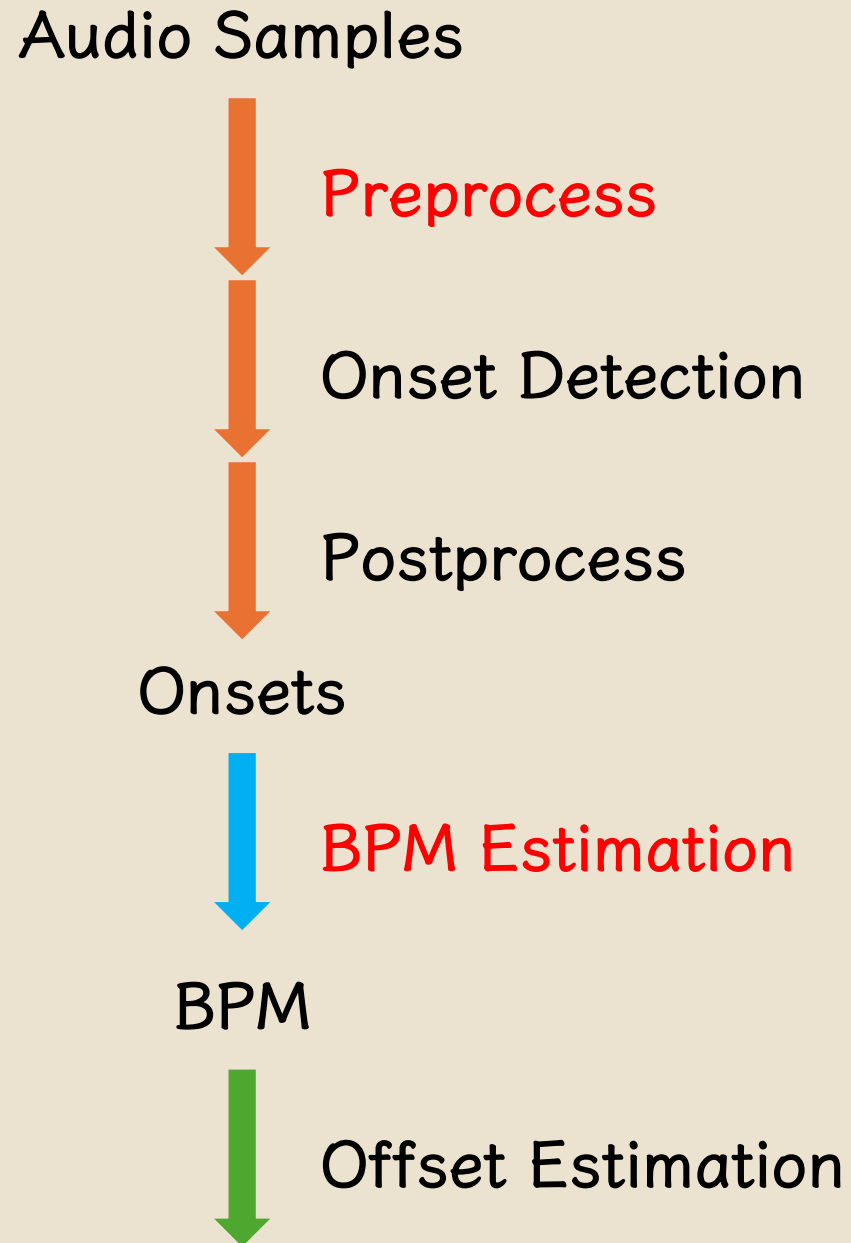


- CS Project Progress Report -

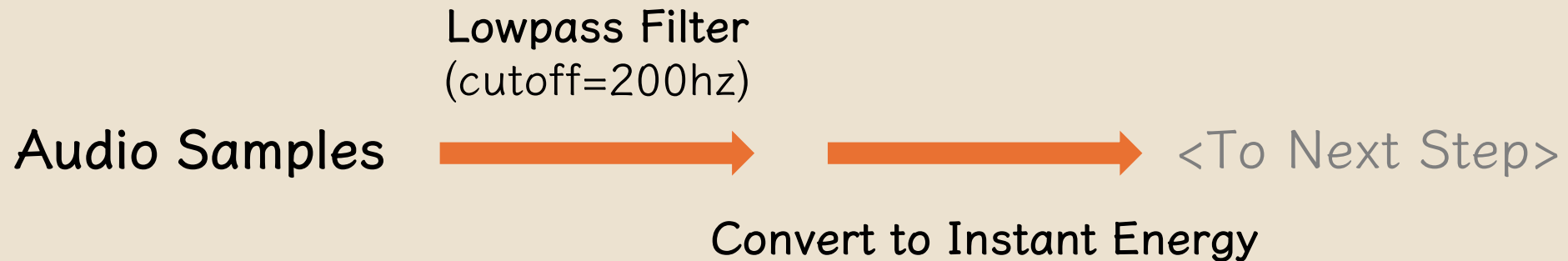
111550037 嚴偉哲

Process



Preprocess – Lowpass Filtering

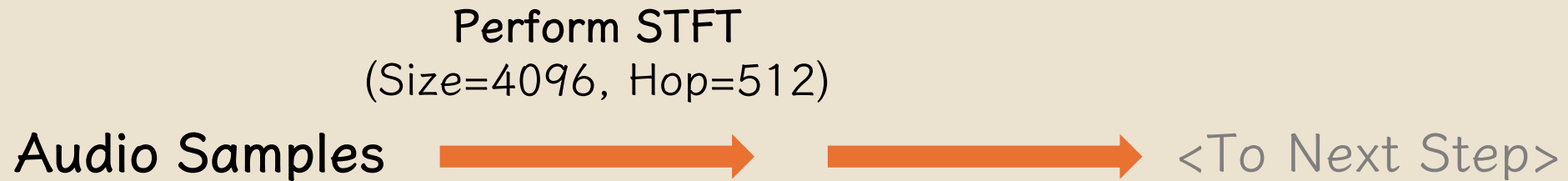
The feeling of beats is mainly made by **drum sounds** (Low Frequency)



$$E[i] = \frac{1}{512} \sum_{k=512i}^{512(i+1)-1} (x_L[k]^2 + x_R[k]^2)$$

Preprocess – Weighted Energy

Make the energy biased towards the **high frequency** content

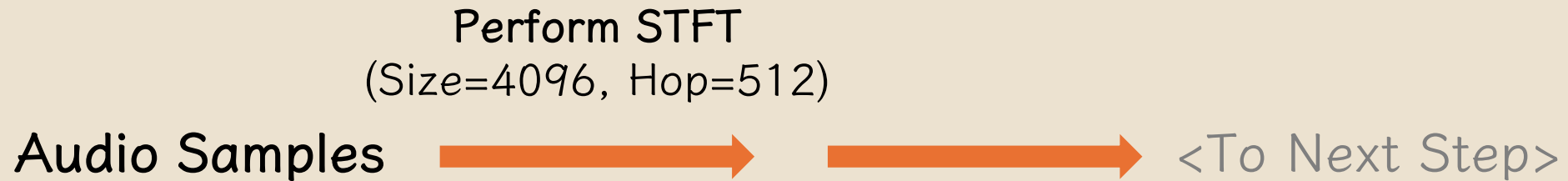


Convert to Weighted Energy

$$E[k] = \sum_{\mu=2}^{2048} (x[k, \mu]^2 \cdot \mu)$$

Preprocess – Percussivity

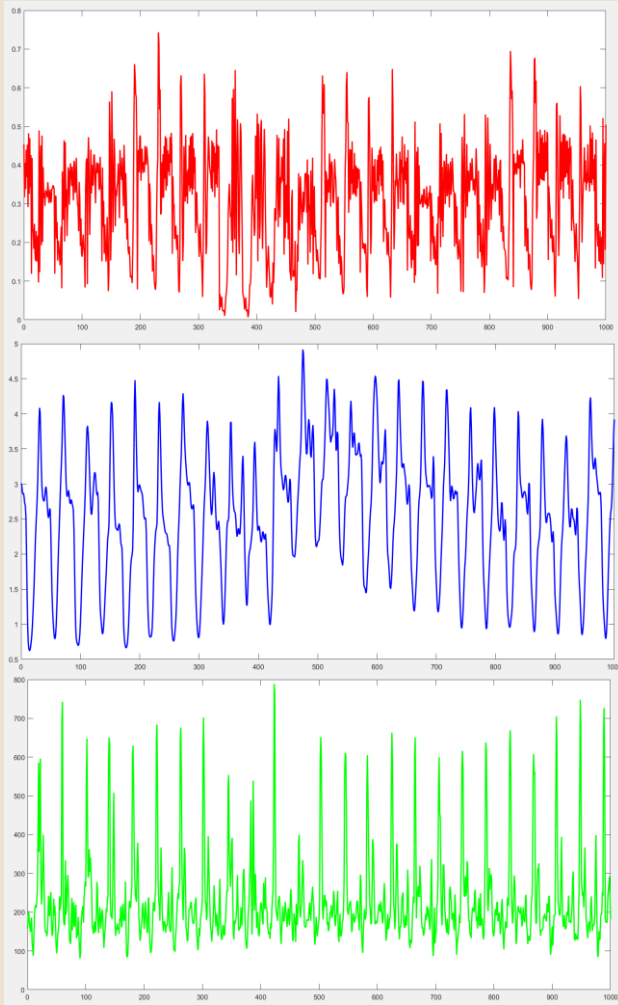
Percussion instruments have spectrum with rapid **broadband** onset



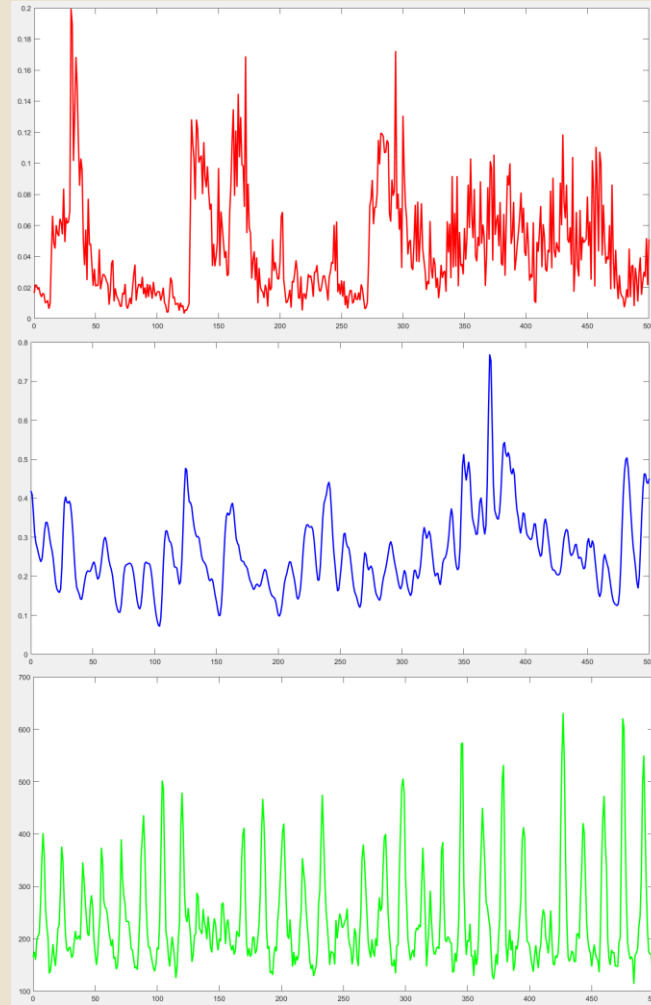
Compute Percussivity

$$P[k] = \sum_{\mu=20}^{2048} \begin{cases} 1, & \text{if } \frac{|x[k, \mu]|}{|x[k-1, \mu]|} > 5 \\ 0, & \text{otherwise} \end{cases}$$

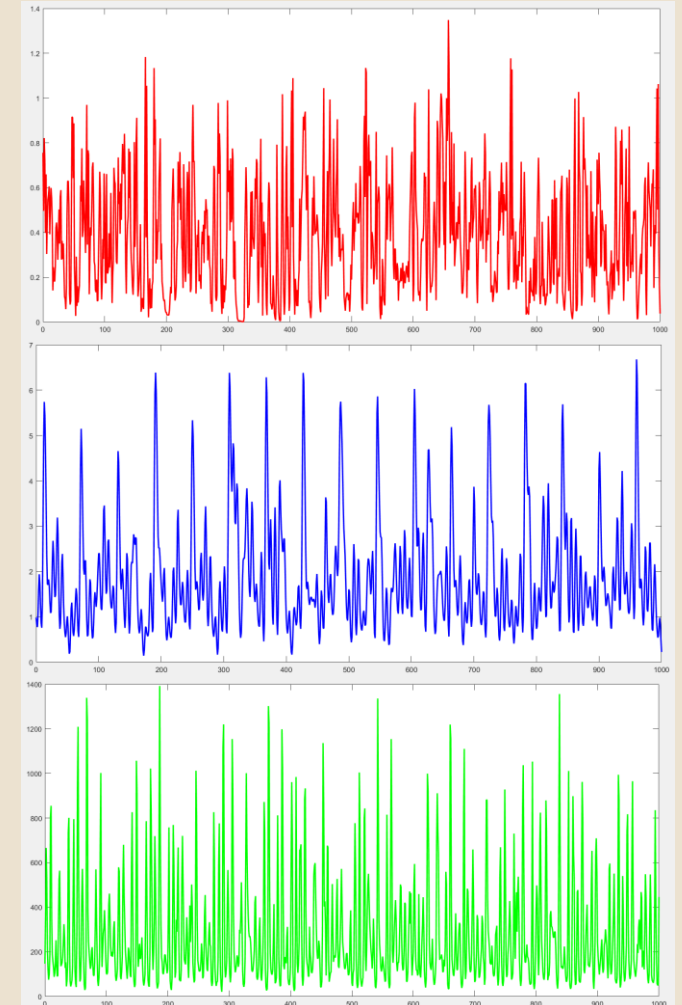
Preprocess – Comparison



Soda Sphere & PIKASONIC - Neverland



PYKAMIA - Fantasia Sonata Overdevil Concerto



Silentroom - Nhelv

Onset Detection

Use the **0.5 second** samples' average (~43 instant blocks) as the reference

Simulate how we perceive **sound intensity**



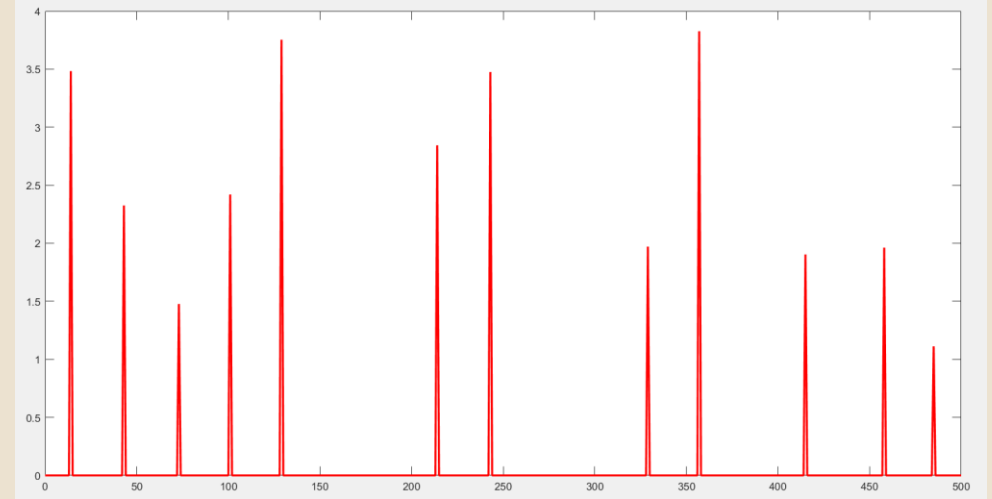
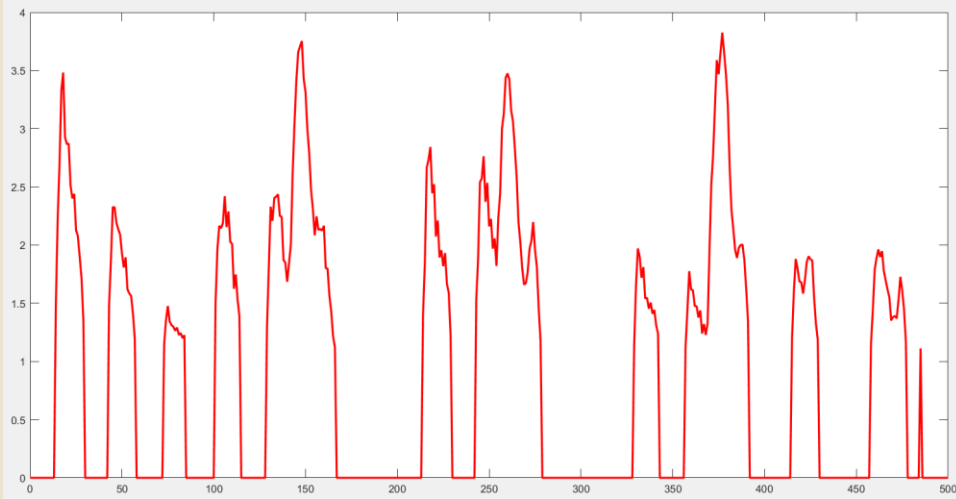
Compute Instant-to-Average
Ratio

$$R[i] = \frac{E[i]}{\frac{1}{43} \sum_{k=i-42}^i E[k]}$$

Onset Detection – Cluster Removal

Slow / soft onsets (string / riser) or **noise-like sound** (cymbals / white noise) may trigger the detector multiple times and form clusters

Set a duration for closest two onsets possible and remove all except the first one



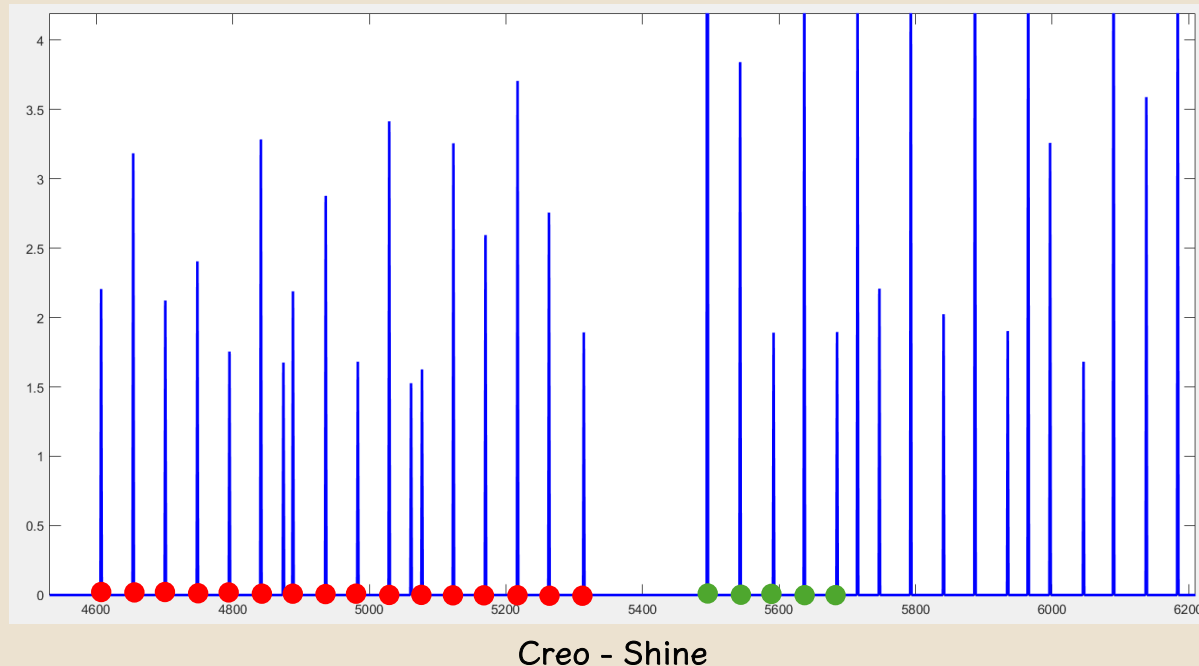
Rabbit House – Ad Astra Per Aspera
(Onset Resolution = 1/15 seconds)

BPM Detection

A comparison technique called **pulse-train** is used.

For every onsets, check the **consecutive matched onsets** with a target BPM (delay).

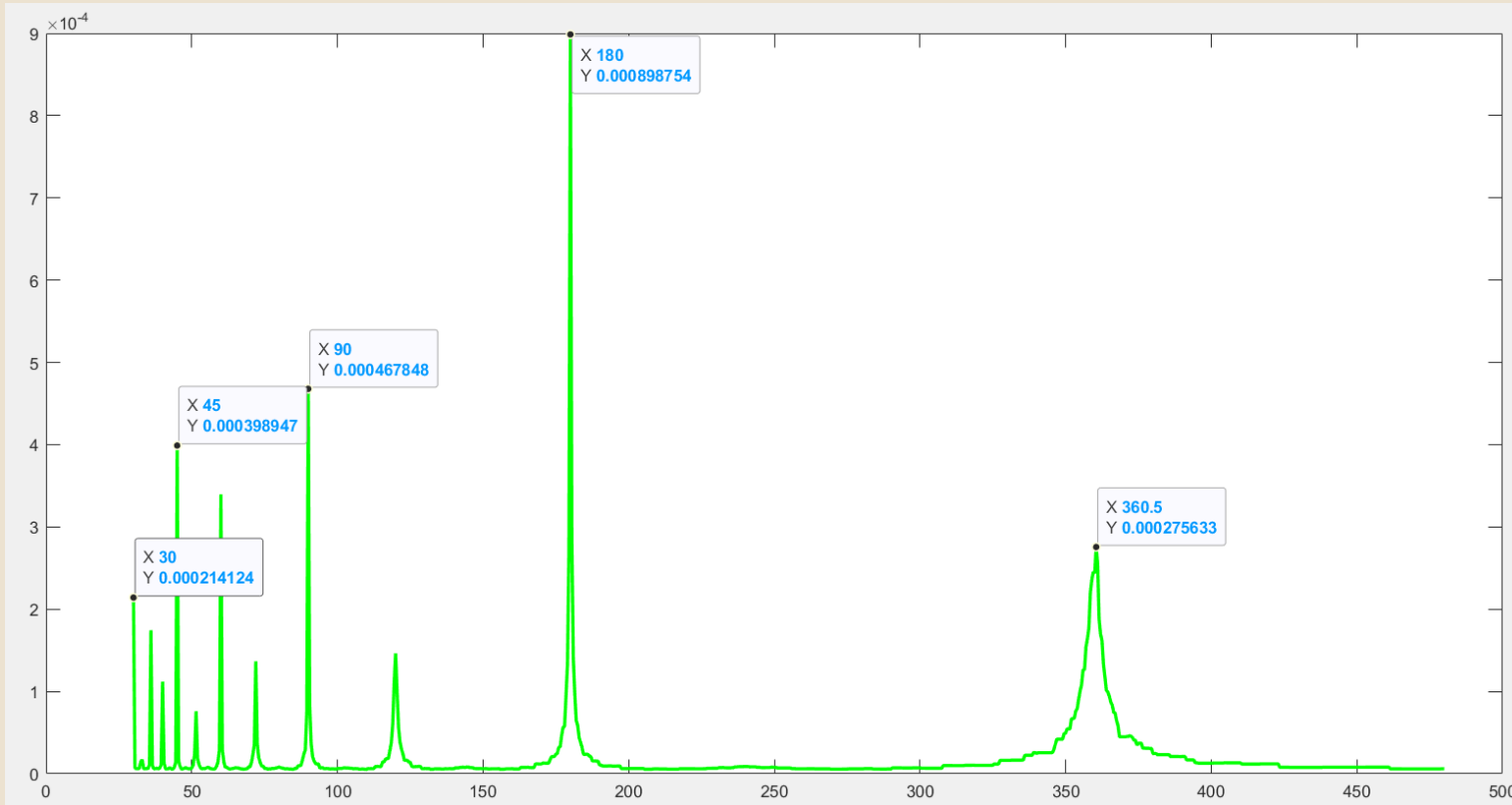
Longer onset chain would be rewarded with way higher score.



$$S(bpm) = \sum_{i=0}^n \left(\frac{\text{Length of onset chain}}{\text{Total onsets}} \right)^2$$

BPM Detection – Result

Once every BPM is tested, the one with the **highest score** is picked

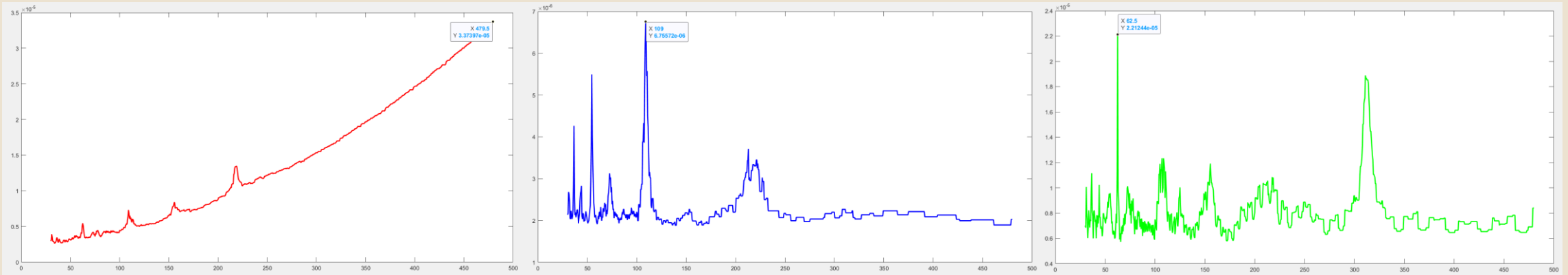


Notice that the BPM of the **multiples of 180** are also spiked up

BPM estimation for ARForest – Inverted World

BPM Detection – Comparison

Three ways of preprocessing all have good performance, but using **percussivity** has the most accurate result overall.



BPM estimation for LeaF – 弼 (125 BPM)

BPM Detection – Confidence

To know how sure the estimation is, I designed a formula to compute its confidence.

First, the **peaks** of the BPM estimation is extracted

Then, it use the **harmonics** of the **selected BPM**, and see the difference of the **harmonic BPM** and the closest **peak BPM**

Sum up the error and get the confidence score