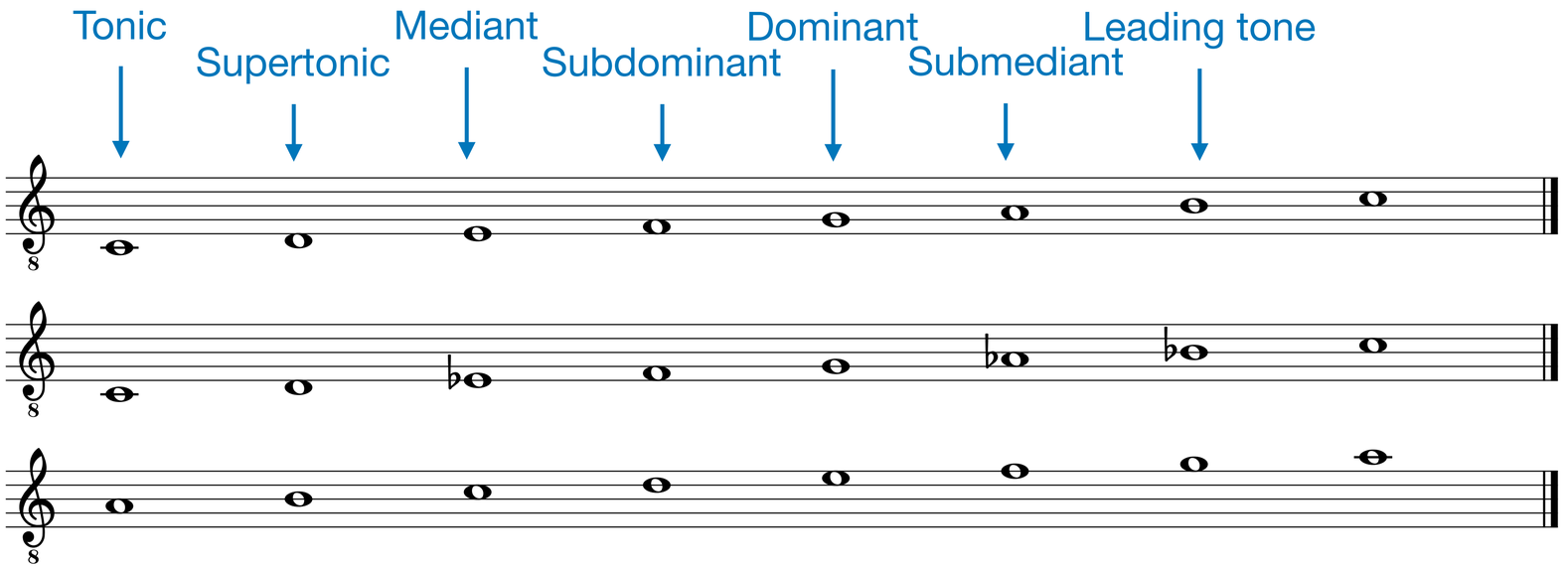
Homework 1

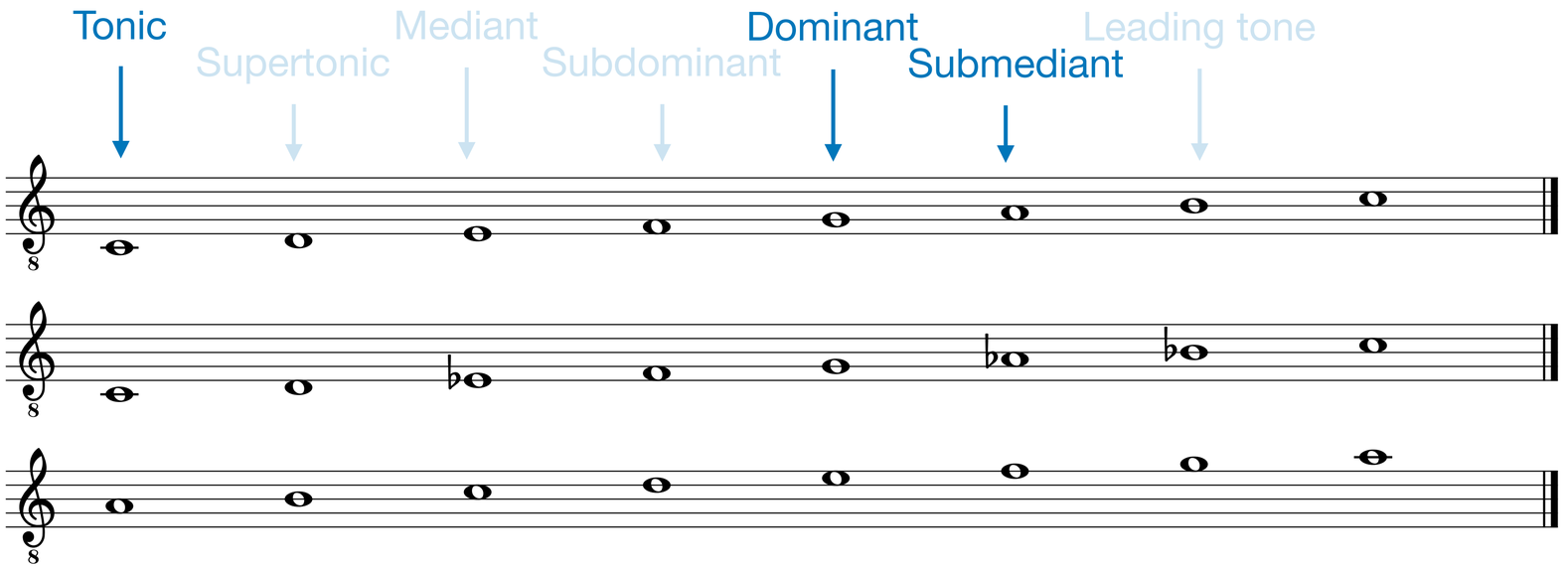
2020/04/21

Key Finding Algorithm

Key

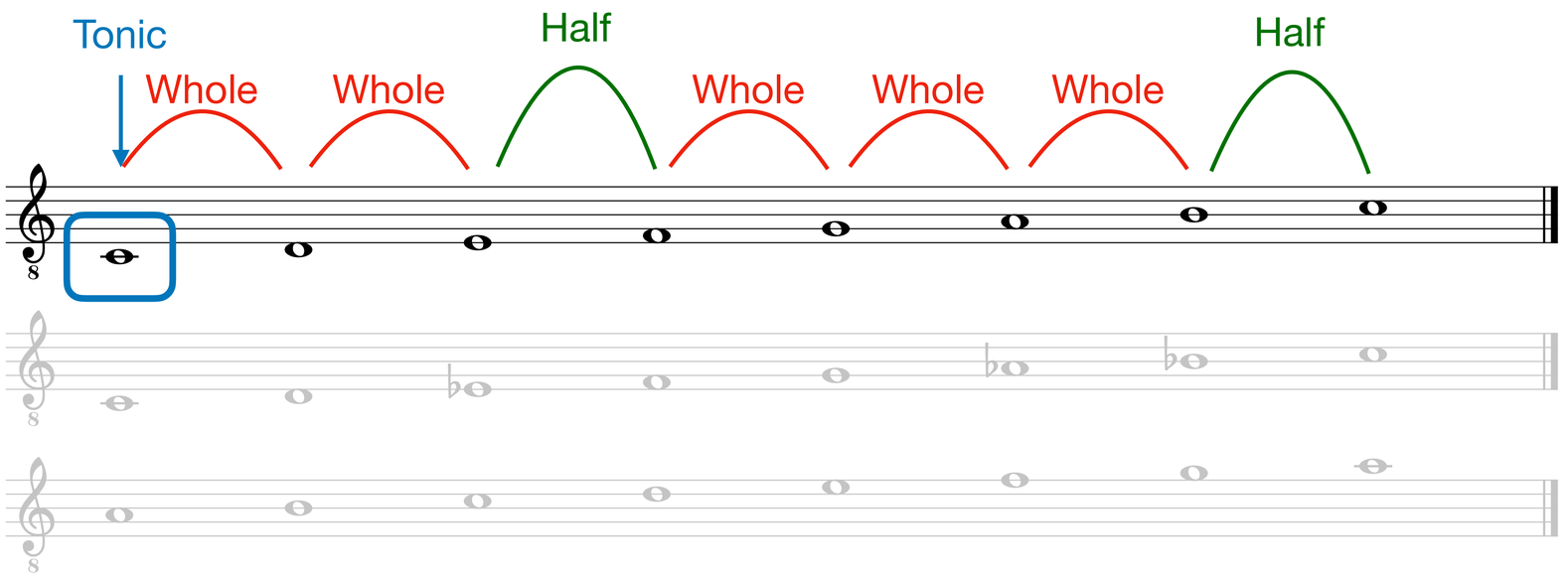

Tonic
Supertonic
Mediant
Subdominant
Dominant
Submediant
Leading tone

Key


Tonic
Supertonic
Mediant
Subdominant
Dominant
Submediant
Leading tone

Key

Major Scale

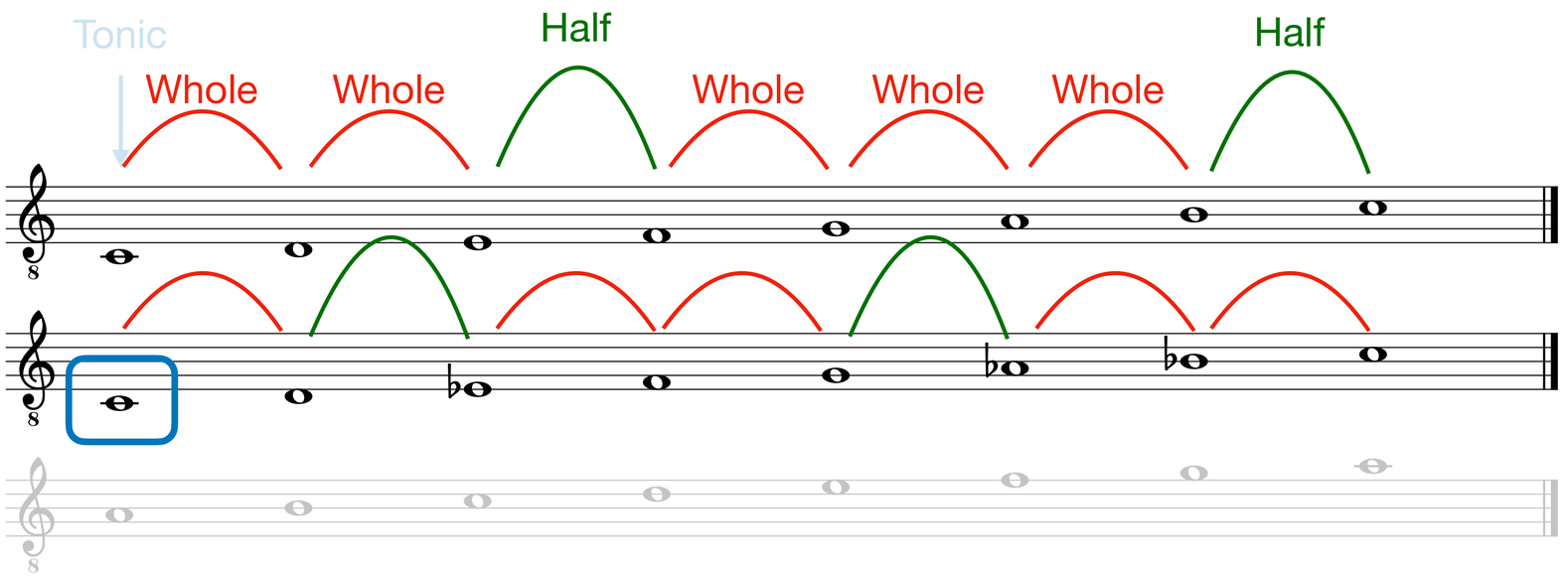


C Major Scale

Key

Major Scale

Parallel Minor Scale



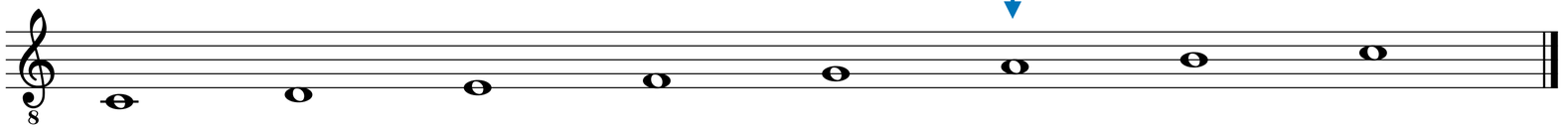
C Major Scale

C Minor Scale

Key

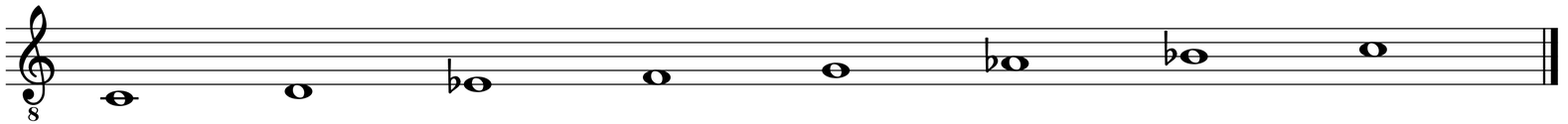
Submediant

Major Scale

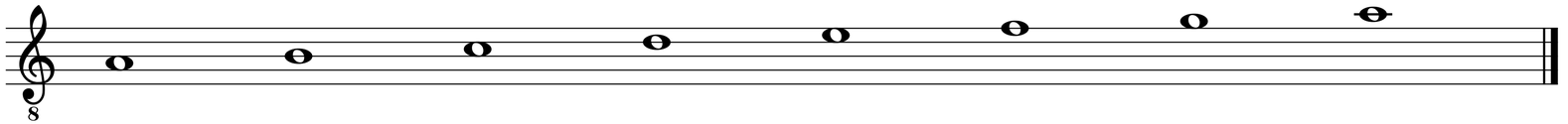


C Major Scale

Parallel Minor Scale



C Minor Scale

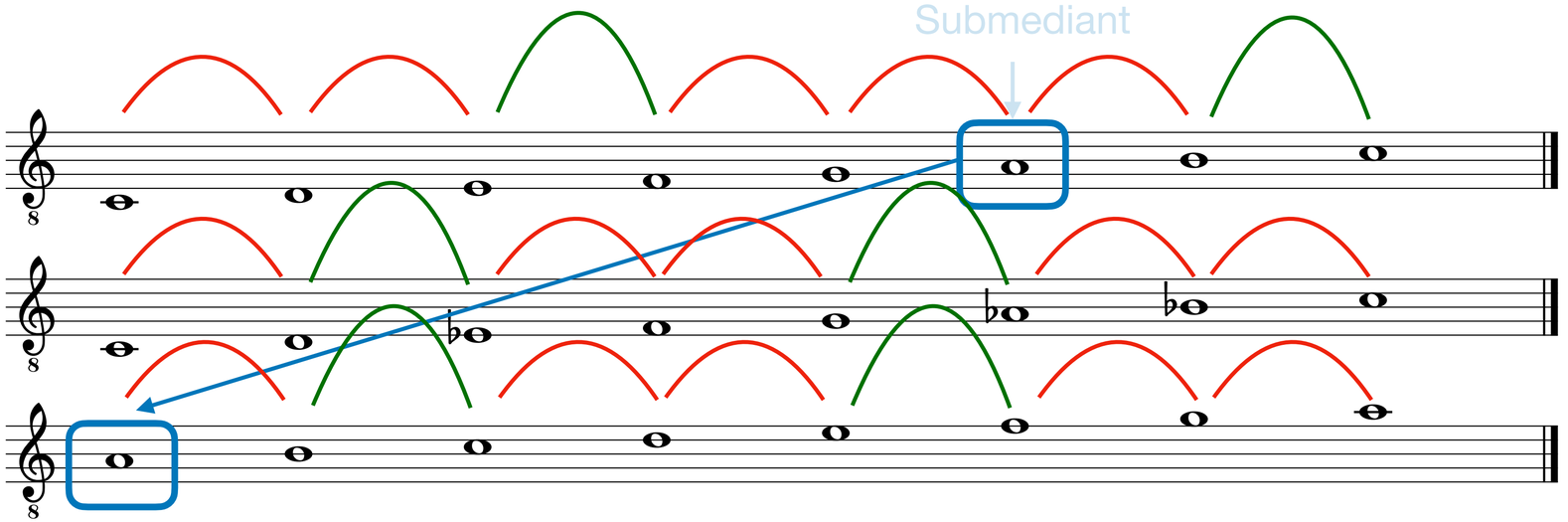


Key

Major Scale

Parallel Minor Scale

Relative Minor Scale



C Major Scale

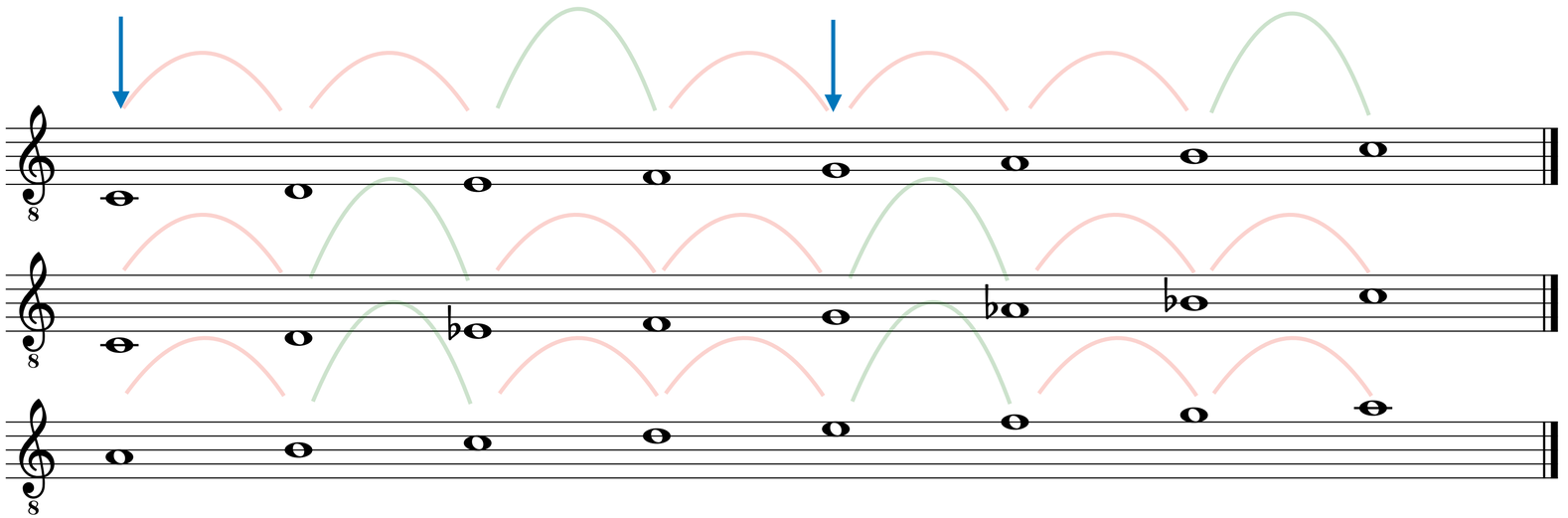
C Minor Scale

A Minor Scale

Key

Tonic

Dominant



Major Scale — C Major Scale

Parallel Minor Scale — C Minor Scale
Same Tonic

Relative Minor Scale — A Minor Scale
Same Key Signature

Prerequisite

- librosa : librosa.github.io/librosa/
- pretty-midi : craffel.github.io/pretty-midi/
- mir_eval : https://craffel.github.io/mir_eval/
- Numpy
- Scipy
- GTZAN : Dataset, Annotation
- BFS-FH : https://drive.google.com/open?id=1gEV87HsdM_4K1EuaOEDjs2yL5-G37UvJ
- GiantSteps : https://github.com/GiantSteps/giantsteps-key-dataset
- A-Maps : https://drive.google.com/drive/folders/1IKMUAqsLTy8sBmbCaaDiK9yWowlHLI8z?usp=sharing

Task 1

Binary template matching for global (or clip-level) key detection for audio data.
Q1/Q2/Q3

1. Find the tonic note.

2. Find major/minor mode.

3. Report the accuracy.

Task 1

Binary template matching for global (or clip-level) key detection for audio data.
Q1/Q2/Q3

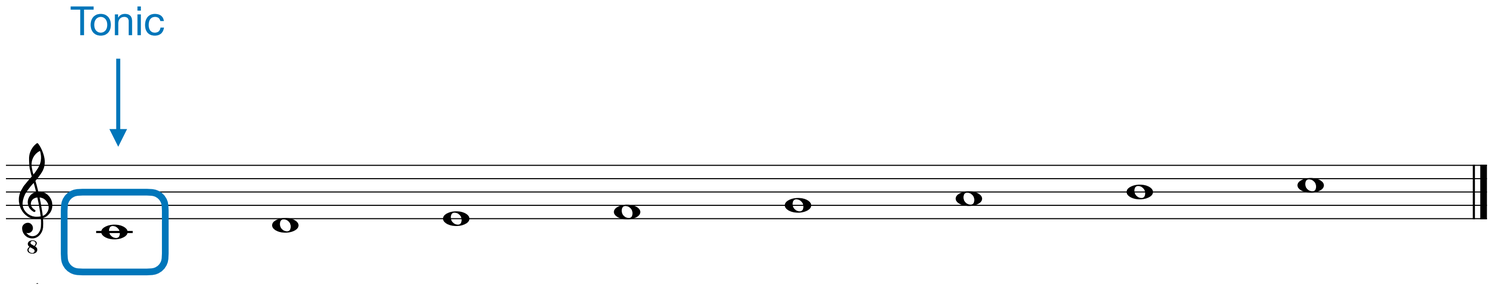
- Find the tonic note:

1. Sum up the chroma features into one chroma vector.

Chroma Features $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N], \ \mathbf{z}_i \in \mathbb{R}^{12}$ *librosa.feature.chroma_stft*

Chroma Vector $\mathbf{x} = \sum_{i=1}^{N} \mathbf{z}_i$

2. Find the maximal value in the chroma vector → Tonic Note



Task 1

Binary template matching for global (or clip-level) key detection for audio data.
Q1/Q2/Q3

- Find major/minor mode:

1. Template matching, find the correlation coefficient between templates of major/minor scales.

Templates

$$\mathbf{y}_{\text{C Major key}} = [1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1]$$

$$\mathbf{y}_{\text{c minor key}} = [1\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0]$$

Correlation Coefficient

$$R(\mathbf{x},\mathbf{y}) = \frac{\sum_{k=1}^{12}(\mathbf{x}_k - \bar{\mathbf{x}})(\mathbf{y}_k - \bar{\mathbf{y}})}{\sqrt{\sum_{k=1}^{12}(\mathbf{x}_k - \bar{\mathbf{x}})^2 \sum_{k=1}^{12}(\mathbf{y}_k - \bar{\mathbf{y}})^2}}$$

Use existed packages, such as

scipy.stats.pearsonr

2. If the coefficient of major key is greater than minor key, we say the piece is in major key, and vice versa.

Task 1

Binary template matching for global (or clip-level) key detection for audio data.
Q1/Q2/Q3

- Report the accuracy.

Annotation

A	A#	B	C	C#	D	D#	E	F	F#	G	G#
0	1	2	3	4	5	6	7	8	9	10	11
a	a#	b	c	c#	d	d#	e	f	f#	g	g#
12	13	14	15	16	17	18	19	20	21	22	23

Example:

G Major → 10

G minor → 22

Accuracy

$$\mathrm{ACC} = \frac{\#\ \text{of correct detection}}{\#\ \text{of all music pieces}}$$

Task 1

Binary template matching for global (or clip-level) key detection for audio data.
Q1/Q2/Q3

1. Find the tonic note. → A
2. Find major/minor mode. → A Minor
3. Report the accuracy. → Predicted: 13
   GT : 13

Task1 - Q1

Binary template matching for global (or clip-level) key detection for audio data.
Q1/Q2/Q3

- Find the key of all pieces of the **GiantSteps** dataset, and the 9 genres*(exclude the classical genre)* in the **GTZAN** dataset.

- Overall accuracy of the two datasets? The accuracy of each genre in **GTZAN**? Which genre have lower accuracy? Why?(from musical point of view)

- Exclude pieces with unknown key labels.

Task1 - Q2

Binary template matching for global (or clip-level) key detection for audio data.
Q1/Q2/Q3

- Repeat Q1 with *factor of logarithmic compression* γ, with $\gamma = 1, 10, 100, 1000$.

- Discuss results.

Task1 - Q3

Binary template matching for global (or clip-level) key detection for audio data.
Q1/Q2/Q3

- Some error detection results are closer to the ground truth label. Example:

Label	Results
C major	G Major (Perfect fifth)
	C minor (parallel)
	A minor (Relative)

- Therefore, use the scoring rule of MIREX key detection competition:

Relation to correct key	Points
Same	1.0
Perfect fifth	0.5
Relative major/minor	0.3
Parallel major/minor	0.2
Other	0.0

Task1 - Q3

Binary template matching for global (or clip-level) key detection for audio data.
Q1/Q2/Q3

- New accuracy:

$$\mathrm{ACC} = \frac{\#\ \mathrm{Same} + 0.5(\#\ \mathrm{Fifth}) + 0.3(\#\ \mathrm{Relative}) + 0.2\ \ (\mathrm{Parallel})}{\#\ \mathrm{of\ all\ music\ pieces}}$$

mir_eval.key

- Repeat Q1 with the new accuracy, discuss results.

Task 2

Krumhansl-Schmuckler key-finding algorithm.
Q4

- Instead of using binary templates, assign values to templates according to human perceptual experiments.

Major key			Minor key		
Name	Binary	K-S	Name	Binary	K-S
Tonic	1	6.35	Tonic	1	6.33
	0	2.23		0	2.68
Supertonic	1	3.48	Supertonic	1	3.52
	0	2.33	Mediant	1	5.38
Mediant	1	4.38		0	2.60
Subdominant	1	4.09	Subdominant	1	3.53
	0	2.52		0	2.54
Dominant	1	5.19	Dominant	1	4.75
	0	2.39	Submediant	1	3.98
Submediant	1	3.66		0	2.69
	0	2.29	Leading tone	1	3.34
Leading tone	1	2.88		0	3.17

- No need to probe the tonic note first. Instead, find the maximal correlation coefficient among the major/ minor scales and the 12 notes.

Task 1

1 note (The probed tonic note) X Major/minor templates

Task 2

12 note X Major/minor templates

Task 2 - Q4

Krumhansl-Schmuckler key-finding algorithm.
Q4

- Repeat Q1/Q2/Q3 with *Krumhansl-Schumuckler's method*.

- Discuss the results: Which feature is better? Is there any limitation of these method? Is there any limitation of using GTZAN dataset for key finding?

- **BoNUs!** Use *librosa.feature.chroma_cens* and *librosa.feature.chroma_cqt*. Compare the results.

Task 3

Local key detection
Q5

- We assumed that one piece has only on key, yet key of a piece is likely to change over time.
- So in this task, we detect local keys.

Task 3 - Q5

Local key detection
Q5

- Detect local keys of a music piece, and output the key of the piece every second.

- Work on two datasets: **BFS-FH** and **A-MAPS**.

  - **A-MAPS**: MIDI files. Use *pretty_midi.Instrument.get_chroma* to get the chroma vector; use *pretty_midi.KeySignature* to read the key label of each piece.

  - **BFS-FH**:

Time(sec)	Key Name
1	F
2	F
3	F
4	F
…	…
31	F
32	A-
33	A-
…	…

Task 3 - Q5

Local key detection
Q5

- Accuracy:

$$\mathrm{ACC} = \frac{\#\ \text{of correct detection}}{\#\ \text{of time instances (detections) in all music pieces}}$$

$$\mathrm{ACC} = \frac{\#\ \text{Same} + 0.5(\#\ \text{Fifth}) + 0.3(\#\ \text{Relative}) + 0.2\ \ (\text{Parallel})}{\#\ \text{of all time instances (detections) in all music pieces}}$$

- Discuss results.

- If you use machine-learning-based methods, use the following partition and report the result of the test set.

BPS-FH	Train	1, 3, 5, 11, 16, 19, 20, 22, 25, 26, 32
	Validation	6, 13, 14, 21, 23, 31
	Test	8, 12, 18, 24, 27, 28
A-MAPS	Train	a~l (more information)
	Validation	m~p
	Test	s~z

BoNUs! - Q6

Design an algorithm! ♪♪♪ ヽ(･ˇ∀ˇ･ゞ)

Q6

- Design an algorithm that can outperforms the introduced algorithms in at least two of the five genres*(Pop, Blues, Metal, Hip Hop, Rock)* in this assignment.

Reminder

(´△｀)♪

- Submit your .zip file containing (1) Report. (2) Codes. to ILMS before the deadline.
- Do NOT contain datasets in your zip file!
- Zip file title: *HW1_<your student ID>.zip*
- Report Title: *HW1_<your student ID>.pdf*
- Deadline : *May 12, Grading policy of late submission will be announced soon.*