COMPUTERGESTÜTZTE MUSIKFORSCHUNG 1

Institut für Musikinformatik und Musikwissenschaft
Wintersemester 2025–26

Christophe Weis
christophe.weis@stud.hfm.eu

Woche 04
04.11.2025

Organisation

wöchentlich, Di. 14.30–16.00, K10 Raum 309

Modul **Music Processing**

- **BA MI (HF)/MW (EF), wiss. Schwerpunkt:** Pflicht (4. Semester)
- **BA MI (HF)/MW (EF), künstl. Schwerpunkt:** Wahlpflicht (6. Semester)
- **BA MW (HF)/MI (EF):** Pflicht (4. Semester) – reduzierter Arbeitsaufwand
- **BA MI/MW (KF):** Pflicht (4. Semester)
- **BA:** Wahlfach

Projektarbeit

- eine selbstständige praktische Arbeit aus den Bereichen Musikkodierung, symbolbasierte Musikverarbeitung und –analyse mit Dokumentation (ca. 5000 Zeichen)

Übungen

- Tutorin: Joanna Friedrich-Sroka
- wöchentlich, Di. 11.15–12.45, K10 Raum 309

09.

music21– Chords & Keys

Chord-Objekte

- Akkorde als Chord-Objekte:

```
c_major_chord1 = m21.chord.Chord(["C3", "G3", "E4", "E5"])
c_major_chord2 = m21.chord.Chord(["E4", "G4", "C5"])
```

- Attribute und Methoden zur Akkord-Analyse:

```
c_major_chord1.commonName → 'major triad'
c_major_chord2.commonName → 'major triad'
c_major_chord1.pitchedCommonName → 'C-major triad'
c_major_chord2.pitchedCommonName → 'C-major triad'

c_major_chord1.isMinorTriad → False

c_major_chord1.root() → <music21.pitch.Pitch C3>
c_major_chord2.root() → <music21.pitch.Pitch C5>
c_major_chord1.bass() → <music21.pitch.Pitch C3>
c_major_chord2.bass() → <music21.pitch.Pitch E4>
c_major_chord1.pitches → …

c_major_chord1.inversion() → 0
c_major_chord2.inversion() → 1
```

Pitch-Class-Sets

- Erstellen von Chord-Objekten mithilfe von **Pitch-Class-Sets**:

```
c_major_chord3 = m21.chord.Chord([0, 4, 7])

c_major_chord3.pitchCommonName → 'C-major triad'
c_major_chord3.bass() → <music21.pitch.Pitch C>
```

- Verwendung von **Forte-Numbers** als Werkzeug zur Unterscheidung/Analyse von Akkord-Typen mit music21:

```
my_chord1 = m21.chord.Chord(["C4", "D4", "E4", "F4", "F#4"])
my_chord2 = m21.chord.Chord(["C4", "D4", "E-4", "F4", "F#4"])

my_chord1.forteClass → '5-9B'
my_chord2.forteClass → '5-10B'
```

Chordify

Die Methode **.chordify()** reduziert den musikalischen Inhalt eines Score-Streams (bestehend aus einzelnen Part-Streams) zu einer Folge von Akkorden, die als ein Part-Stream ausgegeben werden.
Dabei löst jedes neue Notenereignis eine neue Darstellung aller gleichzeitig aktiven Noten als Akkord aus.

Beispiel:

- `chorale = m21.converter.parse('chorale.mid')`
  `print(chorale)` → `<music21.stream.Score 0x1c42e600a50>`

- `chorale_chordified = chorale.chordify()`
  `print(chorale_chordified)` → `<music21.stream.Part 0x1c42e4cc210`

- `chorale_chords = chorale_chordified.recurse().getElementsByClass(m21.chord.Chord)`
  oder:
  `chorale_chords = chorale_chordified.flatten().getElementsByClass(m21.chord.Chord)`

  `chorale_chords[0]` → `<music21.chord.Chord B1 B2 D4 F#4>`

  ```
  for chord in chorale_chords:
      print chord.isMajorChord()
  ```
  → …

music21.key.Key

- Tonarten können in music21 über die Klasse *music21.key.Key* definiert.
  Die Implementierung von **Dur-Tonarten** erfolgt mithilfe von **Großbuchstaben**, von **Moll-Tonarten** mithilfe von **Kleinbuchstaben**.

- *Beispiele:*

  - `c_major_key = m21.key.Key('C')`

    ```
    c_major_key → <music21.key.Key of C major>
    c_major_key.mode → 'major'
    c_major_key.tonic → <music21.pitch.Pitch C>
    ```

  - `d_minor_key = m21.key.Key('d')`

    ```
    d_minor_key → <music21.key.Key of d minor>
    d_minor_key.mode → 'minor'
    d_minor_key.tonic → <music21.pitch.Pitch D>
    ```

music21.roman

- Das Modul *music21.roman* erlaubt es, Akkorde in Kontext einer angegebenen Tonart zu **analysieren** und zu **beschriften**.

- *Beispiel:*

```
c_major_key = m21.key.Key('C')
f_major_triad = m21.chord.Chord(['F3', 'C4', 'A4'])

label = m21.roman.romanNumeralFromChord(f_major_triad, c_major_key)

label
→ <music21.roman.RomanNumeral IV in C major>

label.figure
→ IV
```

music21.roman

- Umgekehrt kann aus jeder Akkord-Beschriftung im Rahmen einer angegebenen Tonart ein Beispiel-Akkord gebildet werden.

- *Beispiel :*

```
pitches_of_some_chord = m21.roman.RomanNumeral('V6', c_major_key).pitches

pitches_of_some_chord
→ (<music21.pitch.Pitch B4>, <music21.pitch.Pitch D5>, <music21.pitch.Pitch G5>)
```

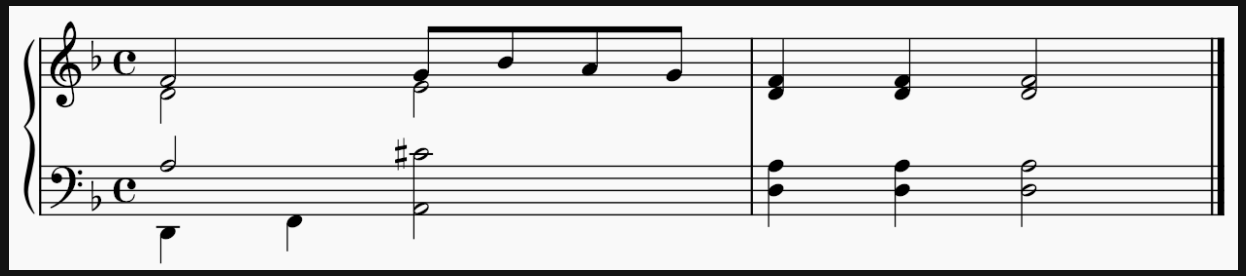
Automatische Bestimmung von Tonarten

Bestimmung von Tonarten erfolgt in music21 mithilfe des **Krumhansl-Schmuckler-Algorithmus**
(Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press, 1990)

Krumhansl-Schmuckler-Algorithmus

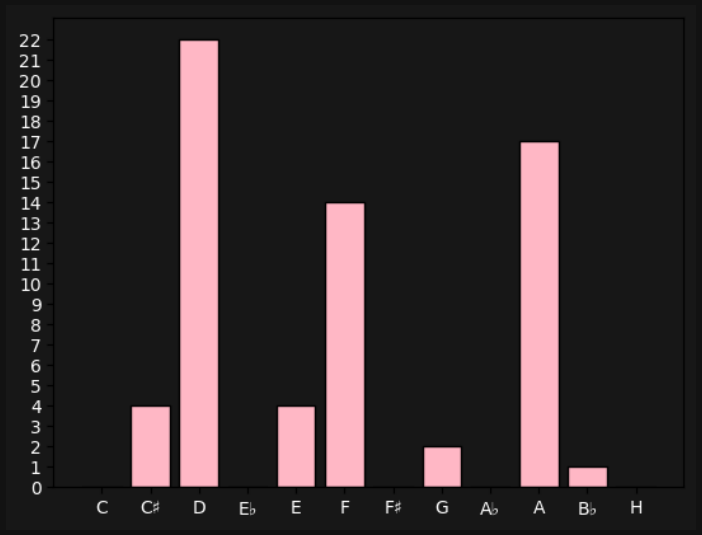
Bestimmung von Tonarten erfolgt in music21 mithilfe des **Krumhansl-Schmuckler-Algorithmus**
(Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press, 1990)

- *Idee:*
  - Für das zu analysierende Stück: statistische Auswertung der **Dauern allervorkommenden Pitch-Classes**



Gesamtdauern der verschiedenen Pitch-Classes:

C:	0 ♪	F♯ / G♭:	0 ♪
C♯ / D♭:	4 ♪	G:	2 ♪
D:	22 ♪	G♯ / A♭	0 ♪
D♯ / E♭:	0 ♪	A:	17 ♪
E:	4 ♪	A♯ / B♭	1 ♪
F:	14 ♪	H:	0 ♪



Krumhansl-Schmuckler-Algorithmus

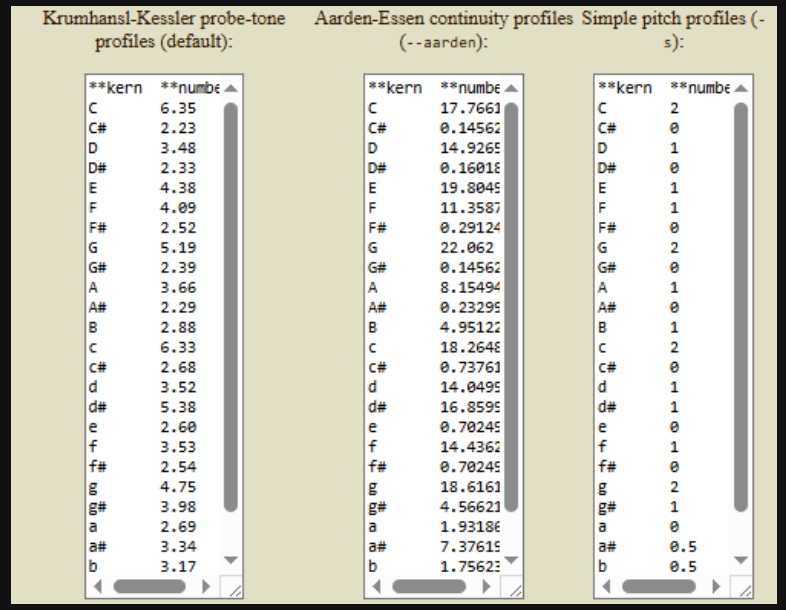
Bestimmung von Tonarten erfolgt in music21 mithilfe des Krumhansl-Schmuckler-Algorithmus
(Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press, 1990)

- *Idee:*
  - Für das zu analysierende Stück: statistische Auswertung der **Dauern allervorkommenden Pitch-Classes**
  - Abgleichen mit Referenzwerten für alle Tonarten

Krumhansl-Schmuckler-Algorithmus

Bestimmung von Tonarten erfolgt in music21 mithilfe des Krumhansl-Schmuckler-Algorithmus
(Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press, 1990)

- *Idee:*
  - Für das zu analysierende Stück: statistische Auswertung der **Dauern allervorkommenden Pitch-Classes**
  - Abgleichen mit Referenzwerten für alle Tonarten
    (→ Auswahlmöglichkeit zwischen **verschiedenen Profilen**, in denen die verschiedenen Tonhöhen unterschiedliche Gewichtungen haben)


Krumhansl-Kessler probe-tone profiles (default):

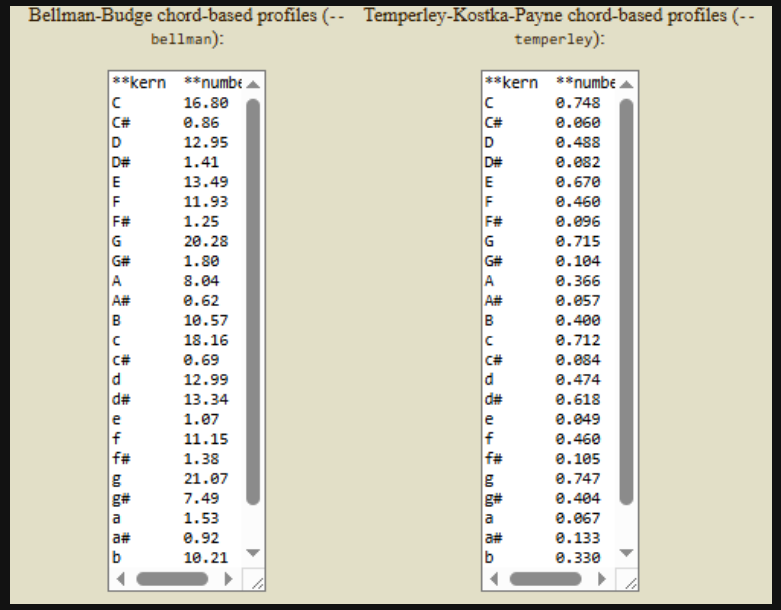
**kern	**numbe
C	6.35
C#	2.23
D	3.48
D#	2.33
E	4.38
F	4.09
F#	2.52
G	5.19
G#	2.39
A	3.66
A#	2.29
B	2.88
c	6.33
c#	2.68
d	3.52
d#	5.38
e	2.60
f	3.53
f#	2.54
g	4.75
g#	3.98
a	2.69
a#	3.34
b	3.17

Aarden-Essen continuity profiles (--aarden):

**kern	**numbe
C	17.7661
C#	0.14562
D	14.9265
D#	0.16018
E	19.8045
F	11.3587
F#	0.29124
G	22.062
G#	0.14562
A	8.15494
A#	0.23295
B	4.95122
c	18.2648
c#	0.73761
d	14.0495
d#	16.8595
e	0.70245
f	14.4362
f#	0.70245
g	18.6161
g#	4.56621
a	1.93186
a#	7.37615
b	1.75623

Simple pitch profiles (-s):

**kern	**numbe
C	2
C#	0
D	1
D#	0
E	1
F	1
F#	0
G	2
G#	0
A	1
A#	0
B	1
c	2
c#	0
d	1
d#	1
e	0
f	1
f#	0
g	2
g#	1
a	0
a#	0.5
b	0.5


Bellman-Budge chord-based profiles (--bellman):

**kern	**numbe
C	16.80
C#	0.86
D	12.95
D#	1.41
E	13.49
F	11.93
F#	1.25
G	20.28
G#	1.80
A	8.04
A#	0.62
B	10.57
c	18.16
c#	0.69
d	12.99
d#	13.34
e	1.07
f	11.15
f#	1.38
g	21.07
g#	7.49
a	1.53
a#	0.92
b	10.21

Temperley-Kostka-Payne chord-based profiles (--temperley):

**kern	**numbe
C	0.748
C#	0.060
D	0.488
D#	0.082
E	0.670
F	0.460
F#	0.096
G	0.715
G#	0.104
A	0.366
A#	0.057
B	0.400
c	0.712
c#	0.084
d	0.474
d#	0.618
e	0.049
f	0.460
f#	0.105
g	0.747
g#	0.404
a	0.067
a#	0.133
b	0.330

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

Krumhansl-Schmuckler-Algorithmus

Bestimmung von Tonarten erfolgt in music21 mithilfe des Krumhansl-Schmuckler-Algorithmus
(Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press, 1990)

- *Idee:*
  - Für das zu analysierende Stück: statistische Auswertung der **Dauern allervorkommenden Pitch-Classes**
  - Abgleichen mit Referenzwerten für alle Tonarten
    (→ Auswahlmöglichkeit zwischen **verschiedenen Profilen**, in denen die verschiedenen Tonhöhen unterschiedliche Gewichtungen haben)
  - Ergebnis: **Angabe der wahrscheinlichsten Tonart(en)** für das analysierte Stück

→ ausführliche Beschreibung des Algorithmus: https://extras.humdrum.org/man/keycor/

Krumhansl-Schmuckler-Algorithmus

Bestimmung von Tonarten erfolgt in music21 mithilfe des Krumhansl-Schmuckler-Algorithmus
(Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press, 1990)

- *Idee:*
  - Für das zu analysierende Stück: statistische Auswertung der **Dauern allervorkommenden Pitch-Classes**
  - Abgleichen mit Referenzwerten für alle Tonarten
    (→ Auswahlmöglichkeit zwischen **verschiedenen Profilen**, in denen die verschiedenen Tonhöhen unterschiedliche Gewichtungen haben)
  - Ergebnis: **Angabe der wahrscheinlichsten Tonart(en)** für das analysierte Stück

  → ausführliche Beschreibung des Algorithmus: https://extras.humdrum.org/man/keycor/

- 2 Nutzungsmöglichkeiten in music21:
  - Möglichkeit 1: mithilfe der Methode `.analyze('key')`
  - Möglichkeit 2: mit dem Modul `music21.analysis.discrete`

Automatische Bestimmung von Tonarten

Möglichkeit 1: Krumhansl-Schmuckler-Analyse mit der Methode `.analyze('key')`

- *Schritt 1:* Laden einer musikalischen Sequenz in ein Stream-Objekt

```
my_stream = …
```

- *Schritt 2:* Automatische Bestimmung der Tonart

```
estimated_key = my_stream.analyze('key')
```

Automatische Bestimmung von Tonarten

Möglichkeit 2: Krumhansl-Schmuckler mit dem Modul `music21.analysis.discrete`:

Referenz in Dokumentation:
https://web.mit.edu/music21/doc/moduleReference/moduleAnalysisDiscrete.html

- *Schritt 1:* Laden einer musikalischen Sequenz in ein Stream-Objekt

  ```
  my_stream = …
  ```

- *Schritt 2:* Definition eines Analyse-Prozessors und Auswahl eines Profils mit Referenz-Gewichtungen für den Krumhansl-Schmuckler-Algorithmus (z.B. der Aarden-Essen-Gewichtungen)

  ```
  analysis_processor = m21.analysis.discrete.AardenEssen()
  ```

- *Schritt 3:* Automatische Bestimmung der Tonart

  ```
  analysis_processor.getSolution(my_stream)
  ```

Einige Fragestellungen

- Frage 1: „Wie viele Moll-Werke im Bach-Corpus enden mit einer Picardischen Terz?“

  → siehe „CMF 1 – Woche 04_Vorlesung_Notebook“

- Frage 2: „Ist die Angabe des häufigsten Akkords ein geeignetes Werkzeug, um die Tonart eines Stücks zu bestimmen?“

  → siehe „CMF 1 – Woche 04_Vorlesung_Notebook“