你的 code 充滿時間：
使用 Python 進入聲音訊號的世界

蘇黎 副研究員

中央研究院資訊科學研究所

音樂與文化科技實驗室
Music&Culture Technology Lab

引言：關於時間

Time is what we want most, but what we use worst.

-- William Penn (1644-1718)

Powered by Canva

- 從資料科學的角度來看，時間是
  - hard to model 難以捉摸
  - hard to control 難以控制
  - but ubiquitous 但無所不在

MCTL

關於聲音/音樂：時間的藝術

- 使用 Python 進入音訊 (O) 音頻 (X) 處理 (audio signal processing) 的世界
- 先從讀檔開始：以鋼琴單音的單聲道 .wav 檔為例

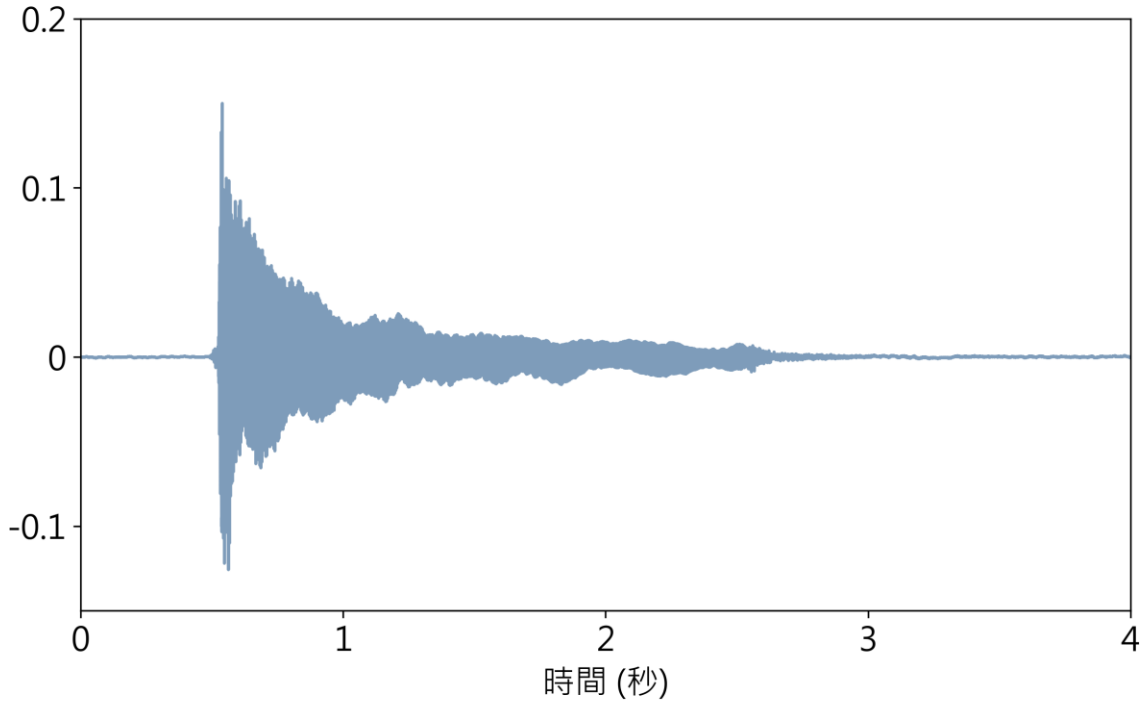
```
>>> import librosa
>>> x, sr = librosa.load('piano.wav', sr = 44100)
>>> x
array([-1.0681152e-04, -1.2207031e-04, -9.1552734e-05, ...,
       -3.3569336e-04, -3.3569336e-04, -3.3569336e-04], dtype=float32)
>>> x.shape
(182000,)
```

MCTL

觀察聲音訊號的細節

- 我個人的習慣：當分析新的訊號/資料，假如沒有看過，一定先把它畫出來看看
- 重點：採樣頻率 sampling frequency (sr) 和時間軸 (t)

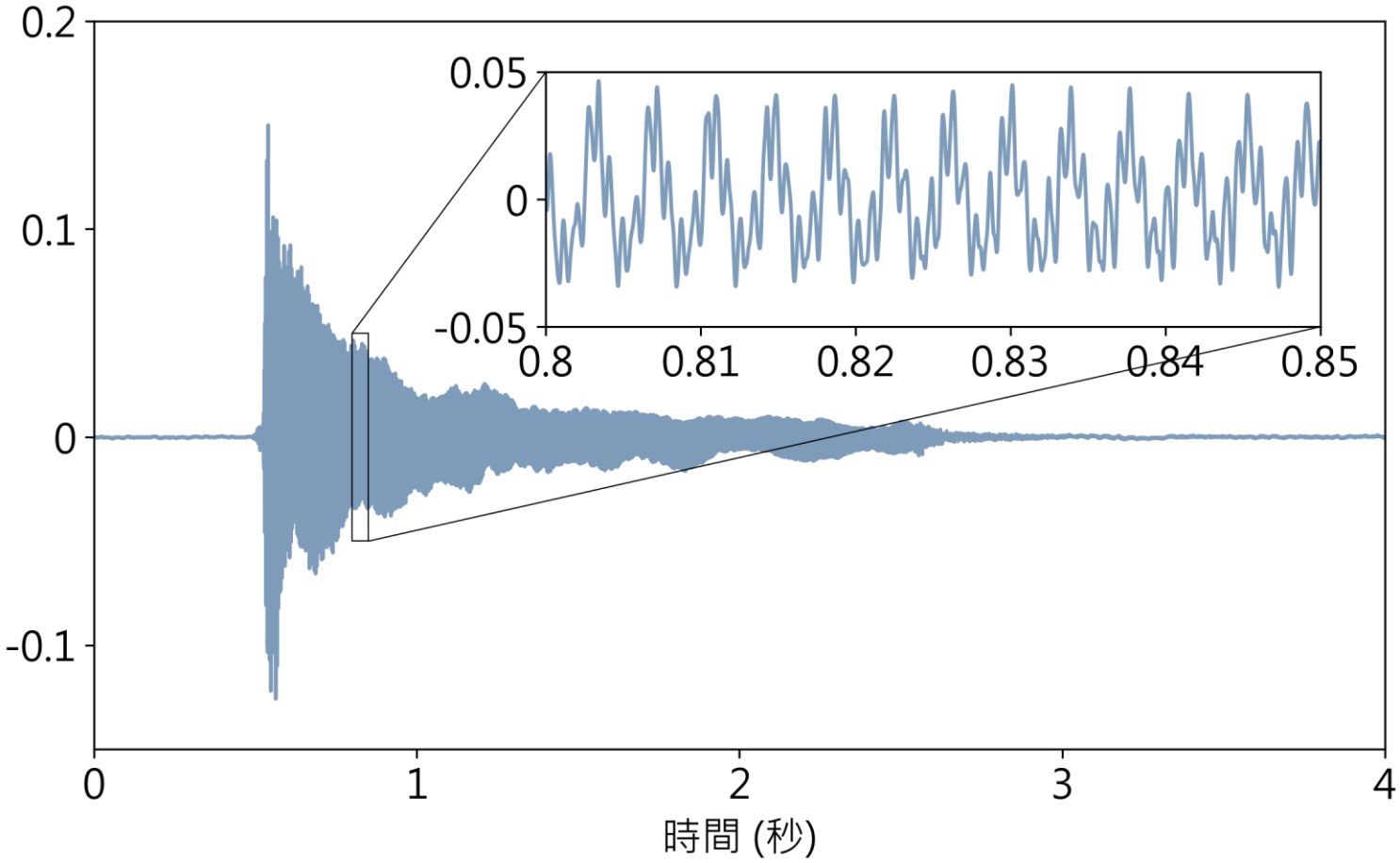
```
import matplotlib.pyplot as plt
import numpy as np
import librosa

x, sr = librosa.load('piano_60.wav', sr = 44100)
t = np.arange(1/sr, x.shape[0]/sr, 1/sr)

plt.plot(t,x)
```



MCTL

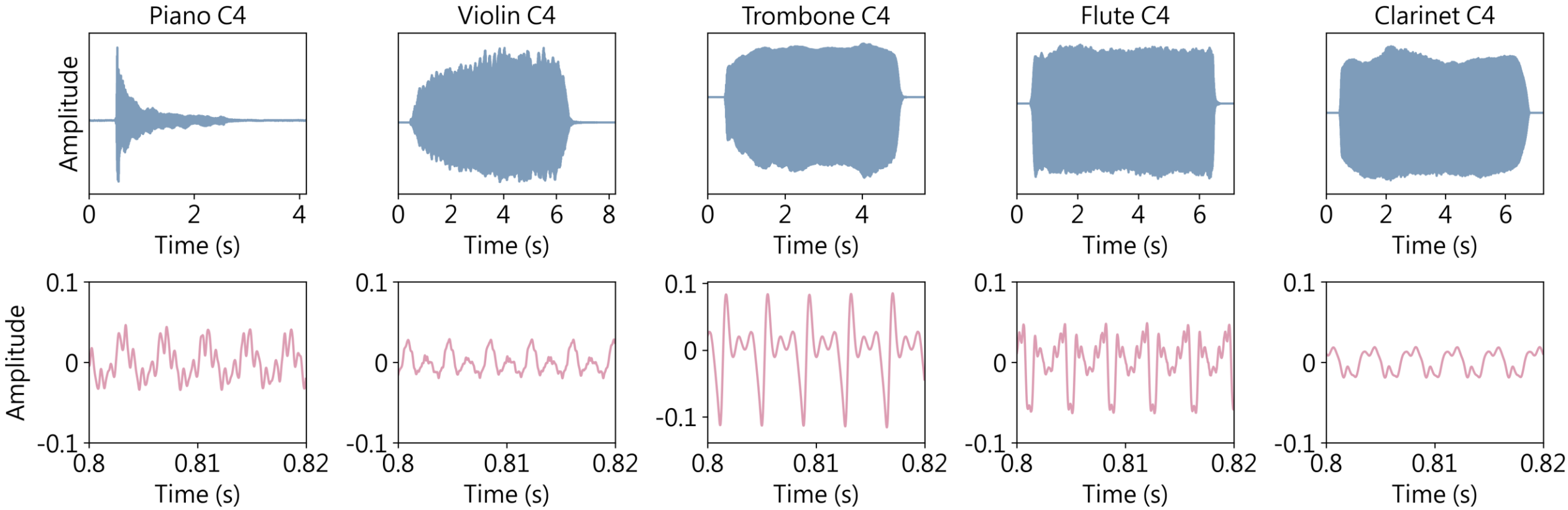
觀察聲音訊號的細節

- 這波形中有哪些特徵？振動週期的意義？



觀察聲音訊號的細節

- 相同音高 C4 (相同訊號週期)、不同的樂器



把各種訊號拿來聽聽看

- 製造隨機音樂

```
import librosa
import numpy as np
import soundfile as sf

num_of_notes = 50
x = np.zeros(0)

for i in range(num_of_notes):
    pitch = np.random.uniform(low=200, high=500) # Hz
    duration = np.random.uniform(low=0.1, high=0.5) # sec
    note = librosa.tone(pitch, duration=duration)
    x = np.append(x, note)

sf.write('random_music.wav', x, samplerate = 22050)
```

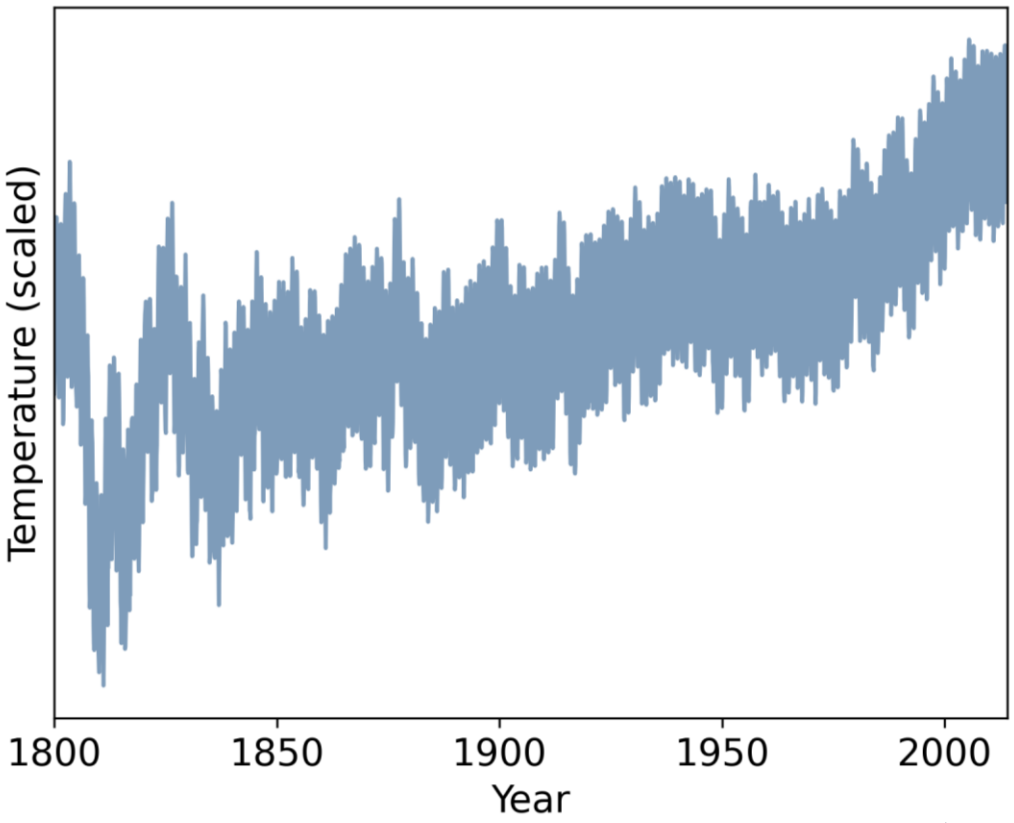
"Infinite Monkey Music" – powered by Canva

MCTL

把各種訊號拿來聽聽看

- 資料聲音化 (sonification)

```
import mir_eval
import numpy as np
import soundfile as sf

# (given the time series of the global temperature)
months = np.linspace(0, 15, len(temperature)) # 15-sec sound
# 2-year (24-month) moving average
t_avg = np.convolve(temperature, np.ones(24)/24, mode='valid')
#  a scaling emphasis the temperature increment
temperature = 100*(t_avg-5) + 10*np.array(temperature[12:-11])

sr = 22050 # sampling rate
x = mir_eval.sonify.pitch_contour(months, temperature, sr)
sf.write('sonified_temperature.wav', x, sr)
```



MCTL

頻譜 spectrum

Mathematics compares the most diverse phenomena and discovers the secret analogies that unite them.

Joseph Fourier (1768 - 1830)

Source: wikipedia

- 訊號在不同頻率 (frequency) 上的分布
- 傅立葉變換 (Fourier transform): 所有的訊號都可以表示成不同頻率正/餘弦波的疊加

$$F(\omega) \coloneqq \int_{-\infty}^{\infty} f(t) e^{-j\omega t} \mathrm{d}t$$

MCTL

頻譜的演算法

Be approximately right rather than exactly wrong.

John Tukey (1915 - 2000)

Source: wikipedia

- 數位訊號的傅立葉變換計算方式：

$$X_k := \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}} = \sum_{n=0}^{N-1} x_n \left[\cos\left(\frac{2\pi}{N} kn\right) - j\sin\left(\frac{2\pi}{N} kn\right)\right]$$

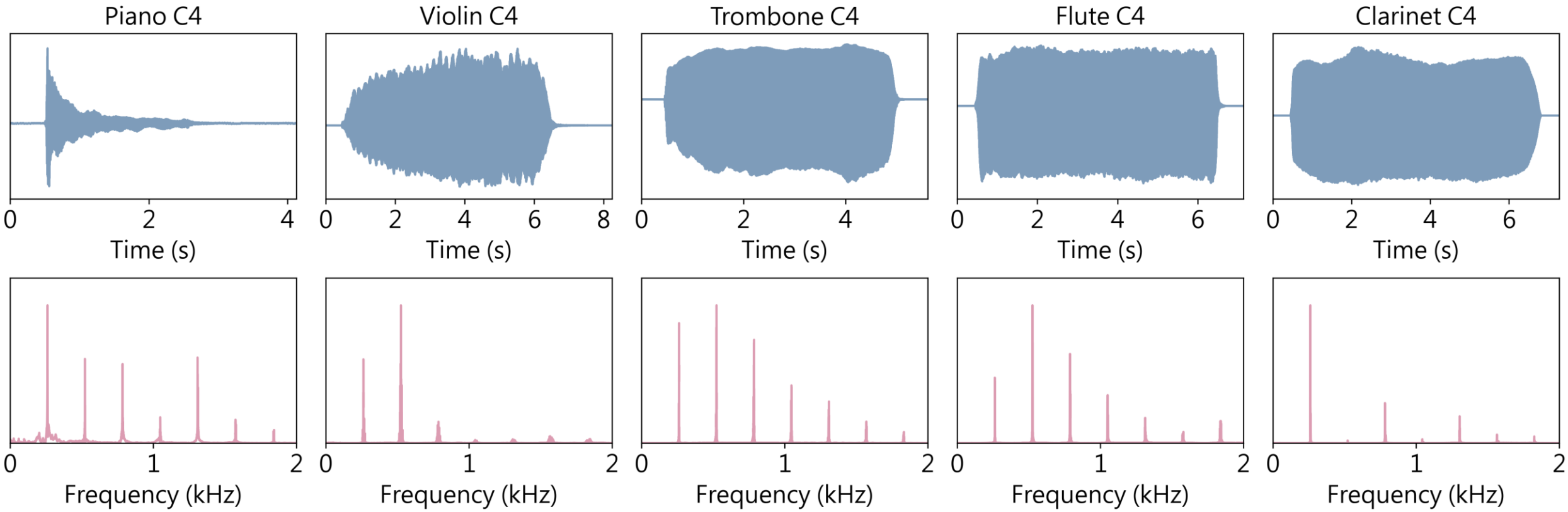
- 快速傅立葉變換 (Fast Fourier Transform, FFT): 最常見的頻譜演算法

```
scipy.fft.fft()
```

MCTL

觀察聲音訊號的頻譜

- The characteristics of different instruments



時頻圖 (spectrogram)

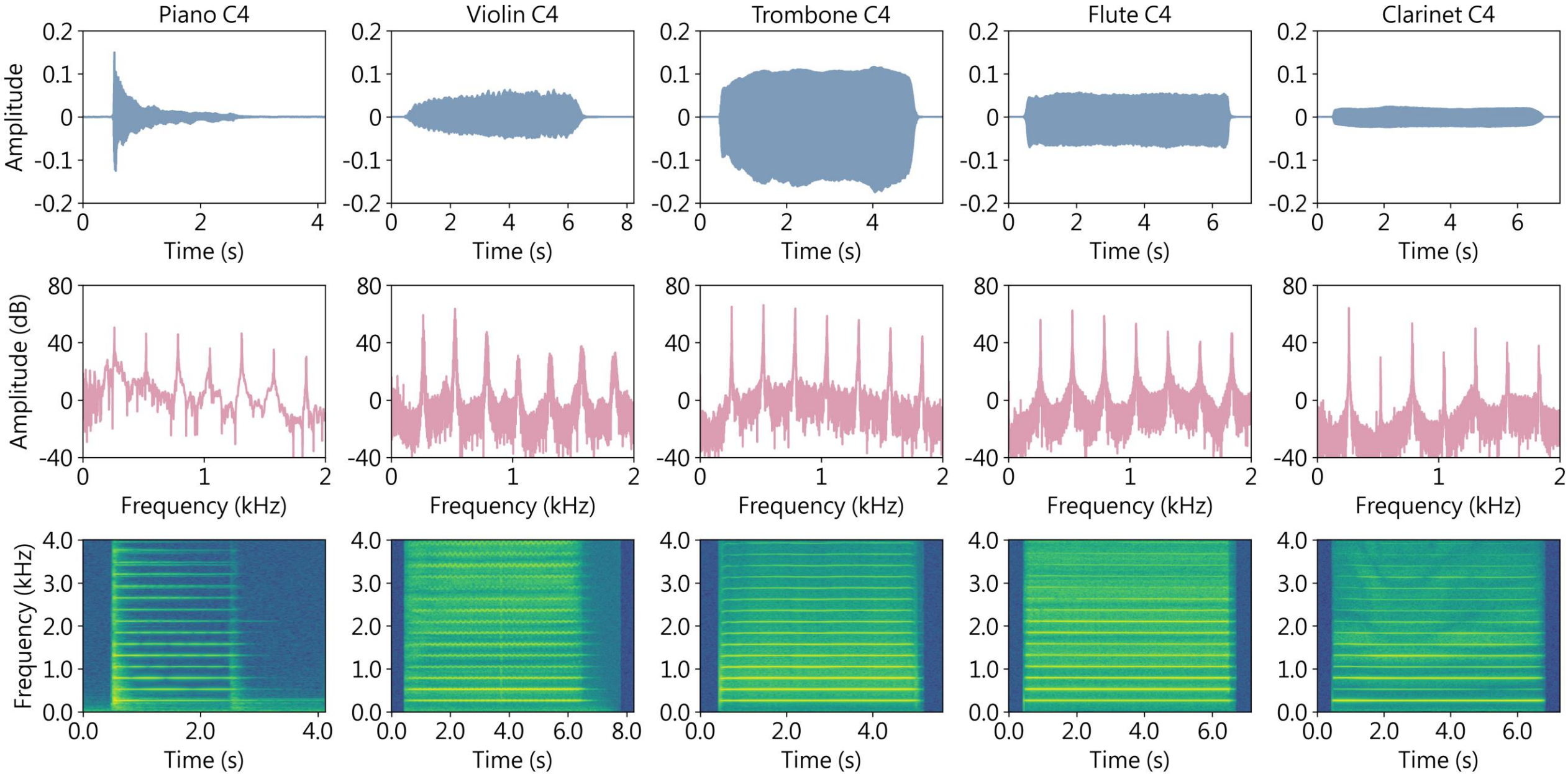
- Dennis Gabor (1900 - 1979): 訊號由聲音粒子 (sound grains) 組成
- 時頻分析 (time-frequency analysis) 和量子力學的淵源
- 短時間傅立葉變換 (short-time Fourier transform)

$$X[k,q] = \sum_{n'=\lceil -N/2 \rceil}^{\lceil N/2 \rceil - 1} x[n' + qH]w[n']e^{\frac{-j2\pi kn'}{N}}$$

Source: wikipedia

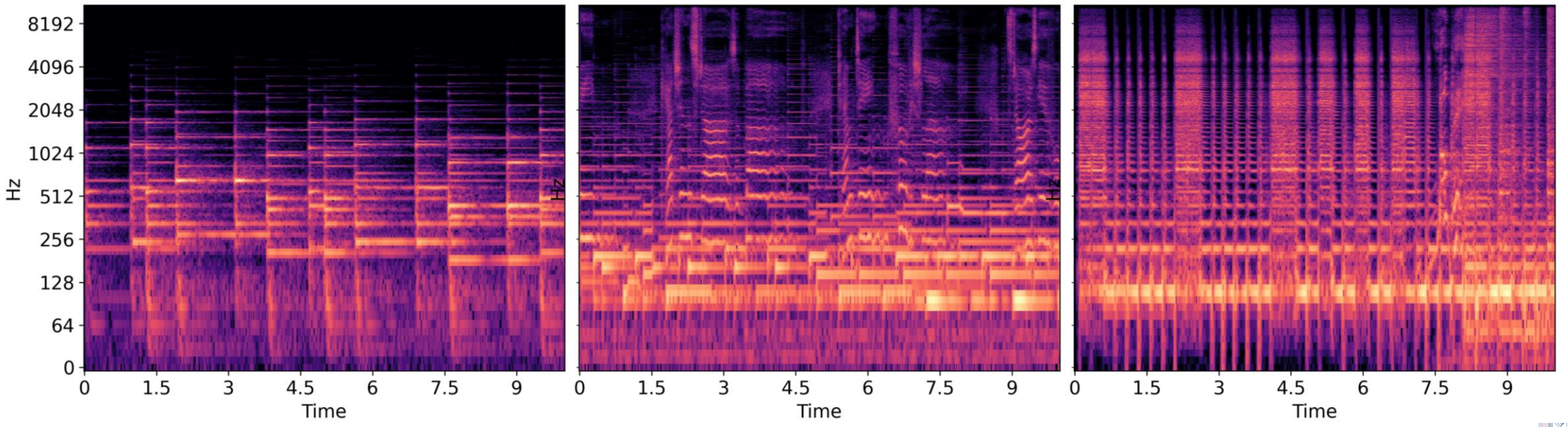
```
import librosa
import numpy as np

x1, sr = librosa.load('piano.wav')
X1 = librosa.stft(x1)
spectrogram = np.abs(X1)
```

MCTL


Piano C4
Violin C4
Trombone C4
Flute C4
Clarinet C4
Amplitude
Amplitude (dB)
Frequency (kHz)
Time (s)

從時頻圖猜歌

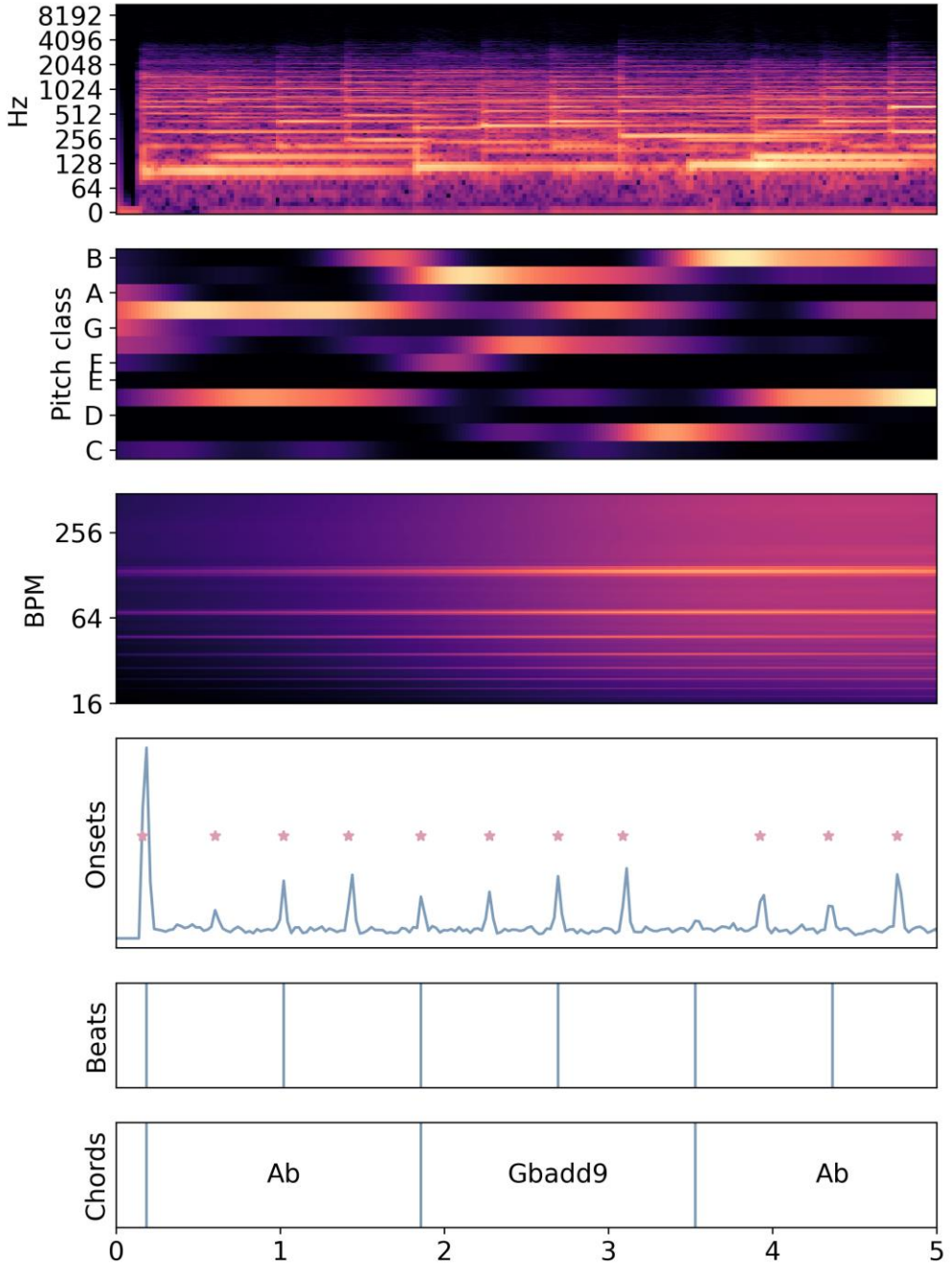
- 1) 鋼琴獨奏 2) 人聲 3) 電吉他



MIR: 用 python 懂音樂！

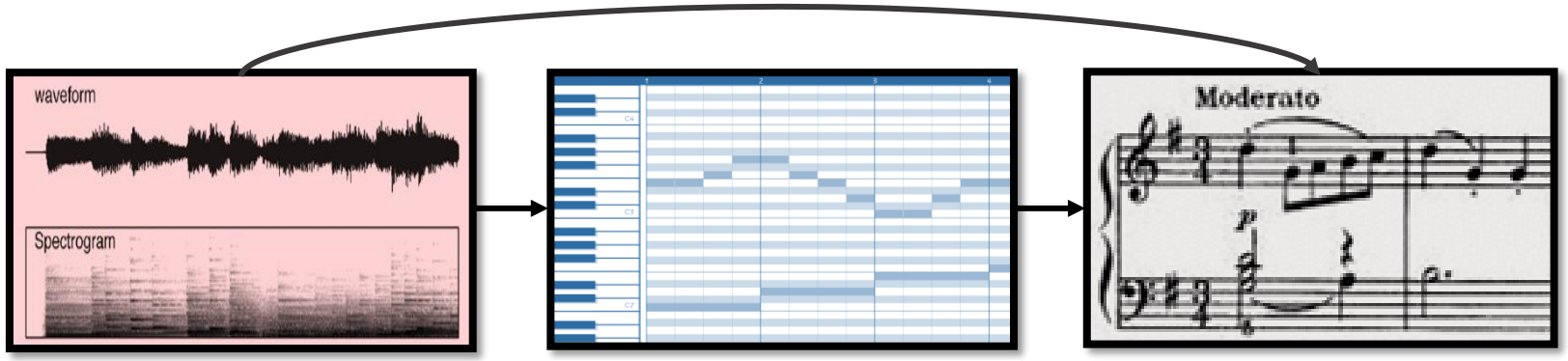
- 音樂資訊檢索 (music information retrieval, MIR) 成為 21 世紀的新興領域
  - 使用 pYIN 偵測音高
  - 使用 spectral flux 偵測聲音事件
  - 使用 tempogram 做節拍的偵測
  - 使用 chromagram 做和弦辨識

```
y, sr = librosa.load('piano.wav')
# spectrogram, chromagram, onset, tempogram, chord, beats
spectrogram = librosa.stft(y)
chroma = librosa.feature.chroma_cens(y=y, sr=sr)
tempo, beats = librosa.beat.beat_track(y=y, sr=sr, start_bpm = 70)
tempogram = librosa.feature.tempogram(y=y, sr=sr)
onset_strength = librosa.onset.onset_strength(y=y, sr=sr)
onsets = librosa.onset.onset_detect(y=y, sr=sr)
onsets = librosa.frames_to_time(onsets, sr=sr)
y_beats = librosa.clicks(frames=beats, sr=sr)
```



音樂採譯問題 automatic music transcription

- 莫札特的故事：music transcription = music piracy?
- 跨模態 (cross-modal) 資訊處理問題：audio-MIDI-score



音訊 (audio) 層面: 被演奏出的時域 (time-domain) 訊號或時頻圖

符號 (symbolic) 層面: 起音點 (onset)、結束點 (offset)、每個音的音高 (pitch) 和聲部 (如旋律、配器) 資訊

樂譜 (score) 層面: 調性與調式、拍號與小節線、力度 (dynamics)、表情 (expressivity) 等所有重要的東西

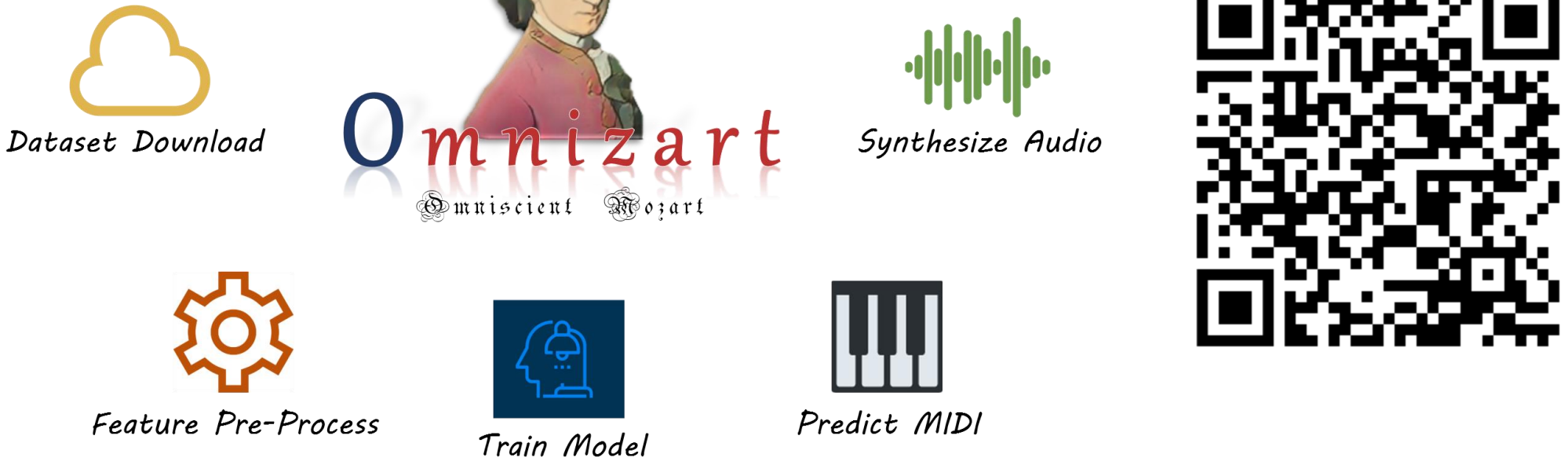
(尚有) 其他高階資訊: 結構 (structure), 曲風 (genre), 風格 (style), …

Wolfgang Amadeus Mozart
(1756 - 1791)
Source: wikipedia

MCTL

The omnizart project

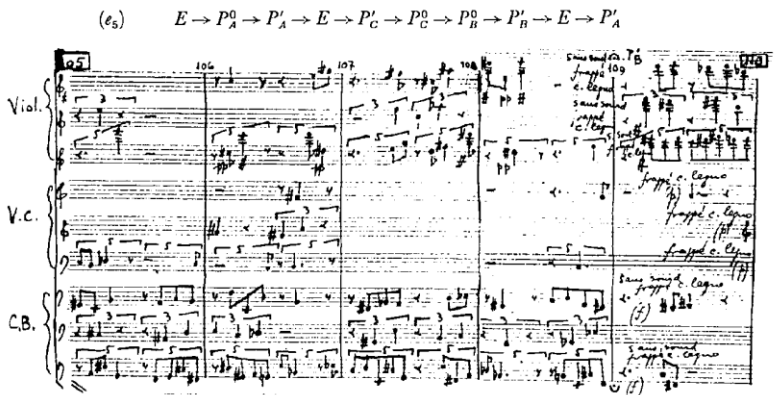
- Omnizart is a python-based library for automatic music transcription!
- Main credit: **Yoyo**



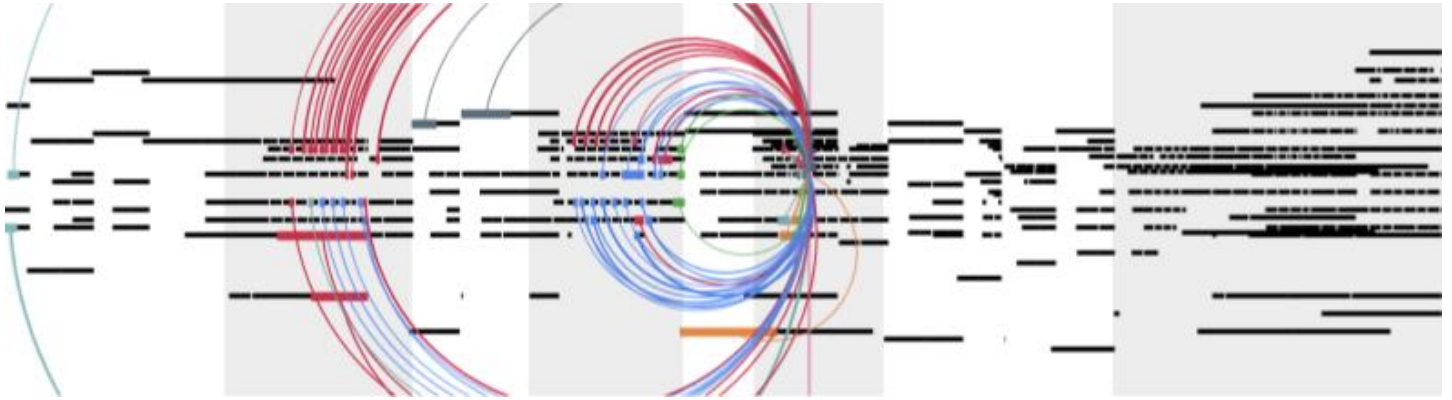
MCTL

音樂生成問題 music generation

- 隨機音樂：風鈴、莫札特骰子遊戲 (Musikalisches Würfelspiel)
- 機率音樂：Iannis Xenakis 使用「聲音粒子」的馬可夫鏈 (Markov Chain) 創作 (1959)
- 數據音樂：Google Magenta's Music Transformer 預測「下一個音樂 token」(2017)



Iannis Xenakis' *Analogues A*

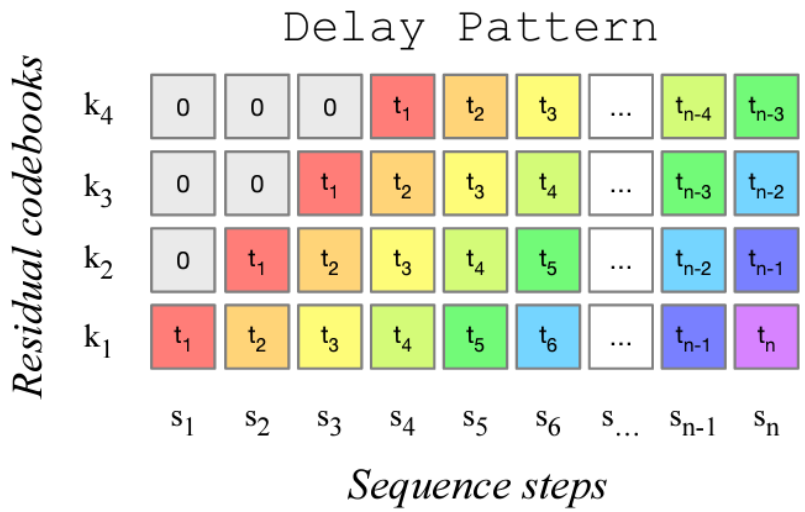


Music Transformer

MCTL

目前的發展

- 這首歌還算好聽吧！SUNO 幫我生的
  - Text-to-music generation
  - Large Language Model (LLM)
  - VAE with Residual Vector Quantization (RVQ)
  - Neural Codec
- 在聲音層次上，這樣的技術成就其實是整合/延伸了所有前面提的概念，逐漸發展來的
  - Sound grains -> sound tokens
  - Markov chain (short-term dependency) -> Transformer (long-term dependency)
  - Random -> large-scale training data



Codebook patterns in Meta's MusicGen model (2023)

MCTL

當代 AI：衝擊與焦慮

- 人類對新科技的焦慮已經很久了
  - 三種焦慮的面向

在像我們這樣的時代，當工程師的天才達到如此令人難以想像的程度時，我們能夠像買一杯啤酒一樣輕鬆地聆聽名曲。當任何人都可以隨意喚醒這被保留在唱片中的魔力，我們難道不應該擔憂這種「對聲音的馴化 (domestication du son) 」嗎？這難道不意味著，一向被認為堅不可摧的藝術之神秘力量，終將消失嗎？

Claude Debussy in *La Revue S. I. M.* (1913)

Claude Debussy (1862 - 1918)

Source: wikipedia

MCTL

當代 AI：衝擊與焦慮

- Marvin Minsky 的「預言」

The problem with music theorists is that they generate papers (theories) only once every five years or so, when they should be concentrating on intelligent systems that can come up with a theory every five minutes.

Marvin Minsky (1927 - 2016)

Marvin Minsky

Source: wikipedia

MCTL

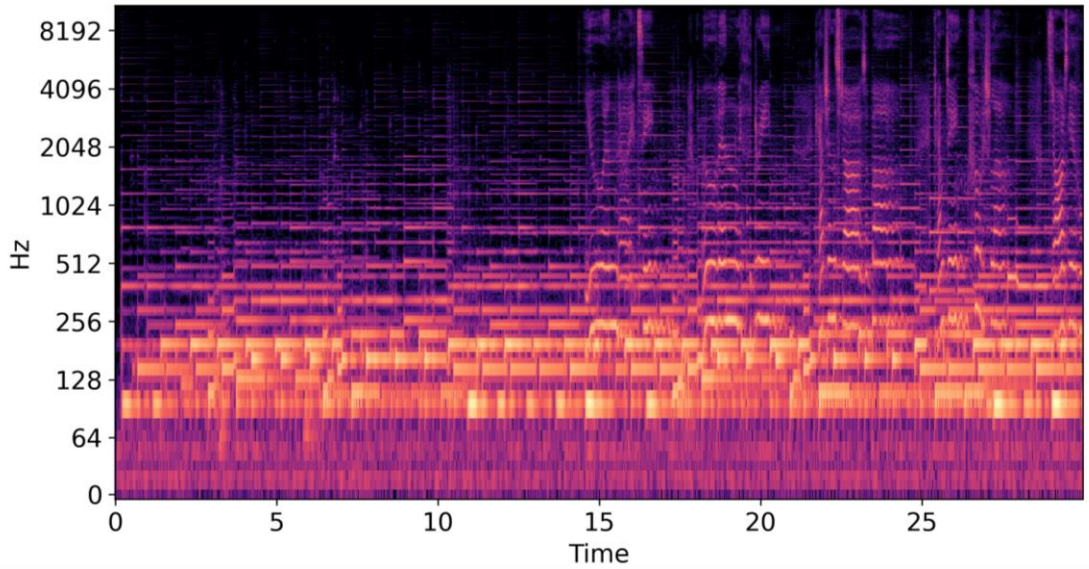
問題與限制

- 會不會 AI 音樂家和人類音樂家其實怕同一件事

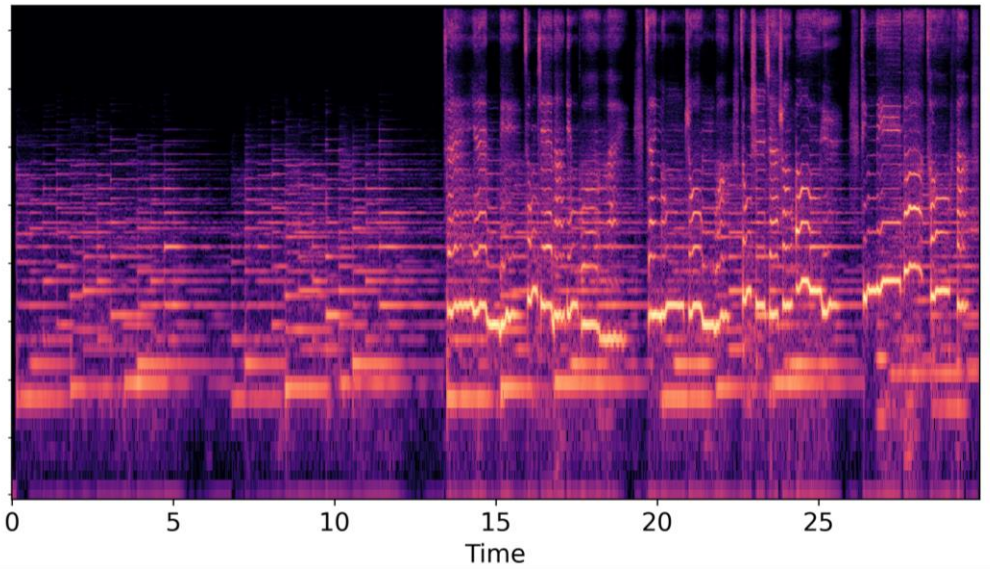
問題與限制

- 音樂生成：時頻圖的比較

Real-world



Generated



MCTL

結語

- 「 不要輕易得出任何一代不如一代的結論，但我們必須注意那已經吞噬了許多美好事物的機器。如果我想滿足這頭怪獸，就把舊的曲目丟給它吧！」

- 「為什麼他們不明白，我們身後已有這麼多世紀的音樂，從這一偉大的知識遺產中受益而幼稚地設法改寫歷史，真不值得。相反，我們的責任難道不是找到我們這個時代所需要的，進步、勇敢和勝利所需要的交響樂公式嗎？……不要讓藝術的捍衛者排在研究人員隊伍的最後而裹足不前；願他們不會被天才的工程師拋在後面。」

Claude Debussy in *La Revue S. I. M.* (1913)

MCTL


MCTL