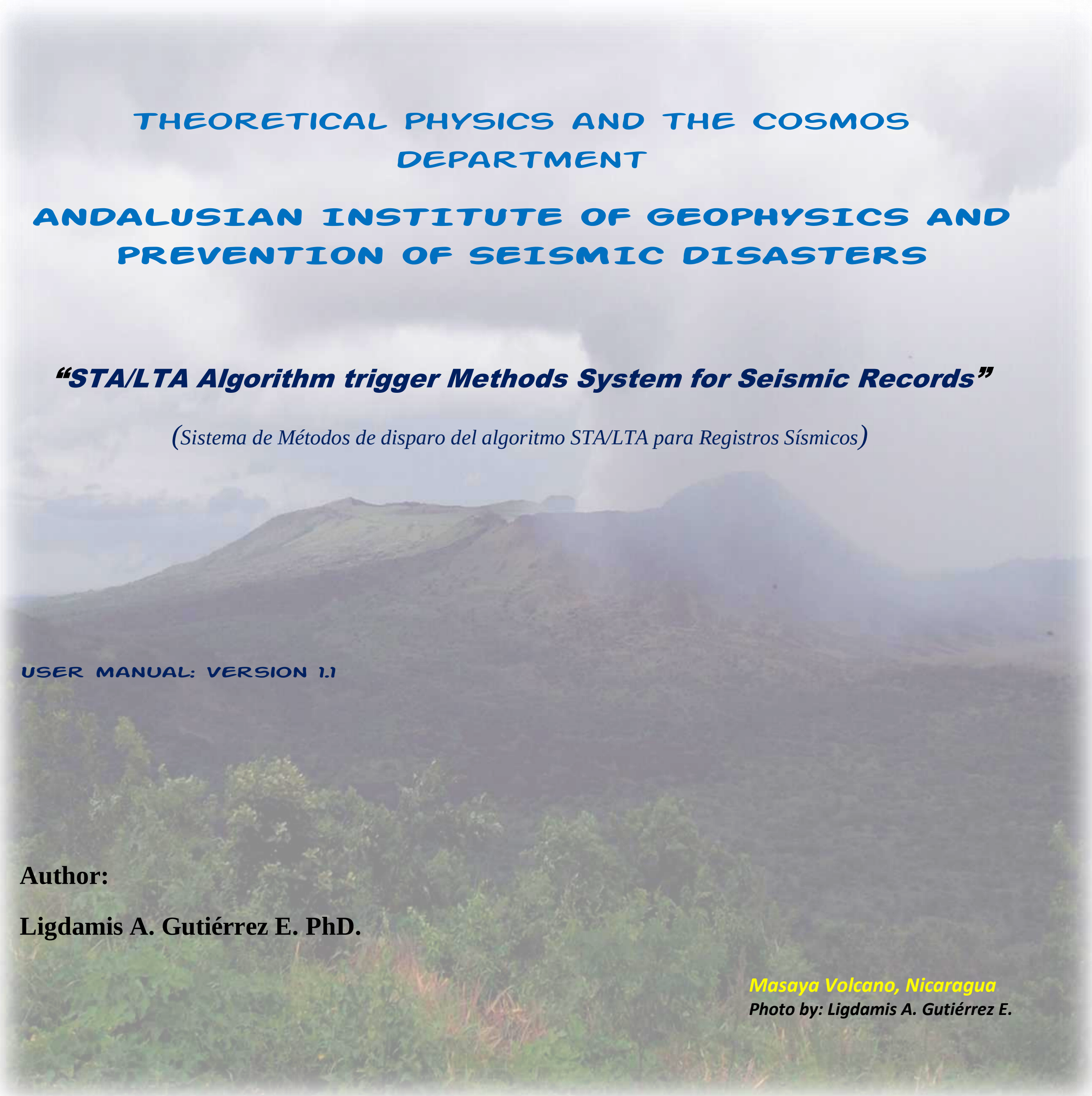
GRANADA UNIVERSITY (UGR)

Theoretical and Cosmos Physics Department

Geophysics Andalusian Institute



THEORETICAL PHYSICS AND THE COSMOS DEPARTMENT

ANDALUSIAN INSTITUTE OF GEOPHYSICS AND PREVENTION OF SEISMIC DISASTERS

"STA/LTA Algorithm trigger Methods System for Seismic Records"

(Sistema de Métodos de disparo del algoritmo STA/LTA para Registros Sísmicos)

USER MANUAL: VERSION 1.1

Author:

Ligdamis A. Gutiérrez E. PhD.

Masaya Volcano, Nicaragua
Photo by: Ligdamis A. Gutiérrez E.

Granada, Spain 2021 - 2023

General Index

1.- Introduction

Back to Index

The "*STA/LTA Algorithm Trigger Methods System for Seismic Records*" is a user-friendly interface designed for the easy and efficient management of the STA/LTA[1] (*Sistema de Métodos de disparo del algoritmo STA/LTA para Registros Sísmicos*) algorithm, which is the most commonly used method for calculating the arrival of P and S waves in seismic records. The algorithm uses the STA/LTA ratio to identify an increase in the short-time STA window relative to the preceding long-time LTA window, as determined in a seismic trace.

The main interface provides a version of the system in English. The appendices include information about the folder structure and its contents. The easy access to two STA/LTA triggering methods: "*recursive*" and "*classic or typical*", combined with the ability to use filtering techniques, provides a reliable automatic tool to assist the operator in identifying the arrival of P and S waves in seismic signals.

The application, through its incorporated libraries, allows for the reading of various seismic formats such as SAC, MSEED, GSE2, EVT, WAV, among others. It can then apply various filtering techniques and the STA/LTA triggering algorithm, automatically providing added value to the operator's knowledge for determining the onset of an event and thereby establishing the arrival of the seismic wave with greater speed and accuracy than manual methods. In this new version, the creation of ".txt" files is added, showing the results of the STA and LTA arrival time values (blue lines), as well as a final count of the marked events.

Both the first and second versions of this system consist of a single interface that includes tools for signal filtering and the "recursive" and "classic or typical" methods of the STA/LTA algorithm. Additionally, the system provides the capability to save graphical results in various formats, such as PNG, JPG, EPS, PS, PDF, RAF, TIF, among others.

The module and the entire system have been developed in Python, version 3.8.6. (The set of libraries is compatible with version 3.10.10). Additionally, a series of open-access libraries are included, which, in conjunction with Python, enable the use of graphical and analytical tools, providing ease of use and enhancing computational power for the user. Some of the main elements and libraries used are listed below:

- **Matplotlib**: Used for creating static, animated, and interactive visualizations in Python. (https://matplotlib.org/stable/users/index.html)
- **NumPy**: A library for numerical operations in Python. (https://numpy.org/doc/stable/user/quickstart.html)
- **PyQt5**: A tool that links with the graphical library Qt5 in C++ (https://pypi.org/project/PyQt5/)
- **Obspy**: A Python toolbox for seismology. (https://docs.obspy.org/)
- **Tkinter**: Graphical User Interface (GUI) (https://docs.python.org/3/library/tkinter.html)

Another key feature of the system is its definition as a cross-platform application, meaning it can operate on various platforms or operating systems, such as Windows (7, 8, 10, 11) in both 32-bit and 64-bit versions. It also supports Linux systems, such as Ubuntu and other similar systems (Debian, Red Hat, Fedora, SUSE, etc.), macOS, and Android for tablets and mobile devices (with Python appropriately adapted for these devices).

NOTE: In the appendices of this document (*as well as in the Readme.txt and Initial_requirements.txt files*), you can find general information on installation for Windows and Linux systems, as well as guidelines for installing the main programs and additional libraries required by Python to properly execute the developed programs in its environment.

[1] Allen, R. (1982). Automatic phase pickers: their present use and future prospects. *Bulletin of the Seismological Society of America*, *72*(6B), S225-S242.

[Back to Index](#)

2.- Initial Screen of the System

In the appendices of this document and in the "**README.txt**" file included in the "***Documents***" folder, you will find instructions for installing the system on Windows (*the process on Linux systems is similar*). Essentially, you need to perform two actions:

a) Copy the "Set_tools_System_1_1" folder to "My Documents" on Windows.
b) Copy the "Set_tools_System_1_1.bat" file to the Windows desktop.

Additionally, there are instructions for installing the necessary Python libraries on the system. Once "Set_tools_System_1_1.bat" has been copied to the desktop, you need to right-click on it and select "Run as administrator."

Fig. 1 Popup Window when Right-Clicking on the "Set_tools_System_1_1.bat" File

In the window that opens, click the "Yes" button when prompted with "*Do you want to allow this app to make changes to your computer*?" This is a warning message. However, the application does not make any changes, so you should trust its execution.

Upon clicking "*Yes*," the following command window opens, welcoming you to the system.

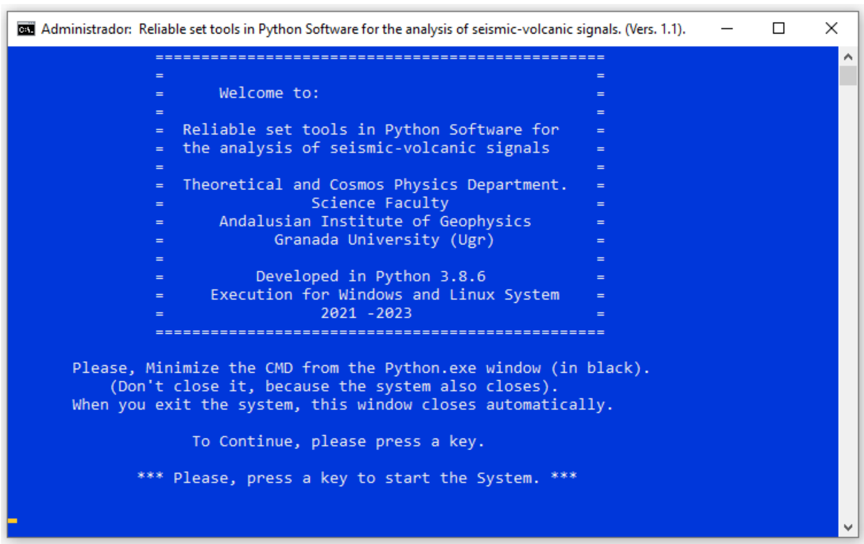


Fig. 2 Welcome Screen and Instructions for Loading the System.

After reading the information in the window, you simply need to press any key to access the system's initial screen. The folder should already be copied to "***My Documents***," and the "**Set_tools_System_1_1.bat**" file contains all the loading instructions.

The system's initial screen is "**Menu.py**." It appears when any key is pressed on the Welcome screen. Additionally, the Python command window or console is displayed, similar to the following:

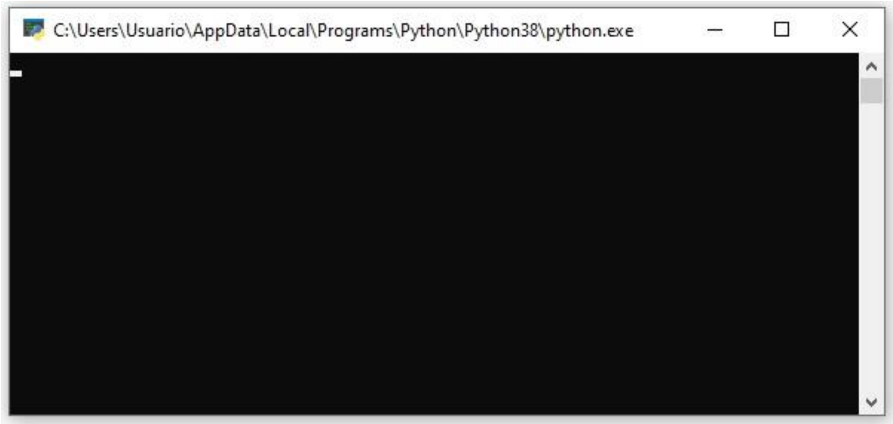


Fig. 3 Python Console (CMD) Window (*Should be minimized*)

To avoid obstructing the view, you can and should "minimize" this screen. Do **not** close it, as this would also close the system's startup window. Once you have finished working with the system, this window will close automatically. The initial presentation screen of the system (the module menu) is as follows:

NOTE: When pressing or clicking on a module button, the startup window will close and the module window will open (*this may take a little time depending on the PC's memory. It is recommended to have at least 8 GB of memory in the system, with 16 GB being ideal*).

Fig. 4 **Main Menu Screen.** The module to be worked on is highlighted. Module 2 (*STA/LTA triggering methods*).

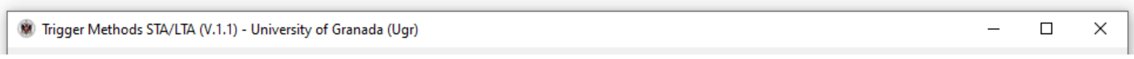
Back to Index

2.1.- Elements of the Initial Screen

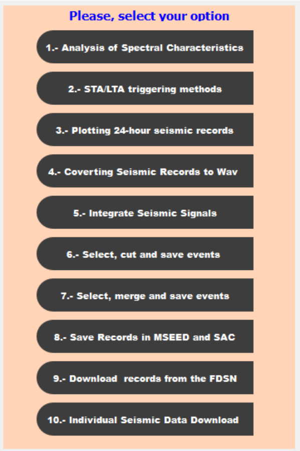
As observed in the previous figure, the initial or presentation screen is a simple window composed of:

a) A top toolbar with basic information about the module.
b) On the left side, there are 10 execution buttons or command buttons for each module of the system.
c) At the bottom, there is a command button that allows for exiting the system.
d) Additionally, it features a background image representing a volcano (Masaya in Nicaragua), and three images with the logos of the University of Granada, the Andalusian Institute of Geophysics, and the Department of Theoretical and Cosmic Physics.

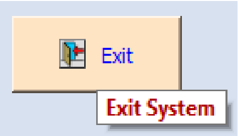
a) At the top, the icon of the University is visible, along with the module title and a reference to the University of Granada (UGR).



b) On the left side, there are 10 execution buttons or command buttons for each module of the system. When the mouse pointer is placed over each button, it is highlighted in white to indicate that it is being selected. Clicking on a button closes the startup menu window and opens the window for the indicated module (this may take a little time depending on the PC's memory).



c) At the bottom, there is a command button: **Exit**. When the mouse pointer is placed over each button, a text appears indicating the action of that button (Exit System, Start System).



If you click the "Exit" button, a window will appear asking the user if they are sure they want to leave the system.

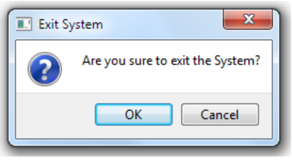


Fig. 5 Text Box Indicating Whether You Want to Exit the System.

If you click "**OK**," the screen will close and the system will exit. If you click "Cancel," you will remain on the initial screen.

Back to Index

3.- Analysis Screen.

The "*main analysis screen*" is the primary interface of the module, where activities related to reading records, filtering and analyzing, calculating, and presenting the STA/LTA triggering methods (classic and recursive) for seismic records are carried out. This screen is composed of the following 4 parts:

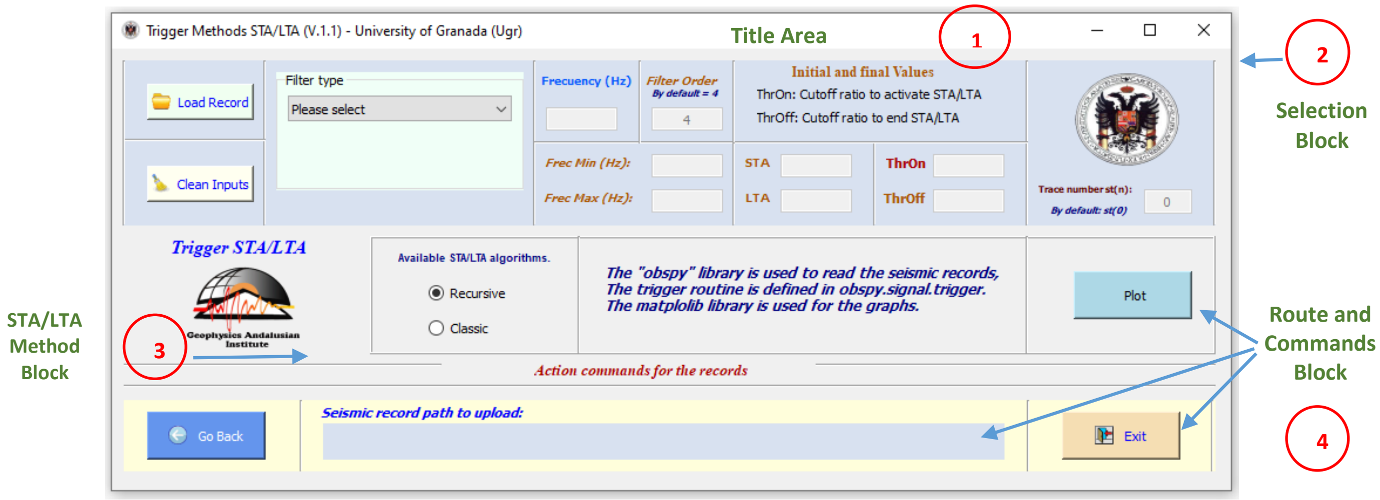


Fig. 6 Elements of the System Analysis Screen.

1. Title Area
2. Selection Block: a) Command Buttons: Load Record, Clean Input, b) Filter Types, c) Data Entry.
3. STA/LTA Triggering Methods Block (Recursive and Classic or Typical).
4. Path and Commands Block: a) Physical Path of the Record to be Analyzed, b) Command Buttons (Plot, Go Back, Exit)

The screen is composed of various elements for use. At the top, you can see the program name, icon, and university name, as the title (1).

The elements that make up the main screen are detailed below.

3.1.- Elements of the Analysis Screen.

In addition to number (1), the 3 elements of the analysis screen are distributed into three main blocks, numbered from (2-4) in the red circles.

3.1.1.- Selection Block. (2)

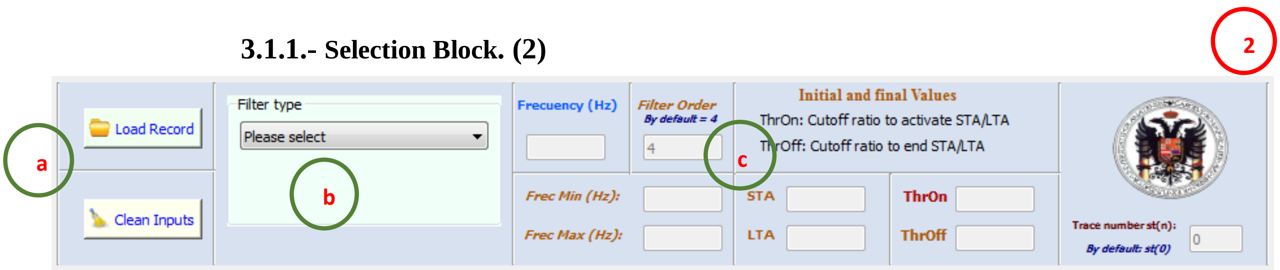
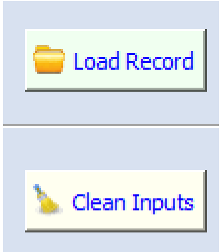


Fig. 7 Selection Elements Block. In The green circles: a) Command Buttons: Load Record, Clean Input, b) Filter Types, c) Data Entry.

Back to Index

3.2.- Elements of the selection block.

This block is configured (*green circles in the previous figure*) by, first, the section that groups the action buttons for loading the record and cleaning the data: (*Command buttons: Load Record and Clean Input*).



When hovering the mouse pointer over these two action buttons, a message displays the actions they perform.

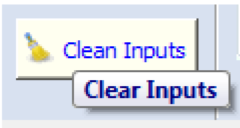
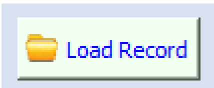


Fig. 8 Record and Data Cleaning Buttons. The action messages for each button can be seen.

As observed, the "*Load Record*" button performs the search and loading of seismic records in various formats. The "*Cleaning*" button clears or deletes the input elements, as well as closes any existing graphs, returning the analysis screen to its initial state, ready for a new search and analysis of seismic events.

3.2.1.- Load Record Button.



The action of the "*Load Record*" button allows, upon clicking, to open a file explorer window (*by default, the path is set to the root directory "**C**" of the PC*), displaying options for the various types of formats to search for and enabling the search within the computer's directory. This can be seen in the following screen.

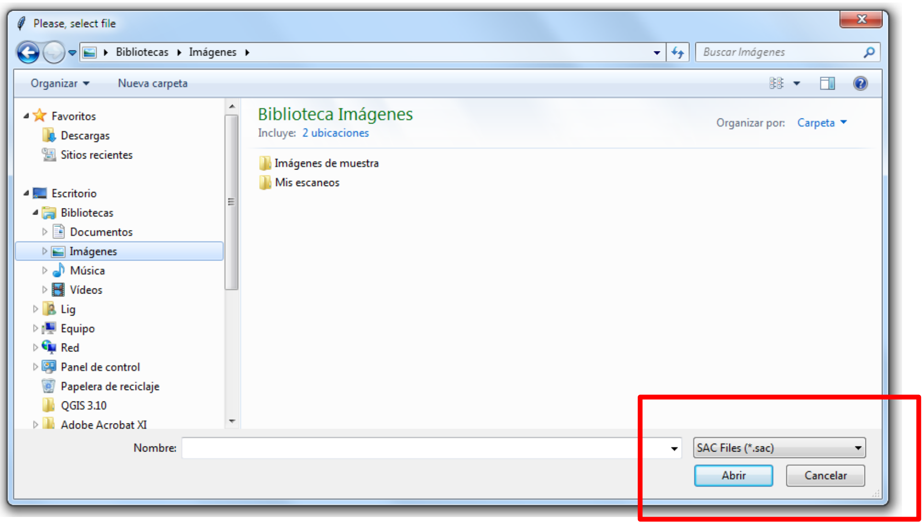


Fig. 9 Record Selection Screen.

On this screen (*the language is determined by the operating system*), records are selected according to the desired format (*red box*), such as SAC, MSEED, GSE2, EVT, etc. This is made possible through the use of the seismic format reading library "**Obspy**".

Back to Index

Once selected, click the "Open" button, and the file will be loaded onto the analysis screen. Otherwise, click the "Cancel" button to return to the analysis screen. The process of selecting a record is shown in the following screen.

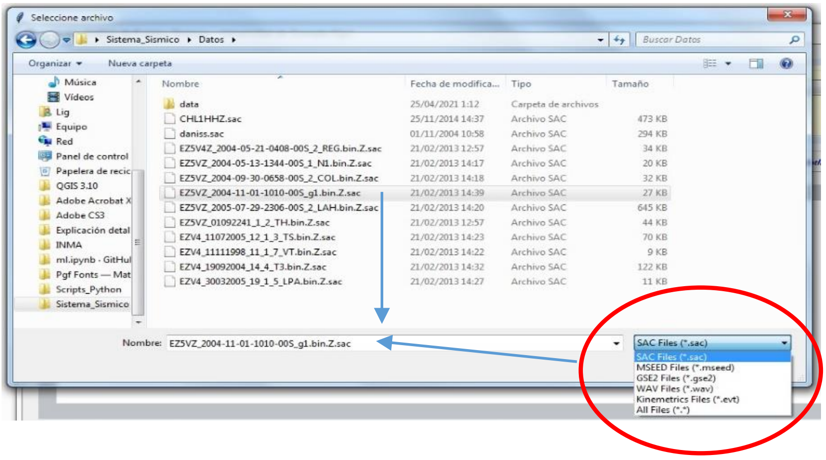


Fig. 10 Example Screen for Selecting a "SAC" Format Record.

On the screen, in the bottom right corner, indicated by the red circle and displayed by the arrow, is a list of the most common seismic formats supported and/or used by observatories and institutes worldwide (*SAC*, *MSEED*, *GSE2*, *WAV*, *EVT*).

When a specific type is selected, records are displayed according to that format. For example, the "SAC" files stored in "Data_examples" are shown. By clicking on the desired record, as seen, it is placed in the "Name" box. At this point, clicking the "Open" button, as shown in the previous screen, will load the path or location of the record in the system. This path will be displayed in the "File Path" box, located at the bottom of the analysis screen.

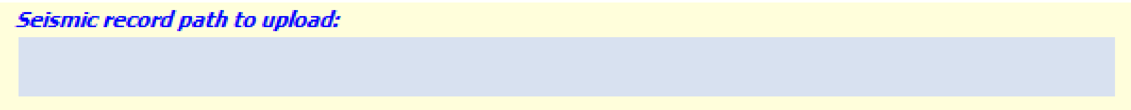
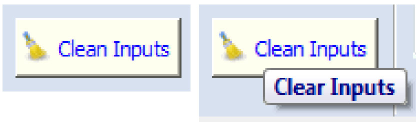


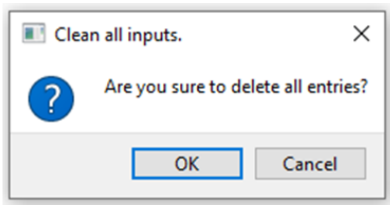
Fig. 11 File Path Box, displaying the location of the record.

This is an important aspect, as it determines whether the physical file location of the stored record on the computer can be located for analysis later. If the file is invalid or cannot be found, a validation window will appear to indicate this (*see Fig. 25, Page 20*).

3.2.2.- Cleaning Button.



Clicking this button clears all entries on the screen. It will clean the data entry text boxes and the path or folder where the record to be downloaded will be stored, and it will delete all active entries with data at that moment.



It restores the initial values of the main interface (*see Fig. 6*). When you place the mouse pointer over it, a message appears indicating its function. Clicking the button opens a window asking the user if they are sure they want to delete the data entries. If confirmed, it clears all entries and returns the interface to its initial state. Otherwise, the current entries remain in the interface.

Back to Index

3.2.3.- Filter Type Selection.

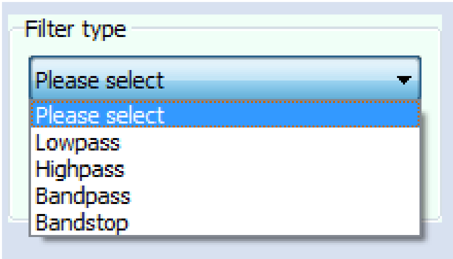


Fig. 12 Filter Type Selection via List

The previous figure shows a dropdown list with various types of filters available for analyzing records. As observed, it includes filter types such as *Lowpass*, *Highpass*, *Bandpass*, and *Bandstop*. Selecting each of these items will activate one or more of the checkboxes in the continuous section, which correspond to the data inputs required for performing the calculations. At the initial value "*Please Select*", the inputs return to their default values, as does the STA/LTA triggering method of the algorithm.

3.2.4.- Input Selection.

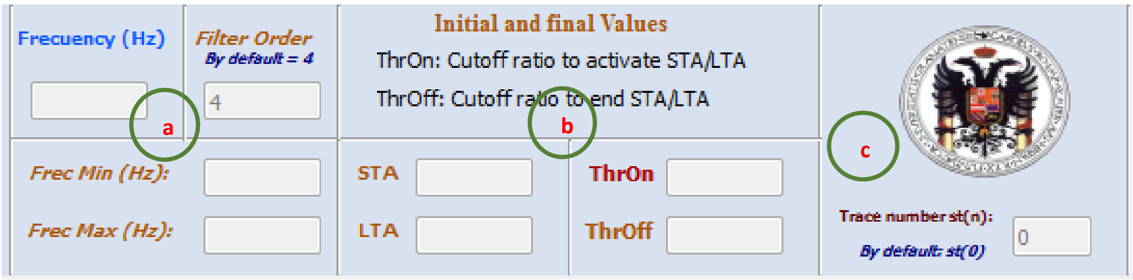


Fig. 13 Selection of the input to calculate and the trace number of the record (default = 0).

The inputs correspond to the selected filter type specified by the operator (default filter order = 4). Initially, all checkboxes are deactivated and will only be activated based on the chosen filter type. The trace number is set to default = 0. If there are multiple traces, such as in MSEED or SEISAN records with three components or various values, the desired trace number, starting from zero, should be entered in this text box. The inputs are validated to accept only numbers, including decimals and negative values. The activation/deactivation responds to the following:

I) The checkboxes in the area (a): {Frecuency (Hz), Filter Order, Frec. Min (Hz), Frec. Max (Hz)}

1.- *(Lowpass), and (Highpass) filter.*

Activate: {Frecuency (Hz), Filter Order}.

Deactivate: {Frec. Min (Hz), Frec. Max (Hz)}.

2.- *(Bandpass) and (Bandstop) filter.*

Activate: {Frec. Min (Hz), Frec. Max (Hz)}.

Deactivate: {Frecuency (Hz), Filter Order}.

II) The checkboxes in area (b): {STA, LTA, ThrOn, ThrOff} are activated with any type of filter.

III) The checkboxes in area (c): {Trace number st(n)} are activated with any type of filter.

Back to Index

3.2.5.- *Lowpass*[2] *filter type.*

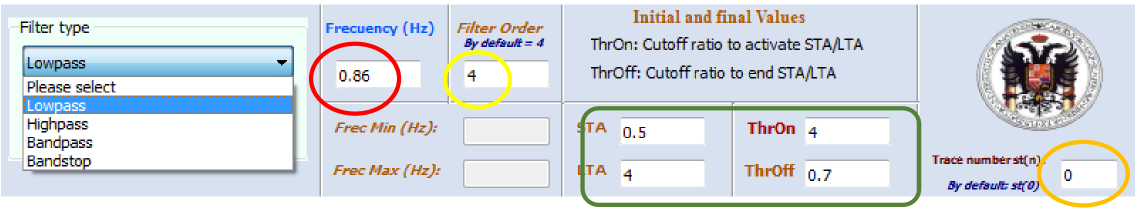


Fig. 14 Example of Lowpass Filter Selection. In the red circle: "**Frequency**" input, in the yellow circle: "**Filter Order**", in the orange circle: "**Trace Number**", and in the green rounded rectangle: (**STA, LTA, ThrOn, ThrOff**).

When selecting this type of filter, the following checkboxes are activated:

a) Filter Frequency (Hz): Floating-point value. Here, you should enter a valid frequency value for the filter calculation (Red circle). In the example, the frequency value entered is "0.86 Hz.".

b) Filter Order: Integer value. This field should be set to a valid filter order for the calculation. Typically, this is order 2 or 4 (by default, it is set to 4 for all analyses) (Yellow circle).

c) STA (Short-Time-Average): Floating-point value. Represents the short-term dynamic time window average, measuring the "instantaneous" amplitude of a seismic signal to detect potential earthquakes. In the example, the value is "0.5.".

d) LTA (Long-Time-Average): Floating-point value. Represents the long-term dynamic time window average, measuring the "average amplitude of current seismic noise." In the example, it is "4.".

e) ThrOn: Floating-point value. Indicates the cutoff ratio to activate the STA/LTA algorithm. It determines the "maximum parameter for peaks and the arrival of the P-wave." Its value generally matches or is in the same range as the "LTA" value. In the example, the value is "4.".

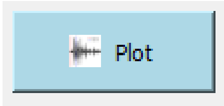
f) ThrOff: Floating-point value. Indicates the cutoff ratio to end the STA/LTA algorithm. It determines the "minimum parameter for peaks and the arrival of the S-wave." Its value generally matches or is in the same range as the "STA" value. In the example, the value is "0.7.".

g) Trace number st (0): Integer value. Determines the trace number to be analyzed. By default, and in formats like "SAC," this value is set to "0." For formats containing multiple traces, such as "MSEED" or "SEISAN," the user selects the trace number to analyze.

The "Frec. Min (Hz)/Minimum Frequency" and "Frec. Max (Hz)/Maximum Frequency" checkboxes remain inactive.

---

[2] The Lowpass filter blocks high-frequency signals and allows low-frequency signals (frequencies below the cutoff frequency) to pass through.



Once the actions are performed and the desired triggering method (Recursive - Classic or typical) is selected, click the “Plot” button. If there is an issue with the input (such as invalid entry, nonexistent record, or out-of-range format) and you attempt to plot by clicking the “Plot” button, a validation dialog box will appear. This dialog box will inform you of the necessary action to correct the issue. These dialog boxes ensure that the program can continue running without severe system interruptions due to data issues. The validations for incorrect entries are as follows:

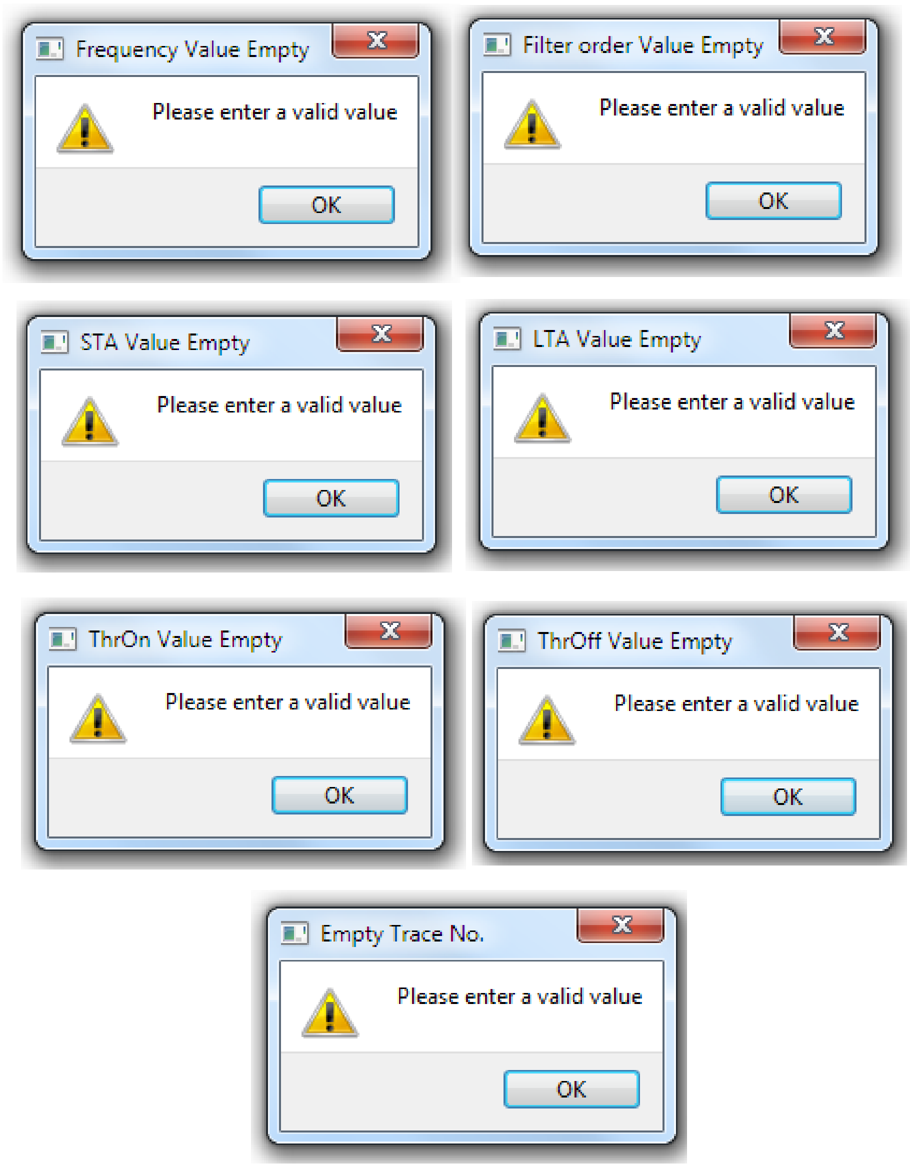


Fig. 15 Validation of entries for the *Lowpass* filter.

Once the inputs are corrected by entering a valid value or one within the permitted range, the program can continue and display the corresponding graph.

3.2.6.- *Highpass*[3] *Filter Type.*

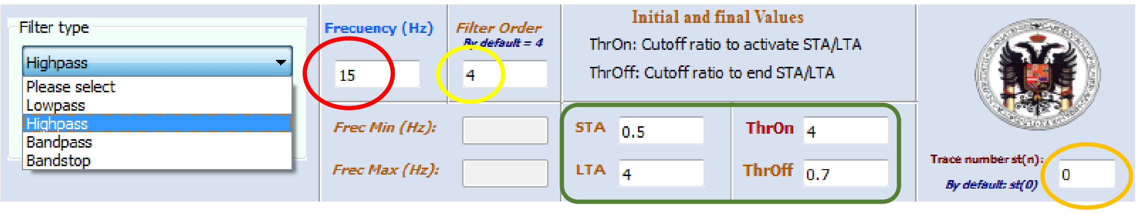


Fig. 16 Example of *Highpass* Filter Selection. In the red circle: "**Frequency**" input, in the yellow circle: "**Filter Order**", in the orange circle: "**Trace Number**", and in the green rounded rectangle: (**STA, LTA, ThrOn, ThrOff**).

Similar to the previous filter, the following checkboxes are activated:

a) Filter Frequency (Hz): Floating-point value. Here, you should enter a valid frequency value for the filter calculation (Red circle). In the example, the frequency value entered is "0.86 Hz.".

b) Filter Order: Integer value. This field should be set to a valid filter order for the calculation. Typically, this is order 2 or 4 (by default, it is set to 4 for all analyses) (Yellow circle).

c) STA (Short-Time-Average): Floating-point value. Represents the short-term dynamic time window average, measuring the "instantaneous" amplitude of a seismic signal to detect potential earthquakes. In the example, the value is "0.5.".

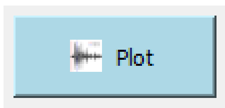
d) LTA (Long-Time-Average): Floating-point value. Represents the long-term dynamic time window average, measuring the "average amplitude of current seismic noise." In the example, it is "4.".

e) ThrOn: Floating-point value. Indicates the cutoff ratio to activate the STA/LTA algorithm. It determines the "maximum parameter for peaks and the arrival of the P-wave." Its value generally matches or is in the same range as the "LTA" value. In the example, the value is "4.".

f) ThrOff: Floating-point value. Indicates the cutoff ratio to end the STA/LTA algorithm. It determines the "minimum parameter for peaks and the arrival of the S-wave." Its value generally matches or is in the same range as the "STA" value. In the example, the value is "0.7.".

g) **Trace number st (0):** Integer value. Determines the trace number to be analyzed. By default, and in formats like "SAC," this value is set to "0." For formats containing multiple traces, such as "MSEED" or "SEISAN," the user selects the trace number to analyze.

The "Frec. Min (Hz)/Minimum Frequency" and "Frec. Max (Hz)/Maximum Frequency" checkboxes remain inactive.



Once the actions are performed and the desired triggering method (Recursive - Classic or typical) is selected, click the “Plot” button. If there is an issue with the input (such as invalid entry, nonexistent record, or out-of-range format) and you attempt to plot by clicking the “Plot” button, a validation dialog box will appear. This dialog box will inform you of the necessary action to correct the issue.

These dialog boxes ensure that the program can continue running without severe system interruptions due to data issues. The validations for incorrect entries are as follows:

[3] The **Highpass Filter** blocks low-frequency signals and allows high-frequency signals (*frequencies higher than the cutoff frequency*) to pass through.

Fig. 17 Validation of entries for the *Highpass* filter.

Once the inputs are corrected by entering a valid value or one within the permitted range, the program can continue and display the corresponding graph.

3.2.7.- *Bandpass*[4] *Filter Type.*

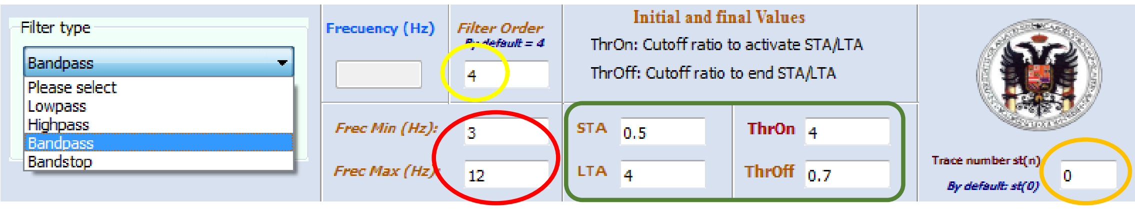


Fig. 18 **Bandpass Filter Selection Example.** In the red circle, the fields "Minimum Frequency (Hz) and Maximum Frequency (Hz)" are highlighted; yellow circle -> "Filter Order"; orange circle -> "Trace No."; green rounded rectangle -> (STA, LTA, ThrOn, ThrOff).

[4] The Bandpass filter allows spectral content to pass only within a range around the central frequency. This range is defined by a minimum frequency and a maximum frequency. It removes noise associated with both low and high frequencies that are outside this specified range.

Unlike the previous two filters, the following checkboxes are activated:

a) **Minimum Filter Frequency (Hz)**: Floating-point value. Enter a valid minimum frequency to define the start of the central window (Red circle). In the example, the value is “3 Hz.”.

b) **Maximum Filter Frequency (Hz)**: Floating-point value. Enter a valid maximum frequency to define the end of the central window (Red circle). In the example, the value is “12 Hz.”.

c) **STA (Short-Time-Average)**: Floating-point value. Represents the short-term dynamic time window average, measuring the "instantaneous" amplitude of a seismic signal to detect potential earthquakes. In the example, the value is “0.5”.

d) **LTA (Long-Time-Average)**: Floating-point value. Represents the long-term dynamic time window average, measuring the "average amplitude of current seismic noise." In the example, the value is “4”.

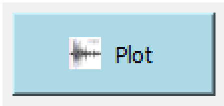
e) **ThrOn**: Floating-point value. Indicates the cutoff ratio to activate the STA/LTA algorithm, determining the "maximum parameter for peaks and the arrival of the P-wave." Its value generally matches or is in the same range as the LTA value. In the example, the value is “4”.

f) **ThrOff**: Floating-point value. Indicates the cutoff ratio to end the STA/LTA algorithm, determining the "minimum parameter for peaks and the arrival of the S-wave." Its value generally matches or is in the same range as the STA value. In the example, the value is “0.7”.

g) **Filter Order**: Integer value. Specify a valid filter order for the calculation. Typically, this is order 2 or 4 (default is set to 4 for all analyses) (Yellow circle).

h) **Trace number st (0):** Integer value. Determines the trace number to be analyzed. By default, and in formats like "SAC," this value is set to "0." For formats containing multiple traces, such as "MSEED" or "SEISAN," the user selects the trace number to analyze.

The "Frequency (Hz)/Filter Frequency" checkbox remains inactive.



Once the actions are performed and the desired triggering method (*Recursive - Classic or typical*) is selected, click the “Plot” button. If there is an issue with the input (*such as invalid entry, nonexistent record, or out-of-range format*) and you attempt to plot by clicking the “Plot” button, a validation dialog box will appear.

This dialog box will prompt you to correct the issue. These dialog boxes ensure that the program can continue running without severe system interruptions due to data issues. The validations for incorrect entries are described on the following page.

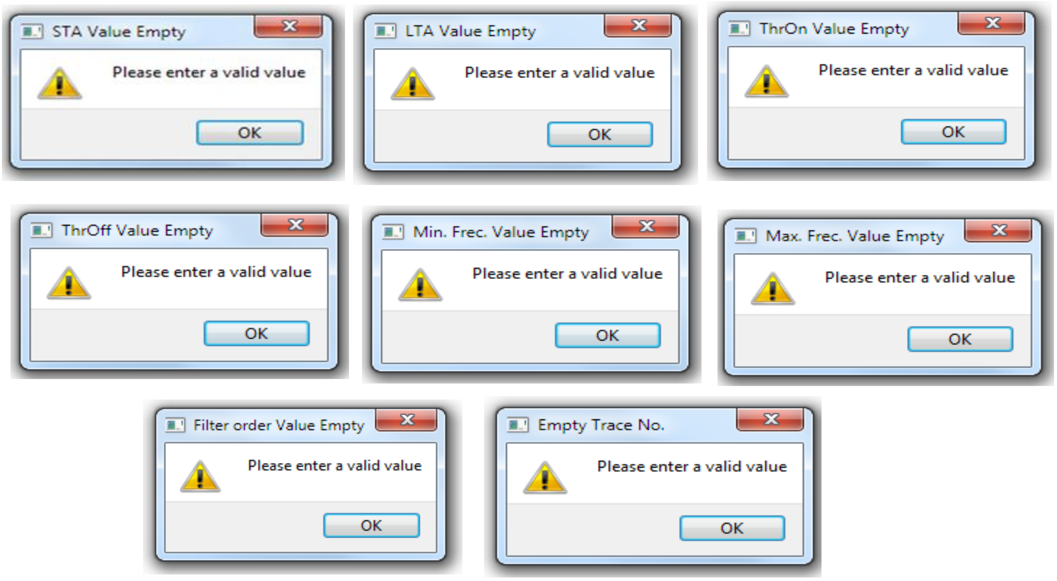


Fig. 19 Validation of entries for the *Bandpass* filter.

3.2.8.- *Bandstop*[5] *Filter Type.*

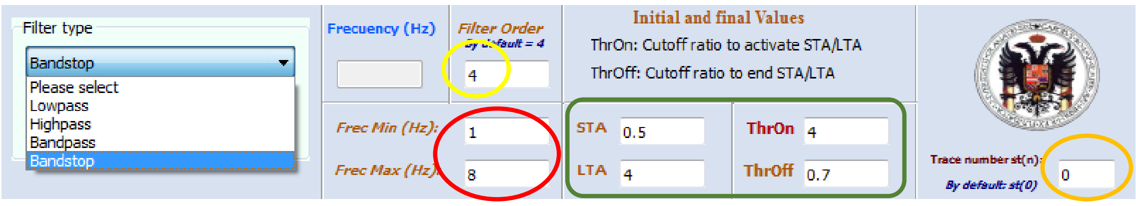


Fig. 20 Example of Bandstop Filter Selection Type. In the red circle are the inputs "**Minimum Frequency** (Hz) and **Maximum Frequency** (Hz)," yellow circle -> "**Filter Order**" orange circle -> **Trace Number**, green rounded rectangle -> (**STA, LTA, ThrOn, ThrOff**)

Similar to the Bandpass filter, the following checkboxes are activated for the Bandstop filter:

a) **Minimum Filter Frequency (Hz)**: Floating-point value. Enter a valid minimum frequency to define the start of the central window (Red circle). In the example, the value is "3 Hz.".

b) **Maximum Filter Frequency (Hz)**: Floating-point value. Enter a valid maximum frequency to define the end of the central window (Red circle). In the example, the value is "12 Hz.".

c) **STA (Short-Time-Average)**: Floating-point value. Represents the short-term dynamic time window average, measuring the "instantaneous" amplitude of a seismic signal to detect potential earthquakes. In the example, the value is "0.5".

d) **LTA (Long-Time-Average)**: Floating-point value. Represents the long-term dynamic time window average, measuring the "average amplitude of current seismic noise." In the example, the value is "4".

[5] The Bandstop filter blocks signals with frequencies between the upper and lower cutoff frequencies. In other words, it removes or attenuates a specific band of frequencies.

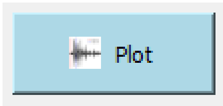
e) **ThrOn**: Floating-point value. Indicates the cutoff ratio to activate the STA/LTA algorithm, determining the "maximum parameter for peaks and the arrival of the P-wave." Its value generally matches or is in the same range as the LTA value. In the example, the value is “4”.

f) **ThrOff**: Floating-point value. Indicates the cutoff ratio to end the STA/LTA algorithm, determining the "minimum parameter for peaks and the arrival of the S-wave." Its value generally matches or is in the same range as the STA value. In the example, the value is “0.7”.

g) **Filter Order**: Integer value. Specify a valid filter order for the calculation. Typically, this is order 2 or 4 (default is set to 4 for all analyses) (Yellow circle).

h) **Trace number st (0):** Integer value. Determines the trace number to be analyzed. By default, and in formats like "SAC," this value is set to "0." For formats containing multiple traces, such as "MSEED" or "SEISAN," the user selects the trace number to analyze.

The "Frequency (Hz)/Filter Frequency" checkbox remains inactive.



Once the actions are performed and the desired triggering method (Recursive - Classic or typical) is selected, click the “Plot” button. If there is an issue with the input (such as invalid entry, nonexistent record, or out-of-range format) and you attempt to plot by clicking the “Plot” button, a validation dialog box will appear.

This dialog box will prompt you to correct the issue. These dialog boxes ensure that the program continues running without severe interruptions due to data issues. The validations for incorrect entries are as follows:

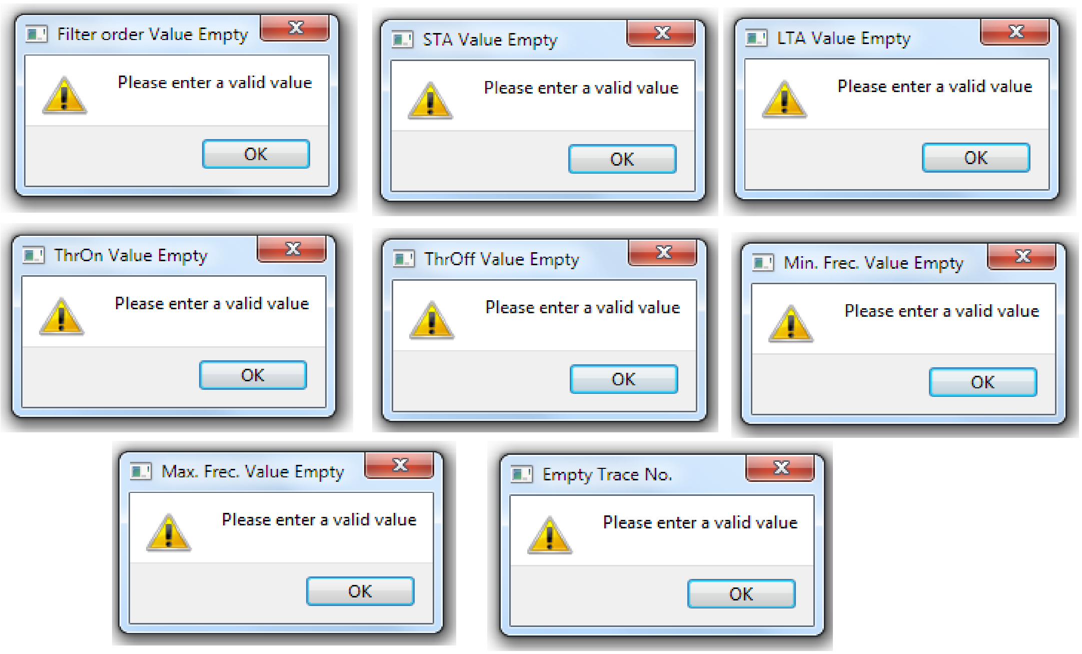


Fig. 21 Validation of entries for the *Bandstop* filter.

Back to Index

4.- STA/LTA Triggering Methods Block (Recursive and Classic or Typical). 3

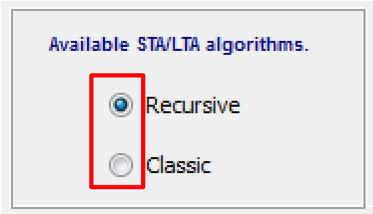


Fig. 22 Selection of Triggering Method Type in the STA/LTA Algorithm (Recursive and Classic or Typical)

This section presents the two available methods for the STA/LTA algorithm: the "Recursive" method and the "Classic" method.

To select a method, click the buttons indicated for each method (red box in the figure). Clicking (activating) one of them will deactivate the other, making method selection straightforward.

By default, the "Recursive" method is selected at the start. The observer will determine which method to select. The choice of method is accompanied by the selection of the filter type for the signal. Clicking the "Clean" button will revert to the default initial value, which is the "Recursive" method

5.- Path and Commands Block. 4

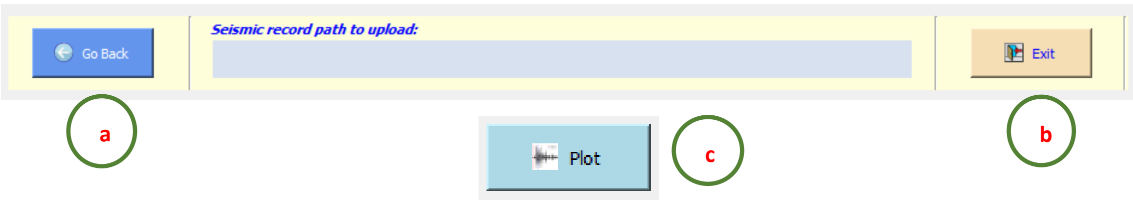
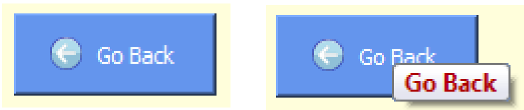


Fig. 23 Physical File Path Block and Command Buttons { (a) Go Back, (b) Exit, and (c) Plot }

This last block consists of the following elements:

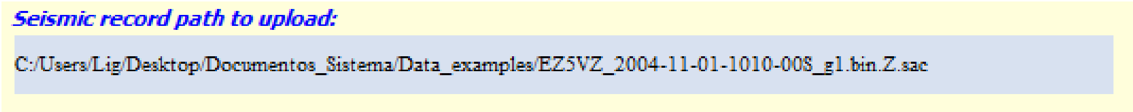
a) **"Go Back" Button**: Allows returning to the initial system presentation screen. Hovering the mouse pointer over it displays a message indicating its function.



b) **File Path Display Area (Seismic record path to upload)**: This area shows the path (Drive/folder/file) of the physical file (where it is stored on the computer) to be accessed by the system for the necessary calculations.

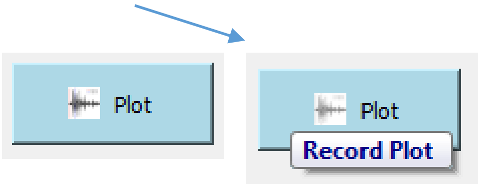
Back to Index

An example of the output when performing the calculations can be seen in the following image.



The "**Path**" (on the computer) of the registry location is shown.

c) "**Plot** button": Once the filter types and STA/LTA triggering method are selected, calculations are performed and the corresponding graphs are presented. Hovering the mouse pointer over the button displays a message indicating its function.



d) "**Exit button**": Allows for a complete exit from the system (*after displaying a prompt asking if you want to leave the system*). Hovering the mouse pointer over the button displays a message indicating its function.

In the same way as on the home screen, if the "**Exit button**" is clicked, a window will appear asking the user if they are sure they want to leave the system.

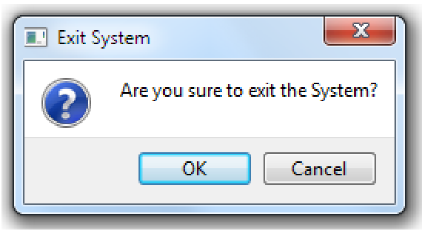


Fig. 24 Text box asking if you want to exit the system.

Clicking "**OK**" closes the screen and completes the system exit. "Cancel" returns to the analysis screen.

5.1.- Error validation in records or entries.

Clicking the "**Plot button**" will trigger an error validation if an invalid entry, non-existent record, or out-of-range format occurs. A dialog box will appear with an alert message indicating the issue. This allows the user to modify the entries or select a valid record without causing the system to crash or halt. The screen displayed is as follows:

Back to Index

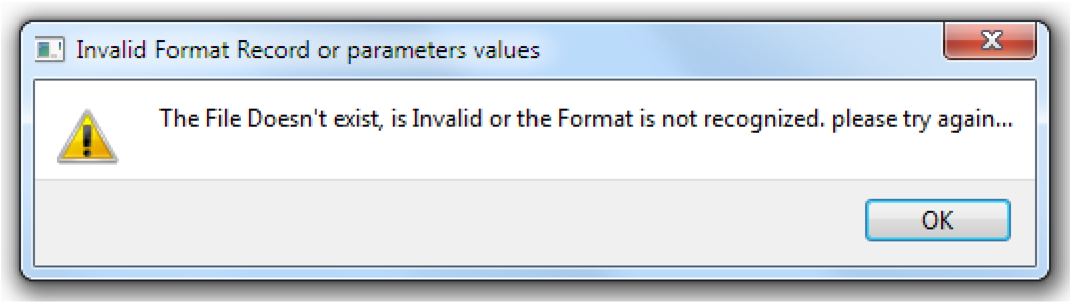


Fig. 25 Validation of invalid entries, formats, or incorrect records.

According to the message in the previous figure, an error has occurred because the format is not recognized, or the record does not exist. Additionally, the parameters or entries might be out of the allowed range according to the signal being analyzed. By pressing the "**OK button**", you return to the system to select a valid file or correct the erroneous entries. This way, the program continues to run without issues.

6.- Results of Filter and Analysis Sections.

Next, examples of the final results from the filtering and analysis process using the STA/LTA method will be presented.

6.1.- Example of results for filter selection and Recursive method.

According to the entire process described earlier, the procedure for filtering records is very simple and consists of the following steps:

a) Open or select a specific record (the file path of the record is displayed as "Seismic record path to upload." By default, the initial path is in the root directory "C" of the PC, whether on Windows or Linux).

b) Select the filter.

c) Enter the filter parameters.

d) Select the triggering method of the STA/LTA algorithm (Recursive).

e) Click the "Plot" button to graph the record.

All of this will produce the output of the analysis, which will consist of graphs of the original signal, the filtered signal, and their respective thresholds. You can zoom in on the graphs and save them in several formats.

The interface with the output elements is shown on the following page.

Back to Index

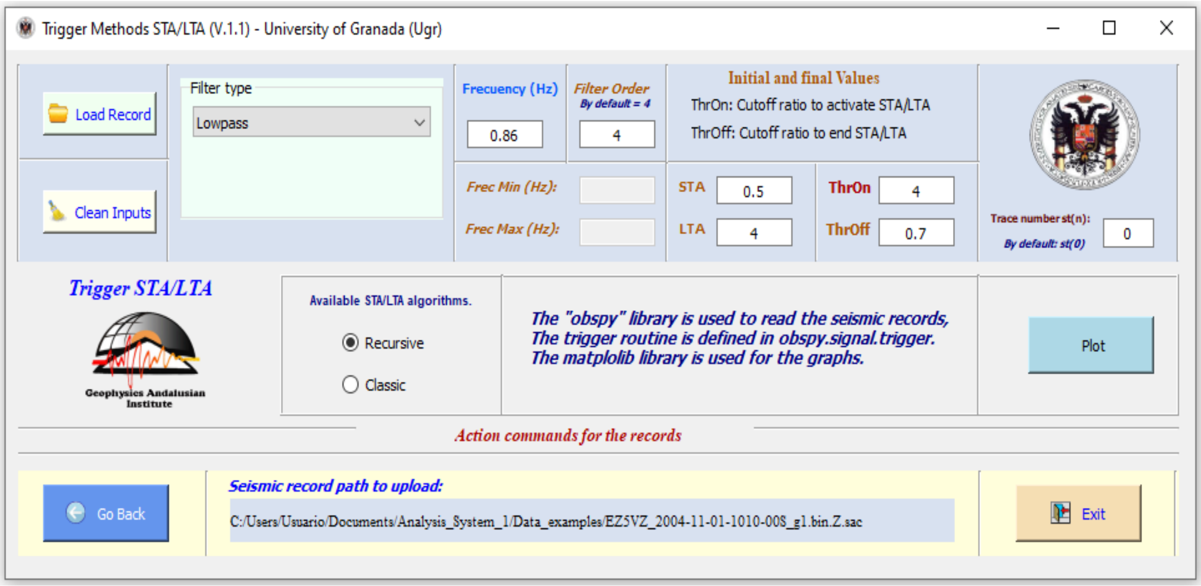


Fig. 26 Selection of parameters for calculating the STA/LTA algorithm of a SAC record, with a *Lowpass* filter and "**Recursive**" method.

In the figure, under "File Path/Ruta Archivo", the complete path where the record to be analyzed is stored is shown. In the "Filter Type/Tipo de Filtro" list, the *Lowpass* filter has been selected. In the input fields, a frequency of 0.86 Hz and a filter order of 4 have been set. The values are STA = 0.5, LTA = 4, ThrOn = 4, and ThrOff = 0.7. The record contains only one trace, the default trace "0".

6.2.- Example of results for filter selection and Classic or Typical method.

According to the entire process described earlier, performing an analysis with the Classic method for the records is very simple and consists of the following steps:

a) Open or select a specific record (the file path of the record is displayed as "Seismic record path to upload." By default, the initial path is in the root directory "C" of the PC, whether on Windows or Linux).
b) Select the filter.
c) Enter the filter parameters.
d) Select the triggering method of the STA/LTA algorithm (Classic).
e) Click the "Plot" button to graph the record.

All of this will produce the output of the analysis, which will consist of graphs of the original signal, the filtered signal, and their respective thresholds. You can zoom in on the graphs and save them in several formats.

The interface with the output elements is shown on the following page.

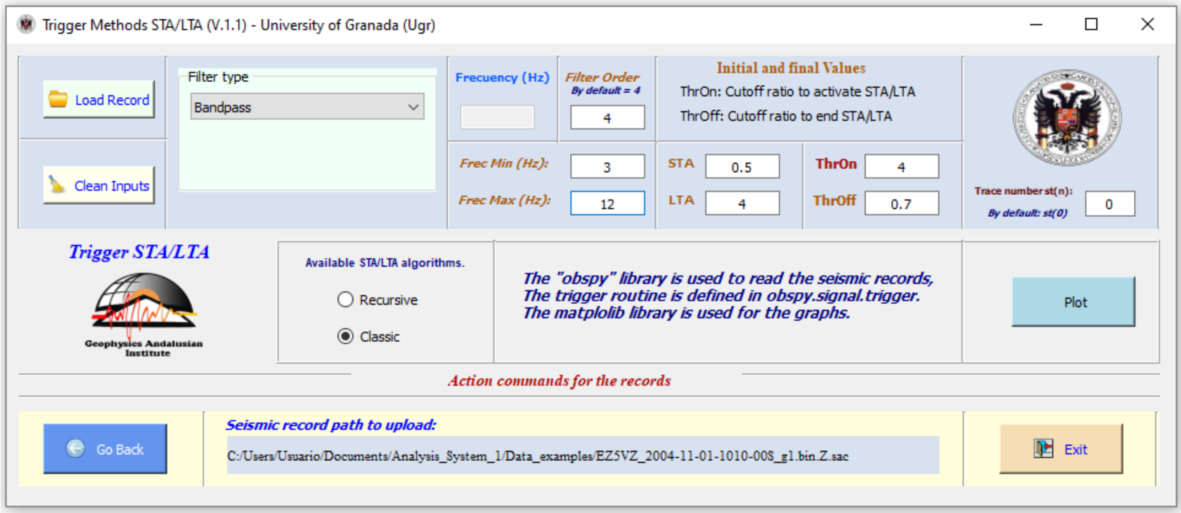


Fig. 27 Selection of parameters for calculating the STA/LTA algorithm of a SAC record, with a *Bandpass* filter and "**Classic**" method.

In the figure, under "File Path/Ruta Archivo", the complete path where the record to be analyzed is stored is shown. In the "Filter Type/Tipo de Filtro" list, the **Bandpass** filter has been selected. In the input fields, a minimum frequency of 3 Hz and a maximum frequency of 12 Hz have been set, with the default filter order left at 4. The values are STA = 0.5, LTA = 4, ThrOn = 4, and ThrOff = 0.7. The record contains only one trace, the default trace "0".

The graphical results can be seen in the following sections.

7.- Resulting graphs.

The execution of the program presents two resulting graphs: one of the original signal and one of the signal processed using the selected filter. The time unit values indicate that every 1000 points correspond to 20 seconds of record duration (*time intervals should be adjusted for each record*). Examples of graphical results will be presented next, showing the filtering process and the selection of the STA/LTA algorithm triggering method (*recursive and classic or typical*).

7.1.- Filter Graphs and Zoom of Records.

As examples, the graphical result of the original record and the filtered record in SAC format is presented for both methods (*Recursive - Classic or Typical*). Similarly, a zoom (*using the Zoom tool [Magnifying Glass]*) on the resulting graphs is shown (*cf. Matplotlib Tools, pp. 41-46*).

Note that when zooming in on one of the two sections of the graph (*original and filtered*), the other updates automatically. That is, zooming in on just one section adjusts the time vector in both graphs. This allows for a better analysis of P and S wave arrivals in the signals. On the other hand, it is important to reiterate that the parameters used here in the examples are arbitrary, so it is up to the operator to adjust the values and thresholds to better define the arrival of P and S waves for each event.

Back to Index

7.1.1.- Graphs of Filters and Zoom of the records by the “Recursive” method.

a) SAC record graph with a “*Lowpass*” filter and “*recursive*” method. Original signal.

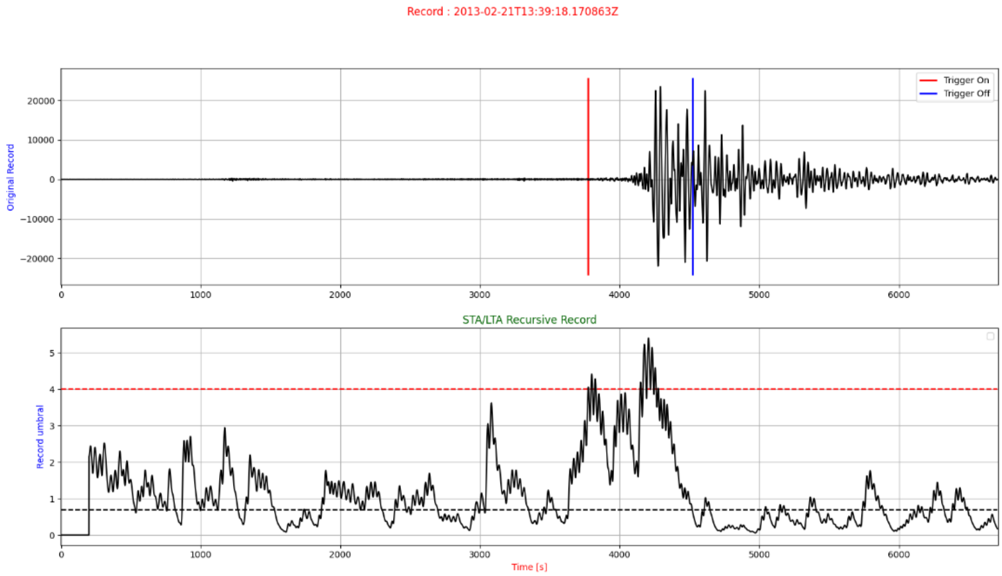


Fig. 28 Example of SAC record graph, with a Lowpass filter at 0.86 Hz and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

b) Zoom of the SAC record graph with a “*Lowpass*” filter and “*recursive*” method. Original signal.

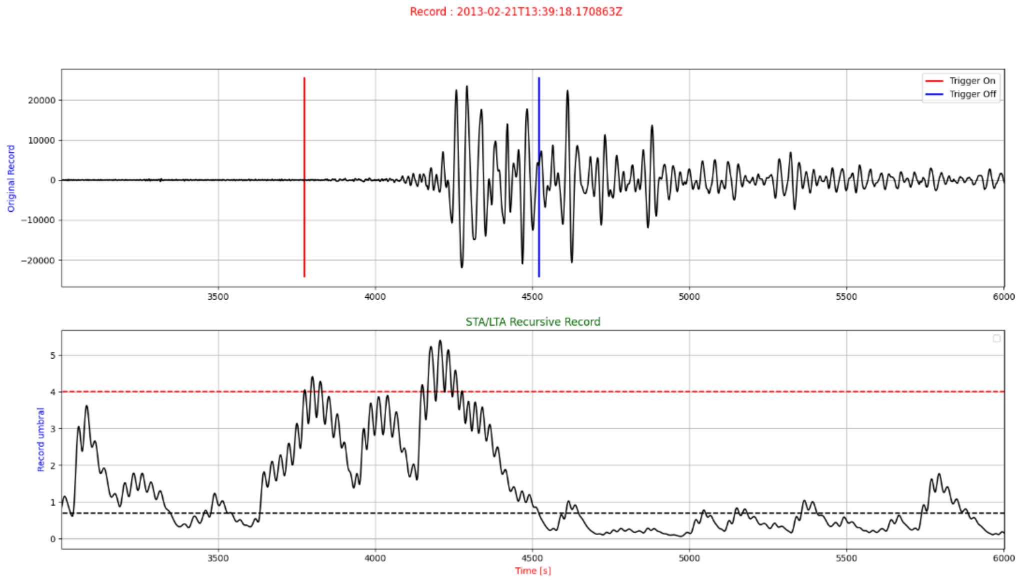


Fig. 29 Example of Zoom of the SAC record graph with a Lowpass filter at 0.86 Hz and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

c) Graph of SAC record with a "*Lowpass*" filter and "*recursive*" method. Filtered signal.

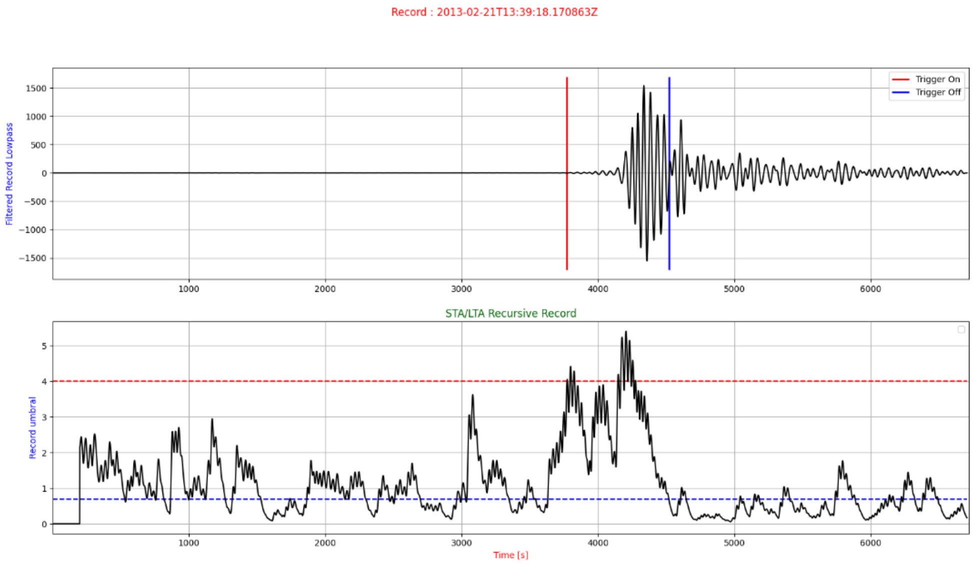


Fig. 30 Example of SAC record graph, with a Lowpass filter at 0.86 Hz and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

d) Zoom of the SAC record graph with a "*Lowpass*" filter and "*recursive*" method. Filtered signal.

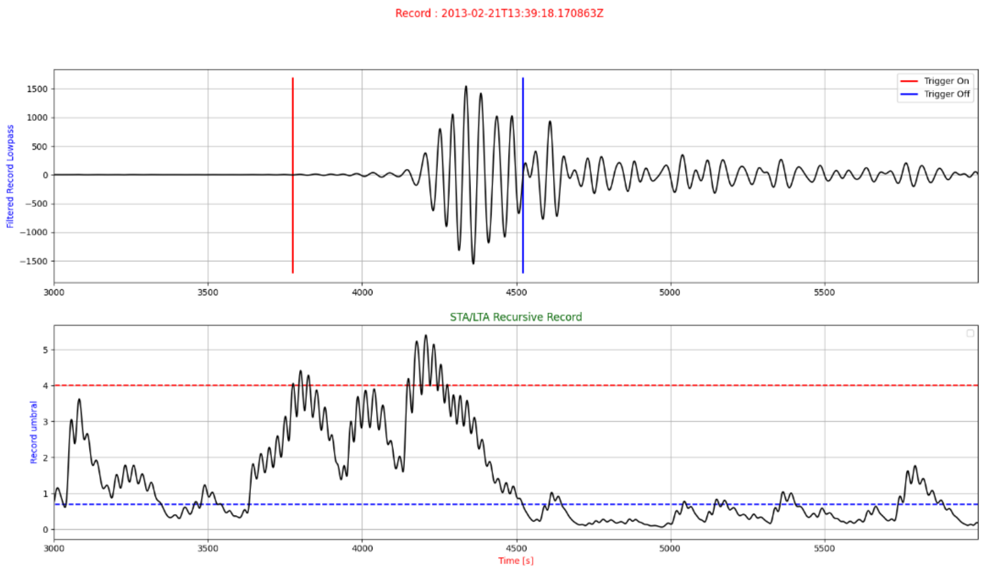


Fig. 31 Example of Zoom of the SAC record graph with a Lowpass filter at 0.86 Hz and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first..

e) SAC record graph with a "*Highpass*" filter and "*recursive*" method. Original signal.

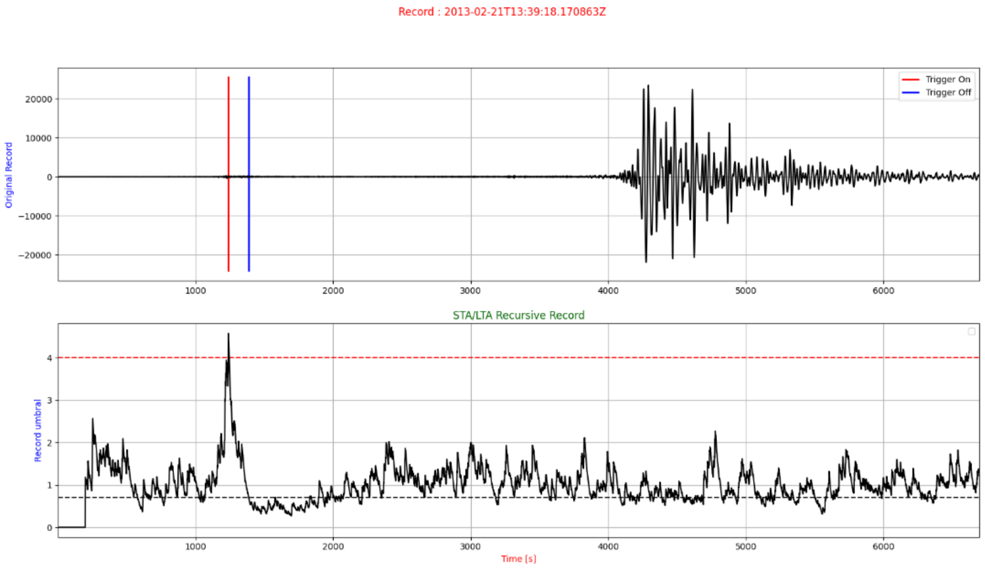


Fig. 32 Example of SAC record graph with a Highpass filter at 15 Hz and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

f) Zoom of the SAC record graph with a "*Highpass*" filter and "*recursive*" method. Original signal. 

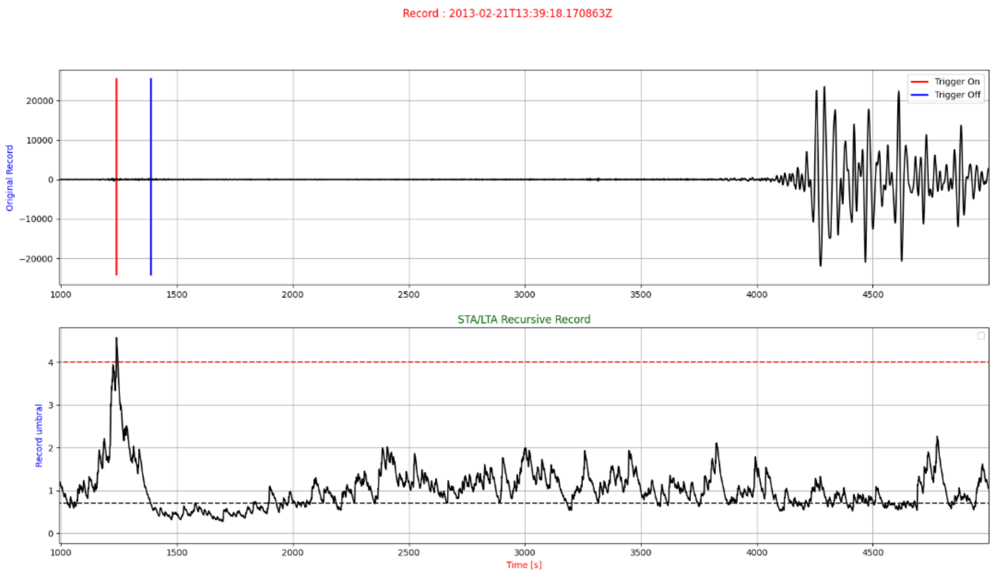


Fig. 33 Example of Zoom of the SAC record graph with a Highpass filter at 15 Hz frequency and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

g) Graph of SAC record with a "*Highpass*" filter and "*recursive*" method. Filtered signal.

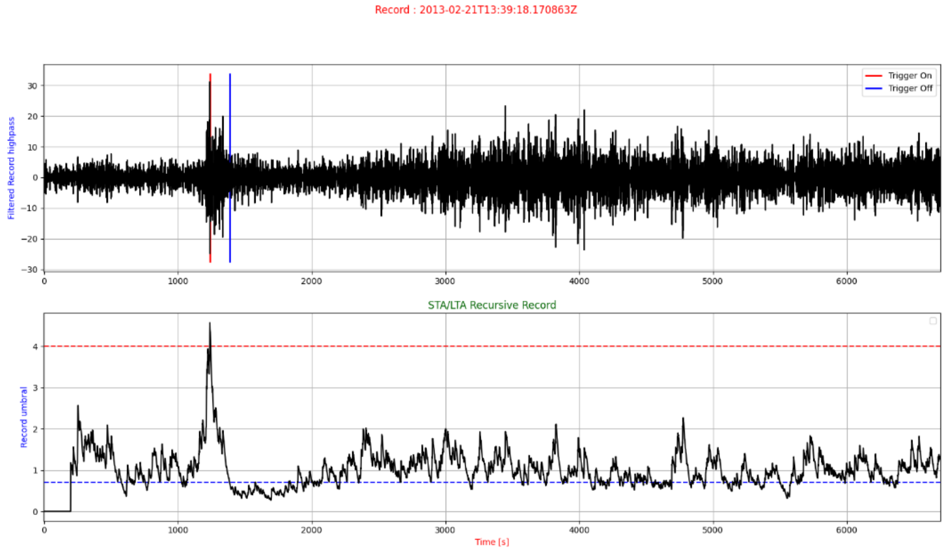


Fig. 34 Example of SAC record graph with a Highpass filter at 15 Hz and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

h) Zoom of the SAC record graph with a "*Highpass*" filter and "*recursive*" method. Filtered signal.

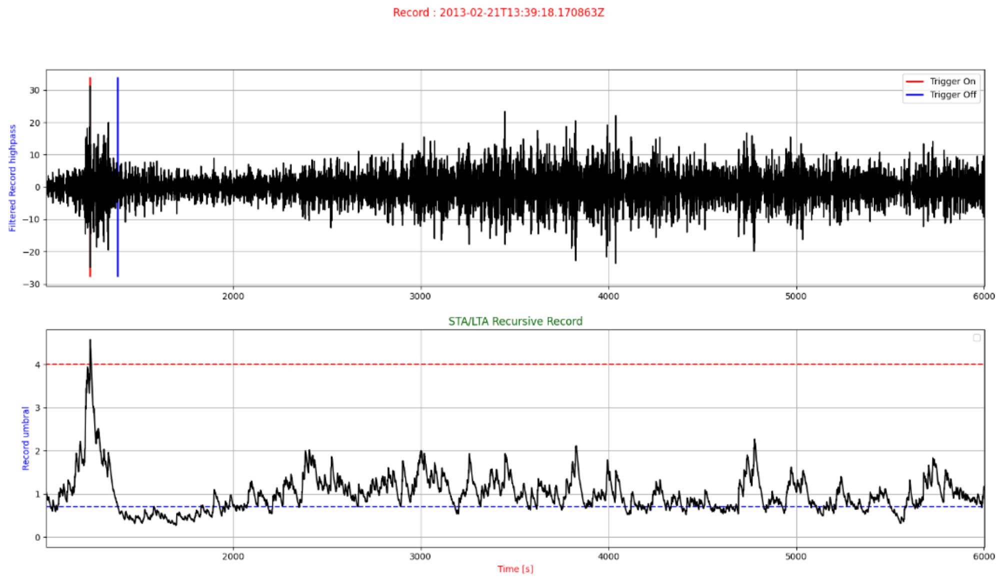


Fig. 35 Example of Zoom of the SAC record graph with a Highpass filter at 15 Hz frequency and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

i) SAC record graph with a "*Bandpass*" filter and "*recursive*" method. Original signal.

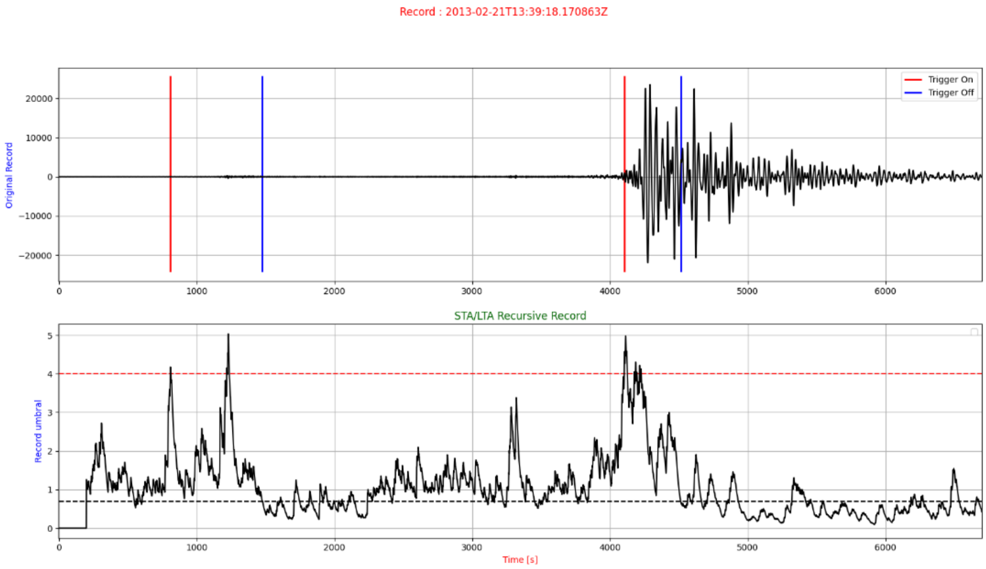


Fig. 36 Example of SAC record graph with a Bandpass filter: minimum frequency at 3 Hz, maximum frequency at 12 Hz, and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

j) Zoom of the SAC record graph with a "*Bandpass*" filter and "*recursive*" method. Original signal.

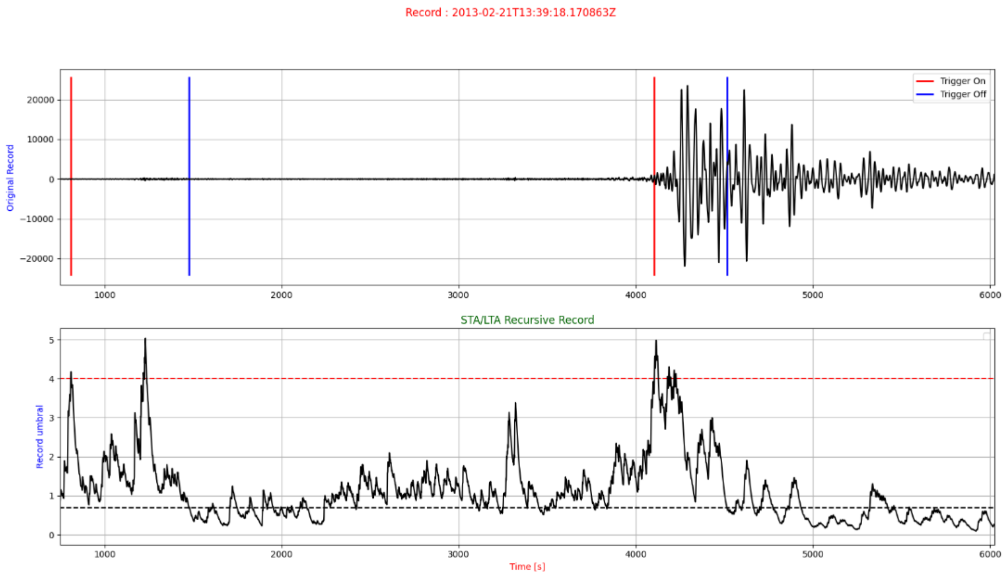


Fig. 37 Example of Zoom of the SAC record graph with a bandpass filter: minimum frequency at 3 Hz, maximum frequency at 12 Hz, and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

k) Graph of SAC record with a "*Bandpass*" filter and "*recursive*" method. Filtered signal.

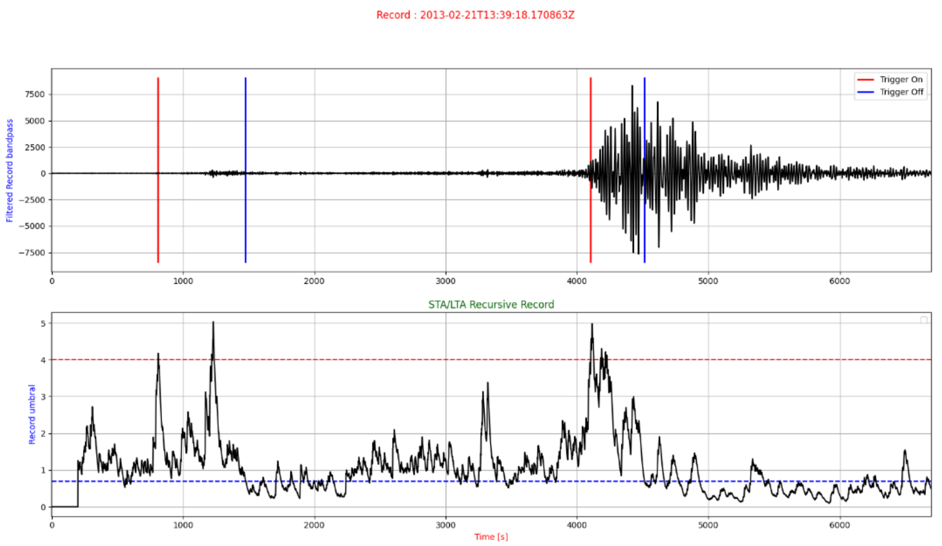


Fig. 38 Example of SAC record graph with a bandpass filter: minimum frequency at 3 Hz, maximum frequency at 12 Hz, and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

l) Zoom of the SAC record graph with a "*Bandpass*" filter and "*recursive*" method. Filtered signal. 

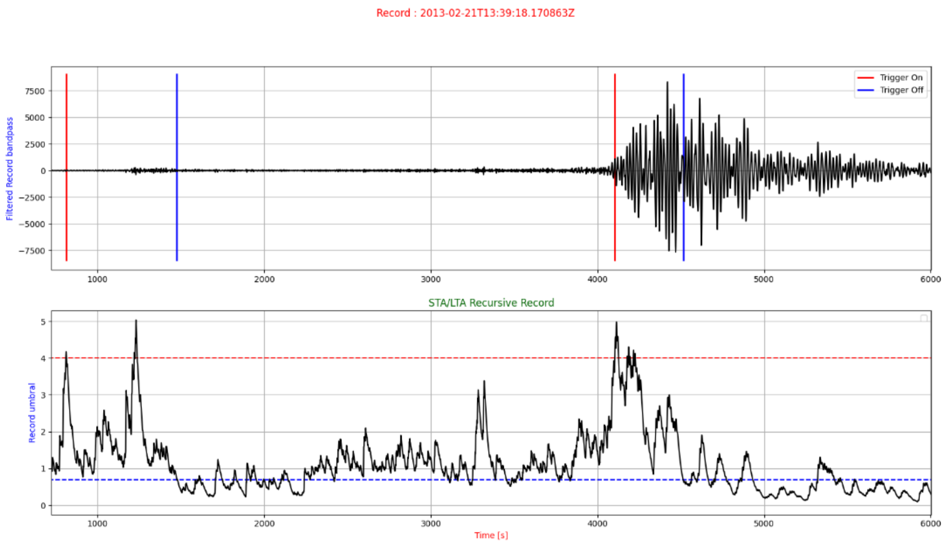


Fig. 39 Example of Zoom of the SAC record graph with a bandpass filter: minimum frequency at 3 Hz, maximum frequency at 12 Hz, and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

m) SAC record graph with a "*Bandstop*" filter and "*recursive*" method. Original signal.

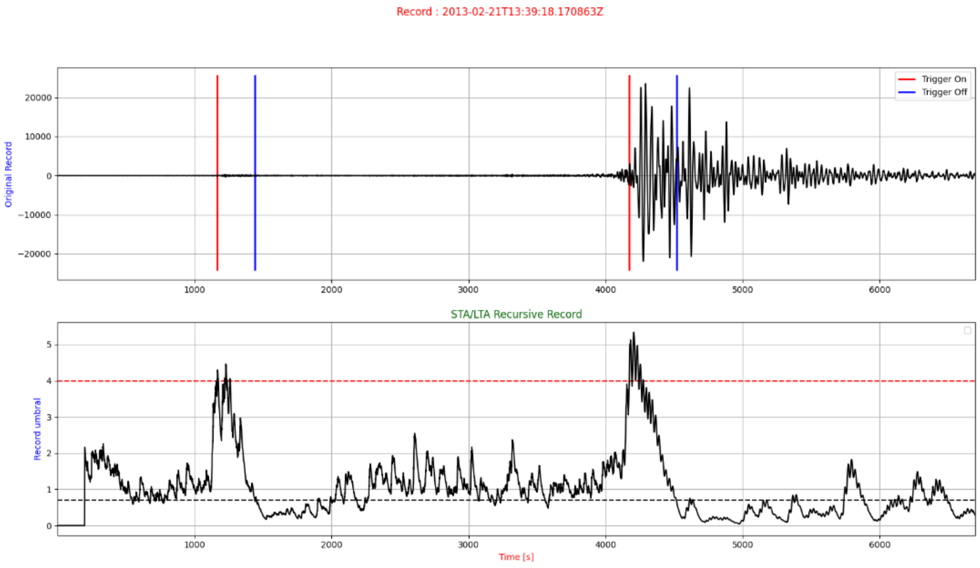


Fig. 40 Example of SAC record graph with a Bandstop filter: minimum frequency at 1 Hz, maximum frequency at 8 Hz, and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

n) Zoom of the SAC record graph with a "*Bandstop*" filter and "*recursive*" method. Original signal.

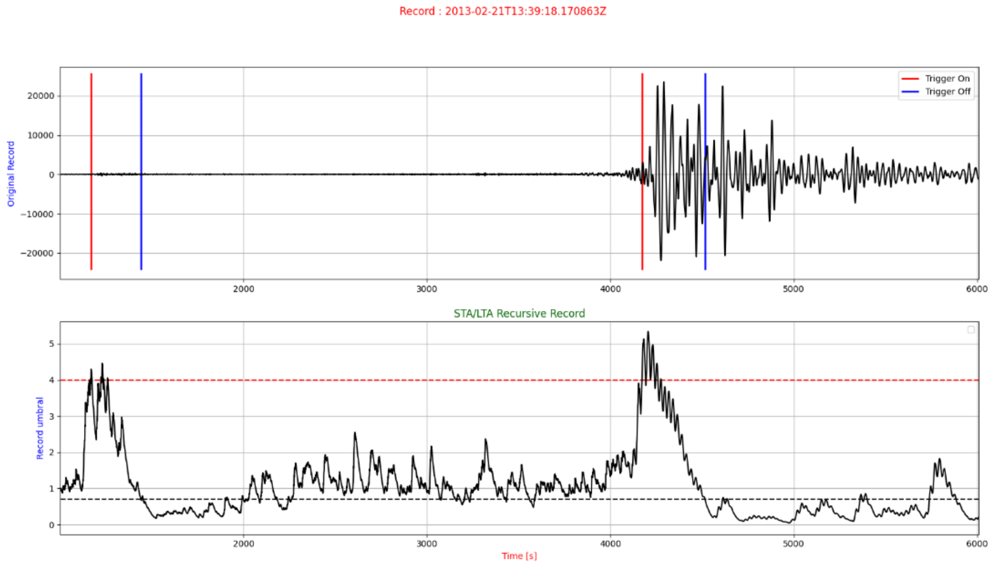


Fig. 41 Example of Zoom of the SAC record graph with a Bandstop filter: minimum frequency at 1 Hz, maximum frequency at 8 Hz, and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

o) Graph of SAC record with a "*Bandstop*" filter and "*recursive*" method. Filtered signal.

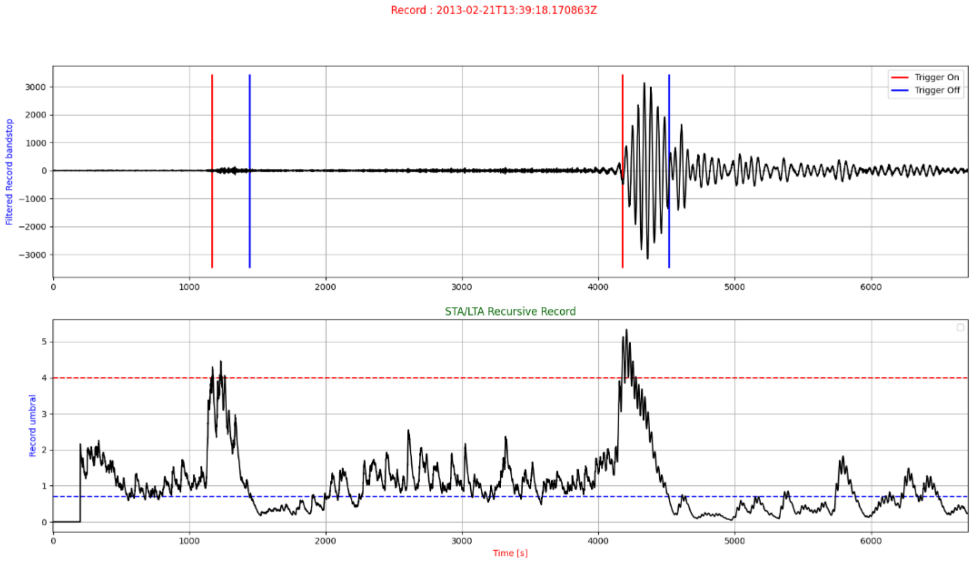


Fig. 42 Example of SAC record graph with a Bandstop filter: minimum frequency at 1 Hz, maximum frequency at 8 Hz, and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

p) Zoom of the SAC record graph with a "*Bandstop*" filter and "*recursive*" method. Filtered signal.

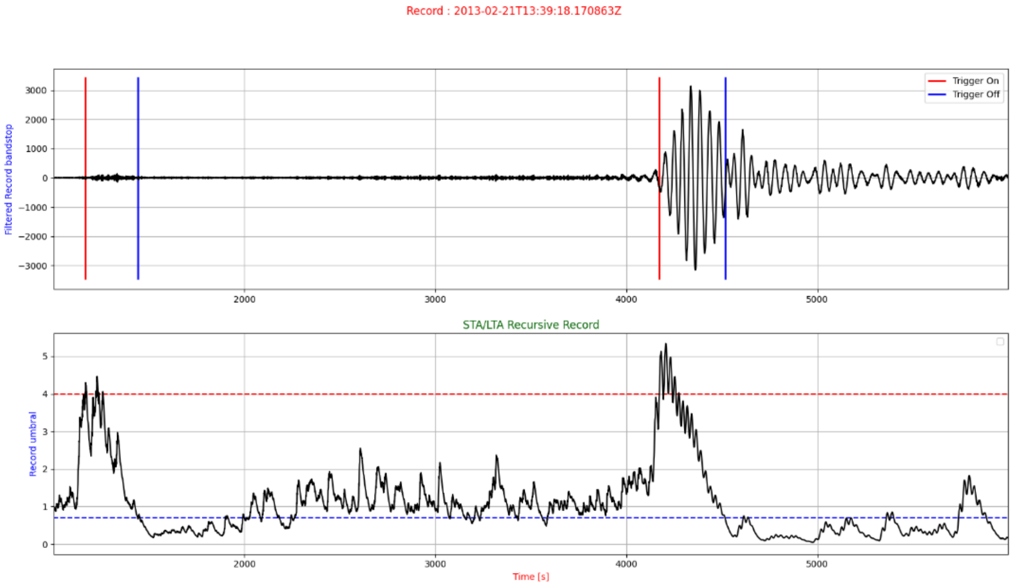


Fig. 43 Example of Zoom of the SAC record graph with a Bandstop filter: minimum frequency at 1 Hz, maximum frequency at 8 Hz, and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

7.1.2.- **Graphs of Filters and Zoom of the records by the “Classic” method.**

a) SAC record graph with a “*Lowpass*” filter and “*Classic*” method. Original signal

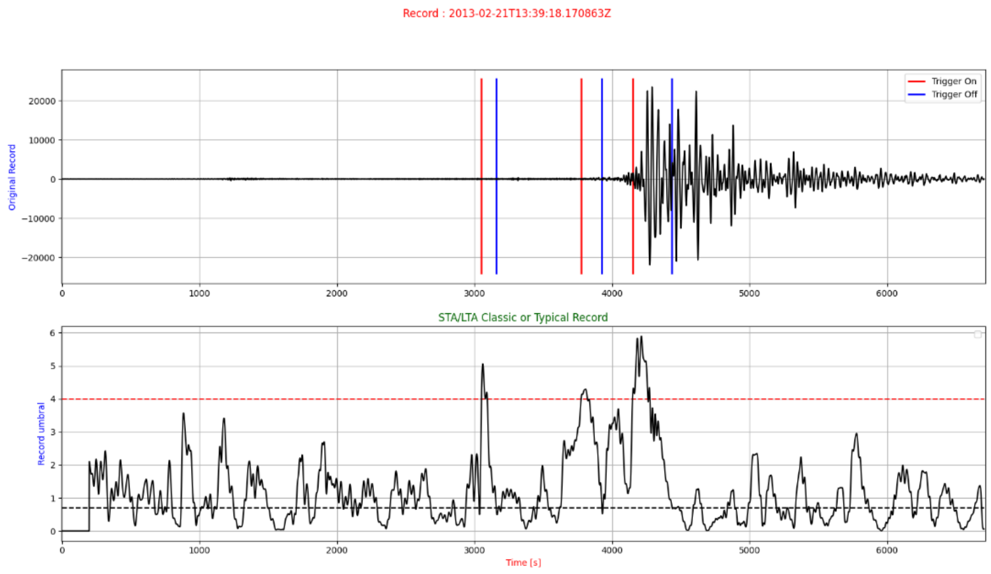


Fig. 44 Example of SAC record graph with a Lowpass filter at 0.86 Hz and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

b) Zoom of the SAC record graph with a “*Lowpass*” filter and “*Classic*” method. Original signal.

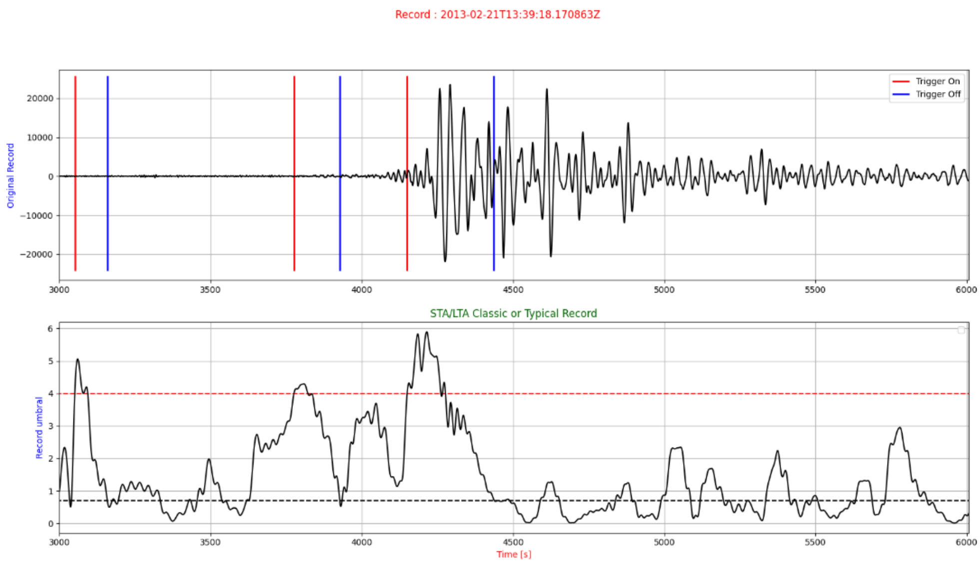


Fig. 45 Example of Zoom of the SAC record graph with a Lowpass filter at 0.86 Hz and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

c) SAC record graph with a "*Lowpass*" filter and "*Classic*" method. Filtered signal.

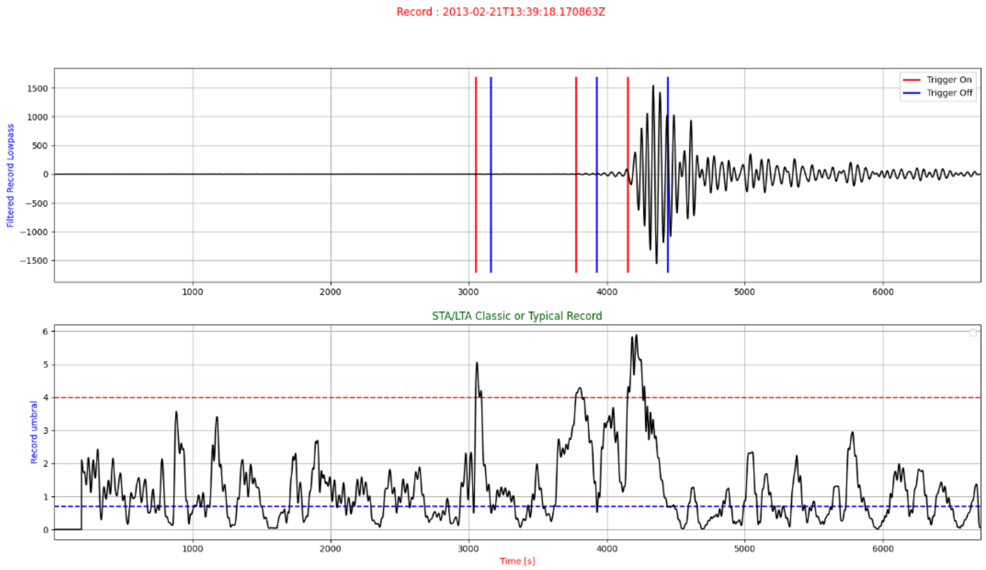


Fig. 46 Example of SAC record graph with a Lowpass filter at 0.86 Hz and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

d) Zoom of the SAC record graph with a "*Lowpass*" filter and "*Classic*" method. Filtered signal. 

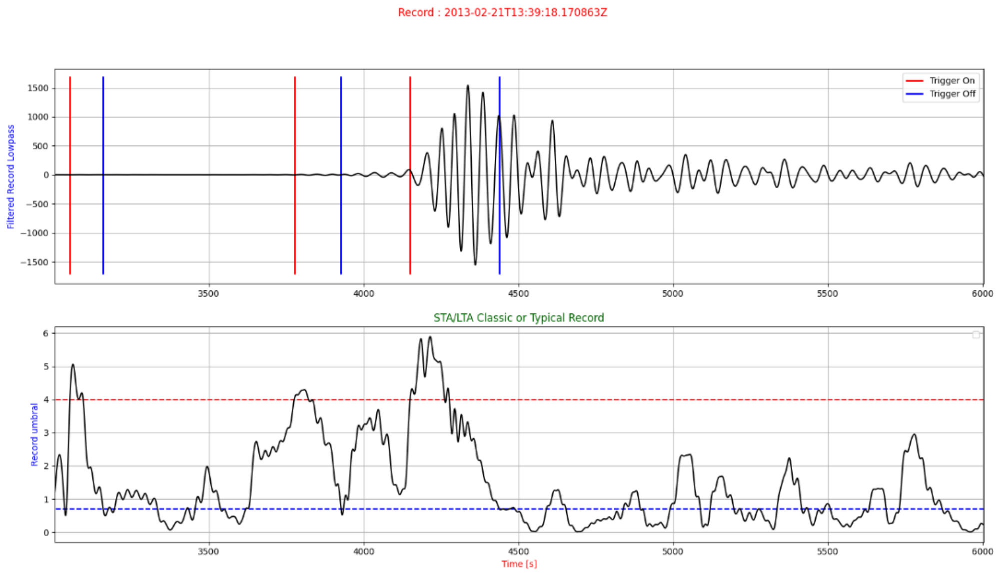


Fig. 47 Example of Zoom of the SAC record graph with a Lowpass filter at 0.86 Hz and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

e) SAC record graph with a "*Highpass*" filter and "*Classic*" method. Original signal.

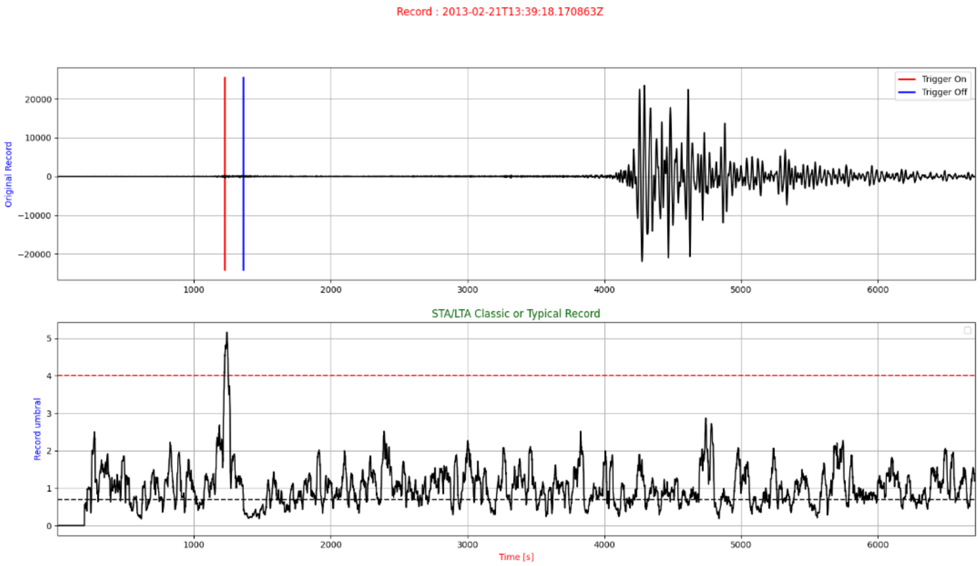


Fig. 48 Example of SAC record graph with a Highpass filter at 15 Hz and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

f) Zoom of the SAC record graph with a "*Highpass*" filter and "*Classic*" method. Original signal.

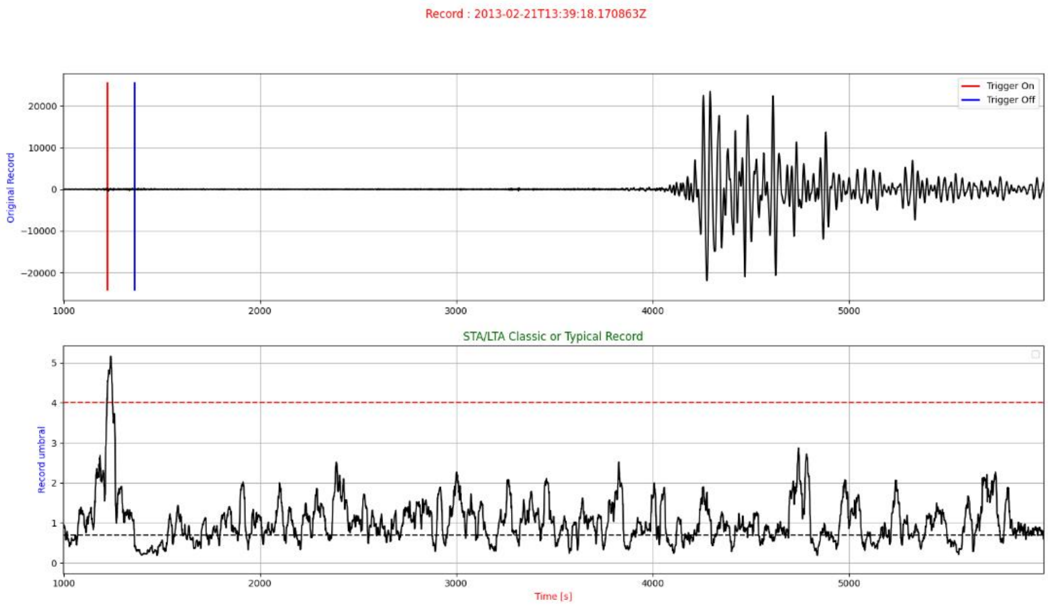


Fig. 49 Example of Zoom of the SAC record graph with a high-pass filter at 15 Hz frequency and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

g) SAC record graph with a "*Highpass*" filter and "*Classic*" method. Filtered signal.

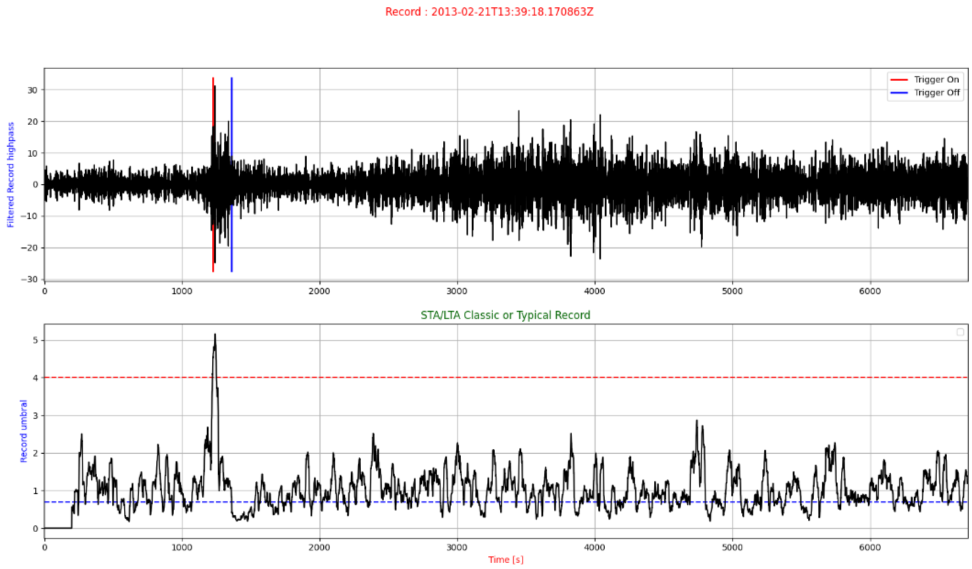


Fig. 50 Example of SAC record graph with a Highpass filter at 15 Hz and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

h) Zoom of the SAC record graph with a "*Highpass*" filter and "*Classic*" method. Filtered signal.

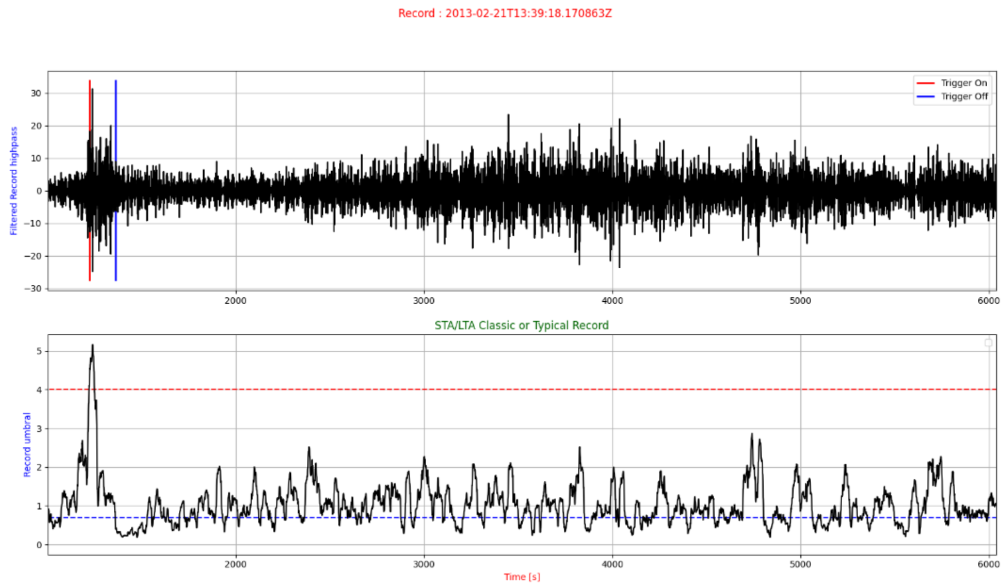


Fig. 51 Example of Zoom of the SAC record graph with a high-pass filter at 15 Hz frequency and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

i) SAC record graph with a "*Bandpass*" filter and "*Classic*" method. Original signal.

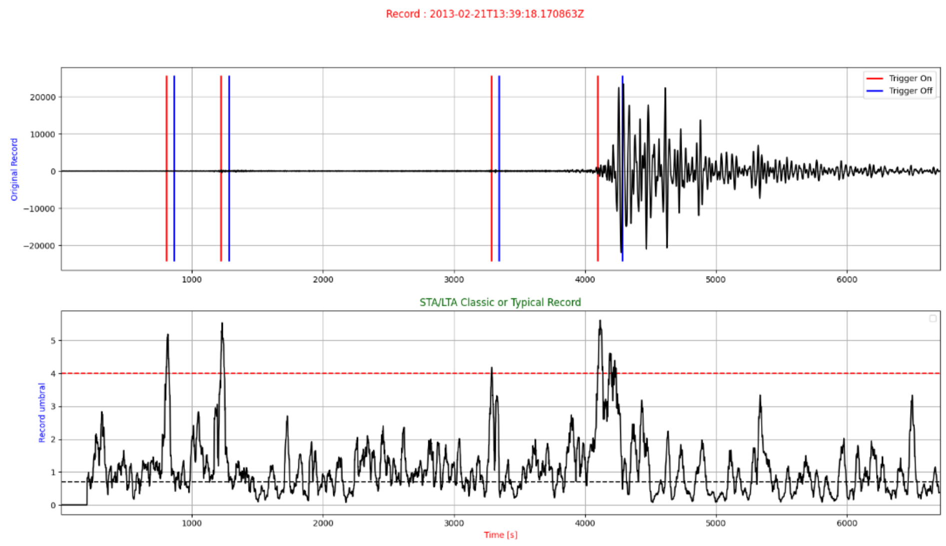


Fig. 52 Ejemplo de Gráfica del registro SAC, con filtro *paso-banda* (*Bandpass*): Frecuencia mínima a 3, frecuencia máxima a 12 y orden de filtro 4. Valores de los parámetros para el algoritmo STA/LTA son: STA=0.5, LTA=4, ThrOn = 4, ThrOff = 0.7, Número de traza = 0.

j) Zoom of the SAC record graph with a "*Bandpass*" filter and "*Classic*" method. Original signal.

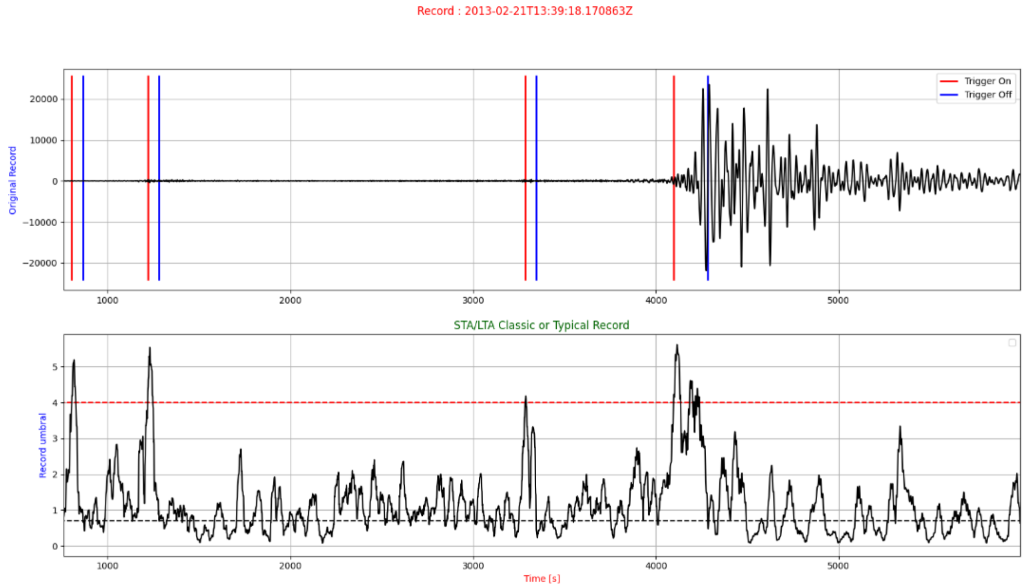


Fig. 53 Ejemplo de Zoom de la Gráfica del registro SAC, con filtro *paso-banda* (*Bandpass*): Frecuencia mínima a 3, frecuencia máxima a 12 y orden de filtro 4. Se selecciona la herramienta Zoom de Matplotlib, trazando un cuadro en cualquiera de los dos segmentos de la gráfica, el segundo con el vector de tiempo se adapta al zoom del primero.

Back to Index

k) SAC record graph with a “*Bandpass*” filter and “*Classic*” method. Filtered signal.

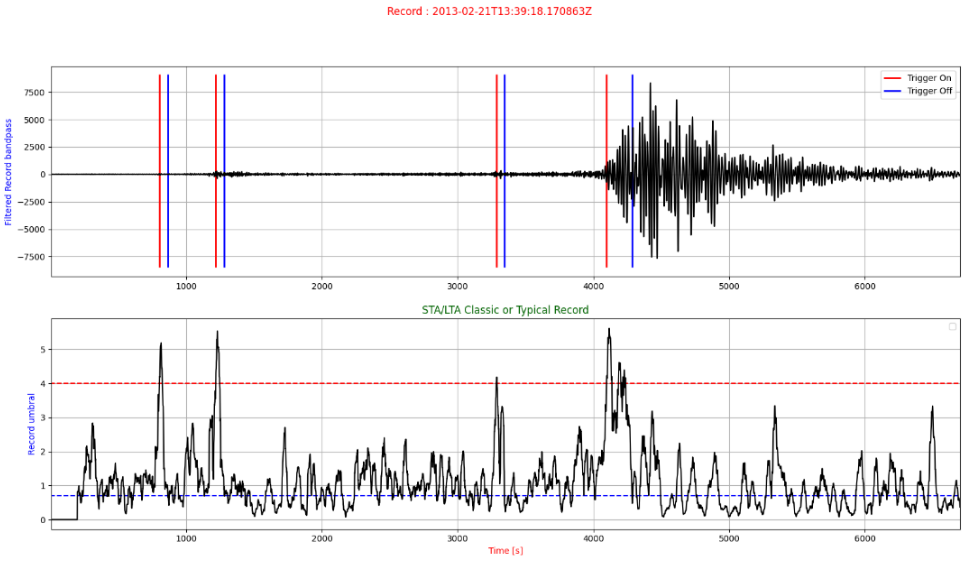


Fig. 54 Example of SAC record graph with a Bandpass filter: minimum frequency at 3 Hz, maximum frequency at 12 Hz, and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

l) Zoom of the SAC record graph with a “*Bandpass*” filter and “*Classic*” method. Filtered signal.

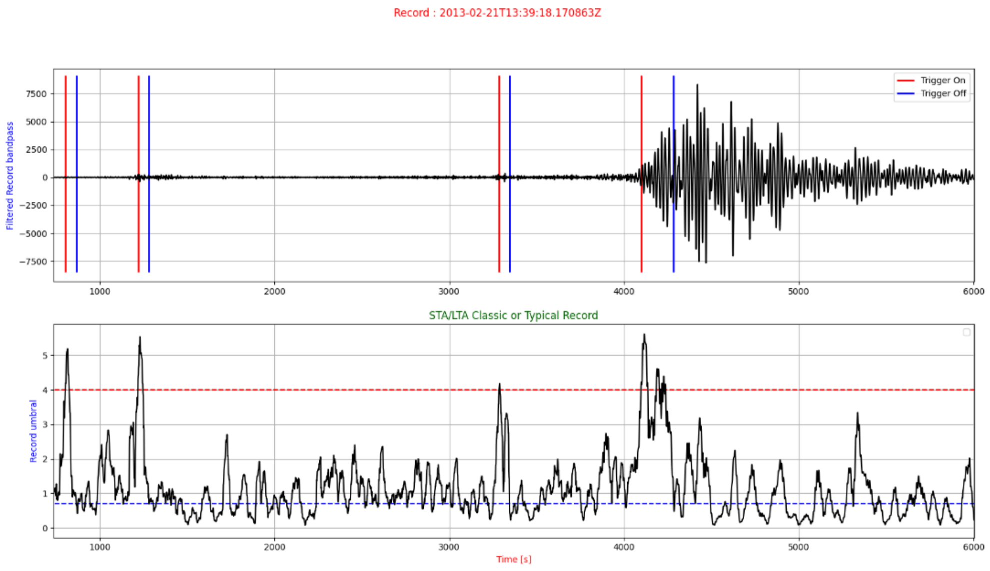


Fig. 55 Example of Zoom of the SAC record graph with a Bandpass filter: minimum frequency at 3 Hz, maximum frequency at 12 Hz, and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

m) SAC record graph with a "*Bandstop*" filter and "*Classic*" method. Original signal.

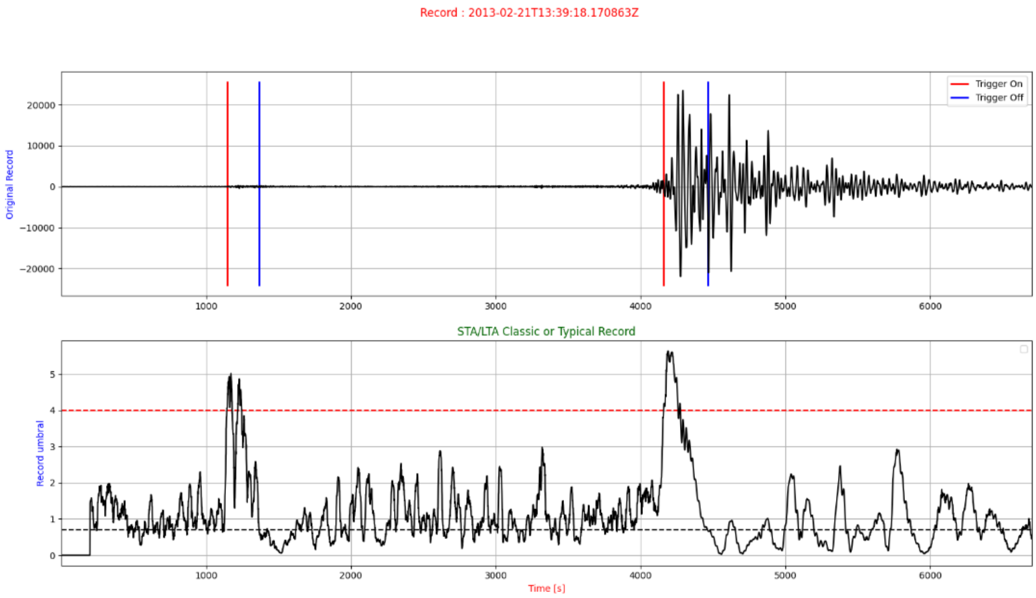


Fig. 56 Example of SAC record graph with a Bandstop filter: minimum frequency at 1 Hz, maximum frequency at 8 Hz, and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

n) Zoom of the SAC record graph with a "*Bandstop*" filter and "*Classic*" method. Original signal.

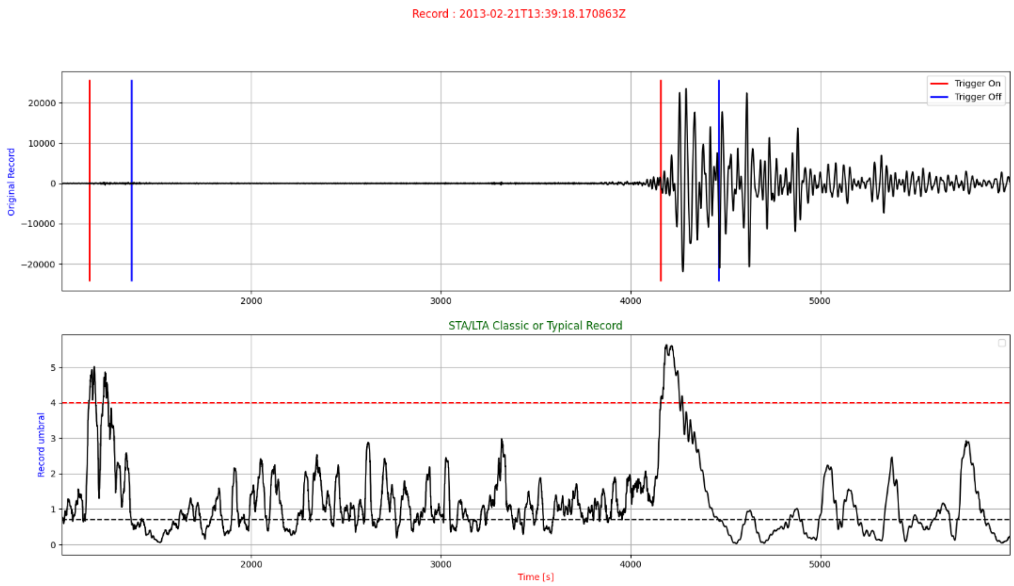


Fig. Example of Zoom of the SAC record graph with a Bandstop filter: minimum frequency at 1 Hz, maximum frequency at 8 Hz, and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

Back to Index

o) SAC record graph with a "*Bandstop*" filter and "*Classic*" method. Filtered signal.

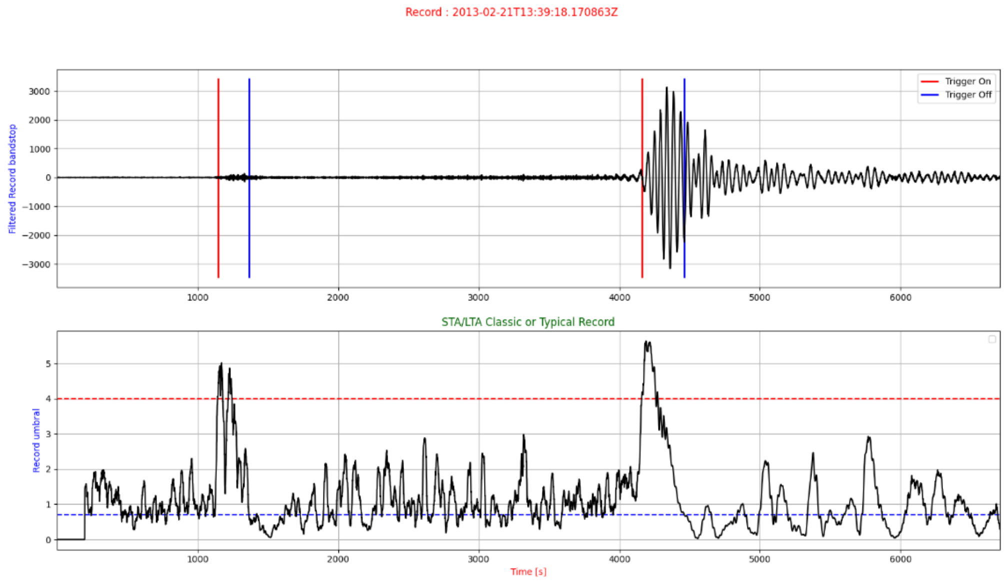


Fig. 58 Example of SAC record graph with a Bandstop filter: minimum frequency at 1 Hz, maximum frequency at 8 Hz, and filter order 4. Parameter values for the STA/LTA algorithm are: STA=0.5, LTA=4, ThrOn=4, ThrOff=0.7, Trace number=0.

p) Zoom of the SAC record graph with a "*Bandstop*" filter and "*Classic*" method. Filtered signal.

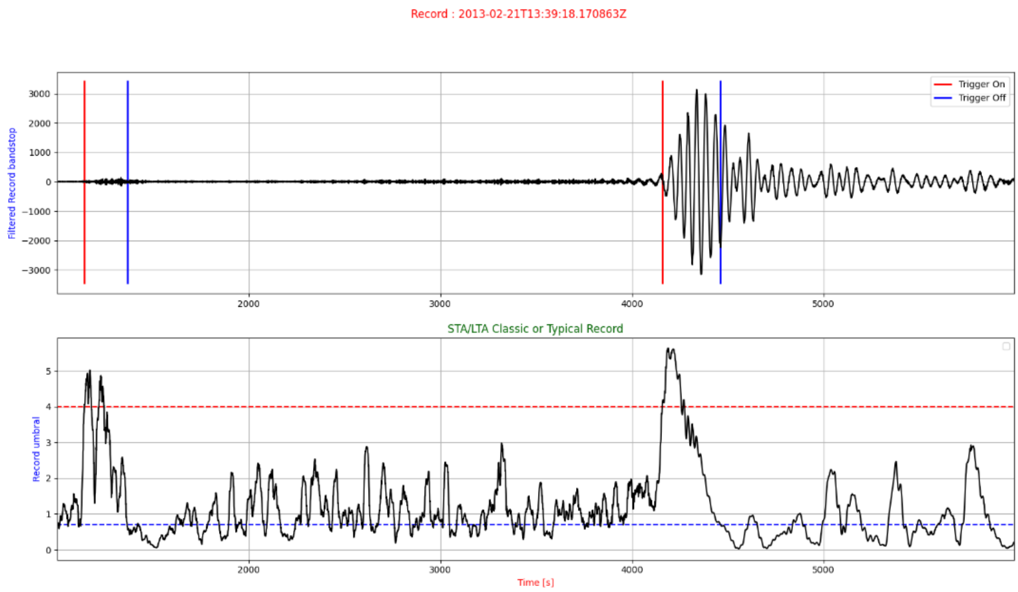


Fig. 59 Example of Zoom of the SAC record graph with a Bandstop filter: minimum frequency at 1 Hz, maximum frequency at 8 Hz, and filter order 4. The Zoom tool from Matplotlib is selected, drawing a box in either of the two segments of the graph, with the second segment's time vector adapting to the zoom of the first.

8.- Creation of text files (.txt) with final STA and LTA values.

For this new version (1.1) of the software, an additional utility has been added compared to the previous version. The update includes the creation of two text files (.txt) that store:

a) **The results of the STA arrival time values in the seismic record.**
b) **The results of the LTA arrival time values in the seismic record.**

At the end of each file, the count of events detected by the algorithm is included.

For each type of filter, these files are created. For example, for the "*Lowpass*" filter, the following files will be created:

a) Choosing the *Recursive method*

Sta_Results_Lowpass_File_Recursive.txt
Lta_Results_Lowpass_File_Recursive.txt

b) Selecting the *Classic or Typical method*.

Sta_Results_Lowpass_File_Classic.txt
Lta_Results_Lowpass_File_Classic.txt

Similarly, text files will be created for the other filter types: *Highpass*, *Bandpass*, and *Bandstop*, according to the selected method.

The creation of these files and the arrival time values can be important when calculating the differences in arrival times between different phases. This needs to be refined when setting the threshold parameters in the interface (STA, LTA, Trigger On, Trigger Off).

For example, a record in which a selection has been made using the Classic method, with a *Lowpass* filter (similar to that in Figure 44 but with some parameter modifications), results in two graphs: the first showing the filtered record and the second showing the original record. They are shown in the following images.

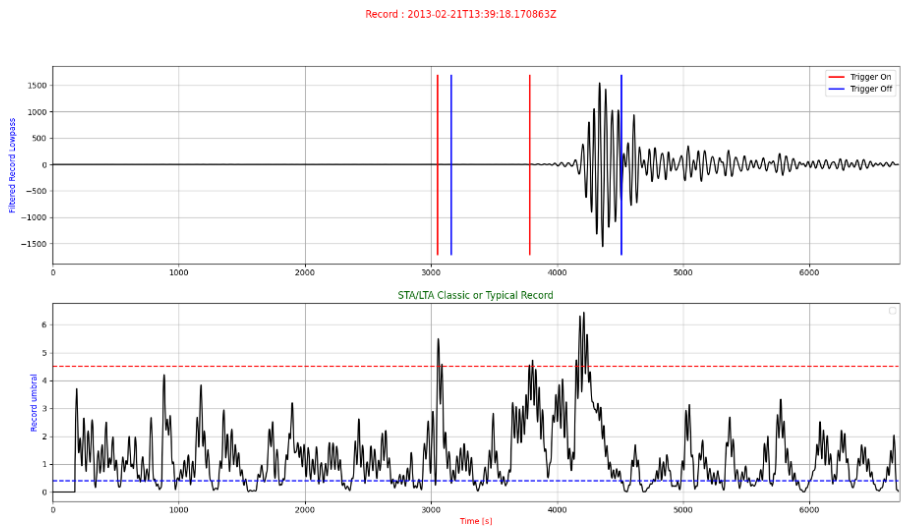


Fig. 60 Example of SAC Record Plot, with Lowpass filter at 0.86 Hz and filter order 4. Parameters for the STA/LTA algorithm are: STA=0.4, LTA=3.55, ThrOn = 4.5, ThrOff = 0.4, Trace number = 0.

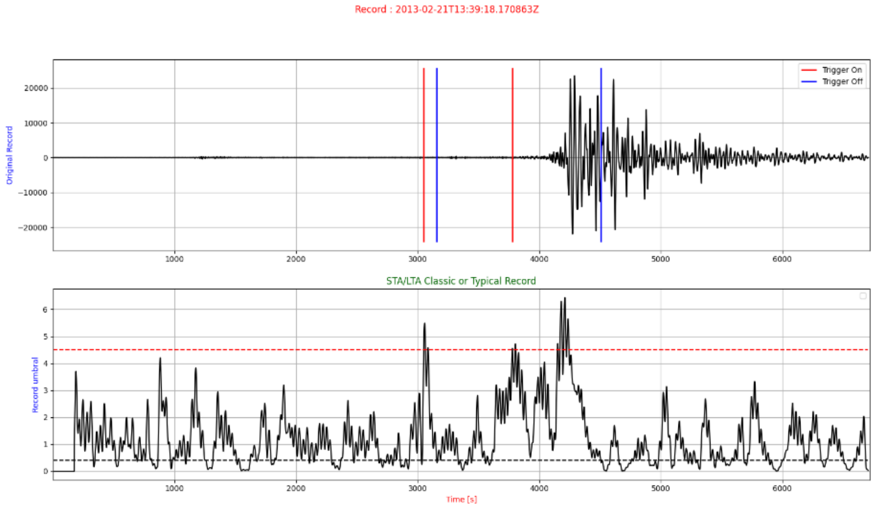


Fig. 61 Example of Original Plot (without filter) of the SAC record, with filter order 4. Parameters for the STA/LTA algorithm are: STA=0.4, LTA=3.55, ThrOn = 4.5, ThrOff = 0.4, Trace number = 0

The text files generated, accompanying the plots, are as follows:

a) Sta_Results_Lowpass_File_Classic.txt

EQU STA Time Values
[3051 3779]

EQU number:
2

b) Lta_Results_Lowpass_File_Classic.txt

EQU LTA Time Values
[3157 4507]

EQU number:
2

It indicates that 2 events have been detected, with their respective STA and LTA arrivals presented in time values.

In the previous record (*Fig. 60*), 1000 units correspond to 20 seconds of time. Therefore, in STA, the first mark of 3051 corresponds to 61.02 seconds. For LTA, the mark is at 3157, which corresponds to 63.14 seconds. This shows that the time interval between STA and LTA for the first detected event is 2.12 seconds.

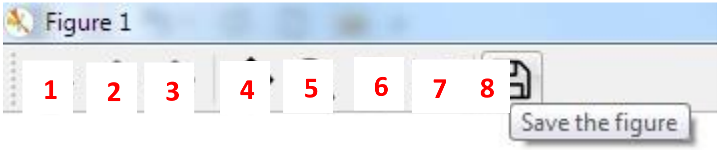
For the second event (*Fig.61*), the STA indicates 3779, which corresponds to 75.58 seconds, with an LTA of 4507 indicating 90.14 seconds. From this, it is deduced that the difference between STA and LTA is 14.58 seconds.

This information could be useful both for counting detected events at a certain threshold and for calculating the time differences in the triggering algorithm between STA and LTA arrivals.

Back to Index

9.- Toolbar of Graphs (Matplotlib Library).

In the construction of graphs, the *Matplotlib* library's graph screen has a set of very useful tools that allow you to visualize, edit, and save graphs in various formats. At the top of the Matplotlib graph screen that appears when a graph is created, there is a toolbar similar to the following:



From left to right, the icons representing the actions to be performed are:

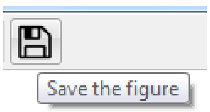
1. Reset original view: Restores all graphs to the initial state.
2. Back to previous view: Preview of the selected graph.
3. Forward to next view: Forward view of the next image.
4. Left button pans, Right button zooms, x/y fixes axis, CTRL fixes aspect. Moves the graph and axes left or right, zooms in or out, fixes the x/y axis, and fixes the aspect with CTRL.
5. Zoom to rectangle: Through a rectangle, zooms in on the selected graph.
6. Configure subplots: Configuration of subplots (Borders and spacings).
7. Edit axis, curve and image parameters: Editing the parameters of the graph. Select the axes or graph and edit elements such as title, coordinates (X, Y), and curve parameters (lines, markers) in styles, colors, and size.
8. Save the figure: Saves the graph in several formats.

This document does not delve into each of them; it only highlights the use of those that are generally more commonly used, such as (1, 2, 5, 7, and 8).

In the above graphs, the use of the zoom tool (5) has been observed. Tools 2 and 3 allow zooming in or out individually for each graph. Option 1 allows restoring all elements or subplots of the graph to their initial values (*each individual graph or part of the window*). As for option 8, it allows saving the graph in various formats. The rest are straightforward, and it is up to the user to explore each of them. Now, the processes for "***editing***" and "***saving*** or ***storing***" the graphs (*Numbers 7 and 8*) are detailed below.

9.1.- *Saving Graphs.*

The process of saving graphs is very simple. Click on the icon of tool number 8 (Save the figure).



This opens an explorer window, similar to those in Windows (depending on the language or system used), where you can select the folder or directory where the graph will be saved.

Back to Index

Additionally, provide a name and select the desired format type. This can be done at the bottom of the explorer window (red circle in the image), where various format types available for saving are selected. The screen resembles the following.

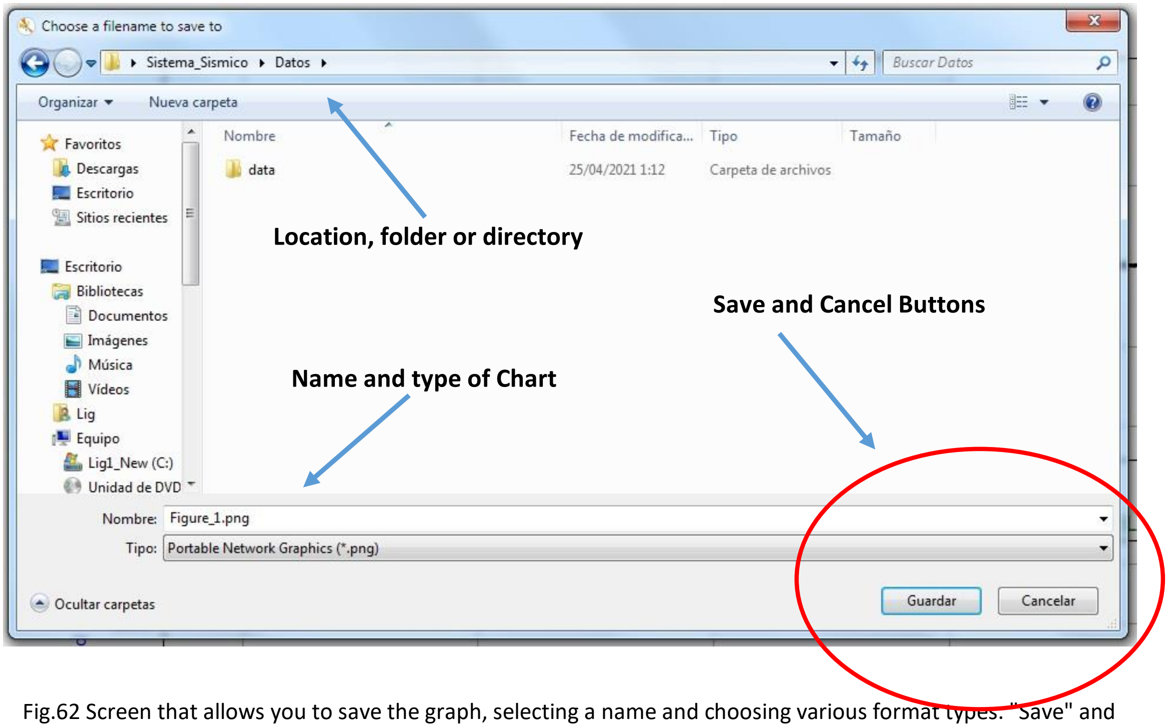


Fig.62 Screen that allows you to save the graph, selecting a name and choosing various format types. "Save" and "Cancel" buttons are provided to complete or cancel the process.

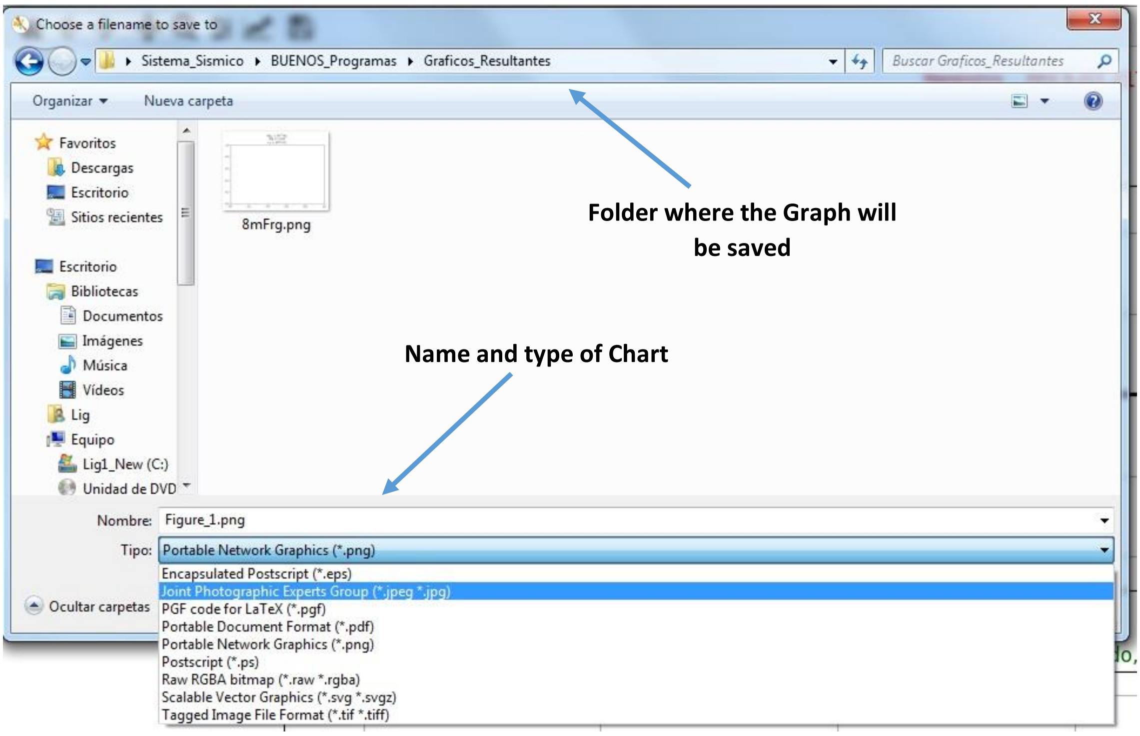


Fig. 63 Screen where you can see the types of formats available to save the graph.

The previous figure shows a list of the available file formats, the following image presents this list in more detail:

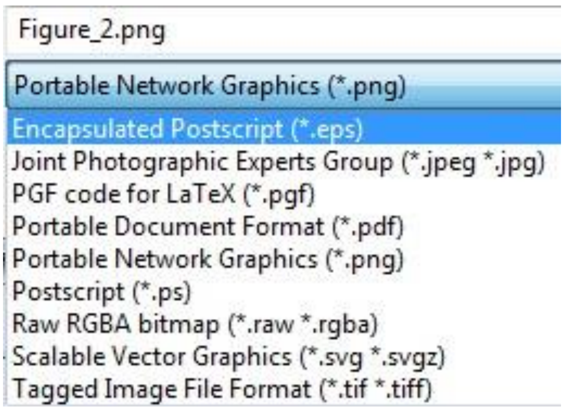
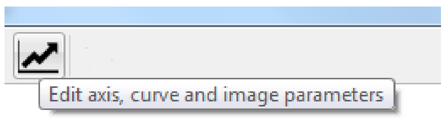


Fig. 64 List of available formats for saving the graph generated by the analysis.

Once you have selected both the name and the desired file format, and the location of the folder or directory where the graph will be saved, click the "*Save*" button (*See Fig. 62*), and the graph will be stored and available for further use as needed.

6.2.- Editing Axes and Images in Graphs.

Through the "Edit" button, point 7 (*See page 41*) of the graph toolbar (*Edit axis, curve and image parameters*), it is possible to edit or modify the parameters of the axes, images, and curves of the graphs.



For example, to modify the parameters of the image of a spectrogram, click on this command button. A "*Customize*" dialog box appears, indicating which of the "*axes*" in the graph areas you want to edit or modify. After selecting, click the "OK" button. This dialog box is similar to the following:

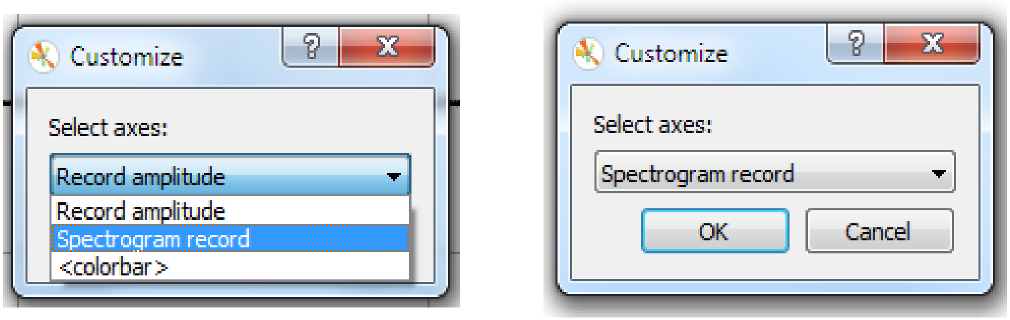


Fig. 65 Customize dialog box, the spectrogram axis has been selected.

Once the desired axis is selected, and the "OK" button is clicked, a new window with the options in the figure is presented. Here, various values of the selected axis are edited, in this case, the spectrogram (*Axes and Images*). The dialog box is as follows:

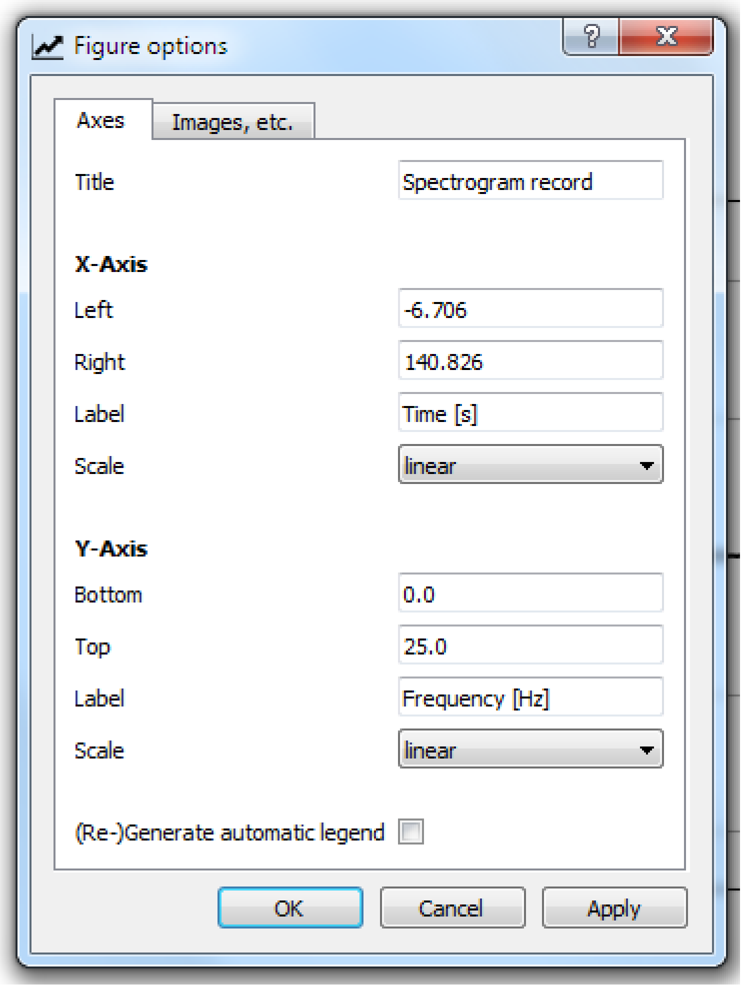


Fig. 83 Dialog box for editing options for the title and axes of the figure.

In this "*Axes*" section, as you can see, you can edit or modify the values or parameters of the title and the "X" and "Y" axes of the graph. For our example, we want to modify the image, so we will select the tab indicating this option. The image presented is as follows:

Back to Index

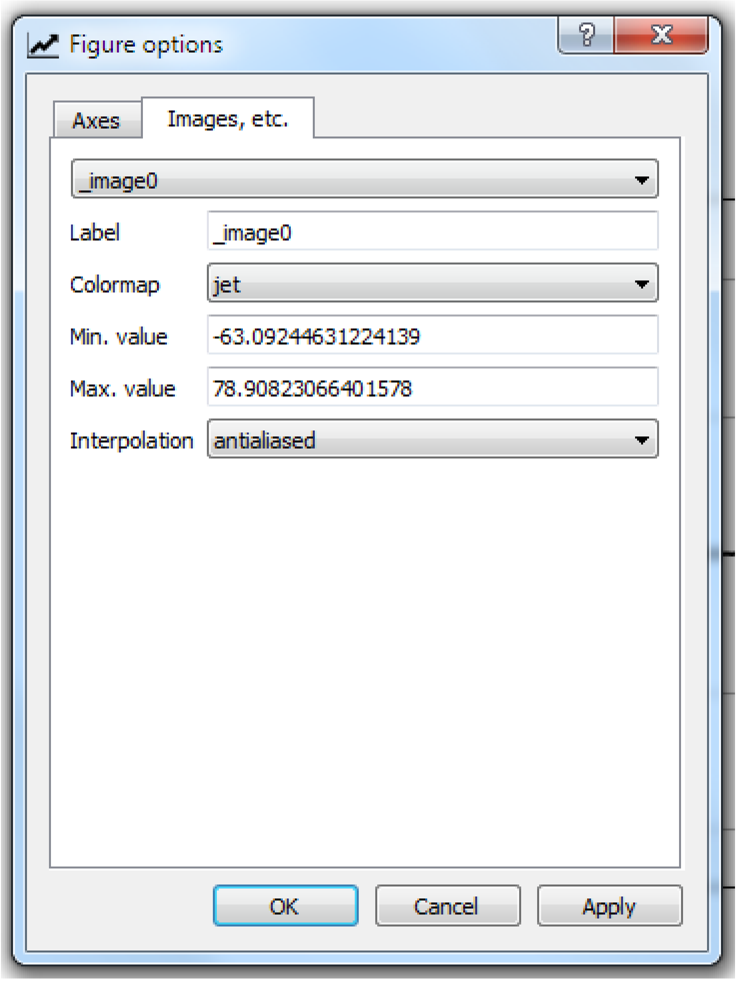


Fig. 67 Dialog box for editing options for image parameters.

As seen in the image, various parameters can be modified, including labels, the color map or "*Colormap*", used in the spectrogram, minimum and maximum values, and interpolation. The default value for the "*Colormap*" is set to "**jet**". The minimum and maximum values for this color map and the interpolation used are assigned by default to the image, but they can be modified according to the operator's interest.

The list of editable parameter values for both "*Colormap*" and "*Interpolation*" is presented in the figure on the next page.

Back to Index

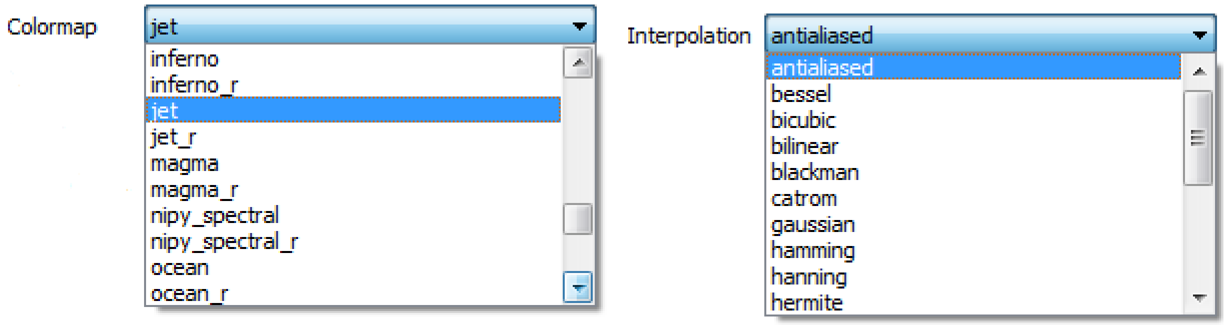


Fig. 68 Dialog boxes for editing some of the parameters of "*Colormap*" and "*Interpolation*" to select in the graph.

CONCLUSION: The system is designed to be a tool that is easy to use, access, and understand. It features a user-friendly interface that offers reliable technological assistance to the human operator in the analysis of both tectonic and volcanic seismic records. The simplicity of this first version lies in the fact that it consists of a single module, which includes several filter analyses and STA/LTA triggering methods, commonly used in the study of a specific seismic signal. In versions or updates following the current version (1.1), additional modules may be added, containing various types of analyses, functionalities, or different STA/LTA algorithm methods to enhance research and study within the scientific community

Acknowledgments:

This software and its documentation are the result of research from Spanish projects:

a) PID2022-143083NB-I00, “LEARNING”, funded by MCIN/AEI /10.13039/501100011033
b) JMI and LG were partially funded by the Spanish project PROOF-FOREVER (EUR2022.134044)
c) PRD was funded by the Ministerio de Ciencia e Innovación del Gobierno de España (MCIN), Agencia Estatal de Investigación (AEI), Fondo Social Europeo (FSE), and Programa Estatal de Promoción del Talento y su Empleabilidad en I+D+I Ayudas para contratos predoctorales para la formación de doctores 2020 (PRE2020-092719).
d) Spanish Project PID2022-143083NB-100 founded by MCIN/AEI/10.13039/501100011033 and by FEDER (EU) “Una manera de hacer Europa”.

PLEC2022-009271“"DigiVolCa”", funded by MCIN/AEI, funded by MCIN/AEI/10.13039/501100011033 and by EU «NextGenerationEU/PRTR», 10.13039/501100011033.

END of the document.

Ligdamis A. Gutiérrez E. PhD.
Theoretical and Cosmos Physics Department, Faculty of Sciences.
Geophysics Andalusian Institute.
Granada University (Ugr)
Granada, Spain – 2023

Theoretical and Cosmos Physics Department

Geophysics Andalusian Institute

Back to Index

APPENDIX A

A1.- Installation of Python and Additional Libraries

A1.1. Package Content.

The main folder "***Analysis_System_1***", It contains two folders with the programs (codes and interfaces in English and documentation in Spanish and English), organized as follows:

a) Folder "***Analysis_System_1***": (*seismic analysis system assembly*). This folder must be copied to "*My Documents*". Contains the following elements:

   a. Subfolder: "***Images***" " Images necessary for program interfaces.

   b. Program: ***Menu.py***. Startup program, presentation and calling of individual modules.

   c. Program: "***Sta_Lta_trigger_methods.py***". Main program (*interface*), for the analysis of seismic records, using the both trigering methods (*Recursive and classical*) of the STA/LTA algorithm, with and without filter (*current program, plus the rest of the modules*).

b) Folders (*Document_ES and Document_EN, depending on the language; English or Spanish*). For *English* "***Document_EN***": It consists of the following items:

   a. User Manual for the module: "***2_Manual_Sta_Lta_trigger_methods_EN.pdf***" in PDF, written in English, with the necessary documentation for the use of the system interfaces. In addition, the rest of the manuals for the other modules (1-10) that make up the system are found.

   b. "***Initials Requirements.txt***" file. File containing the libraries needed to be installed on Windows through "Pip", once Python is installed.

   c. File "***README.txt***": File with general instructions for system installation.

   d. File "***Set_tools_System_1_1.bat***", batch processing executable file. It must be copied to the desktop, from there by right clicking "run as administrator", it will start the system by calling the main menu. The file will automatically search for the startup program (Menu.py) that is located in the "Set_tools_System_1_1" folder that has been previously copied to "My Documents" and will start Python, executing said program.

The system has all the elements (*programs and interfaces*) in English, except for the user manual, which is written in both Spanish and English. To install on Windows, two main actions should be taken after downloading and extracting the ". Rar" files. The first is to copy the entire folder (a) to the "*My Documents*" folder on the PC.

a) From the main folder (**Analysis_System_1**), copy the subfolder "**Document_EN**" to "**My Documents**" in Windows.

b) Copy the file "**Set_tools_System_1_1.bat**", from the "(Document/Document_ES or Document_EN)", depending on the version (Spanish or English), to the Windows desktop.

This ensures the proper use of the program. Now, we will proceed with the installation of the Python language and additional Python libraries on Windows.

A1.2.- Installing Python on Windows.

Python is an interpreted, multi-platform, and multiparadigm programming language (*it works on various operating systems, including Windows, Linux, and Mac*), utilizing two or more programming paradigms within a program-object-oriented, reflective, imperative, and functional.

In addition, Python can be enriched by a large number of programming modules, libraries, packages, or libraries installed through its package manager, "**Pip**." On Linux, the Python program and its manager "**Pip**" are installed together with the operating system. In Windows systems, however, where Python is not a native language, it is necessary to install this language beforehand by downloading the appropriate version from the Python distribution website at the following address: https://www.python.org/downloads/

On the website, the correct version should be selected based on the type of operating system on the computer, including whether it is 32 or 64 bits.

To be installed on both 32 and 64-bit systems, it is essential to note that this document and the software were created with the version available at that time, which was "Python 3.8.6", and many more versions have emerged since then. A more modern and adaptable version to the software (recommended) is "Python 10.10".

Users need to check if more advanced versions do not interfere with some of the installed libraries, such as "*Obspy*," for example. This is because everything related to Linux systems is constantly changing with updates that Python and Linux-based systems make. It is advisable to visit the website and download the most stable or tested updated version of Python that works well with this software.

Once downloaded, run it as an administrator (*right-click and "run as administrator"*), and the software installation wizard will guide you through the necessary steps (*just follow the instructions*).

The process takes only a few minutes. It is "recommended" to indicate during the process, when asked, to include an access path in the system's "**Path**" so that Python can be accessed from any location in Windows. If this is not done during the installation process, it must be done manually by modifying the environment variables (more complicated) to include the path from where Python is installed. This will not be necessary (*if indicated at the beginning*) through the installation wizard.

A1.3.- Installation of Additional Libraries.

The next step is to ensure that Python and its file manager or package manager (**Pip**) have been installed correctly. "**Pip**" (*file and library manager*) is crucial because it allows the installation of additional libraries that Python needs to run the created programs. To do this, open the Windows console window, or "*CMD*." The CMD, or command prompt, is a command-line interpreter.

Accessing the CMD is possible by typing, searching for the Windows logo key (a window), located between the "Ctrl" and "Alt" keys at the bottom left of the keyboard. Pressing this key, plus (+) the letter "**R**" key, will open a "Run" program window, similar to the following.

Back to Index

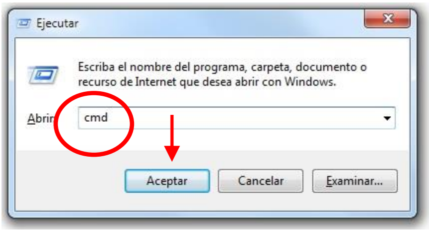


Fig. A1 Screen run in Windows. In the red circle, type "cmd" and click "OK."

As seen in the figure above, type "cmd," click "OK," which will open the Windows command prompt window.

Another way to do this is at the bottom of the desktop, in (W7) or next to (W10) the Windows "Start" button. There is the search section, indicated by the magnifying glass icon. This indicates a search for programs, similar to the following.

Fig. A2 Windows Program Search Screen.

In the box that says "Search programs and files" (Windows 7) or "*Type here to search*" (Windows 10), type "cmd" as well. This action or the previous one will bring up the Windows command prompt (CMD), similar to the following (W7).

Fig. A3 Windows 7 Command Prompt (CMD) Screen.

The same applies to versions: Windows 10 (W10) or Windows 11 (W11).

Back to Index

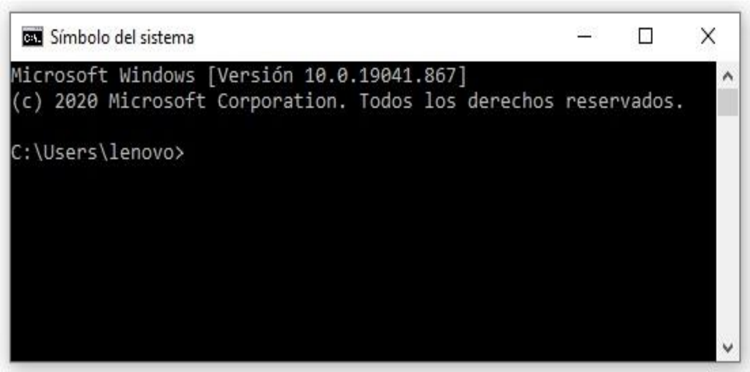


Fig. A4 Command Prompt (CMD) Screen in Windows 10.

Once on this screen, to verify that both Python and its package manager "pip" have been installed correctly, type the following commands: Python -V, and to check "pip," type: pip -V. This is shown in the following figure.

Fig. A5 CMD Screen, indicating Python and pip versions in Windows.

The output of typing "-V" in Python indicates invoking the installed version. In this case, it can be seen that it is "3.8.6" (*The library set is compatible with 3.10.10 version*). This has been possible from any location in the system because the Python script has been installed, remember, in the "**path**" or route found in the system's environment variables. Also, after typing "pip -V", it can be seen that the Pip version is "*20.2.1.*" At this point, it is recommended to update this version since, by default, "Pip" is installed along with "Python", but it does not install the latest or most up-to-date version. To do this, in the CMD window or console, type the following command (*Windows/Linux*): On Windows, type "python," and on Linux, type "python3".

Windows: > python –m pip install –upgrade pip | Linux: $ sudo python3 –m pip install –upgrade pip

This indicates that "Pip" will be updated to its most recent version (*On Linux, as a "superuser," i.e., with "sudo" at the beginning*). It is shown in the following screen.

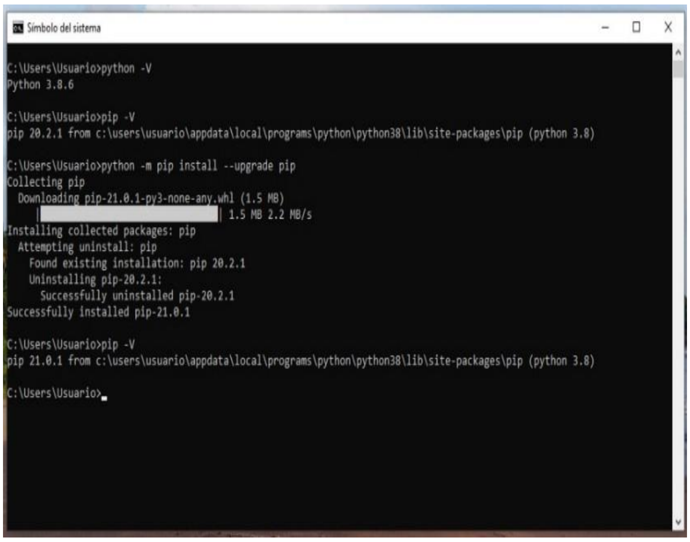


Fig. A6 Screen showing the update and verification of the new version of pip in Windows.

As can be seen, when typing again (**pip -V**), once Pip is updated, the version is 21.0.1. With this, Python and Pip are already installed and updated. Pip, as mentioned, is very important because with this manager, all the necessary libraries and packages are installed so that Python applications can be executed correctly and without errors. To use the system, you must proceed to install the necessary packages or libraries through Pip.

Next, we will proceed to explain how, in a simple and completely automatic way, the most commonly used and general libraries that Python needs will be installed on the system. Libraries such as, for example, "*obspy*," which is the open-source library or software based on Python for processing seismological data. Also, "***matplotlib***," which is a library for generating graphics from data contained in lists or arrays in Python and its mathematical extension "***NumPy***," among others, which the system needs for its execution (*See Annex B*).

A1.4 Automatic Installation of Libraries on Windows and Linux from PIP.

The advantage of having already installed and updated Pip in Windows is that you can install all the libraries that Python needs to run the system.

Additionally, in the "***Document***" folder, the "***Readme.txt***" file contains instructions for this installation. So the user only needs to follow the instructions, and the necessary packages will be installed on the computer (PC) automatically by Pip, both on Windows and Linux. The required libraries are in the file called "***Initial_requirements.txt***", included in the "***Document***" folder of the downloaded installation files and in **Annex B**.

In a Windows Command Prompt (**Cmd**) window, actions are taken for each of the commands indicated in the file, following the instructions. The installation should not present problems on Windows and Linux systems. If any library encounters an error during installation (*shown in red in CMD*), you should consult the documentation for that library or check if the correct or recommended version of Python is being installed (*version 3.8.6 and/or 3.10.10*). The installation on Linux systems (*See README.txt*) is similar and simpler. Copy the main folder to the desktop, the personal folder, etc. From that location, open a command prompt, and simply type:

"***$ python3 Menu.py***" to start the system.

Back to Index

APPENDIX B:

INSTALL PYTHON LIBRARIES FOR THE PROPER FUNCTIONING OF THE SYSTEM.

1.- **PIP**: The **Pip** (*Preferred Installer Program*) is the package or package management manager used to install and manage software packages written in Python. When installing Python, Pip is installed by default. To check the version of Python or PIP, type the following in a console or CMD:

python - V / pip – And to see the list of installed pip packages: -> pip list

Usually, you need to update the version of pip with which Python is installed. For this, type the following command in the command prompt (CMD). In Linux and Mac systems, "**sudo**" is placed at the beginning to indicate super-user permissions.

Python –m pip install - - upgrade pip / (LINUX) -> sudo python –m pip install - - upgrade pip

Once downloaded and installed, you can check the version again with the first command, and you will see that the version has changed and been updated. Now that pip is updated, we will proceed to install the necessary packages for Python to work correctly with the applications.

2.- .- **PyQt Installation**: This is a Python binding for the Qt library written in the C++ language. It is used for creating and using graphical user interfaces (GUI) in Python. Type the following in the command prompt (CMD).

pip install PyQt5 / (LINUX & Mac) -> sudo python install PyQt5

3.- **Matplotlib library Installation**. Matplotlib is the library that allows the creation and visualization of graphics. Type the following:

pip install matplotlib / (LINUX & Mac) -> sudo python install matplotlib

4.- Install the **Obspy** library. This library is for handling seismic signals. Type the following:

pip install obspy / (LINUX & Mac) -> sudo python install obspy

5.- Install **Thinter**: Thinter is a graphical user interface (GUI). Type the following:

pip install tk / (LINUX & Mac) -> sudo python install tk

6.- Install **quantecon**: This library is used for spectrum estimation, Periodogram, Fourier transform. Type the following:

pip install - - upgrade quantecon / (LINUX & Mac) -> sudo python install - - upgrade quantecon

7.- Update a library for **matplotlib**. To avoid problems with graphics, install the following:

pip install msvc-runtime / (LINUX & Mac) -> sudo python install msvc-runtime

8.- Install **easygui** for the graphical interface:

pip install easygui / (LINUX & Mac) -> sudo python install easygui

9 Install **PyWavelets** for CWT handling.

pip install PyWavelets / (LINUX & Mac) -> sudo python install PyWavelets

10.- Install **plotly** for handling and assisting with graphics along with Matplotlib.

pip install plotly / (LINUX & Mac) -> sudo python install plotly

11.- Install "**pyaudio**", for audio management. Python bindings for PortAudio v19, the cross-platform audio I/O library

python -m pip install pyaudio / (LINUX & Mac) -> sudo apt-get install python3-pyaudio

Finally, type "pip list" to see the installed libraries. Optionally, you can create a file called "***requirements.txt***" that will contain all the libraries that the PC will use. The "***requirements.txt***" file must be in the current directory. The instruction to do this is as follows:

pip freeze > requirements.txt