Pytheas User's Guide

Release 0.1.0

Ioannis Spingos, George Kaviris

Apr 16, 2019

CONTENTS

1 What is Pytheas? 2

1.1 Who is Pytheas for? . . . 2
1.2 Highlights . . . 2
1.3 Acknowledgements . . . 3
1.4 Got any feedback? . . . 3

2 Installation 4

2.1 Compatibility . . . 4
2.2 Dependencies and Installation . . . 4
2.3 Download . . . 5
2.4 System Requirements . . . 5

3 Getting Started 6

3.1 Running an example . . . 6
3.2 Acquiring data and metadata . . . 6

4 Basic Description 8

4.1 Available Methods . . . 8
4.2 Visual Inspection (MAN) . . . 8
4.3 Semi-automatic Methods (EV, MC and RC) . . . 9
4.4 Cluster Analysis (CA) . . . 10

5 Other Features 11

5.1 TauP . . . 11
5.2 Database . . . 11
5.3 Figures . . . 12

6 Preferences 13

6.1 General . . . 13
6.2 Grading . . . 14
6.3 Clustering . . . 14
6.4 TauP . . . 16
6.5 AR-AIC . . . 16

7 Controls 17

7.1 Toolbar Options . . . 17
7.2 Key bindings . . . 19

8 Frequently Asked Questions (FAQ) 21

The current User's Guide documents the functionality of the corresponding software version.

CHAPTER

ONE

WHAT IS PYTHEAS?

Pytheas is an open-source software solution for local shear-wave splitting studies. Shear-wave Splitting (SwS) is the phenomenon of two, orthogonally polarized, shear-waves, produced from the same original wave, propagating in an anisotropic medium. SwS has been explored in various branches of Seismology, such as earthquake precursors (Crampin et al., 1999) and fracture characterization in reservoirs (Al-Harassi et al., 2010).

After performing multiple SwS studies with a cumbersome blend of Shell scripts and Fortran programs (e.g. Kaviris et al., 2017), we decided to start developing a new application for analysis, back in 2016. We selected the Python language due to its open-source character, high level features and increasing popularity in Seismology. Even though we started with the intention to simply develop a comfortable GUI for the manual method of visually inspecting particle motion diagrams, the program started to evolve into something more. Fast forward ~3 years (and a MSc thesis) later, we developed a fully-fledged SwS application that includes several popular methods of analysis and aims to introduce a new workflow scheme, by mixing both manual and automatic methods, under a unified and intuitive GUI. By achieving the above, Pytheas' goal is to make analyzing SwS in local recordings easier, more comfortable and more effective.

1.1 Who is Pytheas for?

Pytheas was developed with both beginners and experts in mind. Despite offering a plethora of customizing options, the application should have a smooth learning curve. The intuitive implementation of the visual inspection method can be used both as a research (e.g. for establishing a quality control sample of measurements before using any automatic methods) and a teaching (actually 'seeing' the polarization direction in the particle motion diagrams and then 'moving' the waveforms in time can have a vivid impression on students and help them better understand how splitting works) tool. Striving for the best user-experience, we hope to avoid intimidating students and early-career scientists with requiring terminals and commands to operate the software. Experienced researchers can benefit from the combination of multiple methods to obtain more robust results, while the integration of common file formats will trivialize the hassle of acquiring and converting data.

1.2 Highlights

- Includes the most popular SwS analysis methods (Visual inspection of particle motion diagrams, Rotation-Correlation, Eigenvalues, Minimum Energy, Cluster Analysis).
- All functionality implemented through the GUI.
- Permits the seamless use of different methods, offering a new approach to the quality control of results.
- Compatible with standard formats for both data (e.g. SAC and mSEED) and metadata (QuakeML and StationXML).

- Provides a results-management framework that includes databases, the production of publication-ready figures and comprehensive results files.
- Open-source, which means the ability for user-customization.

1.3 Acknowledgements

One of the projects that had a major impact on the development team of Pytheas is Obspy. Obspy managed to offer a very valuable tool to Seismology and provide the foundations required to build sophisticated Seismological software in Python.

Every modern programmer would be omissive to not acknowledge the StackOverflow community, which has been a great help in getting started with programming.

Finally, we would like to thank the developers of Sphinx for providing the tools to generate this lovely documentation.

1.4 Got any feedback?

If you have any issue, comment or suggestion feel free to contact us at ispingos@geol.uoa.gr. You can also use the software's dedicated GitHub page at https://ispingos.github.com/pytheas/

CHAPTER

TWO

INSTALLATION

2.1 Compatibility

Pytheas has been tested on various systems running Windows 10, using Python 3.7.2. The program has been tested on the following versions of individual packages:

- obspy 1.1.1
- PyQt5 5.11.3
- scikit-learn 0.20.2
- matplotlib 3.0.2
- numpy 1.16.0
- scipy 1.2.0

2.2 Dependencies and Installation

To use Pytheas you first need to install Python 3 and then the following dependencies:

- obspy
- PyQt5
- scikit-learn
- configparser

You can use the provided scripts to install dependencies. For Windows users this is `install_reqs_windows.bat` and for Linux users `install_reqs_linux.sh`.

Alternatively, you can install the packages by `pip` using the following commands in a terminal (Linux/MacOS) or command prompt (Windows):

```
pip install obspy
pip install PyQt5
pip install scikit-learn
pip install configparser
```

Remember to also update `numpy`, `scipy` and `matplotlib`:

```
pip install --upgrade numpy
pip install --upgrade scipy
pip install --upgrade matplotlib
```

Another option is using a distribution like Anaconda.

No matter which one of the above you prefer, the program runs by using the `LaunchPytheas.pyw` file. Windows users can use `LaunchPytheas.exe` instead.

2.3 Download

You can download the Python code and related files from https://www.github.com/ispingos/pytheas-splitting .

2.4 System Requirements

Hardware requirements are largely dependent on the analyzed waveforms. We tested Pytheas on four different systems with the following results.

Table 1: **System Requirements**

System	CPU	RAM	GPU	User Experience*
ASUS Vivobook X5600UD	Intel Core i7 8550U	8 GB	Nvidia GTX 1050 (4 GB)	A+
Custom Desktop	Intel Core i5 8400	8 GB	Nvidia GTX 950 (2 GB)	A+
Dell Inspiron 15 5567-1753	Intel Core i7 7200U	8 GB	AMD Radeon R7 M445 (4 GB)	A
Custom Desktop	Intel Core i5 3470	4 GB	Nvidia GT 640 (2 GB)	B
Toshiba Satellite m50-a-110	Intel Core i5 4200U	4 GB	Nvidia GT 740M (2 GB)	B-

User Experience is based on subjective criteria, among three different analysts.

A dedicated GPU is generally recommended, especially if you intend to work extensively with the MAN method.

CHAPTER

THREE

GETTING STARTED

3.1 Running an example

Important Before you start the process, go to the `example` folder and unzip the data in the `data.zip` archive.

Start the program. You (might) get a warning message stating that no Station Information file was found. In this case, open the Preferences menu (either through the toolbar in File->Preferences or with Shift+P) and then go the TauP tab. Browse to the example folder included in the release and select the StationXML file (*wgoc_stations_eida.xml*). **Restart the program for the changes to take effect.**

After restarting, you will get no new warning message. To open the example, use the Open Catalogue menu option (either through the toolbar in File->Open Catalogue or with Shift+O). A new window will ask for:

1. The path to the main data directory (*<path>/<to>/<pytheas>/example/data*)
2. The path to the Catalogue file (*<path>/<to>/<pytheas>/example/wgoc_events_example_NKUA.xml*)
3. The path where the Database will be stored (*<path>/<to>/<pytheas>/example/data/wgoc_events_example_NKUA.db3*)

Browse, in each case, to the files included in the example folder. You can now press *Ok* to start the analysis! Keep in mind that the example file also includes a `rigo_crustal.nd` file, which is a local velocity model for the Western Gulf of Corinth (where the sample waveforms and events are from).

3.2 Acquiring data and metadata

Pytheas requires three inputs to work:

1. **A directory containing the data:** This can be of any name, but the final branch of the directory tree should contain folders named after the events in a %Y-%m-%d-%H-%M-%S format. For example, *~/my_data/2019/04/01/2019-04-01-22-19-32* and *~/my_data/2019-04-01-22-19-32* are the same for Pytheas.
2. **A compatible Catalogue file: This can be either of a QuakeML format or a simple ASCII file containing event information**

   ```
   2019 04 14 21 00 00.000 38.0100 23.8314 13.0 6.0
   ```

3. **A compatible Station Information file: This can be either of a StaionXML format or a simple ASCII file containing station**

   ```
   ATHU HA 37.9665 23.7845 308
   ```

You can find scripts for downloading data and metadata from the FDSN in the *acquisition_scripts* folder, inluded with Pytheas.

QuakeML and StationXML

It is **highly** recommended to use QuakeML and StationXML files. QuakeML can contain additional information vital for the use of Pytheas (i.e. S arrivals, S-P times, angle of incidence). StationXML can contain equally important information about the stations (i.e. the instrument orientation). You can acquire those types of files through popular services (e.g. WebDC3 and IRIS DMC) via the provided scripts. In a future update, we plan to implement this functionality through Pytheas itself. For Greece, you can acquire event metadata in QuakeML format from the National and Kapodistrian University of Athens - Seismological Laboratory website by going to the detailed event information of each event.

Some parameters will need to be calculated for Pytheas to work, if a QuakeML is not provided. The angle of incidence will be calculated geometrically, considering a linear ray path. In such cases, it is important to use TauP to (re)calculate accurate angles.

CHAPTER

FOUR

BASIC DESCRIPTION

Pytheas offers a lot of functionality in terms of analyzing shear-wave splitting. The user can benefit from manual, semi- and fully- automatic methods, used separately or in tandem.

4.1 Available Methods

The following shear-wave splitting methods are available through Pytheas. The references provided are indicative and by no means cover the topic of each technique or its application.

Table 1: **Methods**

Name	Short Description	Reference	Notation
Visual Inspection	The manual method of visually inspecting particle motion diagrams	Kaviris et al. (2017)	MAN
Rotation-Correlation	Semi-automatic method which seeks the ϕ-t_d pair that provides the highest similarity of the horizontal components	Bowman and Ando (1987)	RC
Eigenvalue	Semi-automatic method which seeks the ϕ-t_d pair that minimizes the second eigenvalue of the covariance matrix of the two horizontal components	Silver and Chan (1991)	EV
Minimum Energy	Semi-automatic method which seeks the ϕ-t_d pair that minimizes the energy of the transverse component	Silver and Chan (1991)	ME
Cluster Analyis	Method for automating RC, EV and ME by seeking the most stable solution over a range of measurements	Teanby et al. (2004)	CA

Hereafter, the notation for the semi-automatic methods uses a ‘M’ prefix if the signal window for analysis is provided by the user (**MRC, MEV and MME**) or a ‘A’ prefix if the signal window is obtained from CA (**ARC, AEV and AME**).

4.2 Visual Inspection (MAN)

The MAN method demands the full involvement of the analyst in all steps. It displays the motion in the horizontal plane defined by the N-S and E-W instrument components to enable the determination of polarization and, then, measure the time-delay. It can be broken down to three steps:

1. **Initial Stage:** Pick the arrival of the S_{fast} phase in one of the horizontal components. The respective vector in the polarigram will also be selected and the ϕ value at the top will change accordingly.
2. **Rotated Stage:** Rotate the waveforms to the axial system defined by ϕ and its orthogonal S_{slow} polarization direction (Ctrl+2). Temporally shift the S_{slow} component (with the Arrow Keys) to match the arrival of the

shear-waves in both horizontal components. In the meantime, the t_d and *pol* values at the top will change dynamically. The latter refers to the polarization direction of the corrected (for anisotropy) waveforms.

3. **Corrected Stage:** Rotate the wavefroms back to the NE axial system (Ctrl+3) to validate visually the measurement. The S-wave should now arrive simultaneously in both horizontal channels and the polarigram around its arrival should present parallel vectors. A qualitative grade must be given by the user (Ctrl+G).

Manual Grading

The manual grade is a synthesis of multiple factors, such as clarity of the initial S_{fast} arrival, the parallelism of vectors around the arrival in the FS system and the degree of correction achieved as decided by the analyst. If no grade is given by the user, the default 'X' will be set. In that case, the measurement will **not be stored in the database**, since 'X'-graded results are not considered as final measurements.

Even though the above description is how the method has been used so far, Pytheas gives the opportunity to skip the *Rotated Stage* and perform the time-delay determination in the *Corrected Stage*. The program will automatically rotate the waveforms to the *Rotated Stage*, apply the time-delay and re-rotate them to the *Corrected Stage*, without the involvement of the user. This new addition to the method can facilitate small improvements in measuring the time-delay.

Pytheas also offers the ability to set a maximum angle of incidence. This quantity is often used to exclude station-event pairs where there is potential for secondary phases to distort the arrival of the direct S-wave. Thus, by setting a maximum angle of incidence (*shear-wave window*), the analyst can simply use the keybinding to navigate to the next available station-event pair, with the ones outside the shear-wave window being automatically skipped.

4.3 Semi-automatic Methods (EV, MC and RC)

The application principle for all three methods is the same. The analyst tries different signal windows around the arrival of the S_{fast} to obtain the one that provides the best measurement, with the start of the window usually being somewhat steady. In Pytheas, after the user has specified the window and selected the analysis method, a new automatic grading algorithm provides a score and qualitative grade for the measurement. Moreover, a figure is shown to permit the quality assesment of the result by the user.

Grading Algorithm

To enable the use of a fully automated process, Pytheas employs a new grading algorithm. The quality of the measurement is determined from five factors; (a) the error of ϕ, (b) the error of t_d, (c) the Signal-to-Noise Ratio (SNR), (d) the Correlation Coefficient (CC) for the corrected waveforms in the FS axial system and (d) the CC for the corrected waveforms in the NE axial system. First, a check is performed to ensure the measurement's SNR is above the minimum accepted. For each of the other parameters, an individual score is calculated, based on the ratio of the parameter and user-defined accepted tresholds. If any of them exceeds 1, the measurement gets an 'E' grade. The mean of the scores provides the final score and, consequently, grade. Most of the above can be customized by the user in the Preferences. However, it is important to note that the final score (since it's normalized to all four parameters) cannot exceed 1. Thus, a perfect measurement would have a final score tending to 0 and a very bad quality one would have a score close to 1. There are two special cases of grades. The 'N' grade is given when a measurement is considered null, i.e. when ϕ is either (sub)parallel or (sub)perpendicular to *pol*. The 'F' grade is used privately by the program to characterize event-station pairs that existed but yielded no result or an error occured during Cluster Analysis.

To conduct a measurement with the EV and ME methods, the waveforms are first rotated to the ray-defined axial system. Pytheas permits the rotation of waveforms to either ZRT (i.e. rotated only by the backazimuth) or LQT (i.e.

rotated by both the backazimuth and the angle of incidence).

4.4 Cluster Analysis (CA)

Cluster Analysis automates the window selection process described above. The program conducts measurements over a large number of candidate windows. The measurements, which form a space defined by ϕ and t_d, are then clustered. The optimal window is the one that belongs to the most constrained cluster and features the smallest errors. If it is not already provided, the Pytheas user needs only to pick the arrival of the S_{fast}. The grading algorithm provides a qualitative grade and two figures are generated; one detailing the clustering process and one showcasing the measurement in the optimal signal window.

CA can be used in two ways. Either after opening a new station-event pair or by selecting the Catalogue Cluster Analysis option, where the whole catalogue will be processed. The latter offers some refinement options, i.e. the selection of specific stations, the setting of the shear-wave window and a universal filter. Moreover, it offers the ability to skip already analyzed pairs or ones that where previously processed but yielded no result (*failed* pairs graded with 'F'). Keep in mind that for EV and me, the waveforms will be rotated in either ZRT or LQT, as defined in the Splitting menu from the toolbar.

CHAPTER

FIVE

OTHER FEATURES

Pytheas also offers a wide variety of secondary features that enable the faster and more effective analysis of shear-wave splitting.

5.1 TauP

The TauP software (Crotwell et al., 1999) is a collection of tools used for estimating phase arrivals and ray-paths. In Pytheas, we use Obspy's implementation of TauP to calculate arrivals of shear-waves and the angle of incidence of each ray. The former parameter is useful when identifying the S_{fast} visually, as noise can complicate this. The latter is used to calculate the precise angle of incidence, a quantity of great importance for shear-wave splitting studies. This angle is used for rejecting potential secondary arrivals misidentified as direct S-waves by setting the shear-wave window (Booth and Crampin, 1985) and for rotating the waveforms to a ray-defined system for the EV and ME methods.

The velocity model used by TauP, defined in the Preferences, can either take a value of builtin models (e.g. `IASP91` or `ak135`) or a custom model file can be specified. In this case, the file format should be either of `.nd` or `.tvel` type. For more information about custom model building please see the original TauP documentation. A simple `.nd` file example is as follows:

```
# Ceiling_depth_km Vp Vs Density
0.0 5.0 3.0 2.7
20 5.0 3.0 2.7
20 6.5 3.7 2.9
33 6.5 3.7 2.9
mantle
33 7.8 4.4 3.3
410 8.9 4.7 3.5
410 9.1 4.9 3.7
670 10.2 5.5 4.0
670 10.7 5.9 4.4
2891 13.7 7.2 5.6
outer-core
2891 8.0 0.0 9.9
5149.5 10.3 0.0 12.2
inner-core
5149.5 11 3.5 12.7
6371 11.3 3.7 13
```

5.2 Database

We have integrated the Python implementation of SQLite to enable reading, writing and seeking results faster. The database is comprised by three tables: (a) *Event* contains the information concerning the event (i.e. origin time and location information), (b) *Station* contains information concerning the event-station pair (e.g. epicentral distance in km, backazimuth, angle of incidence, observed/theoretical/automatic S-arrivals) and (c) *Method* contains information concerning the splitting measurement(s) for each pair. Tables are connected through the event, station and method codes. Each measurement is characterized by a unique combination. If the program finds the combination in the database, it will overwrite the entry. We have not implemented an in-program way to remove entries from the database, to avoid erroneous removals. Thus, any such action should either be conducted on the exported results files or through a third-party database editor (see below).

Each database is saved locally on a user-specified path. There are various actions available through the Database toolbar menu, such as copying the database, loading a new one or extracting information in more user-friendly formats. Concerning the latter, there is the option to extract a common CSV file, which is easily processed with any spreadsheet editor, such as OpenOffice's Calc. The user can also export a custom text format (*spl*) which contains less information and is more human-readable.

The database, except simple values corresponding to each measurement such as ϕ, t_d and their corresponding errors, also contains any arrays generated through processing concerning the EV, ME, RC and CA methods, such as the distribution of the second eigenvalue per parameters pair and the cluster labels for each number of clusters. This information is used to generate figures on demand, through the toolbar menu. However, it is not possible to access the arrays themselves through the program (as of now). If a user wishes to do so (and they have a rudimentary knowledge of Python and SQLite) they can look up the data conversion code in the *db_handler* module.

A relative simple way to access the database itself is through third-party software. DB Browser is a great tool that offers this, exclusively through a GUI. You can add, remove or edit entries and even export specific selections from the database.

5.3 Figures

Figures are an essential part of every scientific publication. Thus, we have implemented the generation of figures that meet the specifications set by Elsevier to produce publication-ready figures. These specifications include:

- Font sizes of at least 7 pt.
- The dpi set at 720 (maximum accepted for line art is 1000).
- Figure size of 7 x 6" (177.8 x 152.4 mm), meeting the *Double Column* width size (i.e. less than 190.0 mm)

To avoid further dependecies we are restricted to the image format types compatible with the used Matplotlib backend, i.e. Qt5. The available formats for figure generation are: PS, EPS, SVG, PDF, PNG and JPG. Sorry, no TIFF option available!

CHAPTER

SIX

PREFERENCES

6.1 General

Table 1: **General**

Setting	Description
Clean Logs	Sets the amount of days prior for which to delete log files.

Table 2: **Waveforms**

Setting	Description
Offset for matching event folders to origin times	Sets the maximum accepted difference (in s) when attempting to match event folder codes to origin times from the catalogue.
Trim waveforms	Select whether or not you want waveforms to be trimmed when displayed. This is very important as it can dramatically increase performance.
Start time from S-arrival	Time from the S-arrival (in s) to use as the trimming window start point.
End time from S-arrival	Time from the S-arrival (in s) to use as the trimming window end point.
Channel Code Preference	Provide a comma-separated list of the first two channel code letters (according to SEED) in the sequence Pytheas will preferrentially look for them. For example, if both HH and HN type channels exist in the dataset for the same station/event pair and the user has provided the list 'HH,HN', the HH channels will be used. If none of the preferred codes is found, the program will read any channels it'll find first.
Instrument Orientation Correction	Select whether to perform orientation correction on non-oriented waveforms. Requires the use of a StationXML file with information down to the Channel layer.

Table 3: **SNR**

Setting	Description
Start of noise window	Time from the S-arrival (in s) to use as the noise window start point. The end point is defined as the S-arrival.
End of signal window	Time from the S-arrival (in s) to use as the signal window end point. The start point is defined as the S-arrival.

Table 4: **Splitting**

Setting	Description
Maximum accepted t_d	The maximum time-delay (in ms) to use when defining test ranges in the EV, ME and RC methods. This does not affect any measurements with the MAN method.

6.2 Grading

Table 5: **Thresholds**

Setting	Description
Polarization difference for nulls	Sets the angular maximum differences for which the S_{fast} direction and the corrected S-wave polarization (*pol*) will be considered parallel or perpendicular. In other words the value *offset* specified here is used as: 0 - *offset* <= ϕ - *pol* <= 0 + *offset* OR 90 - *offset* <= ϕ - *pol* <= 90 + *offset*. If any of these conditions are met the measurement is considered null ('N' grade).
SNR_{min}	Set the minimum accepted SNR.
$\delta\phi_{max}$	Set the maximum accepted error for the S_{fast} polarization direction (in degrees).
δt_{dmax}	Set the maximum accepted error for the time-delay (in ms).
Minimum Correlation Coefficient (FxS)	Set the minimum accepted correlation coefficient for the correlation in the FS axial system.
Minimum Correlation Coefficient (NxE)	Set the minimum accepted correlation coefficient for the correlation in the corrected NE axial system.
Grades	Set the upper bounds for which the grading algorithm will match scores to grades. The maximum attainable score is 1.0, so individual values should be set accordingly. Any score greater than the one specified for 'D' will be graded as 'E'.

6.3 Clustering

Cluster Analysis

To fully understand the settings in this section we **strongly** recommend reading the article of Teanby et al. (2004). For the parameters concerning the definition of the **Windows Bounds** range you can also see Savage et al. (2010).

Table 6: **Thresholds**

Setting	Description
Minimum window start point (T_{beg1})	Set the minimum accepted window start (i.e. the closest to the S-arrival), relative to the S-arrival (in s).
Step for window start points (DT_{beg})	Set the step with which the start points of candidate windows will be generated (in s).
Step for window end points (DT_{end})	Set the step with which the end points of candidate windows will be generated (in s).
Minimum window end point (T_{end0})	Set the minimum accepted window end (i.e. the closest to the S-arrival), relative to the S-arrival (in s).

Table 7: **Windows Bounds**

Setting	Description
Maximum t_S-t_P	The maximum accepted S-P time (in s). This constraint is use in case of errors or overestimation in the provided times. If an error occurs during the process (e.g. there is no provided S-P time in the catalogue) or the given time exceeds this limit the S-P time used is this. This parameter is used to determine the maximum window start point (T_{beg0}) which is the half of the S-P time (i.e. T_{beg0}=(t_S-t_P)/2).
Time from S-arrival for period determination	Set the end of the time window used for determining the S-wave's period (in s).
Minimum S-wave period	The minimum accepted S-wave period. If a measurement exceeds the limit this will be the value used.
Maximum S-wave period	The maximum accepted S-wave period. If a measurement exceeds the limit this will be the value used.
Period factor	Set the factor (f_T) used to obtain the maximum window end point (T_{end1}) which is defined as: T_{end1} = f_T *T_S where T_S the period of the S-waves.
$C_{critical}$	The critical value for which to reject the null hypothesis in the Duda and Hart (1974) criterion.
Maximum number of clusters	Set the maximum number of clusters the program can look for.
Minimum number of points per cluster	Any clusters containing less than the number of points set here will be rejected.
Linkage Criterion	Set which of the available criteria will be used when linking clusters. For more information you can see sci-kit learn's page.

6.4 TauP

Table 8: **TauP**

Setting	Description
Model	Select the velocity model used for TauP. Either a path to an available file (in the `.nd` or `.tvel` formats) can be provided or a string with the name of one of the default models (see Basic Usage in Obspy's documentation).
Station Information File	Either a StationXML file (including the Channel layer) or a simple space-delimited text file where each row represents a station in the following format: `Station Network Latitude_in_dd Longitude_in_dd Elevation_in_m`.
Calculate Incidence	Select whether the angle of incidence will be calculated anew through TauP for each pair or the one obtained from the catalogue will be used.

6.5 AR-AIC

Auto Regressive - Akaike Information Criterion Picker

The Obspy's implementation of AR-AIC is used in the current software. For further details concerning the settings please see their documentation. For more information about the algorithm itself, see Akazawa (2004).

CHAPTER

SEVEN

CONTROLS

7.1 Toolbar Options

Various functions can be accessed through the toolbar.

Table 1: **File Menu**

Option	Description
Open Catalogue	Starts the analysis process by asking for the data, catalogue and database locations.
Preferences	View and edit the available Preferences of Pytheas.
Quit	Exits the program.

Table 2: **Database Menu**

Option	Description
Save as…	Copies the existing database to a new location and, hereafter, uses the new file instead.
Load New	Loads a new database and uses it instead.
Save Solution to Database	Save active solution to the database.
Load Solution from database	Loads an existing solution from the database. A dialog window will open with a drop-down menu oferring all the available methods from the database.
Export…	Exports the selected event-station-method pairs from the database to a text file (either `spl` or `csv`).
Export Active	Exports only the active solution to a text file.
Export Figure	Exports a figure summarizing the results of the selected method of the active event-station pair.

Table 3: **View Menu**

Option	Description
Grid	Show a grid overlaying the waveform and polarigram plots.
Center S	Centers the current view around the picked S-arrival.
Set X Axis Limits	Prompts the user to set custom limits for the time axis.

Table 4: **Manual Menu**

Option	Description
Initial Stage	Go to the Initial Stage of the MAN method.
Rotated Stage	Go to the Rotated Stage of the MAN method.
Corrected Stage	Go to the Corrected Stage of the MAN method.
Set phi	Prompts the user to set a custom ϕ value.
Set time-delay	Prompts the user to set a custom t_d value.
Set grade	Prompts the user to select a grade from the available in the drop-down menu.
Set comment	Prompts the user to set a custom comment for the measurement.

Table 5: **Splitting Menu**

Option	Description
Set Shear-wave Window	Prompts the user to select a maximum angle of incidence (shear-wave window) for the analysis. Defaults to infinite (i.e. no angle set).
ZRT/LQT	Select to which system waveforms should be rotated for the EV and ME methods. This option persists when using any CA implementation.
Eigenvalue	Use the EV method.
Minimum Energy	Use the ME method.
Rotation-Correlation	Use the RC method.
Cluster Analysis (EV)	Use the EV method with CA in the active event-station pair.
Cluster Analysis (ME)	Use the ME method with CA in the active event-station pair.
Cluster Analysis (RC)	Use the RC method with CA in the active event-station pair.
Catalogue Cluster Analysis	Opens a dialog to set options and initiate the CA method for the whole catalogue.

Table 6: **Navigate Menu**

Option	Description
Station List	Opens the Station List window.
Next Station	Go to the Next Station.
Previous Station	Go to the Previous Station.
Events List	Opens the Events List window.
Next Event	Go to the Next Event.
Previous Event	Go to the Previous Event.

Station and Event Windows

These windows offer the ability to the user to select which event or station to analyze. Each window displays some additional information for the selection and each header (if clicked) sorts the available events/stations according to the respective parameter. The *font* for each entry works as follows:

1. **Bold:** The event/station exists in the dataset and is within the shear-wave window.
2. Normal: The event/station exists in the dataset, but is not within the shear-wave window.
3. *Strikethrough:* The event/station pair does not exist in the dataset.

Please keep in mind that events in this window are acquired from the catalogue given. If a catalogue does not contain an event that exists in the data directory, this event will not be recognized by Pytheas. The same is true for stations. Only picked stations will be shown. If a QuakeML is not provided or it does not contain S-arrivals, all of the stations included in the provided Station Information file will be shown in the Stations Window (with the font coding still in effect).

If a station or an event is not found, the user will be prompted by the program to skip it and go to the next/previous, until one is found.

Finally, the program uses any angles of incidence obtained from loading the catalogue for the above process. If you select to recalculate angles with TauP, it is possible that the displayed angle is different than the one in the selection window. Also, it is possible that some rays outside the shear-wave window end up in it.

Table 7: **Tools Menu**

Option	Description
AR-AIC	Toggle to turn the AR-AIC auto-picker on or off.
Bandpass Filter	Apply a bandpass filter to the waveforms. A user prompt will ask for the filter boundaries.
Bandpass 1-20 Hz	Apply a bandpass filter in the 1-20 Hz range.
Bandpass 1-10 Hz	Apply a bandpass filter in the 1-10 Hz range.
Recommmend Filter	Apply a filter with recommended frequency bounds defined as 0.5*T_S and 1.5*T_S Hz.
Remove Filter	Remove any filter and display the raw waveforms.
Show Horizontal Spectra	Display the spectrum of each active horizontal component. Requires a window to first be picked.

7.2 Key bindings

The following key bindings are offered in Pytheas.

Table 8: **Visual Inspection Controls**

Description	Key
Increase t_d by 1 sample	Left Arrow
Decrease t_d by 1 sample	Right Arrow
Increase t_d by 5 samples	Shift + Left Arrow
Decrease t_d by 5 samples	Shift + Right Arrow
Adjust amplitude axis	Mouse Wheel
Adjust time axis	Shift + Mouse Wheel
Pick for splitting measurement	Ctrl + Left Mouse
Pick for S arrival	Ctrl + Right Mouse
Set time window start	Shift + Left Mouse
Set time window end	Shift + Right Mouse
Remove time window	Shift + Middle Mouse

Table 9: **Keys**

Description	Key
Open Catalogue	Shift + O
Open Preferences	Shift + P
Save Current Solution to Database	Ctrl + S
Go to Initial Stage	Ctrl + 1
Go to Rotated Stage	Ctrl + 2
Go to Corrected Stage	Ctrl + 3
Set time-delay	Ctrl + D
Set grade	Ctrl + X
Set comment	Ctrl + G
Apply MRC	Ctrl + K
Apply MEV	Ctrl + L
Apply MME	Ctrl + E
Apply ARC	Ctrl + Y
Apply AEV	Ctrl + T
Apply AME	Ctrl + U
Catalogue Cluster Analysis	Shift + A
Station Selection Window	Shift + S
Next Station	Ctrl + W
Previous Station	Ctrl + Q
Event Selection Window	Shift + E
Next Event	Shift + N
Previous Event	Shift + B
Apply Bandpass Filter	Ctrl + B
Apply 1-20 Hz BP Filter	Ctrl + F
Apply 1-10 Hz BP Filter	Ctrl + H
Remove Filter	Ctrl + R

CHAPTER

EIGHT

FREQUENTLY ASKED QUESTIONS (FAQ)

- **The program doesn't start all! I can't see any splash screen or window!** Please check the requirements in the relevant documentation page and make sure you have a suitable version of Python (3.7+) and each dependency. If the problem persists, please send us a description of the problem and the latest generated log file located in *~pytheas/logs/*.
- **The program simply stopped working. Even when pressing any keyboard keysm or using the mouse, nothing happens.** An underlying exception has occured. Please restart the program and send us a description of the problem, as well as the relevant log file.
- **How does Pytheas handles angles of incidence?** If a QuakeML is provided that includes the `takeoff_angle` field in the `Arrival` field, this angle will be used, after using the given velocity model, to estimate the incidence angle. If this condition is not met, the QuakeML has no `Arrival` or a simple text catalogue is provided instead, the angle will be calculated assuming a linear path between the focus and the station (assumed to be at sea-level). This value will be used for selecting events based on the shear-wave window. However, if the respective option is seleced in Preferences, the incidence angle will be calculated from TauP with the given velocity model upon opening the event-station pair. In this case, the TauP value will be used instead.
- **How does the program find stations if no Arrivals are specified in the QuakeML or a text catalogue is given?** In those cases, the station codes will be retrieved from the station information file provided. If a station name is not present in the QuakeML or the station file, but the waveforms exist in the event directory, it will be ignored by the program. Be sure to at least include all stations in the station file!