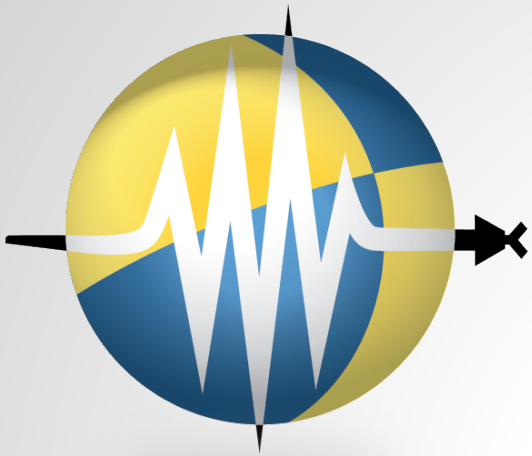


ObsPy

A Python Framework for Seismology

Introduction
Recent Developments
Future Plans

**Tobias Megies, Lion Krischer, Elliott Sales de Andrade,
Robert Barsch, Moritz Beyreuther (...)**

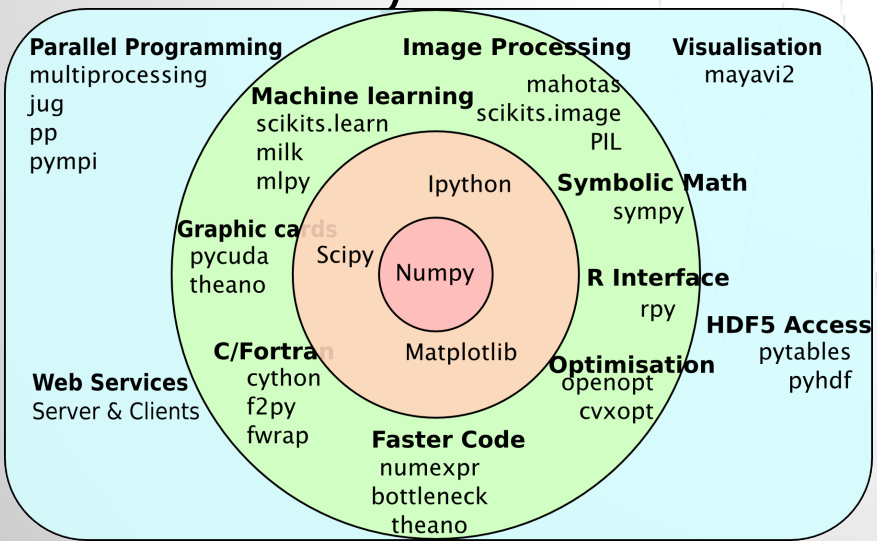
September 2015

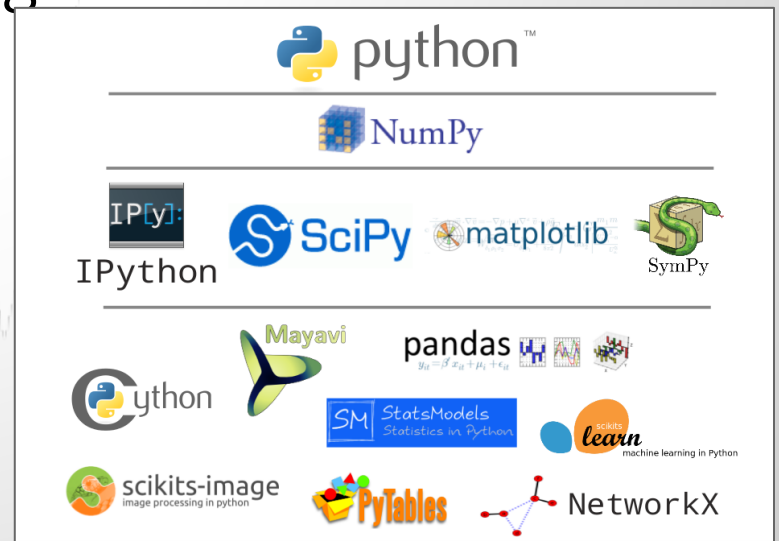
Why Python makes sense for seismology..

- easy to get started
  - one-click, 5 min, all-inclusive installation, binary package distribution (Anaconda)
  - simple, concise, and easy-to-read syntax
  - no compilation steps
  - interactive shell (with help system, tab-completion, ..)
  - speed of development over speed of execution

Why Python makes sense for seismology..

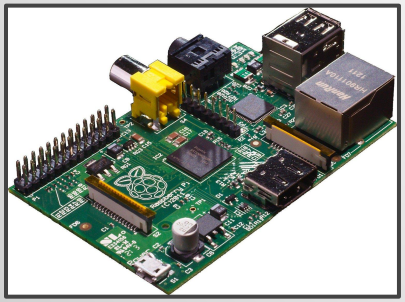
- easy to get started
- high potential
  - general purpose programming language
  - excellent support for scientific purposes
    (big data, signal analysis, visualization, ..)
  - tons of small, special purpose scientific packages
  - easy to interact with existing C/Fortran codes

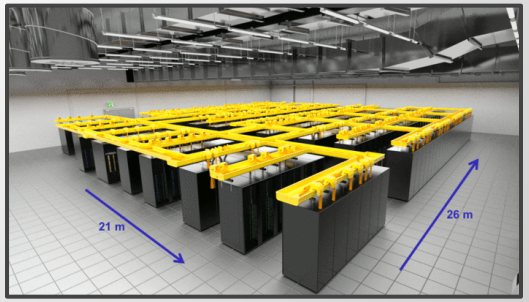




Why Python makes sense for seismology..

- easy to get started
- high potential
- high sustainability/value
  - free and open source
  - readable, reusable, extendable code
  - browser-based notebooks for teaching/sharing
  - cross-platform, cross-architecture
    - Linux, Mac, Windows
    - from Raspberry Pi to SuperMUC





Why Python makes sense for seismology..

- easy to get started
- high potential
- high sustainability/value
- strong, inherent bonds to seismology

Sir Bedevere:

...and that, my liege,
is how we know the Earth
to be banana-shaped.

King Arthur:

This new learning amazes
me, Sir Bedevere.
Explain again how sheep's
bladders may be employed
to prevent earthquakes.

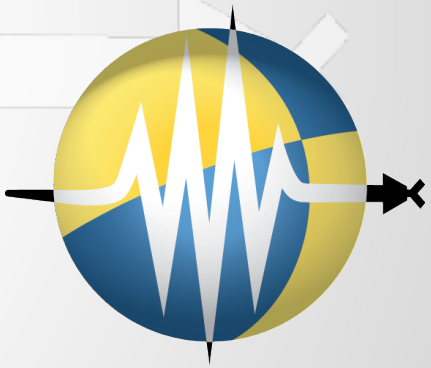
What is ObsPy?

Python library to work with seismological data

- waveform / station / event data

⇒ **teach Python to “talk seismology”**

⇒ **a bridge for seismologists into the scientific Python ecosystem**



Facilitates development

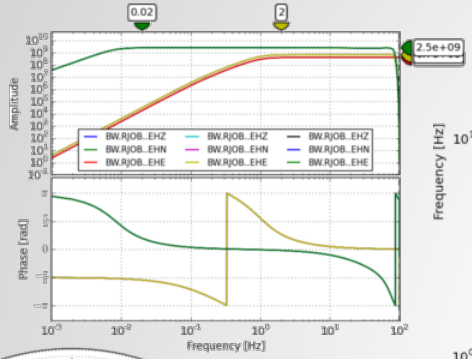
- from short code snippets
- to complex processing workflows

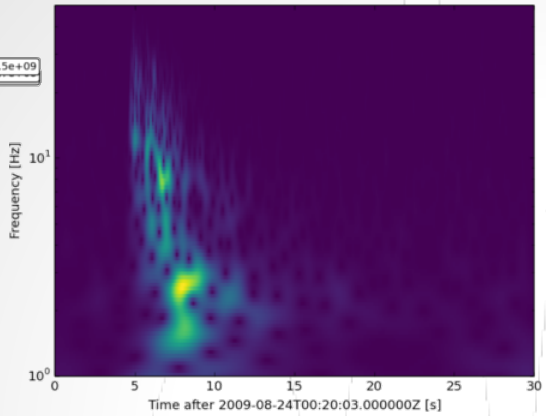
⇒ **make every day seismology tasks easy, complex things possible**

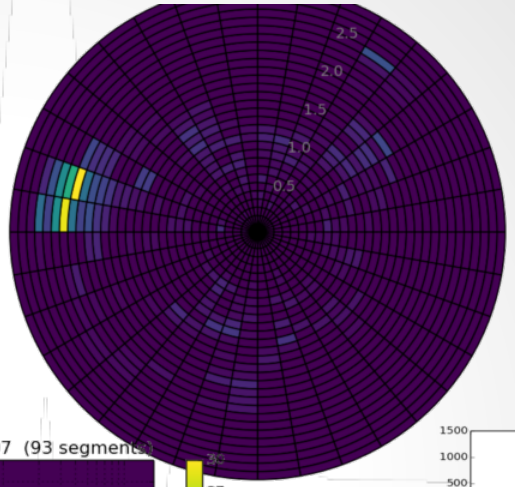
Functionality?

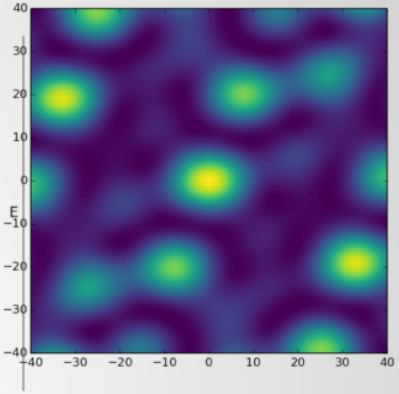
- read/write support
  - waveforms (MiniSEED, SAC, GSE, SEISAN, SEG-Y, SEG2, Q, CSS, ...)
  - station metadata (StationXML, dataless/full SEED, RESP, SACPZ)
  - event metadata (QuakeML, NDK, NLLOC, ZMAP, CNV, MCHEDR)
- data center access
  (FDSN WS, ArcLink, SeedLink, Earthworm, NEIC, ..)
- unified object classes
- signal processing routines
  - trim, merge, taper, rotate, resample, interpolate, ..
  - filtering, instrument correction, ...
  - theoretical travel times/raypaths, array analysis, CC routines, PPSD, event detection, TF misfits, ...
- visualization capabilities

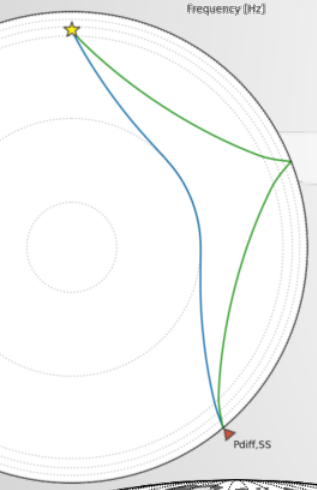
Functionality?

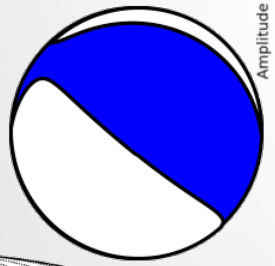


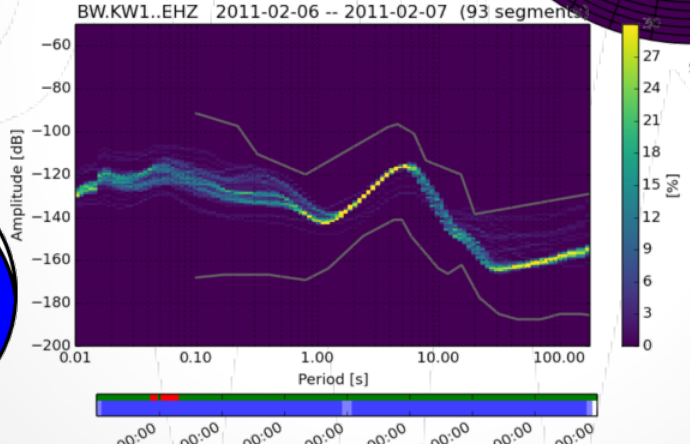


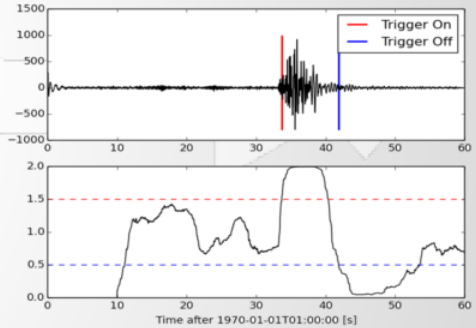


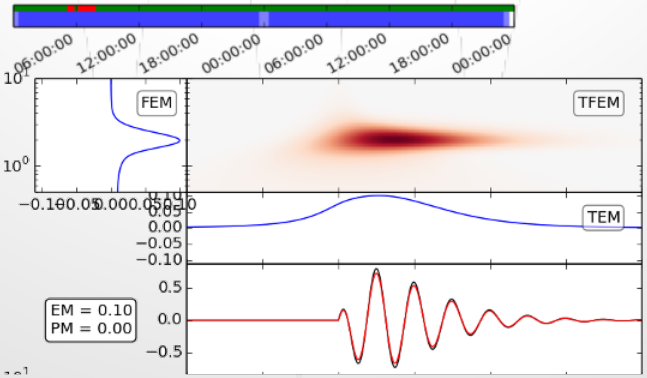


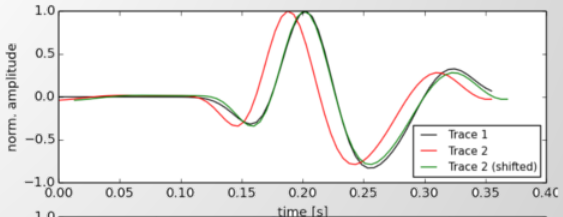


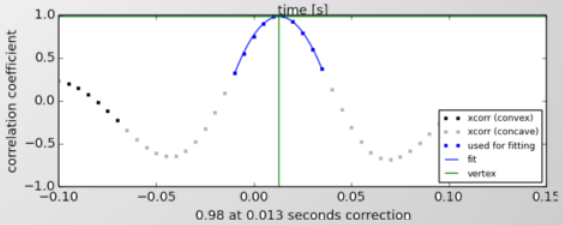


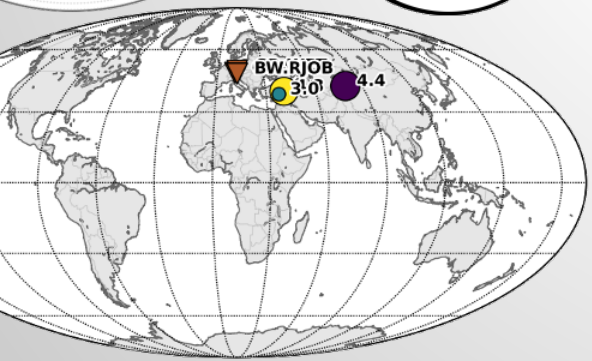






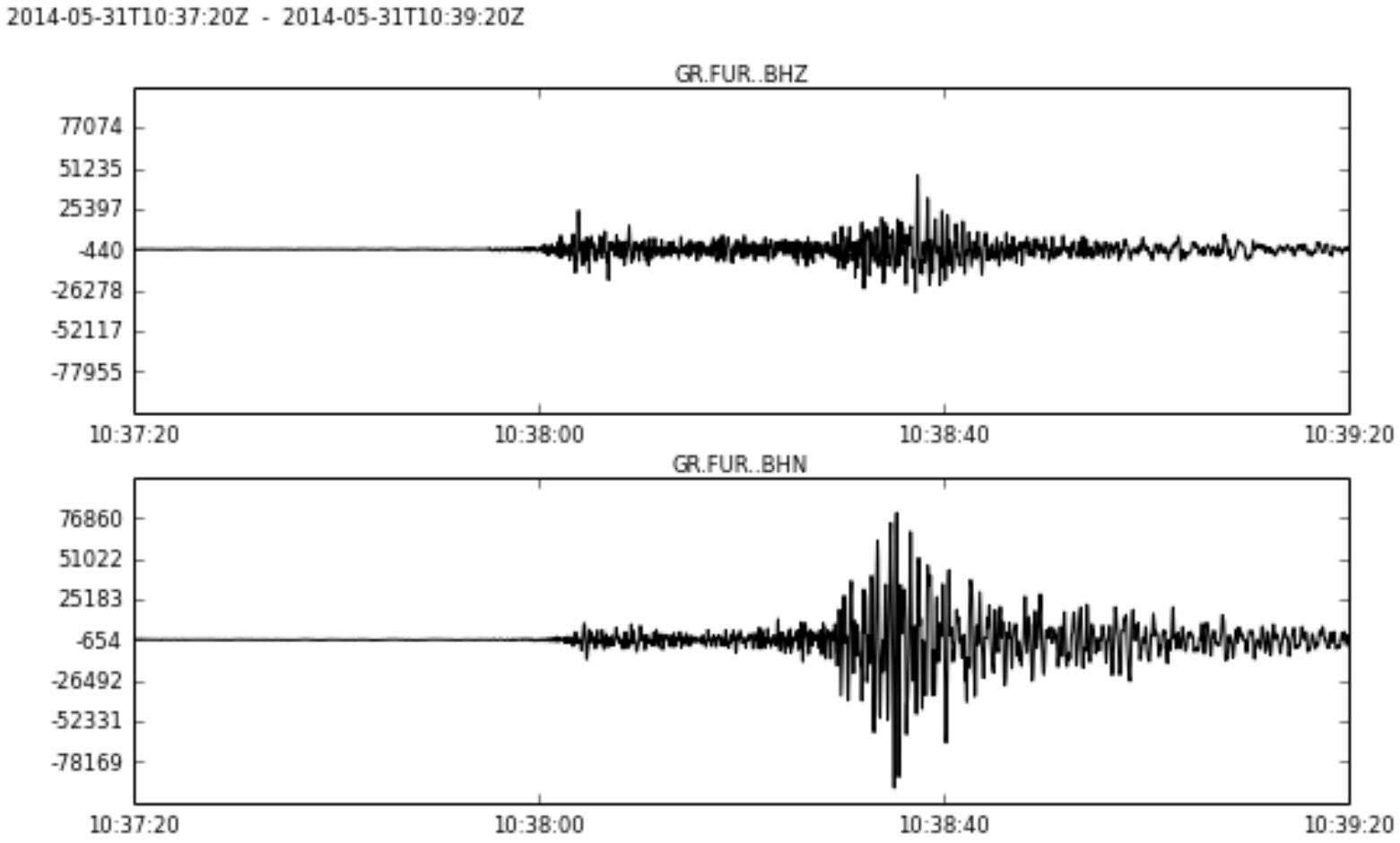






Functionality?

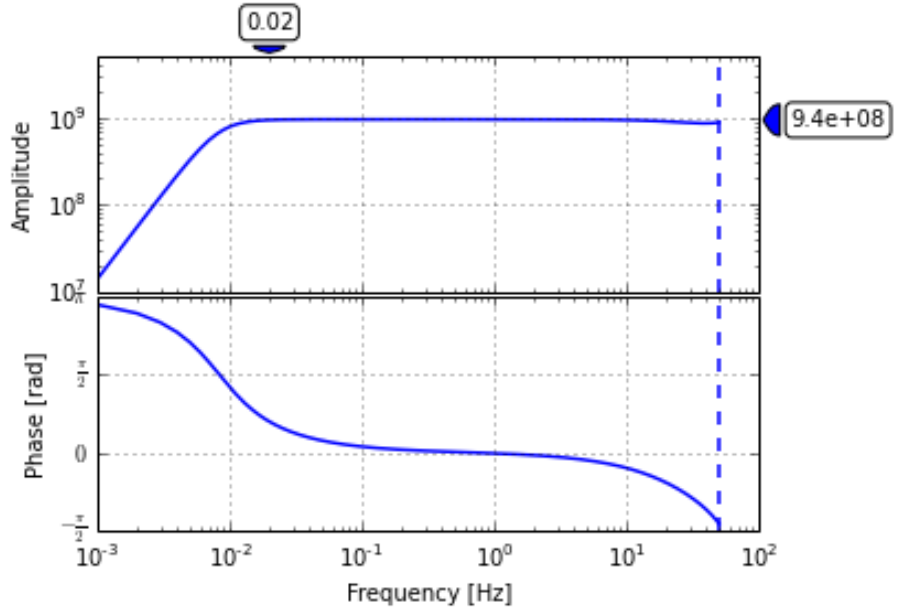
```
In [2]: from obspy import read
        stream = read("waveform.mseed")
        stream.plot()
```



Functionality?

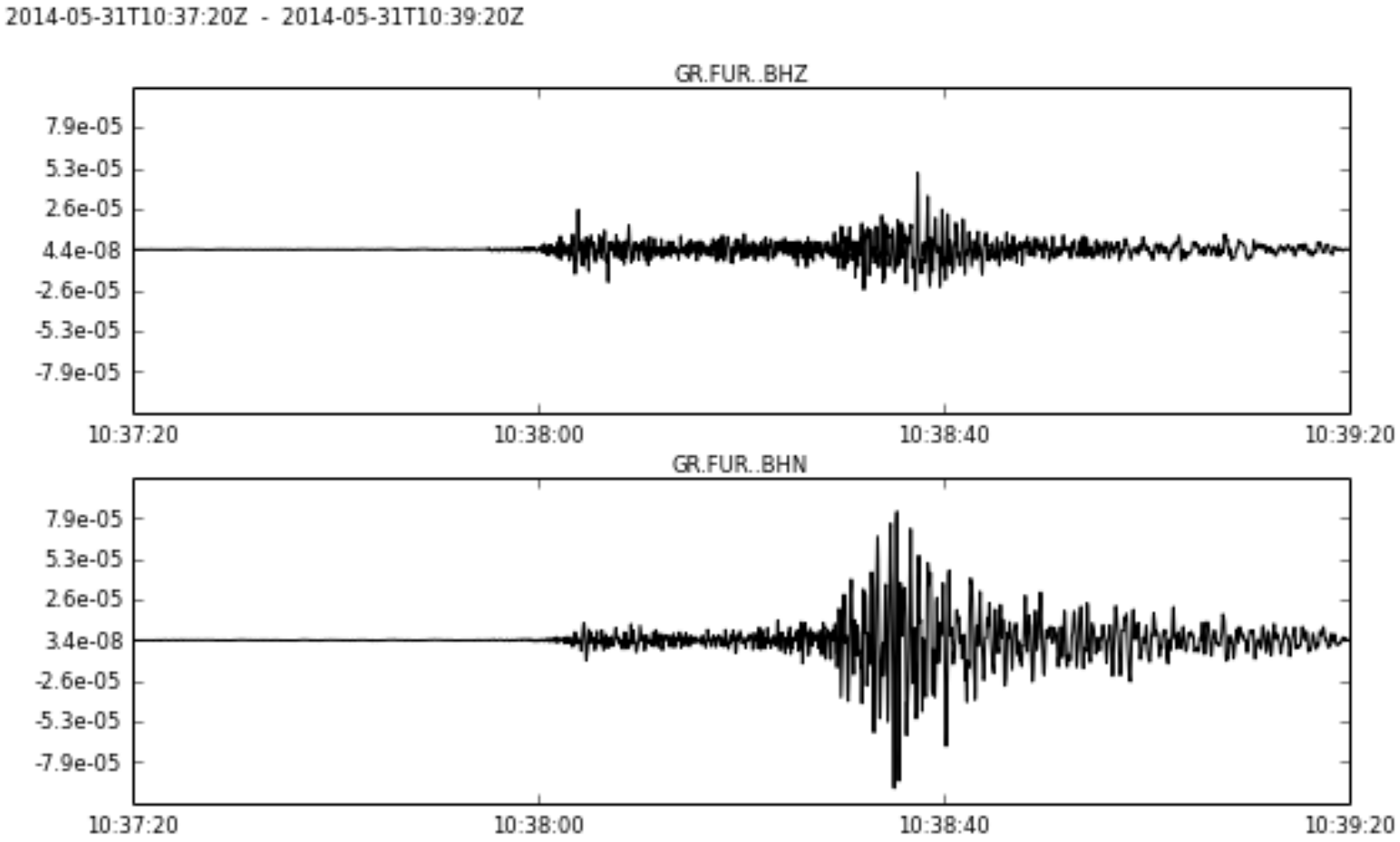
In [3]:
```
from obspy import read_inventory
inventory = read_inventory("station.xml")
channel = inventory[0][0][0]
print channel
channel.plot(0.001)
```

```
Channel 'HHZ', Location ''
        Timerange: 2006-12-16T00:00:00.000000Z - --
        Latitude: 48.16, Longitude: 11.28, Elevation: 565.0 m, Local Depth: 0.0 m
        Azimuth: 0.00 degrees from north, clockwise
        Dip: -90.00 degrees down from horizontal
        Channel types: TRIGGERED, GEOPHYSICAL
        Sampling Rate: 100.00 Hz
        Sensor: Streckeisen STS-2/N seismometer
        Response information available
```



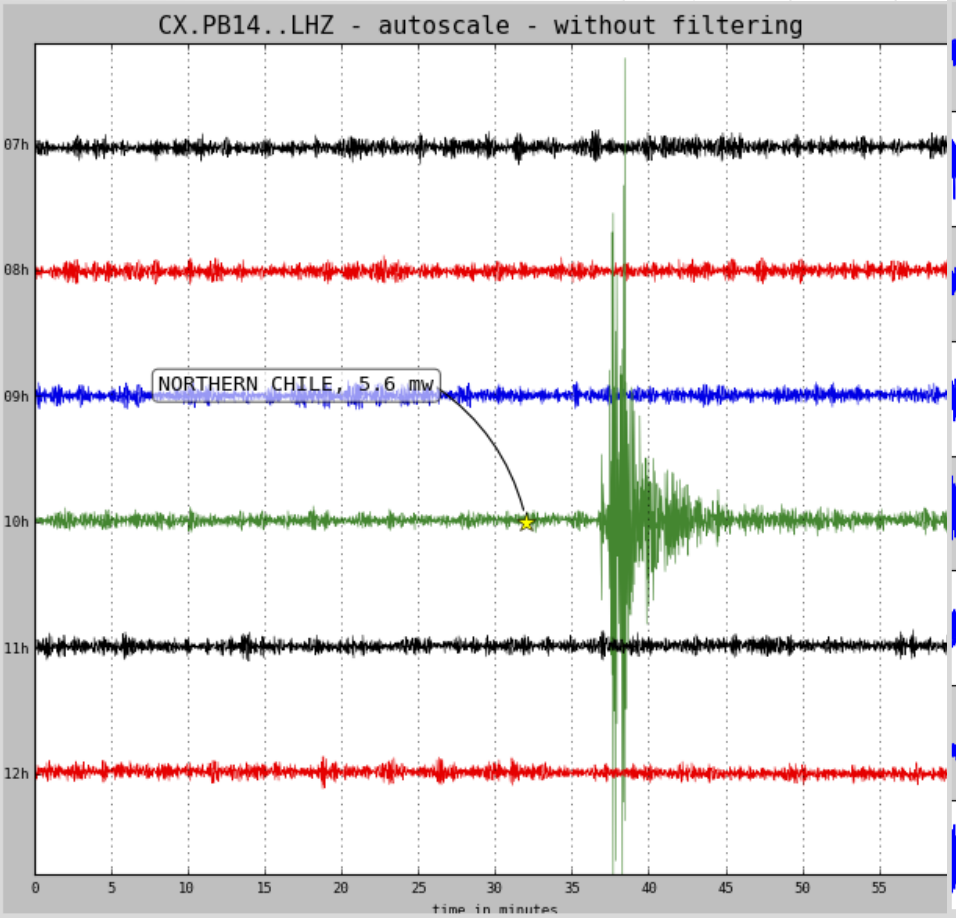
Functionality?

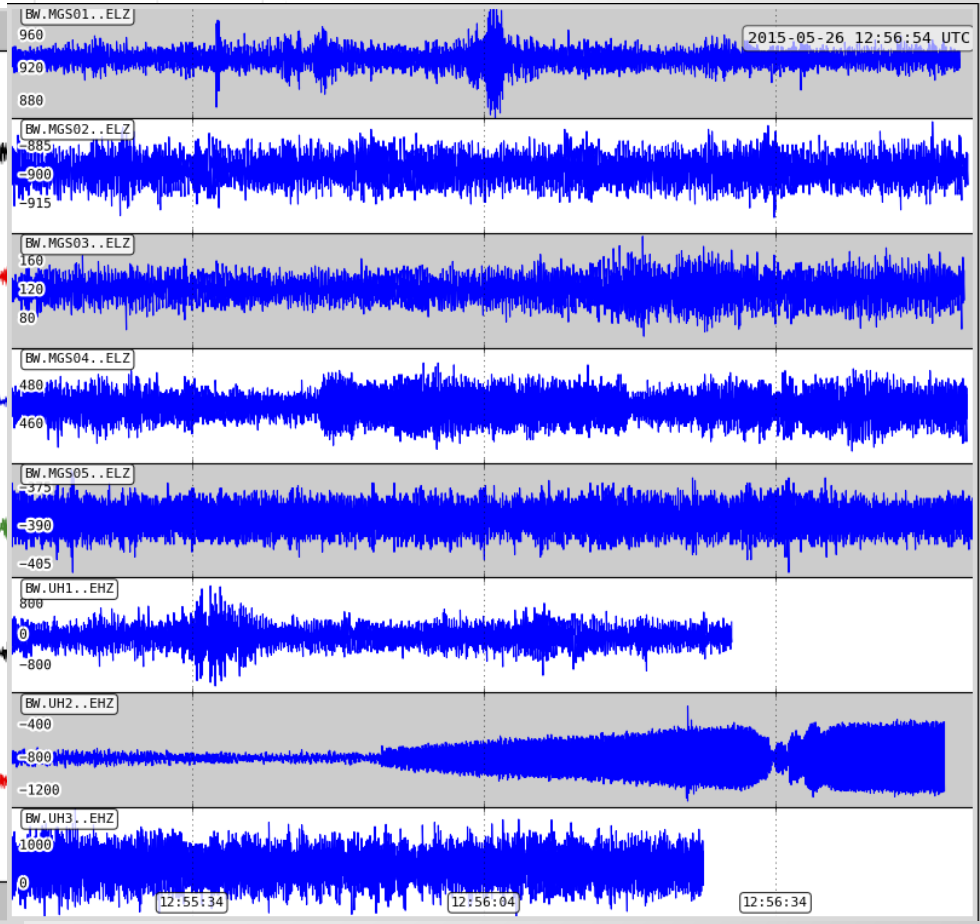
In [4]:
```
stream.attach_response(inventory)
stream.remove_response(output="VEL")
stream.plot()
```



Use cases..

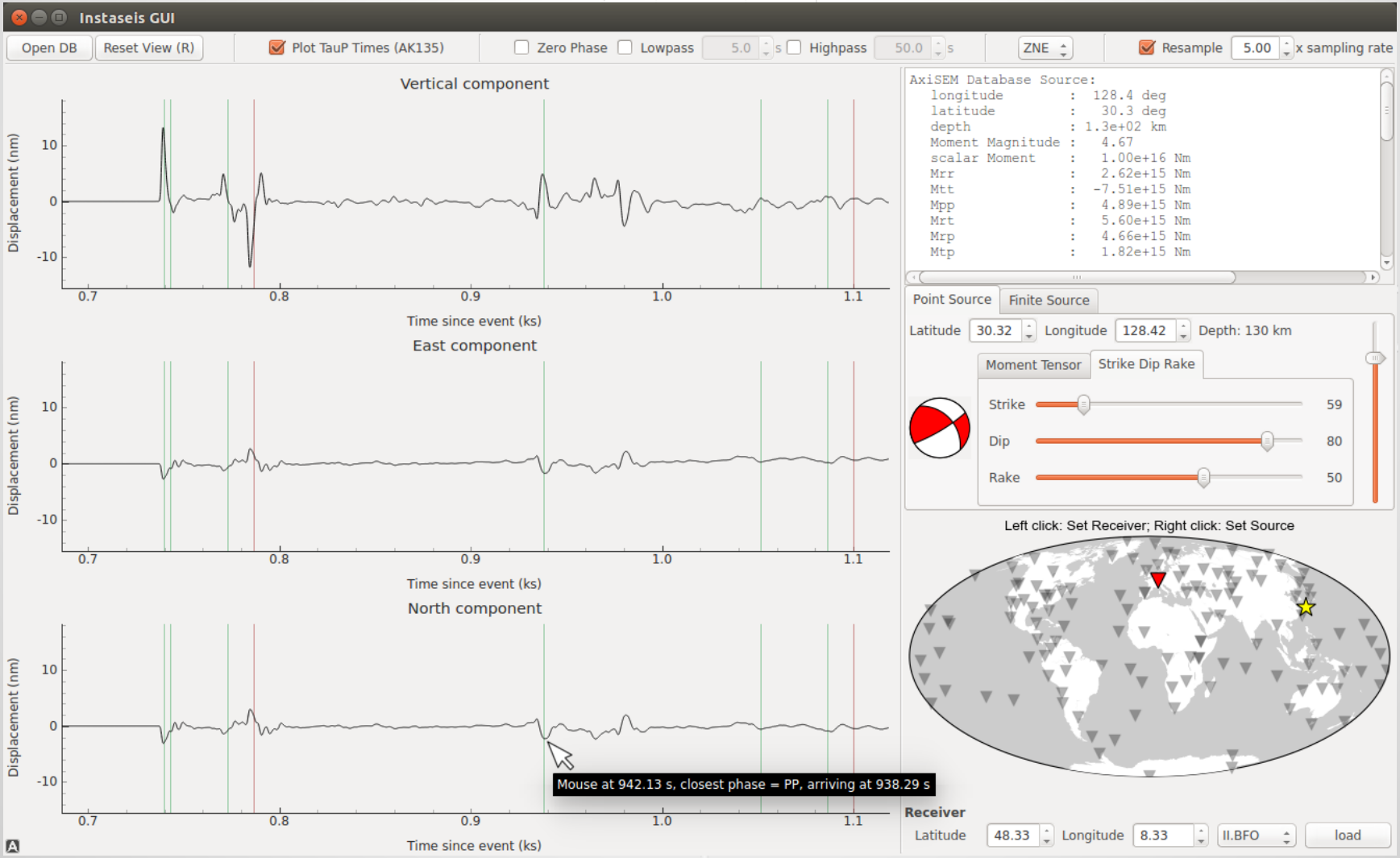
- seedlink plotter





Use cases..

- instaseis



Development goals: try to make it..

..easy to use

- binary packages, system-specific packaging
- documentation: http://docs.obspy.org
- tutorial
- gallery
- mailing list

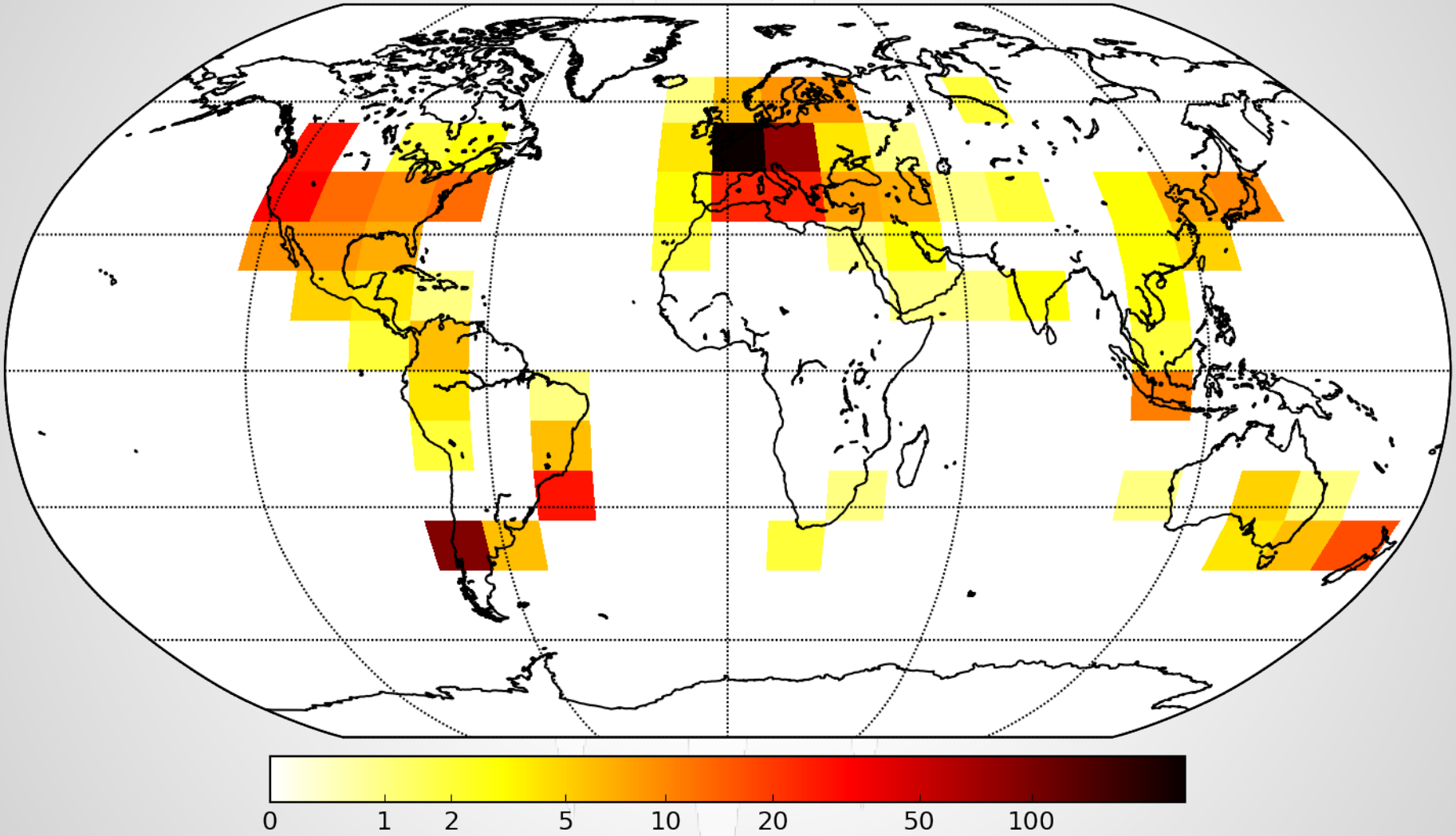
..reliable

- test-driven development
  >1300 tests, 86% Test Coverage, Travis/AppVeyor CI

..transparent and accessible to external contributors

- git: distributed version control, branching/merging
- github: central platform for hosting,
  “social coding”, ticket system

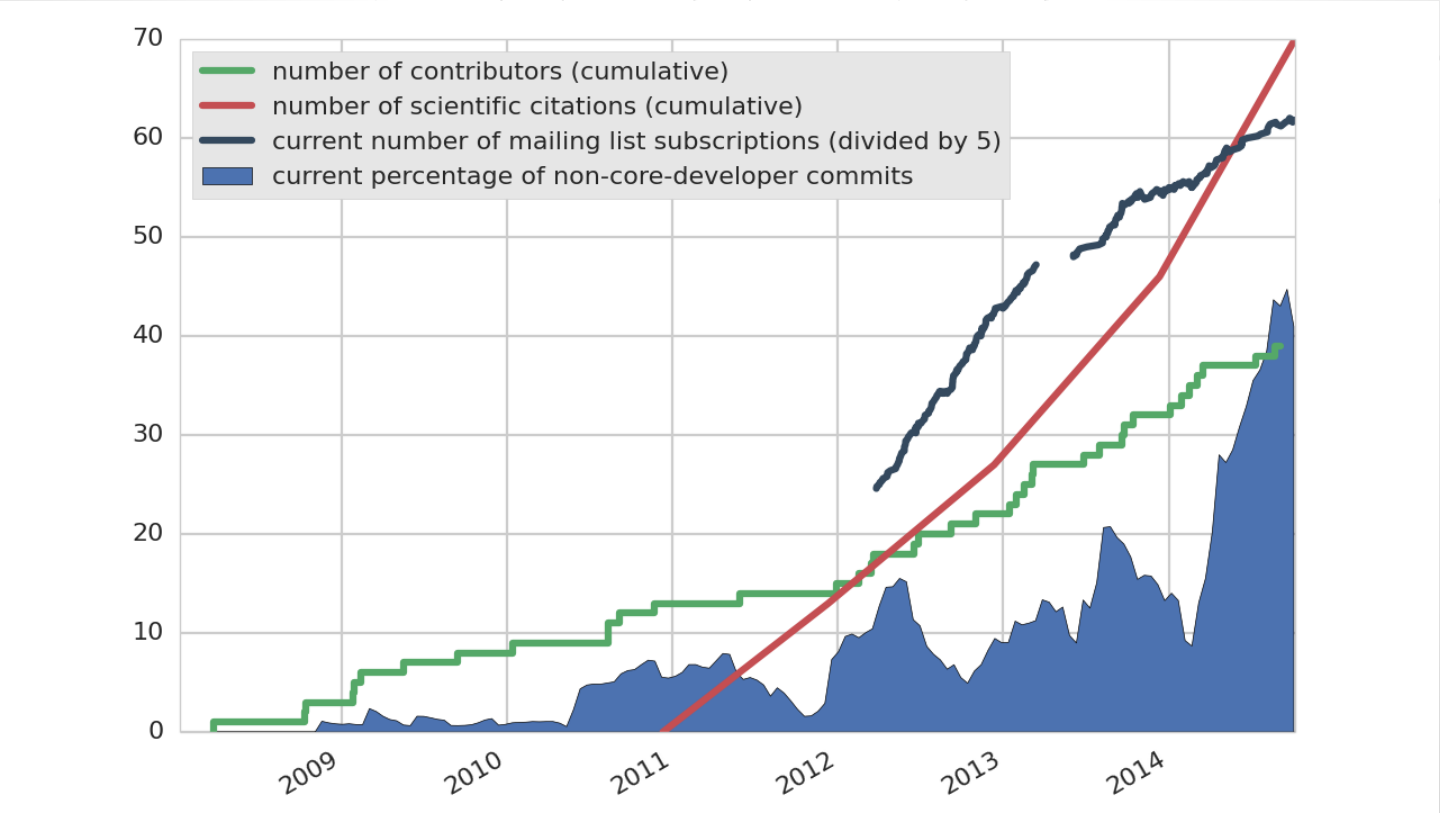
Usage statistics..



ObsPy 0.9.2 Debian/Ubuntu package downloads (821 unique IPs in 449 days)
This represents only a small subset of all Obspy downloads.

Usage statistics..

- >350 people on users mailing list
- 132 scientific citations
  Beyreuther et al 2010, Megies et al 2011, Krischer et al 2015
- 40 contributors

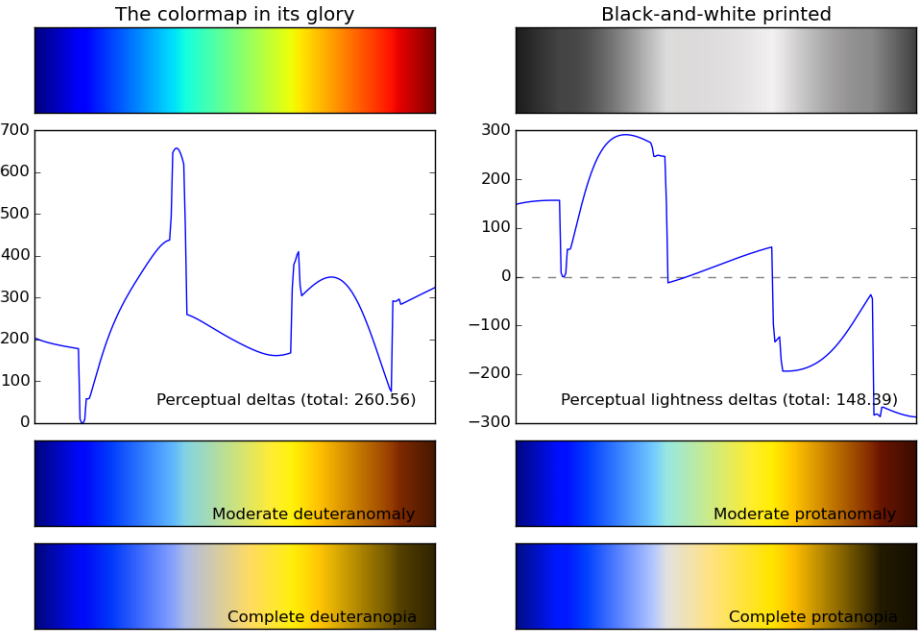


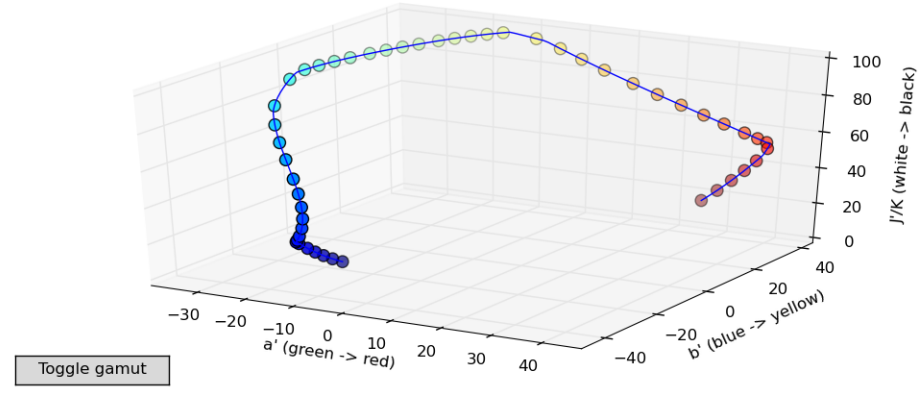
Recent Developments

- Python 2+3 support
- more stable filters (backported from latest scipy release)
- new default colormaps (aka “don’t use jet”)
  - "perceptually uniform", etc.
  - backported from upcoming matplotlib release
- response removal plot
- PPSD changes
  - improved algorithm (esp. affects very long periods)
  - remove full response
  - cumulative / “non-exceedance” plot
- Lanczos interpolation/resampling
- new input/output plugins,
  e.g. shapefiles (GIS), kml (Google Earth), CMTSOLUTION, ...
- ...

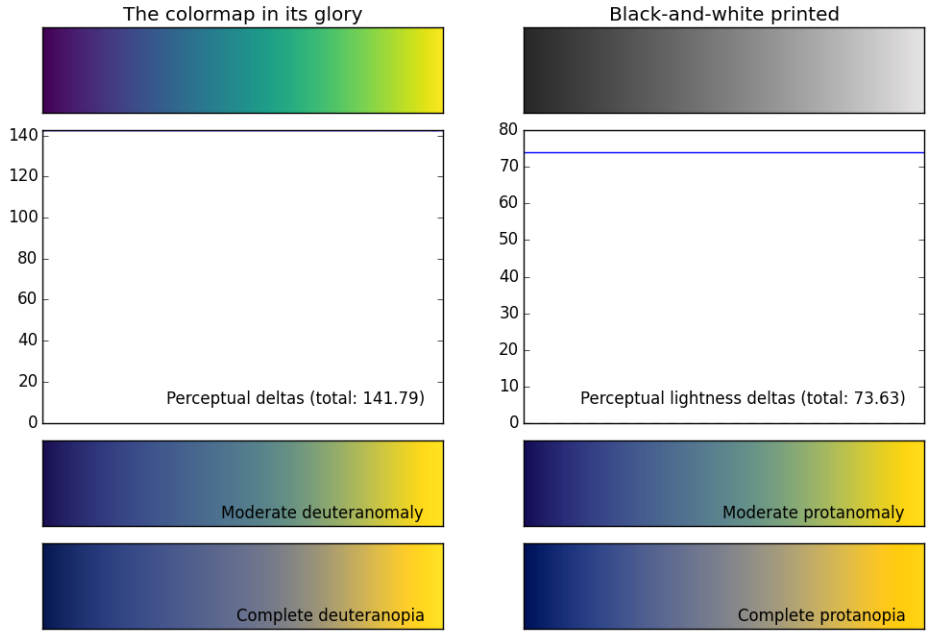
Recent Developments

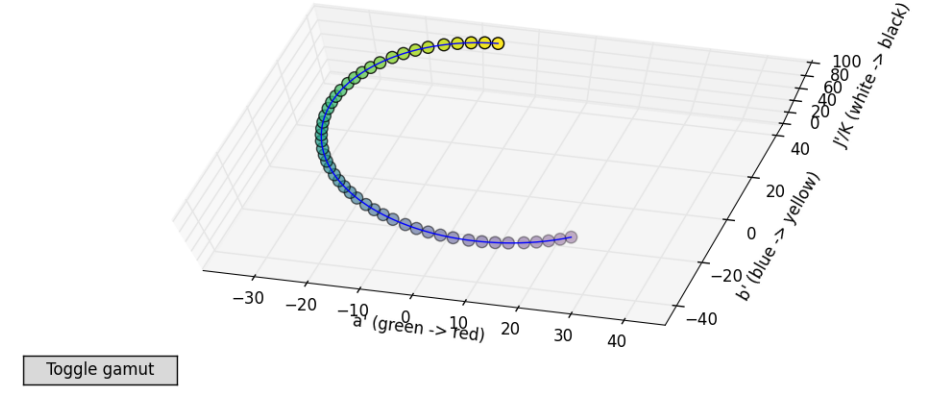
jet





viridis

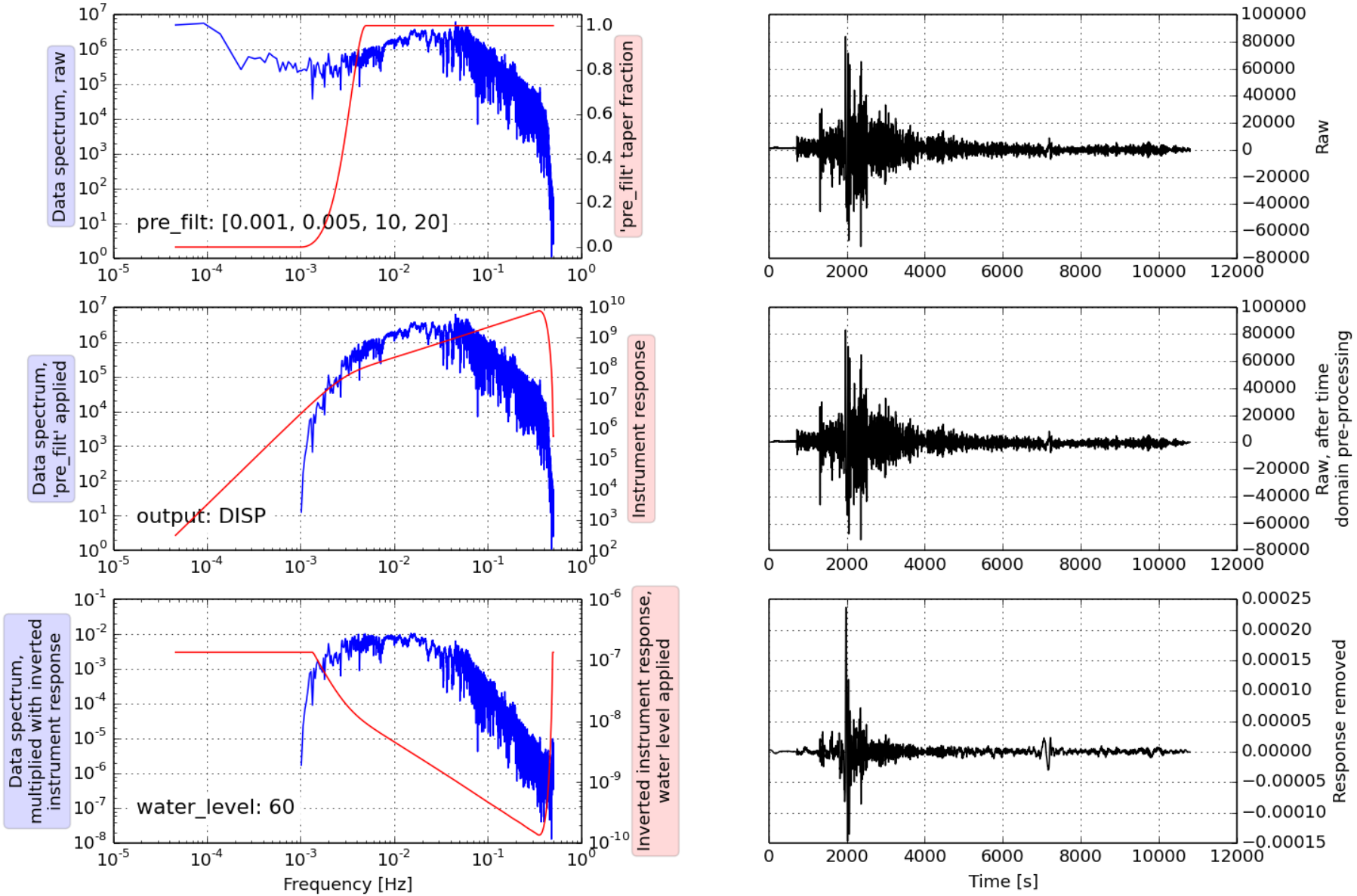




Recent Developments

- Python 2+3 support
- more stable filters (backported from latest scipy release)
- new default colormaps (aka “don’t use jet”)
  - "perceptually uniform", etc.
  - backported from upcoming matplotlib release
- response removal plot
- PPSD changes
  - improved algorithm (esp. affects very long periods)
  - remove full response
  - cumulative / “non-exceedence” plot
- Lanczos interpolation/resampling
- new input/output plugins,
  e.g. shapefiles (GIS), kml (Google Earth), CMTSOLUTION, ...
- ...

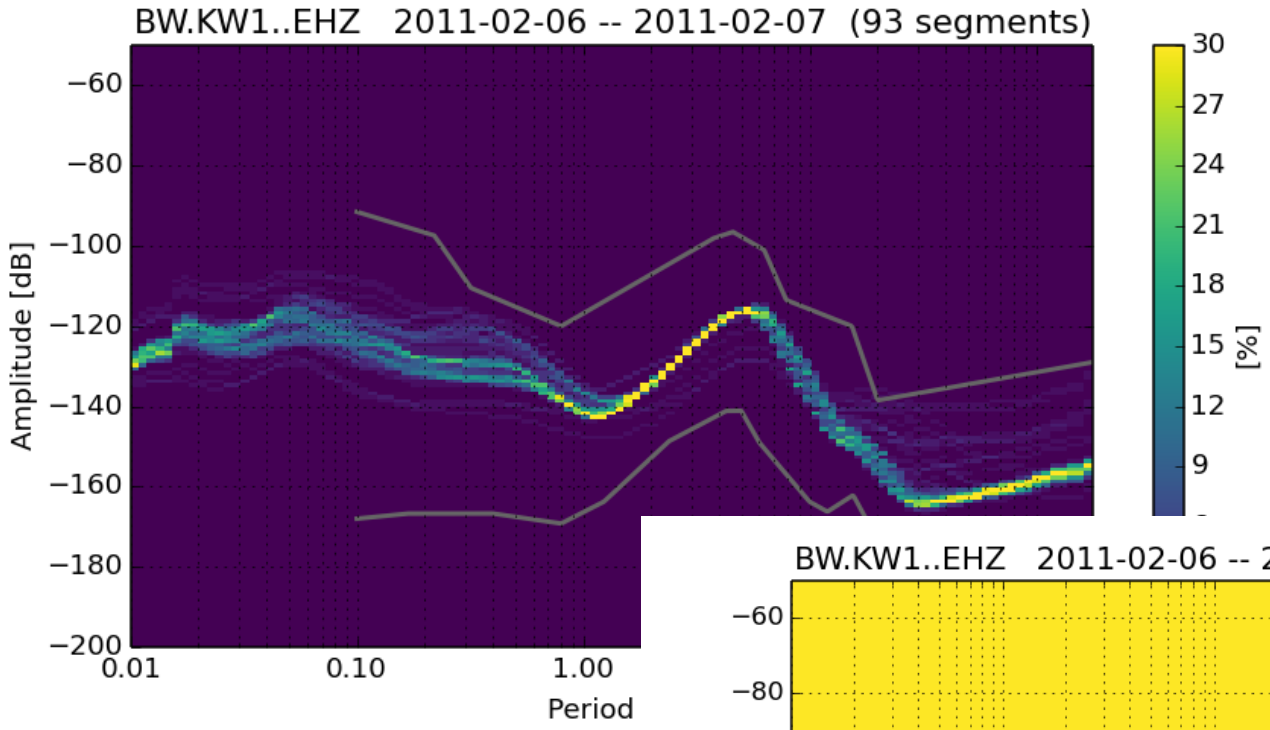
Recent Developments

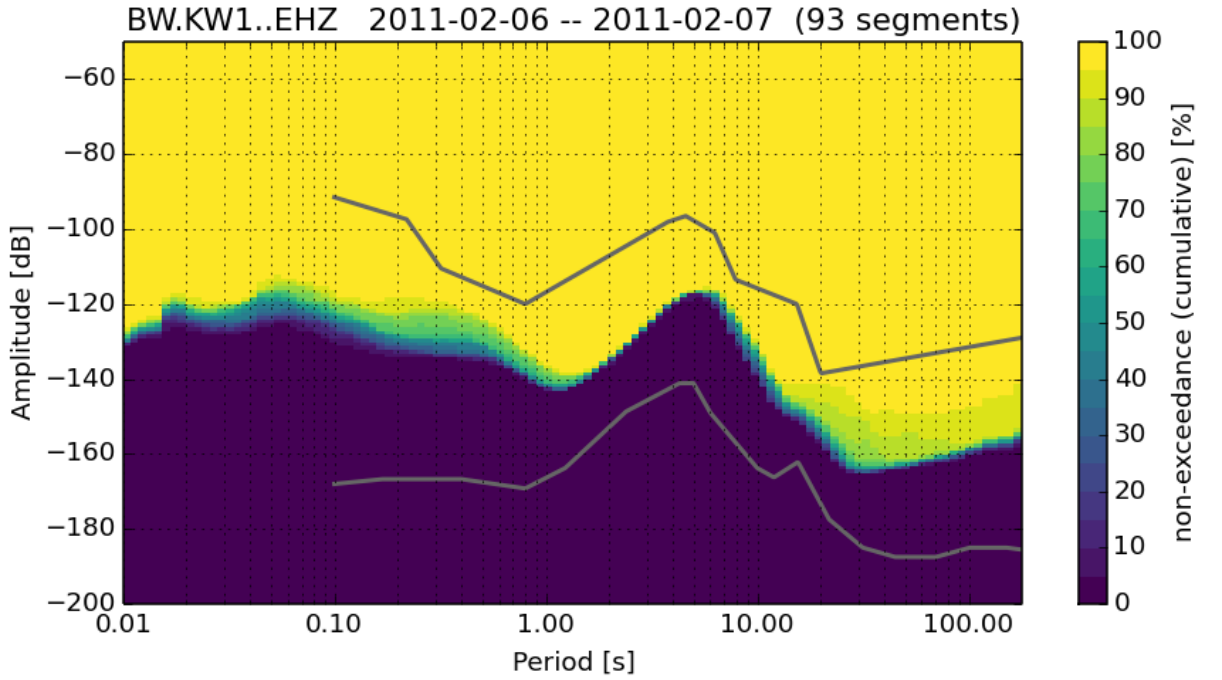


Recent Developments

- Python 2+3 support
- more stable filters (backported from latest scipy release)
- new default colormaps (aka “don’t use jet”)
  - "perceptually uniform", etc.
  - backported from upcoming matplotlib release
- response removal plot
- PPSD changes
  - improved algorithm (esp. affects very long periods)
  - remove full response
  - cumulative / “non-exceedence” plot
- Lanczos interpolation/resampling
- new input/output plugins,
  e.g. shapefiles (GIS), kml (Google Earth), CMTSOLUTION, ...
- ...

Recent Developments



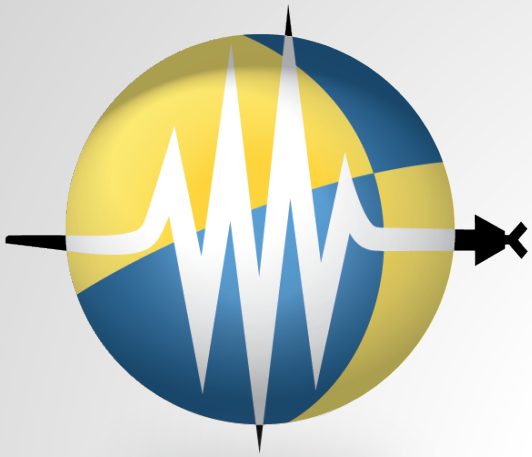


Recent Developments

- Python 2+3 support
- more stable filters (backported from latest scipy release)
- new default colormaps (aka “don’t use jet”)
  - "perceptually uniform", etc.
  - backported from upcoming matplotlib release
- response removal plot
- PPSD changes
  - improved algorithm (esp. affects very long periods)
  - remove full response
  - cumulative / “non-exceedence” plot
- Lanczos interpolation/resampling
- new input/output plugins,
  e.g. shapefiles (GIS), kml (Google Earth), CMTSOLUTION, ...
- ...

Future Plans

- submodule restructuring, major cleanup, consistent naming scheme
  - group all data center / server access plugins
    `obspy.fdsn` → `obspy.clients.fdsn`
  - group all I/O plugins
    `obspy.mseed` → `obspy.io.mseed`
  - ...
- more convenient, canonical handling for frequency domain operations
- better interconnection of waveforms / station /event data (data set object?)
- ...



ObsPy

A Python Framework for Seismology

Thanks for your attention!

www.obspy.org