

# ObsPy: A Python Toolbox for Seismology/Seismological Observatories

Tobias Megies<sup>1</sup>, Robert Barsch<sup>1</sup>, Moritz Beyreuther, Lion Krischer<sup>1</sup>, Joachim Wassermann<sup>1</sup> and The ObsPy Development Team

<sup>1</sup> Department of Earth and Environmental Sciences, Ludwig-Maximilians-Universität München

Contact: [dews@obspy.org](mailto:dews@obspy.org) <http://www.obspy.org>



## Summary

Python combines the possibilities of a full-blown programming language with the flexibility of an interactive scripting language. Its extensive standard library and many freely available high quality scientific modules cover most needs in developing scientific processing workflows.

ObsPy extends Python's capabilities to fit the specific needs that arise when working with seismological data. It *a*) provides read and write support for many common [waveform<sup>1</sup>](#) and [metadata file formats<sup>2</sup>](#), *b*) enables access to [data centers](#), [webservices](#) and [databases<sup>3</sup>](#) to retrieve waveform data and station/event metadata and *c*) comes with a continuously growing [signal processing toolbox<sup>4</sup>](#) that covers the most common tasks in seismological analysis.

In combination with widely used, free Python packages like [NumPy<sup>5</sup>](#), [SciPy<sup>6</sup>](#), [Matplotlib<sup>7</sup>](#), [IPython<sup>8</sup>](#) and [PyQt<sup>9</sup>](#), ObsPy makes it possible to **develop complete workflows** in Python, ranging from reading/requesting data via signal analysis and data processing to visualization in GUI applications, output of modified/derived data and creating **publication-quality figures**. All functionality is [extensively documented<sup>10</sup>](#) and the [ObsPy Gallery/Tutorial<sup>10</sup>](#) give a good impression of the wide range of use cases. ObsPy is tested and running on **Linux**, **MacOSX** and **Windows XP/Vista/7** and comes with installation routines for these systems. ObsPy is developed in a test-driven approach and is available under the **GPL/LGPLv3** licences.

Users are welcome to request help, report bugs or propose enhancements via the user [mailing list<sup>11</sup>](#) or the Trac [ticket system<sup>12</sup>](#).

<sup>1</sup> MiniSEED, SAC, GSE2, SEISAN, SEGY, ...

<sup>2</sup> SEED, Dataless SEED, XML-SEED, RESP, ...

<sup>3</sup> IRIS, ArcLink/WebDC, EMSC, VEBSN, SeisHub, ...

<sup>4</sup> filtering, instrument correction/simulation, spectrograms, triggering, array analysis, probabilistic power spectral densities, instrument calibration, ...

<sup>5</sup> <http://numpy.scipy.org>

<sup>6</sup> <http://scipy.org>

<sup>7</sup> <http://matplotlib.sourceforge.net>

<sup>8</sup> <http://ipython.scipy.org>

<sup>9</sup> <http://www.riverbankcomputing.co.uk/software/pyqt>

<sup>10</sup> <http://docs.obspy.org>

<sup>11</sup> <http://lists.obspy.org>

<sup>12</sup> <http://obspy.org>

## Basic Example

```
import obspy.core
import obspy.neries
date = obspy.core.UTCDateTime(2011, 10, 23)
client = obspy.neries.Client(user='test@obspy.org')
# get magnitude 7+ event for this date (Turkey) from NERIES webservice
event = client.getEvents(min_magnitude=7,
                         min_datetime=date,
                         max_datetime=date+24*60*60)[0]
# get 2000 seconds of waveform data starting at origin time
start = event['datetime']
stream = client.getWaveform('GR', 'FUR', '', 'BH*', start, start+2000)
stream.filter('lowpass', freq=0.5)
stream.plot() # the resulting plot is shown to the right ----->
```

## Installation

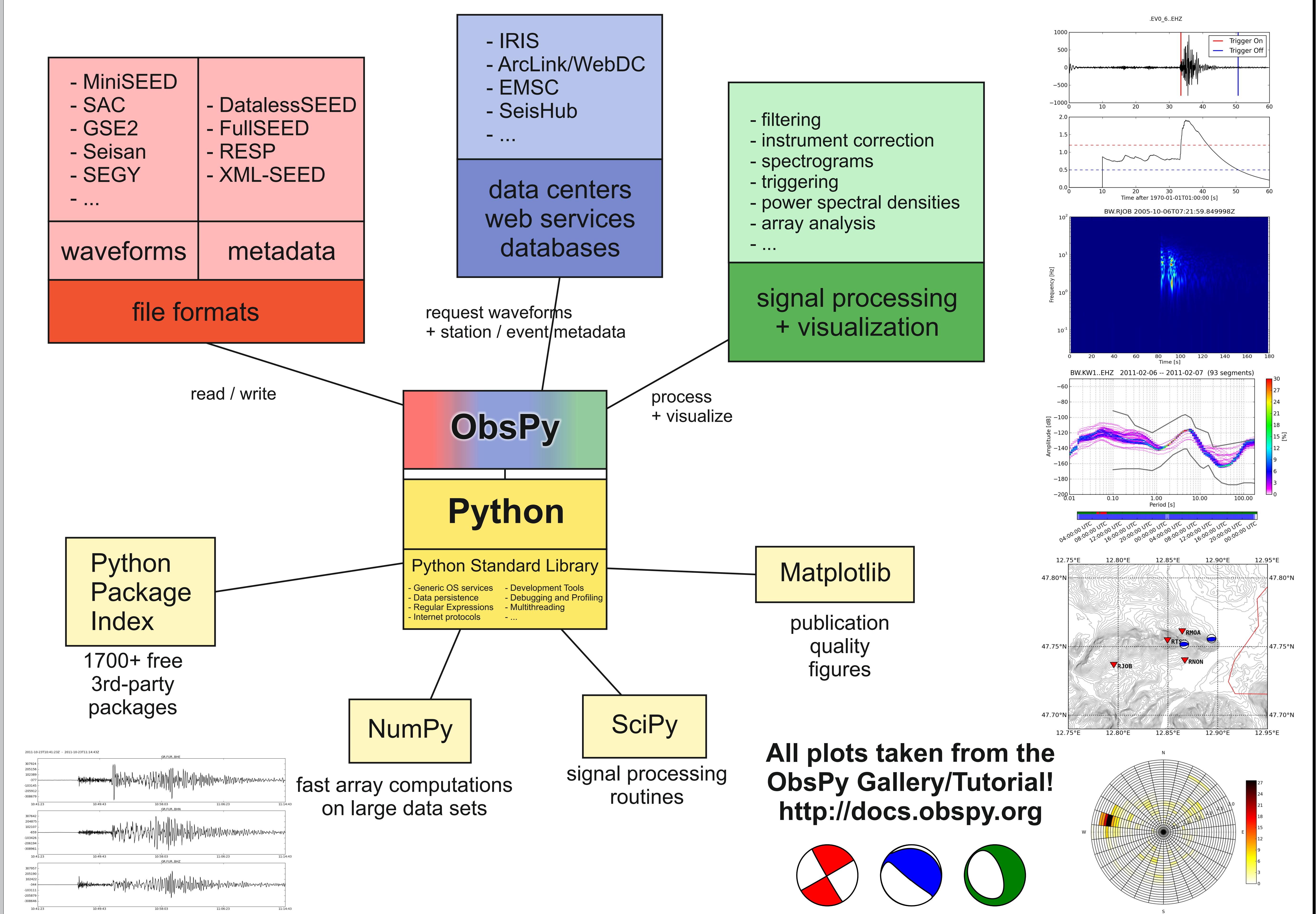
Automated installation routines for the latest stable version are available for Debian/Ubuntu Linux, MacOSX and Windows.

★ **Debian/Ubuntu:** via package management (<http://deb.obspy.org>)

★ **MacOSX:** one-click-install application (contains all dependencies)

★ **Windows:** Windows installer (automatically installs all dependencies)

For the most recent additions and bug fixes, the current developer version can be installed using the Python Package Index (see detailed instructions on our homepage).



## References

BEYREUTHER, M., R. BARSCH, L. KRISCHER, T. MEGIES, Y. BEHR and J. WASSERMANN (2010)

**ObsPy: A Python Toolbox for Seismology**, Seismological Research Letters, 81(3):530-533r. doi:10.1785/gssrl.81.3.530

MEGIES, T., M. BEYREUTHER, R. BARSCH, L. KRISCHER and J. WASSERMANN (2011)

**ObsPy – What can it do for data centers and observatories?**, Annals Of Geophysics, 54(1), 47-58. doi:10.4401/ag-4838

