Ba*It*: an Iterative Baer picker

—

User Manual

v2.1.6

Matteo Bagagli – ETH, Zurich

April 24, 2019

Contents

1 Introduction

Identifying the arrival time of seismic energy at each offset receiver location has been fundamental to several seismic methods from purely scientific il prospecting since the beginnings of exploration seismology. To remove the burden of manual identification and picking of such transients, automatic picking softwares has been developed consistently throughout the last 50 yr producing increasingly precise results.

In addition, it's safe to say that each seismic dataset can differ strongly from each others in terms of background noises, type of events (i.e. microseismic, teleseismic) and sensor's site effects. Rarely, the usage of a single picking algorithm approach with fixed parameter configuration will provide satisfying results for the entire dataset. For this reason internal and intermediate evaluations steps are suggested to have a more stable tool for seismic phase picking.

The Python package here presented has the aim of overcome *false* detection of seismic phase arrival, and provide an IRIS standard error uncertainty classification of the

measurement. The idea of this simple but efficient tool emerged during the repicking procedure of the GAPSS dataset (M. Bagagli et al. in review) for a local earthquake traveltime tomography study. Because the waveforms were obtained by a temporary seismic network deployed over an exploited geothermal area, the final dataset was showing a lot of antropic and mechanical noises (ie spikes, small tremors) that were deceiveing regular pickers algorithms and therefore missing the true phase-arrival.

The **Ba***lt* code use as main picking algorithm the Baer-Kradolfer (BK) picking algorithm (see Baer and Kradolfer 1987) in a recursive manner throughout the waveform *cuts* provided. Every time a new detection is made, a set of fully-customazible test functions could be run, in order to validate the actual seismic nature of the pick. The strength of this package comes, in fact, from the possibility to fully define *ad-hoc* datasets-dependent function based on user's needs. An additional AIC picker can be optionally run on a closer time window around the validated picks to refine the onset measurament. This additional step will help also to overcome the systematic delay of the BK algoritmh.

Due to its ability in avoiding false detection and custom properties, this Python package is distributed with the hope that could be useful to the seismological community as well in different scenarios. The code can be downloaded from GitHub (provide link) and for any bug reports or open issues, please mail to `matteo.bagagli@erdw.ethz.ch`.

2 Installation

A number of dependecies are requested and are listed in the `requirements.txt`. In order to avoid major issues and libraries conflicts it's strongly recommended to use a *virtual environment*; for this task you could either use conda or pip–virtualenv as your preference. To semplify this passage, a file is provided under the `config` folder (`bait_env.yml`), that you could use to generate the environment by typing:

```
$ cd /where/the/package/is
$ conda env create -f ./config/bait_env.yml
```

Strictly speaking, the installation part is trivial: once the environment has been set-up, all you need to do is to open a terminal and type:

```
$ cd /where/the/package/is
$ pip intall .
```

Once the package is successfully installed, you could verify the software integrity by tiping:

```
$ cd /where/the/package/is
$ pytest
```

After all the test are passed, the software is ready to perform without any problems. In case you would like to discuss specific issues, send few lines with the reported error to `matteo.bagagli@erdw.ethz.ch`

3 The code

3.1 Overview

The **Ba***It* software has been developed in Python–3 and is distributed under the *GPL-v3* license guidelines. The code is written in object-oriented programming and can be easily embedded in already existing codes or routines. The package is composed by 3 different modules:

- `bait.py`: this module contains all the main **Ba***It* class and can be considered the *core* of this package.
- `bait_customtests.py`: this module contains all the test-functions aimed at the pick validation and could be expanded with customizable test by the user (see Sect.4).
- `bait_plots.py`: this module contains the necessary function to plot the waveforms and relative **Ba***It* charachteristic functions and picks.
- `bait_errors.py`: in this modules are contained various type of errors raised by the software. This module can be expanded with additional errors if the user require so (see Sect.4).

The workflow of the picker is schematized in Fig.**??**. At the actual stage the **Ba***It* picking algorithm is able to detect properly the P-first arrival times and could detect sometimes the secondary as well, due to its recursive approach. As a common ground rule, is better to have a pre-processing stage prior the picking one.

The core picking-algorithm adopted by **Ba***It* , is the BK implemented in the ObsPy library. During the picking stage, the user could take advantage of the 2 different BK parameters: the first one (*main*) is used on the first picking attempt only, while the second one (*auxiliary*) is used for all the further steps. In case of a missing *auxiliary* set of picker-parameter, the *main* one will be used also on the recursive step. The picking process is pursued throughout the waveform cut until the BK algorithm isn't able to identify further transients.

3.2 Picks evaluation stage

Once a possible phase-arrival is detected, the pick evaluation stage will come in play to validate the effective seismic nature of the transient. This validation process is driven by the test functions and relative parameters chosen. All the tests are performed over a carachteristic function derived overall the waveform stream given as input (Eq.1). The **Ba***It* package comes with 2 already defined: *SignalAmp* and *SignalSustain*. For the following explanation, Fig.**??** will be used as reference.

$$CF_i = \left| \frac{2 \cdot (wave_i - min(wave))}{max(wave) - min(wave)} - 1 \right| \tag{1}$$

3.2.1 SignalAmp

This simple function will analyze and compare the maximum value of a so called *signal window* defined in seconds after the analyzed pick against a threshold. If the maximum value of the window CF exceed the threshold parameter, the test is passed (Eq.2).

$$max(W_1) \geqslant PAR_1 \tag{2}$$

3.2.2 SignalSustain

This function will analyze the sustained energy of the detected transient. It analyze the mean *signal to noise* ratio between each of the n window slices (same temporal length) after the analyzed pick against a so called *noise window* of a fixed length before the anlyzed pick (Eq.3). The test is passed if the relation explained in the equation is valid for all the window slices.

$$mean(W_n) \geqslant PAR_2 \cdot mean(W_0) \tag{3}$$

4 Customization

One of the interesting feature that this object oriented package provide is the ability to costumize, choose and develop test-functions for the validation process itself. The user can select from a list of already defined functions or create new ones to tackle particular issues related to specific *false* detection of seismic transient.

In order to define a validation test functions, open the `bait_customtests.py` and append a function definition at the end of the file, as:

```
def my_validation_function(wt, par1,par2):
        """ Simple doc-string for usage """
        if par1 > par2:
                return True
        else:
                return False
```

This is an useless and silly function, but it serves as reference. Two main points must be followed when creating new functions: the first parameter should point to an instance of an ObsPy.Trace class that contains the necessary waveforms, and the returning value should be booolean True or False in case the test is passed or not respectively.

References

Baer, M. and U. Kradolfer (Aug. 1, 1987). "An automatic phase picker for local and teleseismic events". In: *Bulletin of the Seismological Society of America* 77.4, pp. 1437–1445. ISSN: 0037-1106. URL: `https://pubs.geoscienceworld.org/ssa/bssa/article-abstract/77/4/1437/119016/an-automatic-phase-picker-for-local-and` (visited on 04/18/2019).