

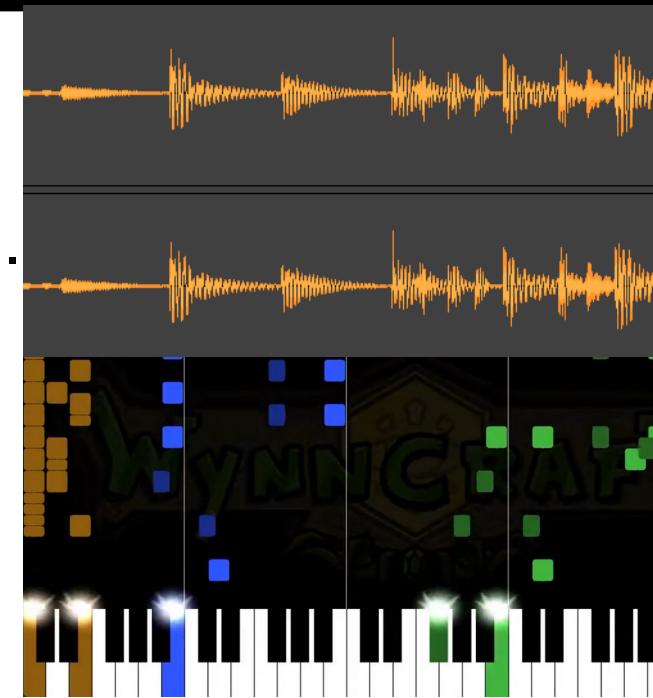
# Note Block hangminták felismerése zenefájlban

Mérnökinformatikus BSc, Önálló laboratórium



# Hangfájlok formátuma

- Wave: wav/mp3/ogg/...
- Float értékek másodpercenként 44100-szor pl.
- Midi: ~kotta
- Zenehang kezdet- és végidőpontok
- Konverzió nem egyértelmű, nem könnyű (főleg wave → midi)



# Note Block Studio

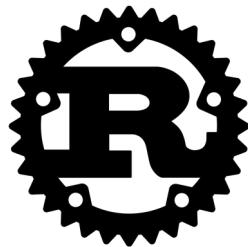


- Minecraft Note Block-ok mint minták
- Note Block Studio: szerkesztőprogram
- NBS fájlformátum
- ~Midi, de speciálisabb: egy időpillanat, konkrét minták



# Projekt célja

- Bemenet: Wave fájl
- Hangminták felismerése adott időben, hangmagasságban, hangerővel
- Kimenet: Nbs fájl
- Hasonlít a zene lekottázásra, annak speciális esete
- Megvalósítás Rust-ban



# Videó

- Note Block hangminták
- Hullámos fájl (bemenet)
- Spektrogram (konverzióhoz fontos)
- Midi vizualizáció (motiváció része)
- Lejátszás a játékban (motiváció része)
- Eredeti Nbs fájl (elvárt kimenet)
- Felismert Nbs fájl (tényleges kimenet)



File Edit Settings Help

00:00:02;556 / 00:00:06:400 5.00 t/s 1.4.1

00:00:02;000 00:00:03:000 00:00:04:000 00:00:05:000 00:00:06:000 00:00:07:000

8 12 16 20 24 28 32 36

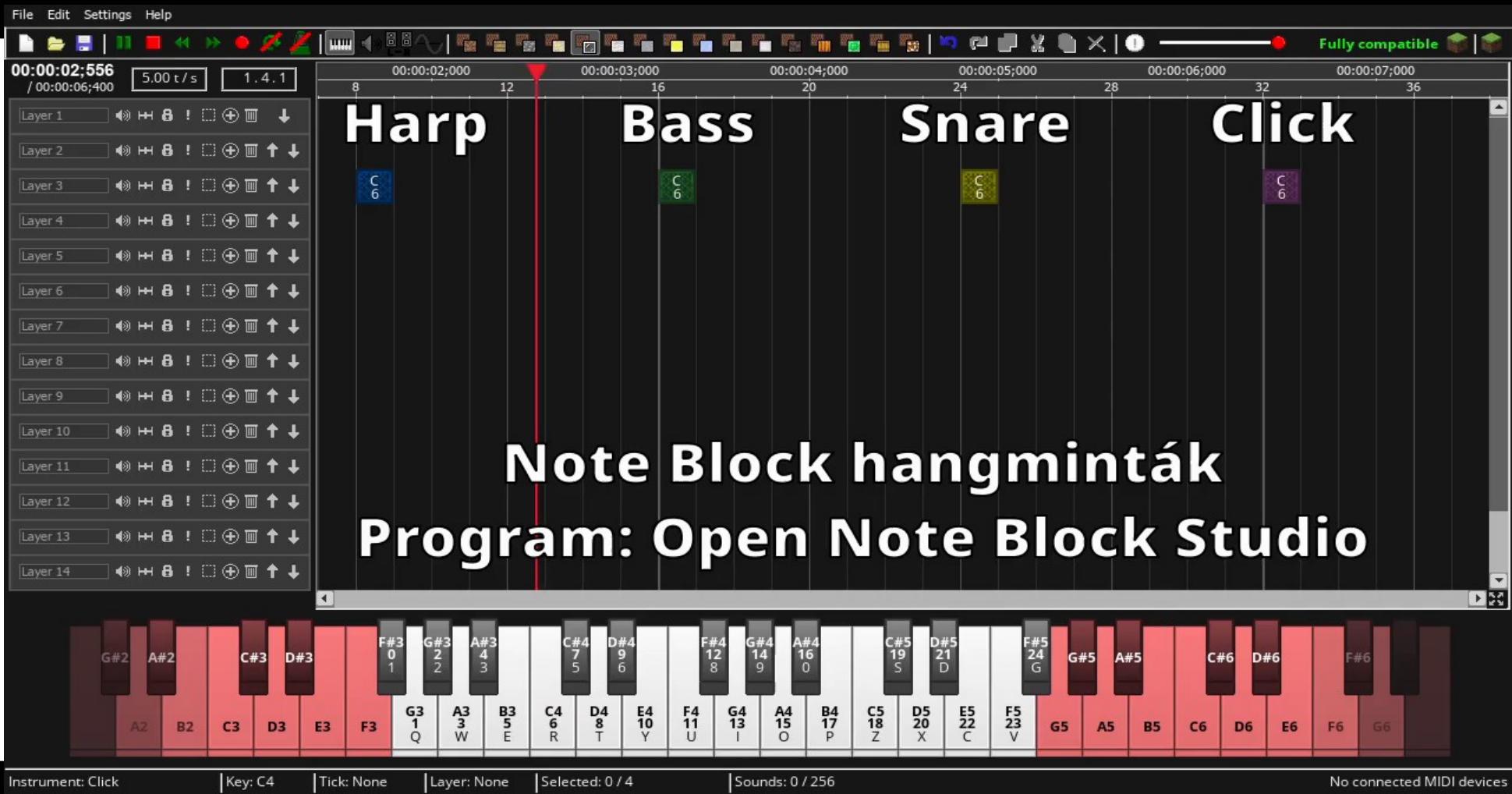
**Harp** **Bass** **Snare** **Click**

C 6 C 6 C 6 C 6

Note Block hangminták  
Program: Open Note Block Studio

G#2 A#2 C#3 D#3 F#3 G#3 A#3 C#4 D#4 F#4 G#4 A#4 C#5 D#5 F#5 G#5 A#5 C#6 D#6 F#6  
A2 B2 C3 D3 E3 F3 G3 A3 B3 C4 D4 E4 F4 G4 A4 B4 C5 D5 E5 F5 G5 A5 B5 C6 D6 E6 F6 G6

Instrument: Click | Key: C4 | Tick: None | Layer: None | Selected: 0 / 4 | Sounds: 0 / 256 | No connected MIDI devices



# Motiváció

- Sok szerző készít ilyen zenét
  - Vizualizáció
  - Remix / átdolgozás
  - Kottakészítés
  - Lejátszani hangszeren
- 
- Wave → Midi → Nbs nagyon pontatlan, közbeavatkozás kell



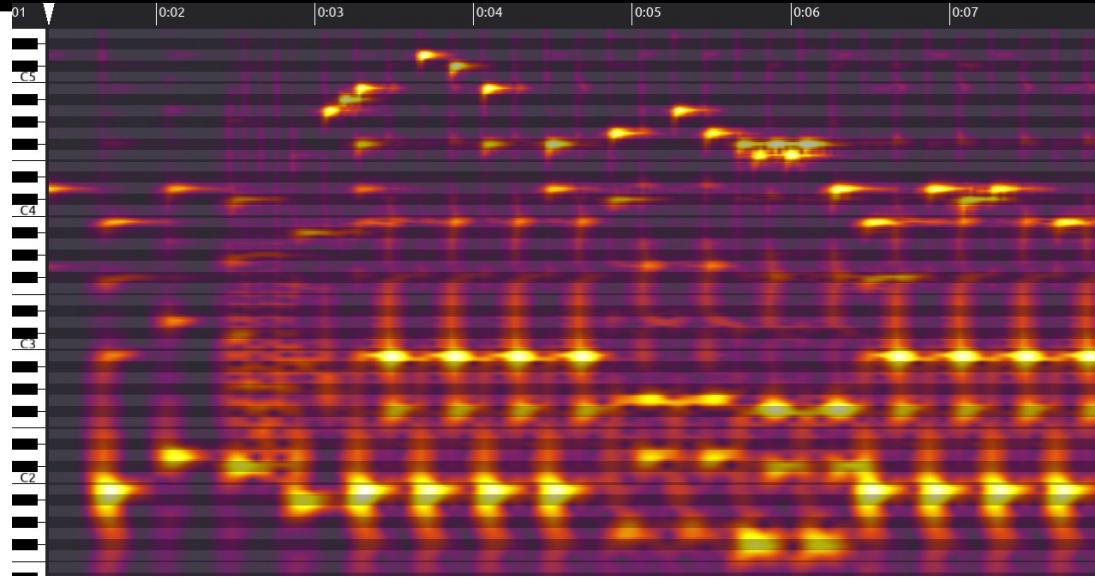
# Adat tulajdonságai

- Különböző hangszerek (5..16), jelenleg 4-gyel van tesztelve
- 2 oktávnyi hangmagasság (25)
- Hangerő 0.0 és 1.0 között
- Időzítés tps-ben (tick: legkisebb időegység), ONBS-ben oszlop

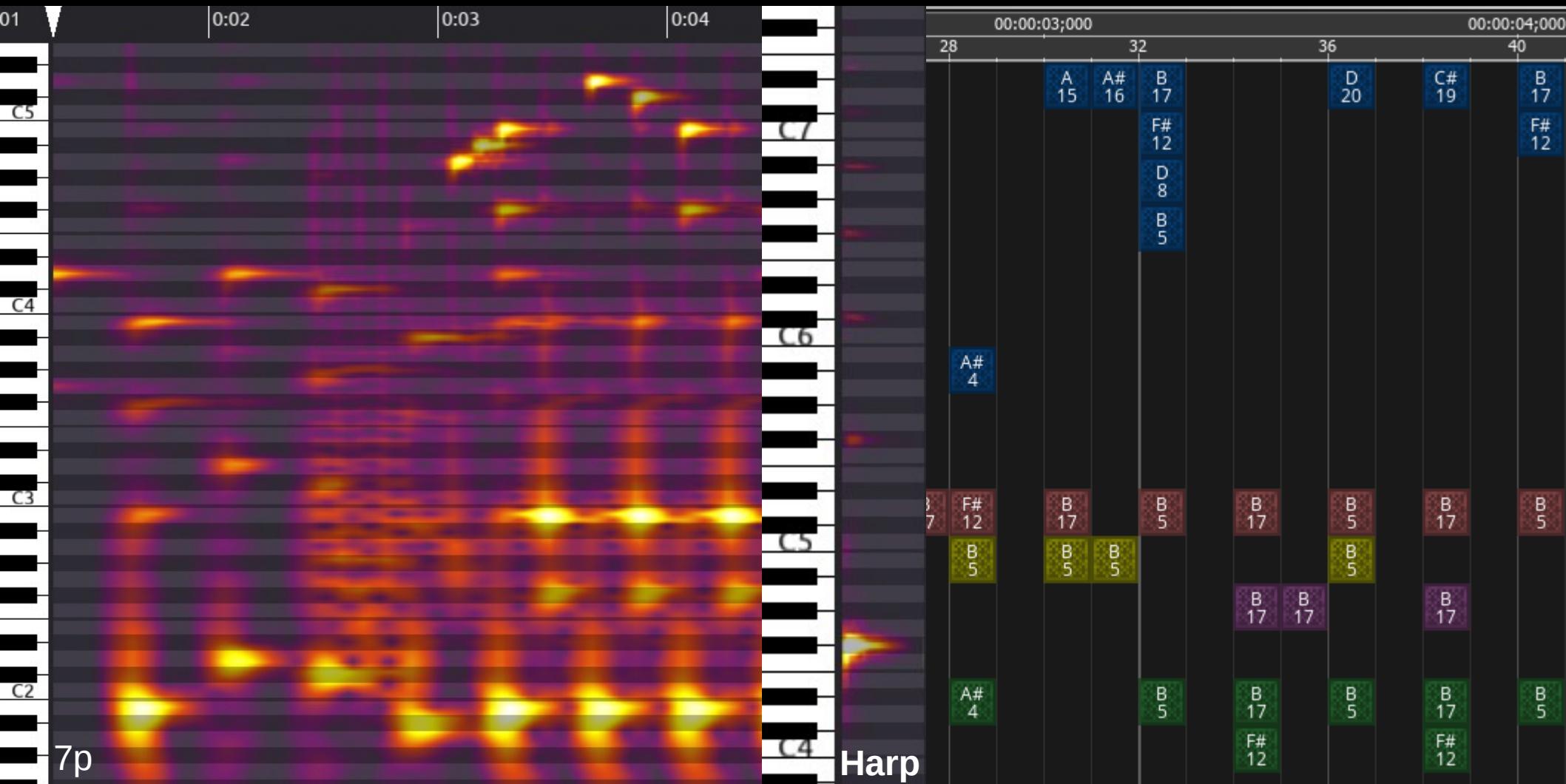


# Konverzió, keresés módszere

- Spektrogram:  
Fourier-transzformációk  
fix időközönként
- Vizsgálandó időpillanatok
- Előfelismerés az állapottér  
csökkentése érdekében
- Optimalizálás Nelder-Mead módszerrel



# Spektrogram (wave-ből generálható), és az eredeti NBS



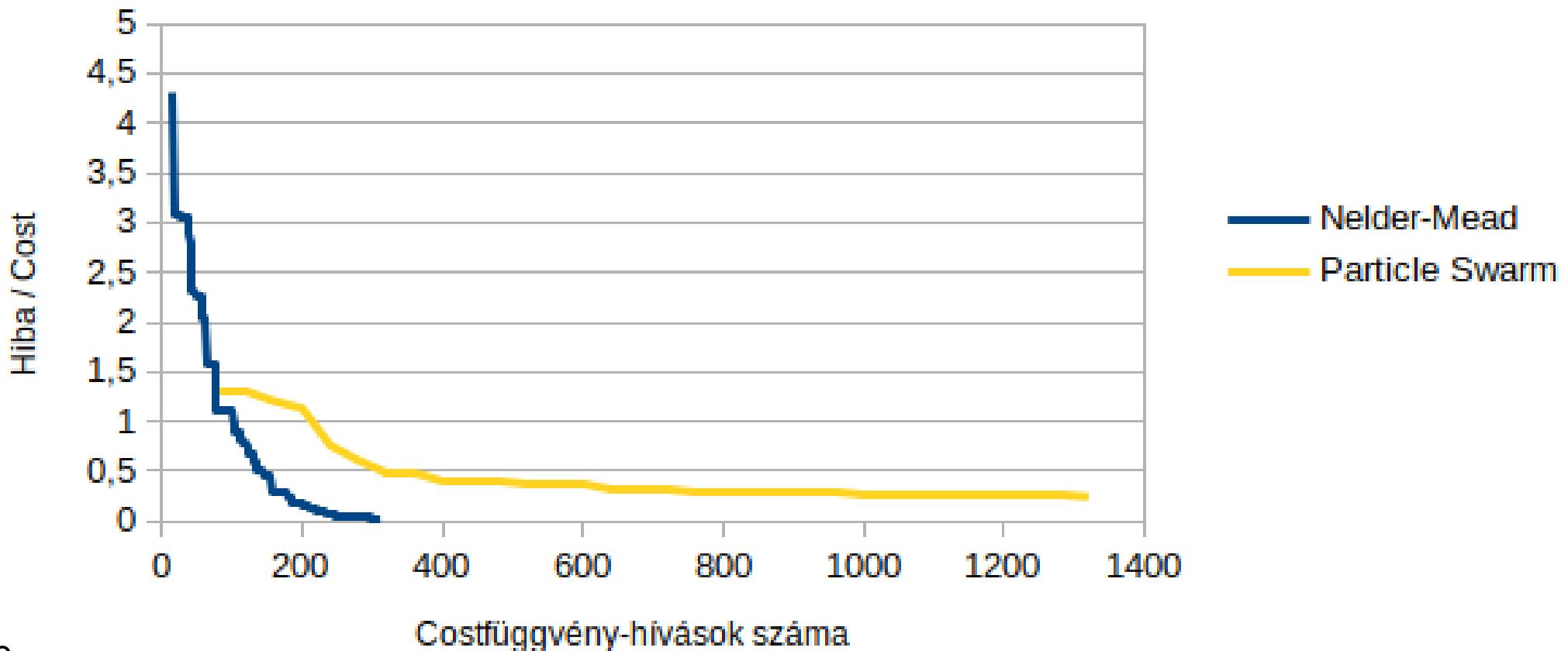
# Költségfüggvény az optimalizációhoz

- Költségfüggvény:  
bemenet: n db float a tippelt hangerőkkel  
kimenet: egy float, a hiba / költség (kisebb = jobb)
- Számítása:  
szintézis: minták hullámainak összeadása a tippelt hangerőn  
spektrogram számítása  
különbség abszolútértéke pixelenként
- Fontos: ne kelljen a deriváltja
- Drága



# Optimalizációs algoritmusok összehasonlítása

Egy konkrét időpillanatban, 11 optimalizálandó paraméterrel



# Értékelés

- Jórészt sikeres felismerés
- Lehetne fejleszteni, sok szempontból



# Fejlesztési lehetőségek, köszönöm a figyelmet!

- több hangszer felismerése
- tesztelés több zenével
- felismerés pontosítása, például előző tickben felismert hangok hullámának hozzáadásával a mostani felismerés közben
- Nelder-Mead optimalizációnak értelmesebb kezdeti érték adása az előfelismerés eredménye alapján
- automata tps, és túlvezérlés-kompenzáció felismerés
- parancssori kapcsolók a gyakran változtatandó paraméterekre
- robusztusabb felismerés másfajta nbs to wave exporterek / módszerek által generált hangfájlokra
- tesztelés más resource pack-kel / soundfonttal
- futásidő optimalizálása
- felismerés pontosságának mérése nagyobb adathalmazon, miután már kellően jó a felismerés
- sztereó felismerés, egyszerre ugyanaz a hang többször
- megpróbálni a layereket felismerni, amiken általában ugyanolyan hangerővel szerepelnek a hangok, ezen hangerők kerekítése a felismert hangerőre, és layer volume-ok használata

