Manuale d’Uso – Analisi dei Canti

Versione: Marzo 2025

# 1. Caricamento, Visualizzazione e avvio delle analisi (main.py)

Questa sezione permette di caricare e visualizzare file WAV, selezionarli, riprodurli e aprire strumenti di analisi.

## 1.1 Caricamento.

Il programma consente due modalità di caricamento dei file .wav:

1. **Caricamento di un singolo file**  
   Questa modalità si utilizza nella **prima fase del flusso di lavoro**, dedicata all’identificazione dei canti all’interno di un file audio continuo (ad esempio una registrazione lunga).  
   È utile per:
   * Esaminare visivamente il segnale.
   * Applicare zoom e selezioni.
   * Individuare e salvare i singoli canti con strumenti dedicati.
2. **Caricamento di un’intera cartella**  
   Questa modalità si impiega **nella fase successiva**, quando i canti sono già stati identificati e salvati in file separati.  
   Caricando l’intera cartella, è possibile:
   * Analizzare automaticamente o semi-automaticamente i canti.
   * Applicare analisi spettrali e parametriche sui singoli frammenti audio.
   * Salvare i risultati in formato JSON/Pickle per ulteriori elaborazioni.

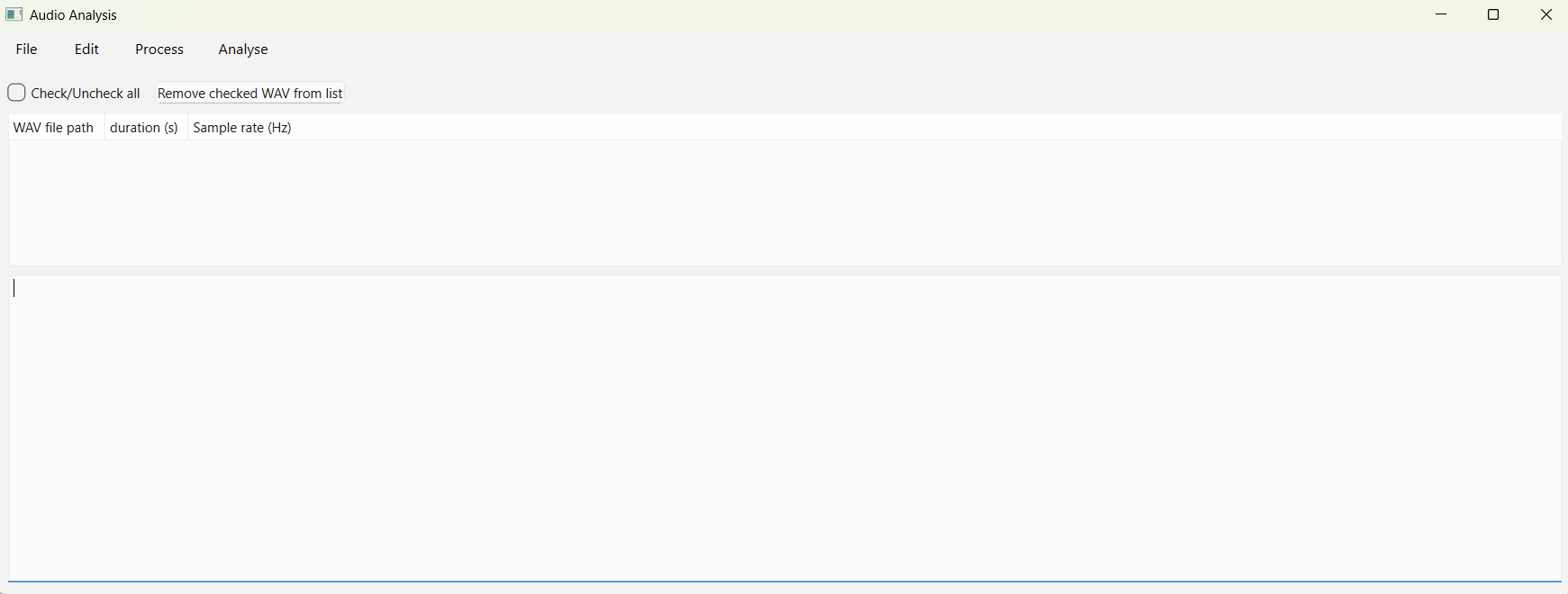


Figura 1 – Avvio del programma.

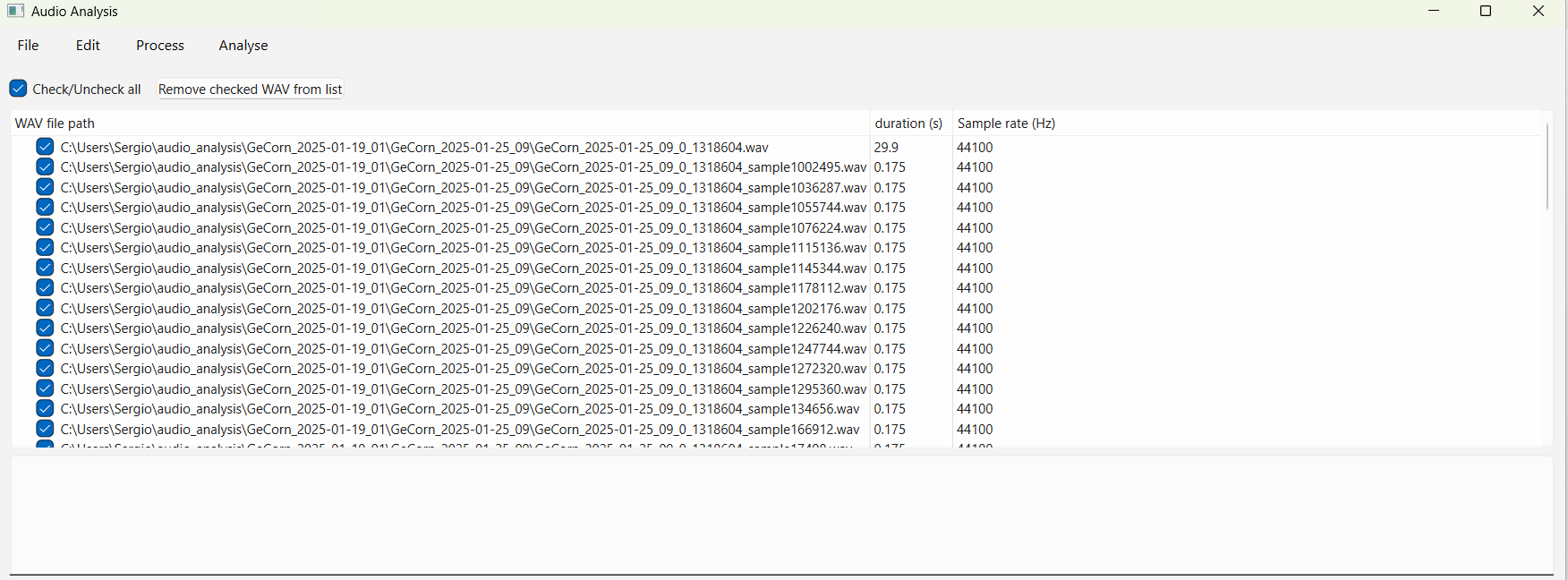


Figura 2 – Lista dei file caricati.

## 1.2 Visualizzazione e editing

Quando si apre un singolo file, il programma apre una finestra in cui è mostrato l’oscillogramma della registrazione. Selezionando con il mouse una regione e cliccando sul pulsante “**Zoom In**”, l’oscillogramma mostrerà solo la regione di interesse. I pulsanti “**Play**” e “**Stop**” attivano e disattivano la riproduzione della regione a schermo. Il tasto “**Zoom out**” consentirà di visualizzare nuovamente l’intero segnale audio.

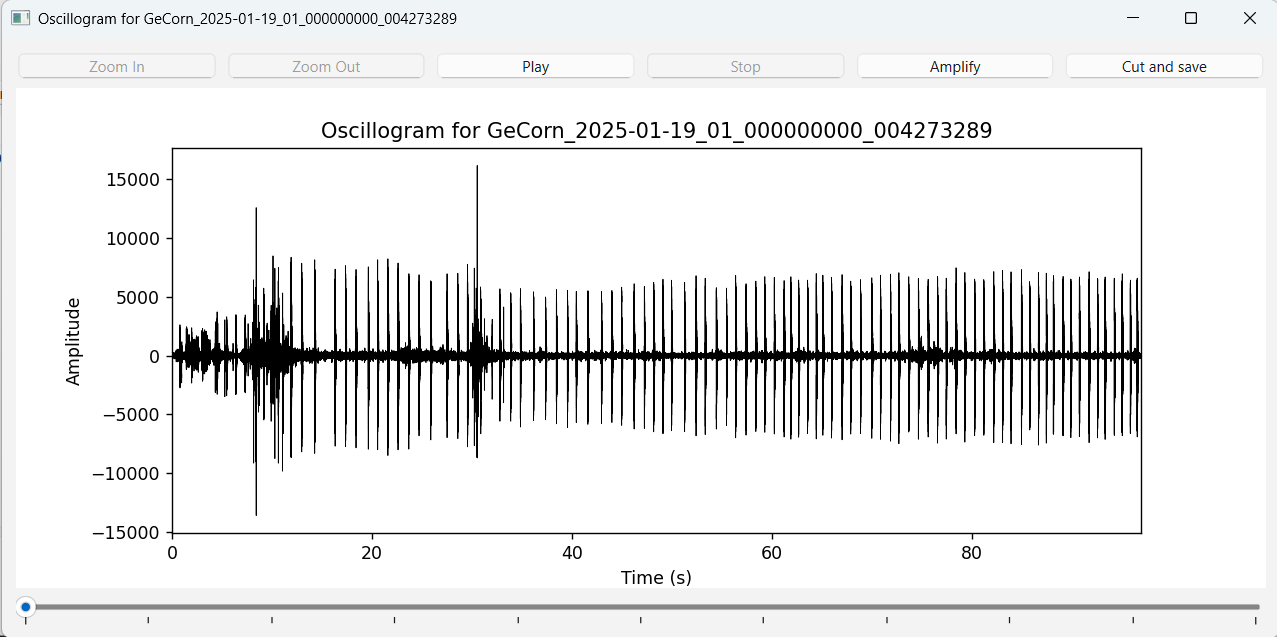


Figura 3 – Oscillogramma di un file.

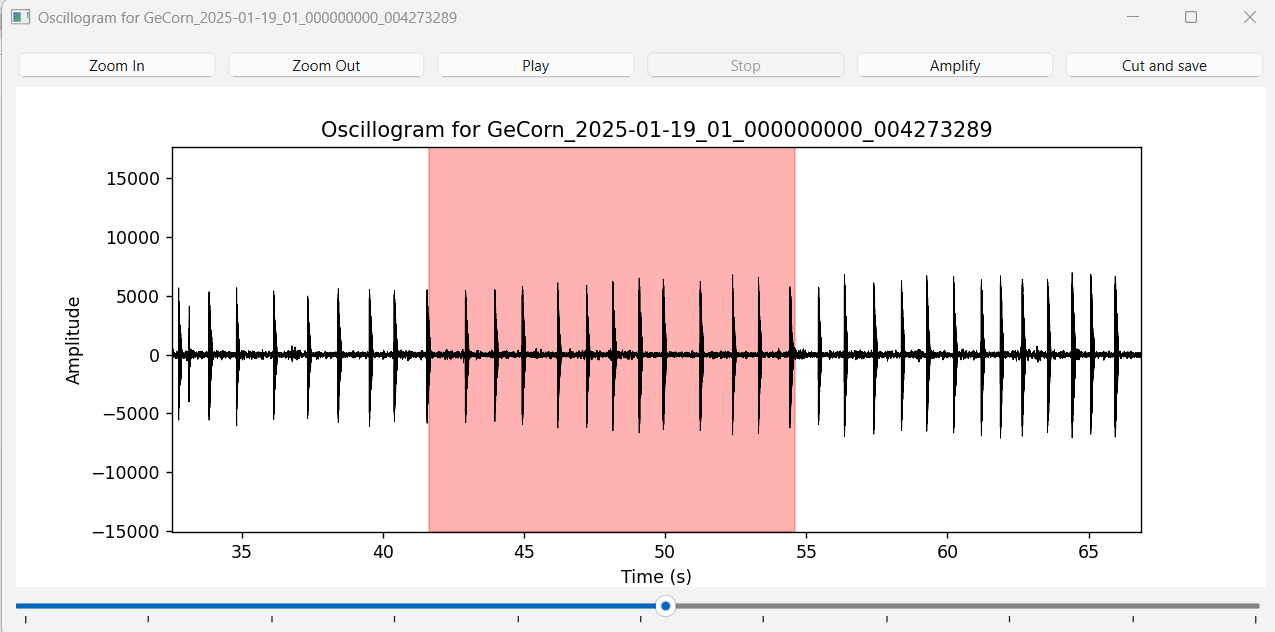


Figura 4 – Zoom sull’oscillogramma con selezione attiva.

Al momento, l’unica funzione di editing implementata è l’amplificazione. Cliccando il tasto “**Amplify**”, si aprirà una finestra di dialogo dove potete indicare il fattore di amplificazione. Valori maggiori di 1, aumenteranno l’intensità del segnale selezionato. Valori inferiori ad 1, la ridurranno. In questo modo, potrete aumentare il rapporto segnale/rumore e rendere più agevoli le successive analisi. I cambiamenti apportati sono salvati su un file temporaneo che potrà essere permanentemente salvato cliccando sul pulsante “**Cut and save**”.

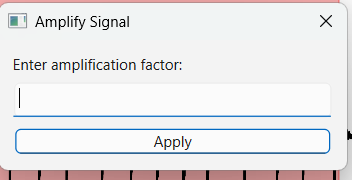


Figura 5 – Finestra di dialogo per la definizione del fattore di amplificazione.

## 1.3 Ritaglio e salvataggio degli spezzoni. Avvio della procedura di analisi.

**Importante**: Il tasto “Cut and save” avvia la procedura di analisi e deve quindi essere sempre utilizzato, indipendentemente che si voglia o meno tagliare il file.

Cliccando il pulsante “Cut and save”, si aprirà una nuova finestra (Figura 6), dove è chiesto di specificare il numero di spezzoni in cui dividere l’intera registrazione. Se la registrazione non è eccessivamente lunga, scegliete il valore di default (uno). Se volete suddividere la registrazione, indicate il numero di spezzoni ed il programma suddividerà il file audio in segmenti di durata simile, individuando il miglior punto di taglio nei pressi di zone silenziose (per evitare il taglio del segnale di interesse). Questa operazione consente inoltre di inizializzare il file JSON in cui verranno salvati tutti i risultati delle successive analisi.

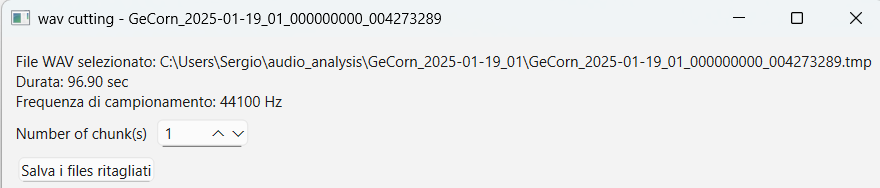


Figura 6 – Parametri per il ritaglio audio.

# 2. Rilevamento e Salvataggio dei Canti (trova\_picchi\_vs2.py)

Permette di calcolare l’inviluppo RMS, identificare i picchi (canti), stimarne la durata e salvarli come file distinti.

## 2.1 Parametri per il calcolo dell’inviluppo (Envelope)

L’inviluppo rappresenta l’andamento generale dell’ampiezza nel tempo e viene ottenuto tramite una media scorrevole dei valori assoluti del segnale.

Due parametri principali ne influenzano la forma:

* Window size (finestra): Indica la durata (in millisecondi) della finestra su cui viene calcolata la media dell’ampiezza assoluta. Una finestra più ampia produce un inviluppo più smussato, utile per ignorare le variazioni interne al canto (es. impulsi) e individuare la forma generale del canto (inizio e fine).
* Overlap (passo) Indica di quanto si sposta la finestra tra un calcolo e il successivo. Un overlap più piccolo genera una curva più continua, mentre uno più grande rende l’inviluppo più “a gradini”.

Per facilitare il rilevamento automatico dei canti nel segnale, bisogna ottenere un inviluppo che segua il profilo del canto, senza evidenziare i modulazioni interne, come gli impulsi.

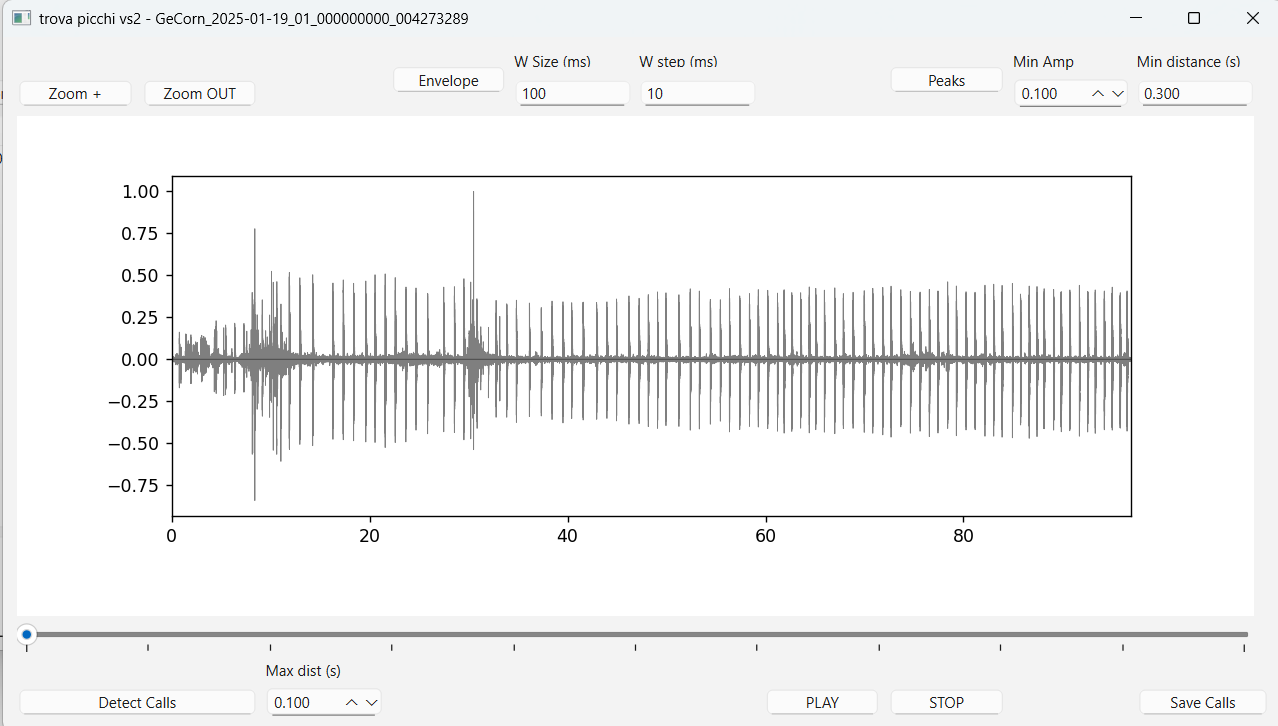


Figura 7 – Inviluppo calcolato.

## 2.2 Parametri per l’individuazione dei picchi (Peaks)

Calcolato l’inviluppo, si procede con la rilevazione dei picchi, che corrispondono al singolo canto. Il rilevamento utilizza due parametri della curva di inviluppo:

* Ampiezza minima (Min Amp). Imposta una soglia e permette di filtrare i picchi sotto di essa. Questo parametro è utile per eliminare piccoli rumori di fondo o variazioni non significative.
* Distanza minima (Min Distance). Impone un intervallo minimo di tempo tra due picchi consecutivi, espresso in secondi. Evita il rilevamento di picchi troppo ravvicinati che potrebbero derivare da fluttuazioni locali dell’inviluppo all’interno di un singolo segnale.

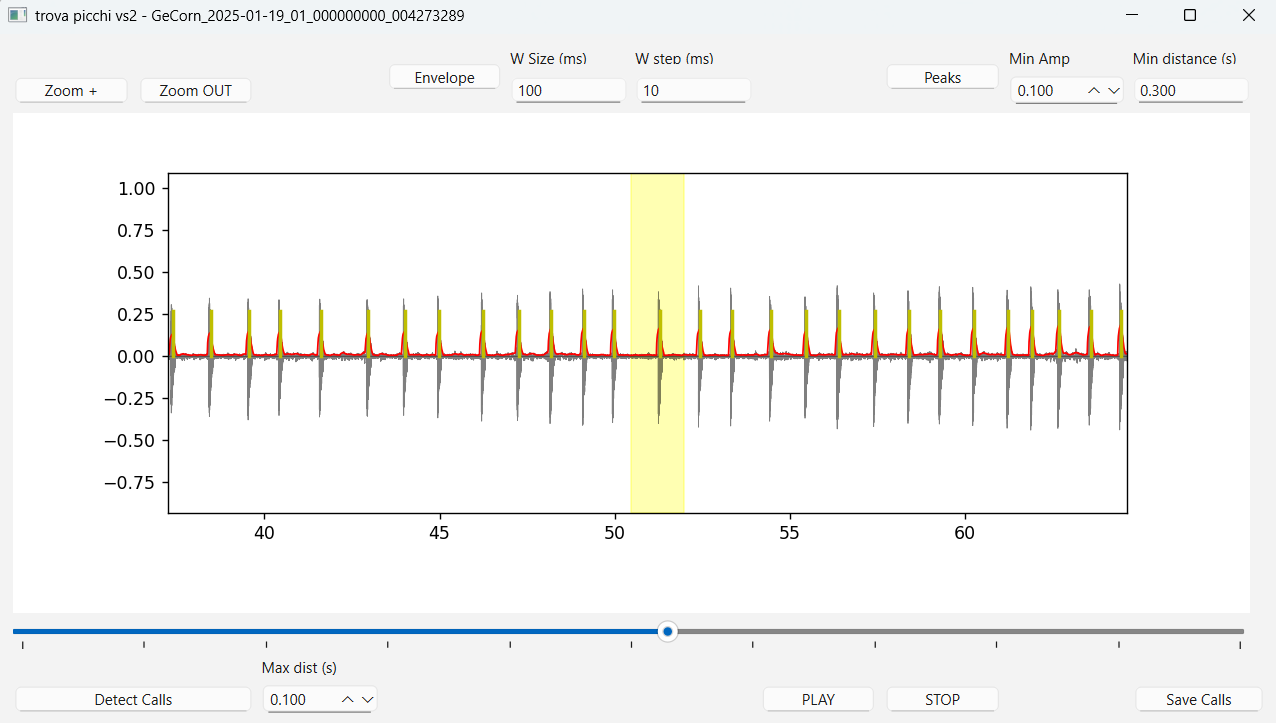


Figura 8 – Picchi rilevati.

## 2.3 Parametri per definire inizio e fine della traccia contenente il singolo canto (Detect Calls)

Trovati i picchi, si utilizza la funzione Detect Calls per individuare inizio e fine di ciascun segmento contenente un canto. La funione usa il parametro “Max dist (ms)” che definisce l’ampiezza della finestra di calcolo utilizzata dall’algoritmo. Questa finestra deve essere all’incirca lunga come (o poco inferiore alla) durata media del canto, ma meno lunga della distanza tra due picchi successivi. L’algoritmo utilizzato è il seguente:

Dove è il campione corrispondente al picco j-esimo dell’inviluppo e *K* è il numero di campioni corrispondenti alla Max\_dist impostata.

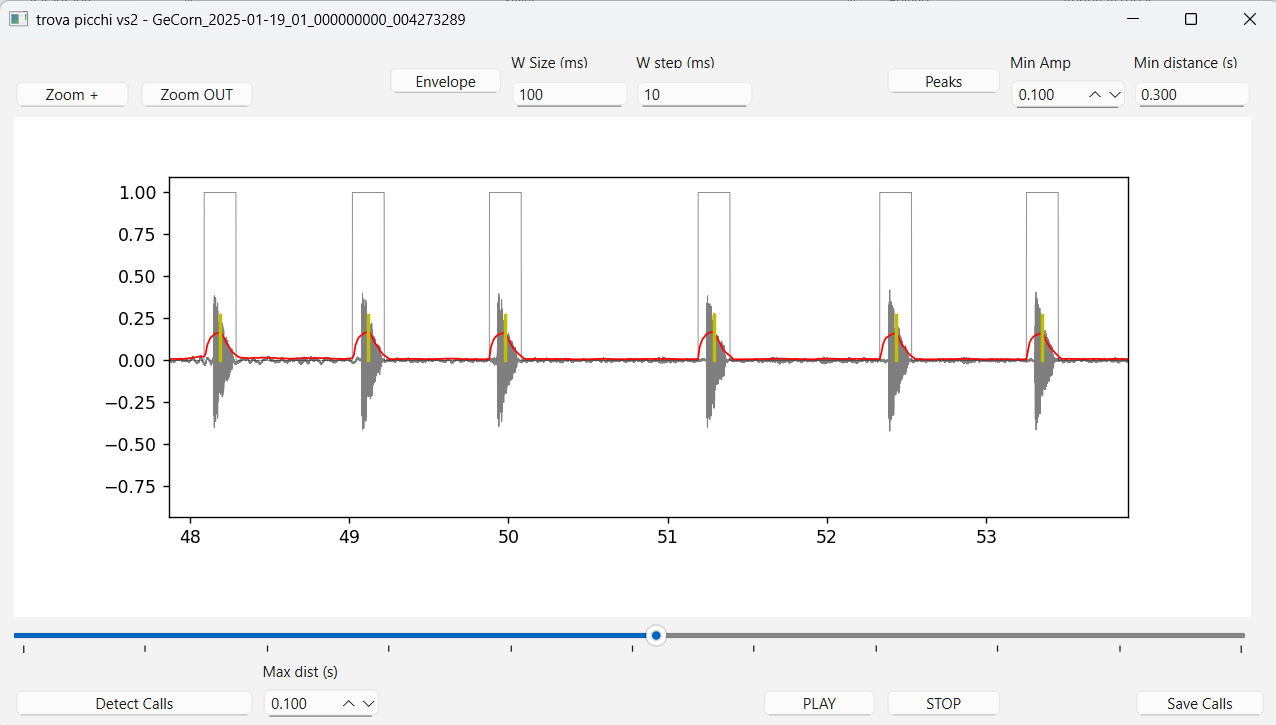


Figura 9 – Canti individuati.

Una volta individuati e controllati i segmenti, si può procedere al taglio e al salvataggio dei singoli spezzoni. Il programma chiede di scegliere una cartella in cui salvare i canti. Il programma, crea quindi una sotto-cartella, il cui nome è ??? e salva al suo interno i singoli spezzoni.

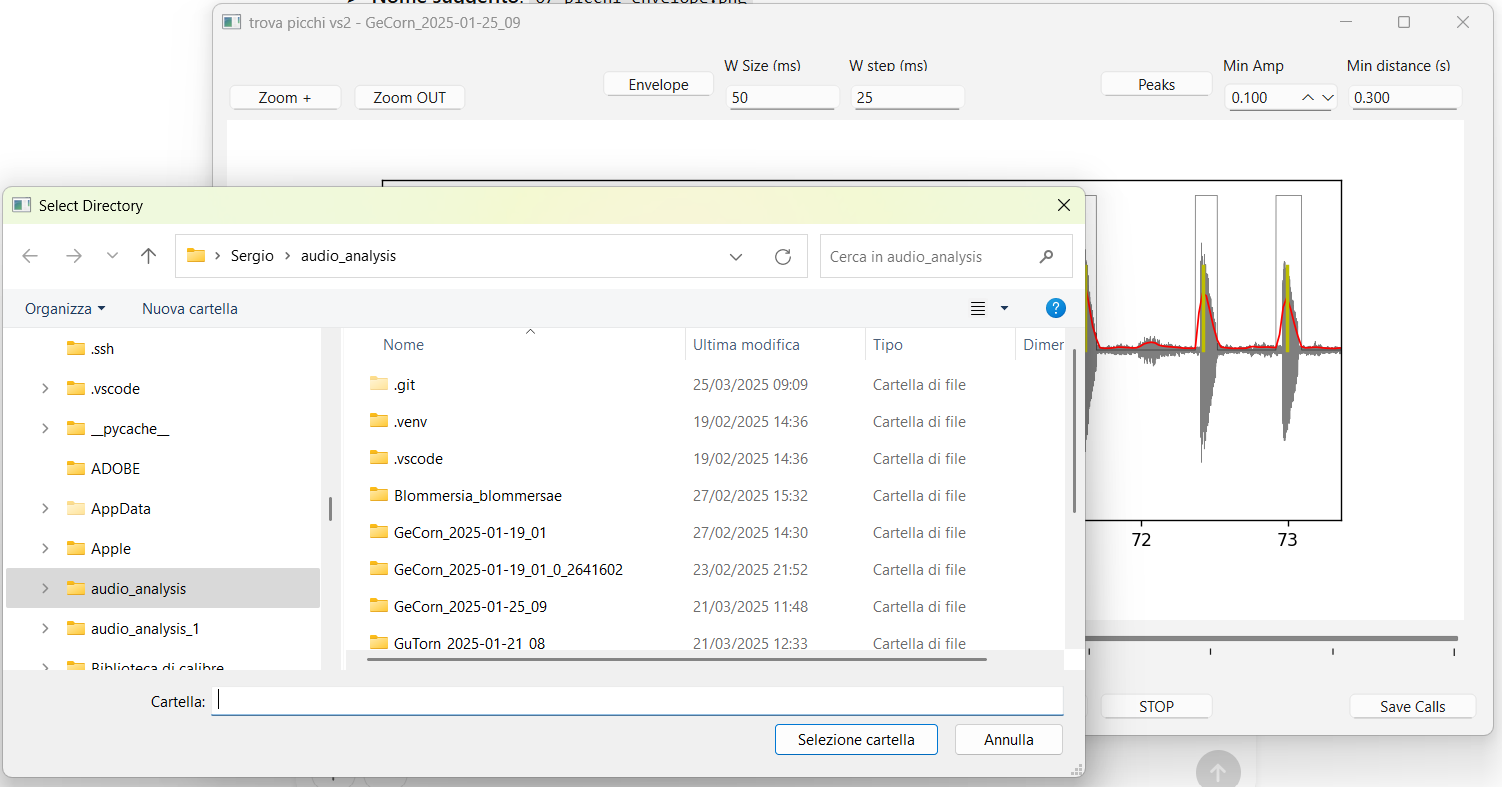
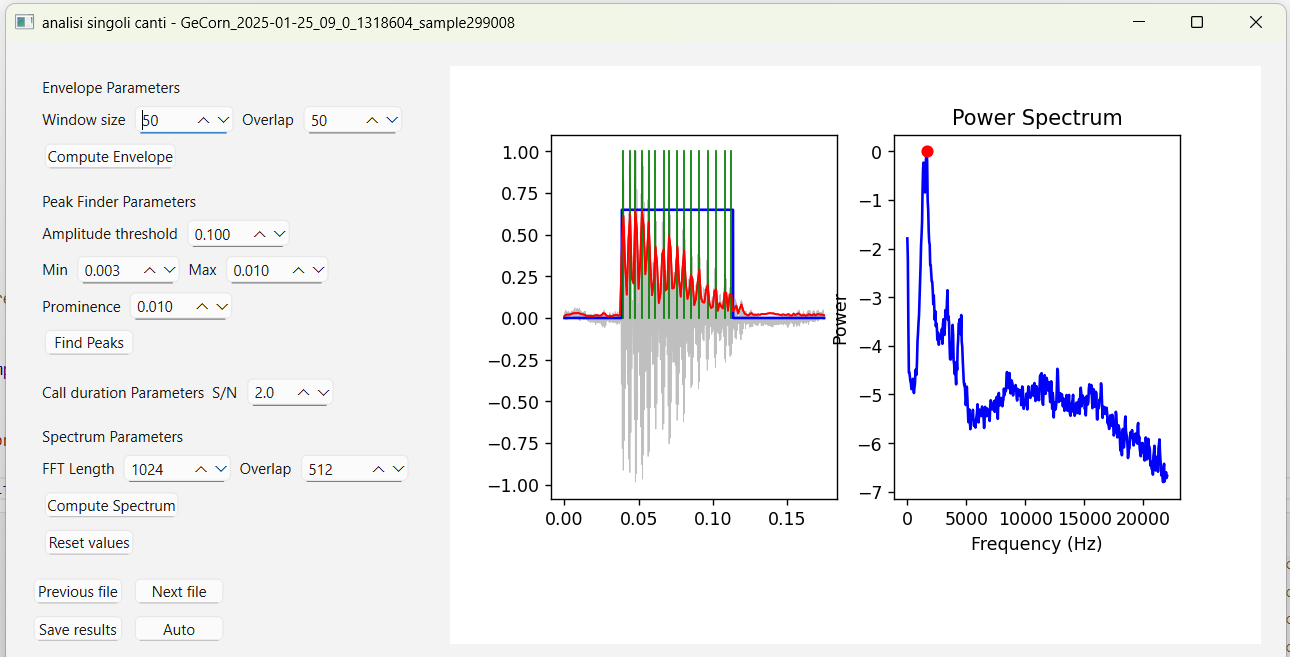


Figura 10 – Finestra di salvataggio dei canti.

# 3. Analisi dei Singoli Canti (analisi\_singoli\_canti.py)

Questo modulo elabora i canti salvati, estrae parametri come durata, numero di impulsi e spettro di frequenza, e salva i risultati in formato JSON. Il programma può procedere in modalità interattiva e non-interattiva (automatica). Nel primo caso, con i tasti [Previous file] e [Next file] si scorre la lista dei files, si modificano i parametri e, se soddisfatti, si salvano i risultati nel file JSON. Nella modalità automatica, il programma processa tutti i files applicando gli stessi parametri e salvando i dati nel file JSON, senza possibilità di modificarli.

Figura 11 – Analisi di un canto (oscillogramma e spettro).



# Parametri per il calcolo dell’inviluppo (RMS)

|  |  |
| --- | --- |
| **Parametro** | **Descrizione** |
| window\_size | Lunghezza della finestra (in campioni) usata per calcolare il valore RMS. |
| overlap | Numero di campioni di sovrapposizione tra una finestra RMS e la successiva. |

# Parametri per la rilevazione dei picchi

|  |  |
| --- | --- |
| **Parametro** | **Descrizione** |
| min\_amplitude | Soglia minima di ampiezza: esclude i picchi con valori RMS troppo bassi. |
| min\_distance | Distanza minima tra due picchi consecutivi (in secondi). |
| max\_distance | Distanza massima per considerare validi due picchi consecutivi. |
| prominence | Prominenza minima richiesta per un picco: quanto si distingue dal contesto. |

# Parametro per identificare l’intero canto

|  |  |
| --- | --- |
| **Parametro** | **Descrizione** |
| signal\_to\_noise\_ratio | Rapporto segnale/rumore usato per stimare inizio e fine del canto in base al livello di fondo dell’inviluppo. |

# Parametri per lo spettro di potenza

|  |  |
| --- | --- |
| **Parametro** | **Descrizione** |
| fft\_length | Numero di campioni su cui calcolare la FFT (risoluzione dello spettro). |
| fft\_overlap | Numero di campioni condivisi tra finestre successive nella stima spettrale. |

Esempio di organizzazione dei risultati per il file GeCorn\_2025-01-25\_09.wav tagliato in 4 frammenti con il primo frammento analizzato:

GeCorn\_2025-01-25\_09

├── GeCorn\_2025-01-25\_09\_000000000\_002328479

│ ├── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_000018963.wav

│ ├── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_000049832.wav

│ ├── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_000074529.wav

│ ├── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_000097461.wav

│ ├── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_000135828.wav

│ ├── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_000168462.wav

│ ├── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_000196245.wav

...

│ ├── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_002236752.wav

│ ├── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_002267622.wav

│ └── GeCorn\_2025-01-25\_09\_000000000\_002328479\_sample\_002301579.wav

├── GeCorn\_2025-01-25\_09\_000000000\_002328479.wav

├── GeCorn\_2025-01-25\_09\_002328480\_004662826.wav

├── GeCorn\_2025-01-25\_09\_004662827\_006991306.wav

├── GeCorn\_2025-01-25\_09\_006991307\_009320446.wav

├── GeCorn\_2025-01-25\_09.json

└── GeCorn\_2025-01-25\_09.pkl