PyMPA Documentation

Release 1.0.0

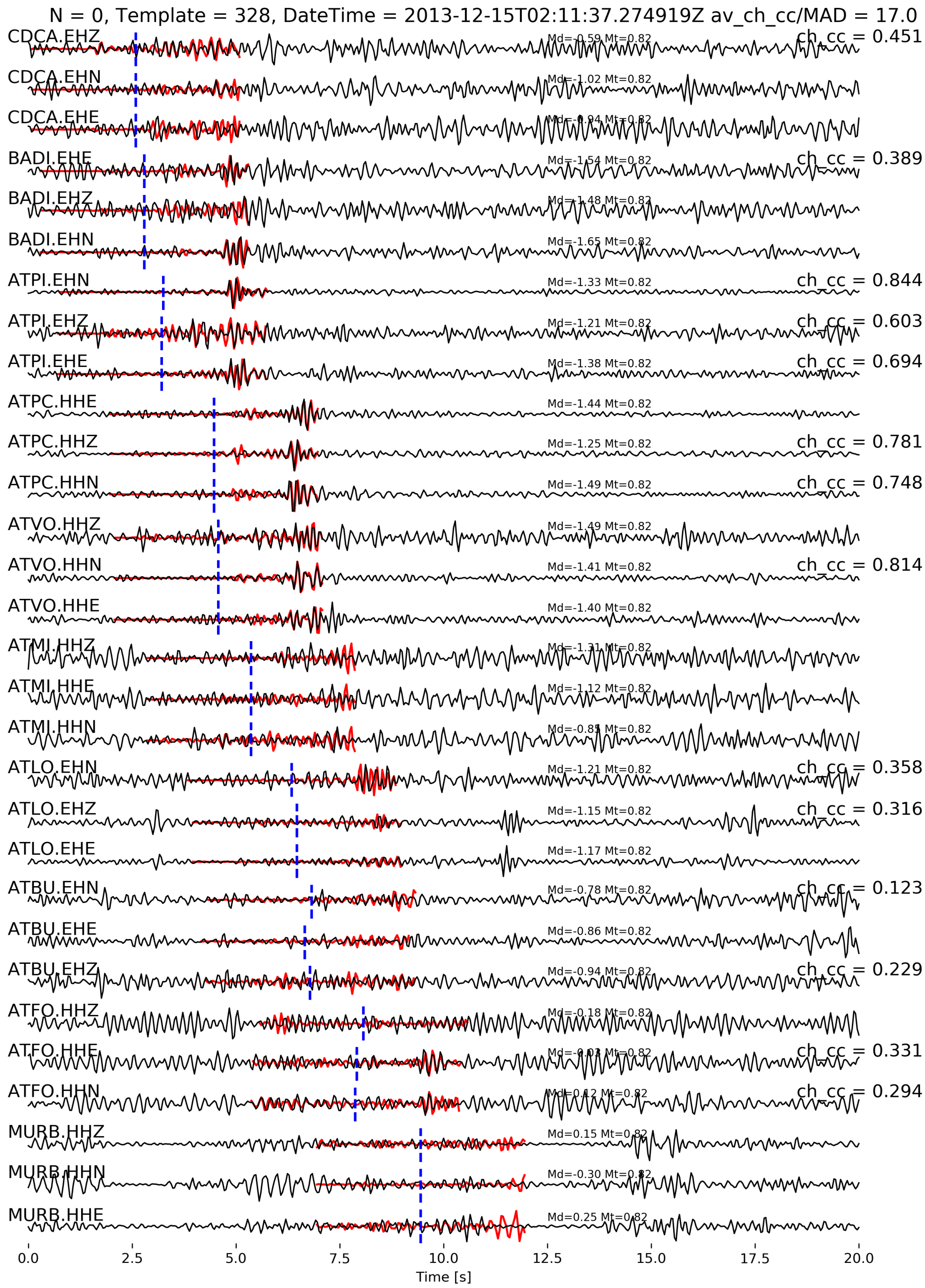
Alessandro Vuan

Apr 09, 2019

CONTENTS

A Python package for the detection of seismicity based on templates. PyMPA contains an efficient code for the detection of microseismicity starting from well located templates. PyMPA is an open source seismological software and consists of some separate utilities for input preparation, the main program, and output post-processing tools to obtain a catalog and verify events. PyMPA is designed to detect microseismicity from the cross-correlation of continuous data and templates.

The code is stored on , and is free to be cloned on your platform. It supports Python 2.7, 3.4, 3.5, 3.6, 3.7 releases and uses for reading and writing seismic data, and for handling most of the event metadata. Matched-filter correlations are calculated using ObsPy v. 1.2.0 released on April 2019. Important: we recommend to use an updated version of ObsPy.



Example of detection using PyMPA. Templates (red waveforms) overlapped on continuous data (black) filtered from 3 to 8 Hz are shown. On the left for the used channels the corresponding cross-correlation value.

This package contains:

- *Routines for downloading data from eida servers*;
- *Routines for creating and trimming templates*;
- *Routines for calculating moveout time for synchronization*;

- *Kurtosis based template verification*;
- *Template matching by using daily estimation of MAD and all the available channels*;
- *Postprocessing routines*;
- *Visual verification of detections*;

This package is written by the PyMPA developers, and is distributed under the LGPL GNU Licence, Copyright PyMPA developers 2019.

CHAPTER

ONE

ACKNOWLEDGEMENTS

The software development was partially funded by a joint research project within the executive program of scientific and technological cooperation between Italy and Japan for the period 2013–2015. Additional funds for software development come from the project "Seismology and Earthquake Engineering Research Infrastructure Alliance for Europe" (SERA), responding to the priorities identified in the call INFRAIA-01-2016-2017 Research Infrastructure for Earthquake Hazard. We thank Monica Sugan for the extensive testing of the codes and Aitaro Kato at the Earthquake Research Institute (ERI) in Tokyo for fruitful discussions. The authors also wish to thank the ObsPy community for the continuous support and constant development of related libraries.

CHAPTER

TWO

CITATION

If you use this package in your work, please cite the following papers:

Vuan A., Sugan M., Amati G., Kato A., 2017 - Improving the Detection of Low-Magnitude Seismicity Preceding the Mw 6.3 L'Aquila Earthquake: Development of a Scalable Code Based on the Cross-Correlation of Template Earthquakes, BSSA https://pubs.geoscienceworld.org/ssa/bssa/article-abstract/525813/improving-the-detection-of-low-magnitude?redirectedFrom=fulltext

Vuan A., Sugan M., Chiaraluce L., Di Stefano R., 2017 - Loading rate variations along a mid-crustal shear zone preceding the MW6.0 earthquake of the 24th of August 2016 in Central Italy, Geophysical Research Letters http://onlinelibrary.wiley.com/doi/10.1002/2017GL076223/full

Sugan, M., Vuan, A., Kato, A., Massa, M., & Amati, G. (2019). Seismic evidence of an early afterslip during the 2012 sequence in Emilia (Italy). Geophysical Research Letters, 46, 625–635. https://doi.org/10.1029/2018GL079617

CHAPTER

THREE

CONTENTS:

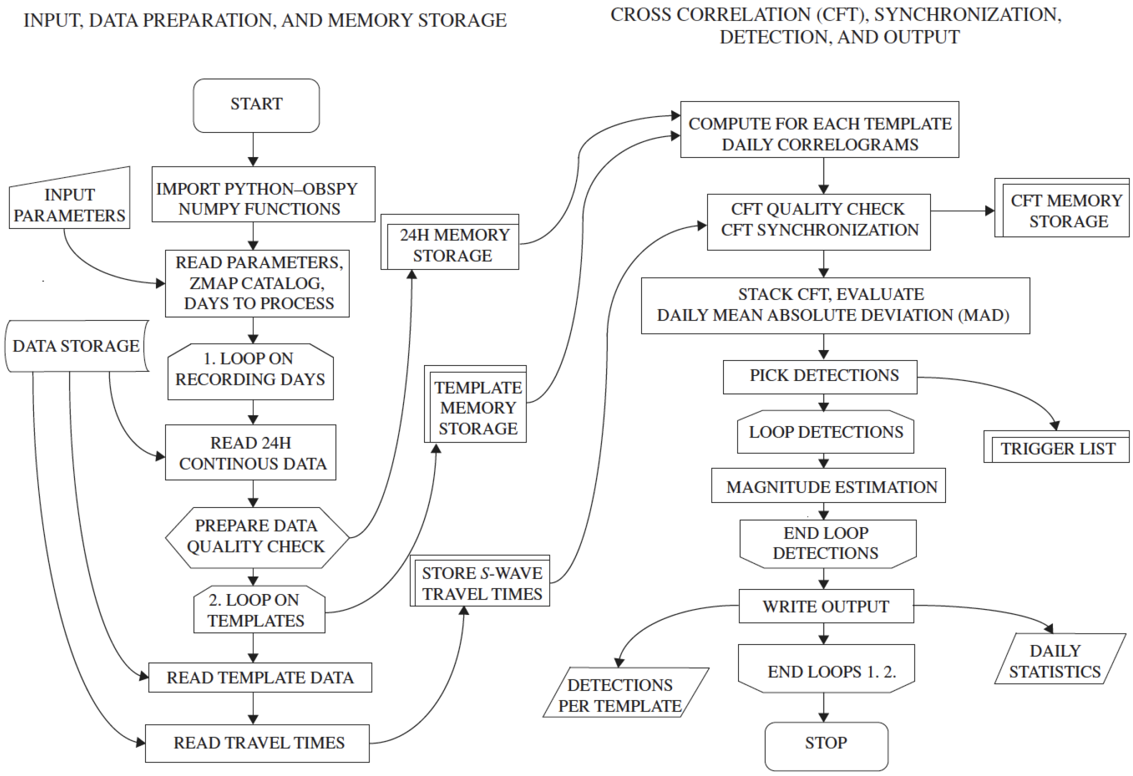
3.1 Introduction to the PyMPA package

This document is designed to give you an overview of the capabilities and implementation of the PyMPA Python package.

3.1.1 Motivation

PyMPA is designed to augment the detection capability of earthquakes, or any seismic signal (explosions or low frequency tremors) by using advanced routines different from the faster standard amplitude-ratio STA/LTA methods.

The technique allows to improve seismic catalogs decreasing the completeness magnitude and is particularly useful in detecting seismicity below the background noise level, and during a strong aftershocks sequence when the network sensitivity is lower. PyMPA is based on MFT search for earthquakes that resemble well-located events, termed templates (e.g., Shelly et al., 2007; Peng and Zhao, 2009; Yang et al., 2009; Kato et al., 2012; Zhang and Wen, 2015). The algorithm, which exploits ObsPy routines (Krischer et al., 2015), is versatile and supports most commonly used seismic data and earthquake catalog formats. A PyMPA flowchart is also shown below.



In addition to PyMPA, we develop other tools external to the main code to manage the input–output preparation and validation for (1) downloading data from Observatories and Research Facilities for European Seismology–European Integrated Data Archive (ORFEUS-EIDA) servers, (2) evaluating data quality, (3) selecting earthquakes as templates from a reference catalog, (4) trimming and filtering them from continuous waveforms, (5) avoiding redundant detections in the output, and (6) validating new events. This repository will continue to grow and develop and any modification will be reported in the github repository.

3.1.2 Supported environments

Linux, OSX and Windows environments running Python 2.7 and 3.x. We will stop support for Python 2.7 in a forthcoming release.

3.1.3 Functionality

Within *input* you will find the routines to generate templates, (*create_template*) select good templates (*template_check*), calculate travel times (*calculate_ttimes*), compute cross-channel correlations from these templates (*pympa*), process_detections (*process_detections*), and apply for visual inspection (*verify_detection*)

3.1.4 Running tests

For running tests examples are provided in the github subdirectories, tests are recalled when modifications are performed to the codes and a TRAVIS CI report is released.

You can also run these tests by yourself locally to ensure that everything runs as you would expect in your environment.

Although every effort has been made to ensure these tests run smoothly on all supported environments , if you do find any issues, please let us know on the page.

3.1.5 References

Shelly, D. R., G. C. Beroza, and S. Ide (2007). Non-volcanic tremor and low frequency earthquake swarms, Nature 446, 305–307.

Peng, Z., and P. Zhao (2009). Migration of early aftershocks following the 2004 Parkfield earthquake, Nature Geosci. 2, 877–881.

Yang, H., L. Zhu, and R. Chu (2009). Fault-plane determination of the 18 April 2008 Mount Carmel, Illinois, earthquake by detecting and relocating aftershocks, Bull. Seismol. Soc. Am. 99, 3413–3420.

Kato, A., K. Obara, T. Igarashi, H. Tsuruoka, S. Nakagawa, and N. Hirata (2012). Propagation of slow slip leading up to the 2011 Mw 9.0 Tohoku-Oki earthquake, Science 335, 705–708.

Zhang, M., and L. Wen (2015). An effective method for small event detection: Match and locate (M&L), Geophys. J. Int. 200, 1523–1537.

Krischer, L., T. Megies, R. Barsch, M. Beyreuther, T. Lecocq, C. Caudron, and J. Wassermann (2015). ObsPy: A bridge for seismology into the scientific Python ecosystem, Comput. Sci. Discov. 8, no. 1, 014003, doi: 10.1088/1749-4699/8/1/014003.

3.2 PyMPA installation

PyMPA is a pure Python package. It runs after the installation of a virtual environment with Numpy, Scipy, Matplotlib and Obspy libraries. Some C extensions of ObsPy toolkit are also used and Bottleneck libraries. Bottleneck is a set of functions inspired by NumPy and SciPy, but written in Cython with high performance in mind. Bottleneck provides separate Cython functions for each combination of array dimensions, axis, and data type.

We heavily recommend installing ObsPy using conda because:

- separate your install from your system default Python, avoiding to have problems with your OS;
- correct compilation is more probable

If you do not have either a miniconda or anaconda installation you can follow the instructions.

If you do not already have a conda environment we recommend creating one with the following:

```
conda create -n obspy python=3.6
conda activate obspy
conda install obspy
conda install bottleneck
```

For installing PyMPA you can simply clone the git repository and try it within your obspy env:

```
git clone git@github.com:avuan/PyMPA37.git
```

On a Linux system for installing conda, obspy, bottleneck and mirror the PyMPA code follow this commands:

```
curl -O https://repo.continuum.io/archive/Anaconda3-5.0.1-Linux-x86_64.sh
chmod +x Anaconda3-5.0.1-Linux-x86_64.sh
./Anaconda3-5.0.1-Linux-x86_64.sh
source ~.bashrc
conda config --add channels conda-forge
conda create -n obspy37 python=3.7
source activate obspy37
conda install obspy
conda install bottleneck
mkdir test_obspy1.2.0
cd test_obspy1.2.0
git clone https://github.com/avuan/PyMPA37
```

3.3 Tutorial

This tutorial is designed to give you an overview of the capabilities and implementation of the PyMPA Python package.

3.3.1 Downloading Seismological Data

To download seismological data from EIDA (European Integrated Data Archive) servers: Data from broad band seismic stations are available from many European Institutions. To download seismological data from EIDA (European Integrated Data Archive) servers and inventory data in STATIONXML format many examples can also be found in ObsPy.

PyMPA requires, continuous data and stations inventories. EIDA servers can easily release data from permanent networks and the corresponding inventories. The examples in the subdirectory input.download_data.dir show the python scripts that allows the download.

In the case your data come from other sources, PyMPA through ObsPy libraries is able to manage most of the seismological data formats (MSEED, SAC, SEISAN, SEGY, etc..). An inventory data file including station information needs to be created by modifying an existing StationXML file.

PyMPA does not use databases and prefers to store single channel daily continuous data in archieves or subdirectories.

Executable files:

- download_data.py (https://github.com/avuan/PyMPA37/tree/master/input.download_data.dir/download_data.py)
- download_inventory.py (https://github.com/avuan/PyMPA37/tree/master/input.download_data.dir/download_inventory.py)

(*download_data*) download data

3.3.2 Create Templates

(*create_template*) create templates

A Python script create_templates.py is used to trim templates from continuous data and inventories stored in an archive. Generally, we use travel times to cut events before and after S-wave arrivals. Thus, a reference 1D velocity model is needed. Trimmed waveforms have to be carefully checked to evaluate the effectiveness of S-wave travel time calculations. Take care that a high sampling rate could result in memory consumption and prolonged times of execution. Input data should be decimated a priori accordingly with your needs and availability of cores. Check the example for running create_templates.py at https://github.com/avuan/PyMPA37/tree/master/input.create_templates.dir

Executable file:

- create_templates.py (https://github.com/avuan/PyMPA37/tree/master/input.create_templates.dir/create_templates.py)

Input parameters:

```
#Line 1 -- list of stations
#Line 2 -- list of channels
#Line 3 -- list of networks
#Line 4 -- Lowpass frequency
#Line 5 -- Highpass frequency
```

(continues on next page)

(continued from previous page)

```
#Line 6 -- Trimmed Time before S-wave
#Line 7 -- Trimmed Time after S-wave
#Line 8 -- UTC precision
#Line 9 -- Continuous data dir
#Line 10 -- Template data dir
#Line 11 -- Processing days list
#Line 12 -- Zmap catalog
#Line 13 -- Starting template
#Line 14 -- Stopping template
#Line 15 -- Taup Model
AQU CAMP CERT FAGN FIAM GUAR INTR MNS NRCA TERO
BHE BHN BHZ
IV MN
2.0
8.0
2.5
2.5
6
24h
template
lista1
templates.zmap
26
27
aquila_kato
```

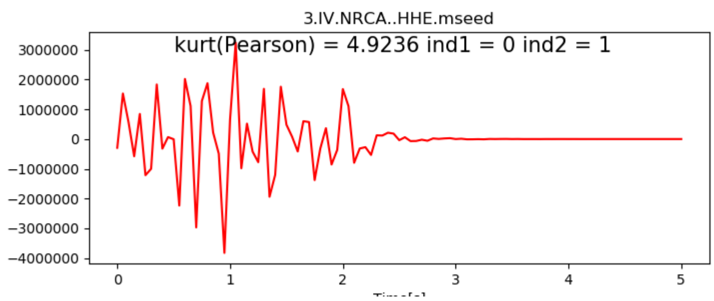
Note that input and output file names, inventories, template catalogs, velocity models are recalled also in the next steps.

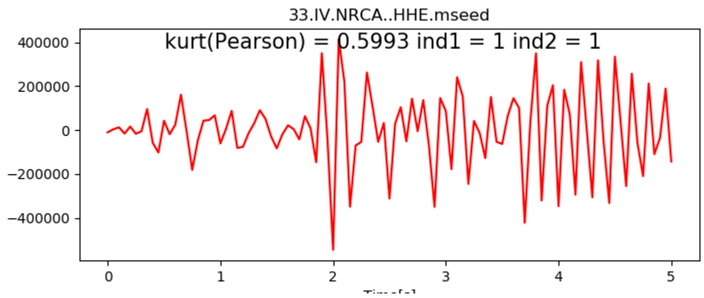
3.3.3 Check Template Quality

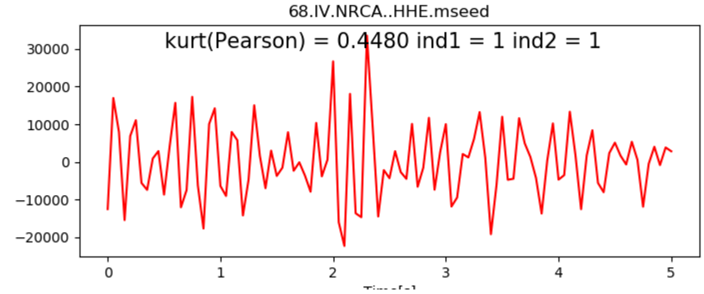
(*template_check*) select good templates

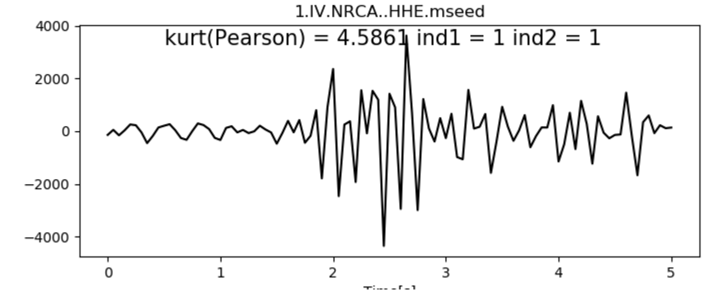
Evaluating template quality allows to input only a good signal to noise ratio avoiding artifacts resulting in unwanted detections. The selection is based on Kurtosis method (https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kurtosis.html) supposing that the waveform is simmetrically trimmed at the first S-wave arrival. Kurtosis evaluates if the time distrbution of amplitudes is simmetric or not excluding data having a low signal to noise ratio. Peak amplitudes at the beginning or in the signal coda are unwanted and a selection is also made to exclude them.

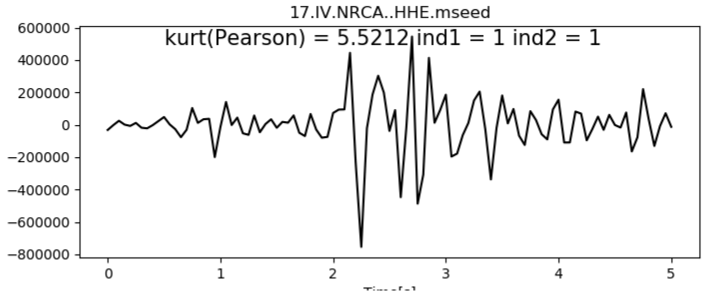
Check examples running test_kurtosis1.py at https://github.com/avuan/PyMPA37/tree/master/input.template_check.dir After running kurtosis based selection, templates are separated in two subdirectories "bad"(red waveforms see figure below) and "good" (black waveforms see figure below)

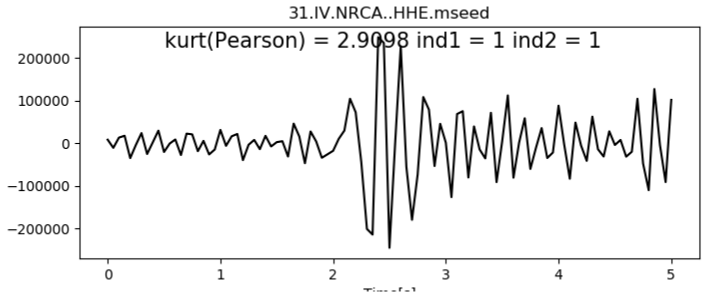












Executable python scripts:

- template_check1.py at https://github.com/avuan/PyMPA37/tree/master/input.template_check.dir/template_check1.py (performs on waveform)
- template_check2.py at https://github.com/avuan/PyMPA37/tree/master/input.template_check.dir/template_check2.py (performs on the absolute values of waveform)

3.3.4 Calculate Travel Times

(*calculate_ttimes*) calculate travel times

Travel time calculation is based on Java TauP Toolkit as implemented in ObsPy (https://docs.obspy.org/packages/obspy.taup.html) Travel times are needed for synchronization to obtain a stacked cross-correlation function. It is supposed that trimmed templates are stored in ./template directory. The same reference 1D velocity model used for trimming templates is needed.

Executable file:

- calculate_ttimes.py at https://github.com/avuan/PyMPA37/tree/master/input.calculate_ttimes.dir/calculate_ttimes.py

Input parameters:

```
#Line 1 -- list of stations
#Line 2 -- list of channels
#Line 3 -- list of networks
#Line 4 -- Lowpass frequency
#Line 5 -- Highpass frequency
#Line 6 -- Trimmed Time before S-wave
#Line 7 -- Trimmed Time after S-wave
#Line 8 -- UTC precision
#Line 9 -- Continuous data dir
#Line 10 -- Template data dir
#Line 11 -- Processing days list
#Line 12 -- Zmap catalog
#Line 13 -- Starting template
#Line 14 -- Stopping template
#Line 15 -- Taup Model
AQU CAMP CERT FAGN FIAM GUAR INTR MNS NRCA TERO
BHE BHN BHZ
IV MN
2.0
8.0
2.5
2.5
6
template
ttimes1
lista1
templates.zmap
26
27
aquila_kato
```

Note that input and output file names, inventories, template catalogs, velocity models are recalled also in the next steps.

3.3.5 Running PyMPA

Template matching code, using cross-correlation based on well located events. The code is embarassingly parallel and different templates/days can be run on different cores. We do not provide the scripts to parallelize jobs preferring to leave to the user to find the best strategy to accomplish the task. We generally prefer to distribute the workload by using Slurm.

Executable files:

- pympa.py (working on daily chunks and with a reduced number of channels). Chunking daily data results in MAD calculated on the working time window.

Input parameters:

Input parameters are described line by line in parameters24 file

```
# input parameters for runnig 27 version
# line1 = stations available,
# line2 = channels available,
# line3 = networks available,
# line4 = low bandpass filter frequency,
# line5 = high bandpass filter frequency,
# line6 = sample tolerance in detecting maximum cft amplitude for each channel,
# line7 = cross-correlation threshold to be overcome by cft,
# line8 = min number of channels overcoming the cross-correlation threshold,
# line9 = template duration(s),
# line10 = UTCDateTime.DEFAULT_PRECISION (number of digits considered in fractions␣
↪of seconds),
# line11 = string variable containing the directory name for continuous 24h␣
↪waveforms,
# line12 = string variable for templates' directory,
# line13 = string variable for travel times directory,
# line14 = filename for day list to process,
# line15 = filename for zmap catalog,
# line16 = template start number,
# line17 = template stop number(if line16==0  and line17==0 all templates are␣
↪processed,
# line18 = multiplying factor for MAD to determine threshold
# line 19 = multiplying factor to remove daily cft channels with std greater than␣
↪(average std from all channels * factor at at line 19)
# line 20 = multiplying factor to remove daily cft channels with std smaller than␣
↪(average std from all channels * factor at at line 20)
# line 21 = maximum number of templates to be used in template matching (choice is␣
↪made preferring the closest channels)
# line 22 = number of chunks per day (1=86400s, 2=43200, 3=28800, 4=21600, 6=14400␣
↪etc... icreasing the factor allows reducing memory consumption)
APEC ATBU ATCA ATCC ATFO ATLO ATPC ATPI ATSC ATVO BADI FOSV FRON MURB NARO PARC␣
↪PIEI PE3 SSFR
EHE EHN EHZ HHE HHN HHZ
IV
3.0
8.0
6
0.35
6
5
6
24h
template
ttimes
lista1
templates.zmap
200
203
8
1.5
0.25
12
6
```

3.3.6 Output Processing

(*output.process_detections*) controls multiple detections in short time windows

A bash script calling python code performs the catalog sythesis. Some templates could concur to the same detection. The detection showing the highest threshold value is preferred in a fix time window (e.g. 6 seconds).

Executable file:

- bash script postproc37.sh
- process_detections.py at https://github.com/avuan/PyMPA37/tree/master/output.process_detections.dir/process_detections.py

Input parameters:

```
# - Line 1 - FlagDateTime (Enter 1 for timestamp format (2009-03-30T21:59:43.
↪616111Z) or Enter 0 for having seconds in a day (for plotting))
# - Line 2 - Flag_cc (1 to have returned the best results in terms of average␣
↪cross-corr using the sample tolerance, 0 results at sample tolerance 0)
# - Line 3 - Input file
# - Line 4 - UTC precison used in time
# - Line 5 - window_length (half time window for searching unique events)
# - Line 6 - min_threshold (threshold used in parameters24 for detecting events, ␣
↪threshold * MAD)
# - Line 7 - min_nch (minimum number of channels to overcome the setup cc value in␣
↪parameters24 e.g. 0.6)
1
1
./dcat
6
3.0
8.0
7
```

3.3.7 Verify Detections

(*output.verify_detection*) for visual verification of events

Produce graphics windows showing the continuous data overlapped by templates events at the detection time.

Executable file:

- verify_detection.py at https://github.com/avuan/PyMPA37/tree/master/output.verify_detection.dir/verify_detection.py

Input parameters:

```
# Line 1 -- list of stations
# Line 2 -- list of channels
# Line 3 -- list of networks
# Line 4 -- Lowpass frequency
# Line 5 -- Highpass frequency
# Line 6 -- Trimmed Time before S-wave
# Line 7 -- Trimmed Time after S-wave
# Line 8 -- UTC precision
# Line 9 -- Continuous data dir
# Line 10 -- Template data dir
# Line 11 -- Ttimes data dir
# Line 12 -- Zmap catalog
# Line 13 -- Starting detection
# Line 14 -- Stopping detection
# Line 15 -- Half of the visualized time window in sec
# Line 16 -- Taup Model
# Line 17 -- Flag_Save_Fig (0 = show, 1 = save figure)
# Line 18 -- Flag_Read_Stats (0 = no stats, 1 = read stats)
# Line 19 -- Tolerance in time between outcat origin time and stats
ATBU ATFO ATLO ATMI ATPC ATPI ATSC ATVO BADI CDCA MURB
EHE EHN EHZ HHE HHN HHZ
IV MN
```

(continues on next page)

(continued from previous page)

```
3.0
8.0
2.5
2.5
6
24h
template
ttimes
templates.zmap
0
5
10
ato
1
1
0.05
```

3.3.8 References

Shelly, D. R., G. C. Beroza, and S. Ide (2007). Non-volcanic tremor and low frequency earthquake swarms, Nature 446, 305–307.

Peng, Z., and P. Zhao (2009). Migration of early aftershocks following the 2004 Parkfield earthquake, Nature Geosci. 2, 877–881.

Yang, H., L. Zhu, and R. Chu (2009). Fault-plane determination of the 18 April 2008 Mount Carmel, Illinois, earthquake by detecting and relocating aftershocks, Bull. Seismol. Soc. Am. 99, 3413–3420.

Kato, A., K. Obara, T. Igarashi, H. Tsuruoka, S. Nakagawa, and N. Hirata (2012). Propagation of slow slip leading up to the 2011 Mw 9.0 Tohoku-Oki earthquake, Science 335, 705–708.

Zhang, M., and L. Wen (2015). An effective method for small event detection: Match and locate (M&L), Geophys. J. Int. 200, 1523–1537.

Krischer, L., T. Megies, R. Barsch, M. Beyreuther, T. Lecocq, C. Caudron, and J. Wassermann (2015). ObsPy: A bridge for seismology into the scientific Python ecosystem, Comput. Sci. Discov. 8, no. 1, 014003, doi: 10.1088/1749-4699/8/1/014003.

3.4 Preprocessing Input

This package contains utilities for:

- *Routines for downloading data from eida servers*;
- *Routines for creating and trimming templates*;
- *Routines for calculating moveout time for synchronization*;
- *Kurtosis based template verification*;

These utilities are written by the PyMPA developers, and are distributed under the LGPL GNU Licence, Copyright PyMPA developers 2019.

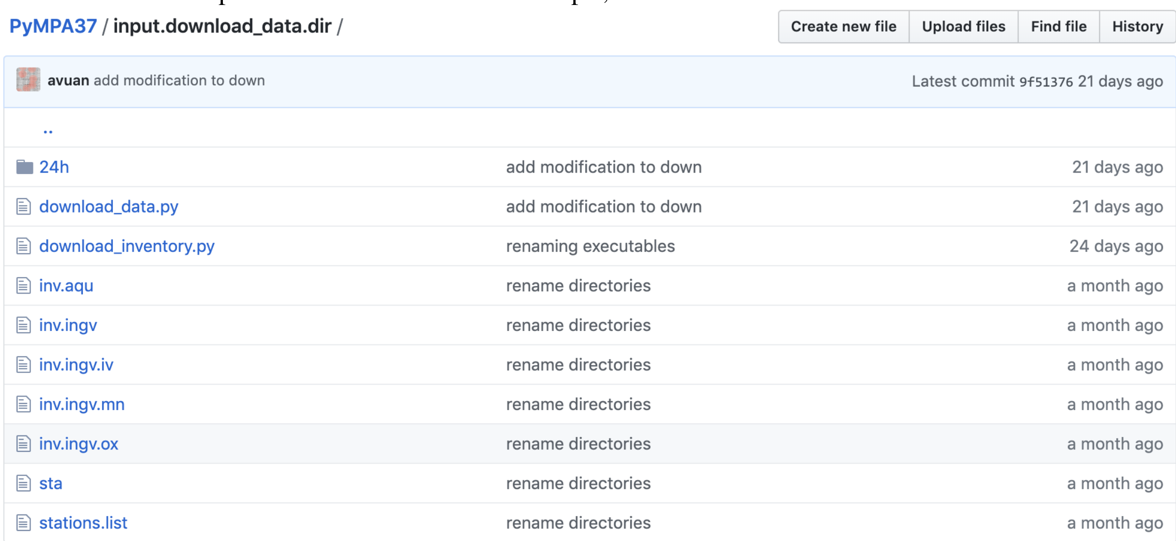
3.4.1 Contents:

Download data

Downloading data from EIDA servers is easily performed by using ObsPy tools and routines. Please see the scripts at https://github.com/avuan/PyMPA37/blob/master/input.download_data.dir/

download_data.py and download_inventory.py allows downloading continuous data and inventories for the selected stations and time period.

For some networks or stations please check before data availability or if there are some restrictions and a token or userid is needed. Some EIDA servers when receiving from the same user many requests in a short time close the door to the user. That is the reason why we introduced a pause command in our scripts, this avoids the shutdown of the connection.



Create templates

Actually templates are created by trimming a fixed time window focused on S-wave theoretical travel times. The length of the time window is established by the inter-statiion network distance and the frequency range used. The user should carefully check to exclude signal deriving from numerical artifacts (e.g. filtering applied to zero padding time windows having no data), or pre and coda signals not connected with the seismic perturbation investigated (e.g. LFEs, earthquakes, icequakes etc. . .) In the next versions the trimming will allow for selecting variable length P and S-waves.

Needed files:

- Events in a catalog: e.g. templates.zmap (quakeml or zmap format) see ObsPy for format

```
13.38347    42.38842    2009    03    30    1.3    11.180    14    04    34.50
13.38981    42.33004    2009    03    30    1.5    9.170     14    15    4.32
13.39447    42.32878    2009    03    30    1.6    9.610     14    16    6.35
13.40577    42.31941    2009    03    30    1.4    5.370     14    17    1.74
13.36791    42.59639    2009    03    30    1.3    5.710     14    35    16.74
13.45651    42.25600    2009    03    30    1.3    5.620     14    42    23.29
13.38686    42.31381    2009    03    30    1.2    6.270     15    10    34.73
13.49527    42.30006    2009    03    30    0.9    6.810     15    21    2.28
```

```
13.32714          42.33182          2009     03       30       1.4      6.210    15␣
↪       40       5.62
13.29606          42.27235          2009     03       30       1.2      8.690    15␣
↪       53       5.35
13.39056          42.31176          2009     03       30       1.0      6.630    15␣
↪       57       21.38
13.33732          42.32800          2009     03       30       1.6      11.580   16␣
↪       06       59.81
13.37035          42.31771          2009     03       30       1.4      10.570   16␣
↪       28       12.31
13.39649          42.31908          2009     03       30       1.4      8.140    17␣
↪       04       31.64
13.33823          42.33349          2009     03       30       1.3      7.820    17␣
↪       15       58.33
13.41437          42.34201          2009     03       30       1.5      7.120    17␣
↪       47       19.88
13.38869          42.30452          2009     03       30       1.0      7.820    18␣
↪       08       28.49
13.39272          42.32332          2009     03       30       1.3      8.670    19␣
↪       27       21.43
13.44121          42.32962          2009     03       30       1.2      2.700    19␣
↪       52       48.49
13.39902          42.34200          2009     03       30       1.2      9.790    20␣
↪       23       36.86
13.37897          42.33234          2009     03       30       1.0      8.200    21␣
↪       35       55.98
13.42597          42.46570          2009     03       30       1.3      19.820   21␣
↪       37       24.47
13.38473          42.31577          2009     03       30       1.2      6.540    22␣
↪       08       12.70
13.38779          42.30452          2009     03       30       1.1      6.690    22␣
↪       18       38.95
13.38168          42.31454          2009     03       30       1.0      7.290    22␣
↪       52       12.88
13.25361          42.27312          2009     03       30       1.0      14.930   23␣
↪       21       59.35
13.38767          42.30401          2009     03       30       1.1      6.470    23␣
↪       55       40.01
```

- Suitable velocity model for computing travel times

```
aquila + ak135
depth P vel. S vel. density older density
     0.000      3.7500      2.1650      2.4500
     1.500      3.7500      2.1710      2.4500
     1.510      4.9400      2.8520      2.7800
     4.510      4.9400      2.8580      2.7800
     4.520      6.0100      3.2790      2.7600
    14.520      6.0100      3.2850      2.7600
    14.530      5.5500      3.3950      2.9100
    29.530      5.5500      3.4010      2.9100
    29.540      5.8800      4.0990      3.1000
    43.540      5.8800      4.1050      3.1000
    43.550      5.8800      4.5610      3.1000
    57.500      5.8800      3.3600      3.1000
    57.500      7.1100      4.0100      3.2300
    93.000      7.1100      4.0100      3.2300
    93.000      7.1000      3.9900      3.3000
```

(continues on next page)

(continued from previous page)

```
 136.500    7.1000    3.9900    3.3000
 165.000    8.1750    4.5090    3.3487
 210.000    8.3000    4.5180    3.3960
 210.000    8.3000    4.5230    3.3960
 260.000    8.4825    4.6090    3.4652
 310.000    8.6650    4.6960    3.5343
 360.000    8.8475    4.7830    3.6034
 410.000    9.0300    4.8700    3.6726
 410.000    9.3600    5.0800    3.7976
 460.000    9.5280    5.1860    3.8612
 510.000    9.6960    5.2920    3.9248
 560.000    9.8640    5.3980    3.9885
 610.000   10.0320    5.5040    4.0521
 660.000   10.2000    5.6100    4.1158
 660.000   10.7900    5.9600    4.3393
 710.000   10.9229    6.0897    4.3896
 760.000   11.0558    6.2095    4.4399
 809.500   11.1353    6.2426    4.4701
 859.000   11.2221    6.2798    4.5029
 908.500   11.3068    6.3160    4.5350
 958.000   11.3896    6.3512    4.5664
1007.500   11.4705    6.3854    4.5970
1057.000   11.5495    6.4187    4.6270
1106.500   11.6269    6.4510    4.6563
1156.000   11.7026    6.4828    4.6849
1205.500   11.7766    6.5138    4.7130
1255.000   11.8491    6.5439    4.7404
1304.500   11.9200    6.5727    4.7673
1354.000   11.9895    6.6008    4.7936
1403.500   12.0577    6.6285    4.8195
1453.000   12.1245    6.6555    4.8448
1502.500   12.1912    6.6815    4.8700
1552.000   12.2550    6.7073    4.8942
1601.500   12.3185    6.7326    4.9182
1651.000   12.3819    6.7573    4.9423
1700.500   12.4426    6.7815    4.9653
1750.000   12.5031    6.8052    4.9882
1799.500   12.5631    6.8286    5.0109
1849.000   12.6221    6.8515    5.0333
1898.500   12.6804    6.8742    5.0553
1948.000   12.7382    6.8972    5.0772
1997.500   12.7956    6.9194    5.0990
2047.000   12.8526    6.9418    5.1206
2096.500   12.9096    6.9627    5.1422
2146.000   12.9668    6.9855    5.1638
2195.500   13.0222    7.0063    5.1848
2245.000   13.0783    7.0281    5.2061
2294.500   13.1336    7.0500    5.2270
2344.000   13.1894    7.0720    5.2481
2393.500   13.2465    7.0931    5.2698
2443.000   13.3018    7.1144    5.2907
2492.500   13.3585    7.1369    5.3122
2542.000   13.4156    7.1586    5.3338
2591.500   13.4741    7.1807    5.3560
2640.000   13.5312    7.2031    5.3776
2690.000   13.5900    7.2258    5.3999
2740.000   13.6494    7.2490    5.4224
2740.000   13.6494    7.2490    5.4224
2789.670   13.6530    7.2597    5.4238
2839.330   13.6566    7.2704    5.4250
2891.500   13.6602    7.2811    5.4265
```

(continues on next page)

(continued from previous page)

```
2891.500      8.0000      0.0000      9.9145
2939.330      8.0382      0.0000      9.9942
2989.660      8.1283      0.0000     10.0722
3039.990      8.2213      0.0000     10.1485
3090.320      8.3122      0.0000     10.2233
3140.660      8.4001      0.0000     10.2964
3190.990      8.4861      0.0000     10.3679
3241.320      8.5692      0.0000     10.4378
3291.650      8.6496      0.0000     10.5062
3341.980      8.7283      0.0000     10.5731
3392.310      8.8036      0.0000     10.6385
3442.640      8.8761      0.0000     10.7023
3492.970      8.9461      0.0000     10.7647
3543.300      9.0138      0.0000     10.8257
3593.640      9.0792      0.0000     10.8852
3643.970      9.1426      0.0000     10.9434
3694.300      9.2042      0.0000     11.0001
3744.630      9.2634      0.0000     11.0555
3794.960      9.3205      0.0000     11.1095
3845.290      9.3760      0.0000     11.1623
3895.620      9.4297      0.0000     11.2137
3945.950      9.4814      0.0000     11.2639
3996.280      9.5306      0.0000     11.3127
4046.620      9.5777      0.0000     11.3604
4096.950      9.6232      0.0000     11.4069
4147.280      9.6673      0.0000     11.4521
4197.610      9.7100      0.0000     11.4962
4247.940      9.7513      0.0000     11.5391
4298.270      9.7914      0.0000     11.5809
4348.600      9.8304      0.0000     11.6216
4398.930      9.8682      0.0000     11.6612
4449.260      9.9051      0.0000     11.6998
4499.600      9.9410      0.0000     11.7373
4549.930      9.9761      0.0000     11.7737
4600.260     10.0103      0.0000     11.8092
4650.590     10.0439      0.0000     11.8437
4700.920     10.0768      0.0000     11.8772
4801.580     10.1415      0.0000     11.9414
4851.910     10.1739      0.0000     11.9722
4902.240     10.2049      0.0000     12.0001
4952.580     10.2329      0.0000     12.0311
5002.910     10.2565      0.0000     12.0593
5053.240     10.2745      0.0000     12.0867
5103.570     10.2854      0.0000     12.1133
5153.500     10.2890      0.0000     12.1391
5153.500     11.0427      3.5043     12.7037
5204.610     11.0585      3.5187     12.7289
5255.320     11.0718      3.5314     12.7530
5306.040     11.0850      3.5435     12.7760
5356.750     11.0983      3.5551     12.7980
5407.460     11.1166      3.5661     12.8188
5458.170     11.1316      3.5765     12.8387
5508.890     11.1457      3.5864     12.8574
5559.600     11.1590      3.5957     12.8751
5610.310     11.1715      3.6044     12.8917
5661.020     11.1832      3.6126     12.9072
5711.740     11.1941      3.6202     12.9217
5813.160     11.2134      3.6337     12.9474
5863.870     11.2219      3.6396     12.9586
5914.590     11.2295      3.6450     12.9688
5965.300     11.2364      3.6498     12.9779
```

(continues on next page)

(continued from previous page)

```
  6016.010     11.2424      3.6540     12.9859
  6066.720     11.2477      3.6577     12.9929
  6117.440     11.2521      3.6608     12.9988
  6168.150     11.2557      3.6633     13.0036
  6218.860     11.2586      3.6653     13.0074
  6269.570     11.2606      3.6667     13.0100
  6320.290     11.2618      3.6675     13.0117
  6371.000     11.2622      3.6678     13.0122
```

- Station inventory (format consistent with ObsPy read_inventory routine see https://docs.obspy.org/packages/autogen/obspy.core.inventory.inventory.read_inventory.html)
- Days to process: one column file including days to process e.g. lista1

```
090330
```

- Set parameters: e.g. trim.par

```
#Line 1 -- list of stations
#Line 2 -- list of channels
#Line 3 -- list of networks
#Line 4 -- Lowpass frequency
#Line 5 -- Highpass frequency
#Line 6 -- Trimmed Time before S-wave
#Line 7 -- Trimmed Time after S-wave
#Line 8 -- UTC precision
#Line 9 -- Continuous data dir
#Line 10 -- Template data dir
#Line 11 -- Processing days list
#Line 12 -- Zmap catalog
#Line 13 -- Starting template
#Line 14 -- Stopping template
#Line 15 -- Taup Model
AQU CAMP CERT FAGN FIAM GUAR INTR MNS NRCA TERO
BHE BHN BHZ
IV MN
2.0
8.0
2.5
2.5
6
24h
template
lista1
templates.zmap
26
27
aquila_kato
```

- Directory i.e. ./24h where 24h continuous data are stored
- Output dir i.e. ./template (find trimmed time series)

Note that input and output file names, inventories, template catalogs, velocity models are recalled also in the next steps and remain almost fixed. Parameters in files .par could change.

Calculate Travel Times

Needed files:

- Events in a catalog: e.g. templates.zmap (quakeml or zmap format) see ObsPy for format

- Suitable velocity model for computing travel times
- Station inventory (format consistent with ObsPy read_inventory routine see https://docs.obspy.org/packages/autogen/obspy.core.inventory.inventory.read_inventory.html)
- Days to process: one column file including days to process e.g. lista1
- Set parameters: e.g. times.par

```
#Line 1 -- list of stations
#Line 2 -- list of channels
#Line 3 -- list of networks
#Line 4 -- Lowpass frequency
#Line 5 -- Highpass frequency
#Line 6 -- Trimmed Time before S-wave
#Line 7 -- Trimmed Time after S-wave
#Line 8 -- UTC precision
#Line 9 -- Continuous data dir
#Line 10 -- Template data dir
#Line 11 -- Processing days list
#Line 12 -- Zmap catalog
#Line 13 -- Starting template
#Line 14 -- Stopping template
#Line 15 -- Taup Model
AQU CAMP CERT FAGN FIAM GUAR INTR MNS NRCA TERO
BHE BHN BHZ
IV MN
2.0
8.0
2.5
2.5
6
template
ttimes1
lista1
templates.zmap
26
27
aquila_kato
```

- Input directory i.e. ./template where trimmed templates are found
- Output dir i.e. ./ttimes (find moveout times from different channels used to synchronize cross-correlation functions)

Template Kurtosis-based Waveform Check

Kurtosis statistics is used to evaluate the simmetry of the time series neglecting from the pool of trimmed templates waveforms that show high simmetry in the S-wave selected time window. It is supposed that low signal to noise ratios have low values of Kurtosis index. This contributes to exclude the template/channel from the estimation of the cross-correlation. The routine avoids also signals ior glitches that are located at the beginning and at the end of the signal (please check carefully the code before using it). Scipy Kurtosis is used as a standard routine (see https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kurtosis.html)

An example using the input_template_check is provided also showing accpted and removed waveforms.

3.5 Main Program

PyMPA procedure provides the detection of microseismicity starting from well located templates by using a cross-correlation function from a network. The code is stored on , and can be freely cloned on your platform. It supports Python 2.7, 3.4, 3.5, 3.6, 3.7 releases and uses for reading and writing seismic data, and for handling most of the event metadata. Matched-filter correlations are calculated using a python normalised cross-correlation function or the ObsPy v. 1.2.0 released on April 2019. Detections can be also obtained using a single station and three channels by modifying the input parameters (thresholds etc..). This version is running on a single core and does not include multiprocessing routines. However, in the need of massive calculations for years and thousands of templates, it could be easily implemented a script using SLURM or other schedulers to submit many jobs to the available processors.

Important: we recommend to use an updated version of ObsPy.

Main packages contains:

- *Template matching by using daily estimation of MAD and all the available channels*;

This package is written by the PyMPA developers, and is distributed under the LGPL GNU Licence, Copyright PyMPA developers 2019.

3.5.1 Contents:

Running PyMPA

The input files are those prepared in advance by using pre-processing tools. The only input file that changes is the parameters24 input file.

```
# input parameters for runnig 27 version
# line1 = stations available,
# line2 = channels available,
# line3 = networks available,
# line4 = low bandpass filter frequency,
# line5 = high bandpass filter frequency,
# line6 = sample tolerance in detecting maximum cft amplitude for each channel,
# line7 = cross-correlation threshold to be overcome by cft,
# line8 = min number of channels overcoming the cross-correlation threshold,
# line9 = template duration(s),
# line10 = UTCDateTime.DEFAULT_PRECISION (number of digits considered in fractions␣
↪of seconds),
# line11 = string variable containing the directory name for continuous 24h␣
↪waveforms,
# line12 = string variable for templates' directory,
# line13 = string variable for travel times directory,
# line14 = filename for day list to process,
# line15 = filename for zmap catalog,
# line16 = template start number,
# line17 = template stop number(if line16==0  and line17==0 all templates are␣
↪processed,
# line18 = multiplying factor for MAD to determine threshold
# line 19 = multiplying factor to remove daily cft channels with std greater than␣
↪(average std from all channels * factor at at line 19)
# line 20 = multiplying factor to remove daily cft channels with std smaller than␣
↪(average std from all channels * factor at at line 20)
# line 21 = maximum number of templates to be used in template matching (choice is␣
↪made preferring the closest channels)
# line 22 = number of chunks per day (1=86400s, 2=43200, 3=28800, 4=21600, 6=14400␣
↪etc... icreasing the factor allows reducing memory consumption)
APEC ATBU ATCA ATCC ATFO ATLO ATPC ATPI ATSC ATVO BADI FOSV FRON MURB NARO PARC␣
↪PIEI PE3 SSFR
EHE EHN EHZ HHE HHN HHZ
```

(continues on next page)

(continued from previous page)

```
IV
3.0
8.0
6
0.35
6
5
6
24h
template
ttimes
lista1
templates.zmap
200
203
8
1.5
0.25
12
6
```

Needed input files:

- Events in a catalog: e.g. templates.zmap (quakeml or zmap format) see ObsPy for format
- Suitable velocity model for computing travel times
- Station inventory (format consistent with ObsPy read_inventory routine see https://docs.obspy.org/packages/autogen/obspy.core.inventory.inventory.read_inventory.html)
- Days to process: one column file including days to process e.g. lista1
- Set parameters: e.g. parameters24
- Input directory ./template where trimmed templates are found
- Input directory ./24h where 24 hours continuous waveforms are stores
- Input directory /ttimes (find moveout times from different channels used to synchronize cross-correlation functions)

Output:

- Output files .cat, .stats, .stats.mag, .except (details on the output)

Detections (.cat)

Detections are listed in a .cat file (e.g. 200.100723.cat)

```
200 2010-07-23T02:20:29.833321Z 0.18 0.624 19.077 0.388 11.855 11
200 2010-07-23T06:22:08.273321Z 0.4 0.565 16.951 0.289 8.667 10
200 2010-07-23T06:46:09.553321Z 0.55 0.652 19.56 0.351 10.526 11
200 2010-07-23T22:20:57.553321Z 1.88 1.0 28.431 0.361 10.273 12
200 2010-07-23T22:32:10.713321Z -0.35 0.499 14.198 0.318 9.04 10
200 2010-07-23T22:42:34.113321Z -0.02 0.555 15.773 0.292 8.302 11
200 2010-07-23T22:53:24.393321Z 0.72 0.642 18.254 0.282 8.012 12
200 2010-07-23T23:09:24.953321Z 1.77 0.675 19.178 0.407 11.568 12
```

Columns in 200.100723.cat file are:

- Template number corresponding to the python line index in file templates.zmap (event catalog)
- UTC Date and Time (2010-07-23T02:20:29.833321Z) Time precision selection is possible in parameters24 input file

- Magnitude estimated as in Peng and Zhao (2009). The magnitude of the detected event is calculated as the median value of the maximum amplitude ratios for all channels between the template and detected event, assuming that a 10-fold increase in amplitude corresponds to a one-unit increase in magnitude.
- Average cross-correlation estimated from the channels that concurred to the detection. This value is estimated using a time shift between the channels that optimized the stacked CFT.
- Threshold value (ratio between the amplitude of the CFT stacking and the daily MAD Median Absolute Deviation). The higher the threshold the most probable the detection. This value is estimated using a time shift between the channels that optimized the stacked CFT.
- Average cross-correlation estimated from the channels that concurred to the detection at no time shift.
- Threshold value (ratio between the amplitude of the CFT stacking and the daily MAD Median Absolute Deviation). No time shift of the signal cross-correlation functions is allowed.
- Number of channels for which the cross-correlation is over a certain lower bound (e.g. 0.35)

Single Channel Statistics is listed in a .stats file (e.g. 200.100723.stats)

```
IV.ATFO HHE -0.010616075281 0.721025616044 -1.0
IV.ATFO HHZ 0.283285750909 0.66306439801 -1.0
IV.ATLO EHE 0.777449171081 0.777449171081 0.0
IV.ATLO EHN -0.0852349513925 0.650466054757 1.0
IV.ATLO EHZ 0.322139846713 0.366398836428 -4.0
IV.ATPI EHE 0.537447491069 0.650271304516 -1.0
IV.ATVO HHE 0.435372478132 0.435372478132 0.0
IV.ATVO HHN 0.573539172544 0.573539172544 0.0
IV.ATVO HHZ 0.133907687217 0.346238209304 -1.0
IV.MURB HHE 0.435748932184 0.792480883079 1.0
IV.MURB HHN 0.452987060234 0.713344389699 1.0
IV.MURB HHZ 0.794985414714 0.794985414714 0.0
100723 200 0 2010-07-23T02:20:29.833321Z 0.18 1.88 11.0 0.0326947876687 0.624 19.
↪077 0.388 11.855 12.0 9.0 5.0 0.0
IV.ATFO HHE 0.044988233346 0.231377904481 -5.0
IV.ATFO HHZ 0.149551602865 0.420184950211 -1.0
IV.ATLO EHE 0.224489982653 0.664636238755 1.0
IV.ATLO EHN 0.631160567096 0.631160567096 0.0
IV.ATLO EHZ 0.033085860272 0.262689503234 -3.0
IV.ATPI EHE 0.0362705954363 0.610553204714 -1.0
IV.ATVO HHE 0.297118217589 0.525933024914 -1.0
IV.ATVO HHN 0.36078302459 0.403840027808 -1.0
IV.ATVO HHZ 0.482921869799 0.482921869799 0.0
IV.MURB HHE 0.0567199403744 0.857500320272 1.0
IV.MURB HHN 0.287008419281 0.828772698562 1.0
IV.MURB HHZ 0.864438991339 0.864438991339 0.0
100723 200 0 2010-07-23T06:22:08.273321Z 0.4 1.88 10.0 0.0333501008197 0.565 16.
↪951 0.289 8.667 10.0 7.0 3.0 0.0
IV.ATFO HHE 0.493563159844 0.732363635003 -1.0
IV.ATFO HHZ 0.502878983763 0.601417302353 -1.0
IV.ATLO EHE 0.854286992577 0.854286992577 0.0
IV.ATLO EHN 0.7996230999 0.7996230999 0.0
IV.ATLO EHZ 0.299188562677 0.324593761331 -3.0
IV.ATPI EHE 0.398638163197 0.777572264208 -1.0
IV.ATVO HHE 0.375558339812 0.664585222306 -1.0
IV.ATVO HHN 0.231316704565 0.610737885036 -1.0
IV.ATVO HHZ 0.26865915923 0.599567887303 -1.0
IV.MURB HHE -0.283092335391 0.607940349505 2.0
IV.MURB HHN -0.193080895955 0.584272892006 1.0
IV.MURB HHZ 0.464909097985 0.671106175089 5.0
100723 200 1 2010-07-23T06:46:09.553321Z 0.55 1.88 11.0 0.0333501008197 0.652 19.
↪56 0.351 10.526 12.0 11.0 4.0 0.0
IV.ATFO HHE 0.322210986536 1.0 -1.0
IV.ATFO HHZ 0.253951355525 1.0 -1.0
```

(continues on next page)

(continued from previous page)

```
IV.ATLO EHE 0.278841350001 1.0 -1.0
IV.ATLO EHN 0.354879813216 1.0 -1.0
IV.ATLO EHZ 0.076424274257 1.0 -1.0
IV.ATPI EHE 0.205024190008 1.0 -1.0
IV.ATVO HHE 0.230482228036 1.0 -1.0
IV.ATVO HHN 0.266001050745 1.0 -1.0
IV.ATVO HHZ 0.232455628227 1.0 -1.0
IV.MURB HHE 1.0 1.0 0.0
IV.MURB HHN 1.0 1.0 0.0
IV.MURB HHZ 0.115532925487 1.0 -1.0
100723 200 0 2010-07-23T22:20:57.553321Z 1.88 1.88 12.0 0.0351725931634 1.0 28.431␣
↪0.361 10.273 12.0 12.0 12.0 12.0
IV.ATFO HHE 0.27835785495 0.27835785495 0.0
IV.ATFO HHZ 0.464215345119 0.464215345119 0.0
IV.ATLO EHE -0.553807627447 0.639548344368 2.0
IV.ATLO EHN 0.595817241945 0.682934804256 -1.0
IV.ATLO EHZ 0.124458920838 0.414792472665 -6.0
IV.ATPI EHE 0.490181937608 0.542870771368 1.0
IV.ATVO HHE 0.159972199467 0.543073938124 1.0
IV.ATVO HHN 0.569331631238 0.581437149146 1.0
IV.ATVO HHZ 0.303750325466 0.303750325466 0.0
IV.MURB HHE 0.435371126677 0.435371126677 0.0
IV.MURB HHN 0.364290089669 0.522644091355 1.0
IV.MURB HHZ 0.583644009364 0.583644009364 0.0
100723 200 1 2010-07-23T22:32:10.713321Z -0.35 1.88 10.0 0.0351725931634 0.499 14.
↪198 0.318 9.04 11.0 7.0 0.0 0.0
IV.ATFO HHE 0.315717260245 0.316496580088 -1.0
IV.ATFO HHZ 0.18226247941 0.538348076552 -1.0
IV.ATLO EHE 0.76548108478 0.76548108478 0.0
IV.ATLO EHN -0.30651106846 0.71107343083 -2.0
IV.ATLO EHZ 0.175140751874 0.395886313878 -5.0
IV.ATPI EHE 0.550221988963 0.550221988963 0.0
IV.ATVO HHE 0.667125057705 0.667125057705 0.0
IV.ATVO HHN 0.465986754229 0.701811155038 -1.0
IV.ATVO HHZ 0.150874830603 0.528232273116 -1.0
IV.MURB HHE 0.336709159822 0.582443814666 -1.0
IV.MURB HHN 0.407275625222 0.407275625222 0.0
IV.MURB HHZ -0.206333673397 0.493003146855 -1.0
100723 200 2 2010-07-23T22:42:34.113321Z -0.02 1.88 11.0 0.0351725931634 0.555 15.
↪773 0.292 8.302 12.0 8.0 3.0 0.0
IV.ATFO HHE 0.597024006837 0.666212214776 -1.0
IV.ATFO HHZ 0.576151427278 0.605192504123 -1.0
IV.ATLO EHE 0.802523422374 0.802523422374 0.0
IV.ATLO EHN 0.794246622277 0.794246622277 0.0
IV.ATLO EHZ 0.261713290462 0.54321668075 1.0
IV.ATPI EHE 0.225795605873 0.768023549282 -1.0
IV.ATVO HHE 0.237369132309 0.610432245585 -1.0
IV.ATVO HHN 0.0826140210104 0.54340758269 -1.0
IV.ATVO HHZ 0.0765783808929 0.544169591212 -1.0
IV.MURB HHE -0.298544446625 0.535326480215 2.0
IV.MURB HHN -0.262083078275 0.608894184398 2.0
IV.MURB HHZ 0.288271942621 0.682827589334 1.0
100723 200 3 2010-07-23T22:53:24.393321Z 0.72 1.88 12.0 0.0351725931634 0.642 18.
↪254 0.282 8.012 12.0 12.0 3.0 0.0
IV.ATFO HHE 0.865690619679 0.865690619679 0.0
IV.ATFO HHZ 0.82915443645 0.82915443645 0.0
IV.ATLO EHE 0.675994128404 0.675994128404 0.0
IV.ATLO EHN 0.878930788701 0.878930788701 0.0
IV.ATLO EHZ 0.52260713279 0.52260713279 0.0
IV.ATPI EHE 0.436441880161 0.819360579542 -1.0
IV.ATVO HHE 0.354424434214 0.756587935097 -1.0
```

(continues on next page)

(continued from previous page)

```
IV.ATVO HHN 0.348394023107 0.744906831956 -1.0
IV.ATVO HHZ 0.467055038428 0.59961848772 -1.0
IV.MURB HHE -0.142054506386 0.361216622713 2.0
IV.MURB HHN -0.230821891517 0.538608343873 2.0
IV.MURB HHZ -0.123187495339 0.501711512207 1.0
100723 200 4 2010-07-23T23:09:24.953321Z 1.77 1.88 12.0 0.0351725931634 0.675 19.
↪178 0.407 11.568 12.0 11.0 6.0 0.0
```

Columns in 200.100723.stats file are:

- Network.Station
- Channel
- Cross-correlation value at no time shift
- Cross-correlation value with time shift (nsamples) as in column 5
- Time shift in nsamples (e.g. -1.0 means that the shift is equal to 0.05 at 20Hz sampling rate)

At the end of each trace id you find other parameters related to the detection in part repeating the detection parameters in .cat file and in part related to the cross-correlations values over some limits (0.3 - 0.5 - 0.7 - 0.9).

- date, template_num, detection_num, date&time, template_magnitude, detection_magnitude, threshold_fixed, MAD, ave_crosscc, threshold_record, ave_crosscc_0, threshold_record_0, num_channels_gt0.3, num_channels_gt0.5, num_channels_gt0.7, num_channels_gt0.9
- 100723 201 0 2010-07-23T22:20:57.712239Z 1.51 0.07 9.0 0.0342230009997 0.486 14.193 0.301 8.796 11.0 5.0 2.0 0.0

```
ATLO.EHZ 0.596105028863
ATLO.EHE 0.170393549956
ATLO.EHN 0.373070244109
ATVO.HHE 0.329951522763
ATVO.HHZ 0.42550877913
ATVO.HHN 0.0483225655661
MURB.HHZ 0.191841464532
MURB.HHE 0.0947942692072
MURB.HHN 0.171838180932
ATPI.EHE 0.250359739024
ATFO.HHZ 0.145329568414
ATFO.HHE 0.201548385896
100723 200 0 2010-07-23T02:20:29.833321Z 0.18 1.88 11.0 0.0326947876687 0.624 19.
↪077 0.388 11.855 12.0 9.0 5.0 0.0
ATLO.EHZ 0.388076513533
ATLO.EHE 0.292051072606
ATLO.EHN 0.495577417282
ATVO.HHE 0.351604524548
ATVO.HHZ 0.379804503709
ATVO.HHN 0.144512967971
MURB.HHZ 0.495658912113
MURB.HHE 0.331121201612
MURB.HHN 0.497348207422
ATPI.EHE 0.151842475707
ATFO.HHZ 0.36823188478
ATFO.HHE -0.00476247218528
100723 200 0 2010-07-23T06:22:08.273321Z 0.4 1.88 10.0 0.0333501008197 0.565 16.
↪951 0.289 8.667 10.0 7.0 3.0 0.0
ATLO.EHZ 0.576184745362
ATLO.EHE 0.679097346202
ATLO.EHN 0.546824188445
ATVO.HHE 0.476992588924
ATVO.HHZ 0.634929765059
ATVO.HHN 0.345685333249
```

(continues on next page)

(continued from previous page)

```
MURB.HHZ 0.618940291946
MURB.HHE 0.349556189062
MURB.HHN 0.53763608087
ATPI.EHE 0.411736225028
ATFO.HHZ 0.539730631169
ATFO.HHE 0.490461193276
100723 200 1 2010-07-23T06:46:09.553321Z 0.55 1.88 11.0 0.0333501008197 0.652 19.
↪56 0.351 10.526 12.0 11.0 4.0 0.0
ATLO.EHZ 1.88
ATLO.EHE 1.88
ATLO.EHN 1.88
ATVO.HHE 1.88
ATVO.HHZ 1.88
ATVO.HHN 1.88
MURB.HHZ 1.88
MURB.HHE 1.88
MURB.HHN 1.88
ATPI.EHE 1.88
ATFO.HHZ 1.88
ATFO.HHE 1.88
100723 200 0 2010-07-23T22:20:57.553321Z 1.88 1.88 12.0 0.0351725931634 1.0 28.431␣
↪0.361 10.273 12.0 12.0 12.0 12.0
ATLO.EHZ 0.00504472888794
ATLO.EHE -0.123587952907
ATLO.EHN -0.219563912481
ATVO.HHE -0.52594517306
ATVO.HHZ -0.291161883007
ATVO.HHN -0.341123672838
MURB.HHZ -0.645272339198
MURB.HHE -0.617399606897
MURB.HHN -0.702125882823
ATPI.EHE -0.492439858968
ATFO.HHZ -0.221244903401
ATFO.HHE -0.0530834564922
100723 200 1 2010-07-23T22:32:10.713321Z -0.35 1.88 10.0 0.0351725931634 0.499 14.
↪198 0.318 9.04 11.0 7.0 0.0 0.0
ATLO.EHZ 0.413064678236
ATLO.EHE 0.232081767138
ATLO.EHN -0.0433873767435
ATVO.HHE 0.0144100500685
ATVO.HHZ 0.28271537475
ATVO.HHN -0.0540216138896
MURB.HHZ -0.12470249651
MURB.HHE -0.263968423814
MURB.HHN -0.380872281083
ATPI.EHE 0.111693437768
ATFO.HHZ 0.00414520780979
ATFO.HHE -0.0528771192165
100723 200 2 2010-07-23T22:42:34.113321Z -0.02 1.88 11.0 0.0351725931634 0.555 15.
↪773 0.292 8.302 12.0 8.0 3.0 0.0
ATLO.EHZ 0.803093035554
ATLO.EHE 0.615888381013
ATLO.EHN 0.748348463219
ATVO.HHE 0.695137600201
ATVO.HHZ 0.960014109901
ATVO.HHN 0.731588902278
MURB.HHZ 0.981265250687
MURB.HHE 0.615341321839
MURB.HHN 0.825145491738
ATPI.EHE 0.656415513667
ATFO.HHZ 0.834391164831
```

(continues on next page)

(continued from previous page)

```
ATFO.HHE 0.682137125844
100723 200 3 2010-07-23T22:53:24.393321Z 0.72 1.88 12.0 0.0351725931634 0.642 18.
↪254 0.282 8.012 12.0 12.0 3.0 0.0
ATLO.EHZ 1.67348690528
ATLO.EHE 1.27352216828
ATLO.EHN 1.76932866181
ATVO.HHE 1.81632078689
ATVO.HHZ 1.85243210825
ATVO.HHN 1.63997793344
MURB.HHZ 1.74803392605
MURB.HHE 1.54245846659
MURB.HHN 1.71741408018
ATPI.EHE 1.75089086943
ATFO.HHZ 1.89092026561
ATFO.HHE 1.88418067696
100723 200 4 2010-07-23T23:09:24.953321Z 1.77 1.88 12.0 0.0351725931634 0.675 19.
↪178 0.407 11.568 12.0 11.0 6.0 0.0
```

Columns in 200.100723.stats.mag file are:

- Station.Channel Mag.
- date, template_num, detection_num, date&time, template_magnitude, detection_magnitude, threshold_fixed, MAD, ave_crosscc, threshold_record, ave_crosscc_0, threshold_record_0, num_channels_gt0.3, num_channels_gt0.5, num_channels_gt0.7, num_channels_gt0.9
- 100723 201 0 2010-07-23T22:20:57.712239Z 1.51 0.07 9.0 0.0342230009997 0.486 14.193 0.301 8.796 11.0 5.0 2.0 0.0

Note that input and output file names, inventories, template catalogs, velocity models are recalled also in the next steps and remain almost fixed. Parameters in files .par could change.

3.6 Creating an output catalog and verify detections

It consists of some separate utilities in , post-processing tools to obtain a catalog and verify events. Many events could be correlated to more than one template in a narrow time window. A fixed time window length can be selected, and within each, the template for which the normalized cross-correlation coefficient is the greates provides the event location and data to determine the magnitude. This process is run by <./sub/output.process_detections> in to two steps, by using the last event origin time as a reference to set the next time window scrutinized. The final catalog should be verified by visual inspection for a number of sampled detections. Generally, we proceed by verification of events having low thresholds to understand a safe value to validate the catalog. The routine <./sub/output.verify_detection> creates graphs of time windows where continuous data and trimmed templates are plotted with info grasped from channel by channel cross-correlation process.

Important: we recommend to use an updated version of .

These utilities contains:

- *Postprocessing routines*;
- *Visual verification of detections*;

This package is written by the PyMPA developers, and is distributed under the LGPL GNU Licence, Copyright PyMPA developers 2019.

3.6.1 Contents:

Process PyMPA Output

From the defined template by cross-correlation for a selected day is obtained a list of possible detections. Since time overlapping detections could be found also by different templates, the procedure allows a search for the template able to detect the events with the highest threshold value that is also related with the highest average cross-correlation value for the used network. From the main program, many cat files are daily sorted and grouped for a scan of the events showing the best detections. A bash script is used to collect data and create the input for the procedure and a python script is used to filter the result on the basis of the fiter.par parameters used. The filter.par defines some additional filtering to detections to possibly overcome a visual validation of the new detections.

Visual verification of detections

Visual verification is needed to evaluate problematic cases or estimate a safe threshold to validate the catalog of detections. Visual verification needs the new detections catalog, the templates.zmap list, the daily template detections (e.g. 230.140913.cat file) and relative statistics (e.g. 230.140913.stats)cat file) and relative statistics (e.g. 230.140913.stats)cat file) and relative statistics (e.g. 230.140913.stats). The verify.par parameters allow for selecting the stations, frequency range, and window length for the visual verification.