Unmixer

An interface for extracting and remixing loops

Jordan B. L. Smith
Yuta Kawasaki
Masataka Goto

AIST

Suppose you are a remix artist...

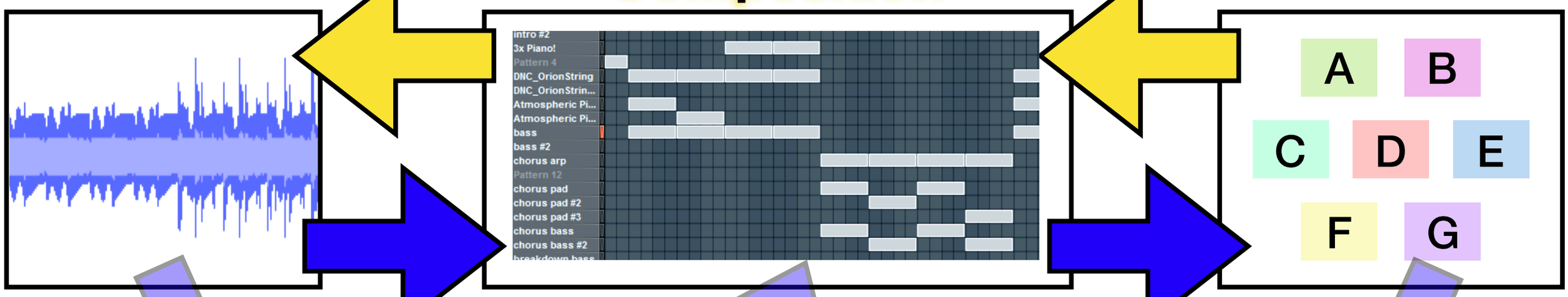
Remix artists have the STEREO MIX...

...but they want the ISOLATED LOOPS.

Song

← Composition ←

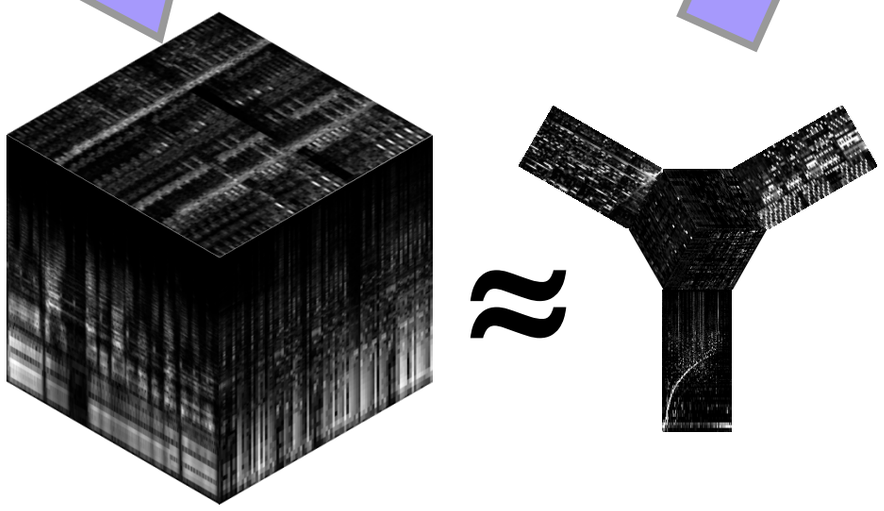
Ingredients



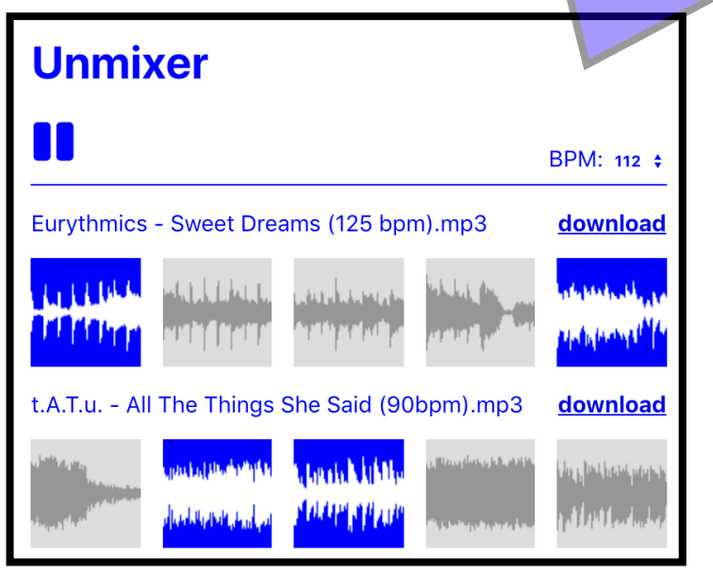
→ De-composition →

Goal of Unmixer: extract loops from mixed audio

Algorithm



Interface



Unmixer Interface

1. User uploads audio file

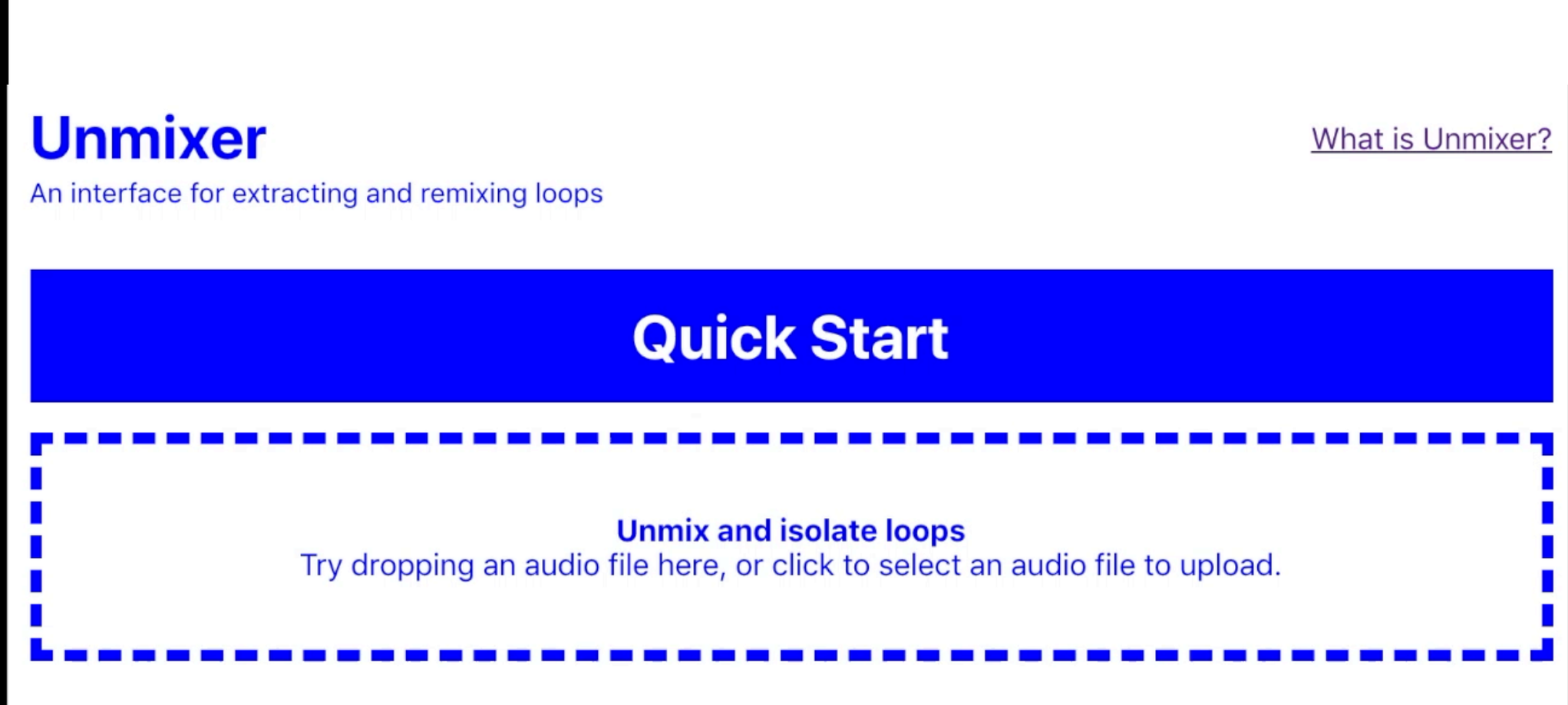
2. Wait for results (several minutes)

3. Click tiles to add or remove them from the mix

4. Add more songs, test out mashups

5. Adjust tempo

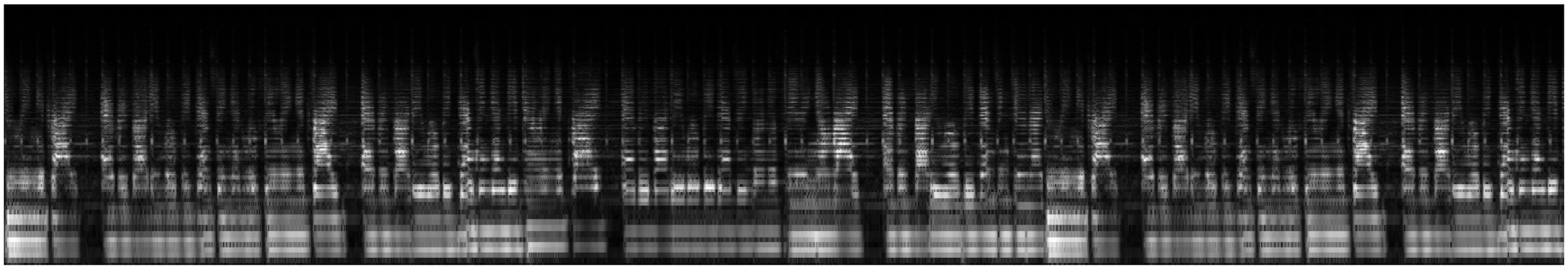
6. Download loops to remix in your favourite DAW



Unmixer Algorithm

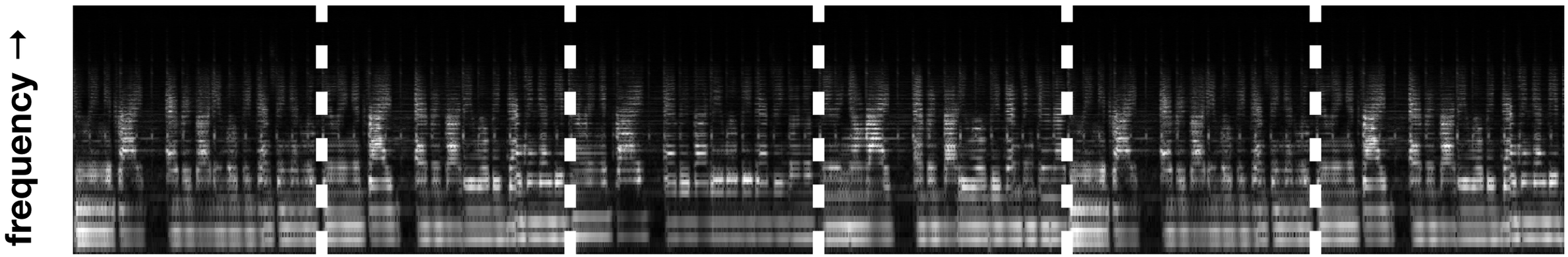
Algorithm

1. Compute spectrum



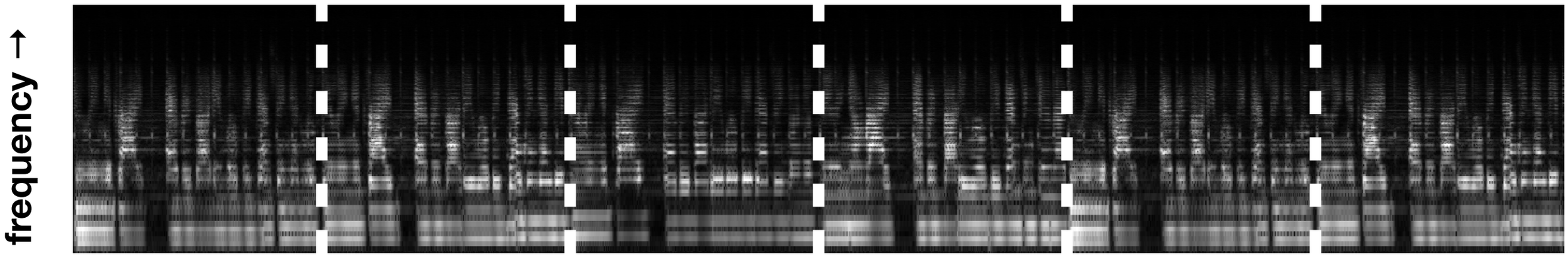
Algorithm

1. **Compute spectrum**
2. **Estimate downbeats (madmom)**



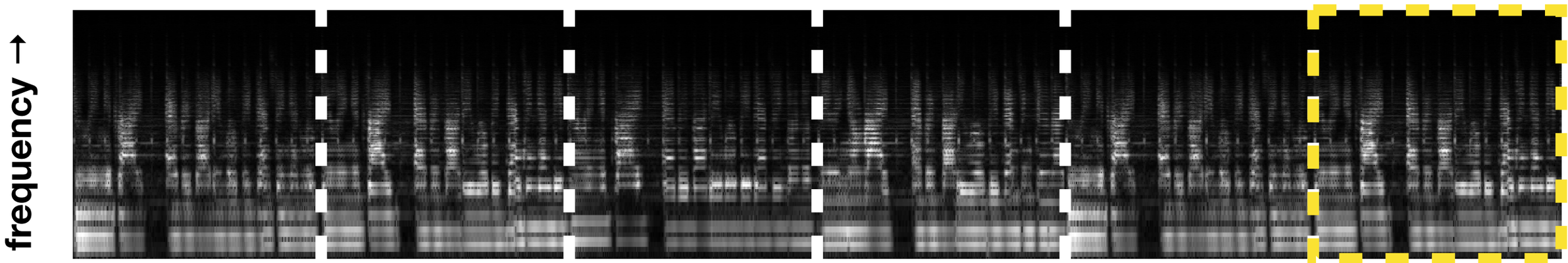
Algorithm

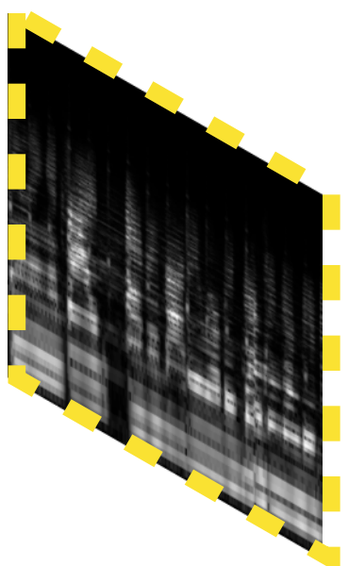
1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**



Algorithm

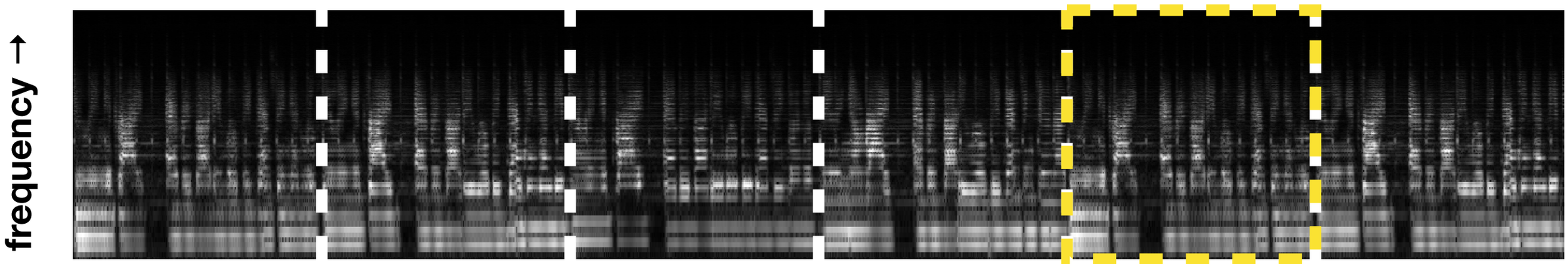
1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**

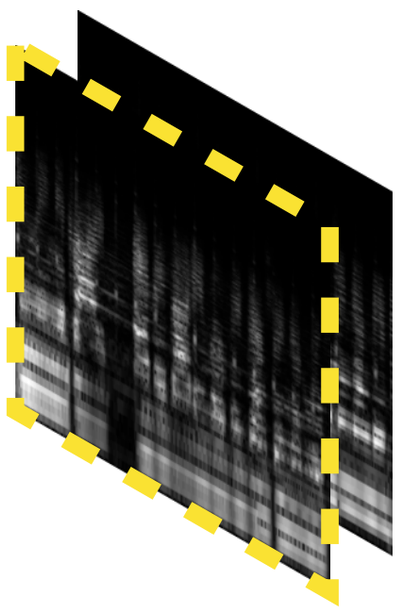




Algorithm

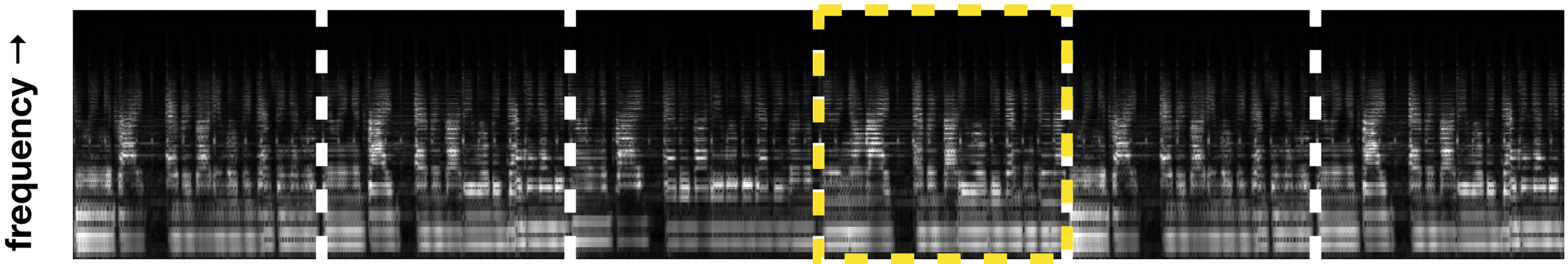
1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**

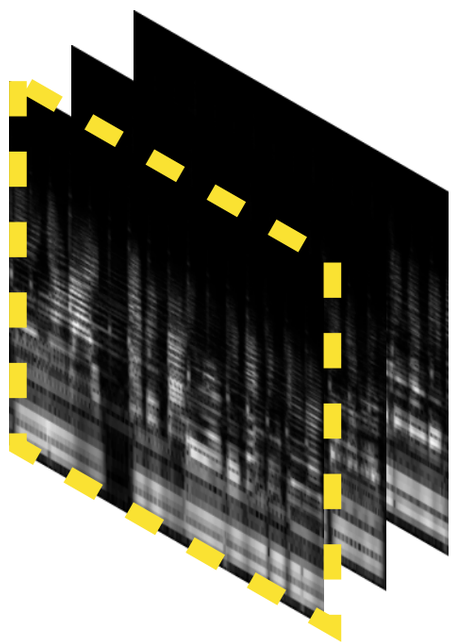




Algorithm

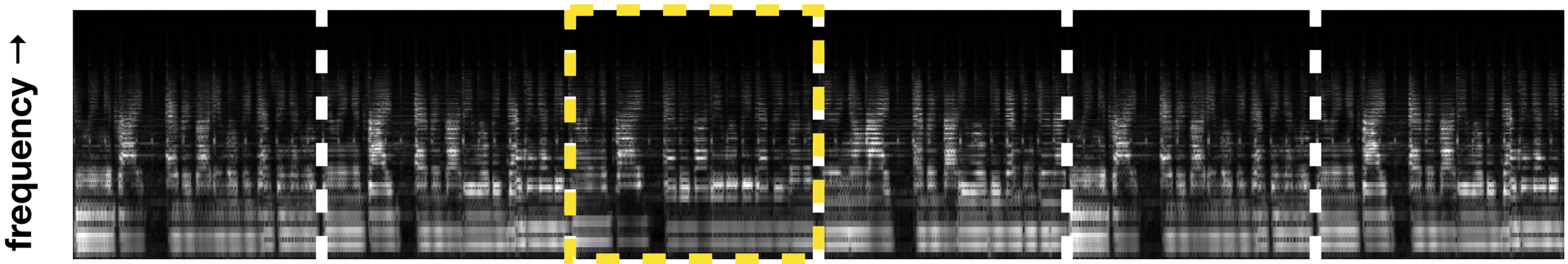
1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**

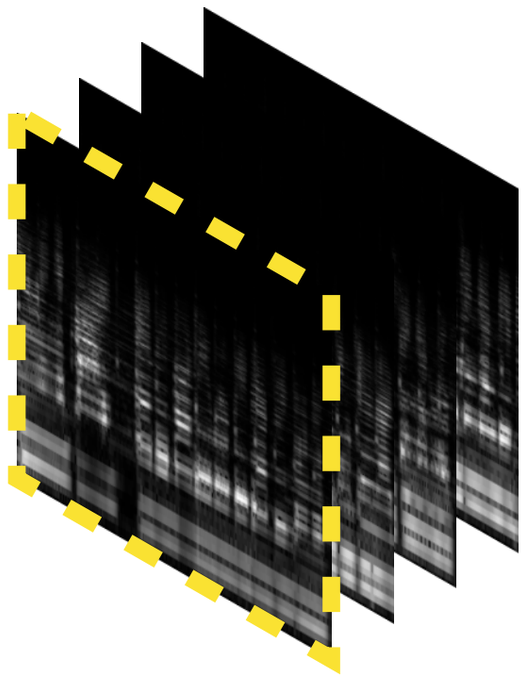




Algorithm

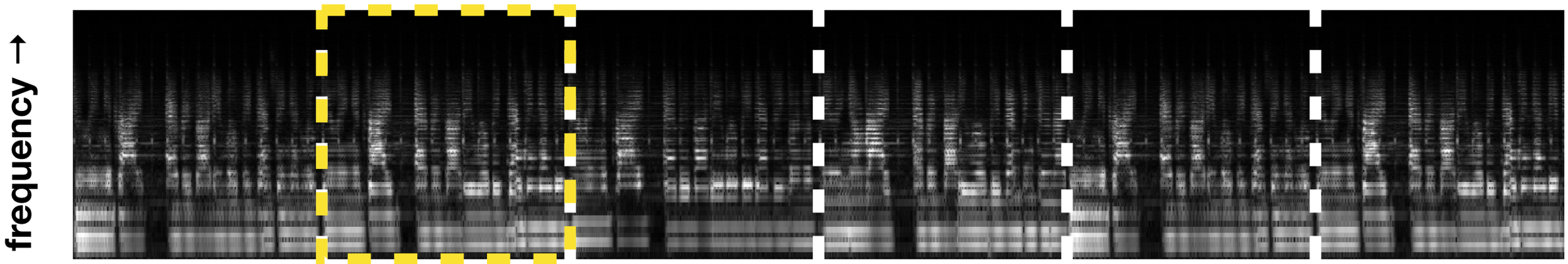
1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**

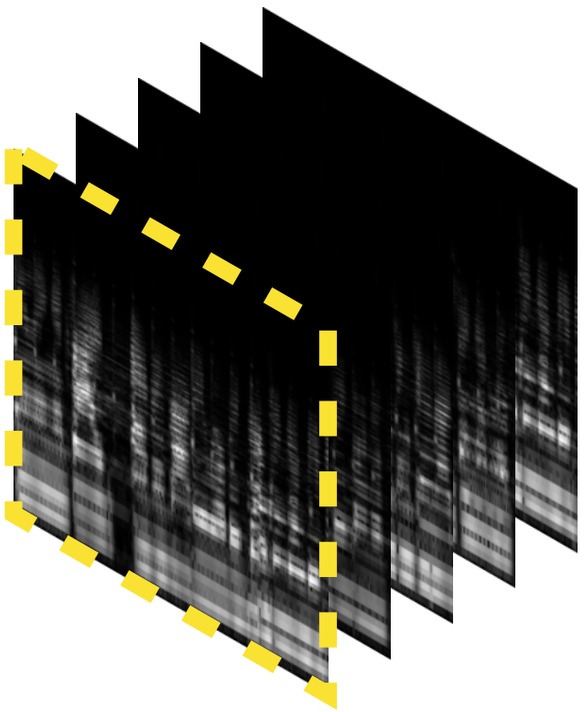




Algorithm

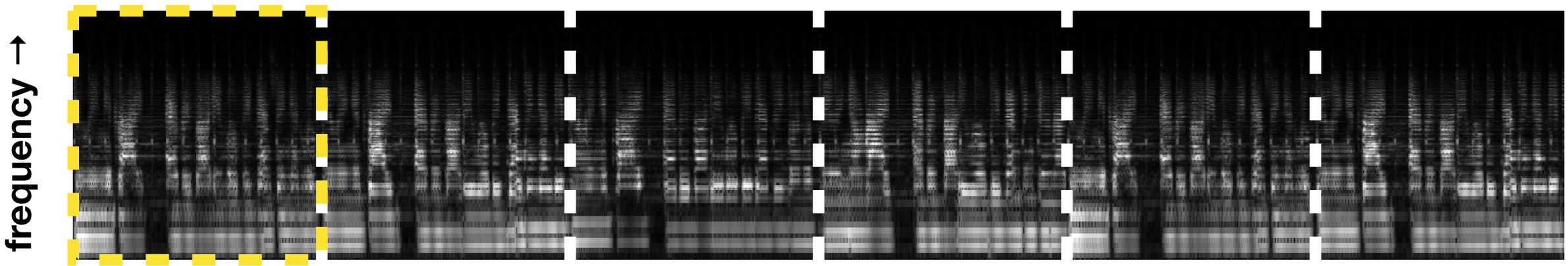
1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**

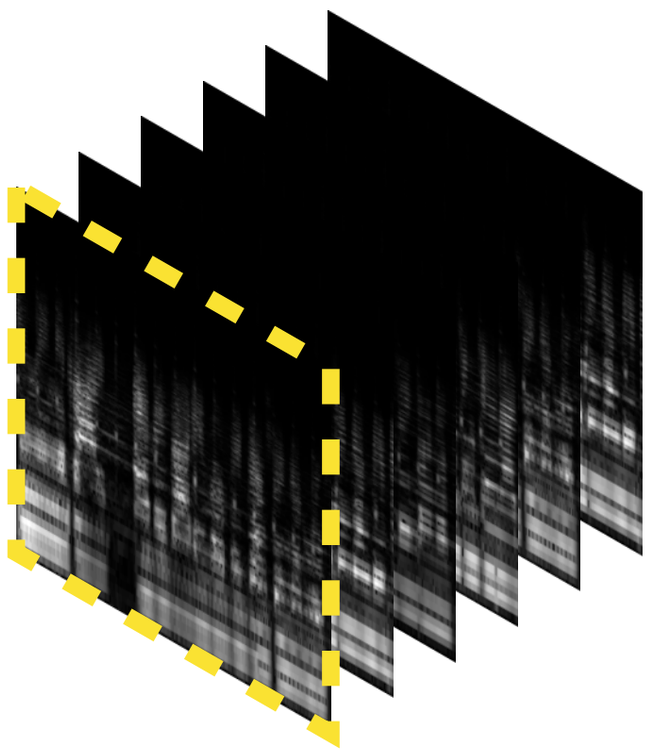




Algorithm

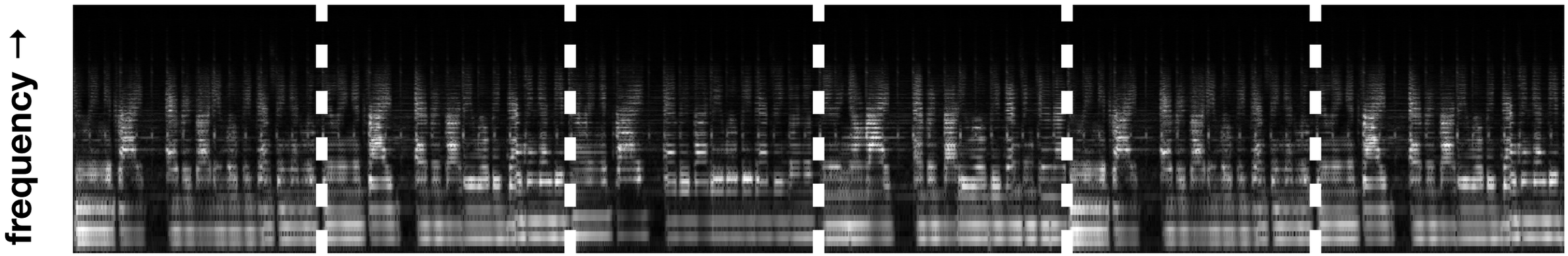
1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**

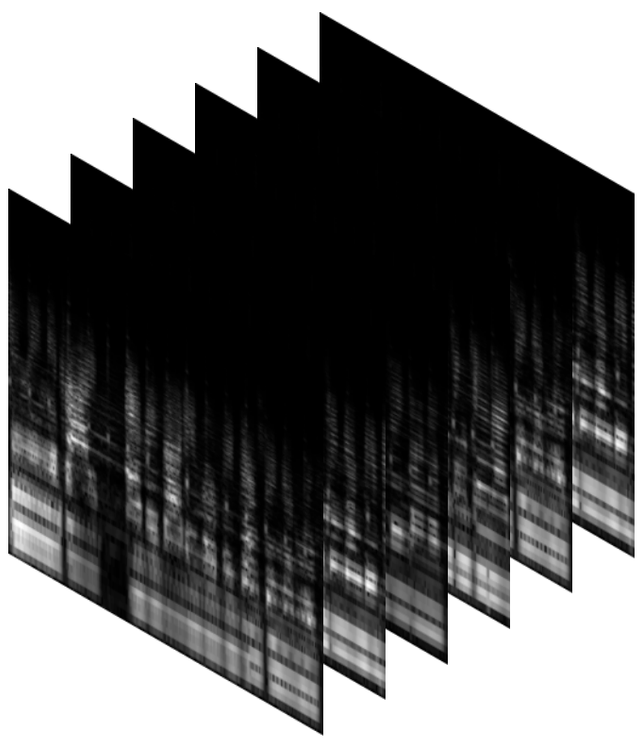




Algorithm

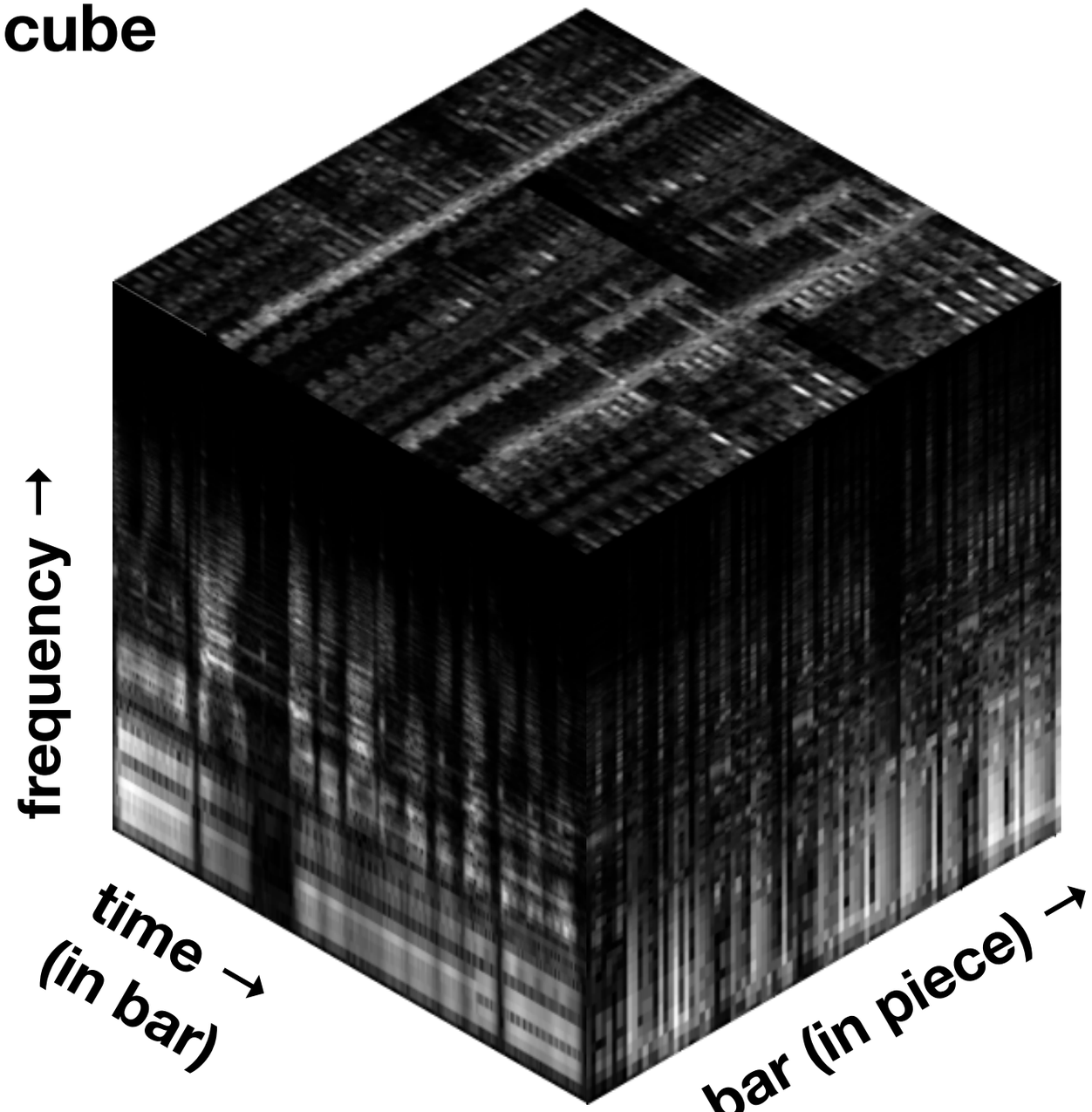
1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**





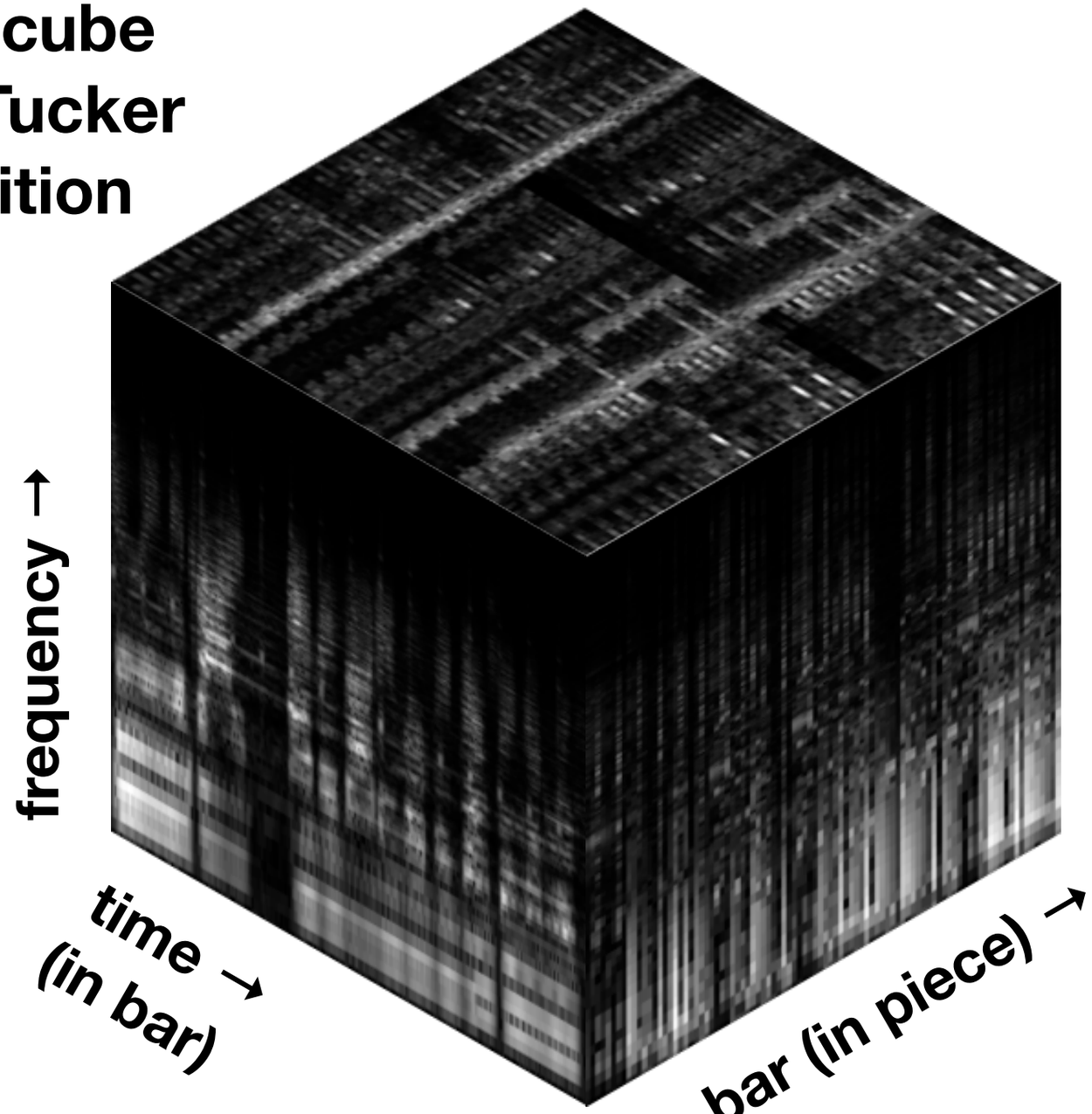
Algorithm

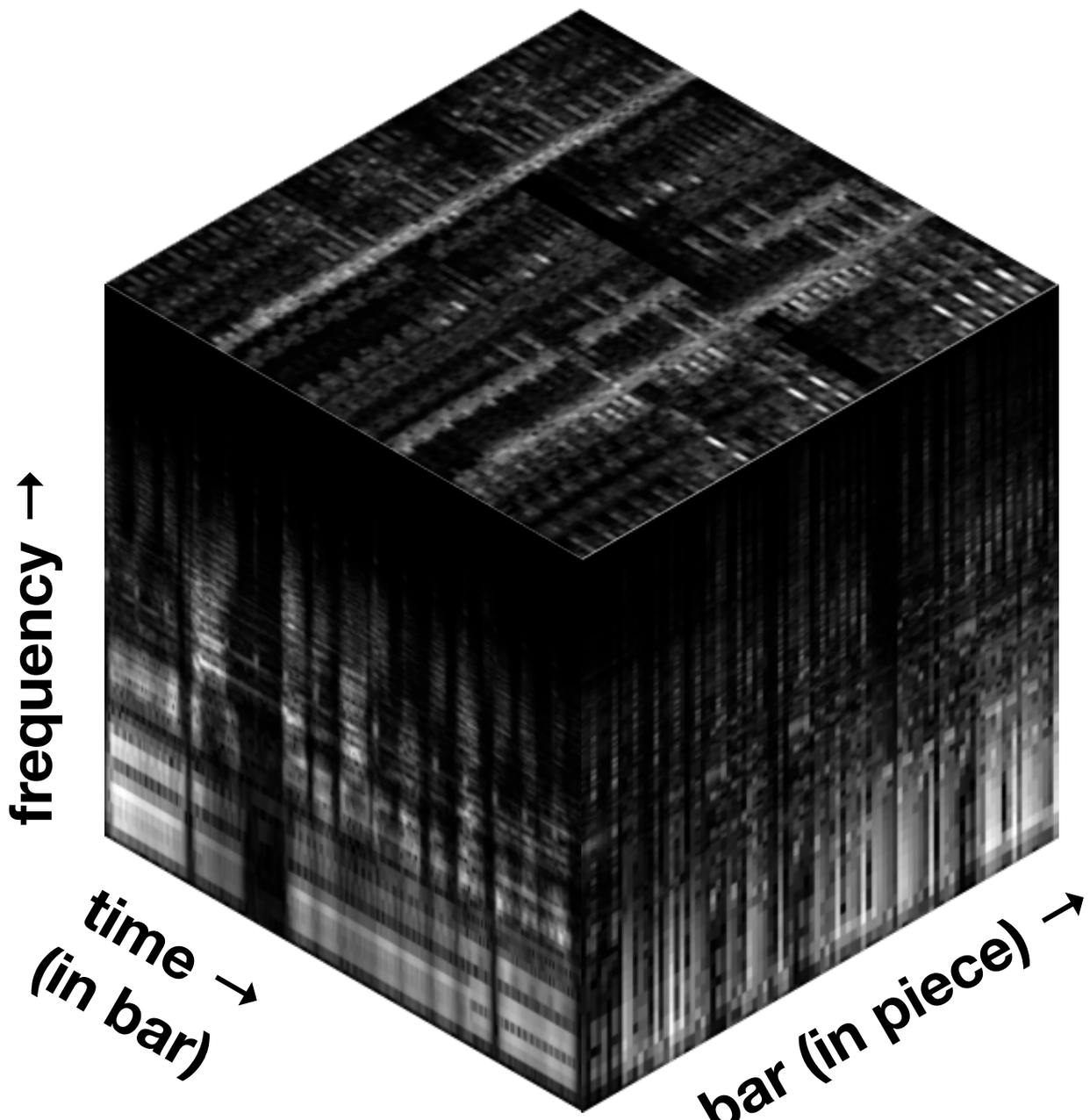
1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**



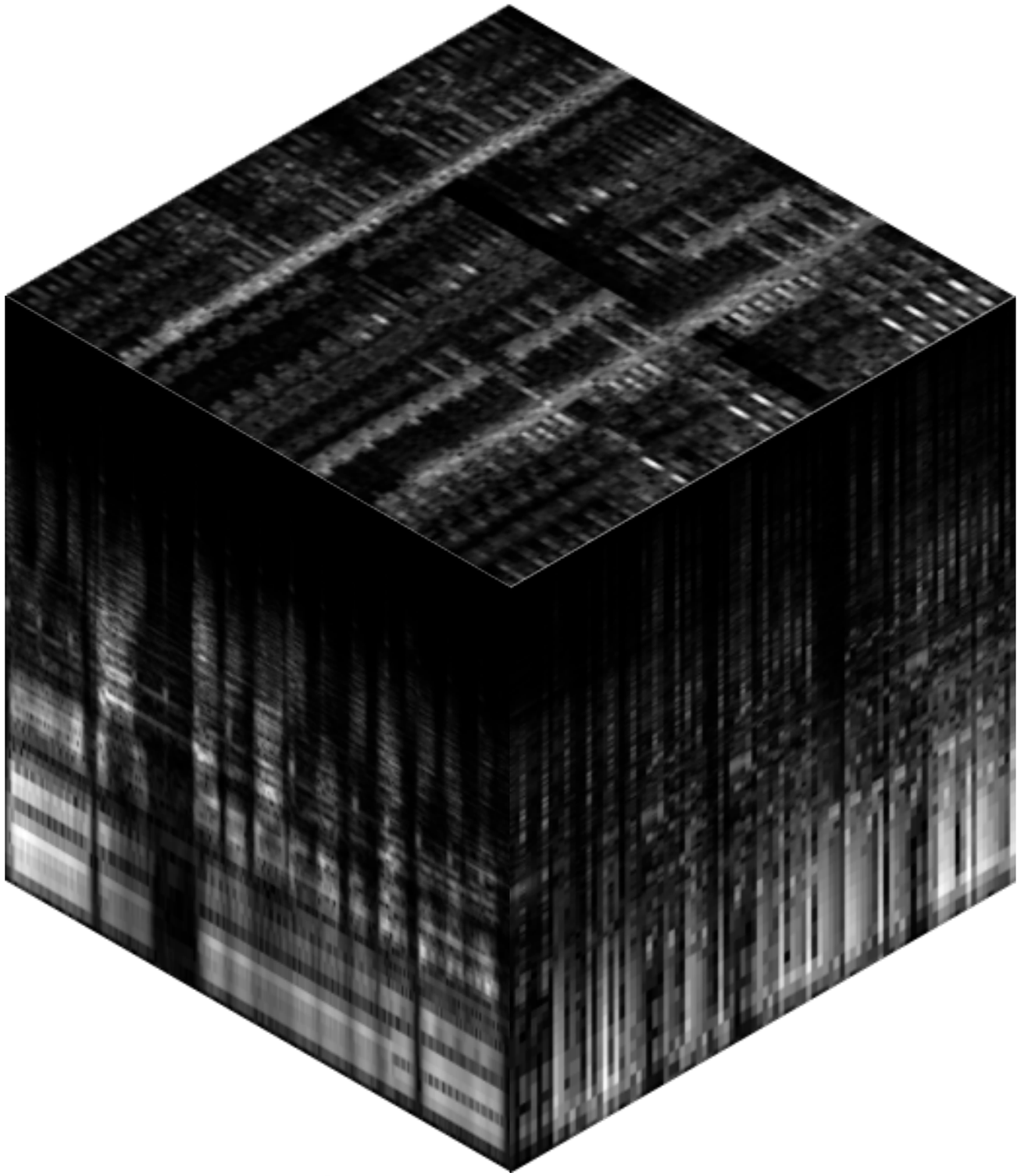
Algorithm

1. **Compute spectrum**
2. **Estimate downbeats (madmom)**
3. **Stack into cube**
4. **Compute Tucker decomposition**

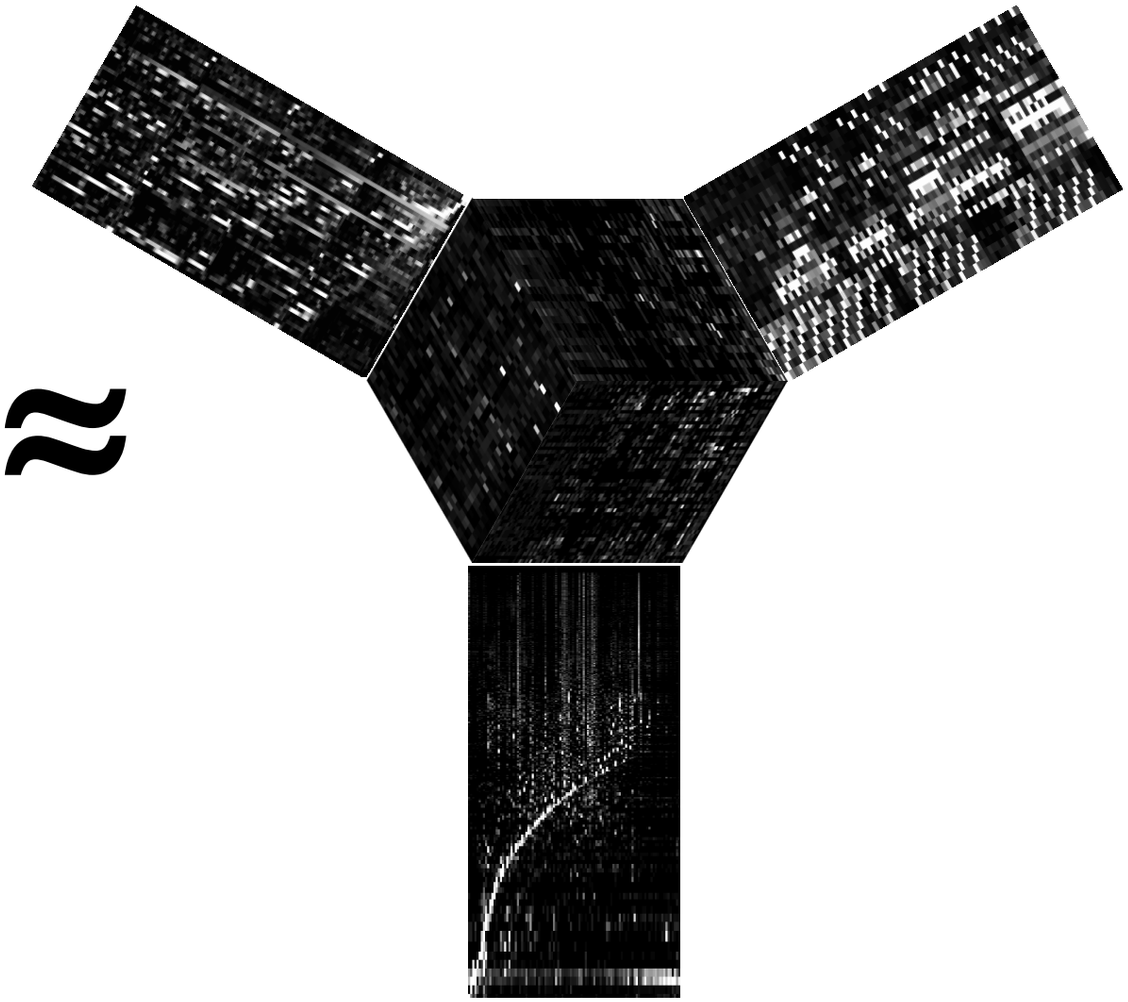



frequency →
time →
(in bar)
bar (in piece) →

Tucker decomposition:



≈

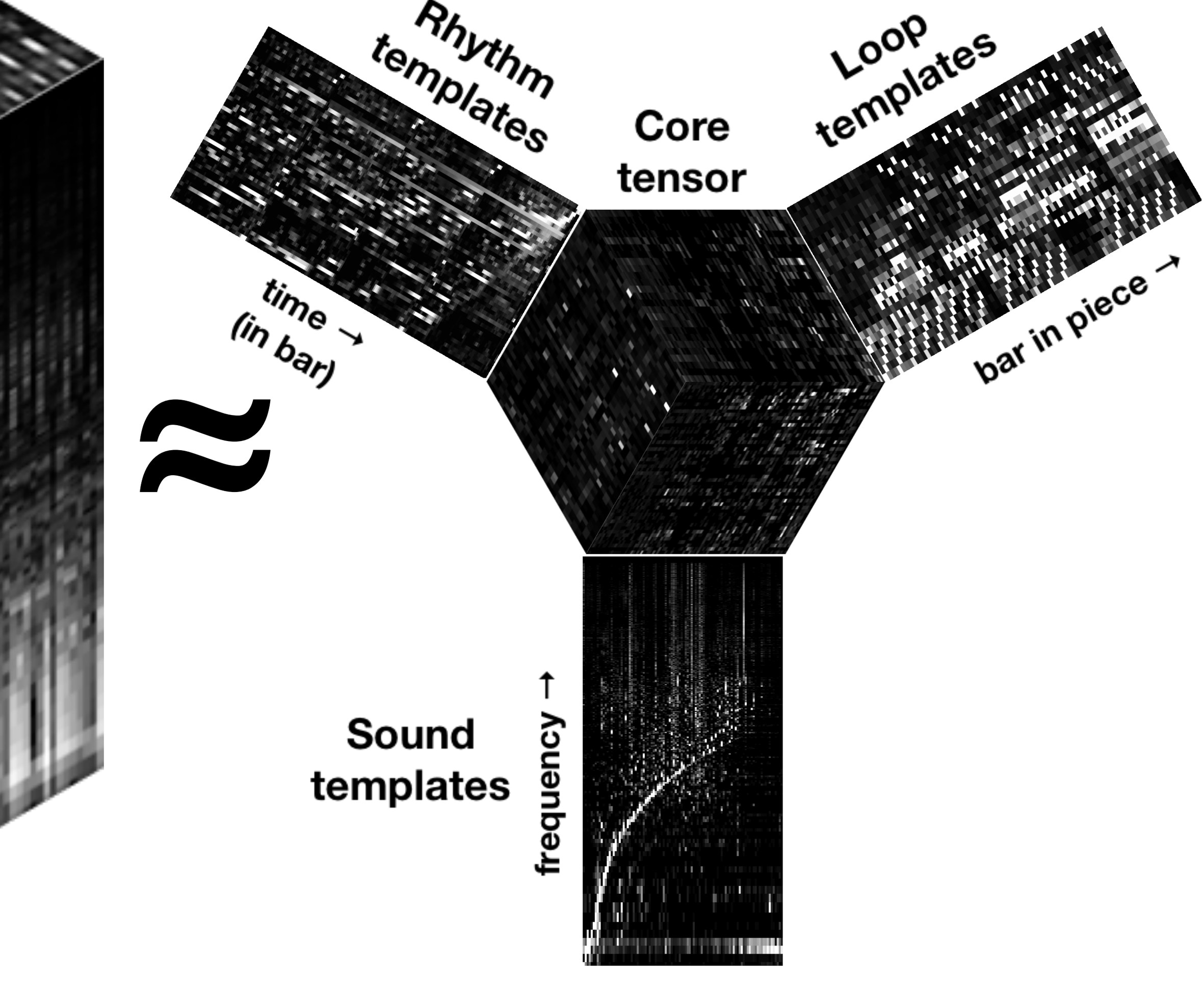


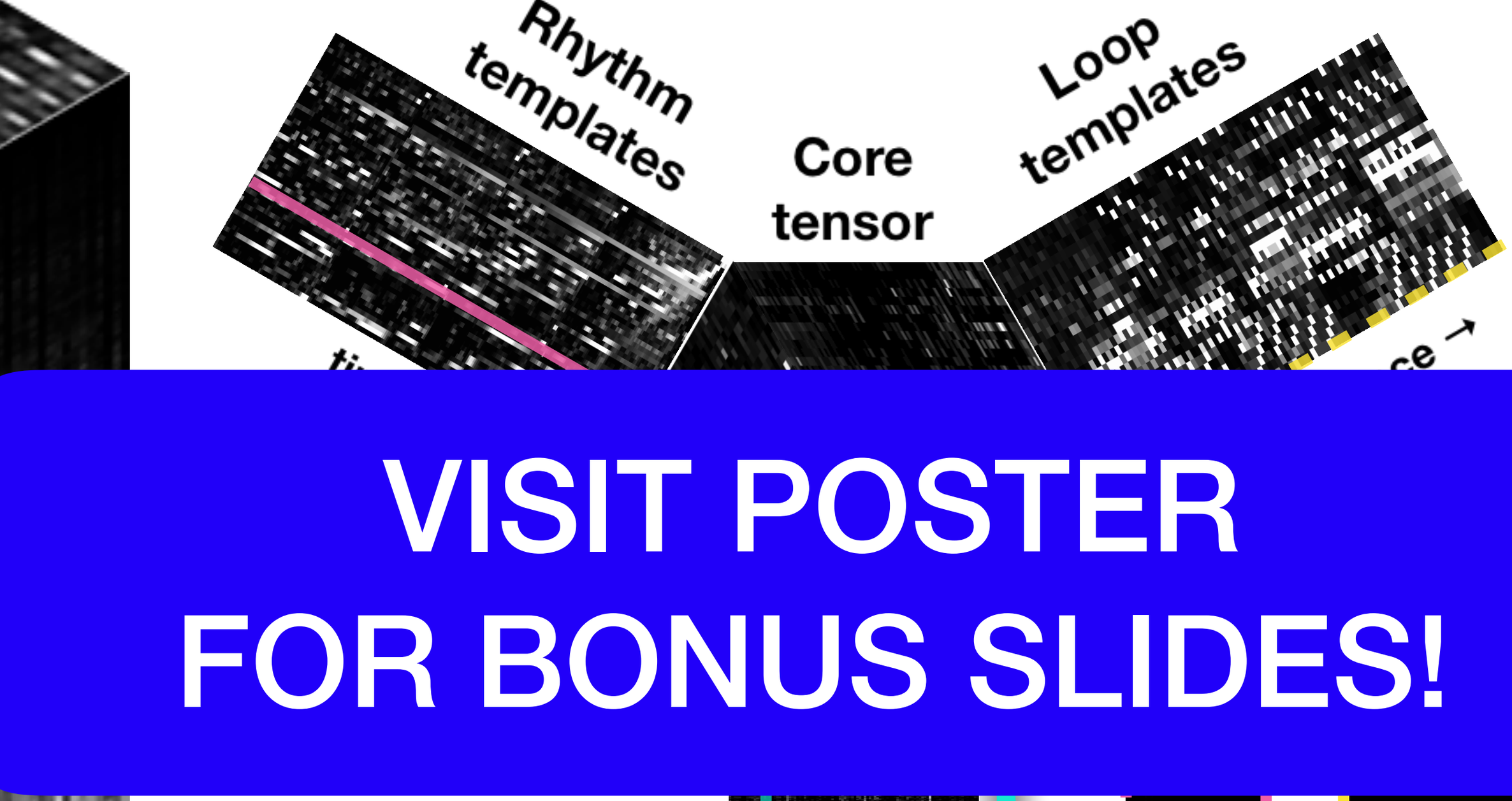
Spectral cube

~20M elements

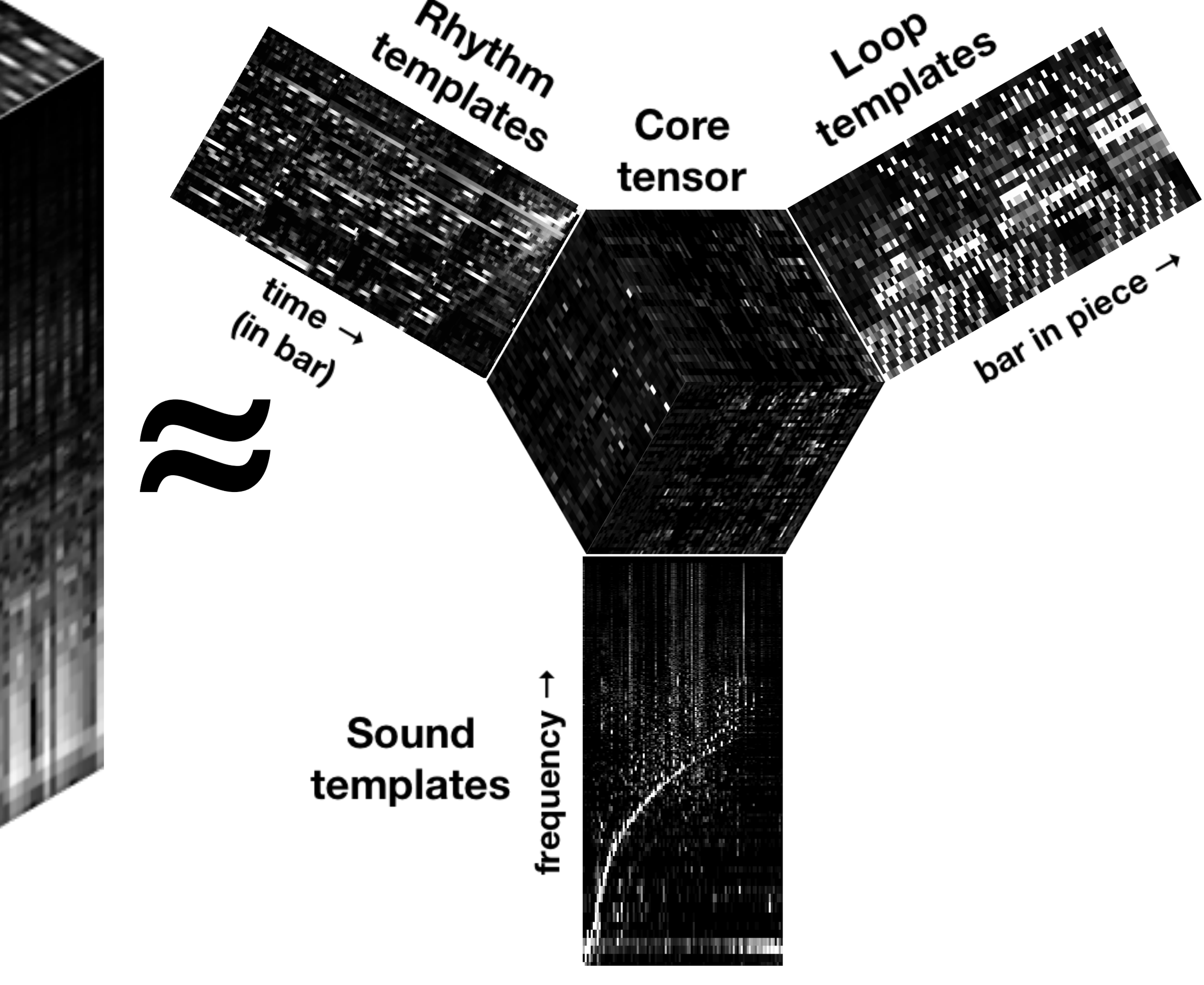
Core tensor and 3 templates

~0.2M elements

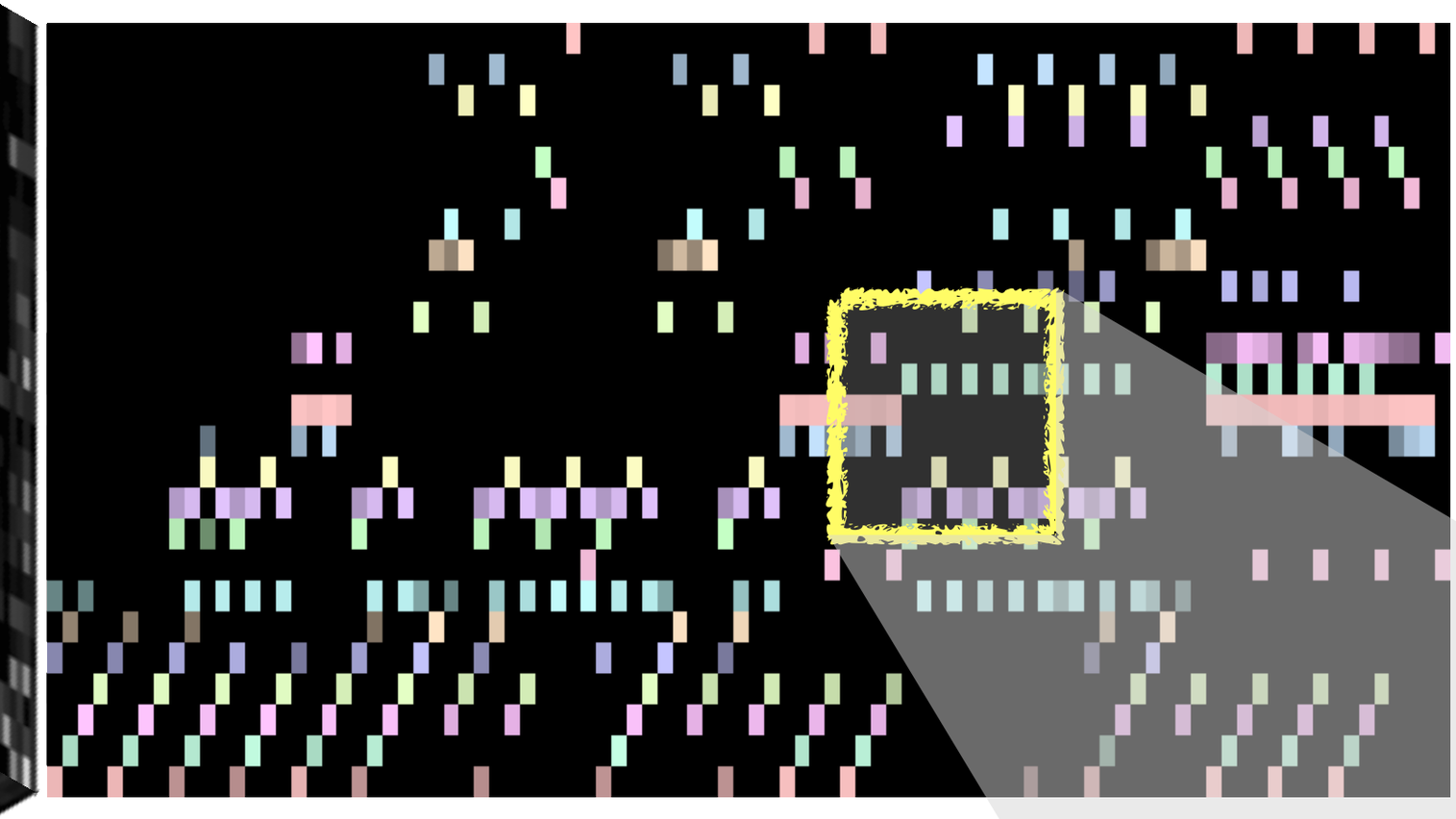

Rhythm templates
Core tensor
Loop templates
time →
(in bar)
bar in piece →
Sound templates
frequency →



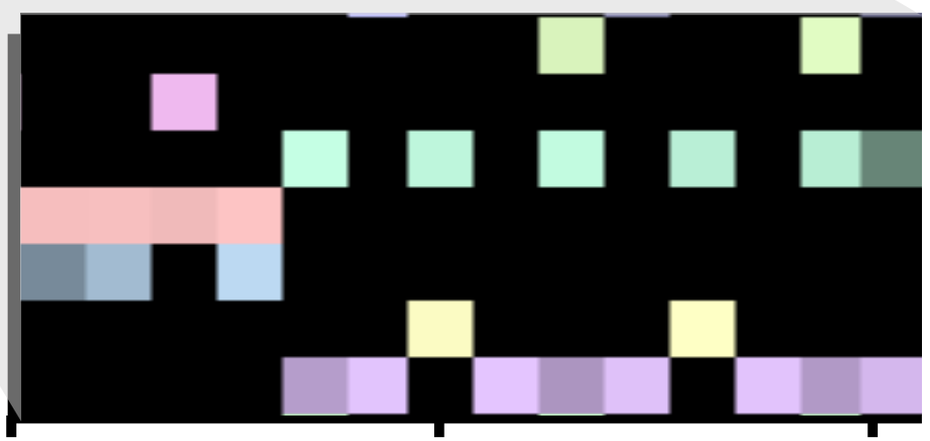
VISIT POSTER FOR BONUS SLIDES!


Rhythm templates
Core tensor
Loop templates
time → (in bar)
bar in piece →
Sound templates
frequency →

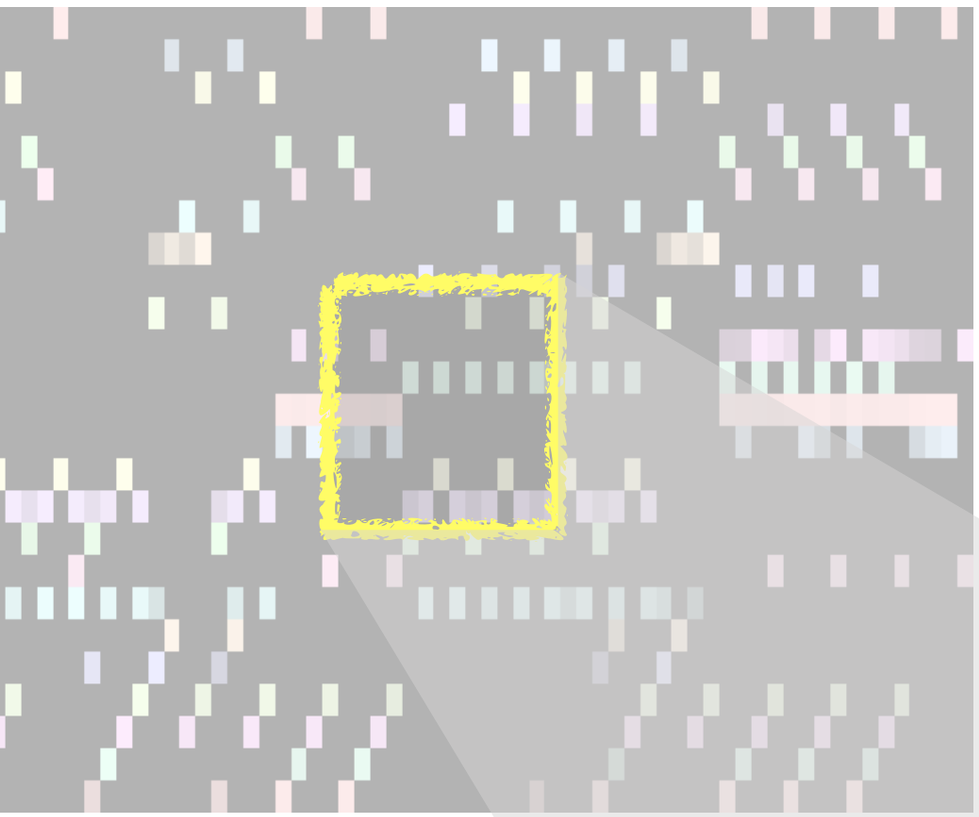
Loop templates



bar in piece →



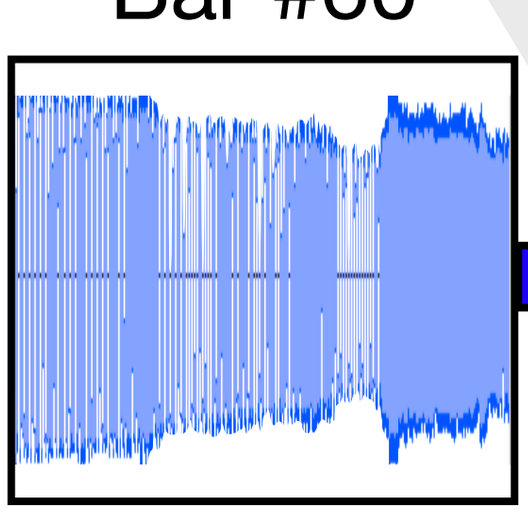
Bar 52 Bar 58 Bar 65

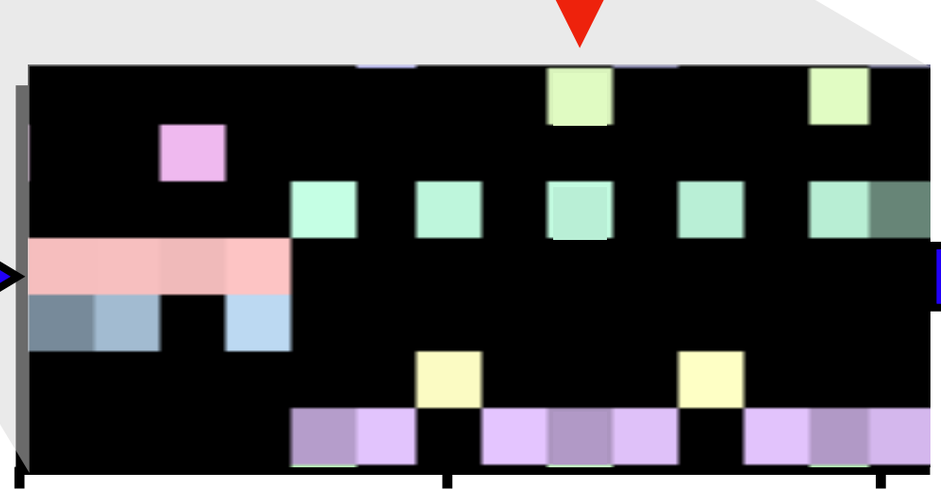


Song: “Doin’ it Right” by Daft Punk

Bar #60

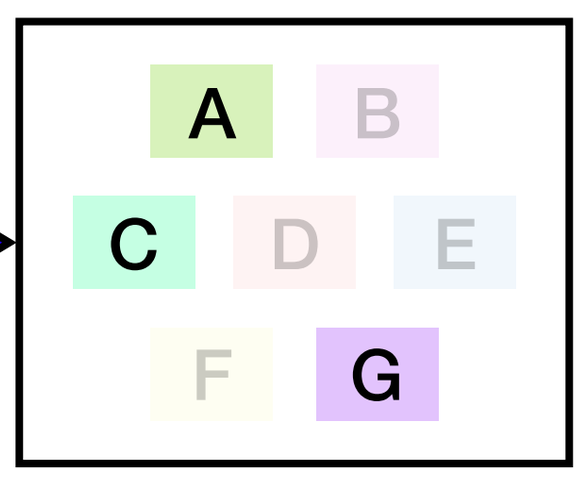
Bar #60





Bar 52 Bar 58 Bar 65

Ingredients for bar #60:

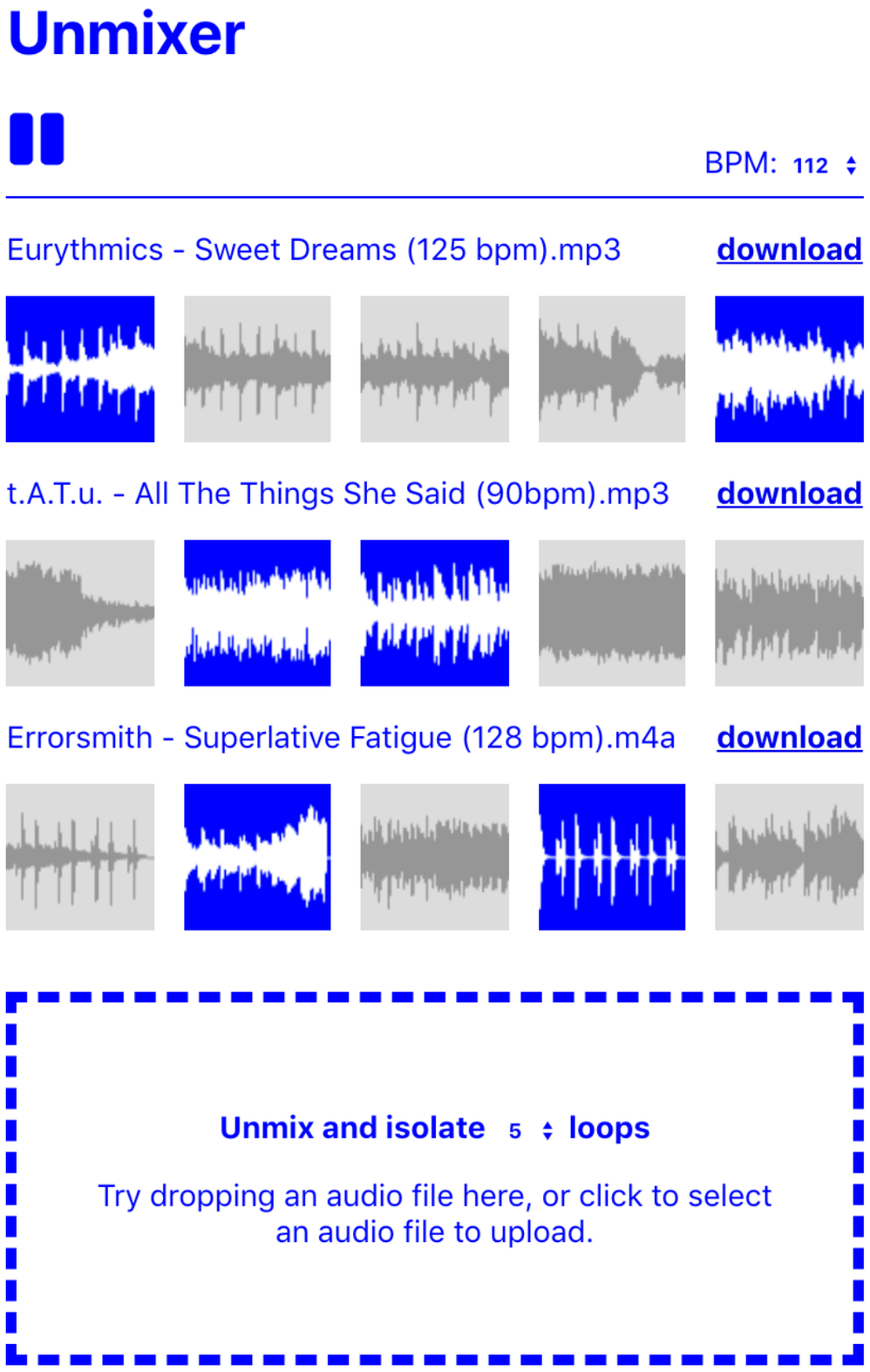


https://unmixer.ongaaccel.jp

Thank you!

At our poster:

- learn how loops are selected;
- learn how “recipes” made more independent;
- use the interface!



This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.