T9	2-3
8GJ	.	.	.	2	[790]	{025}T7	3-7
4F	4e-	8aL	4cc	3	[9035]	{0368}T9	4-27
.	.	8gnXJ	.	4	[0357]	{0357}T0	4-22
8B-L	4d	4f	4dd	5	[A25]	{047}T10	3-11B
8AJ	.	.	.	6	[259]	{037}T2	3-11A
4G	4g	4b-	4dd	7	[7A2]	{037}T7	3-11A
==	==	==	==	==	==	==	==
*-	*-	*-	*-	*-	*-	*-	*-

Chord interpretations

humcat h://371chorales/chor001.krn | tntype -a | tntype -tfa | tntype -Da

**kern	**kern	**kern	**kern	**tnt	**tnf	**description
*Ibass	*Itenor	*Ialto	*Isoprn	*	*	*
*k[f#]	*k[f#]	*k[f#]	*k[f#]	*k[f#]	*k[f#]	*k[f#]
*M3/4	*M3/4	*M3/4	*M3/4	*M3/4	*M3/4	*M3/4
4GG	4B	4d	4g	3-11B	{047}T07	Major Chord
=1	=1	=1	=1	=1	=1	=1
4G	4B	4d	2g	3-11B	{047}T07	Major Chord
4E	8cL	4e	.	3-11B	{047}T00	Major Chord
.	8BJ	.	.	3-11A	{037}T04	Minor Chord
4F#	4A	4d	4dd	3-11B	{047}T02	Major Chord
=2	=2	=2	=2	=2	=2	=2
4G	4G	2d	4.b	3-11B	{047}T07	Major Chord
4D	4F#	.	.	3-11A	{037}T11	Minor Chord
.	.	.	8a	3-11B	{047}T02	Major Chord
4E	4G	4B	4g	3-11A	{037}T04	Minor Chord
=3	=3	=3	=3	=3	=3	=3
4C	8cL	8eL	4.g	3-11B	{047}T00	Major Chord
.	8BJ	8d	.	4-14B	{0457}T07	Perfect-fourth Major Tetrachord
8BBL	4c	8e	.	4-20	{0158}T11	Major-seventh Chord
8AAJ	.	8f#J	8a	3-10	{036}T06	Diminished Chord
4GG	4d	4g	4b	3-11B	{047}T07	Major Chord
=4	=4	=4	=4	=4	=4	=4

Bach chorale sonority types

- Are there more major or minor sonorities in Bach chorales?

By musical description:

```
humcat -s h://371chorales | tntype -D | ridx –H | sort | uniq -c | sort -nr
```

By Forte number:

```
humcat -s h://371chorales | tntype | ridx –H | sort | uniq -c | sort -nr
```

- Do Bach chorales in minor keys have more major or minor sonorities?

```
humcat –s h://371chorales | humsplit
humcat -s `egrep –l "^\*[a-g][#-]?:" chor*.krn` | tntype -D | ridx –H | sort | uniq -c | sort -nr
```

- What is the most common 7th chord sonority?

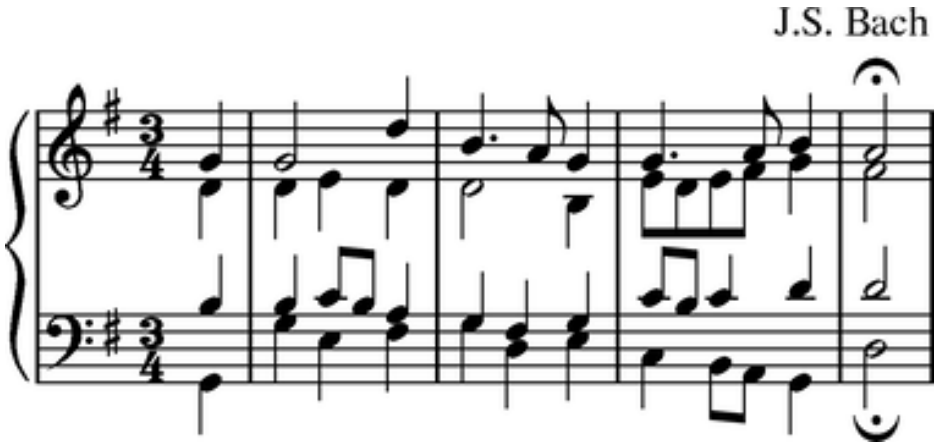
sonority

- similar to tntype program but has more triad-centered descriptions of sonorities

http://extras.humdrum.org/man/sonority

sonority -a h://371chorales/chor001.krn

**kern	**kern	**kern	**kern	**qual
*ICvox	*ICvox	*ICvox	*ICvox	*ICvox
*Ibass	*Itenor	*Ialto	*Isoprn	*
*k[f#]	*k[f#]	*k[f#]	*k[f#]	*k[f#]
*G:	*G:	*G:	*G:	*G:
*M3/4	*M3/4	*M3/4	*M3/4	*M3/4
*MM100	*MM100	*MM100	*MM100	*MM100
4GG	4B	4d	4g	maj:0:G
=1	=1	=1	=1	=1
4G	4B	4d	2g	maj:0:G
4E	8cL	4e	.	maj:1:C
.	8BJ	.	.	min:0:E
4F#	4A	4d	4dd	maj:1:D
=2	=2	=2	=2	=2
4G	4G	2d	4.b	maj:0:G
4D	4F#	.	.	min:1:B
.	.	.	8a	maj:0:D
4E	4G	4B	4g	min:0:E
=3	=3	=3	=3	=3
4C	8cL	8eL	4.g	maj:0:C
.	8BJ	8d	.	X
8BBL	4c	8e	.	majmaj:3:C
8AAJ	.	8f#J	8a	dim:1:F#
4GG	4d	4g	4b	maj:0:G
=4	=4	=4	=4	=4
2D;	2d;	2f#;	2a;	maj:0:D



Starting/Ending sonority

- Do Bach chorales start and end on the same sonority?
- How does the starting/ending chord root relate to the key of the chorale?

humcat -s h://371chorales | humsplit

```
#!/usr/bin/perl

@filelist = @ARGV;

foreach $file (@filelist) {
   processFile($file)
}

sub processFile {
  my ($file) = @_;
  $first_sonority = `sonority $file | ridx -GLIMd | grep -v "::" | head -n 1`;
  $last_sonority  = `sonority $file | ridx -GLIMd | grep -v "::" | tail -n 1`;
  $key = `egrep -i '^\\*[A-G][#-]?:' $file | head -n 1 | sed 's/\t.*//'`;
  chomp $first_sonority;
  chomp $last_sonority;
  chomp $key;
  print "$file\t$key\t$first_sonority\t$last_sonority\n";
}
```

Starting/Ending sonority (2)

chor001.krn	*G:	maj:0:G	maj:0:G
chor002.krn	*A:	maj:0:A	maj:0:A
chor003.krn	*a:dor	maj:0:E	maj:0:E
chor004.krn	*E:	maj:0:E	maj:0:E
chor005.krn	*G:	maj:0:G	maj:0:G
chor006.krn	*F:	maj:0:F	maj:0:F
chor007.krn	*A:	maj:0:A	maj:0:A
chor008.krn	*f:dor	min:0:F	maj:0:F
chor009.krn	*G:	maj:0:G	maj:0:G
chor010.krn	*a:	domsev:3:E	maj:0:E
chor011.krn	*C:	maj:0:C	maj:0:C
chor012.krn	*a:	min:0:A	maj:0:A
chor013.krn	*a:	min:0:A	maj:0:A
chor014.krn	*G:	maj:0:G	maj:0:G
chor015.krn	*d:dor	min:0:D	maj:0:D
chor016.krn	*b:	maj:0:F#	maj:0:F#
chor017.krn	*e:	min:0:E	maj:0:E
chor018.krn	*G:	maj:0:G	maj:0:G
chor019.krn	*g:dor	min:0:G	maj:0:G
chor020.krn	*D:	maj:0:D	maj:0:D
chor021.krn	*a:	domsev:3:E	maj:0:E
chor022.krn	*E-:	maj:0:E-	maj:0:E-
chor023.krn	*a:	min:0:A	maj:0:A
chor024.krn	*D:	maj:0:D	maj:0:D
chor025.krn	*f:dor	min:0:F	maj:0:F

Check for unusual cases

```
#!/usr/bin/perl

@filelist = @ARGV;

foreach $file (@filelist) {
    processFile($file)
}

sub processFile {
   my ($file) = @_;
   $first_sonority = `sonority $file | ridx -GLIMd | grep -v "::" | head -n 1`;
   $last_sonority  = `sonority $file | ridx -GLIMd | grep -v "::" | tail -n 1`;
   $key = `egrep -i '^\\*[A-G][#-]?:' $file | head -n 1 | sed 's/\t.*//'`;
   chomp $first_sonority;
   chomp $last_sonority;
   chomp $key;
   $first_sonority =~ /:([^:]*)$/;
   $first_root = $1;
   $last_sonority =~ /:([^:]*)$/;
   $last_root = $1;
   $key =~ /^\*([A-G][#-]?):/;
   $key_root = uc($1);
   if (($first_root ne $last_root) or ($first_root ne $key_root)) {
      print "$file\t$key\t$first_sonority\t$last_sonority\n";
   }
}
```

Inconsistent start/end/key

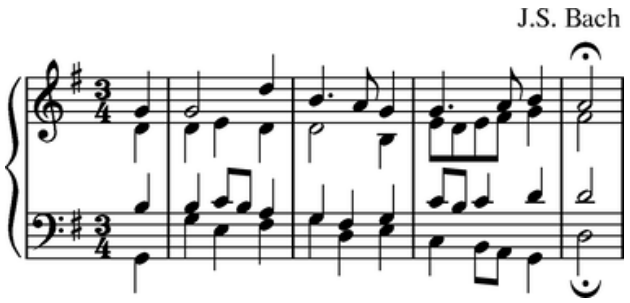
	chor056.krn	*b:	min:0:E	maj:0:F#
	chor057.krn	*a:	maj:0:E	min:0:A
	chor066.krn	*a:	min:0:D	maj:0:A
	chor071.krn	*e:	maj:1:B	maj:0:E
	chor074.krn	*F:	min:0:D	maj:0:F
	chor077.krn	*A:	min:0:F#	maj:0:A
	chor079.krn	*a:	min:0:A	maj:0:E
	chor083.krn	*A:	min:0:F#	maj:0:A
	chor089.krn	*b:	min:0:B	maj:0:F#
	chor119.krn	*c:dor	maj:0:B-	maj:0:C
	chor121.krn	*A:	min:0:F#	maj:0:A
	chor154.krn	*G:mix	min:0:D	maj:0:G
	chor162.krn	*d:dor	maj:0:A	maj:0:E
	chor181.krn	*e:	min:0:E	maj:0:B
	chor205.krn	***C:**	**min:0:E**	**maj:0:E → *e:phr**
	chor208.krn	*e:	min:0:E	maj:0:B
	chor227.krn	*d:	min:0:G	maj:0:D
	chor248.krn	*G:	maj:1:D	maj:0:G
bwv 77/6	**chor253.krn**	***g:**	**maj:0:A**	**maj:0:D**
	chor255.krn	*D:	maj:1:A	maj:0:D
	chor275.krn	*A:	maj:1:C#	maj:0:A
	chor284.krn	*C:mix	maj:0:F	maj:0:C
	chor286.krn	*b:	min:0:B	maj:0:F#
	chor288.krn	*A:mix	min:0:F#	maj:0:A
	chor291.krn	*D:	maj:1:A	maj:0:D
	chor311.krn	*F:	maj:1:C	maj:0:F
	chor314.krn	*e:	maj:0:B	maj:0:F#
	chor315.krn	*G:	min:0:E	maj:0:G
	chor333.krn	*D:	maj:0:F#	maj:0:D
	chor337.krn	*F:	maj:0:A	maj:0:F
	chor341.krn	*A:	min:0:F#	maj:0:A
	chor357.krn	*G:mix	maj:0:C	maj:0:G
	chor359.krn	*b:	maj:0:D	incmaj:0:B
	chor364.krn	*b:	maj:0:F#	maj:0:B
	chor367.krn	*b:	maj:0:D	maj:0:F#

key

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

Estimated key: G major (r=0.9501) confidence: 52.3%

humcat h://371chorales/chor001.krn | key -a



Tonic[0]	major 0.441131	minor 0.0554652
Tonic[1]	major -0.711388	minor -0.415044
Tonic[2]	major 0.775722	minor 0.354884
Tonic[3]	major -0.301544	minor -0.42342
Tonic[4]	major -0.085096	minor 0.540753
Tonic[5]	major 0.00550523	minor -0.515126
Tonic[6]	major -0.407599	minor 0.0398076
Tonic[7]	major 0.9501	minor 0.434019
Tonic[8]	major -0.602254	minor -0.310748
Tonic[9]	major 0.158757	minor 0.224714
Tonic[10]	major -0.11878	minor -0.679797
Tonic[11]	major -0.104554	minor 0.694493

Estimated key: G major (r=0.9501) confidence: 52.3%

keycor

http://extras.humdrum.org/man/keycor

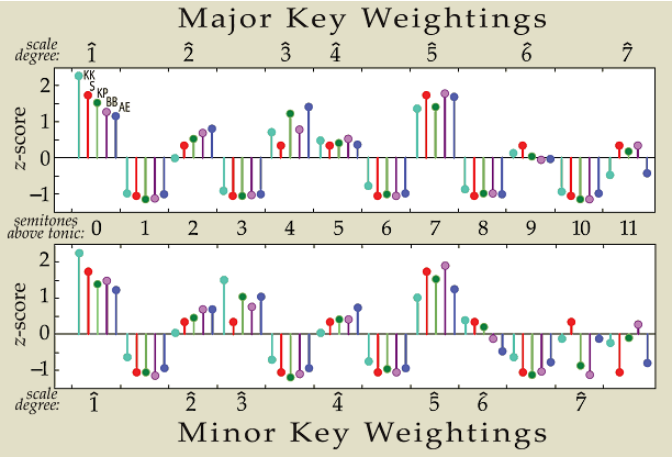
- Generalized version of the Humdrum Toolkit key program.

keycor h://371chorales/chor001.krn

The best key is: G Major

$$R(x,y) = \frac{\sum(x_n - \bar{x})(y_n - \bar{y})}{\sqrt{\sum(x_n - \bar{x})^2 \sum(y_n - \bar{y})^2}}$$

$$\text{key}_k = \arg\max_k R(x, y_k)$$



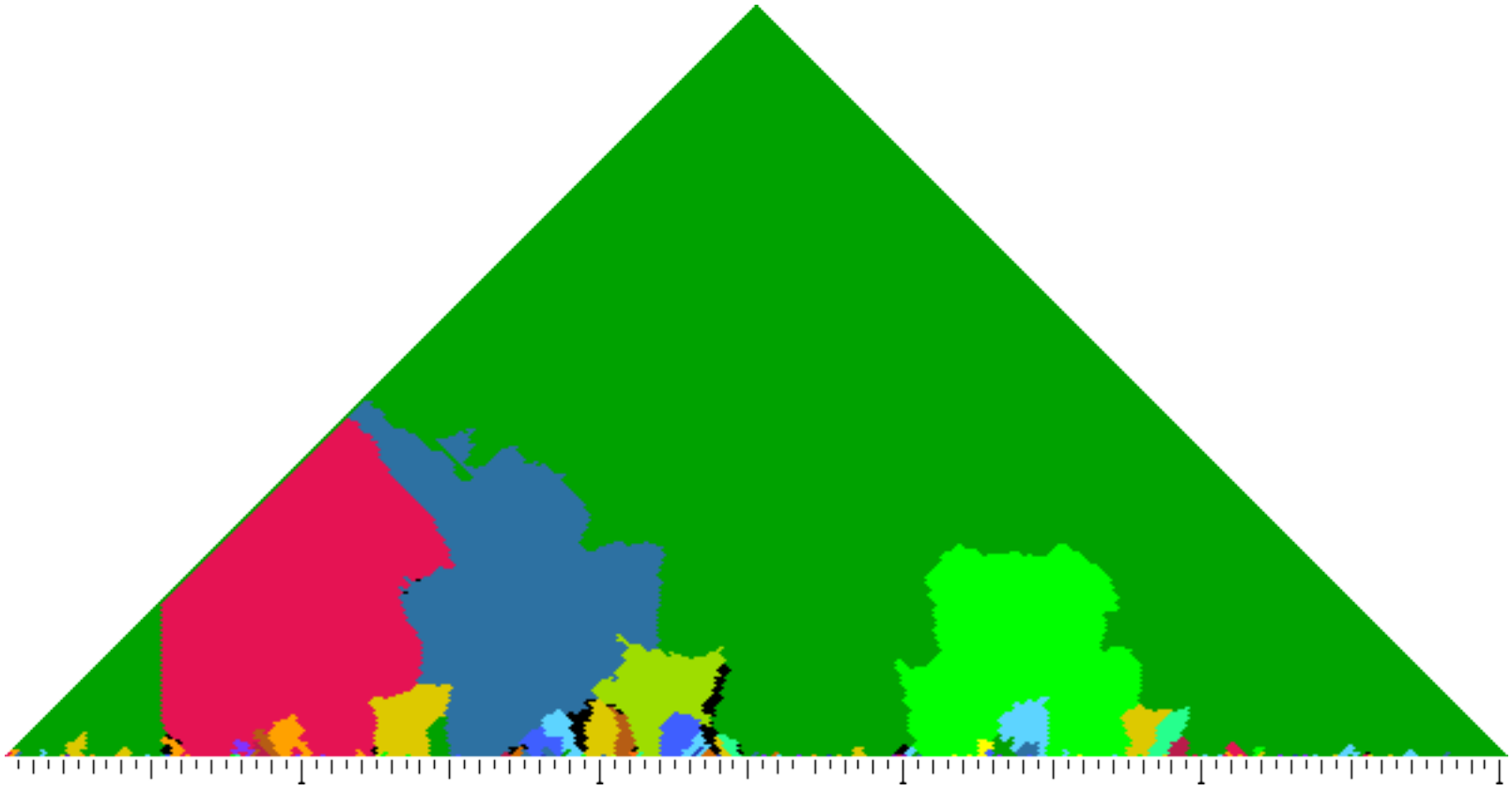
keycor -c h://371chorales/chor001.krn

**key	**rval	**conf	**start	**mid	**end
G	0.952	82	=0	=5	=11
G	0.956	87	=1	=6	=11
G	0.973	87	=1	=6	=12
G	0.974	97	=1	=6	=12
G	0.969	90	=2	=7	=12
G	0.975	94	=2	=7	=13
G	0.971	86	=2	=7	=13
G	0.969	92	=3	=8	=13
G	0.959	92	=3	=8	=14
G	0.959	86	=3	=8	=14
G	0.969	81	=4	=9	=14
G	0.958	71	=4	=9	=15
G	0.959	70	=4	=9	=15
G	0.962	71	=5	=10	=15
G	0.963	64	=5	=10	=16
G	0.960	67	=5	=10	=16
G	0.943	64	=6	=11	=16
G	0.960	69	=6	=11	=17
G	0.954	72	=6	=11	=17
G	0.948	64	=7	=12	=17
G	0.952	66	=7	=12	=18
G	0.966	76	=7	=12	=18
G	0.976	86	=8	=13	=18
G	0.975	83	=8	=13	=19
G	0.965	87	=8	=13	=19
G	0.970	93	=9	=14	=19
G	0.975	88	=9	=14	=20
G	0.972	82	=9	=14	=20
G	0.978	80	=10	=15	=20
G	0.980	78	=10	=15	=21
G	0.972	68	=10	=15	=21
*-	*-	*-	*-	*-	*-

mkeyscape

http://extras.humdrum.org/man/mkeyscape

- Structural analysis of key in a piece of music



* Beethoven's 5th symphony in C minor, first movement

C++ harmony analysis skeleton

git clone https://github.com/craigsapp/humextras
cd humextras
make library

Then place the following code into humextras/src-programs/midinotes.cpp
and then type "make midinotes", then type "bin/midinotes h://371chorales/chor001.krn"

```
**kern  **kern  **kern  **kern
*Ibass  *Itenor *Ialto  *Isoprn
*k[f#]  *k[f#]  *k[f#]  *k[f#]
*G:     *G:     *G:     *G:
*M3/4   *M3/4   *M3/4   *M3/4
4GG     4B      4d      4g
=1      =1      =1      =1
4G      4B      4d      2g
4E      8cL     4e      .
.       8BJ     .       .
4F#     4A      4d      4dd
=2      =2      =2      =2
4G      4G      2d      4.b
4D      4F#     .       .
.       .       .       8a
4E      4G      4B      4g
=3      =3      =3      =3
4C      8cL     8eL     4.g
.       8BJ     8d      .
8BBL    4c      8e      .
```

```
43 59 62 67
55 59 62 67
52 60 64 67
52 59 64 67
54 57 62 74
55 55 62 71
50 54 62 71
50 54 62 69
52 55 59 67
48 60 64 67
48 59 62 67
47 60 64 67
```

midinotes.cpp (1)

```
// This program takes multiple input files or standard input and outputs a
// list of MIDI pitches sounding at a every time (line) in the input score(s).

#include "humdrum.h"

void processSegment      (HumdrumFile& infile);
void processLine         (HumdrumFile& infile, int line);
void addFieldMidiNotes   (Array<int>& notelist, HumdrumFile& infile, int line,
                          int field);

int main(int argc, char** argv) {
   Options options;
   options.process(argc, argv);
   HumdrumFileSet infiles;
   int i;
   int incount = options.getArgCount();
   if (incount < 1) {
      infiles.read(cin);
   } else {
      for (i=0; i<incount; i++) {
         infiles.readAppend(options.getArg(i+1));
      }
   }

   for (i=0; i<infiles.getCount(); i++) {
      processSegment(infiles[i]);
   }

   return 0;
}
```

midinotes.cpp (2)

```
// processSegment -- handle data extraction from one Humdrum file segment
//    (such as a movement, or individual work from a collection).
void processSegment(HumdrumFile& infile) {
   for (int i=0; i<infile.getNumLines(); i++) {
      if (!infile[i].isData()) {
         continue;
      }
      processLine(infile, i);
   }
}

// processLine --  Print notes for one line of data.
void processLine(HumdrumFile& infile, int line) {
   Array<int> notelist;
   notelist.setSize(1000);
   notelist.setSize(0);
   for (int j=0; j<infile[line].getFieldCount(); j++) {
      if (infile[line].isExInterp(j, "**kern")) {
         addFieldMidiNotes(notelist, infile, line, j);
      }
   }
   for (int i=0; i<notelist.getSize(); i++) {
      cout << notelist[i];
      if (i < notelist.getSize()-1) {
         cout << ' ';
      }
   }
   if (notelist.getSize() > 0) {
      cout << '\n';
   }
}
```

midinotes.cpp (3)

```
// addFieldMidiNotes -- Print one or more notes in a Humdrum **kern token.
//     Don't do anything if there is a rest.

void addFieldMidiNotes(Array<int>& notelist, HumdrumFile& infile, int line,
      int field) {
   int k;
   int midinote;
   int tline  = line;
   int tfield = field;
   char buffer[1024] = {0};

   if (strcmp(infile[line][field], ".") == 0) {
      // resolve data represented by null token
      tline  = infile[line].getDotLine(field);
      tfield = infile[line].getDotSpine(field);
   }

   int tcount = infile[tline].getTokenCount(tfield);
   for (k=0; k<tcount; k++) {
      infile[tline].getToken(buffer, tfield, k);
      if (strchr(buffer, 'r') != NULL) {
         // ignore rests
         continue;
      }
      midinote = Convert::kernToMidiNoteNumber(buffer);
      notelist.append(midinote);
   }
}
```