Computers & Geosciences ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Case Study

APASVO: A free software tool for automatic P-phase picking and event detection in seismic traces☆,☆☆

José Emilio Romero*, Manuel Titos, Ángel Bueno, Isaac Álvarez*, Luz García, Ángel de la Torre, Mª Carmen Benítez

Department of Signal Theory, Telematics and Communications, CITIC-UGR, University of Granada, ETS Ingenierías Informática y de Telecomunicación, Periodista Daniel Saucedo Aranda, s/n, 18071 Granada, Spain

ARTICLE INFO

Article history:
Received 15 May 2015
Received in revised form
4 February 2016
Accepted 5 February 2016

Keywords:
Seismology
Software
Automatic
P-phase picking
Detection
STA-LTA
AMPA
Autoregressive models

ABSTRACT

The accurate estimation of the arrival time of seismic waves or picking is a problem of major interest in seismic research given its relevance in many seismological applications, such as earthquake source location and active seismic tomography. In the last decades, several automatic picking methods have been proposed with the ultimate goal of implementing picking algorithms whose results are comparable to those obtained by manual picking. In order to facilitate the use of these automated methods in the analysis of seismic traces, this paper presents a new free, open source, software graphical tool, named APASVO, which allows picking tasks in an easy and user-friendly way. The tool also provides event detection functionality, where a relatively imprecise estimation of the onset time is sufficient. The application implements the STA-LTA detection algorithm and the AMPA picking algorithm. An autoregressive AIC-based picking method can also be applied. Besides, this graphical tool is complemented with two additional command line tools, an event picking tool and a synthetic earthquake generator. APASVO is a multiplatform tool that works on Windows, Linux and OS X. The application can process data in a large variety of file formats. It is implemented in Python and relies on well-known scientific computing packages such as ObsPy, NumPy, SciPy and Matplotlib.

1. Introduction

The precise determination of the arrival time of seismic waves or picking is a major problem in many seismological applications, such as tomographic inversion of active-source data (Ibánez et al., 2008; García-Yeguas et al., 2012), hypocenter location or earthquake early warning systems (Horiuchi et al., 2005; Nakamura et al., 2009). Traditionally, this task has been performed manually by trained personnel. This manual approach has, however, several drawbacks. The analyst must often cope with the processing of a large amount of seismic data, which makes picking a tedious, time consuming task and therefore prone to human errors (Zhang et al., 2003). In addition, the results depend largely on the experience of the analyst (Küperkoch et al., 2011; Ross and Ben-Zion, 2014), and their quality can be obscured by several factors such as background and non-stationary noise processes from diverse sources such as waves, tides, wind, microtremors or human activity (Havskov and Alguacil, 2004).

Another significant shortcoming of manual picking is the difficulty of dealing with the systematic processing of huge amounts of available seismic data incoming from large-scale monitoring, a quantity that grows as the number of seismometers installed worldwide increments every day. The occurrence of earthquakes grows exponentially as the considered magnitude decreases (Gutenberg and Richter, 1956). Thus, while approximately 150 000 earthquakes occur annually with magnitude sufficient to be perceived by the population (Gutenberg and Richter, 1965), a much larger number of small microearthquakes go unnoticed to people daily but are recorded by monitoring devices. Analysis of this microseismicity would be useful to address the study of the seismicity of a region in a more reliable way taking into account only the earthquakes of a greater magnitude, whose occurrence is much lower. In practice, however, addressing the study of such microseismicity using a manual approach becomes a difficult task due to the aforementioned problems. The volume of data to be examined can be quite large, and the small magnitude of the events makes them difficult to distinguish from background noise, even for an experienced analyst.

☆Source code and binary releases are hosted on APASVO's GitHub repository: https://github.com/cageo/Romero-2016.

☆☆APASVO is also available from Python PyPi repository: https://pypi.python.org/pypi/APASVO.

* Corresponding authors.
E-mail addresses: jemromerol@gmail.com (J.E. Romero), isamaru@ugr.es (I. Álvarez).

http://dx.doi.org/10.1016/j.cageo.2016.02.004
0098-3004/

Please cite this article as: Romero, J.E., et al., APASVO: A free software tool for automatic P-phase picking and event detection in seismic traces. Computers & Geosciences (2016), http://dx.doi.org/10.1016/j.cageo.2016.02.004

Limitations of the manual approach make picking a task whose automation is desirable, and have led to the development of many automatic picking algorithms using a variety of different approaches. We have to separate detection methods, where a relatively imprecise estimation of the onset time is sufficient, and picking algorithms, aimed at providing a precise determination of the phase arrival time (Baer and Kradolfer, 1987). The classical *short term average over long term average* method, or STA-LTA (Allen, 1978, 1982), based on estimating the ratio between the average energy of two windows of different length, has been used extensively as a detection algorithm. The naive version of this method can be easily implemented in an efficient way, yet it may have a poor performance as a picking method (Alvarez et al., 2013). Besides, this method provides a great flexibility, both in its parameter configuration (Trnkoczy, 2002) and accepting a significant number of possible modifications (Withers et al., 1998). Several other more sophisticated picking methods have been proposed, including direct use of an envelope function with a dynamic trigger level (Baer and Kradolfer, 1987), higher-order statistics (Saragiotis et al., 2002; Küperkoch et al., 2010; Nippress et al., 2010), polarization approaches (Montalbetti and Kanasewich, 1970; Kurzon et al., 2014), local-maxima distribution (Panagiotakis et al., 2008), wavelet-based methods (Zhang et al., 2003) or even pattern recognition methods (Tong, 1995) such as neural networks (Dai and MacBeth, 1995; Gentili and Michelini, 2006) or computer vision inspired methods (Joswig, 1990).

Autoregressive models have been successfully applied for picking (Takanami and Kitagawa, 1988, 1991; Leonard and Kennett, 1999; Kitagawa et al., 2001; Kurz et al., 2005). The standard approach, commonly named AR-AIC, is based on estimating autoregressive models for both background noise and incoming seismic phase and then finding the phase arrival time by just looking for the time point that attains the minimum value of the Akaike Information Criterion (AIC) of the locally stationary AR model (Akaike, 1998). Autoregressive methods provide very precise picking results. However, they are computationally very intensive, which often limits their application to selected fragments of the input seismic trace, usually taken on the basis of prior estimate onset picking (Alvarez et al., 2013).

The ability to cope with poor signal to noise ratio conditions is another decisive factor when selecting a picking algorithm. The AMPA algorithm (Adaptive Multi-Band Picking Algorithm) (Alvarez et al., 2013) applies a series of filtering stages to the input signal in order to mitigate as far as possible the background noise as well as enhance the arrival of the seismic phase. The algorithm consists of two steps: Firstly, input signal is processed by using a filter bank in order to estimate the signal envelope and equalize noise for each subband. Secondly, the resulting envelope obtained from the previous step is filtered by using a set of filters, each one of a different length, designed to both enhance the phase arrival and lower emergent or impulsive noises. Results obtained by this algorithm achieve high accuracy even under very adverse signal to noise ratio conditions.

Nowadays, while none of the methods proposed so far has managed to completely replace the job of an expert analyst, they provide a valuable tool to support their work. A number of computer applications have been released for this purpose (Goldstein et al., 1998; Havskov and Ottemöller, 1999; Stockwell, 1999; Lesage, 2009; Abdelwahed, 2012; Olivieri and Clinton, 2012). However, there is still demand for simple, easy to use analysis tools that manage to lower the need for training manuals, specially for users not familiar with complex computer environments. The APASVO software introduced in this paper is intended to ease the use of automatic picking methods using an user friendly graphical interface.

2. Program description

APASVO software tool consists of three different applications, named *apasvo-gui*, *apasvo-detector* and *apasvo-generator*. The application *apasvo-gui* is the graphical interface of APASVO, and allows the user to perform automatic onset detection and picking in an interactive way on one or more files simultaneously. The application implements the STA-LTA detection algorithm (Allen, 1982) and the AMPA picking algorithm (Alvarez et al., 2013). An autoregressive AIC-based picking method (Takanami and Kitagawa, 1988) can be applied either to a user selected signal fragment or in combination with the above algorithms. It is written using PySide and ObsPy. PySide is a python binding of the Qt cross-platform application framework, and its design resembles other digital signal processing and audio processing tools so that users can quickly feel comfortable when they start using the application. ObsPy is a python framework for processing seismological data (Beyreuther et al., 2010). Thus, *apasvo-gui* offers functionality to handle data in many seismological data formats. The application includes manual event relocation. P-wave arrival times can be manually manipulated and even new picks can be added by hand. Optionally S-waves can be manually picked.

apasvo-detector is an application that allows us to use APASVO's picking and detection capacities from command line. Like *apasvo-gui*, *apasvo-detector* allows us to process one or multiple files in a single run. The files can be processed either on batch or supervised mode. On supervised mode the application shows graphically on a separate window each one of the picked onset times and requests the user for feedback asking whether to accept the result shown or not, while on batch mode input files are processed without any interaction from the user about the results that are being generated.

Finally, *apasvo-generator* is a command line application, mainly designed for testing purposes, that generates seismic traces containing a synthetic seismic signal contaminated with background noise. It also allows us to add background noise to an existing seismic trace.

2.1. *apasvo-gui's main functionality*

Fig. 1 shows the main components of *apasvo-gui*'s graphical interface. The application can display different traces of files with different seismic formats simultaneously. One trace from a sac data file and four traces from the same segy data file are opened. The file in sac format is selected. At the top, the application includes several toolbars that allow the use of most of the functionality. The *main toolbar* provides tools for opening or closing files, saving results, doing/undoing actions and configuring settings. The *navigation toolbar* comes with tools for zooming, panning, navigating between used views (e.g. returning to the previous view of the signal after zooming on it) or saving the current view of the signal to an image file. The *analysis toolbar* contains tools for applying the picking algorithms implemented by APASVO, as well as turning on/off the detection threshold and changing its value. It also includes filtering functionality. Finally, the *media toolbar* allows play backing the current signal. It also permits fine adjustment of current selection's boundaries.

Under the toolbar area the *event list* is displayed, showing a table with the data of the events picked so far. Each table row reports arrival time and characteristic function value at that time, plus additional information such as algorithm used, picking mode (i.e. manual or automatic), phase (P or S) confirmation status, an identifying name or label, and additional comments.

The central part of the interface is occupied by the *signal's main view*, which shows the portion of the signal being currently displayed using three different plots that share the same time axis.