AIMBAT

• Open-source software package for efficiently measuring teleseismic body wave arrival times for large seismic arrays (Lou et al., 2013).
• Based on MCCC (multi-channel cross-correlation) developed by VanDecar and Crosson (1990).
• Automated picking by aligning via an ICCS (iterative cross-correlation and stack) algorithm.
• Graphical user interface for interactive seismogram quality control.

→ User processing time is reduced while valuable input from a user's expertise is retained.
→ As a byproduct, SAC (Goldstein et al., 2003) plotting and phase picking functionalities are replicated and enhanced.
Welcome to AIMBAT's documentation!

Contents:

- 1. Introduction
  - 1.1. About AIMBAT
  - 1.2. Associated Documents
  - 1.3. Authors' Contacts
- 2. Installing Dependencies
  - 2.1. Github
  - 2.2. Macports
  - 2.3. Homebrew
  - 2.4. Getting your operating system
  - 2.5. Python Dependencies
    - 2.5.1. Optional but recommended
  - 2.6. Installing Basic Python Packages
    - 2.6.1. Enthought Python
    - 2.6.2. If you cannot use Enthought Canopy ...
  - 2.7. Installing Basemap (Python dependency)
  - 2.8. Possible Issues
    - 2.8.1. Macports
Welcome

Sphinx is a tool that makes it easy to create intelligent and beautiful documentation, written by Georg Brandl and licensed under the BSD license.

It was originally created for the new Python documentation, and it has excellent facilities for the documentation of Python projects, but C/C++ is already supported as well, and it is planned to add special support for other languages as well. Of course, this site is also created from reStructuredText sources using Sphinx! The following features should be highlighted:

- **Output formats**: HTML (including Windows HTML Help), LaTeX (for printable PDF versions), ePub, Texinfo, manual pages, plain text
- **Extensive cross-references**: semantic markup and automatic links for functions, classes, citations, glossary terms and similar pieces of information
- **Hierarchical structure**: easy definition of a document tree, with automatic links to siblings, parents and children
- **Automatic indices**: general index as well as a language-specific module indices
- **Code handling**: automatic highlighting using the Pygments highlighter
- **Extensions**: automatic testing of code snippets, inclusion of docstrings from Python modules (API docs), and more

Sphinx uses reStructuredText as its markup language, and many of its strengths come from the power and straightforwardness of reStructuredText and its parsing and translating suite, the Docutils.
LaTeX – A document preparation system
GEOS (Geometry Engine - Open Source) is a C++ port of the Java Topology Suite (JTS) to contain the complete functionality of JTS in C++. This includes all the OpenGIS Simple Spatial Predicates, all spatial predicate functions and spatial operators, as well as specific JTS enhanced topological filters.

GEOS is available under the terms of the GNU Lesser General Public License (LGPL), and is developed by OpenGeo.

Capabilities Include

- Geometries: Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, C R Segment
- Predicates: Intersects, Touches, Disjoint, Crosses, Within, Contains, Overlaps, Equals
- Operations: Union, Distance, Intersection, Symmetric Difference, Convex Hull, Envelope
- Prepared geometries (pre-spatially indexed)
- STR spatial index
- OGC Well Known Text (WKT) and Well Known Binary (WKB) encoders and decoders
- C and C++ API (C API gives long term ABI stability)
- Thread safe (using the reentrant API)

Download

- 2013/08/25 geos-3.4.2.tar.bz2 (Changes)
- 2013/09/04 geos-3.3.9.tar.bz2 (Changes)
- 2011/09/21 geos-3.2.3.tar.bz2 (Changes)
- 2009/06/15 geos-3.1.1.tar.bz2 (Changes)
- Older versions...
- Nightly snapshot: http://geos.osgeo.org/snapshots/
Python - Basemap

 basemap 1.0.7

Plot data on map projections with matplotlib

An add-on toolkit for matplotlib that lets you plot data on map projections with coastlines, lakes, rivers and political boundaries. See http://matplotlib.github.com/basemap/users/examples.html for examples of what it can do.
Installing AIMBAT

- Install pysmo.sac
Installing AIMBAT

- Install `pysmo.aimbat`
.bash_profile

- Edit .profile or .bashrc

```bash
export PATH=$PATH:/Users/lkloh/USAarray/InstallingCommonPrograms

# sod
export PATH=$PATH:/Users/lkloh/sod-3.2.3/bin

export PATH=$PATH:/Users/lkloh/aimbat/scripts
export PATH=$PATH:/Users/lkloh/aimbat/additional-processing-scripts
```
Sort by file Index Name:

- File

Sort by Quality:

- All
- CCC
- SNR
- COH

Sort by Header:

- NPTS
- B
- E
- Delta
- STLA
- STLO
- Dist
- AZ
- BAZ
- GCARC

Waiting for User input

FILENAMES
- File: Sort in alphabetical order by filename

QUALITY:
- All: Weighted Ratio of all quality measures
- CCC: Cross-coefficient Coefficient
- SNR: Signal-to-noise Ratio
- COH: time domain coherence

OTHER HEADERS:
- NPTS: Number of points per data component
- B: Beginning value of the independent variable
- E: Ending value of the independent variable
- Delta: Increment between evenly spaced samples
- STLA: Station latitude (deg, north positive)
- STLO: Station longitude (deg, east positive)
- DIST: Station to event distance (km)
- AZ: Event to station azimuth (deg)
- BAZ: Station to event azimuth (deg)
- GCARC: Station to event great circle arc length (deg)
Butterworth Filter

Select Order:

○ 1
○ 2
○ 3
○ 4
○ 5

Low Freq: 0.005
High Freq: 0.008
Order: 1

Signal vs Time

Amplitude vs frequency

Original
Filtered
Butterworth Filter
Thank you Developers:

Xiaoting Lou
Simon Lloyd
Lay Kuan Lo
Nick Ambruz

Thank you!
Possible Future Development: Using Obspy to download IRIS data from within AIMBAT (volunteers?)

1. `getWaveform()`: The following example illustrates how to request and plot 18 seconds of all three single band channels ("EH\*") of station Jochberg/Hochstaufen ("RJOB") of the Bavarian network ("BW") for a seismic event around 2009-08-20 04:03:12 (UTC).

```python
>>> from obspy import UTCDateTime
>>> from obspy.arclink.client import Client
>>> client = Client(user='test@obspy.org')
>>> t = UTCDateTime("2009-08-20 04:03:12")
>>> st = client.getWaveform("BW", "RJOB", ",", "EH\*", t - 3, t + 15)
>>> st.plot()
```
Possible Future Development: Using GMT-py to make maps within AIMBAT (volunteers?)
AIMBAT exercise 2016 USArray data processing short course