MISSIONE 4
ISTRUZIONE RICERCA

RESTART

THE COBS TOOLKIT

Table of Contents

1 Introduction

This document describes a toolkit specifically designed for the compression of biometric and biomedical signals, developed as part of the COBS4FUN project.

COBS4FUN stands for "Compression Of Biometric Signals for FUture Networks applications" and was a cascade call project of the structural project S7 FUN-Media, coordinated by the Politecnico di Torino (Spoke 4) and funded by the European Union – NextGenerationEU as part of the NRRP – M4C2, Investment 1.3, program "RESearch and innovation on future Telecommunications systems and networks, to make Italy more smART" (RESTART), PE00000001.

The main aim of COBS4FUN was the development of new formats and compression techniques for efficient encoding and transmission of biometric signals.

In particular, a comprehensive framework has been developed for the management of biometric data, with a focus on interoperability, modularity, and efficient storage and transmission.

The framework includes:

- a new standardized format for biometric multimodal data (**BMD**),
- a dedicated tool for its creation and manipulation (**BMDComposer**), and
- a flexible environment for the development and evaluation of compression algorithms for biometric signals and traits (**COBS** - Compression Of Biometric Signals).

This document focuses specifically on **COBS**, summarizing its functionalities and its role within the overall framework.

As will be shown in the following sections, the COBS toolkit enables the creation of compression pipelines by combining predictors, transformations, and entropy encoders. Moreover, it allows for the comparison and evaluation of the performance of multiple compression pipelines in terms of compression efficiency and signal distortion.

The features and implementation details of COBS are briefly described in the next sections of this report.

2 COBS Toolkit Overview

COBS (Compression of Biometric Signals) represents a comprehensive software framework specifically designed for the compression of biometric and biomedical signals. In particular, the tool was developed to assist in the design of compression algorithms and to measure the performance of different algorithms on a large dataset, using uniform metrics and providing easily comparable results.

The toolkit provides an environment that supports the user throughout the whole cycle of design and evaluation of a compression algorithm, including data formatting, compression, generation of graphs and comparative tables.

The main idea behind the toolkit is to provide a simple and intuitive graphical user interface (GUI) that allows users to select and combine commonly used building blocks of a compression pipeline - such as predictors, transformations and encoders - and observe the effects of these choices on relevant performance metrics.

At the end, COBS provides researchers, clinicians, and signal processing professionals with a sophisticated yet intuitive platform for designing and handling compression algorithms for various types of physiological data, including electrocardiograms (ECG), electroencephalograms (EEG), electromyograms (EMG), and other biosignals commonly encountered in medical research, clinical practice and biometrics.

The software architecture supports multiple biomedical file formats including **EDF** (European Data Format), **BDF** (BioSemi Data Format), **WFDB** (Waveform Database Format), **FIFF** (Functional Interchange File Format), **WAV** (16-bits and 24-bits), as well as **CSV** and **MAT** for integration with Excel and MATLAB environments, ensuring seamless integration with existing clinical and research workflows.

Format conversion capabilities implemented in COBS maintain complete metadata integrity, preserving critical technical information such as sampling rates, channel labels, and acquisition parameters that are essential for signal processing applications.

The software architecture is built around a modular five-stage compression pipeline, shown in Fig. 1, consisting of preprocessing, prediction, transformation, quantization, and entropy coding stages.

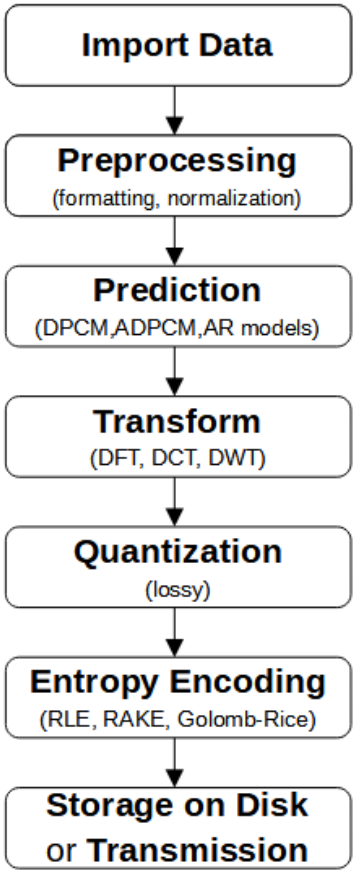

Figure 1: Scheme of a compression pipeline devised for biometric signals.

This systematic approach enables users to construct custom compression algorithms optimized for specific signal characteristics and application requirements, where each stage can be independently configured and optimized.

The graphical user interface emphasizes accessibility through three specialized workspaces. The Data Editor tab facilitates format conversions and preprocessing operations between various biomedical signal formats. The Compress tab enables users to leverage pre-configured compression algorithms with optimized parameter sets for common biomedical signal types. The Designer tab provides users with interactive algorithm development capabilities, featuring real-time preview and performance feedback that enables custom pipeline construction with immediate visualization of compression effects and signal quality metrics.

The system supports both lossless and lossy compression modes, enabling applications ranging from critical clinical data archival where perfect signal reconstruction is mandatory, to efficient telemedicine transmission where controlled quality loss is acceptable for bandwidth optimization.

The modular design philosophy prioritizes component independence, algorithm interchangeability, and system extensibility. The JSON-based algorithm configuration system provides extensibility for research applications while maintaining compatibility with standard analysis tools. This flexibility enables users to add custom prediction algorithms, transformation methods, and coding schemes through well-defined interfaces. The ability to export standalone Python scripts facilitates integration with external processing pipelines and enables reproducible research workflows by generating complete algorithm implementations with embedded parameters that can operate independently of the GUI application.

Block-based processing capabilities enable handling of large datasets that exceed available memory while maintaining compression efficiency through adaptive block size selection. The software supports both real-time processing for immediate feedback during algorithm development and batch operations for efficient processing of large biomedical datasets.

The comprehensive approach addresses diverse requirements across multiple domains within biometric and biomedical signal processing, from academic research requiring algorithm experimentation and validation, to biometric applications demanding reliable format conversion and data archival, to telemedicine systems requiring efficient signal transmission with controlled quality parameters. This versatility, combined with the intuitive interface design and robust technical implementation, establishes COBS as a significant contribution to the biometric and biomedical signal processing community.

3 COBS Pipeline and Building Blocks

The COBS signal processing pipeline implements a sophisticated four-stage compression architecture that systematically reduces data redundancy while maintaining signal integrity and providing precise control over compression characteristics. The pipeline design follows established principles from information theory and signal processing, combining temporal prediction, spectral transformation, controlled quantization, and entropy coding to achieve optimal compression performance for biometric signals.

The pipeline architecture operates on the principle that biometric signals contain multiple forms of redundancy that can be exploited through different algorithmic approaches: temporal redundancy, arising from the predictable nature of physiological processes, is addressed through sophisticated *prediction* algorithms that model signal evolution over time; spectral redundancy, resulting from energy concentration in specific frequency bands, is exploited through *transformation* algorithms that provide compact signal representations in alternative domains.

COBS implements multiple prediction algorithms ranging from simple linear predictors (DPCM, MA), to adaptive predictors (LMS, NLMS, RLS, Kalman), and up to more sophisticated adaptive approaches (Volterra and MLP) that can accommodate non-stationary signal characteristics commonly encountered in physiological monitoring applications.

COBS implements classical frequency-domain transforms (DFT and DCT), and wavelet-based multi-resolution analysis techniques (DWT), enabling optimal algorithm selection based on specific signal characteristics and compression objectives.

A detailed description of the transformations supported by COBS is provided in the Deliverable D3.1.

After transformation, the representation of many signals is suitable for compression: the few higher coefficients contain most of the energy, while the other ones contain information related to noise or not significant details.

At this stage, the signal representation can undergo further processing to increase sparsity, thereby improving its compressibility. Two commonly used techniques are Vertical thresholding (the first k coefficients are retained and the others are set to zero) and Horizontal thresholding (coefficients with values below a given threshold are discarded). COBS allows both thresholding techniques.

The choice of the signal transform and threshold method is a trade-off between noise reduction, compression, and distortion.

Quantization operations provide controlled quality reduction by reducing the precision of transform coefficients or prediction residuals according to user-specified quality parameters. The quantization process creates discrete-valued signals that are more amenable to entropy coding while enabling explicit control over the trade-off between compression efficiency and signal fidelity.

The actual implementation supports uniform scalar quantization with configurable step sizes.

In the future, more sophisticated vector quantization approaches will be integrated that exploit coefficient interdependencies to maximize compression performance while minimizing reconstruction errors.

The *entropy coding* stage performs the final compression operations by exploiting statistical characteristics of quantized signals to generate compressed bit streams. Multiple entropy coding algorithms are available in COBS, optimized for different coefficient distributions and computational requirements. In particular, five families of entropy encoders are available: Huffman encoding, Arithmetic encoding, Run-Length encoding, Golomb-Rice encoding and RAKE encoding.

The coding stage generates compressed data streams along with metadata necessary for reconstruction, completing the compression pipeline and enabling storage or transmission of processed signals.

More details on the set of building blocks available in the framework are described in the Deliverable D3.1 *"COBS: A New Toolkit for Compression of Biometric Signals and Traits"*.

4 Graphical User Interface

COBS Graphical User Interface (GUI) implements a tabbed interface design that organizes functionality into three primary workspace areas. This design philosophy emphasizes task-oriented workflow organization, allowing users to focus on specific aspects of signal processing.

The GUI, implemented using PyQt5, is organized into three tabs, named respectively: **Data Editor** tab (for format conversion), **Designer** tab (for the design and testing of compression pipelines), and **Compress** tab (for executing test campaigns using both the codecs developed within COBS and external ones). This structure reflects the design workflow: from data preparation, to the exploration of solutions, and finally to the evaluation of performance on large-scale datasets. Further details on the functionalities of the three tabs will be provided in the following paragraphs.

4.1 The Data Editor tab

The **Data Editor** tab serves as the primary interface for format conversion operations and signal preprocessing tasks. This workspace provides comprehensive support for biometric signal formats commonly encountered in research environments, offering bidirectional conversion capabilities that preserve metadata integrity and signal quality throughout the conversion process.

The interface organization within the Data Editor tab follows a logical workflow progression from input selection to output configuration. File selection controls support both individual file processing and batch operations on entire directories, with intelligent file type detection that automatically configures appropriate conversion parameters. The source format detection mechanism analyzes file headers and metadata to ensure accurate interpretation of signal characteristics and acquisition parameters.

Conversion configuration options are presented through intuitive control panels that adapt based on the selected source and target formats.

The preview capabilities within the Data Editor tab enable users to verify conversion results before committing to full processing operations.

Batch processing functionality streamlines workflows involving multiple files or entire datasets. The interface provides progress monitoring capabilities with detailed logging of conversion operations, error reporting, and quality metrics calculation.

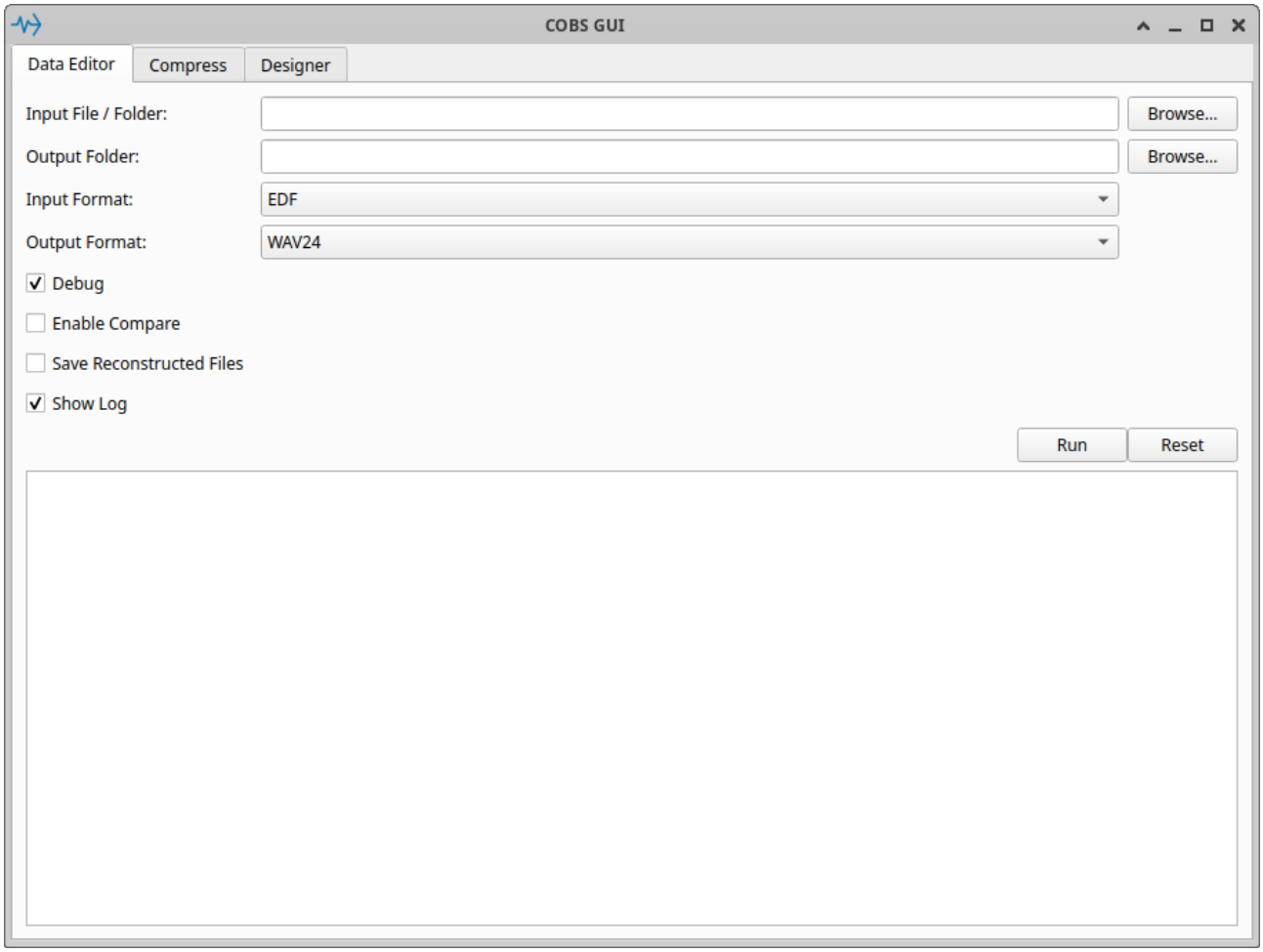

Figure 2: The Data Editor tab.

A few screenshots of the **Data Editor** tab are shown in Figs. 2, 3. More detailed information can be found in deliverable D3.1 *"COBS: A New Toolkit for Compression of Biometric Signals and Traits"*.

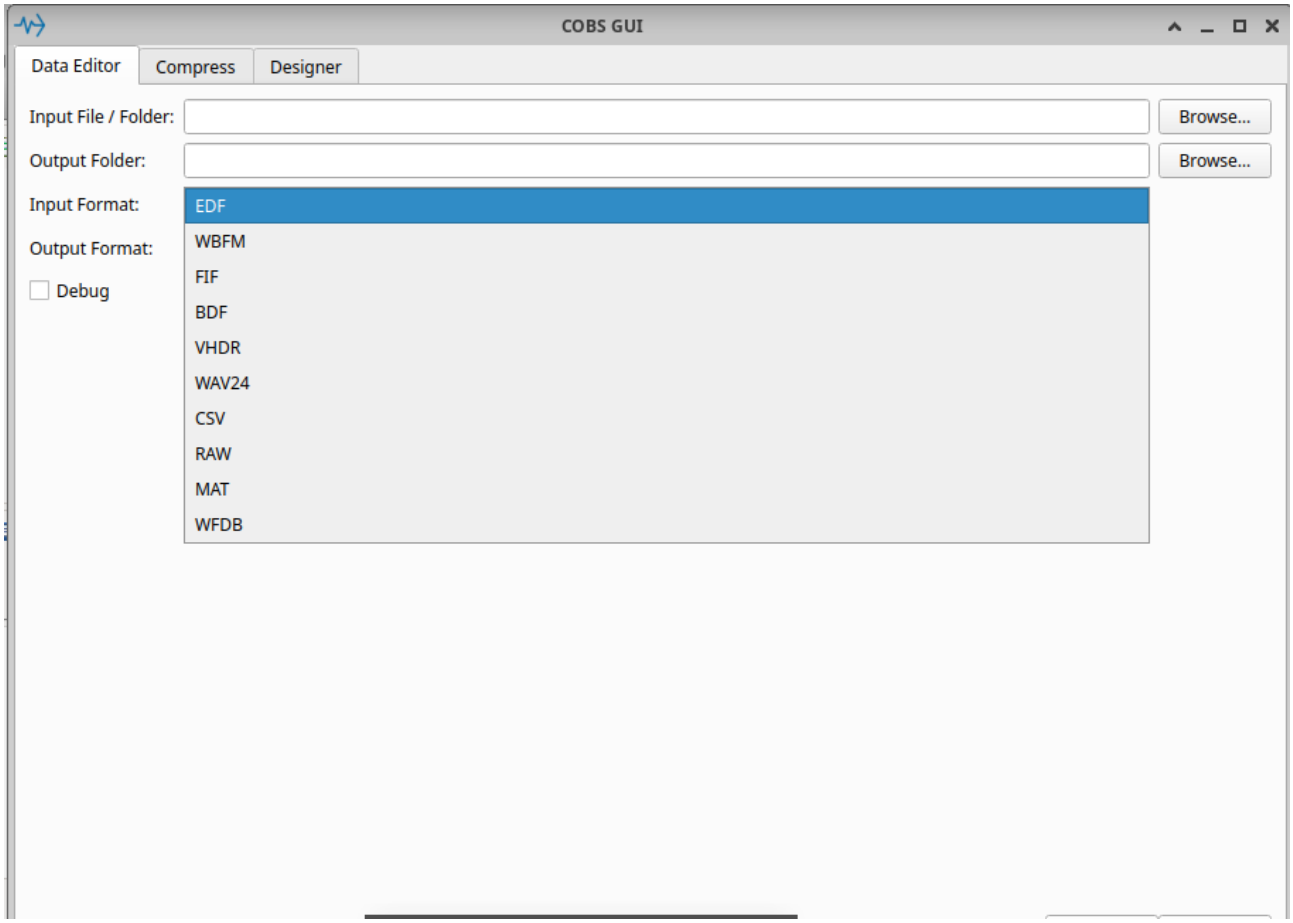

Figure 3: Data Editor tab: supported input formats.

4.2 The Designer tab

The **Designer** tab, shown in Fig. 4, represents the most sophisticated interface component, providing comprehensive tools for custom compression pipeline design and real-time performance evaluation. This workspace enables advanced users to experiment with different algorithm combinations, optimize parameters for specific signal characteristics, and export standalone compression implementations for successive use.

The pipeline configuration interface organizes processing stages into clearly defined sections corresponding to the four-stage compression architecture. Each section provides access to relevant algorithms and parameters, with dependency checking that ensures compatibility between selected components.

Users can load sample signals and observe the effects of different algorithm configurations immediately, without committing to full processing operations. Interactive sliders and numerical input controls enable fine-tuning of algorithm parameters while observing real-time effects on compression performance and signal quality. Interactive parameter adjustment enables users to optimize compression settings through real-time feedback and performance metrics displayed in the Designer tab. An example of pipeline configuration with a DPCM predictor, a Haar wavelet transform, and an arithmetic encoder is shown in Fig. 5. Other possible predictors are shown in Fig. 6.

Interactive plots display signal characteristics at each processing stage, showing prediction residuals, transform coefficients, quantization effects, and final compression results. The same tab also provides information on the compression ratio (CR) and distortion metrics (PRD, NPRD, and MAE) achievable on a test file using specific settings (see Fig. 7).

Once the compression pipeline has been fine-tuned (by selecting the predictor, transform, vertical and horizontal thresholds, etc.), a new pair of scripts - referred to as *Custom scripts* - can be generated by pressing the button *Export Scripts*. These Python scripts, named *mycompressor.py* and *mydecompressor.py*, embed all the selected settings and can be used as a ready-to-use codec for compressing and decompressing signals.

In particular, these scripts can be used to perform compression and decompression on a larger set of signals. To do so, they must first be loaded into the internal library managed by COBS, using a feature available in the **Compress** tab.

More detailed information can be found in deliverable D3.1 *"COBS: A New Toolkit for Compression of Biometric Signals and Traits"*.

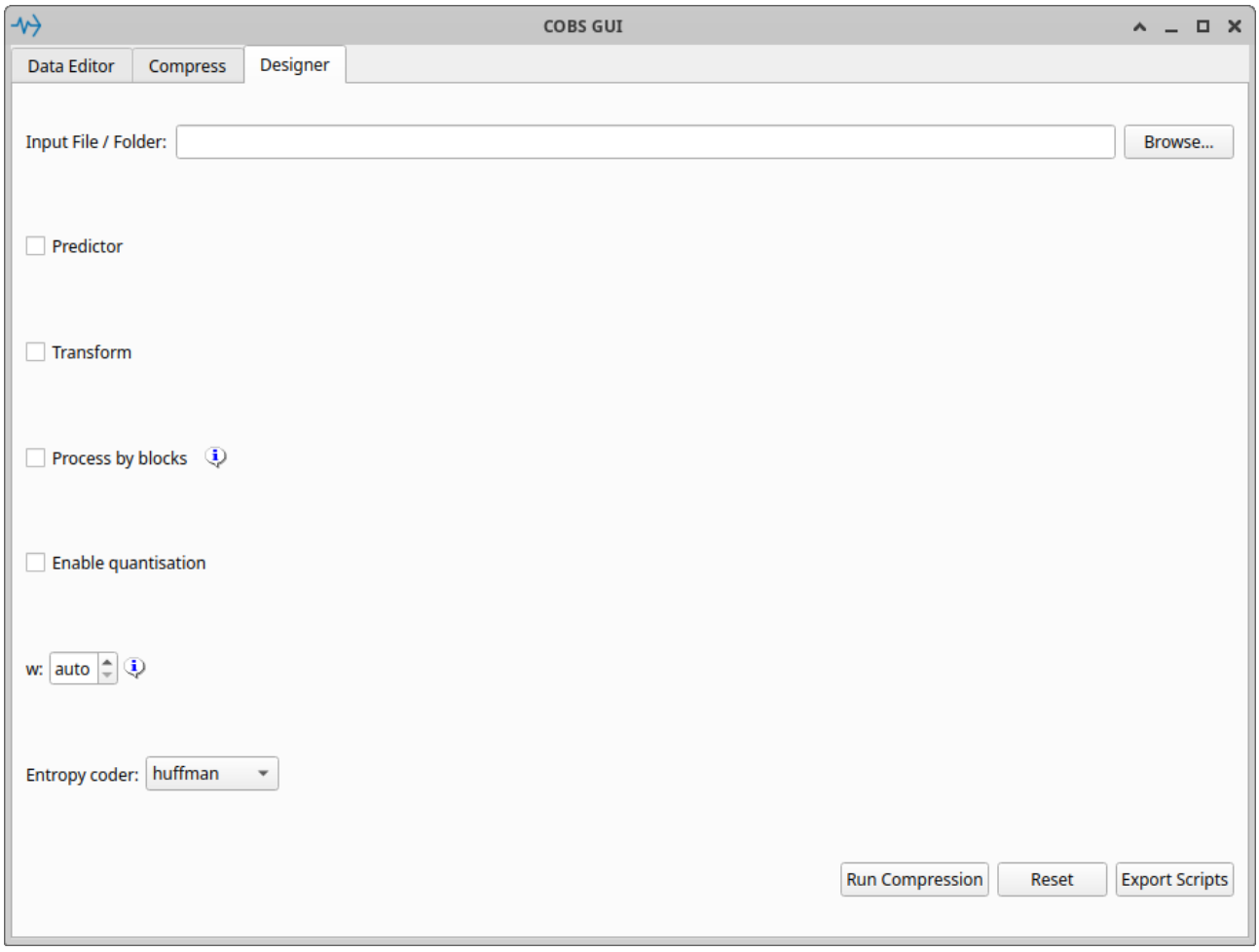

Figure 4: The Designer tab.

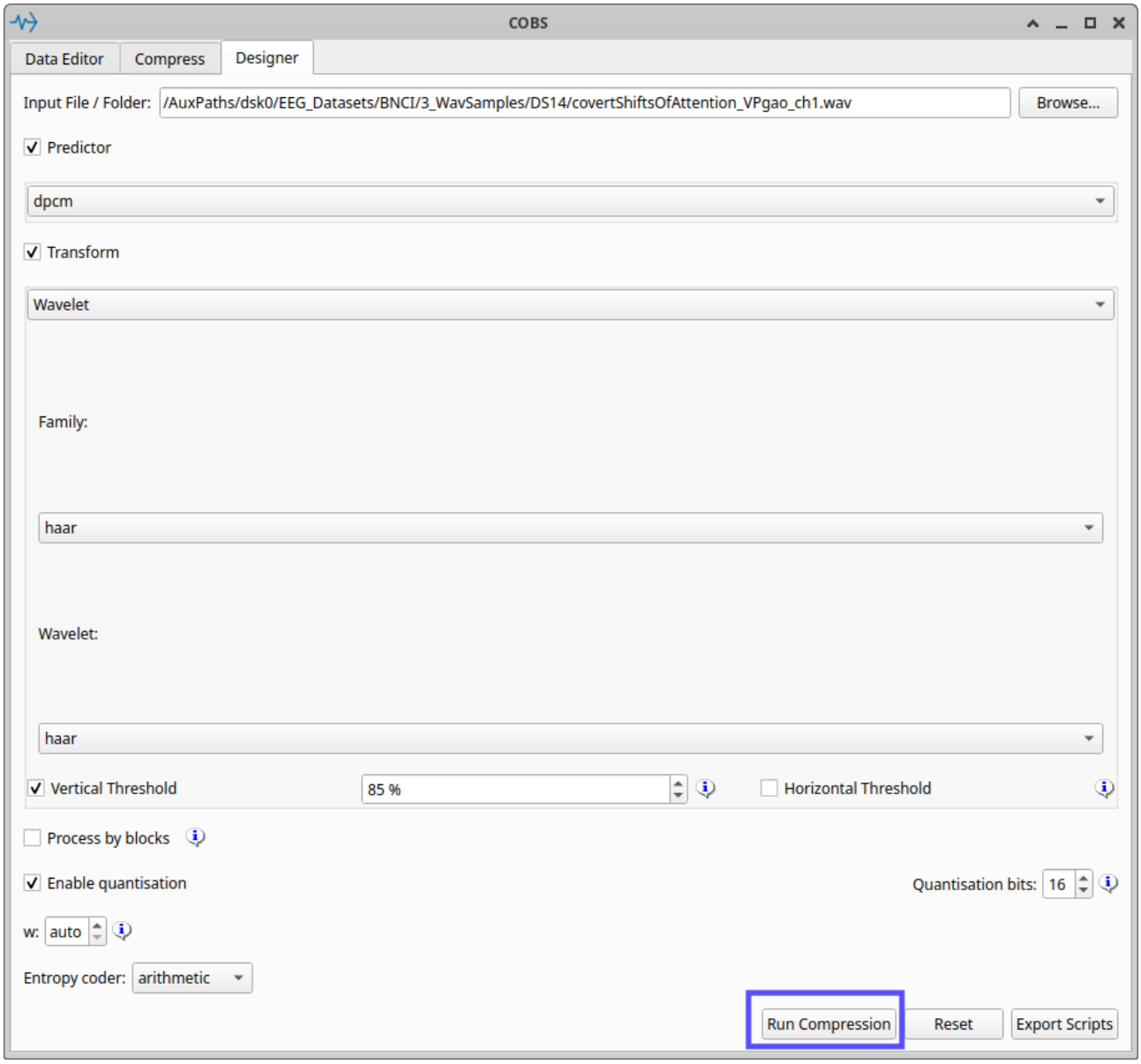

Figure 5: Designer tab: example of pipeline configuration.

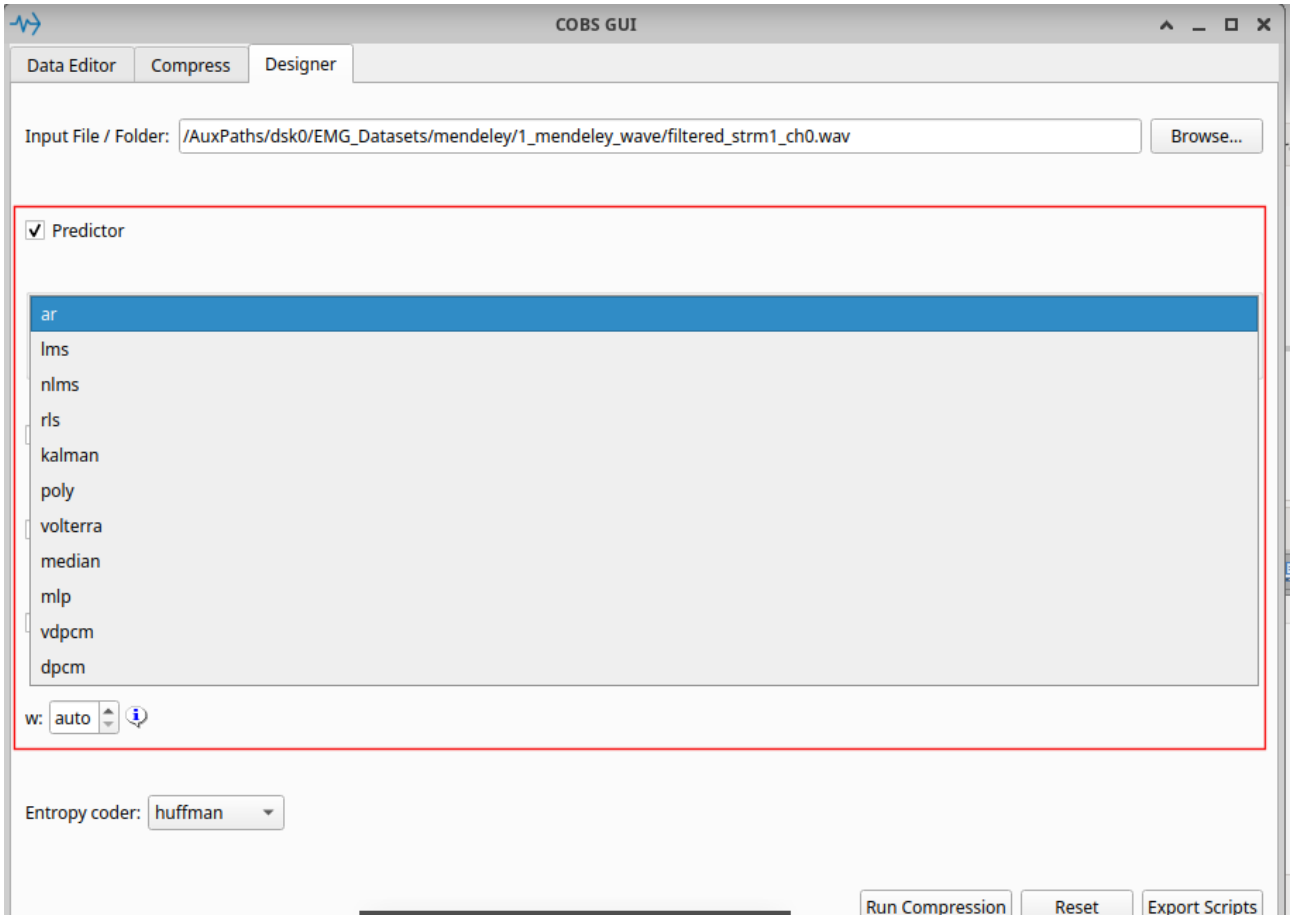

Figure 6: Designer tab. Selection of the prediction algorithm from a listbox.

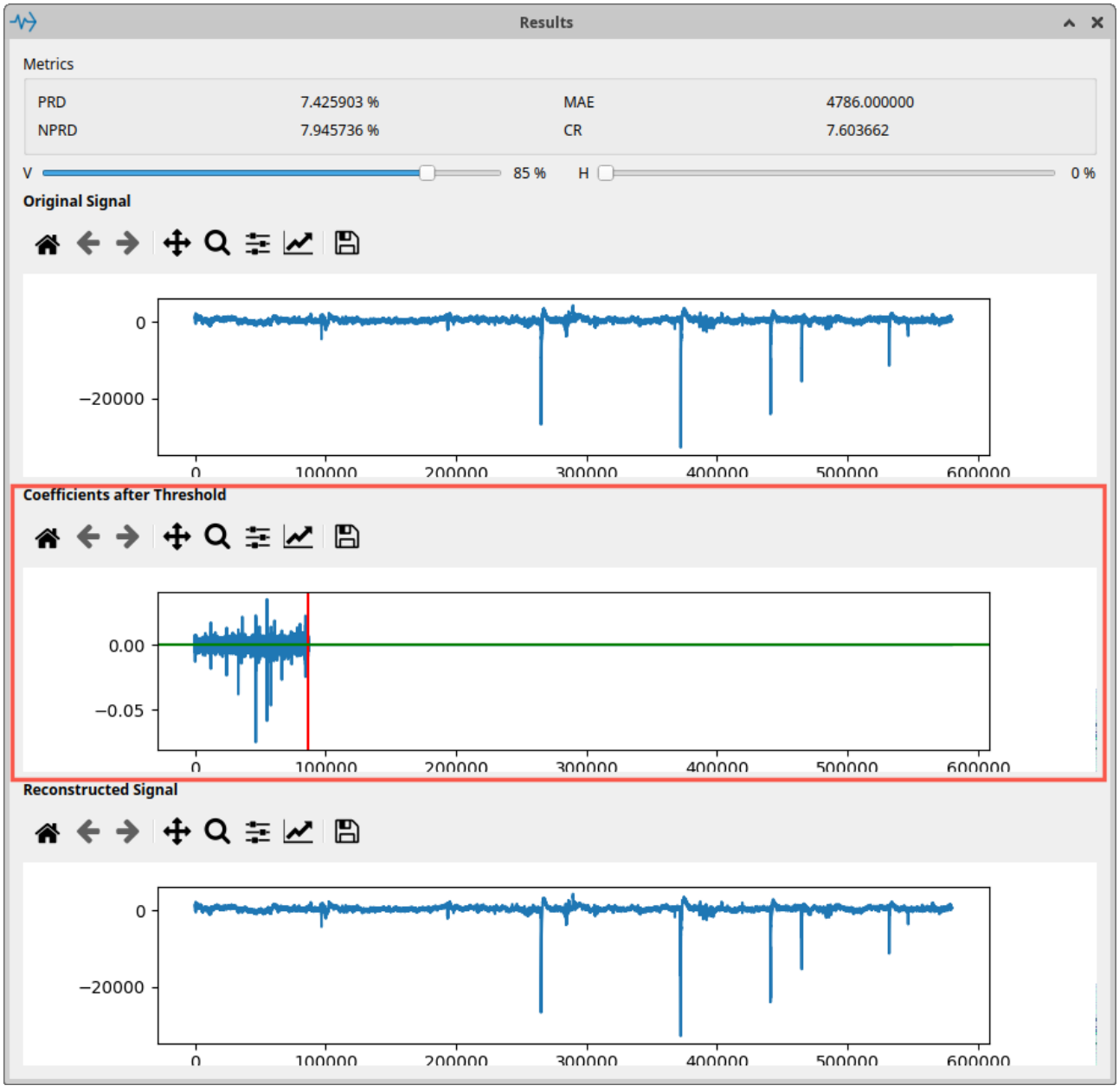

Figure 7: Designer tab: compression results.

4.3 Compress Tab

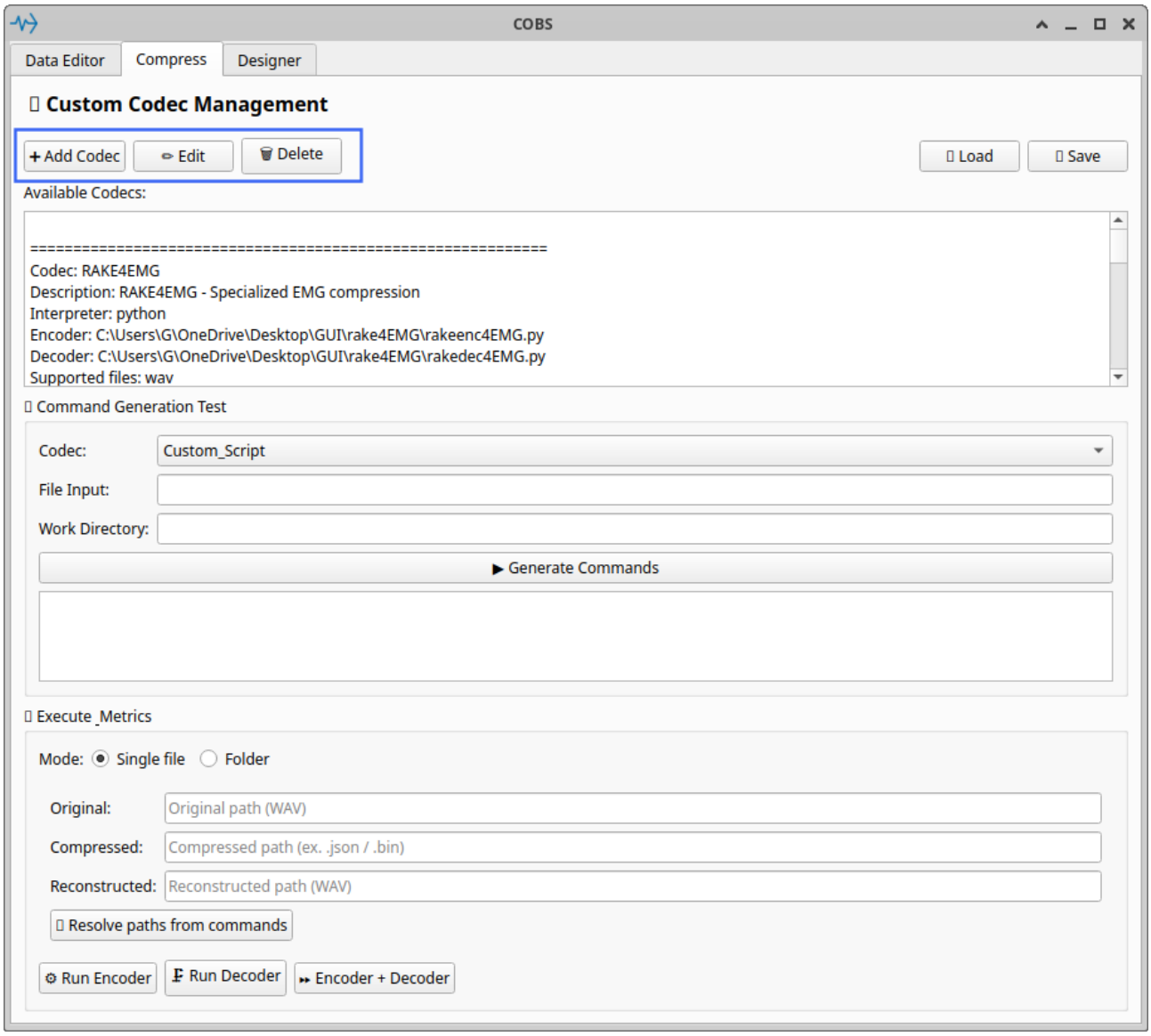

Figure 8: The Compress tab

The **Compress** tab, shown in Fig. 8, provides the tools necessary to apply a previously configured compression pipeline - or even external codecs - to a larger set of input signals. It allows the user to load custom scripts, select input/output folders, and execute batch compression and decompression operations.

This workspace emphasizes ease of use while maintaining access to comprehensive algorithm configuration options, making it suitable for both routine compression tasks and specialized processing requirements.

The tool supports two types of codecs:

- **Custom codecs**. These are codecs developed within COBS, using the functionalities provided by the Designer tab. They reflect configurations created and fine-tuned directly within the GUI.
- **External codecs**. Codecs developed with other tools can also be imported and used. Both

Python scripts and executable files are supported. Currently, three external codecs are integrated into COBS: RAKE4EMG and RAKELS (developed within the COBS4FUN project), and VAWC, developed by Fraunhofer and poised to become a standard for both DICOM and ITU.

For the codecs exported from the Designer tab, it is sufficient to select the Custom script option from the Algorithm field, specify the input folder containing the dataset files, the output folder for the compressed binary files, and the paths to the two Python scripts exported from the Designer.

An example of a Custom Codec configuration is shown in Fig. 9.

Once the codec has been configured, it can be used in the Compress tab to run tests on other files or on an entire dataset that was previously converted, as shown in Figs. 10 and 11.

The compression metric values are displayed in the interface (see Fig. 12) and are also saved in a CSV file.

COBS maintains a library of available codecs that is loaded into memory every time the tool is started. The codec library is managed as a .json file, simplifying the storage, editing, and sharing of codec configurations.

More detailed information can be found in the deliverable D3.1 *"COBS: A New Toolkit for Compression of Biometric Signals and Traits"*.

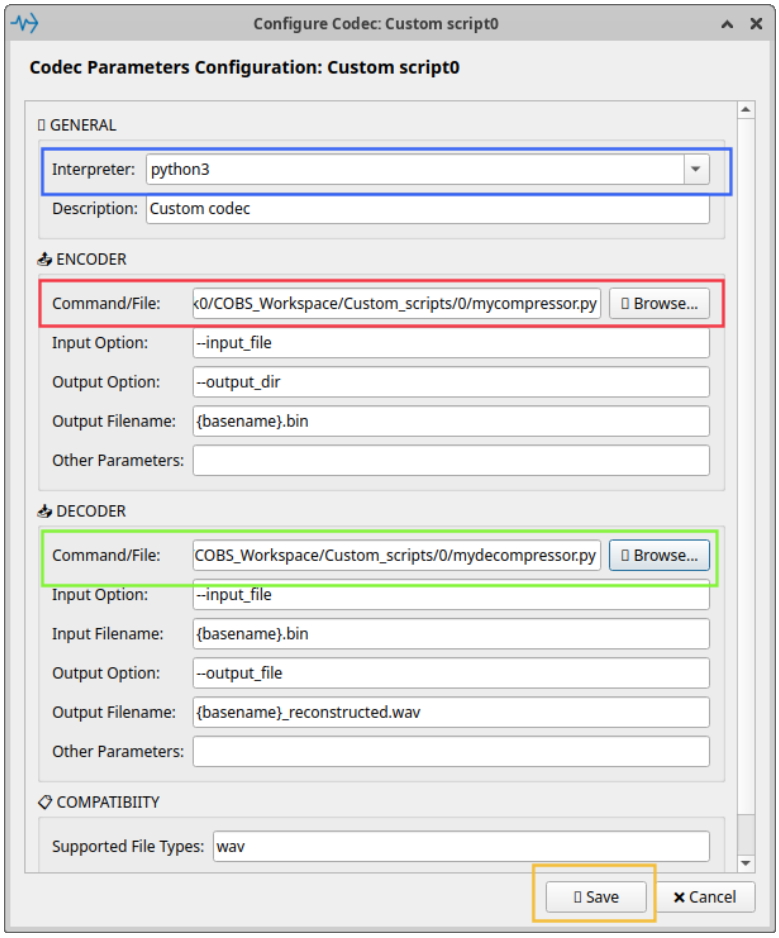

Figure 9: Compress tab. Example of Custom Codec configuration.

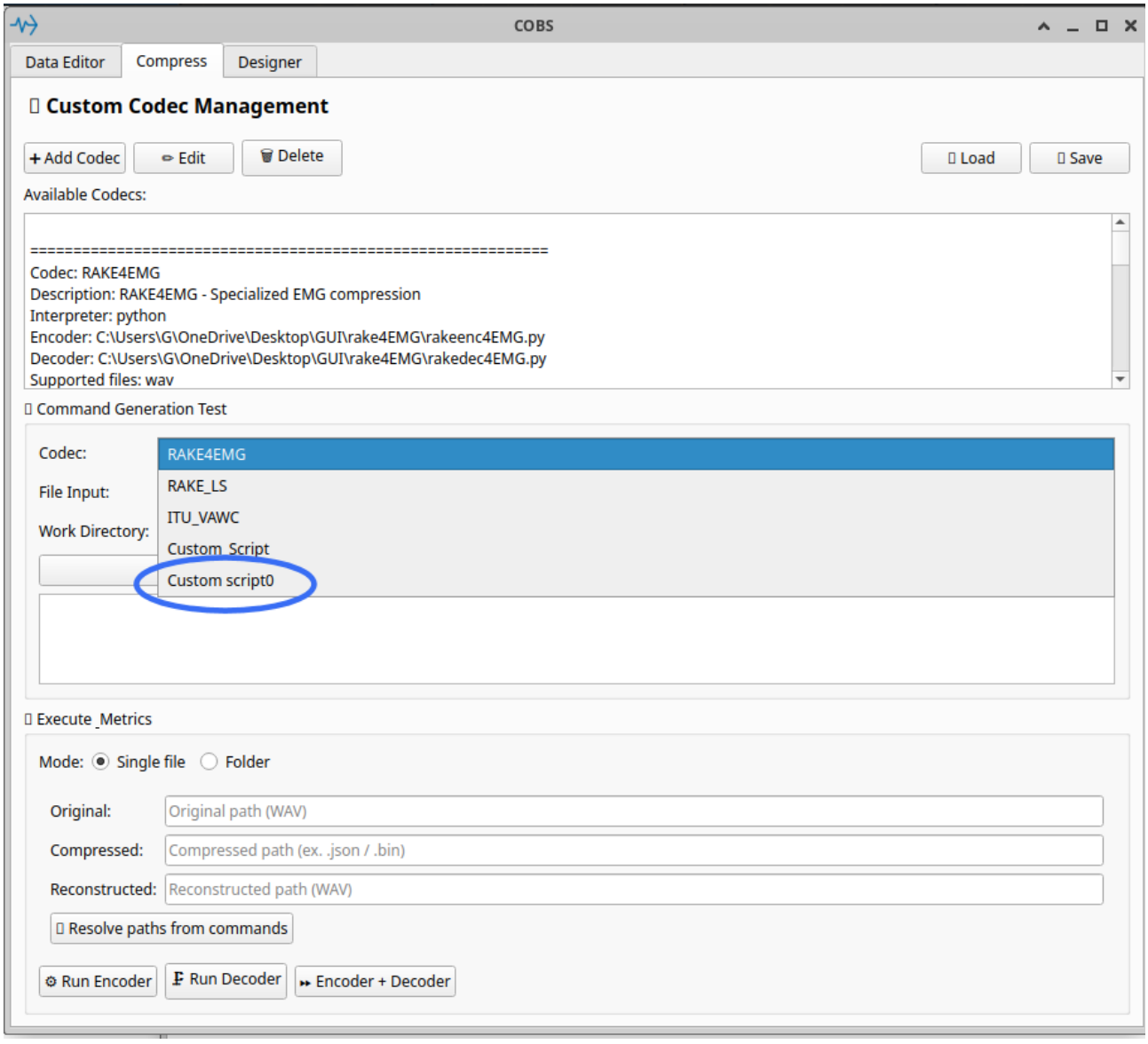

Figure 10: Compress tab: after configuration the new custom codec is now available in the drop-down listbox.

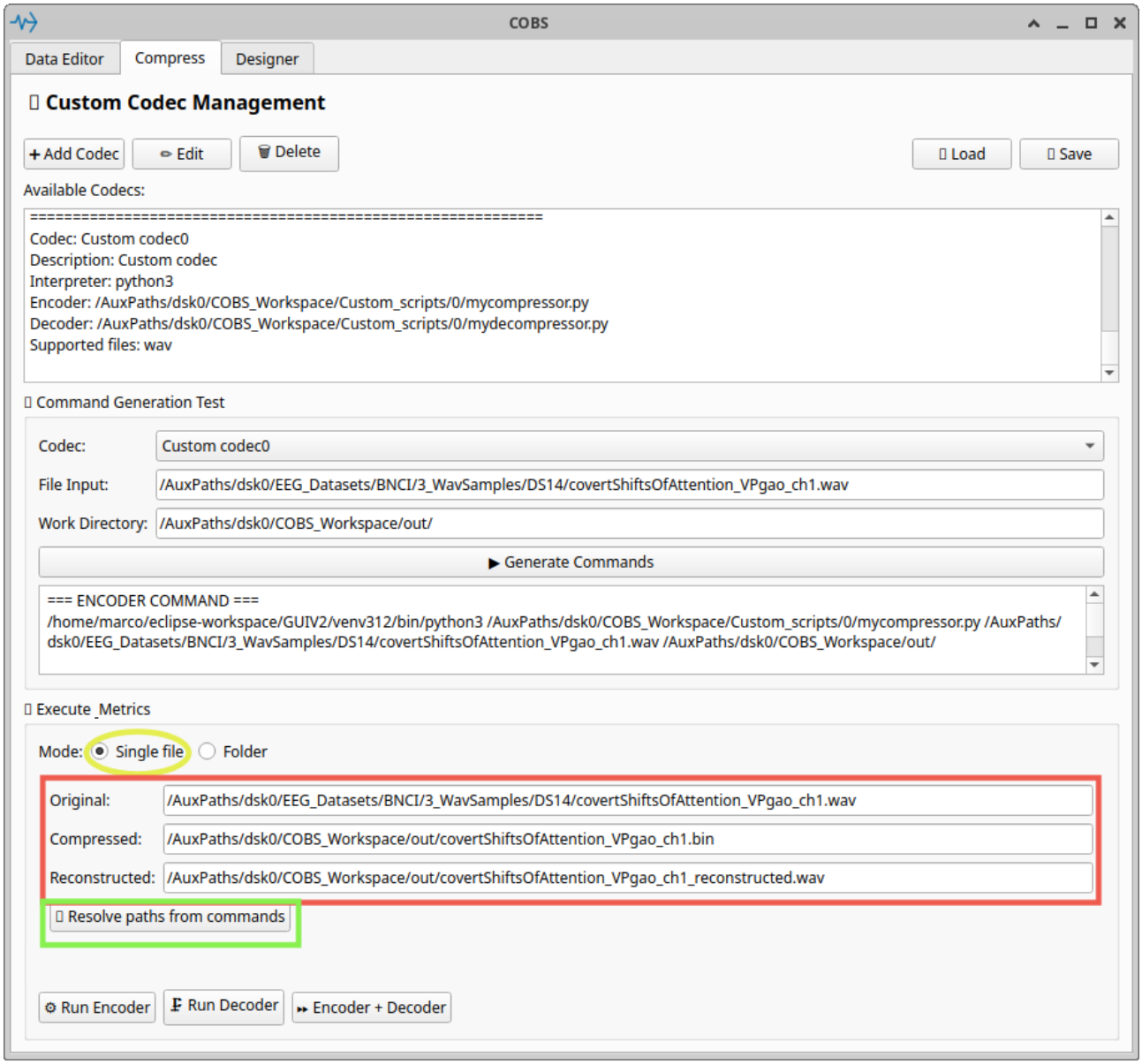

Figure 11: Compress tab: the custom codec is tested using a specific test file.

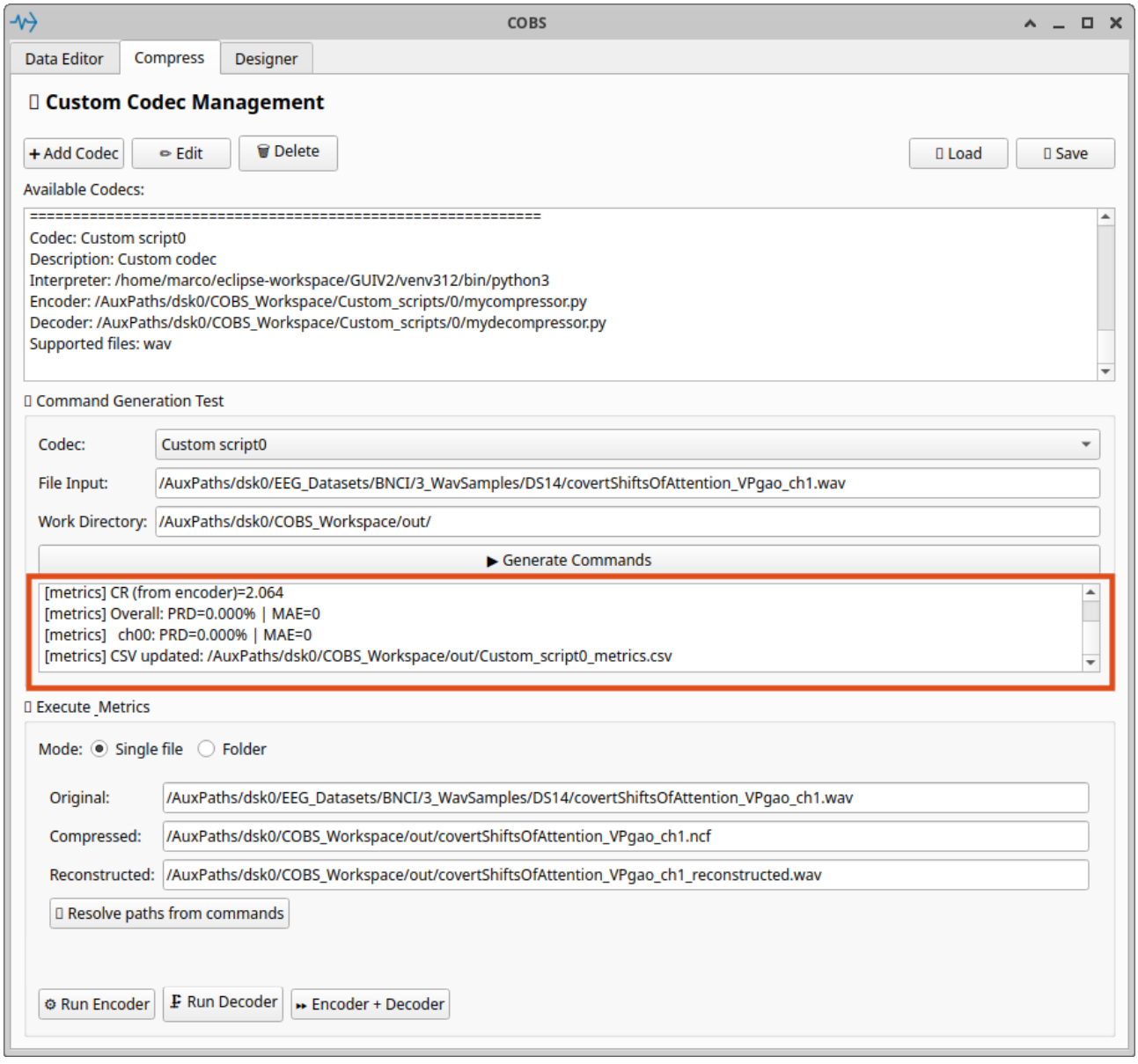

Figure 12: Compress tab: compression metrics reported at the end of encoding and decoding tasks.

5 Acknowledgement

This work was partially supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP E13C22001870001, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART"), and conducted within the framework of the cascade call project "Compression Of Biometric Signals for FUture Network Applications" ("COBS4FUN" - CUP C49J24000250004).