# Echogram.jl: Echogram Bottom Detection and Export

This test script performs a complete workflow using the `EcoSons` package, from loading sonar data and detecting the seafloor (bottom), to visualization and exporting. It uses a configuration file (`params.json`) to control processing parameters.

## 1. Load Dependencies

```
using EcoSons
using Test
using JSON
```

- **EcoSons**: A package for sonar data processing (custom module).
- **Test**: 's built-in testing library.
- **JSON**: Used to parse the configuration file in JSON format.

---

## 2. Define and Start Test Set

```
@testset begin
```

Starts a `Test` block, allowing the entire pipeline to be wrapped in test assertions if needed (though no actual `@test` expressions are used here — it's primarily a script run under the testing framework).

---

## 3. Load Configuration Parameters

```
json_path = joinpath(@__DIR__, "../config/params.json")
config = JSON.parsefile(json_path)
```

- Loads a configuration file that contains paths and processing parameters such as:
  - `data_dir`: Directory containing sonar files.
  - `channel`: Transducer channel to read.
  - `transect`: Index or key to select a specific section of the data.
  - `bottom_detection`: Dict of keyword arguments for the bottom detection algorithm.

---

## 4. Prepare File Paths

```
dir = joinpath(@__DIR__, config["data_dir"])
files = filter(f -> endswith(f, ".raw"), readdir(dir))
full_paths = joinpath.(dir, files)
channel = config["channel"]
```

- Builds a list of full paths to `.raw` sonar data files in the specified directory.

## 5. Load Sonar Data

```
data, dim = load_sonar_data(channel, full_paths)
```

- Loads sonar data from the listed `.raw` files using the specified `channel`.
- Returns:
  - `data`: A vector of ping records, each holding metadata and echo data.
  - `dim`: Dimensional info (number of channels).

## 6. Save Sonar Data

```
 jld2dir = joinpath(@__DIR__, config["JLD2_dir"]["data"])
    saveJLD2("$(jld2dir)", data)
```

• Constructs output path from config.

• Saves the sonar data in JLD2 format for efficient future loading

## 7. Select Transect

```
transect = config["transect"]
```

- Loads the transect index or filter parameters.
- (Commented-out code suggests the option to select data by lat/lon bounding box.)

## 8. Apply Bottom Detection

```
bottom_args = Dict(Symbol(k) => v for (k, v) in
config["bottom_detection"])
data = compute_bottom(data; bottom_args...)
```

- Converts string keys from JSON into symbols for keyword arguments.
- Applies `compute_bottom`, a function that detects the seafloor in the sonar data using configurable parameters (e.g., thresholding, smoothing).

## 9. Inspect Sound Velocity (Optional Debug)

```
println(data[1].Q[2].sampleInterval) # First ping
```

- Debug/inspection step: prints the `sampleInterval` of the second transducer header in the first ping.

```
# set_sound_velocity!(data; velocity = velocityUNESCO(20.0, 35.0,
10.0))
```

- (Commented out): optionally set sound velocity using the UNESCO formula for temperature/salinity/depth.

## 10. Subset Data for Transect

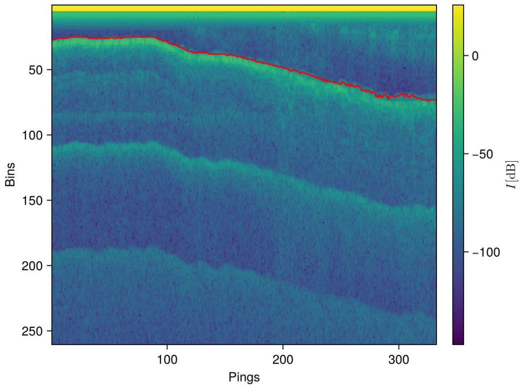
```
data_selected = data[transect]
```

- Extracts only the data for the transect of interest.

## 11. Plot the Echogram with Bottom

```
plot_echobottom(data_selected)
```

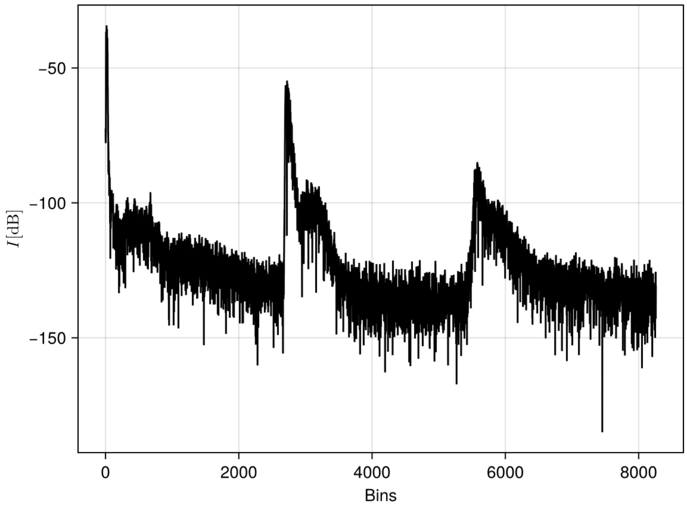
- Displays an echogram (2D grayscale or color image) with detected bottom overlaid:



## 12. Plot a Single Ping

```
ping = 1
plot_ping(1:length(data[transect].P[ping,:]),
data[transect].P[ping,:])
```

- Plots the echo power (amplitude) for a single ping in the transect.
- X-axis: sample indices; Y-axis: signal strength.



## 13. Export Bottom-Detected Data

```
export_name = "echobottom.dat"
export_dir = joinpath(@__DIR__, "..", "data", export_name)
export_echobottom(data_selected, transect, export_dir)
```

- Exports the processed transect (with bottom detections) to a `.dat` file for further use or external analysis.

## Summary

This script automates a sonar data processing workflow using configuration-driven parameters:

- Loads `.raw` sonar data files.
- Applies bottom detection.
- Optionally adjusts physical parameters (e.g., sound velocity).
- Visualizes both echograms and individual pings.
- Exports the annotated data for downstream analysis.