21M.269 – Studies in Western Music History:

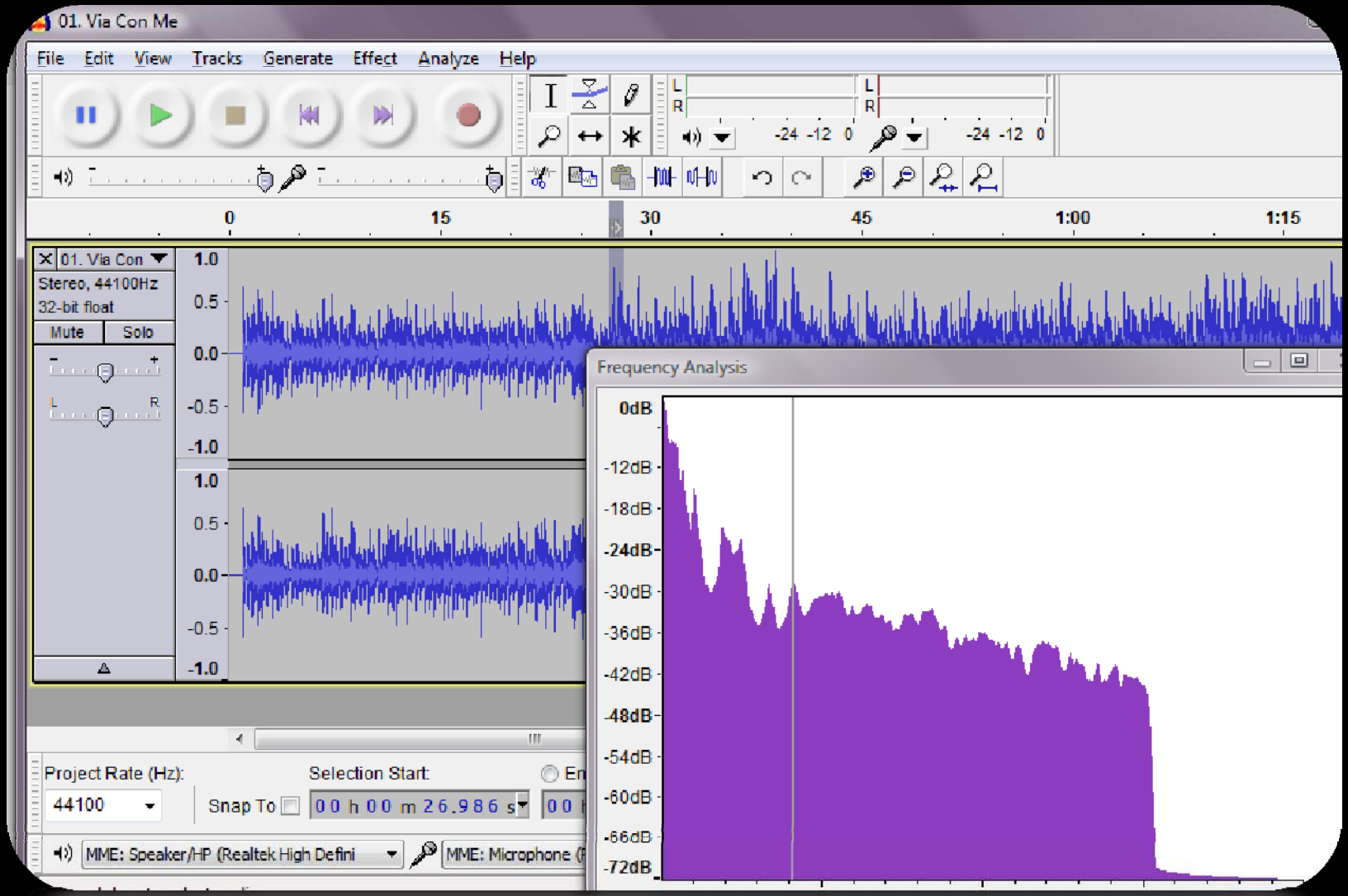
Quantitative and Computational Approaches to Music History

Michael Scott Cuthbert
Associate Professor of Music, M.I.T.

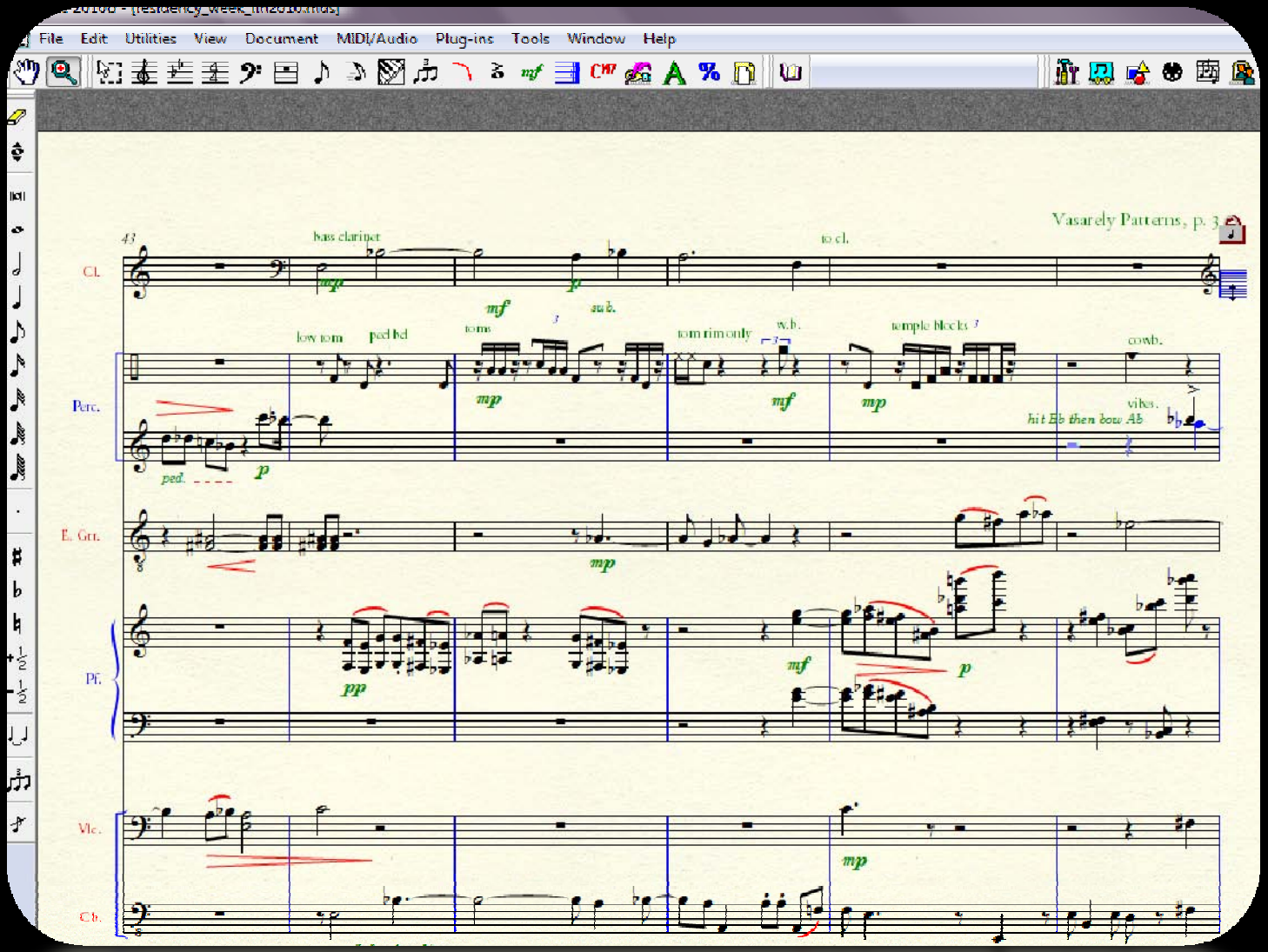
7 February 2012

[Screenshot of iTunes® removed due to copyright restrictions.]

Computers are changing how we acquire and share music



Using computers to analyze and manipulate audio



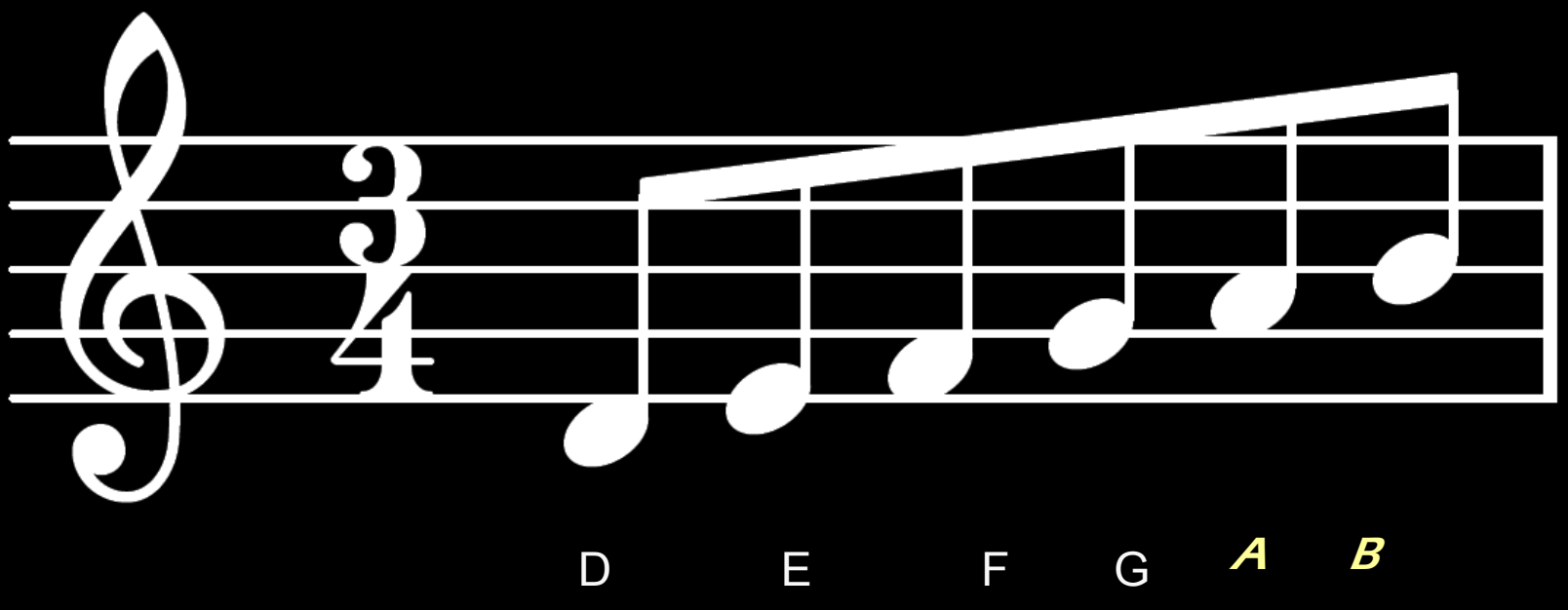
Computers are changing how composers notate their scores

Despite all the advances in using Computers in Music

practically nothing has changed in
how computers are used to analyze scores.

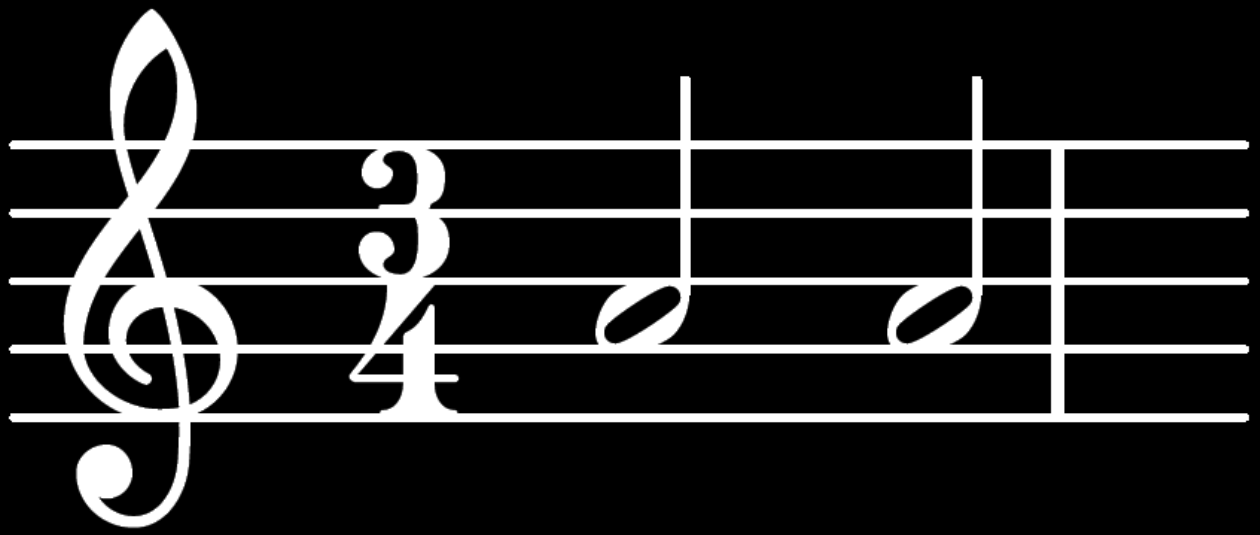
Simple things that a computer should know about music but doesn't…

Scales …



Simple things that a computer should know about music but doesn't…

Meter…

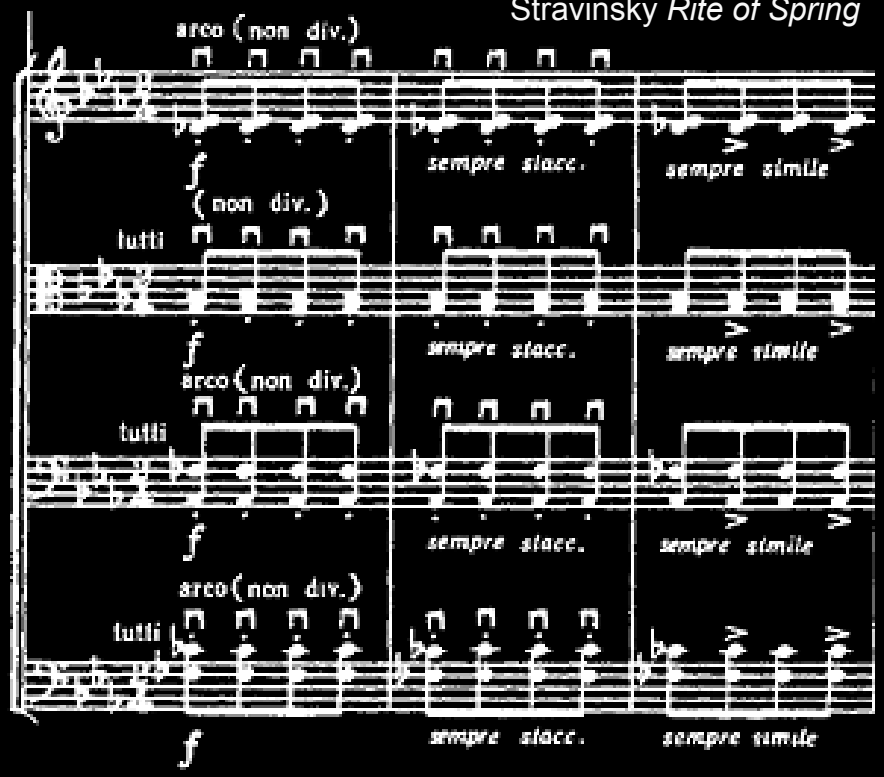


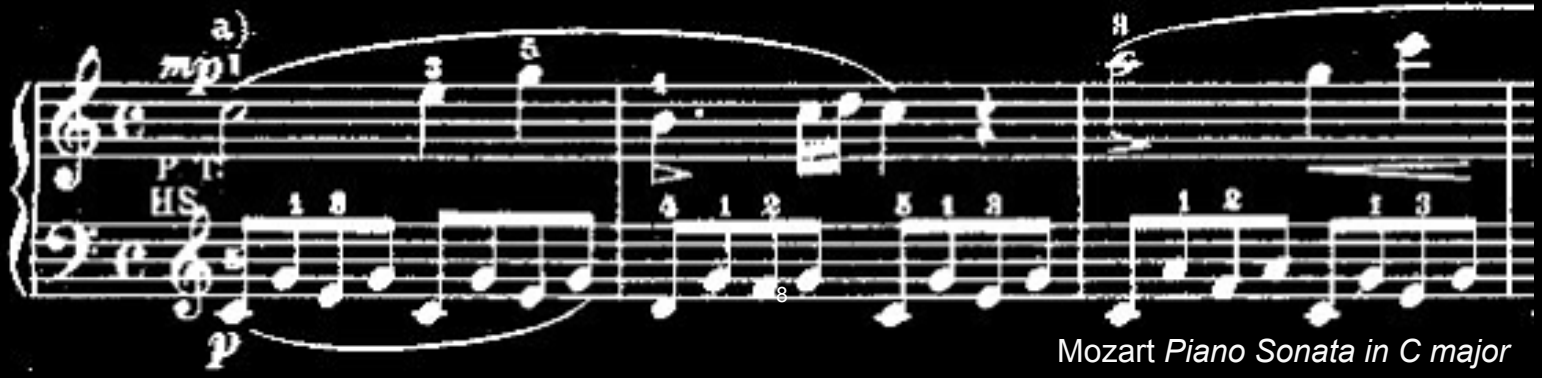
Simple things that a computer should know about music but doesn't

Elementary Harmony:

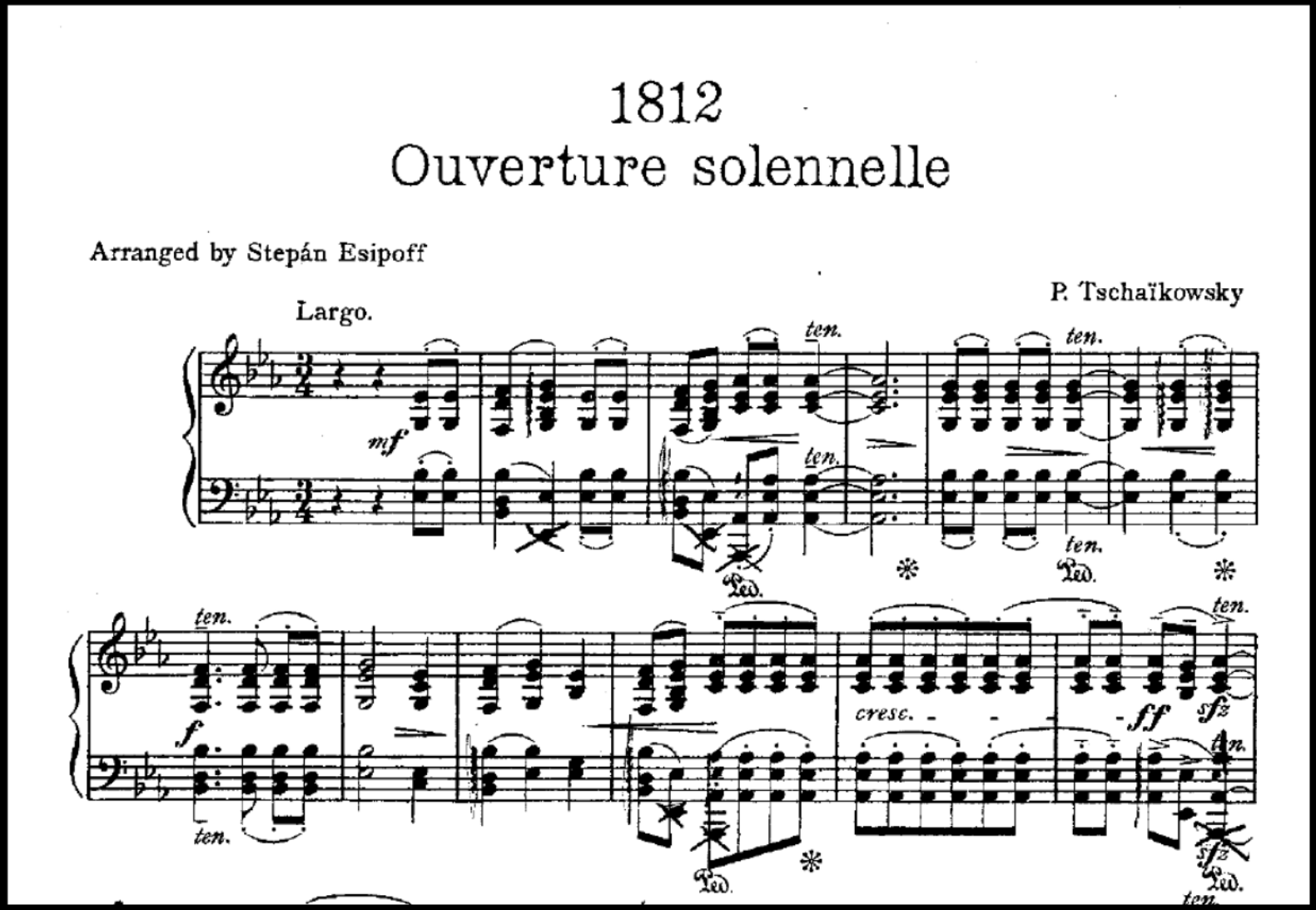
Which of these is more dissonant?

Stravinsky *Rite of Spring*

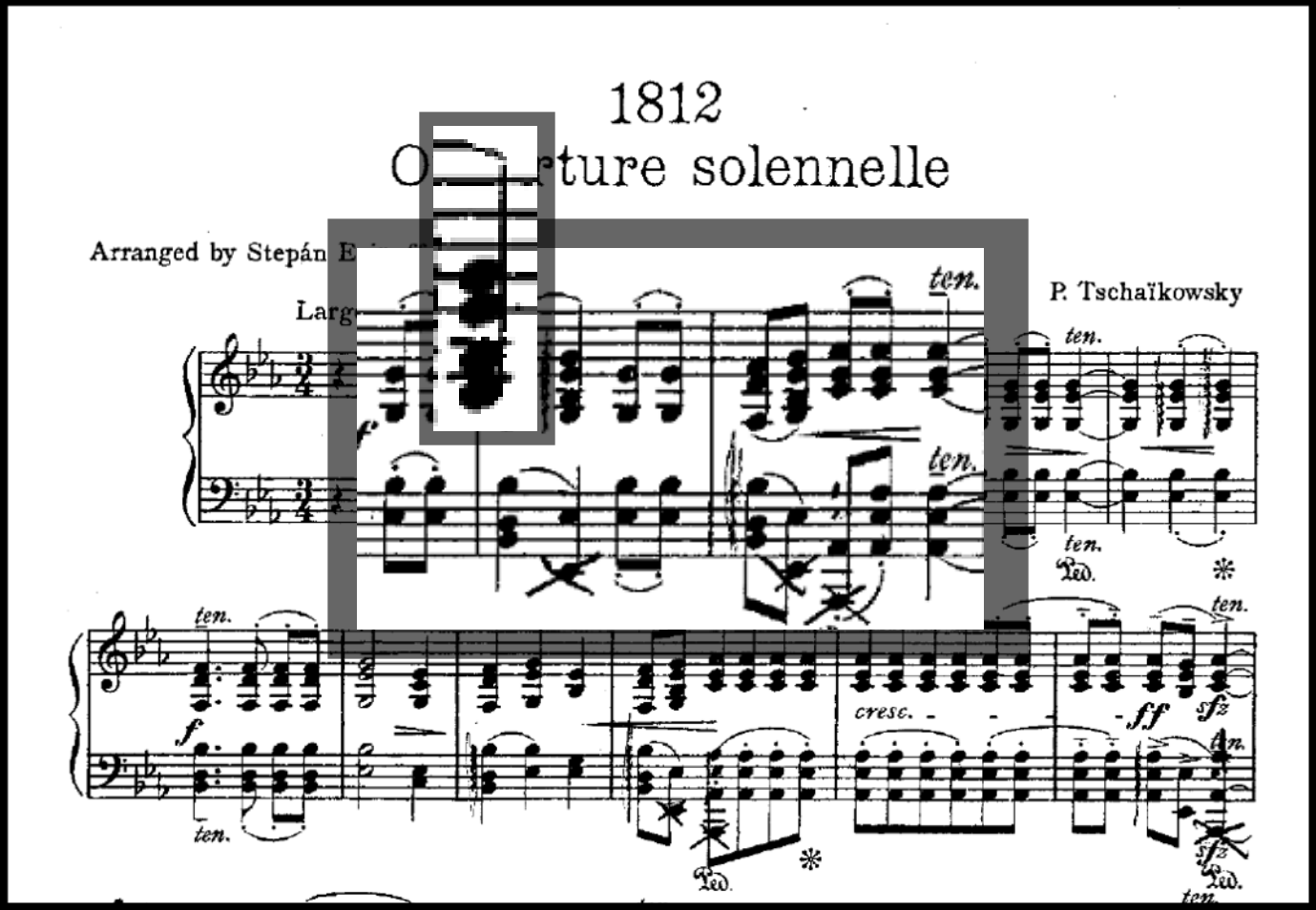




Mozart *Piano Sonata in C major*

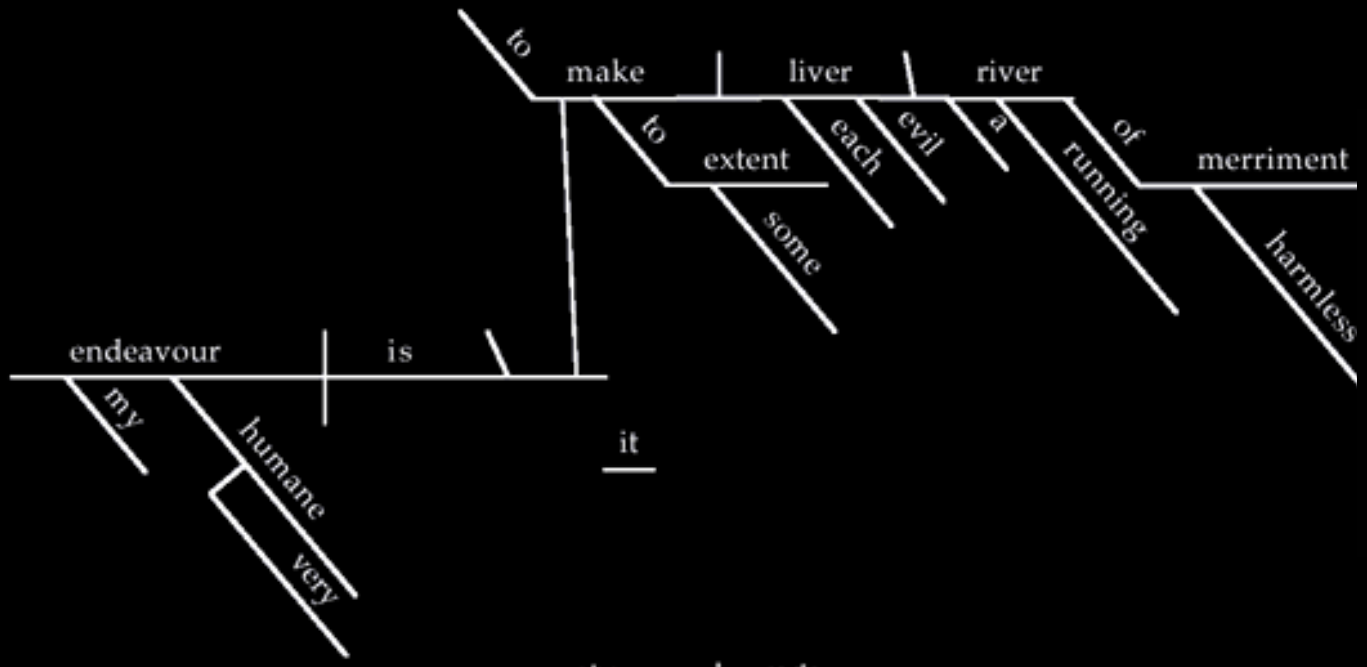


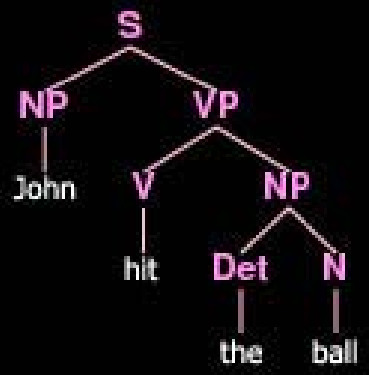
The problem with doing large-scale analysis is that in order to say something useful about, say, the key of a piece, you need a system that can identify and work with a phrase…

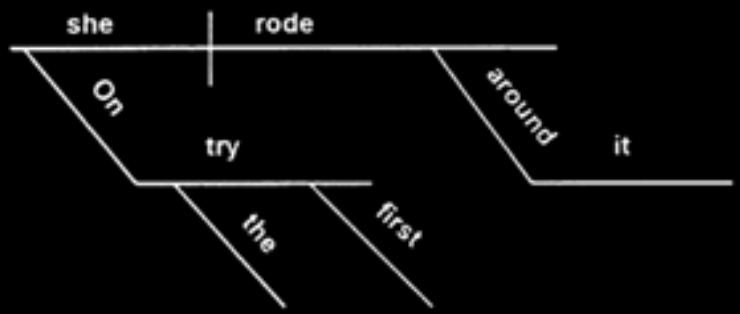


…and in order to work with phrases you need an object model that understands chord progressions. And in order to work with progressions you need to be able to classify individual chords…and to do that you need a robust way of modeling notes. So a small problem becomes a bigger and bigger problem.

Hypothetical Example from Computer-aided Linguistics: Using Sentence Structure to find Spam E-Mails







Hypothetical Example from Computer-aided Linguistics: Using Sentence Structure to find Spam E-Mails

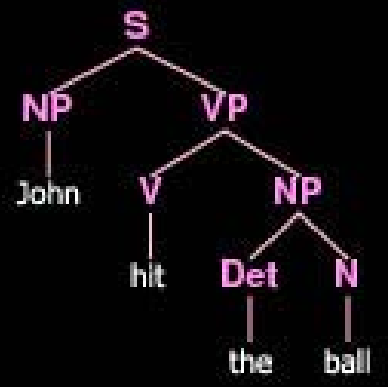
Available Packages for Sentence Parsing (a quick search):

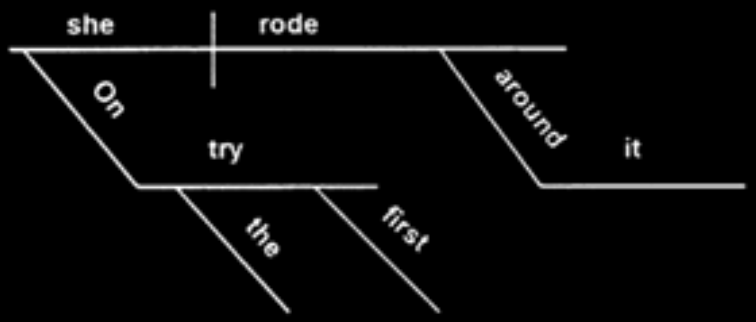
Stanford Natural Language Processing Group

Apple Pie Parser (NYU)

SenDraw @ University of Central Florida

OpenNLP sentence parser





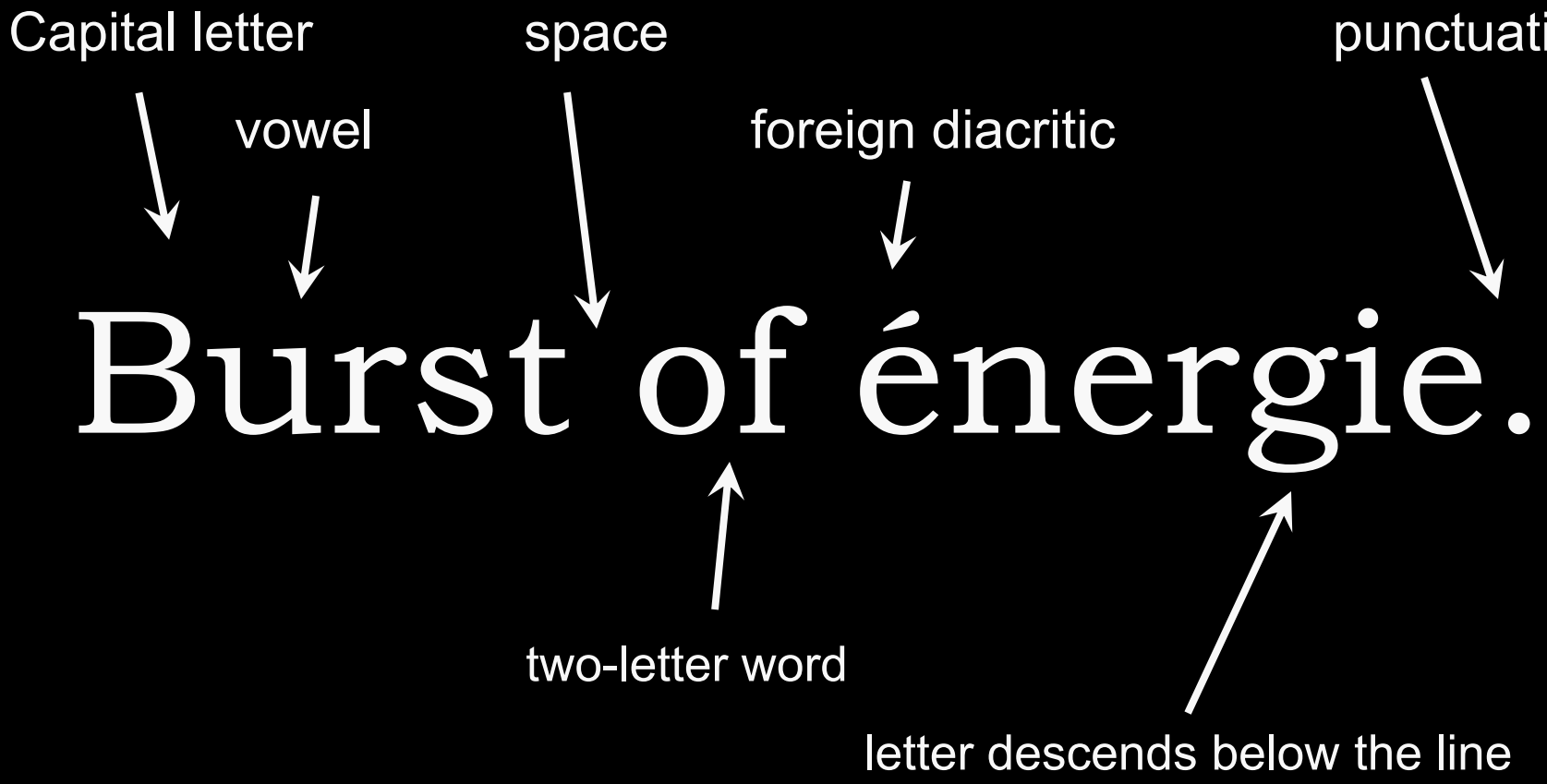
Hypothetical Example from Computer-aided Linguistics: Using Sentence Structure to find Spam E-Mails

[Screenshots of online thesarus, dictionary, and word lists removed due to copyright restrictions.]

Hypothetical Example from Computer-aided Linguistics: Recognition of text characteristics that we take for granted

Burst of énergie.

Hypothetical Example from Computer-aided Linguistics: Recognition of text characteristics that we take for granted



Examples of Humdrum (David Huron) code:

Problem: “Do the words ‘high’, ‘hoch’, or ‘haut’ coincide with high notes in the text?”

Solution: Compare “semits * | text | extract -i '**semits,**text' | ditto -s = \ | egrep -i '^*|high|hoch|haut' | extract -i '**semits' | stats” to “semits * | extract -i '**semits' | ditto -s = | rid -GLI \ | stats”

Problem: “Convert one of Arnold Schoenberg’s 12-tone rows to a 12-tone matrix.”

Solution: X=0 ; while [$X -ne 12] ; do ; reihe -a -P $X $3 | rid -GLId | head -$2 > P$X ; reihe -a -I $X $3 | rid -GLId | head -$2 > I$X ; reihe -a -R $X $3 | rid -GLId | head -$2 > R$X ; reihe -a -RI $X $3 | rid -GLId | head -$2 > RI$X ; let X=$X+1 ; done

music21: a toolkit for computer-aided musicology

What is music21?

Music21 is a set of tools for helping scholars and other active listeners answer questions about music quickly and simply. If you've ever asked yourself a question like, "I wonder how often Bach does *that*" or "I wish I knew which band was the first to use these chords in this order," or "I'll bet we'd know more about Renaissance counterpoint (*or* Indian ragas *or* post-tonal pitch structures *or* the form of minuets) if I could write a program to automatically write more of them," then music21 can help you with your work.

How simple is music21 to use?

Extremely. After starting Python and typing "`from music21 import *`" you can do all of these things with only a single line of music21 code:

Display a short melody in musical notation:

```
tinyNotation.TinyNotationStream("c4 d8 f g16 a g f#", "3/4").show()
```

Hear the interval of a quarter tone (*coming soon!*):

```
interval.ChromaticInterval(0.5).show("midi")
```

Print the twelve-tone matrix for a tone row (in this case the opening of Schoenberg's Fourth String Quartet):

```
print (serial.rowToMatrix([2,1,9,10,5,3,4,0,8,7,6,11]) )
```

or since all the 2nd-Viennese school rows are already available as objects, you can type:

```
print (serial.RowSchoenbergOp37().matrix() )
```

Convert a file from Humdrum's **kern data format to MusicXML for editing in Finale or Sibelius:

```
parse('/users/cuthbert/documents/composition.krn').write('xml')
```

With five lines of music21 code or less, you can:

```
def closedPosition(self):
    '''
    returns a new Chord object with
    >>> chord1 = Chord(["C#4", "G5",
    >>> chord2 = chord1.closedPositi
    >>> print(chord2.lily.value)
    <cis' e' g'>4
    '''
    newChord = copy.deepcopy(self)
    tempChordNotes = newChord.pitche
    chordBassPS = self.bass().ps
    for thisPitch in tempChordNotes:
        while thisPitch.ps > chordBa
            thisPitch.octave = thisP
    newChord.pitches = tempChordNote
```

- Get Started with music21
- Browse the music21 documentation
- Download music21 from Google Code
- Get our latest news and updates at the music21 blog
- Read the Frequently Asked Questions list
- Sign up for the music21list mailing list through Google Groups.

LATEST MUSIC21 NEWS:

music21 at ISMIR

Chris and I will be presenting music21 as an oral paper and poster at the International Society for Music Information Retrieval in Utrecht this August. We hope we're able to meet many of you there! Chris will also be talking about the meterical features of the system at the International Computer Music Conference in NY in early

`music21`: libraries for importing and exporting many formats

In only	*In & Out*	*Out only*
abc	MusicXML	Lilypond
MuseData Stage 2	MIDI	Orange
MD Stage 1	Scala	Weka
Humdrum/KERN		Braille

NoteworthyComposer

romanText (Tymoczko or Clercq/Temperley flavors)

Monophonic audio

Simpler

Problem: “Convert one of Arnold Schoenberg’s 12-tone rows to a 12-tone matrix.”

Humdrum Solution: X=0 ; while [$X -ne 12] ; do ; reihe -a -P $X $3 | rid -GLId | head -$2 > P$X ; reihe -a -I $X $3 | rid -GLId | head -$2 > I$X ; reihe -a -R $X $3 | rid -GLId | head -$2 > R$X ; reihe -a -RI $X $3 | rid -GLId | head -$2 > RI$X ; let X=$X+1 ; done
(produces 48 files on disk)

Simpler

Problem: “Convert one of Arnold Schoenberg’s 12-tone rows to a 12-tone matrix.”

0	2	11	7	8	3	9	1	4	10	6	5
10	0	9	5	6	1	7	11	2	8	4	3
1	3	0	8	9	4	10	2	5	11	7	6
5	7	4	0	1	8	2	6	9	3	11	10
4	6	3	11	0	7	1	5	8	2	10	9
9	11	8	4	5	0	6	10	1	7	3	2
3	5	2	10	11	6	0	4	7	1	9	8
11	1	10	6	7	2	8	0	3	9	5	4
8	10	7	3	4	11	5	9	0	6	2	1
2	4	1	9	10	5	11	3	6	0	8	7
6	8	5	1	2	9	3	7	10	4	0	11
7	9	6	2	3	10	4	8	11	5	1	0

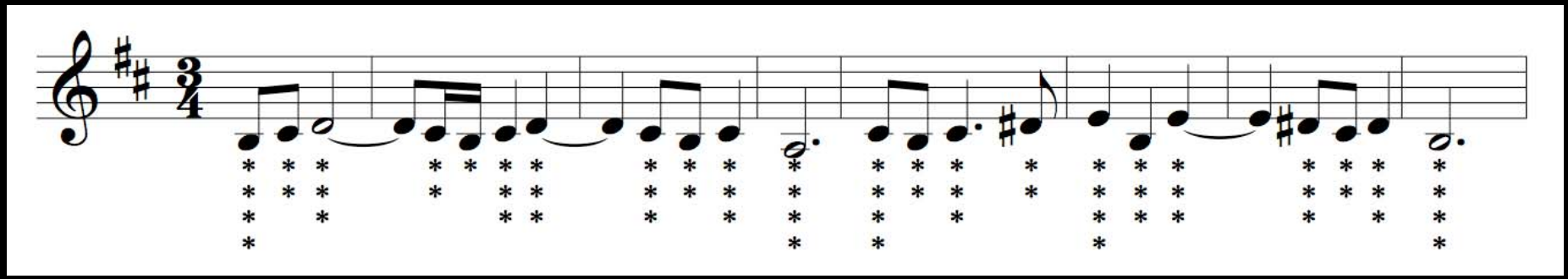
Music21 solution:

```
print serial.rowToMatrix([0,2,11,7,8,3,9,1,4,10,6,5])
```

Built in Analytical Libraries

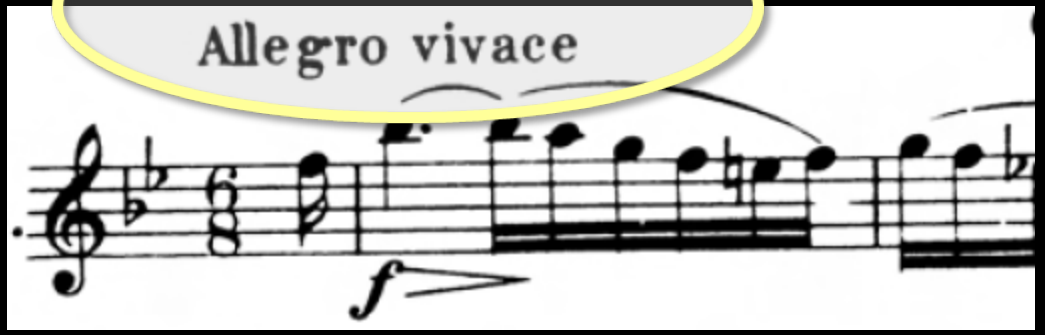
Metrical Analysis

```
>>> from music21 import *
>>> bwv11_6 = corpus.parse('bach/bwv11.6.xml')
>>> alto = bwv11_6.parts['alto']
>>> excerpt = alto.measures(13,20)
>>> analysis.metrical.labelBeatDepth(excerpt)
>>> excerpt.show()
```



Problems of automated CMN analysis

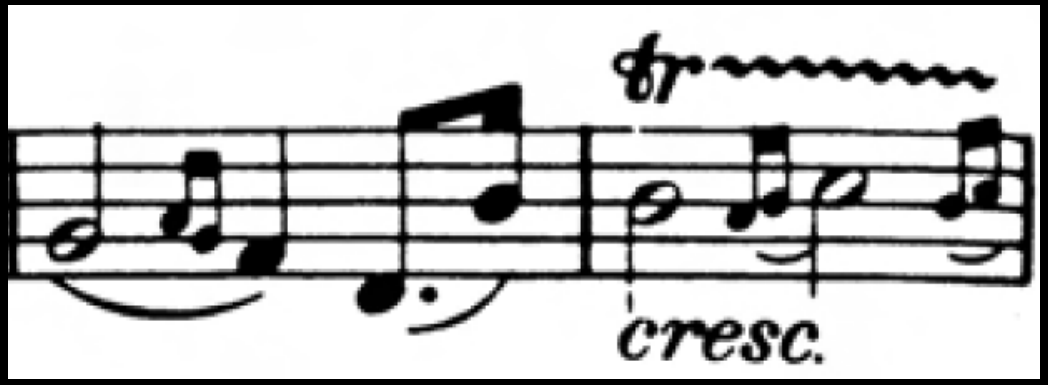
Pickups



Cadenzas

On the beat

Before the beat

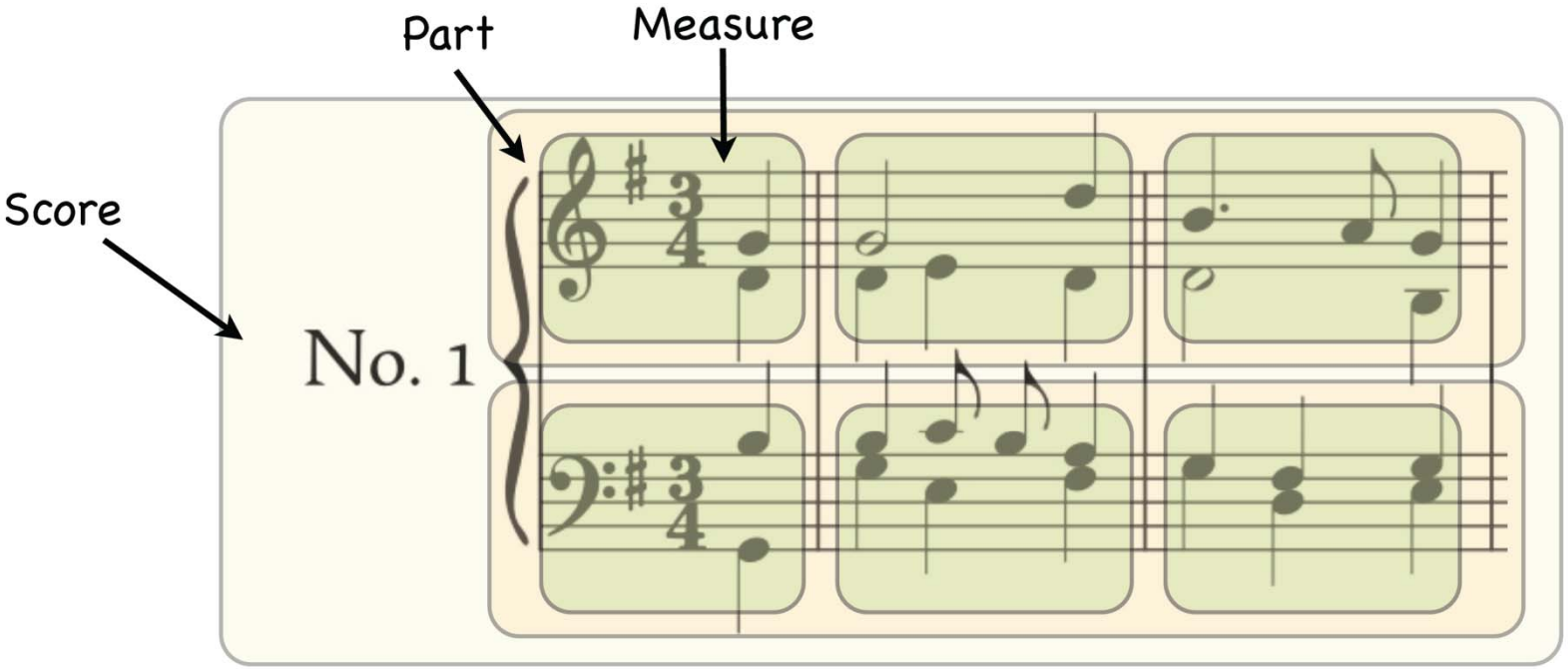


The next 3 slides are by William Andrew Burnson

REPRESENTATION

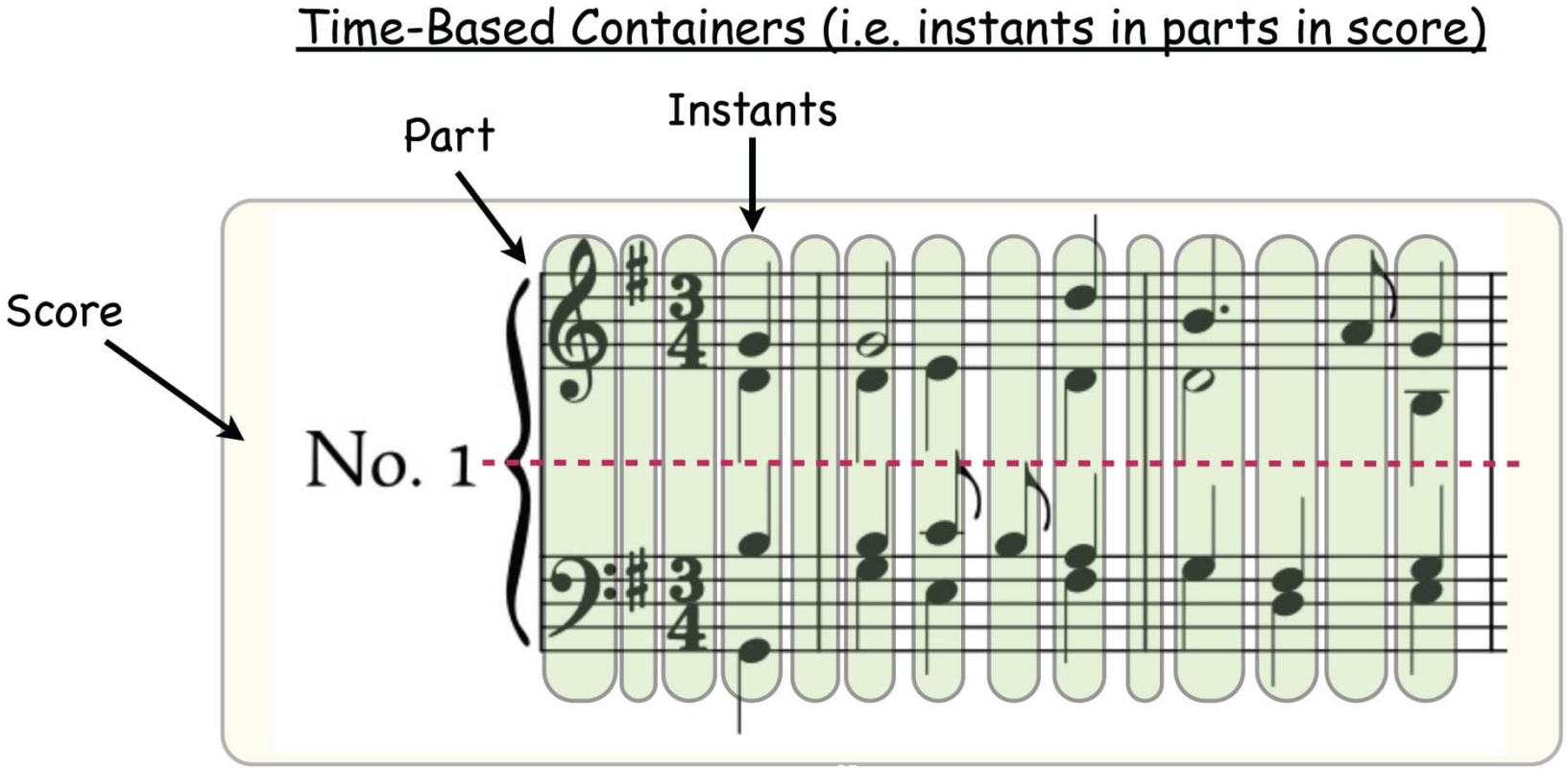
A pervasive approach to representing music comprises part-based (row first) multidimensional containers.

Part-Based Containers (i.e. measures in parts in score)



REPRESENTATION

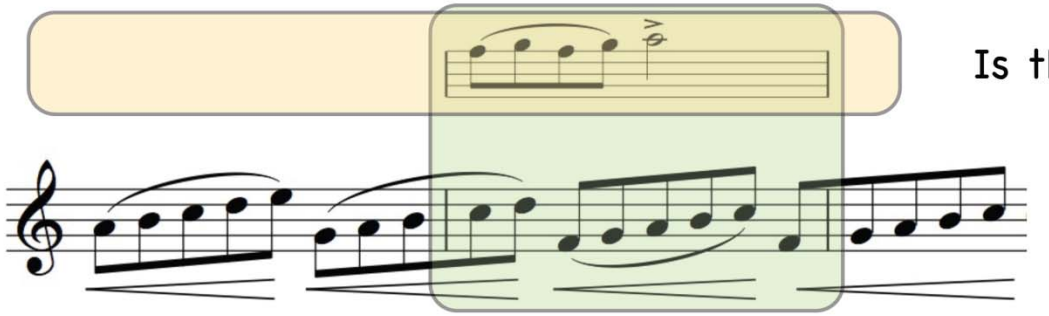
Alternatively, a time-based (column first) container can be used



REPRESENTATION

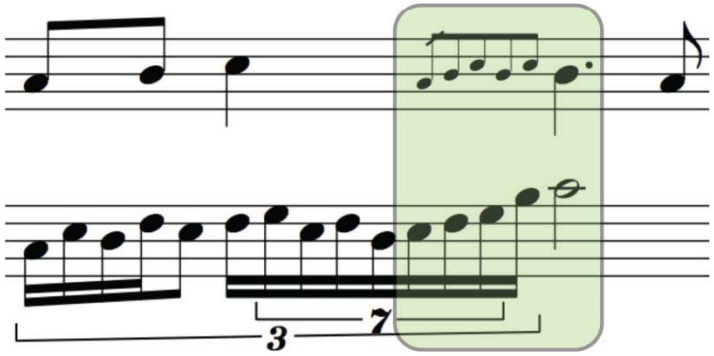
Caveats of a container-based representation:

1) Example of Part-Based failure: Ossia Staff



Is this a part with hidden measures?

2) Example of Time-Based failure: Grace notes



In what order should the notes appear?

If it’s such a pain, why do it?

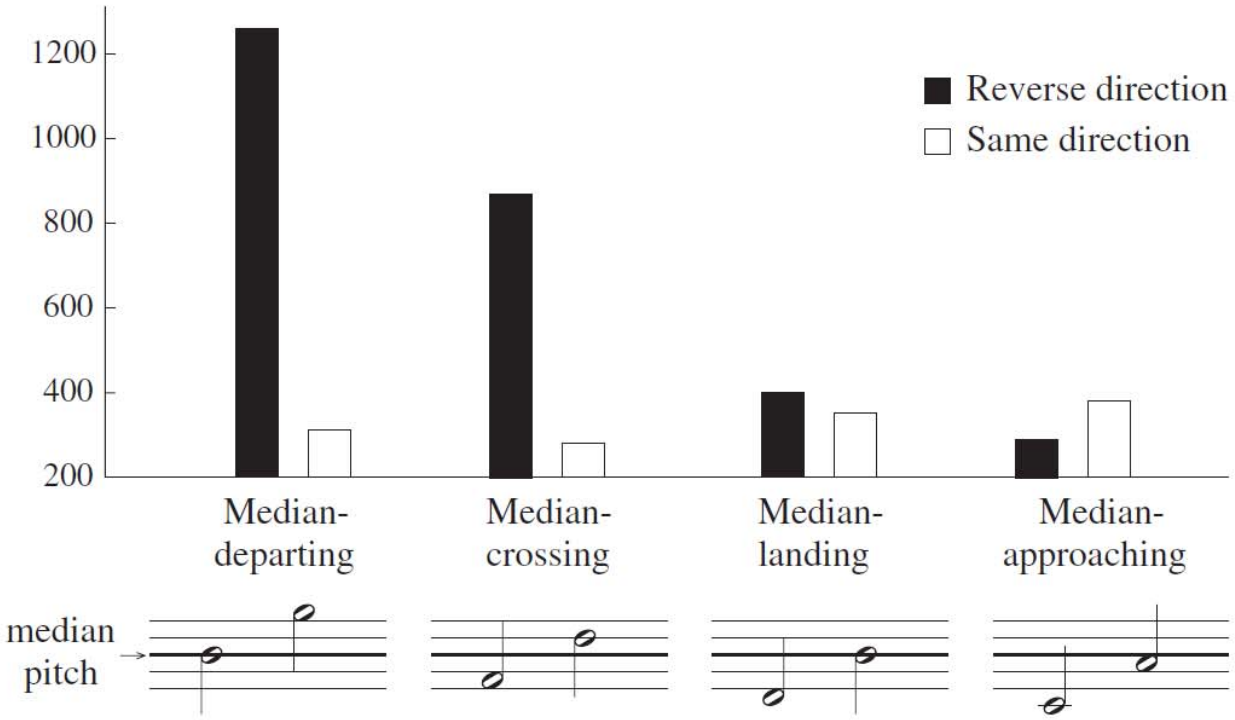
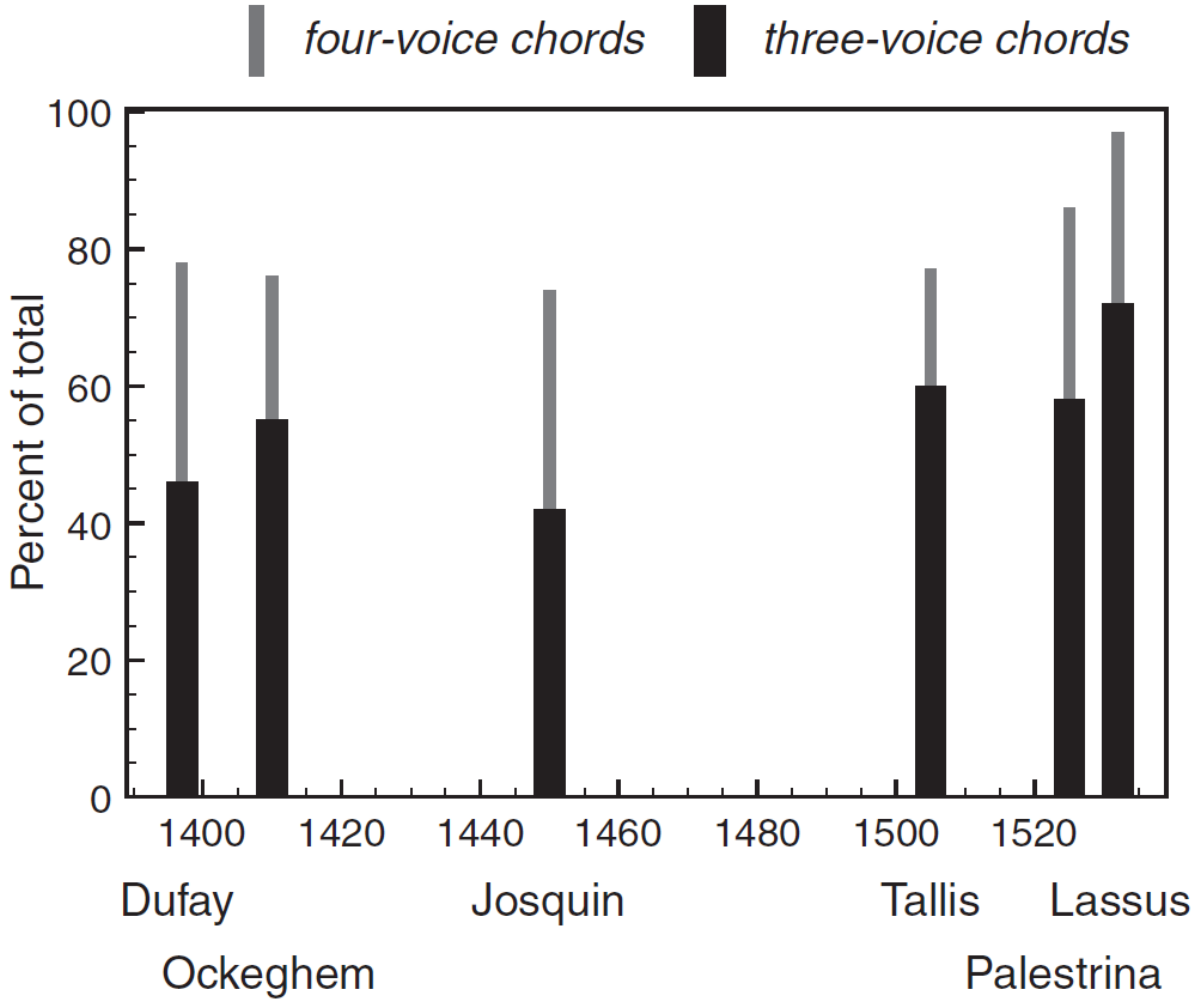


Figure 5.5

Number of instances of various melodic leaps found in a cross-cultural sample of melodies. Most large intervals that approach the median pitch continue in the same melodic direction. Large intervals that land on the median pitch are as likely to continue in the same direction as to reverse direction. Results support the phenomenon of melodic regression and fail to support post-leap reversal. From von Hippel and Huron 2000.

Figure 6.3.5 The percentage of consonant sonorities that are complete triads rather than "doubled" intervals, by composer date of birth. The percentage is reasonably high throughout the Renaissance, and increases over time.



Some domains for Computational Musicology

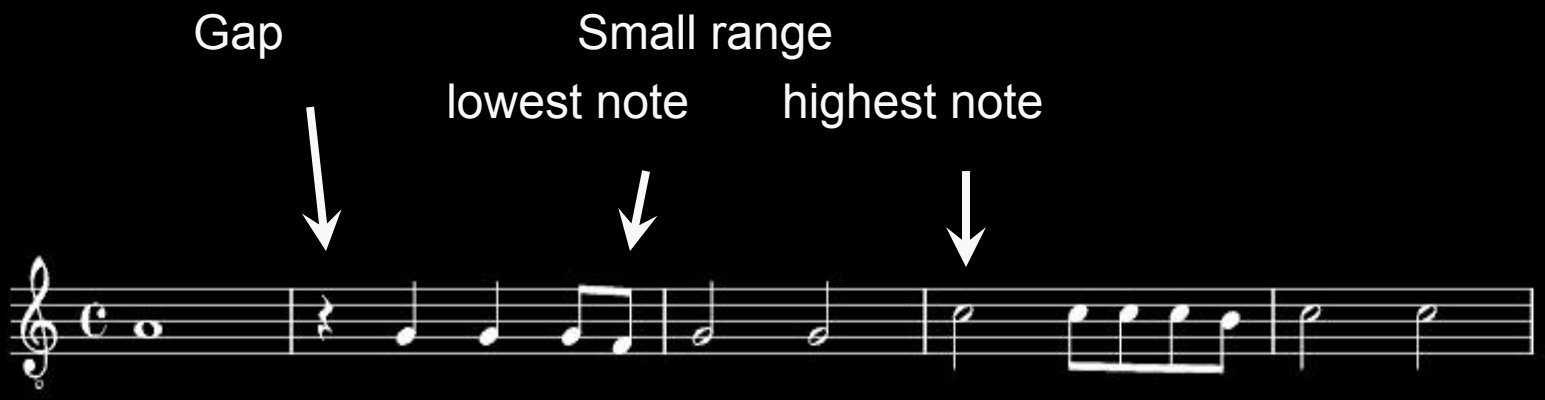
Mozart accompaniment patterns

Aaron Copland's harmonies

GIS and Gregorian Chant

Authenticity and Bach's Works

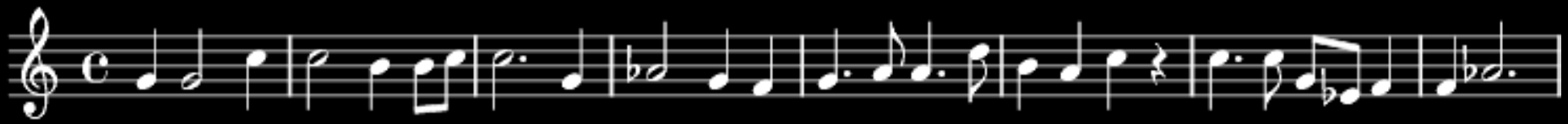
Cavalli, *Gli amori d'Apollo e di Dafne*



Giá dell' Al – ba vi – ci - na L'au- re pre-cor-ri-tri - ci.

Already daybreak is near The morning air outruns us.

Gluck, *Orfeo and Euridice* + Random notes



Gluck's pitches

Randomly generated

Some domains for Computational Musicology

Mozart accompaniment patterns

Aaron Copland's harmonies

GIS and Gregorian Chant

Authenticity and Bach's Works

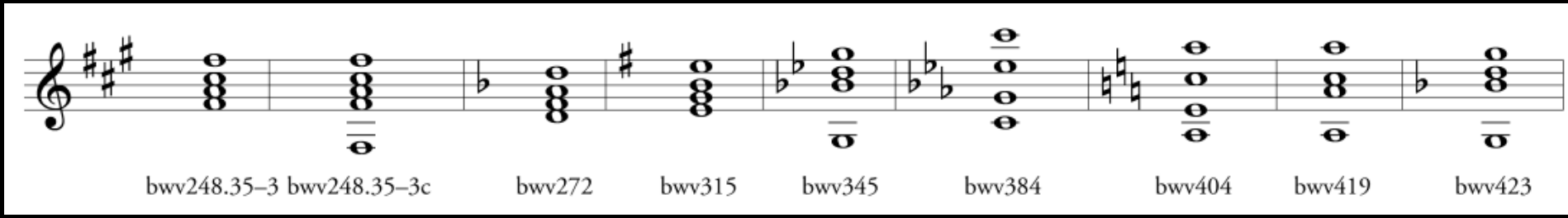
Baroque Opera Arias

Randomness and Musical Style

Algorithmic Composition

Bach usually ends minor-mode pieces in major. When does he end them in minor?

```
results = stream.Stream()
for chorale in corpus.bachChorales:
  analyzedKey = chorale.analyze('key')
  if analyzedKey.mode == 'minor':
    lastChordPitches = []
    for part in chorale.parts:
        lastChordPitches.append(part.flat.pitches[-1])
    lastChord = chord.Chord(lastChordPitches)
    lastChord.duration.type = "whole"

    if lastChord.isMinorTriad() is True:
        m = stream.Measure()
        m.keySignature = chorale.flat.getElementsByClass('KeySignature')[0]
         m.append(lastChord)
         results.append(m)
results.show()
```


bwv248.35–3
bwv248.35–3c
bwv272
bwv315
bwv345
bwv384
bwv404
bwv419
bwv423

Examples and Demonstrations — music21 v0.3.1a6 documentation - Mozilla Firefox

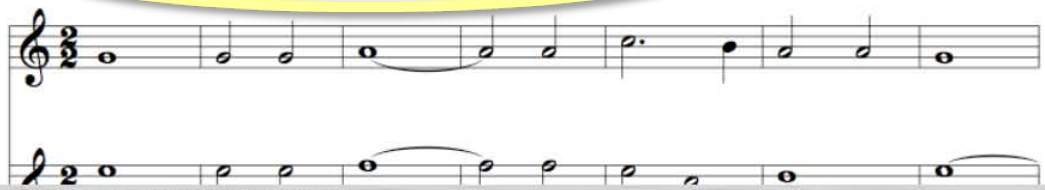
http://mit.edu/music21/doc/html/examples.html#creating-a-reduction-and-labeling-intervals

Michael Scott Cuthbert and cuthber... | Google Desktop: Re: oh right... | Gmail - Search results - cuthbert@g... | Examples and Demonstrations — ...

Creating a Reduction and Labeling Intervals

This example, after parsing a polyphonic work stored as an Opus and creating a Score, presents and labels the intervals of the resultant chords of all distinct harmonies:

```
from music21 import corpus
# Parse an Opus, a collection of Scores
o = corpus.parseWork('josquin/laDeplorationDeLaMorteDeJohannesOckeghem')
# Create a Score from a Measure range
sExcerpt = o.mergeScores().measures(127, 134)
# Create a reduction of Chords
reduction = sExcerpt.chordify()
# Iterate over the Chords and prepare presentation
for c in reduction.flat.getElementsByClass('Chord'):
    c.closedPosition(forceOctave=4, inPlace=True)
    c.removeRedundantPitches(inPlace=True)
    c.annotateIntervals()
# Add the reduction and display the results
sExcerpt.insert(0, reduction)
sExcerpt.show()
```



- Searching the Corpus by Locale
- Finding Chords by Root and Collecting their Successors
- Pitch and Duration Transformations
- Basic Counting of and Searching for Musical Elements

Previous topic

Overview: Meters, Time Signatures, and Processing Beams, Accents, and Beats

Next topic

Installing Music21

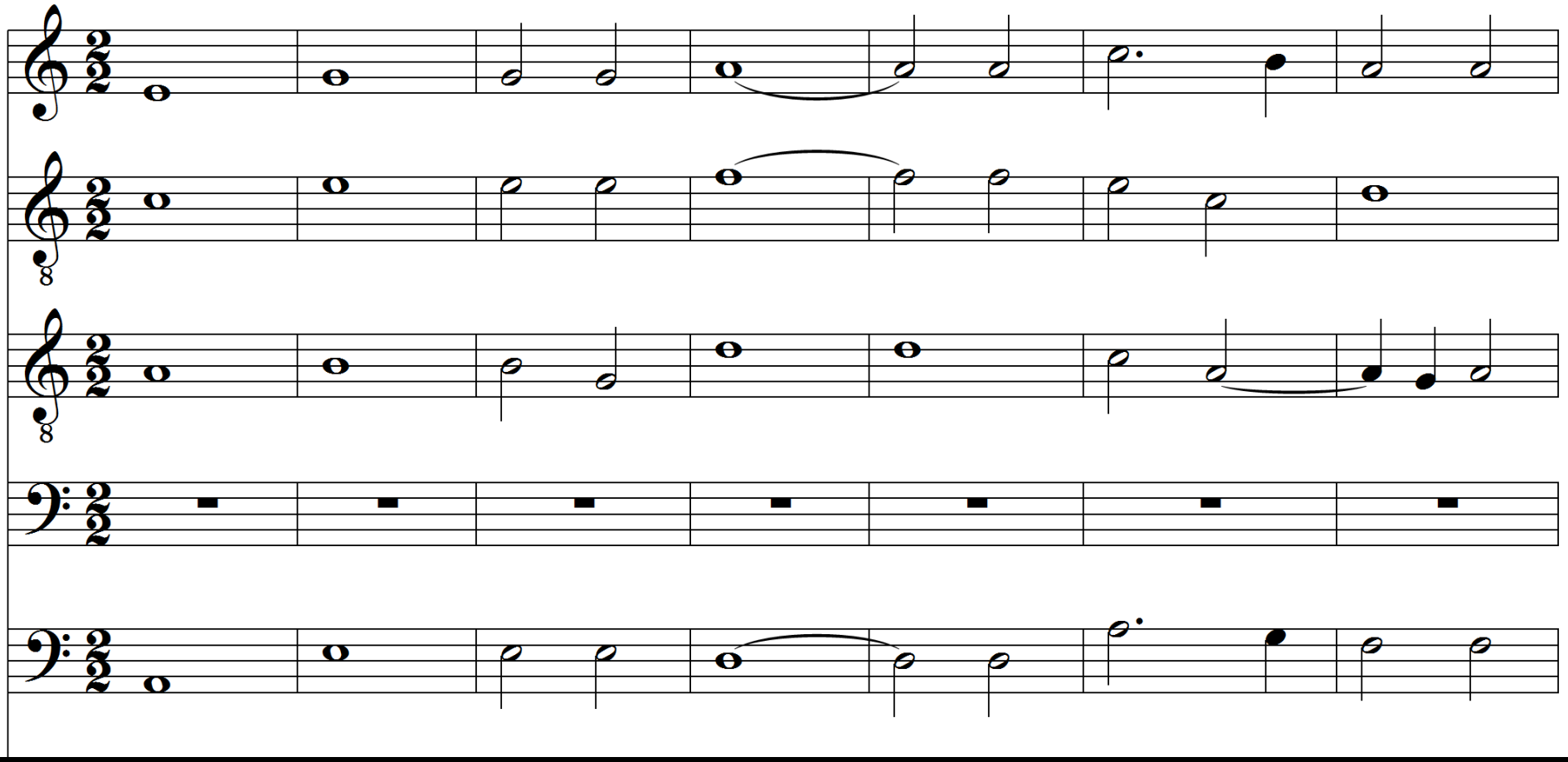
This Page

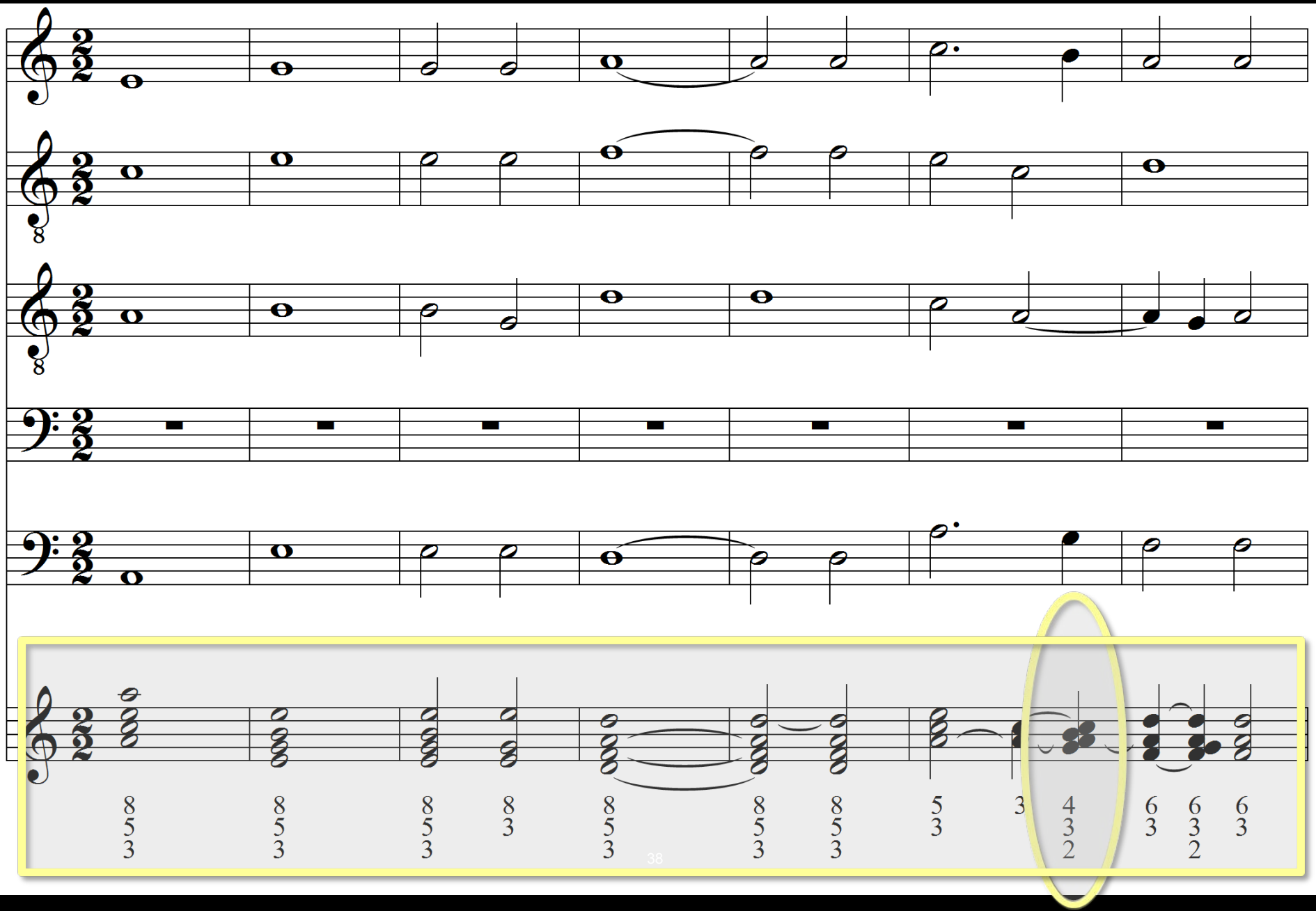
Show Source

Quick search

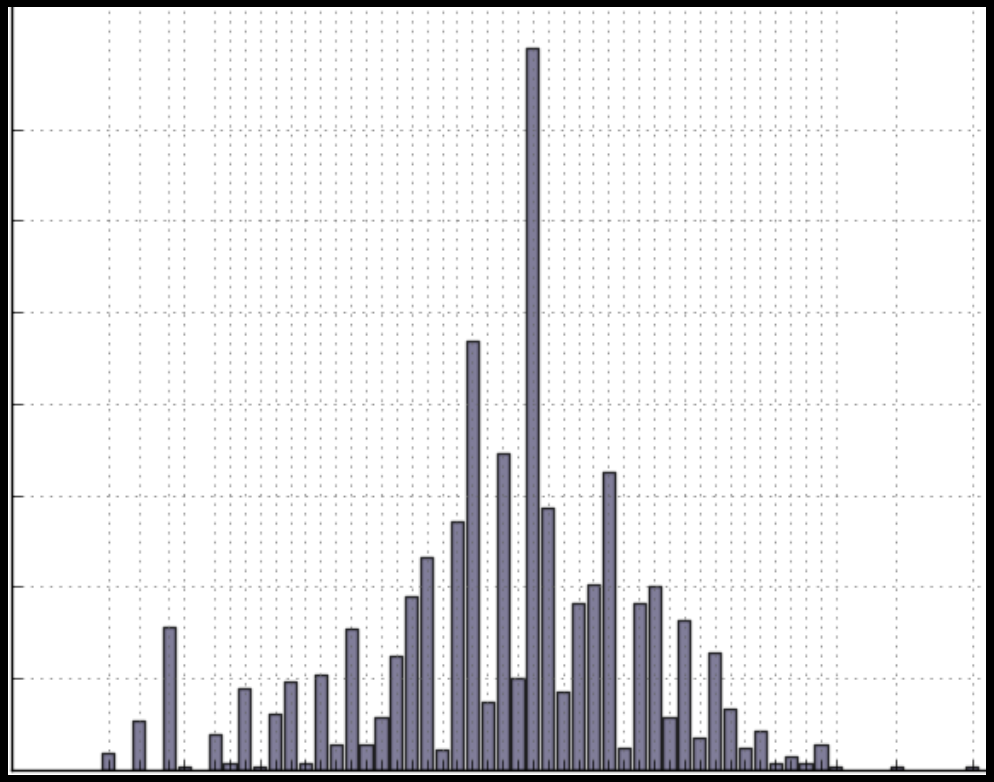
http://mit.edu/music21/doc/html/examples.html#pitch-and-duration-transformations

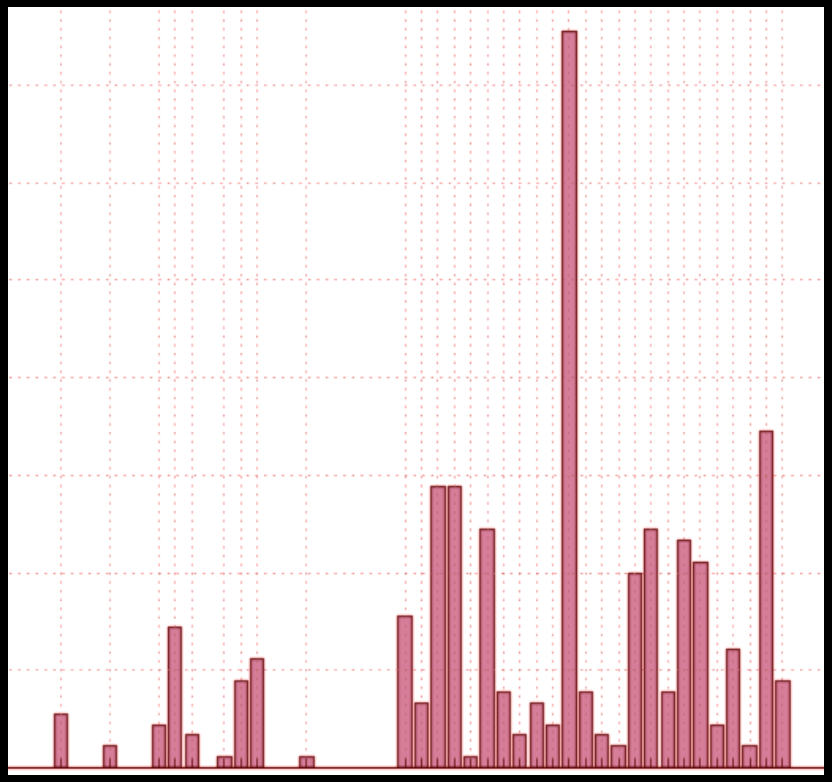
Slide 75 of 79 | English (United States) | AutoPreview




8 5 3
8 5 3
8 5 3
8 3
8 5 3
8 5 3
8 5 3
5 3
3
4 3 2
6 3
6 3 2
6 3

```
schumann =
corpus.parseWork('schumann/opus41no1', 3)
schumann.plot('pitch')
```





Schumann

Chopin

Feature Extraction with `music21`

Why a new Feature Extraction System after jSymbolic?

Why go beyond MIDI?

Some Elements MIDI Lacks

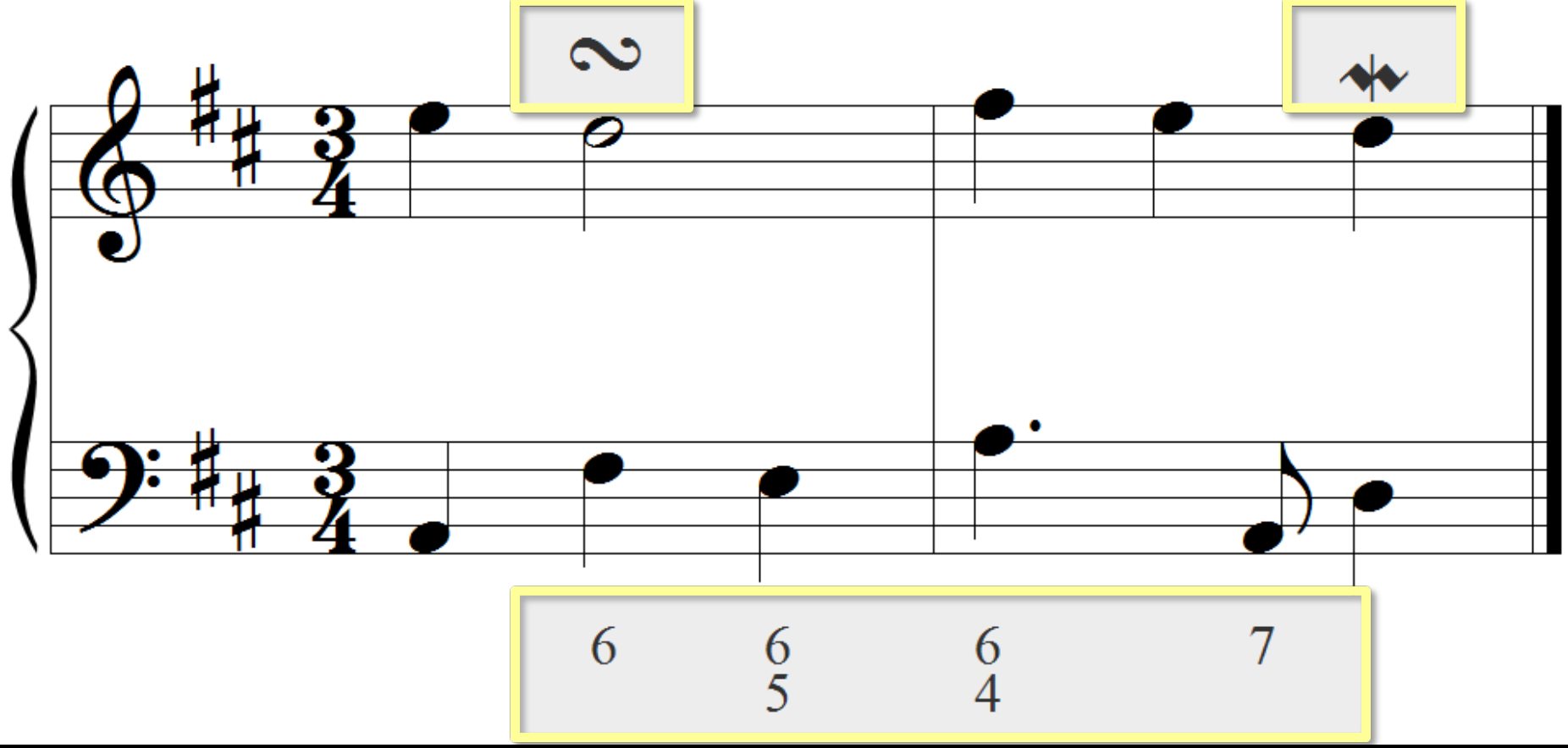
It does not record differences between enharmonic tones (such as E-flat and D-sharp),

It cannot precisely encode microtones (pitch bend objects being notoriously unreliable and not portable),

Few implementations support lyrics,

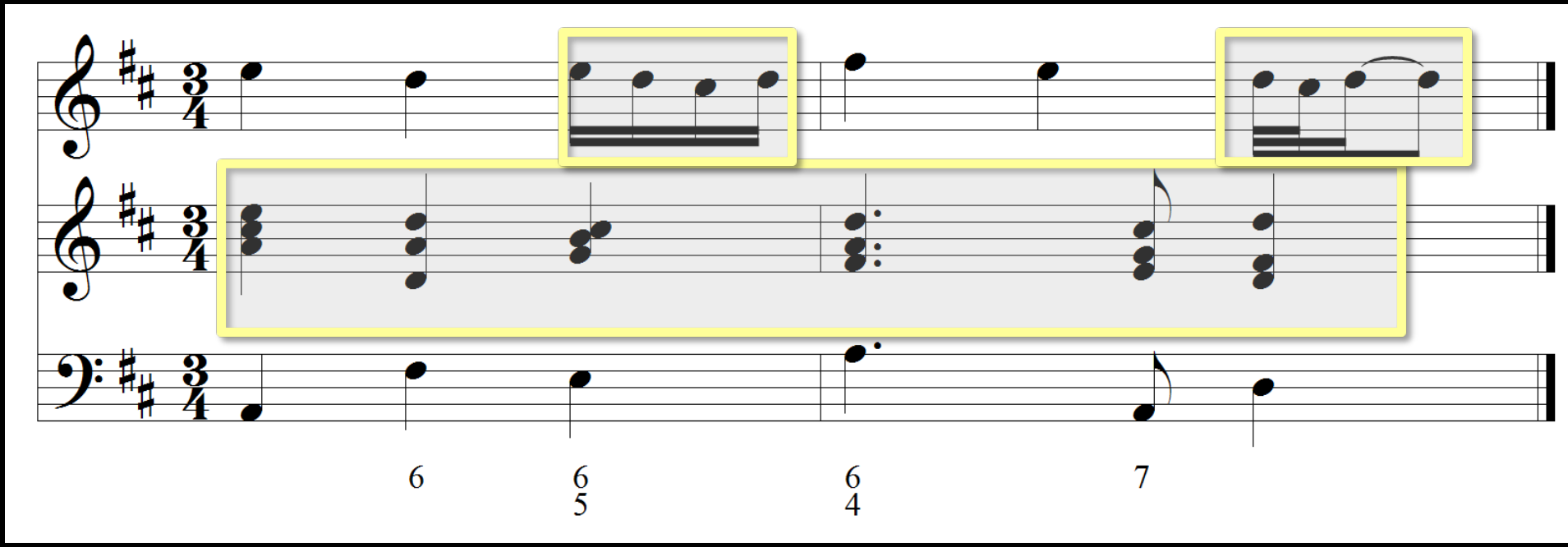
Differences between an eighth note and a staccato quarter is erased

Chord symbols are not supported, and so on.



[0.0, 0.0, 1.0, 0.0, 1.0, 0.0, 0.66…, 0.0, 0.0, 1.0, 0.0, 0.0]

Pitch class frequency (C, C#, D, D#/Eb, E, …) where 1.0 is the most common pitchclass, 0.5 = half as prevalent as the most common, pitch class, etc. C-sharp is highlighted

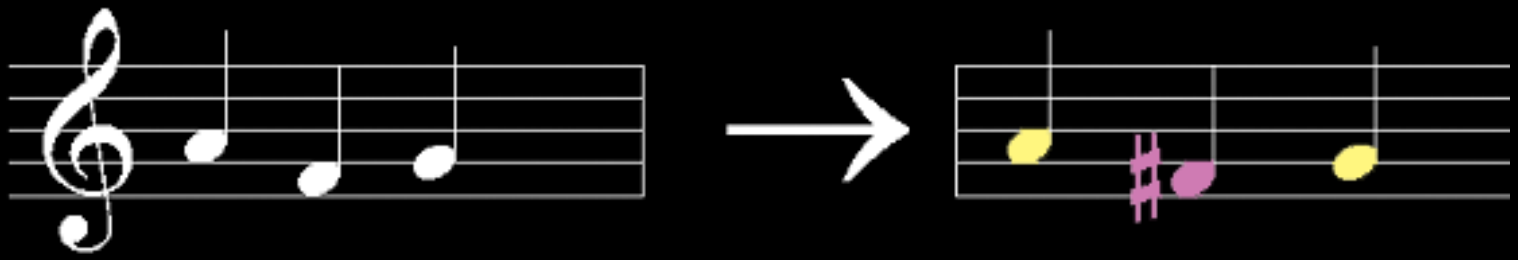


[0.0, 0.5, 1.0, 0.0, 0.6, 0.0, 0.4, 0.2, 0.0, 0.7, 0.0, 0.1]

After pitch realization C-sharp goes from being non-existent to one of the more prominent pitches.

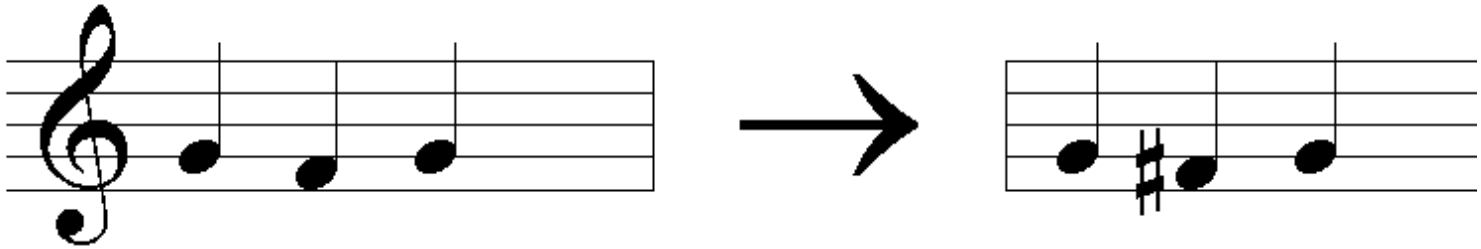
from The rules of “musica ficta” according to Nicolaus de Capua: Rule 3

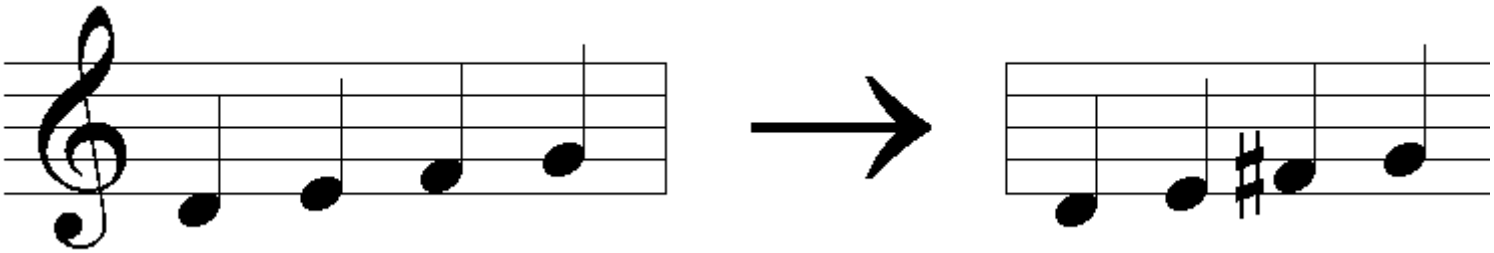
Nota quod quando cantus descendit ditonum sane et inmediate ascendit unam uocem. tunc de ditono debemus facere semiditonum.

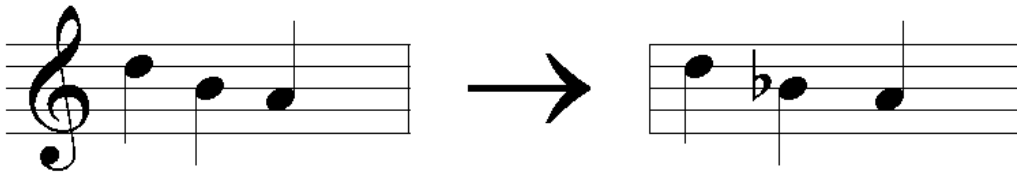


for further information, see Jan Herlinger, “Nicolaus de Capua, Antonio Zacara da Teramo, and *musica ficta*” in Francesco Zimei, editor, *Antonio Zacara da Teramo e il suo tempo* (Lucca: LIM, 2004), pp. 67–89.

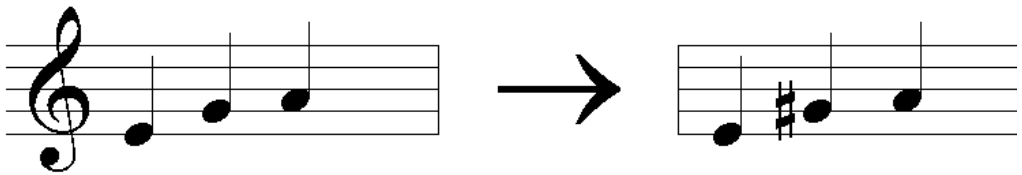
Nicolaus de Capua, the other three rules

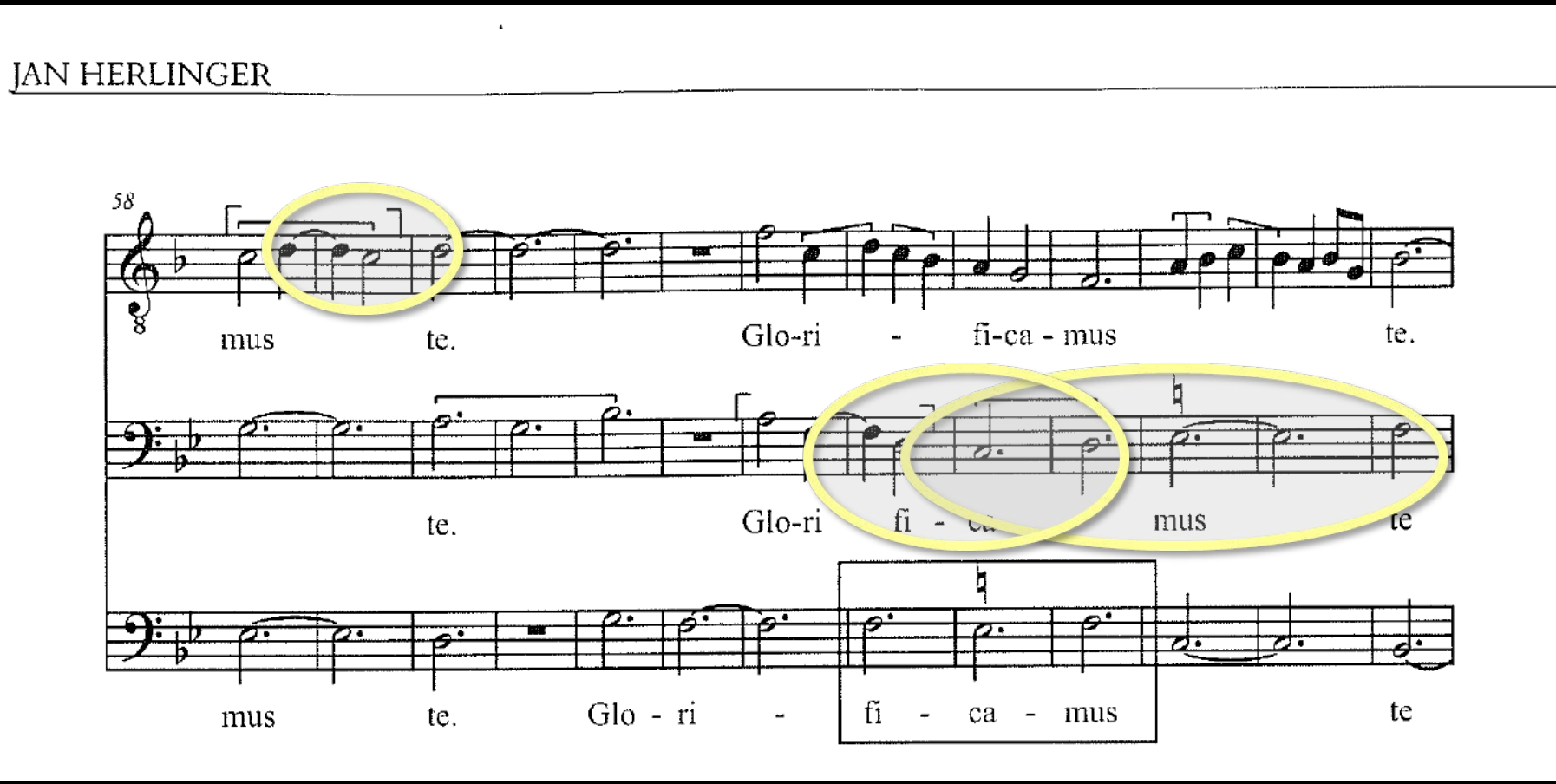






or





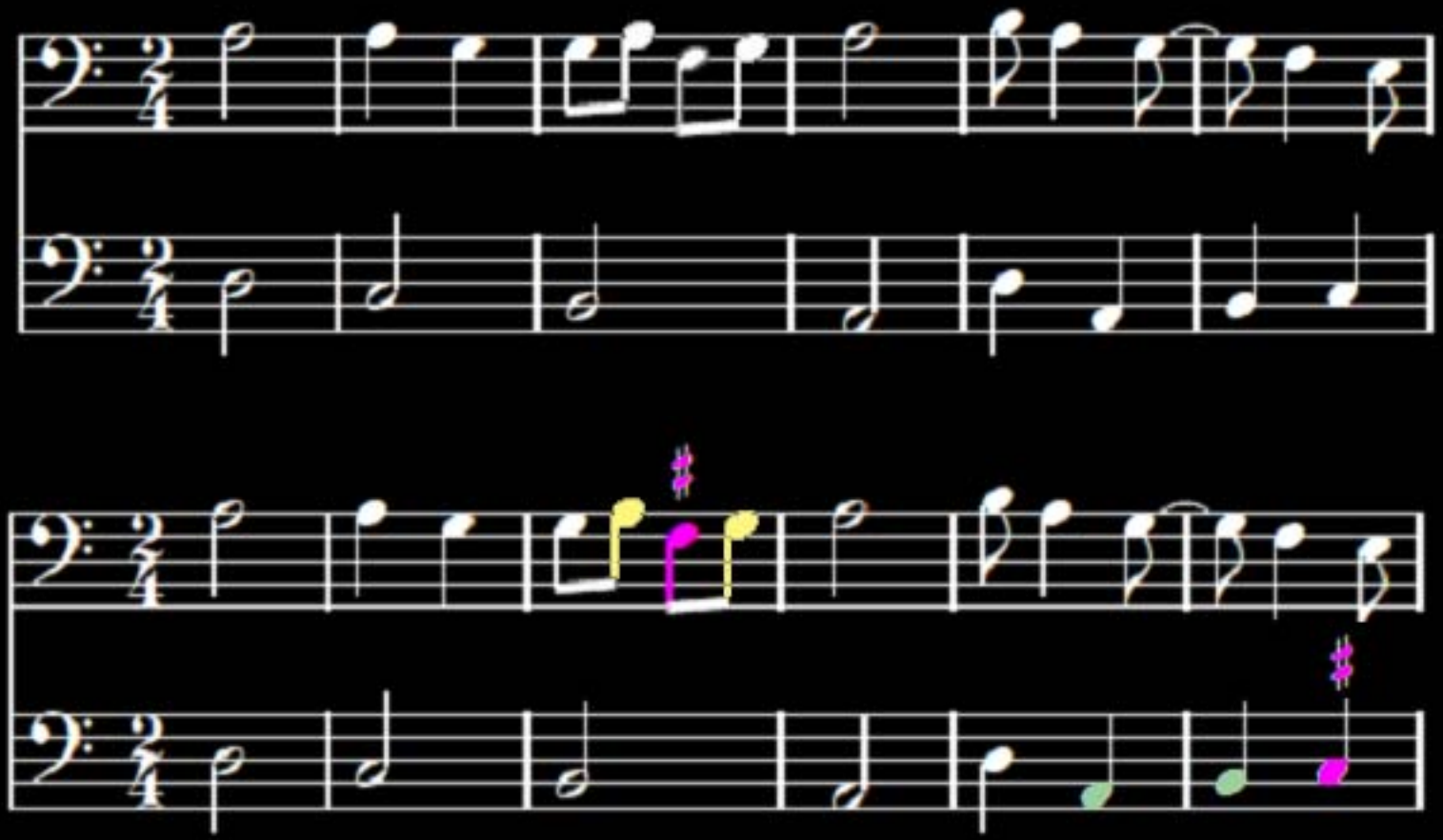
Zachara, Gloria “Rosetta”, PMFC 13. Herlinger, 2004, p. 72

```
def capuaRuleThree(stream):
    '''Applies Capua's third rule to the given stream, i.e. if there is a
    descending major third followed by an ascending major second, the second
    note will be made a half-step higher so that there is a descending minor
    third followed by an ascending minor second. Also changes
    note.editorial.color for rule 3 (green pink green).'''
    for i in range(0, len(stream.notes)-2):
        n1 = stream.notes[i]
        n2 = stream.notes[i+1]
        n3 = stream.notes[i+2]

        if n1.isRest or \
           n2.isRest or \
           n3.isRest:
             continue

        i1 = generateInterval(n1,n2)
        i2 = generateInterval(n2,n3)

        if n1.accidental is not None or \
           n2.accidental is not None or \
           n3.accidental is not None:
             continue

        ### never seems to improve things...
        if n2.step == "A" or n2.step == "D":
             continue

        # e.g., E C D => E C# D
        if i1.directedName  == "M-3" and \
           i2.directedName  == "M2":
            if (n2.editorial.misc.has_key("capua")):
                n2.editorial.misc['capua'] += RULETHREE
            else:
                n2.editorial.misc['capua'] = RULETHREE
            n2.editorial.ficta = Accidental("sharp")
            n2.editorial.misc["capua-ficta"] = Accidental("sharp")
            n1.editorial.color = "pink"
```

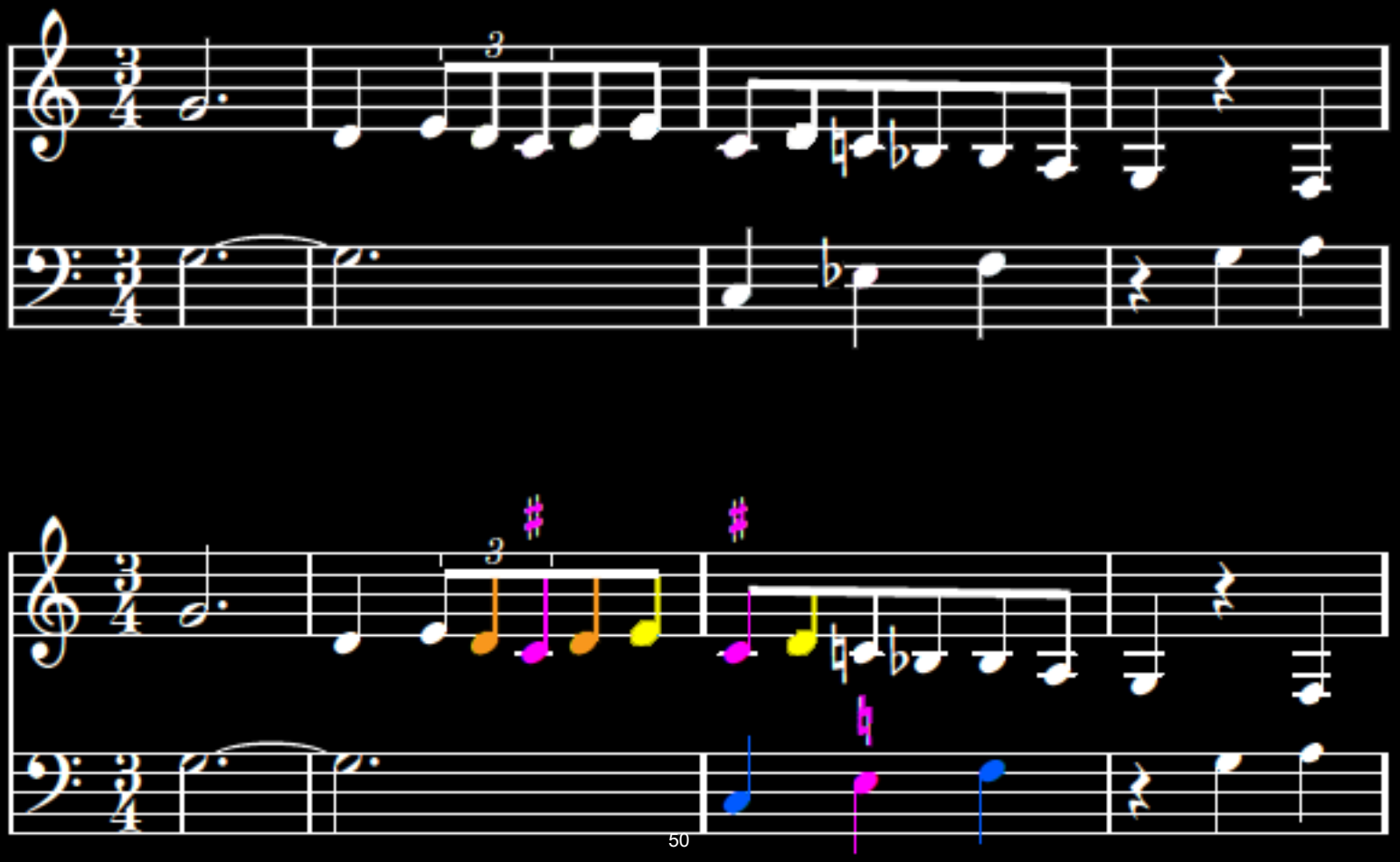
Encoded list of incipits and cadences of Trecento ballate

FischerNu	Incipit	pg start	pg end	begin time	incipit C	incipit T
2	Acurr' uomo	1	1	2/4	r4 a8 b c'2 ~ c' b4 r4	r4 d8 d c2 ~c4. d
3	Ad(d)io, amore mio					
12	Altro che sospirar	2	4	2/4	a2 r4 e r f# g r8 e g f e c	A2~ A A G F2~ F
16	Amor amaro					
22	Amore a lo to aspetto	5	6	2/4	A2 A8 B- c A A4 r8 A16	D2 D D
28	Amor, merçè	7	7	6/8	a4.~ a4 e8 e f d d r g	A2. c4. d4.
29	Amor mi fa cantar	8	8	9/8		
30	Amor mi stringe	9	10	2/4	g2 a4 g f16 e d8 r f	G4 B- A4. c8~ c
32	Amor, per ti sempre ardo	157	157	6/8		
39	Astio non morì may! Nè puo morir	11	12	3/4	d4 f#2 g8 f f e e d trip{d4	G4 D2 C2. F4 E-2
40	A tanti homini	13	13	2/4	d2~ d4 r8 d c d e f e4 r8	G2 D8 G4 F8 E4
41	Aymè, per tutto	14	15	6/8	c4. d4.~ d4 c8 B- A16 B	F2. BB-4 F8 D C4
42	Bel fiore dança					
43	Bench'amar	158	159	3/4	g4 f8 e4 d8 e4 c r e8 f#	c2. G4 A2 A8 A G
50	Bench'l' serva	16	17	3/4	e4 f#2 g4 r8 a f g a4 a8	A2. G4 F G D F G
56	Checc' a tte piaccia	18	20	9/8	d4. e4 d8 e c d e4. f#2.	b4. c4. B2. A2.~
61	Che pensi di me	21	23	6/8	r4. r8 r c d4. r8 r d e4 g8	r4. r8 r c d4. G4.
62	Che ti çova	24	24	3/4		
72	Chi vole amar	27	28	3/4	a2.~ a2. e4 e g8 f g a g	d2.~ d2. c2. a2. g
78	Come 'nfra l'atre donne	32	33	2/4	c2 e4 f a8 g g d d4 e	c2 A4 a G2 G F E
81	Con dollia me ne vo					
83	Con lagreme sospiro	34	35	2/4	d2 d8 e16 d c8 r r d B- c	G2 G4 A8 B- G E
88	Cosa non è ch'a se tanto	29	31	9/8	c2. d4. e4. d4 e8 c4 d8	C2.~ C4. G2. G4
97	Deduto sey a quel	36	40	3/4	g2. a2 g4 a trip{g4 a8} f4	c2. c4 c2 F2. c4

Capua rules applied to *Non creder, donna* by Francesco da Firenze (Landini)



But does it ***always*** work?



Does “capua” fix aug./dim. unisons, 5ths, & 8ves?

Augmented/Diminished 1/5/8 that “capua” left alone:
269
Augmented/Diminished 1/5/8 that “capua” fixed:
26 (9%)

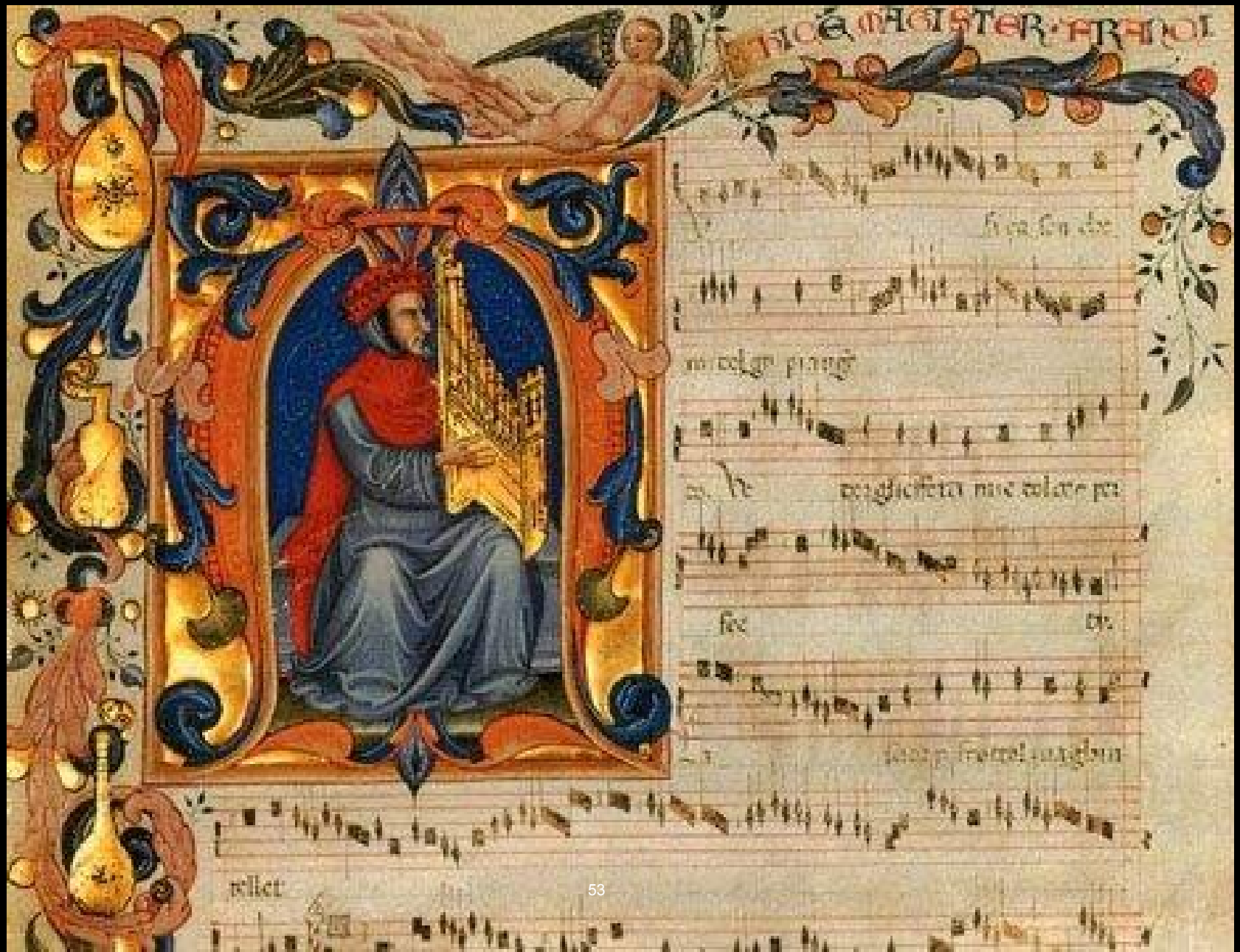
Perfect consonances that “capua” left alone: 11522
Perfect consonances that “capua” made worse: 287 (2.4%)

(Is it better than nothing 3 out of 4 times?
or worse than nothing 9 out of 10 times?)

Reuse your repertory

Exploring tonal closure in ballate

Francesco da Firenze (or 'Landini') *image from the Squarcialupi Codex*



Francesco's Tonality: Summary

Francesco Cadences

Total Same	**64**	**46.4%**
Total Different	**74**	**53.6%**

Other Attributed Composers Cadences

Total Same	**49**	**32.0%**
Total Different	**104**	**68.0%**

Anonymous Cadences

Total Same	**28**	**37.8%**
Total Different	**46**	**62.2%**

Francesco's Tonal usage

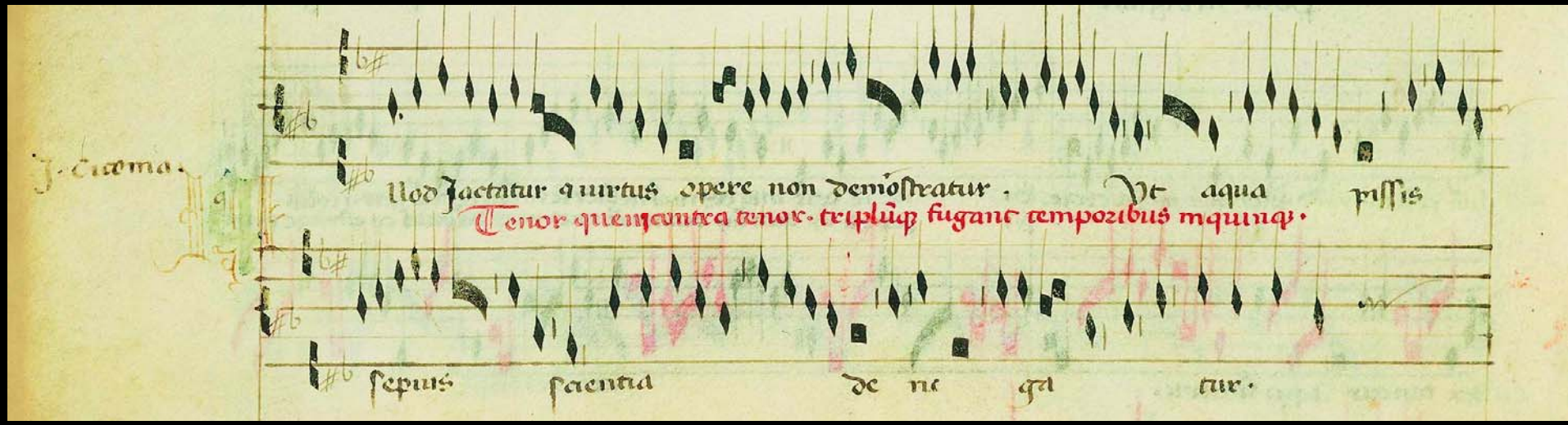
Tenor notes at the beginning and first cadences of ballate

1st note	*Cadence note*	#
A	**same**	**0**
	different	**16**
	total	*16 (12% of total)*
C	**same**	**12**
	different	**7**
	total	*19 (14%)*
D	**same**	**25**
	different	**7**
	total	*32 (23%)*

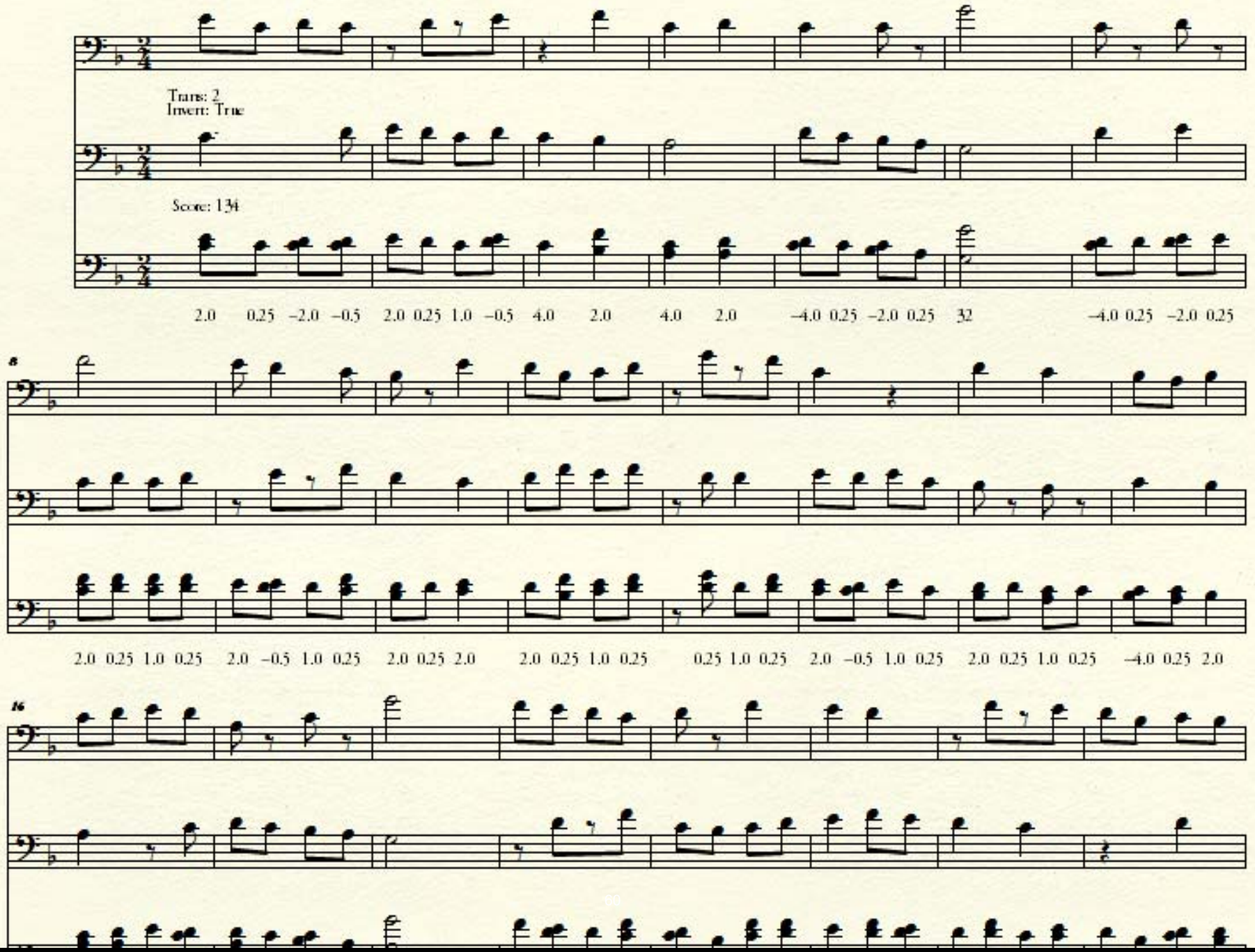
1st note	*Cadence note*	#	
E	**same**	**0**	
	diff.	**2**	
	total	*2*	*(1%)*
F	**same**	**8**	
	diff.	**8**	
	total	*16*	*(12%)*
G	**same**	**19**	
	diff.	**34**	
	total	*53*	*(38%)*

Making a corpus from a single example

A Canon by Johannes Ciconia


J. Ciconia.
Tenor quiescantra tenor. triplū fugant temporibus mquinq;

```
185     return qjPart
186
187 def findRetrogradeVoices(show = True):
188     '''
189     the structure of the piece strongly suggests a retrograde solution
190     (e.g., there is a cadence in m5 and five measures from the end and one
191     at the exact center).  This method tries all transpositions of one
192     voice vs. the other and gives positive points to intervals of 3, 4,
193     5, 6, and 8 (incl. tritones, since they might be fixed w/ other voices;
194     4th is included since there could be a 3rd or 5th below it).
195     '''
196
197     for transpose in [1, 2, -2, 3, -3, 4, -4]:
198         for invert in [False, True]:
199             qj1 = getQJ()
200             qj2 = getQJ()
201             if transpose != 1:
202                 transposeStreamDiatonic(qj2, transpose)
203             if invert is True:
204                 qj2.invertDiatonic(qj2.flat.notes[0], inPlace = True)
205             qj2 = reverse(qj2, makeNotation = False)
206             qj = stream.Score()
207             qj.insert(0, qj2.flat)
208             qj.insert(0, qj1.flat)
209             qjChords = qj.chordify()
210             consScore = 0
211             totIntervals = 0
212             for n in qjChords.flat.notes:
```


Trans: 2
Invert: True
Score: 134
2.0 0.25 −2.0 −0.5 2.0 0.25 1.0 −0.5 4.0 2.0 4.0 2.0 −4.0 0.25 −2.0 0.25 32 −4.0 0.25 −2.0 0.25
8
2.0 0.25 1.0 0.25 2.0 −0.5 1.0 0.25 2.0 0.25 2.0 2.0 0.25 1.0 0.25 0.25 1.0 0.25 2.0 −0.5 1.0 0.25 2.0 0.25 1.0 0.25 −4.0 0.25 2.0
16

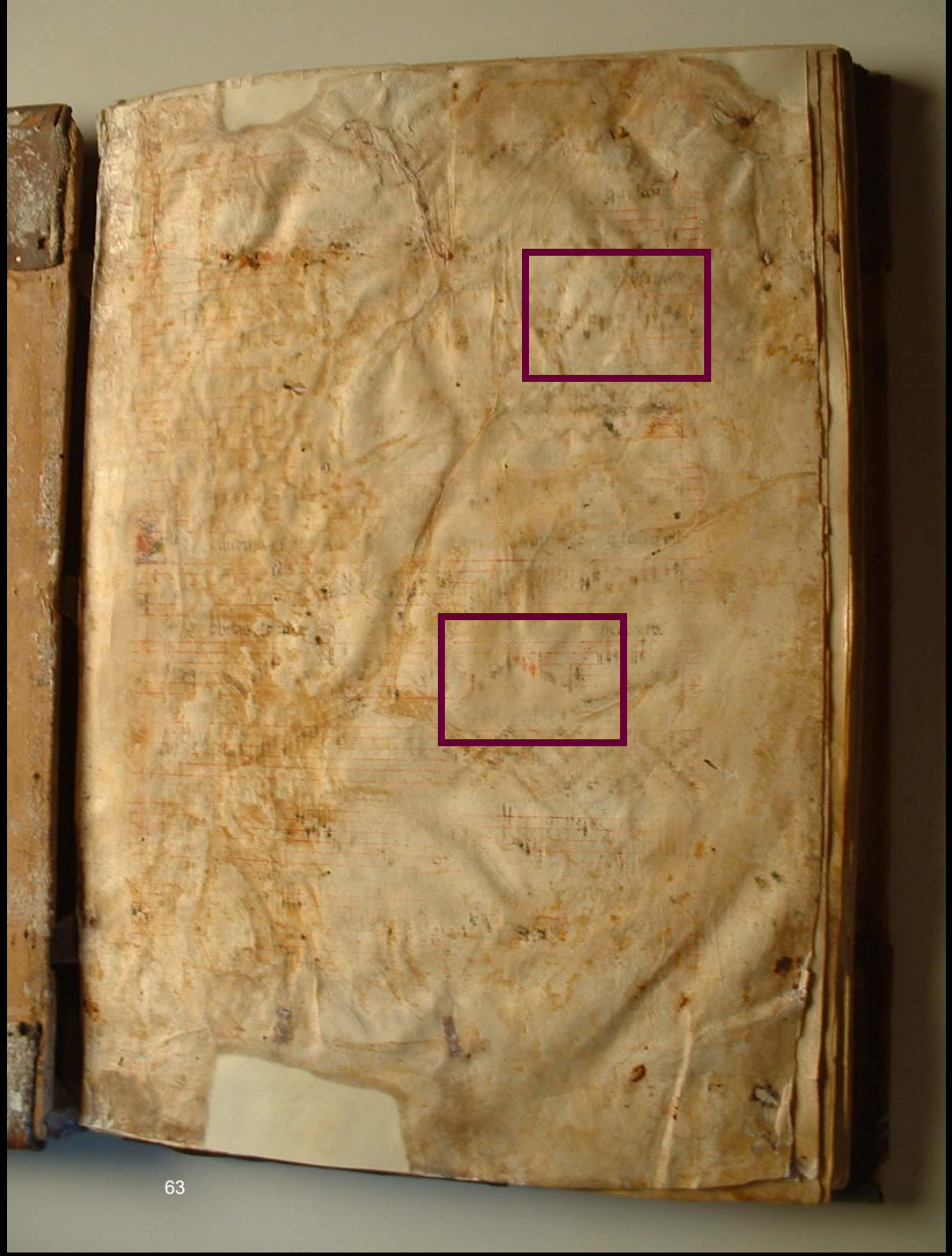
Simultaneous retrograde solutions for Quod Jactatur

INTERVAL	INVERT?	SCORE
1		113
1	Invert	-150
2		-111
2	Invert	134
-2		-119
-2	Invert	90
3		121
3	Invert	-110
-3		103
-3	Invert	92
4		112
4	Invert	123
-4		123
-4	Invert	125

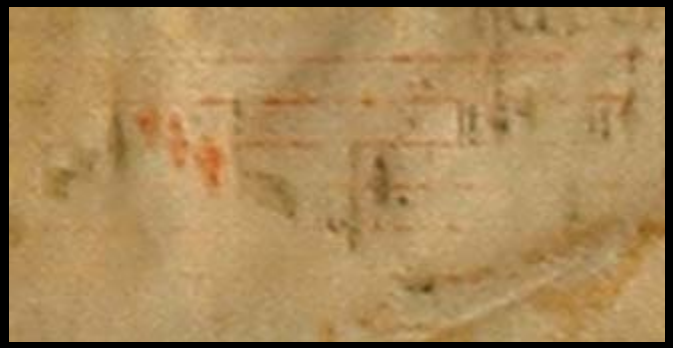
Short case studies

3. Searching in Medieval and Renaissance Music

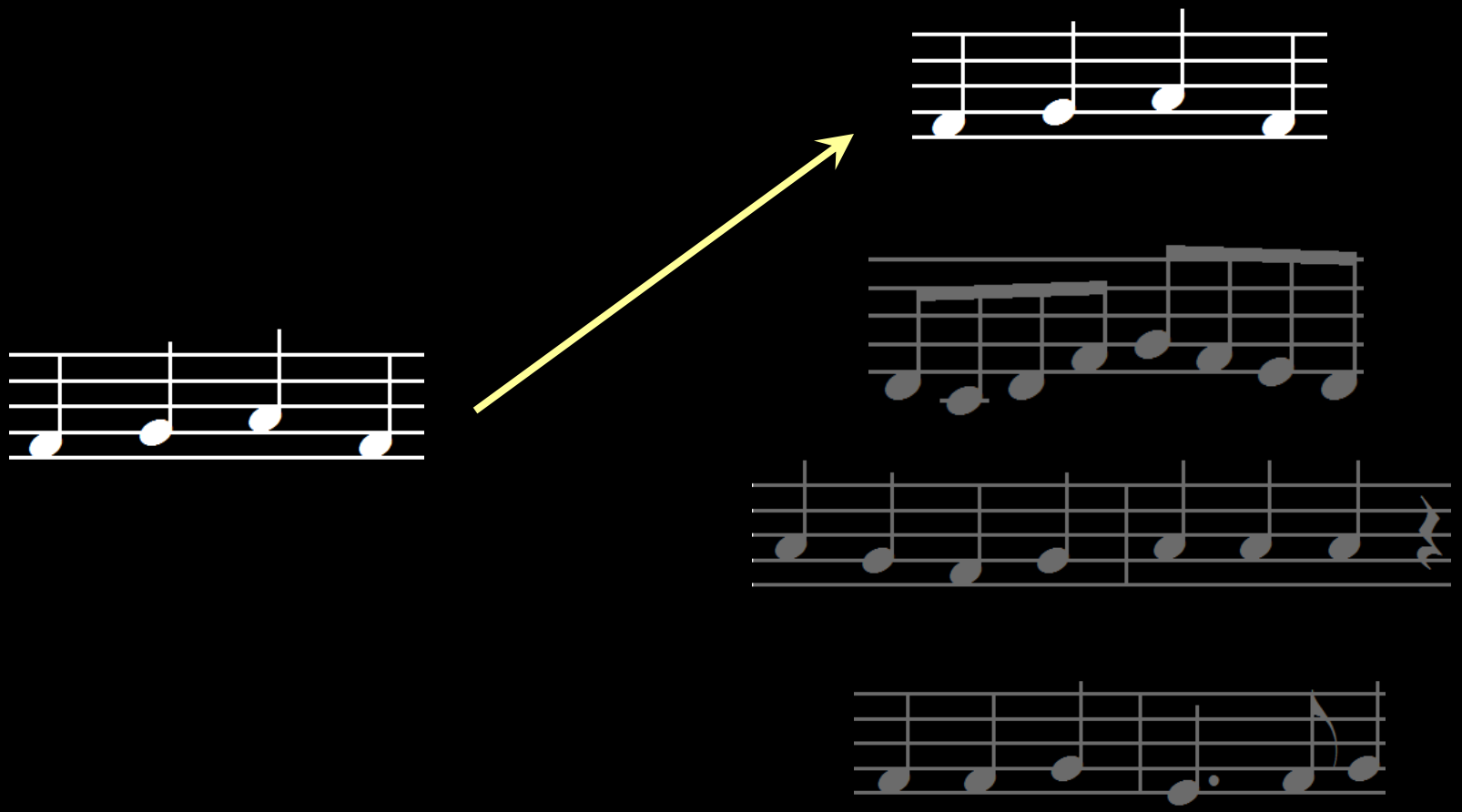
Cividale 98, folio 1r



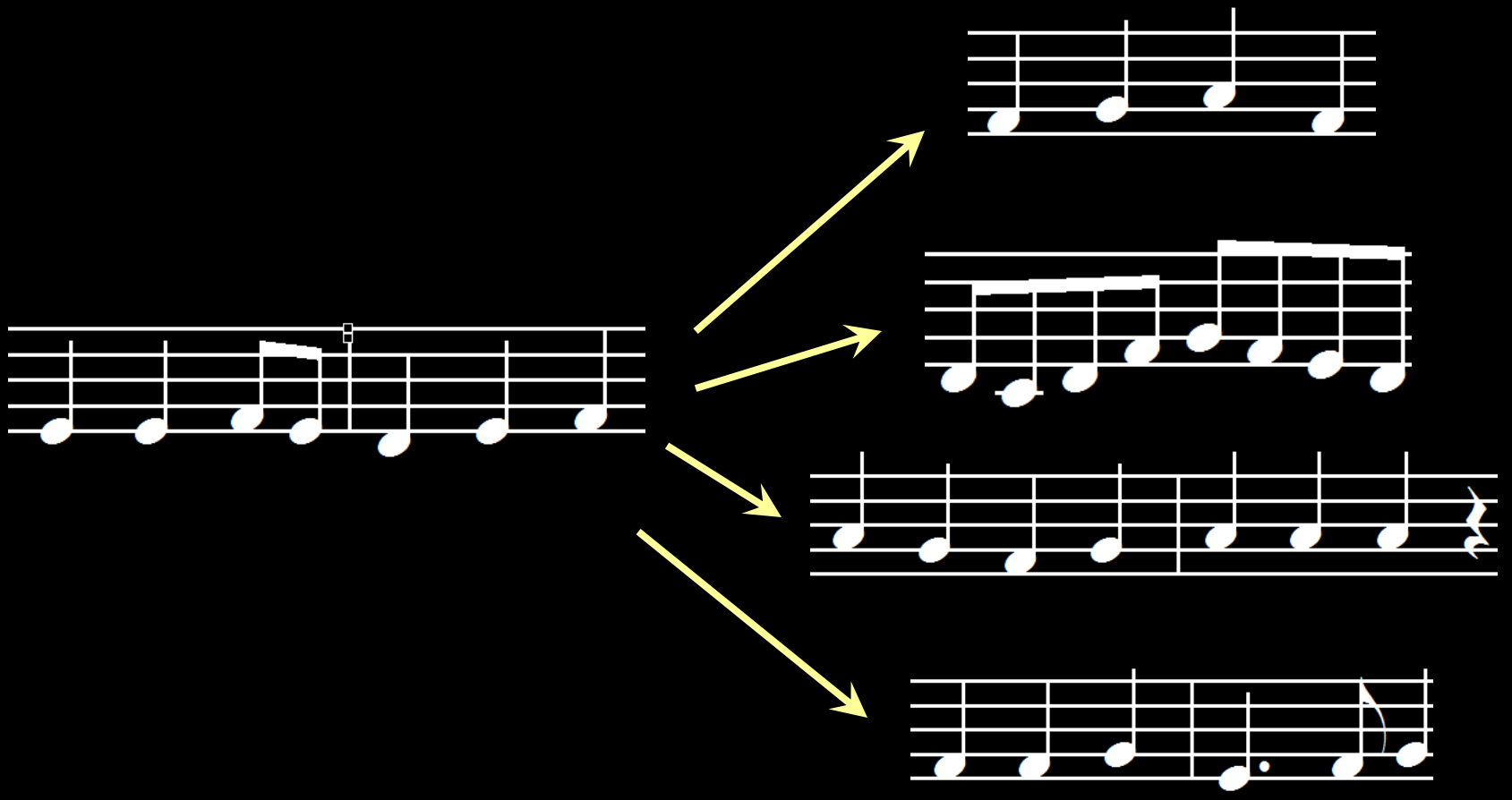
Cividale 98, digitally enhanced



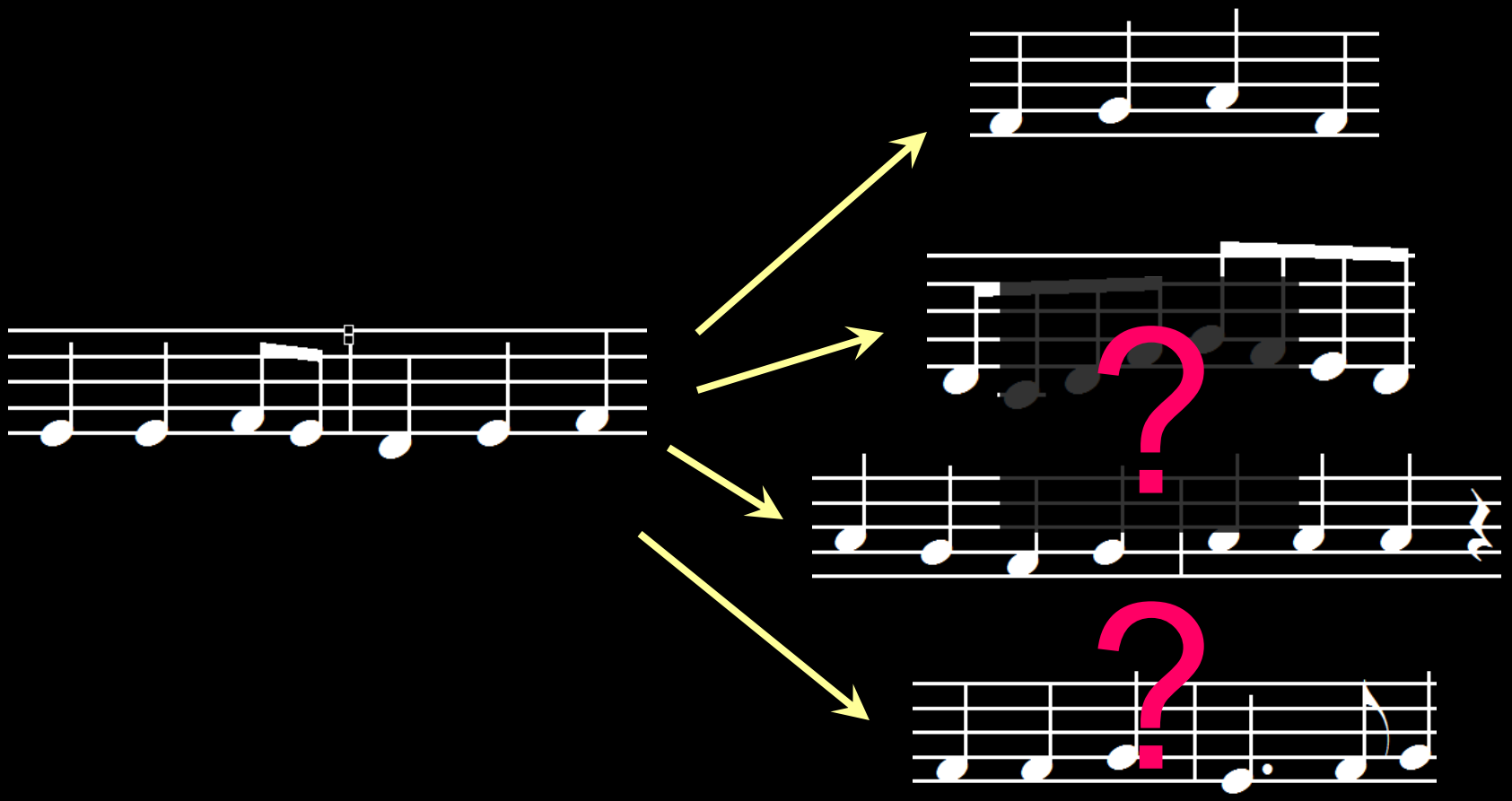
Exact matches are no problem for current databases



But even the smallest musical changes pose problems

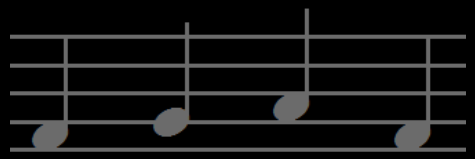


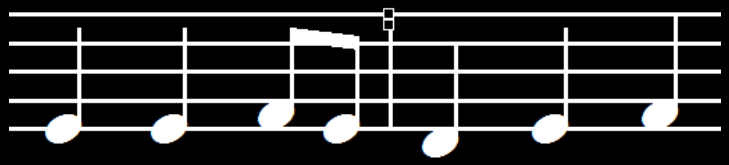
But even the smallest musical changes pose problems



`music21` encodes information that solves these search problems.

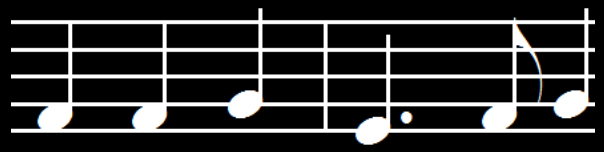
sometimes melodies are higher or lower than in the database





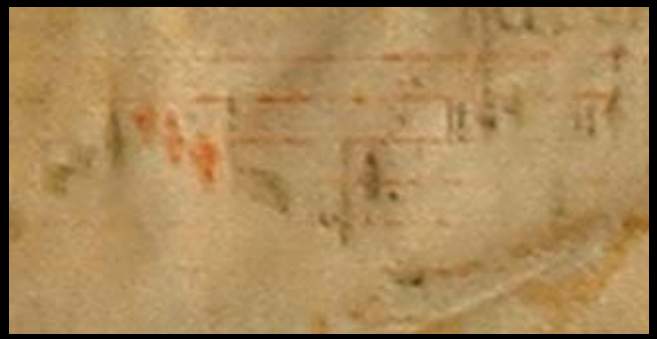
rhythms change slightly

“skips” get filled in



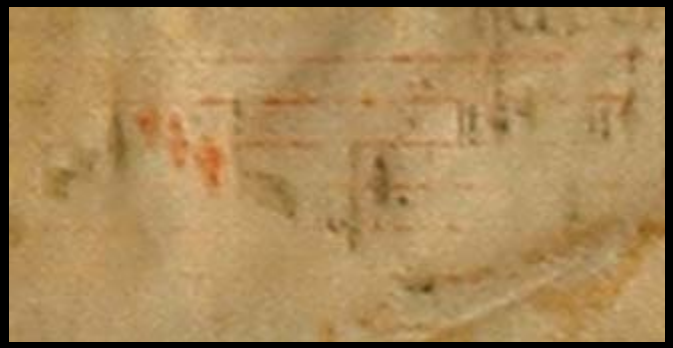
Code: Music21 for searching on Cividale 98 piece

```
score.genre == “Ballade” or score.genre in
    medRen.MassMovements
score.manuscript.notation[“red”] is not False
score.timeSignature in[“6/8”, “9/8”]
for thisPart in score:
    thisPart.notes.containsIntervals(-2, -2, -2)
    search.notes.thisPart(“r8 r8 ?8 ?4. * r8 r8 ?4.”) is not None
```

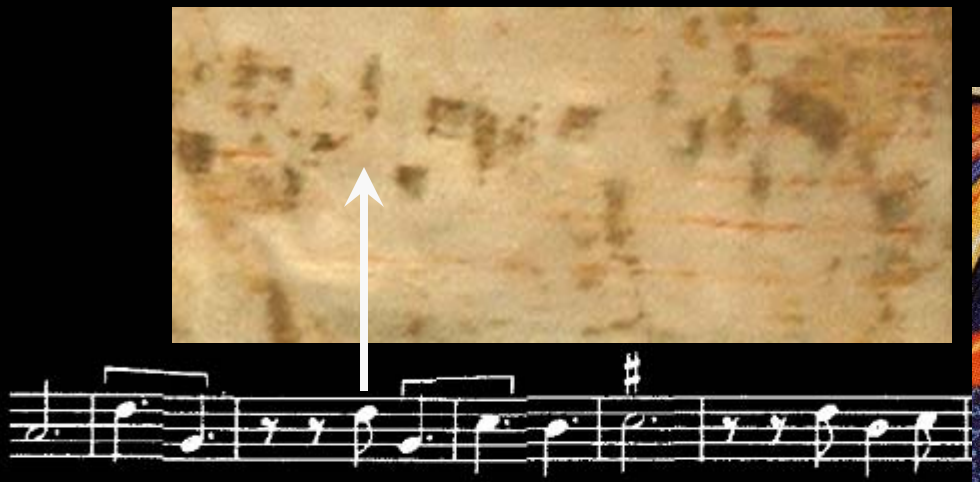


This search has been simplified slightly for rapid presentation.

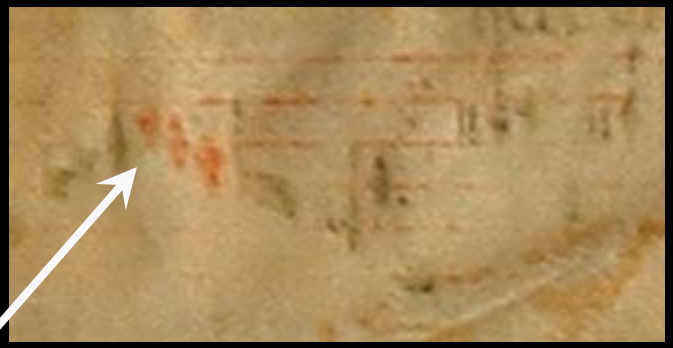
Cividale 98, digitally enhanced



Cividale 98, digitally enhanced and identified as a Credo by Zachara da Teramo



T: mm. 321-329 (+*custos of 330*)



Ct: mm. 316-323

Examples and Demonstrations — music21 v0.3.1a6 documentation - Mozilla Firefox

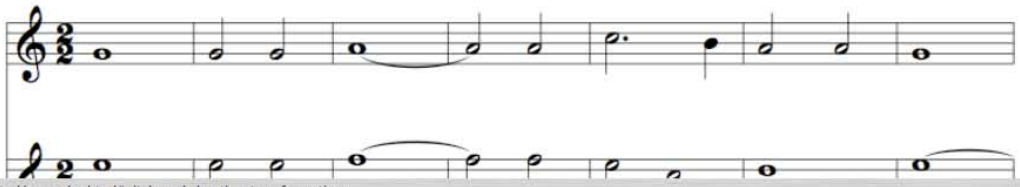
http://mit.edu/music21/doc/html/examples.html#creating-a-reduction-and-labeling-intervals

Michael Scott Cuthbert and cuthber... | Google Desktop: Re: oh right... | Gmail - Search results - cuthbert@g... | Examples and Demonstrations — ...

Creating a Reduction and Labeling Intervals

This example, after parsing a polyphonic work stored as an Opus and creating a Score, presents and labels the intervals of the resultant chords of all distinct harmonies:

```
from music21 import corpus
# Parse an Opus, a collection of Scores
o = corpus.parseWork('josquin/laDeplorationDeLaMorteDeJohannesOckeghem')
# Create a Score from a Measure range
sExcerpt = o.mergeScores().measures(127, 134)
# Create a reduction of Chords
reduction = sExcerpt.chordify()
# Iterate over the Chords and prepare presentation
for c in reduction.flat.getElementsByClass('Chord'):
    c.closedPosition(forceOctave=4, inPlace=True)
    c.removeRedundantPitches(inPlace=True)
    c.annotateIntervals()
# Add the reduction and display the results
sExcerpt.insert(0, reduction)
sExcerpt.show()
```



Previous topic

Overview: Meters, Time Signatures, and Processing Beams, Accents, and Beats

Next topic

Installing Music21

This Page

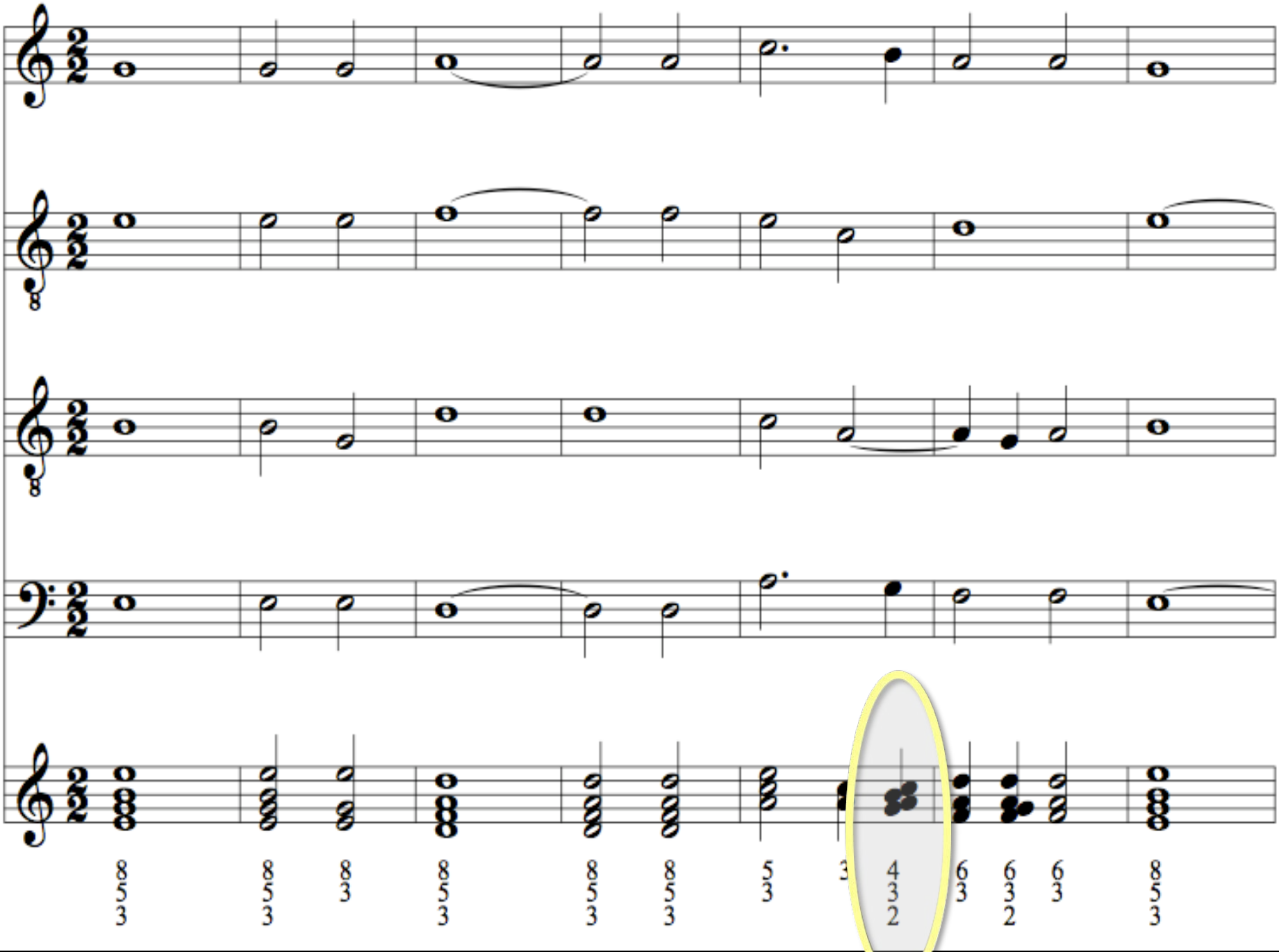
Show Source

Quick search

http://mit.edu/music21/doc/html/examples.html#pitch-and-duration-transformations

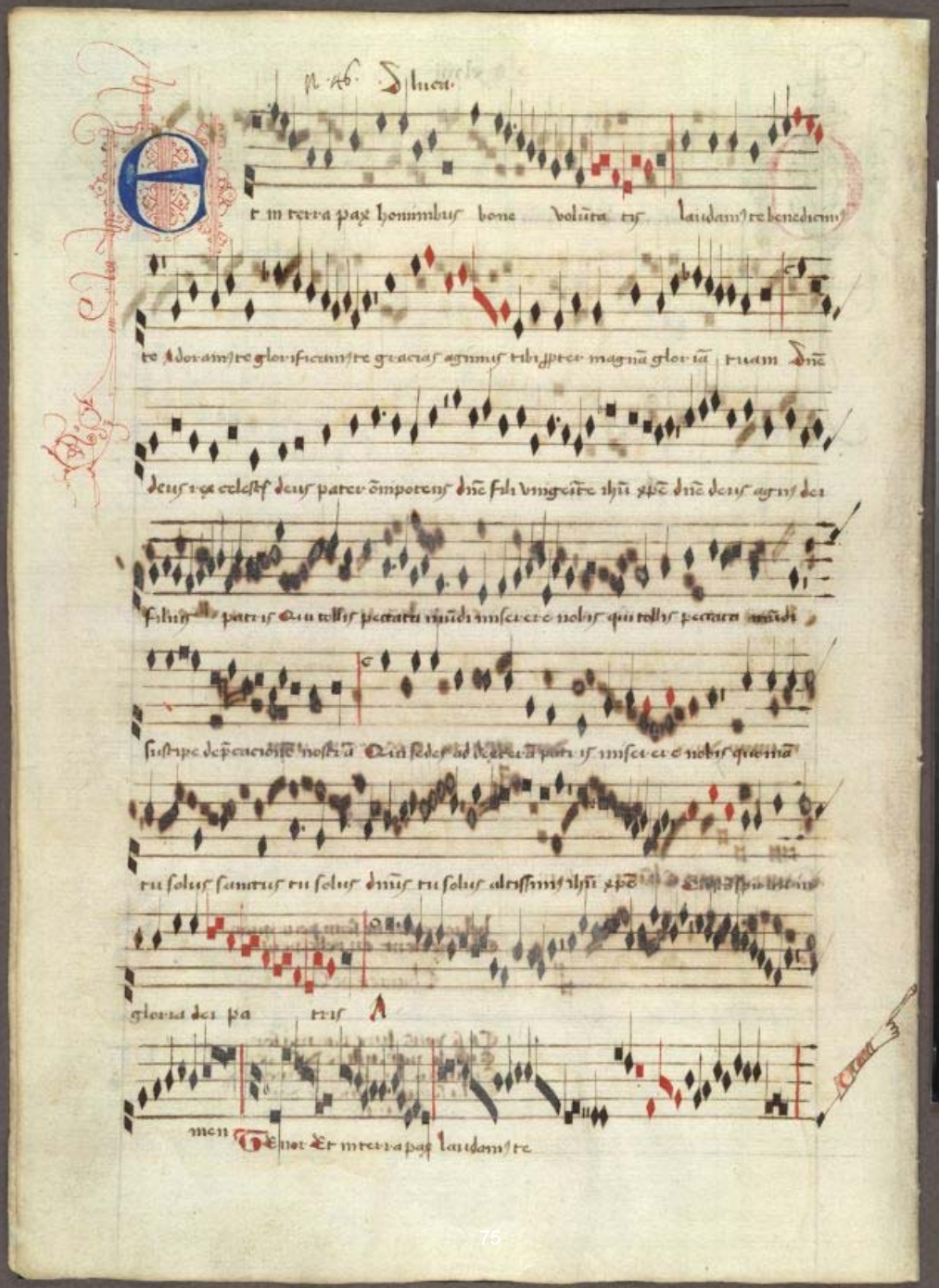
Slide 75 of 79 | "_cuthbert1" | English (United States)

quodJactatur | ModA | harvard_talk.pptx | saved-messages... | Pydev - music2... | Finale 2010b | ModA_20v.jpg -... | Examples and D... | 04_secular-intab... | 16:17

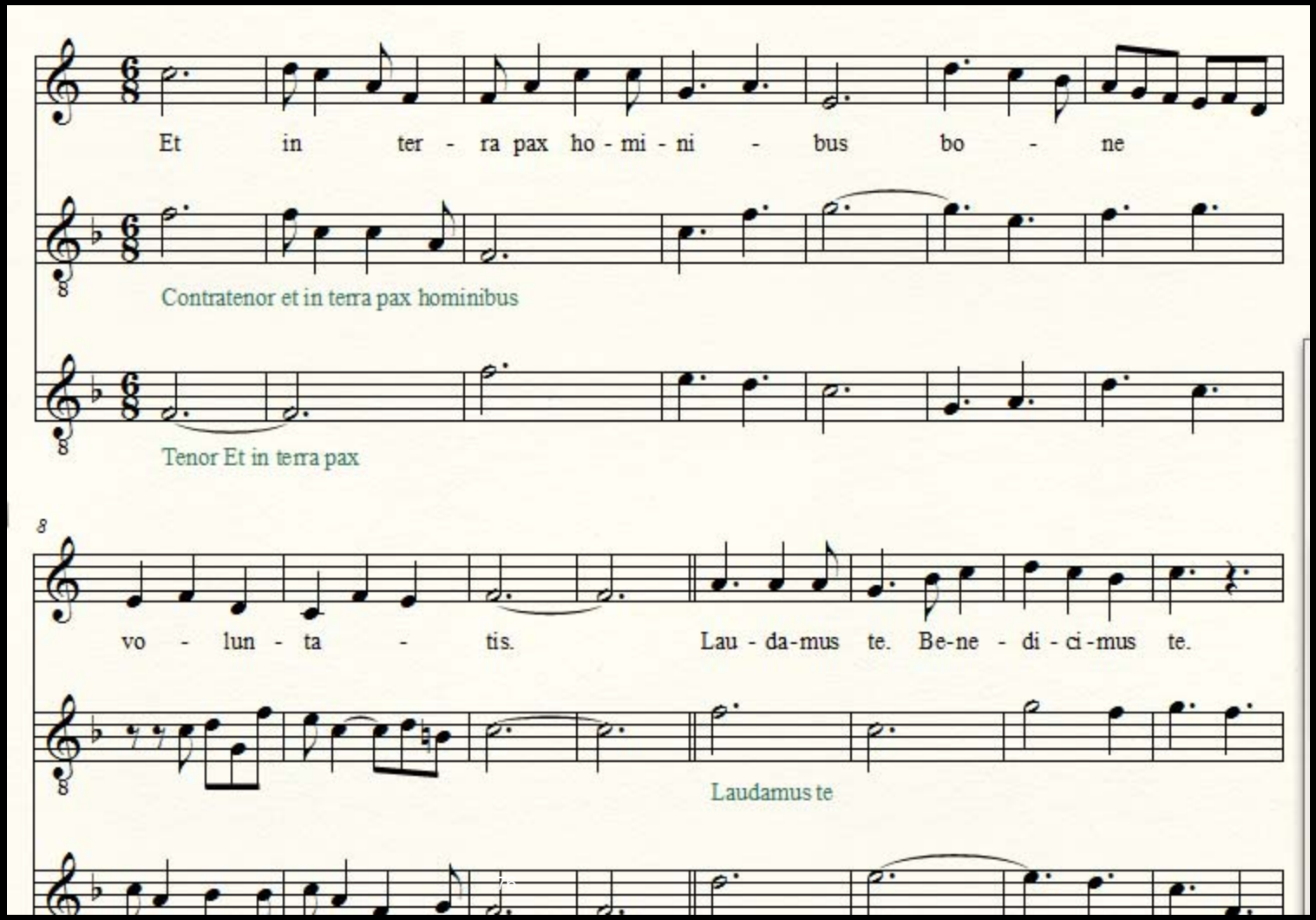

8 5 3
8 5 3
8 3
8 5 3
8 5 3
8 5 3
5 3
3
4 3 2
6 3
6 3 2
6 3
8 5 3

Short case studies

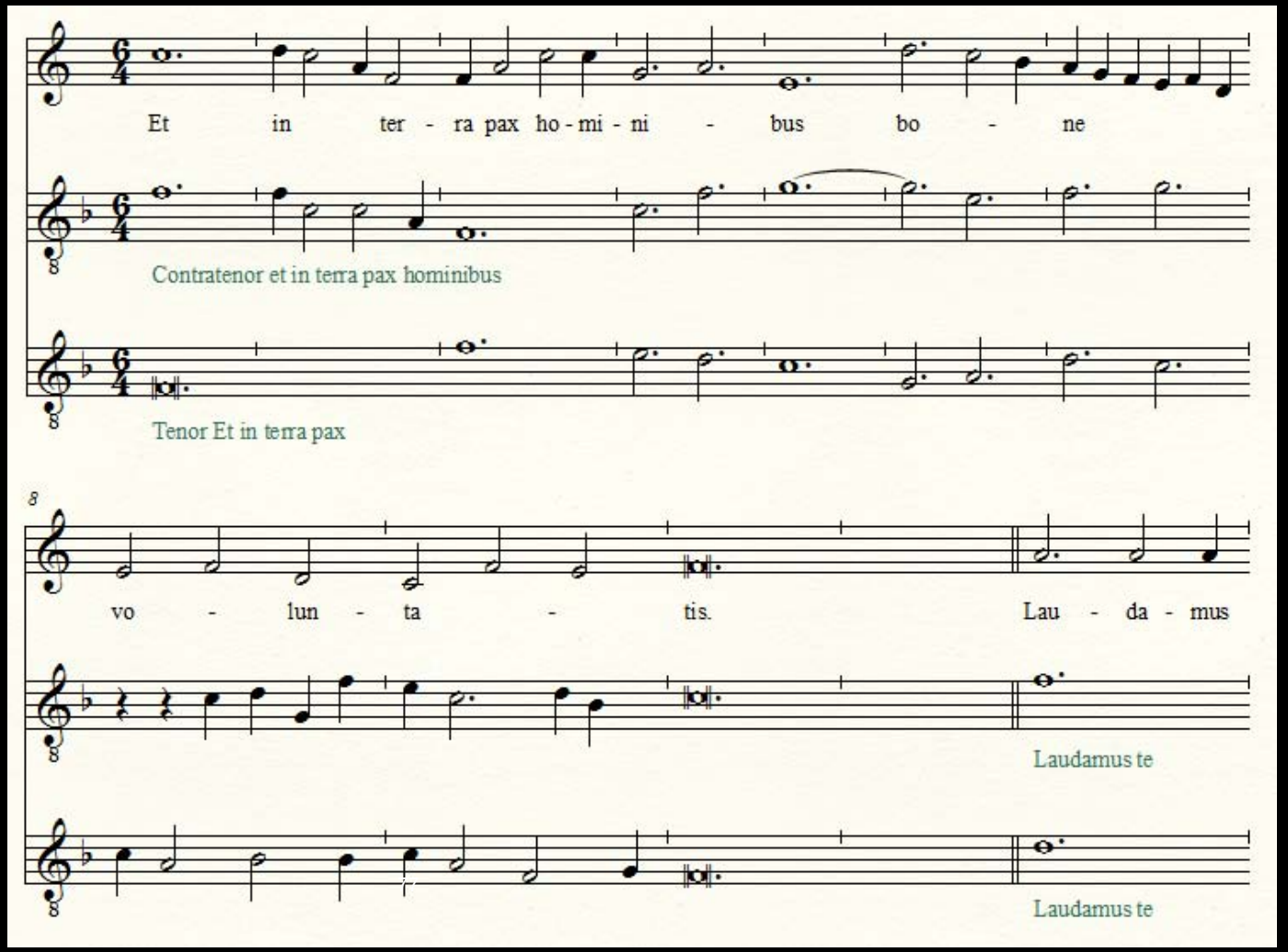
4. Producing digital and print editions in different styles



```
from music21 import *
gloria = corpus.parse('luca/gloria')
gloria.show()
```



```
from music21 import *
gloria = corpus.parse('luca/gloria')
gloriaNew = medren.convertHouseStyle(gloria, durationScale = 2,
    barlineStyle = 'tick', tieTransfer = True)
gloriaNew.show()
```



Counting our Losses

Two questions about Medieval music:

(1) How many manuscripts of medieval polyphony were once copied but no longer exist?

(2) How many pieces of medieval polyphony were once copied but no longer exist?

Counting our Losses

What percentage of pieces of 14th c. Italian polyphony were once copied but no longer exist?

(a) 90% of them?
(b) 99% of them?
(c) 99.9% of them?
(d) 99.9999% of them?

How could we figure out the answer?

N	x	P(m)	=	m
number of pieces originally		probability that any given piece is missing		total # missing pieces

How can we find P(m)?
[probability that we're missing a given piece]

What's the probability that any given piece is in a given manuscript?

$$\frac{r_i}{N}$$

r_i = number of pieces in that manuscript

N = total number of pieces originally copied.

[a number we don't know…]

* unquestioned assumption: pieces have equal probability of being copied.

How can we find P(m)?
[probability that we're missing a given piece]

What's the probability that any given piece is NOT in a given manuscript?

$$1 - \frac{r_i}{N}$$

* unquestioned assumption still in play.

How can we find P(m)?
[probability that we're missing a given piece]

What's the probability that any given piece is NOT in TWO given manuscripts?

$$\left(1 - \frac{r_1}{N}\right)\left(1 - \frac{r_2}{N}\right)$$

* unquestioned assumption still in play.

How can we find P(m)?
[probability that we're missing a given piece]

What's the probability that any given piece is NOT in ANY of all 85 surviving manuscripts?

$$\left(1-\frac{r_1}{N}\right)\left(1-\frac{r_2}{N}\right)\left(1-\frac{r_3}{N}\right)\left(1-\frac{r_4}{N}\right)\left(1-\frac{r_5}{N}\right)\left(1-\frac{r_6}{N}\right)\left(1-\frac{r_7}{N}\right)\left(1-\frac{r_8}{N}\right)\left(1-\frac{r_9}{N}\right)\left(1-\frac{r_{10}}{N}\right)\left(1-\frac{r_{11}}{N}\right)\left(1-\frac{r_{12}}{N}\right)\left(1-\frac{r_{13}}{N}\right)\cdots$$
$$\left(1-\frac{r_{14}}{N}\right)\left(1-\frac{r_{15}}{N}\right)\left(1-\frac{r_{16}}{N}\right)\left(1-\frac{r_{17}}{N}\right)\left(1-\frac{r_{18}}{N}\right)\left(1-\frac{r_{19}}{N}\right)\left(1-\frac{r_{20}}{N}\right)\left(1-\frac{r_{21}}{N}\right)\left(1-\frac{r_{22}}{N}\right)\left(1-\frac{r_{23}}{N}\right)\left(1-\frac{r_{24}}{N}\right)\cdots$$
$$\left(1-\frac{r_{25}}{N}\right)\left(1-\frac{r_{26}}{N}\right)\left(1-\frac{r_{27}}{N}\right)\left(1-\frac{r_{28}}{N}\right)\left(1-\frac{r_{29}}{N}\right)\left(1-\frac{r_{30}}{N}\right)\left(1-\frac{r_{31}}{N}\right)\left(1-\frac{r_{32}}{N}\right)\left(1-\frac{r_{33}}{N}\right)\left(1-\frac{r_{34}}{N}\right)\left(1-\frac{r_{35}}{N}\right)\cdots$$
$$\left(1-\frac{r_{36}}{N}\right)\left(1-\frac{r_{37}}{N}\right)\left(1-\frac{r_{38}}{N}\right)\left(1-\frac{r_{39}}{N}\right)\left(1-\frac{r_{40}}{N}\right)\left(1-\frac{r_{41}}{N}\right)\left(1-\frac{r_{42}}{N}\right)\left(1-\frac{r_{43}}{N}\right)\left(1-\frac{r_{44}}{N}\right)\left(1-\frac{r_{45}}{N}\right)\left(1-\frac{r_{46}}{N}\right)\cdots$$
$$\left(1-\frac{r_{47}}{N}\right)\left(1-\frac{r_{48}}{N}\right)\left(1-\frac{r_{49}}{N}\right)\left(1-\frac{r_{50}}{N}\right)\left(1-\frac{r_{51}}{N}\right)\left(1-\frac{r_{52}}{N}\right)\left(1-\frac{r_{53}}{N}\right)\left(1-\frac{r_{54}}{N}\right)\left(1-\frac{r_{55}}{N}\right)\left(1-\frac{r_{56}}{N}\right)\left(1-\frac{r_{57}}{N}\right)\cdots$$
$$\left(1-\frac{r_{58}}{N}\right)\left(1-\frac{r_{59}}{N}\right)\left(1-\frac{r_{60}}{N}\right)\left(1-\frac{r_{61}}{N}\right)\left(1-\frac{r_{62}}{N}\right)\left(1-\frac{r_{63}}{N}\right)\left(1-\frac{r_{64}}{N}\right)\left(1-\frac{r_{65}}{N}\right)\left(1-\frac{r_{66}}{N}\right)\left(1-\frac{r_{67}}{N}\right)\left(1-\frac{r_{68}}{N}\right)\cdots$$
$$\left(1-\frac{r_{69}}{N}\right)\left(1-\frac{r_{70}}{N}\right)\left(1-\frac{r_{71}}{N}\right)\left(1-\frac{r_{72}}{N}\right)\left(1-\frac{r_{73}}{N}\right)\left(1-\frac{r_{74}}{N}\right)\left(1-\frac{r_{75}}{N}\right)\left(1-\frac{r_{76}}{N}\right)\left(1-\frac{r_{77}}{N}\right)\left(1-\frac{r_{78}}{N}\right)\left(1-\frac{r_{79}}{N}\right)\cdots$$
$$\left(1-\frac{r_{80}}{N}\right)\left(1-\frac{r_{81}}{N}\right)\left(1-\frac{r_{82}}{N}\right)\left(1-\frac{r_{83}}{N}\right)\left(1-\frac{r_{84}}{N}\right)\left(1-\frac{r_{85}}{N}\right)=P(m)$$

* unquestioned assumption still in play.

How could we figure out the answer?

N x	P(m)	= m
number of pieces originally	probability that any given piece is missing	total # missing pieces

* unquestioned assumption still in play.

How could we figure out the answer?

$$N \times \left(1-\frac{r_1}{N}\right)\left(1-\frac{r_2}{N}\right)\left(1-\frac{r_3}{N}\right)\left(1-\frac{r_4}{N}\right)\left(1-\frac{r_5}{N}\right)\left(1-\frac{r_6}{N}\right)\left(1-\frac{r_7}{N}\right)\left(1-\frac{r_8}{N}\right)\left(1-\frac{r_9}{N}\right)\left(1-\frac{r_{10}}{N}\right)\left(1-\frac{r_{11}}{N}\right)\left(1-\frac{r_{12}}{N}\right)\left(1-\frac{r_{13}}{N}\right)\cdots$$
$$\left(1-\frac{r_{14}}{N}\right)\left(1-\frac{r_{15}}{N}\right)\left(1-\frac{r_{16}}{N}\right)\left(1-\frac{r_{17}}{N}\right)\left(1-\frac{r_{18}}{N}\right)\left(1-\frac{r_{19}}{N}\right)\left(1-\frac{r_{20}}{N}\right)\left(1-\frac{r_{21}}{N}\right)\left(1-\frac{r_{22}}{N}\right)\left(1-\frac{r_{23}}{N}\right)\left(1-\frac{r_{24}}{N}\right)\cdots$$
$$\left(1-\frac{r_{25}}{N}\right)\left(1-\frac{r_{26}}{N}\right)\left(1-\frac{r_{27}}{N}\right)\left(1-\frac{r_{28}}{N}\right)\left(1-\frac{r_{29}}{N}\right)\left(1-\frac{r_{30}}{N}\right)\left(1-\frac{r_{31}}{N}\right)\left(1-\frac{r_{32}}{N}\right)\left(1-\frac{r_{33}}{N}\right)\left(1-\frac{r_{34}}{N}\right)\left(1-\frac{r_{35}}{N}\right)\cdots$$
$$\left(1-\frac{r_{36}}{N}\right)\left(1-\frac{r_{37}}{N}\right)\left(1-\frac{r_{38}}{N}\right)\left(1-\frac{r_{39}}{N}\right)\left(1-\frac{r_{40}}{N}\right)\left(1-\frac{r_{41}}{N}\right)\left(1-\frac{r_{42}}{N}\right)\left(1-\frac{r_{43}}{N}\right)\left(1-\frac{r_{44}}{N}\right)\left(1-\frac{r_{45}}{N}\right)\left(1-\frac{r_{46}}{N}\right)\cdots$$
$$\left(1-\frac{r_{47}}{N}\right)\left(1-\frac{r_{48}}{N}\right)\left(1-\frac{r_{49}}{N}\right)\left(1-\frac{r_{50}}{N}\right)\left(1-\frac{r_{51}}{N}\right)\left(1-\frac{r_{52}}{N}\right)\left(1-\frac{r_{53}}{N}\right)\left(1-\frac{r_{54}}{N}\right)\left(1-\frac{r_{55}}{N}\right)\left(1-\frac{r_{56}}{N}\right)\left(1-\frac{r_{57}}{N}\right)\cdots$$
$$\left(1-\frac{r_{58}}{N}\right)\left(1-\frac{r_{59}}{N}\right)\left(1-\frac{r_{60}}{N}\right)\left(1-\frac{r_{61}}{N}\right)\left(1-\frac{r_{62}}{N}\right)\left(1-\frac{r_{63}}{N}\right)\left(1-\frac{r_{64}}{N}\right)\left(1-\frac{r_{65}}{N}\right)\left(1-\frac{r_{66}}{N}\right)\left(1-\frac{r_{67}}{N}\right)\left(1-\frac{r_{68}}{N}\right)\cdots$$
$$\left(1-\frac{r_{69}}{N}\right)\left(1-\frac{r_{70}}{N}\right)\left(1-\frac{r_{71}}{N}\right)\left(1-\frac{r_{72}}{N}\right)\left(1-\frac{r_{73}}{N}\right)\left(1-\frac{r_{74}}{N}\right)\left(1-\frac{r_{75}}{N}\right)\left(1-\frac{r_{76}}{N}\right)\left(1-\frac{r_{77}}{N}\right)\left(1-\frac{r_{78}}{N}\right)\left(1-\frac{r_{79}}{N}\right)\cdots$$
$$\left(1-\frac{r_{80}}{N}\right)\left(1-\frac{r_{81}}{N}\right)\left(1-\frac{r_{82}}{N}\right)\left(1-\frac{r_{83}}{N}\right)\left(1-\frac{r_{84}}{N}\right)\left(1-\frac{r_{85}}{N}\right) = m$$

number of pieces originally

total # missing pieces

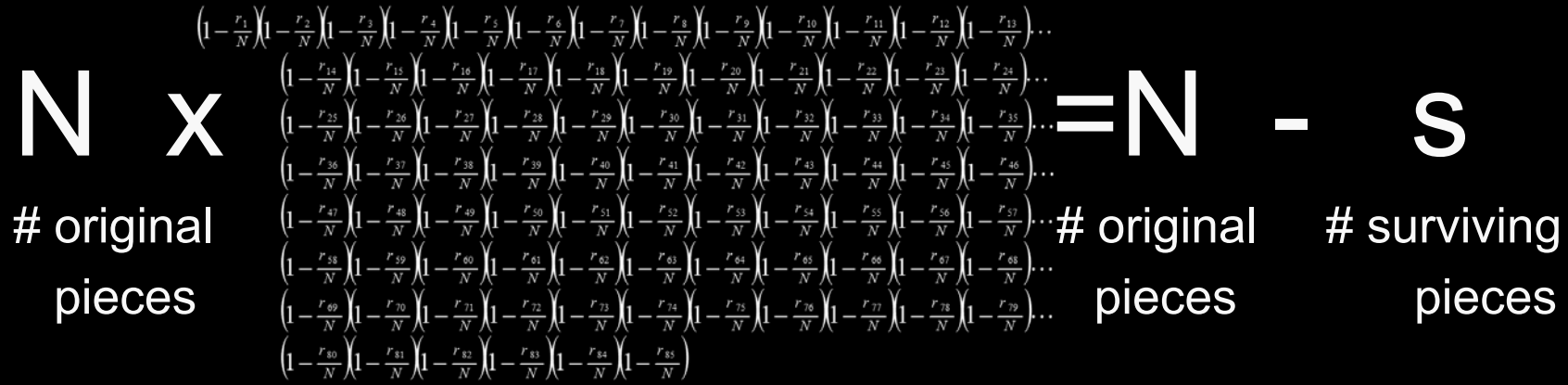
* unquestioned assumption still in play.

What's the number of missing pieces?

m	=	N	-	s
# of missing pieces		# of pieces originally		# of pieces that survive

* unquestioned assumption still in play.

How could we figure out the answer?



* unquestioned assumption still in play.

```
#!/usr/local/bin/perl

##### find_n.pl -- Michael Scott Cuthbert
### Find hypothetical total number of pieces given X1 pieces randomly
### distributed in manuscripts of size N1 N2 N3 N4...

###    ./find_n.pl X1 N1 N2 N3 N4 ...

use strict;
use Math::BigFloat;

my $pieces_surviving_today = shift @ARGV;
my @ms_sizes   = @ARGV;
my $total_number_of_mss = scalar @ms_sizes;

# n * (1/n^(num_of_mss)) * (n - a1) * (n - a2) * ... * (n-ay) = n -
pieces_surviving_today (r)

    ### n = our current guess for the number of original pieces; start by
    ###     supposing we have them all (plus 1 to avoid division by zero).
my $n = $pieces_surviving_today + 1;
```

* unquestioned assumption still in play.

Testing whether our model of “random copying” messes up the number for N.

Holdout cross-validation:

Look at a subset of the data (say all manuscripts discovered before 1955) and see how many pieces we would expect to have today given the new manuscripts discovered since then. And then we compare that number to what we actually have.

Result:

Pretty good! Estimates are about 4% too low.

Original numbers for pieces in MSS 1380-1420

	# today	estimated original total	% missing
Ballate	409	507	19 %
Madrigals	167	177	6 %
Liturgical Pieces	116	196	41 %
Non-Liturgical Latin works	47	105	55 %
Total	739	985	25 %

Counting our Losses

How many pieces of 14th c. Italian polyphony were once copied but no longer exist?

(a) 90% of them?
(b) 99% of them?
(c) 99.9% of them?
(d) 99.9999% of them?
(e) 25% of them!

21M.269 Studies in Western Music History: Quantitative and Computational Approaches to Music History
Spring 2012