ObsPyLoad Manual

Contents

1 Usage

ObsPyLoad is meant to be used from a shell and can be called without the explicit utilization of the *Python* interpreter. The program tries to be intuitively usable, but as seems unavoidable with shell programs that reach a certain complexity, the list of options may be a little overwhelming at first glance.
Getting started with a command-line only tool may take a little more time for less shell-inclined folks, but there are definitely strong advantages over a *GUI-only* tool. For example, tasks like automatically checking for new events, batch processing or remote server-side usage should be feasible with standard shell tools. The author assumes that most seismologists are into shell usage, anyway.
One design inclination has been to not force the user to provide mandatory options. That means that just typing *obspyload.py* is sufficient to download data. Most users probably will not do this more than once, but it seems unavoidable to tell first-time users what the program will do before entering the download procedure, so when no options are given, the program prints this clarifying message:

```
chris@gauss:~$ obspyload.py

Welcome,
you provided no options, using all default values will
download every event that occurred in the last 3 months
with magnitude > 3 from every available station.
```

```
    ObsPyLoad will now create the folder /home/chris/obspyload-data
    and possibly download vast amounts of data. Continue?
    Note: you can suppress this message with -f or --force
    Brief help: obspyload.py -h
    Long help: obspyload.py -H
    [y/N]>
```

The prompt asks the user to enter "y" or "n". As is common practice, the capital *N* indicates that answering "No" is the default. As is stated in the message, this text can be suppressed with *-f*, which is what most people probably want. As also written in the message, answering *yes* would result in using all default values for the data download.
It will be interesting to the reader to know the default behaviour and how to change some or all aspects of it. Here is a list of all default values:

- the data download mode is default, alternative modes like the metadata download mode or the exception file mode must be entered explicitly
- the default datapath in which the data will be saved is *obspyload-data* in the current working directory. In metadata download mode, the default path is *obspyload-metadata*.
- the default starttime is three months ago, the default end time is now.
- the default velocity model for theoretical travel time calculation is *iasp91*.
- the default preset is 5 minutes, which means that data from each event/station combination will be downloaded starting 5 minutes before estimated arrival time.
- the default offset is 80 minutes, which means that data from each event/station combination will be downloaded until 80 minutes after estimated arrival time.
- the default minimum magnitude is 3.
- by default, there is no geographical restriction, events from the whole globe will be downloaded.
- by default, there is no network/station restriction. All available stations will be downloaded, including temporary ones.
- by default, no plot will be created. If the plot will be created, the default internal resolution will be 1200x800x1 and the default timespan for the plot will be 100 minutes. By default, the plot will not be filled with more data than is downloaded anyway.
- by default, if the plot is created, the 'P' and 'S' phases will be plotted on top.

The *ObsPyLoad* command line tool uses a syntax which has been, to some degree, influenced by the *Generic Mapping Tools (GMT)* (`http://gmt.soest.hawaii.edu`). Most options can be given in two different flavours. There usually is one option combining related values as one option divided by some delimiter like a slash or a dot, and another set of options achieving the same thing separately.
It makes sense to have a look at the custom written long help function of *ObsPyLoad* at this time, since this will clarify the concept:

1.1 Long help function

Special effort has been put into the two help functions. This one will be printed if the user types

```
    $ obspyload.py -H
```

```
ObsPyLoad: ObsPy Seismic Data Download tool.
============================================
The CLI allows for different flavors of usage, in short:
--------------------------------------------------------

  e.g.:            obspyload.py -r <west>/<east>/<south>/<north> -t
                   <start>/<end> -m <min_mag> -M <max_mag> -i <nw>.<st>.<l>.<ch>
  e.g.:            obspyload.py -y <min_lon> -Y <max_lon> -x <min_lat> -X
                   <max_lat> -s <start> -e <end> -P <datapath> -o <offset>
                   --reset -f


You may (no mandatory options):
-------------------------------

* specify a geographical rectangle:

  Default:         no constraints.
  Format:          +/- 90 decimal degrees for latitudinal limits,
                   +/- 180 decimal degrees for longitudinal limits.

  -r[--rect]       <min.longitude>/<max.longitude>/<min.latitude>/<max.latitude>
                   e.g.: -r -15.5/40/30.8/50

  -x[--lonmin]     <min.latitude>
  -X[--lonmax]     <max.longitude>
  -y[--latmin]     <min.latitude>
  -Y[--latmax]     <max.latitude>
                   e.g.: -x -15.5 -X 40 -y 30.8 -Y 50


* specify a timeframe:

  Default:         the last 3 months
  Format:          Any obspy.core.UTCDateTime recognizable string.

  -t[--time]       <start>/<end>
                   e.g.: -t 2007-12-31/2011-01-31

  -s[--start]      <starttime>
  -e[--end]        <endtime>
                   e.g.: -s 2007-12-31 -e 2011-01-31


* specify a minimum and maximum magnitude:

  Default:         minimum magnitude 3, no maximum magnitude.
  Format:          Integer or decimal.

  -m[--magmin]     <min.magnitude>
  -M[--magmax]     <max.magnitude>
                   e.g.: -m 4.2 -M 9


* specify a station restriction:

  Default:         no constraints.
  Format:          Any station code, may include wildcards.
  -i[--identity]    <nw>.<st>.<l>.<ch>
                    e.g. -i IU.ANMO.00.BH* or -i *.*.?0.BHZ

  -N[--network]     <network>
  -S[--station]     <station>
  -L[--location]    <location>
  -C[--channel]     <channel>
                    e.g. -N IU -S ANMO -L 00 -C BH*


* specify plotting options:

  Default:          no plot. If the plot will be created with -I d (or -I
                    default), the defaults are 1200x800x1/100 and the default
                    phases to plot are 'P' and 'S'.

  -I[--plot]        <pxHeight>x<pxWidth>x<colWidth>[/<timespan>]
                    For each event, create one plot with the data from all
                    stations together with theoretical arrival times. You may
                    provide the internal plotting resolution: e.g. -I 900x600x5.
                    This gives you a resolution of 900x600, and 5 units broad
                    station columns. If -I d, or -I default, the default of
                    1200x800x1 will be used. If this command line parameter is
                    not passed to ObsPyLoad at all, no plots will be created. You
                    may additionally specify the timespan of the plot after event
                    origin time in minutes: e.g. for timespan lasting 30 minutes:
                    -I 1200x800x1/30 (or -I d/30). The default timespan is 100
                    minutes. The final output file will be in pdf format.

  -F[--fill-
  plot]
                    When creating the plot, download all the data needed to fill
                    the rectangular area of the plot. Note: depending on your
                    options, this will approximately double the data download
                    volume (but you'll end up with nicer plots ;-)).

  -a[--phases]      <phase1>,<phase2>,...
                    Specify phases for which the theoretical arrival times should
                    be plotted on top if creating the data plot(see above, -I
                    option). Default: -a P,S. To plot all available phases, use
                    -a all. If you just want to plot the data and no phases, use
                    -a none.
                    Available phases:
                    P, P'P'ab, P'P'bc, P'P'df, PKKPab, PKKPbc, PKKPdf, PKKSab,
                    PKKSbc, PKKSdf, PKPab, PKPbc, PKPdf, PKPdiff, PKSab, PKSbc,
                    PKSdf, PKiKP, PP, PS, PcP, PcS, Pdiff, Pn, PnPn, PnS, S,
                    S'S'ac, S'S'df, SKKPab, SKKPbc, SKKPdf, SKKSac, SKKSdf,
                    SKPab, SKPbc, SKPdf, SKSac, SKSdf, SKiKP, SP, SPg, SPn, SS,
                    ScP, ScS, Sdiff, Sn, SnSn, pP, pPKPab, pPKPbc, pPKPdf,
                    pPKPdiff, pPKiKP, pPdiff, pPn, pS, pSKSac, pSKSdf, pSdiff,
                    sP, sPKPab, sPKPbc, sPKPdf, sPKPdiff, sPKiKP, sPb, sPdiff,
                    sPg, sPn, sS, sSKSac, sSKSdf, sSdiff, sSn
                    Note: if you select phases with ticks(') in the phase name,
                    don't forget to use quotes (-a "phase1',phase2") to avoid
                    unintended behaviour.


* specify additional options:
   -n[--no-
   temporary]
                    Instead of downloading both temporary and permanent networks
                    (default), download only permanent ones.

   -p[--preset]     <preset>
                    Time parameter given in seconds which determines how close
                    the data will be cropped before estimated arrival time at
                    each individual station. Default: 5 minutes.

   -o[--offset]     <offset>
                    Time parameter given in seconds which determines how close
                    the data will be cropped after estimated arrival time at each
                    individual station. Default: 80 minutes.

   -q[--query-
   resp]
                    Instead of downloading seismic data, download instrument
                    response files.

   -P[--datapath]   <datapath>
                    Specify a different datapath, do not use do default one.

   -R[--reset]
                    If the datapath is found, do not resume previous downloads as
                    is the default behaviour, but redownload everything. Same as
                    deleting the datapath before running ObsPyLoad.

   -u[--update]
                    Update the event database if ObsPyLoad runs on the same
                    directory for a second time.

   -f[--force]
                    Skip working directory warning (auto-confirm folder
                    creation).

 Type obspyload.py -h for a list of all long and short options.


 Examples:
 ---------

   Alps region,     obspyload.py -r 5/16.5/45.75/48 -t
   minimum          2007-01-13T08:24:00/2011-02-25T22:41:00 -m 4.2
   magnitude of
   4.2:

   Sumatra          obspyload.py -r 90/108/-7/7 -t "2004-12-24
   region,          01:23:45/2004-12-26 12:34:56" -m 9
   Christmas
   2004,
   different
   timestring,
   mind the
   quotation
   marks:
```

```
Mount            obspyload.py -r 12.8/12.9/47.72/47.77 -t
Hochstaufen      2001-01-01/2011-02-28
area(Ger/Aus),
default
minimum
magnitude:

Only one         obspyload.py -s 2011-03-01 -m 9 -I 400x300x3 -f -i IU.YSS.*.*
station, to
quickly try
out the plot:

ArcLink          obspyload.py -N B? -S FURT -f
Network
wildcard
search:

Downloading      obspyload.py -q -f -P metacatalog
metadata from
all available
stations to
folder
"metacatalog":

Download         obspyload.py -E -P thisOrderHadExceptions -f
stations that
failed last
time (not
necessary to
re-enter the
event/station
restrictions):
```

1.2 Specifying a geographical rectangle

As can be seen above, to clearly lay out different ways of usage, this help function is structured thematically. For example, if the user wants to have a look at all events that happened (roughly) in the Japan region, these two commands are equivalent:

```
$ obspyload.py -x 125 -X 150 -y 30 -Y 48
$ obspyload.py -r 125/150/30/48
```

1.3 Specifying a time frame

If the user wants to download all events that occurred in the time from February 6th to May 20th 2011, he can achieve it in these two ways:

```
$ obspyload.py -s 2011-02-06 -e 2011-05-20
$ obspyload.py -t 2011-02-06/2011-05-20
```

There are many more ways to provide a time string. As stated in the help function printed above, any *obspy.core.UTCDateTime* recognizable string can be given. The interested user may have a look at the documentation of this module, since it is not appropriate to echo all of its examples in this thesis. Here is just an incomplete list of selected ways of usage:

- ISO8601 string, calendar date: *-s 20091231T122334.5 -e 2009-12-31T12:23:34+01:15*
- ISO8601, ordinal date, two ways: *-s 2009-365T12:23:34.5 -e 2009365T122334.5*
- ISO8601, week date: *-s 2009-W53-7T12:23:34.5*
- other string: *-t 1985-12-02 12:23:34/2007-08-22 13:37:13*

Since the default *end time* is always now, to download as many events as possible it is sufficient to change the *start time* from the default value of *3 months ago* further into the past, e.g. by specifying *-s 1970-001*.

1.4 Further options

As can be seen in section 1.1, there are numerous other options available, adding capabilities to restrict event magnitudes (*-m* (minimum magnitude) and *-M* (maximum magnitude)), to restrict stations and networks (*-i, -N, -S, -L, -C*) and to add a plot of all station data and theoretical arrival time for each event(*-I, -F, -a*).

1.5 Combining options

Options and restrictions can be combined in any arbitrary way. For instance, if the user wants to download all events that occurred in the Japan region in the time from February 6th to May 20th 2011, he can use one of these commands:

```
$ obspyload.py -s 2011-02-06 -e 2011-05-20 -x 125 -X 150 -y 30 -Y 48
$ obspyload.py -r 125/150/30/48 -t 2011-02-06/2011-05-20
```

Figure 1 has been created to provide an optical guideline to the process of quickly finding a complete *ObsPy-Load* command without forgetting any options for a particular task. It should be useful to both regular and casual users of this tool.

1.6 A listing of all possible options: OptionParser help function

A list of all possible options, including customized help texts, is generated by the *OptionParser* module and can be accessed via

```
$ obspyload.py -h
```

Since the output of this command overlaps with the custom written long help function, it is not necessary to include it in this chapter. It can be found in Section 5.

2 Output

2.1 Shell output

2.1.1 Data download mode

When entering the data download procedure, the output of the script to the shell looks similar to the one seen below. The structure varies a bit depending on whether the plot option is used and whether extra data is downloaded to fill the plot area.

```
chris@gauss:~/data$ obspyload.py -f -m 9 -i BW.WETR.*.BH* -I d
Keypress capture thread initialized...
Press 'q' at any time to finish the file in progress and quit.
Downloading NERIES eventlist... done.
Received 1 event(s) from NERIES.
Downloading ArcLink inventory data... done.
Received 3 channel(s) from ArcLink.
```

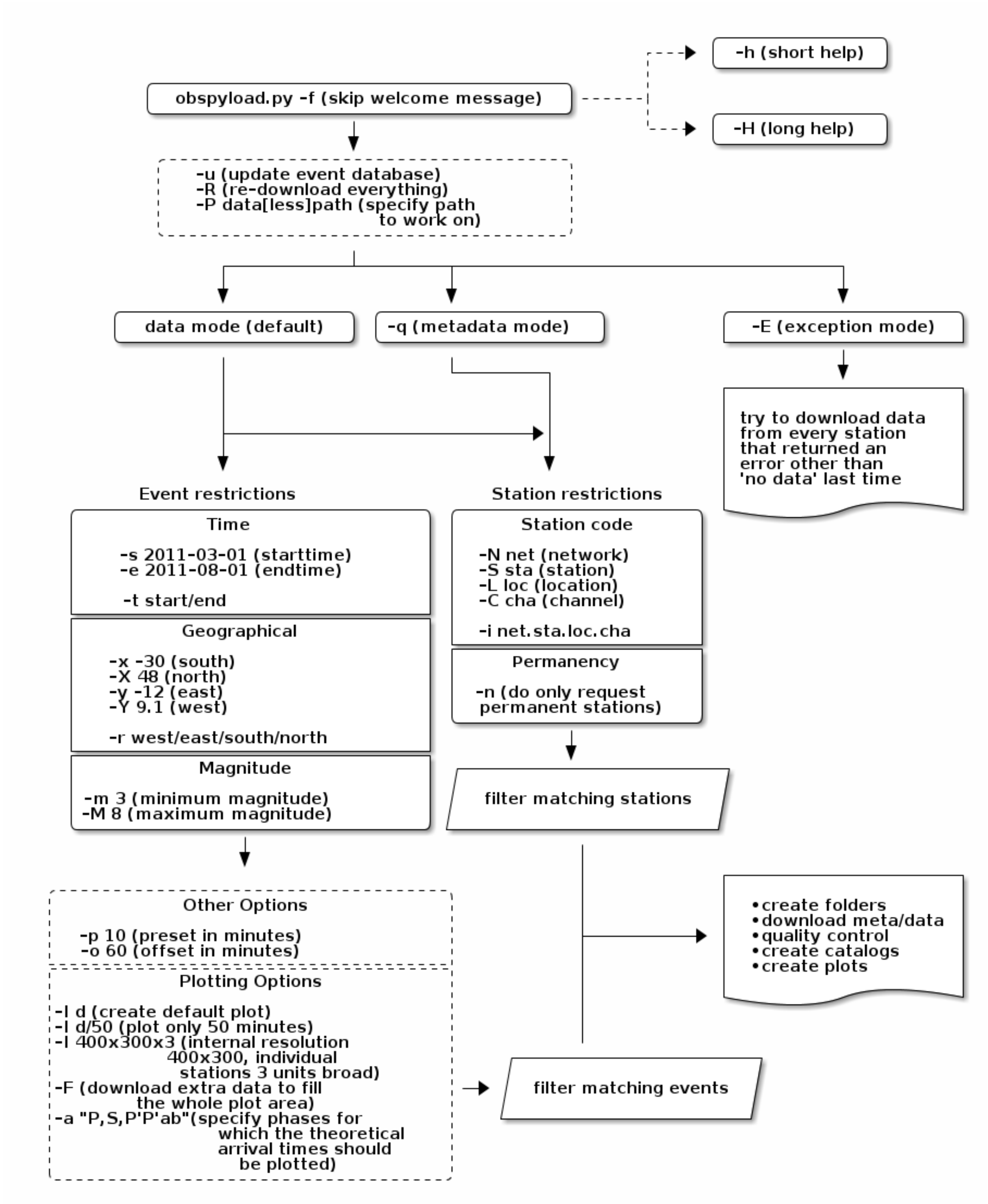
-h (short help)
obspyload.py -f (skip welcome message)
-H (long help)
-u (update event database)
-R (re-download everything)
-P data[less]path (specify path to work on)
data mode (default)
-q (metadata mode)
-E (exception mode)
try to download data from every station that returned an error other than 'no data' last time
Event restrictions
Time
-s 2011-03-01 (starttime)
-e 2011-08-01 (endtime)
-t start/end
Geographical
-x -30 (south)
-X 48 (north)
-y -12 (east)
-Y 9.1 (west)
-r west/east/south/north
Magnitude
-m 3 (minimum magnitude)
-M 8 (maximum magnitude)
Station restrictions
Station code
-N net (network)
-S sta (station)
-L loc (location)
-C cha (channel)
-i net.sta.loc.cha
Permanency
-n (do only request permanent stations)
filter matching stations
Other Options
-p 10 (preset in minutes)
-o 60 (offset in minutes)
Plotting Options
-I d (create default plot)
-I d/50 (plot only 50 minutes)
-I 400x300x3 (internal resolution 400x300, individual stations 3 units broad)
-F (download extra data to fill the whole plot area)
-a "P,S,P'P'ab"(specify phases for which the theoretical arrival times should be plotted)
filter matching events
• create folders
• download meta/data
• quality control
• create catalogs
• create plots

Figure 1: obspyload.py command line parameters helper diagram

```
Downloading IRIS availability data...
IRIS returned to matching stations.
Downloading quakeml xml file for event 20110311_0000010... done.
Downloading event 20110311_0000010 from ArcLink BW.WETR..BHE... done.
Scaling data for station plot... done.
Downloading event 20110311_0000010 from ArcLink BW.WETR..BHN... done.
Scaling data for station plot... done.
Downloading event 20110311_0000010 from ArcLink BW.WETR..BHZ... done.
Scaling data for station plot... done.
Done with event 20110311_0000010, saving plots...
Downloaded 665707 bytes in 41 seconds.
Done, press any key to quit.
```

In this example, due to restricting to only one station and setting the minimum magnitude to 9, the number of events was (fortunately) limited to one and the runtime was very short.

2.1.2 Metadata download mode

The metadata download mode, which can be used with the *-q* command line option, will first use the *availability* webservice from *IRIS*, then the *inventory* webservice from *ArcLink*. After printing how many stations have been returned from each data provider, *resp* files are downloaded from *IRIS* and *Dataless SEED* files are downloaded from *ArcLink*.

```
chris@gauss:~/data$ obspyload.py -q -s 1970-001
ObsPyLoad will download resp and dataless seed instrument files and quit.

Keypress capture thread initialized...
Press 'q' at any time to finish the file in progress and quit.
Downloading IRIS availability data... done.
Parsing IRIS availability xml to obtain nw.st.lo.ch... done.
Received 10678 station(s) from IRIS.
Received 100622 channel(s) from IRIS.
Downloading ArcLink inventory data... done.
Received 10866 channel(s) from ArcLink.
Downloading Resp file for 3A.L002..HHE.resp from IRIS... done.
Downloading Resp file for 3A.L002..HHN.resp from IRIS... done.
Downloading Resp file for 3A.L002..HHZ.resp from IRIS... done.

(...)

Downloading dataless seed file for AI.ESPZ..BHN.seed from ArcLink... done.
Downloading dataless seed file for AI.ESPZ..BHZ.seed from ArcLink... done.
Downloading dataless seed file for AI.JUBA..BHE.seed from ArcLink... done.

(...)
```

2.2 Folder and data structure

2.2.1 Data download mode

When downloading waveform data, *ObsPyLoad* creates the data folder specified with the option *-P* (or creates the default folder *obspyload-data*). Inside this folder, event directories, the event catalog file *catalog.txt*, the exception file *exceptions.txt* and some temporary files can be found (see Figure 2).
The event catalog file *catalog.txt* contains a list of all events in plain text formated as following[1].

[1]The *catalog.txt* file as included here has been shortened by the columns *datetime* and *origin_id* to fit on this paper format at reasonable font size.

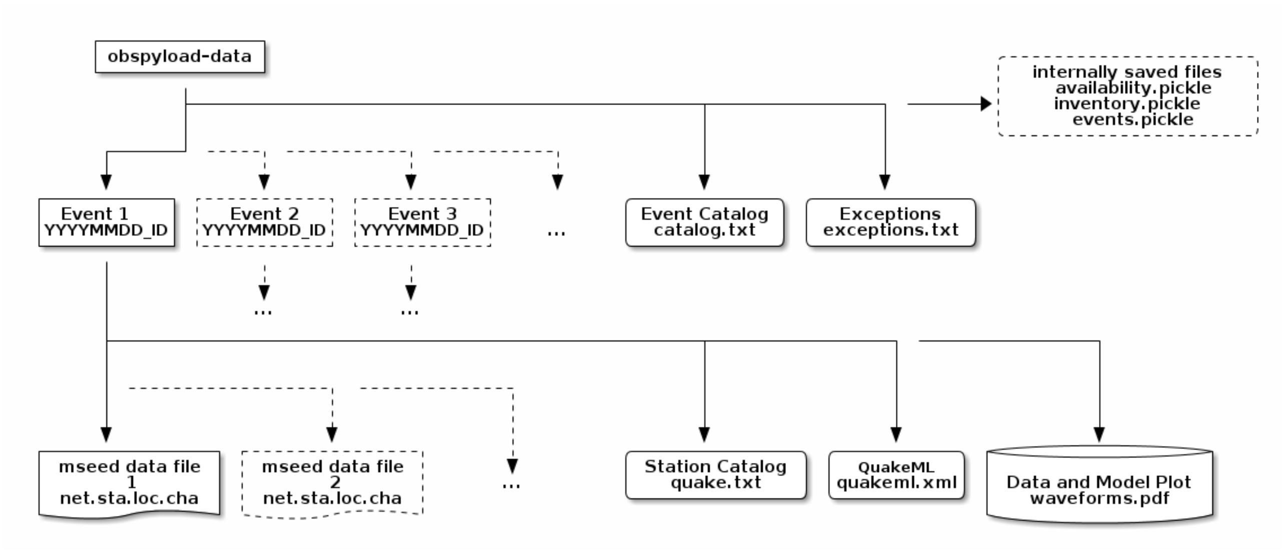

Figure 2: File structure inside the data folder

```
event_id;author;flynn_region;latitude;longitude;depth;magnitude;magnitude_type;DataQuality;TimingQualityMin
#############################################################################################################

20110409_0000021;CSEM;KYUSHU, JAPAN;30.023;131.764;-30.0;6.0;mw;0 (OK);100.00
20110411_0000023;CSEM;EASTERN HONSHU, JAPAN;36.998;140.452;-20.5;6.7;mw;0 (OK);100.00
20110411_0000078;CSEM;NEAR EAST COAST OF HONSHU, JAPAN;35.388;140.719;-5.0;6.4;mw;0 (OK);100.00
```

The last two columns of this file concern **quality control**. The column *Data Quality* uses the *obspy.mseed.libmseed.getDataQualityFlagsCount* method which counts all data quality flags of a MiniSEED file set by the digitizer (see the libmseed module documentation, `http://docs.obspy.org/packages/auto/obspy.mseed.libmseed.html`). The individual flags and MiniSEED files are not distinguished in this catalog, this number is merely a sum of all data quality flags of all MiniSEED data files for this event. Ideally it is zero, and *(OK)* will be written next to it. A large number in this field indicates quality problems in the data for this event, and *FAIL* will be written next to it.
The column *TimingQualityMin* is the lowest of all minima entries of the *timing quality information in Blockette 1001* in all files for this event. It ranges from *0* to *100*, with large values being better.
If this number is unsatisfactory, since this is the minimum of all files, it might be worth the time to have a look at the *quake.txt* file inside the concerning event folder. In this file, all stations and their minimum timing quality are listed separately, so it is possible to track the bad data and act as necessary.
The file *exceptions.txt*, which can also be found in the top-level of the data structure, provides a log of errors that occurred while downloading the data. Those may be errors like "*no data available*", which is fine, or connection problems like timeout errors, which is unfavorable.

```
event_id;data provider;station;starttime;endtime;exception
##########################################################

20110311_0000010;ArcLink;IU.ANTO.00.BHE;2011-03-11T05:53:24.128784Z;2011-03-11T07:18:24.128784Z;No data available
20110311_0000010;ArcLink;IU.ANTO.00.BHN;2011-03-11T05:53:24.128784Z;2011-03-11T07:18:24.128784Z;No data available
20110311_0000010;IRIS;IU.AFI.20.HN1;2011-03-11T05:52:17.247314Z;2011-03-11T07:17:17.247314Z;No waveform data available
```

If this file reveals an unpleasant amount of connection problems, it may be worth to try again to download the missing data with the *exception file mode* (*-E*). When continuing an interrupted download, this file is used to skip former event/station combinations that resulted in an exception.
The temporary files (*availability.pickle*, *inventory.pickle*, *events.pickle*) are files saved as byte stream and therefore of no immediate use outside the program.

Inside each event-folder, which is named according to the *NERIES* event id, all the data *MiniSEED* files for this event are saved. The file *quake.txt* first contains a header depicting once more the event info line as seen in the *event catalog* (see above). Following this, a catalog of all station channels from which data has been downloaded can be found (see the provided sample below). It also features the data provider and three columns of *quality control information*.

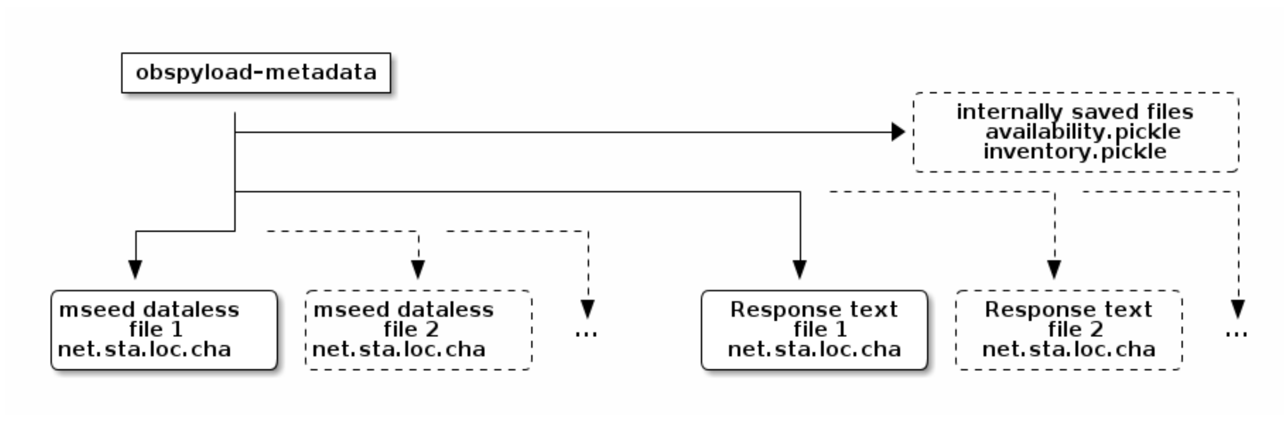
obspyload-metadata
internally saved files
availability.pickle
inventory.pickle
mseed dataless file 1 net.sta.loc.cha
mseed dataless file 2 net.sta.loc.cha
...
Response text file 1 net.sta.loc.cha
Response text file 2 net.sta.loc.cha
...

Figure 3: File structure inside the metadata folder

```
Station;Data Provider;TQ min;Gaps;Overlaps
###########################################

IU.SFJD.10.BHZ;ArcLink;None;0;0
IU.AFI.00.BHZ;IRIS;100.0;0;0
IU.ANMO.00.BHZ;IRIS;80.0;0;0
IU.CASY.10.BHZ;IRIS;85.0;0;0
IU.CHTO.00.BH1;IRIS;0.0;1;0
IU.HKT.00.BHZ;IRIS;45.0;0;0
```

The column *TQ min* is the minimum timing quality as described above for the *catalog.txt* file, but for each station separately. If no timing quality is available for a particular station, *None* will be printed in that field. The column *Gaps* is the number of gaps found in the data, whereas *Overlaps* counts the number of overlaps occurring in the MiniSEED file.
For each event, the relevant *QuakeML* xml file is saved as *quakeml.xml*.
If the program has been instructed to create a plot for each event, those will be contained inside the event folders as *waveforms.pdf*. Additionally, a stack of all events will be included in the top level data directory.

2.2.2 Metadata download mode

Unless specified otherwise, the metadata download mode creates the folder *obspyload-metadata*. Inside, besides some internal saved files, all downloaded dataless MiniSEED files and instrument response files are contained at top level (see Figure 3). Their corresponding file extensions are *.mseed* and *.resp*.

3 Examples

3.1 Strong Events around the 2011 Tohoku earthquake

Just before the development of *ObsPyLoad*, the 2011 Tohoku earthquake in Japan with a magnitude of 9.0 M_w caused tremendous tragedy. Before and after this event, some strong earthquakes (M_w > 7) occurred in the same area. For these events, this example will download all broadband vertical-component data (*BHZ*) solely from stations for which the device is set to *00*, both from the *IRIS* and *ArcLink* webservices.

```
$ obspyload.py -P tohoku2011 -m 7 -t 2011-02-27/2011-04-19 -r 140/146/30/42
                                 -i *.*.00.BHZ -I 1200x800x5/60 -a P,S,PP
```

The resulting folder *tohoku2011* can be found on the accompanying CD inside the folder *examples*. The command as seen above produced plots with a time frame of 60 minutes, a station column width of 3 and plotted theoretical arrival times of the *P, S, PP* wave phases on top. The example of Figure 4 shows the devastating earthquake that occurred on March 11th, 2011.

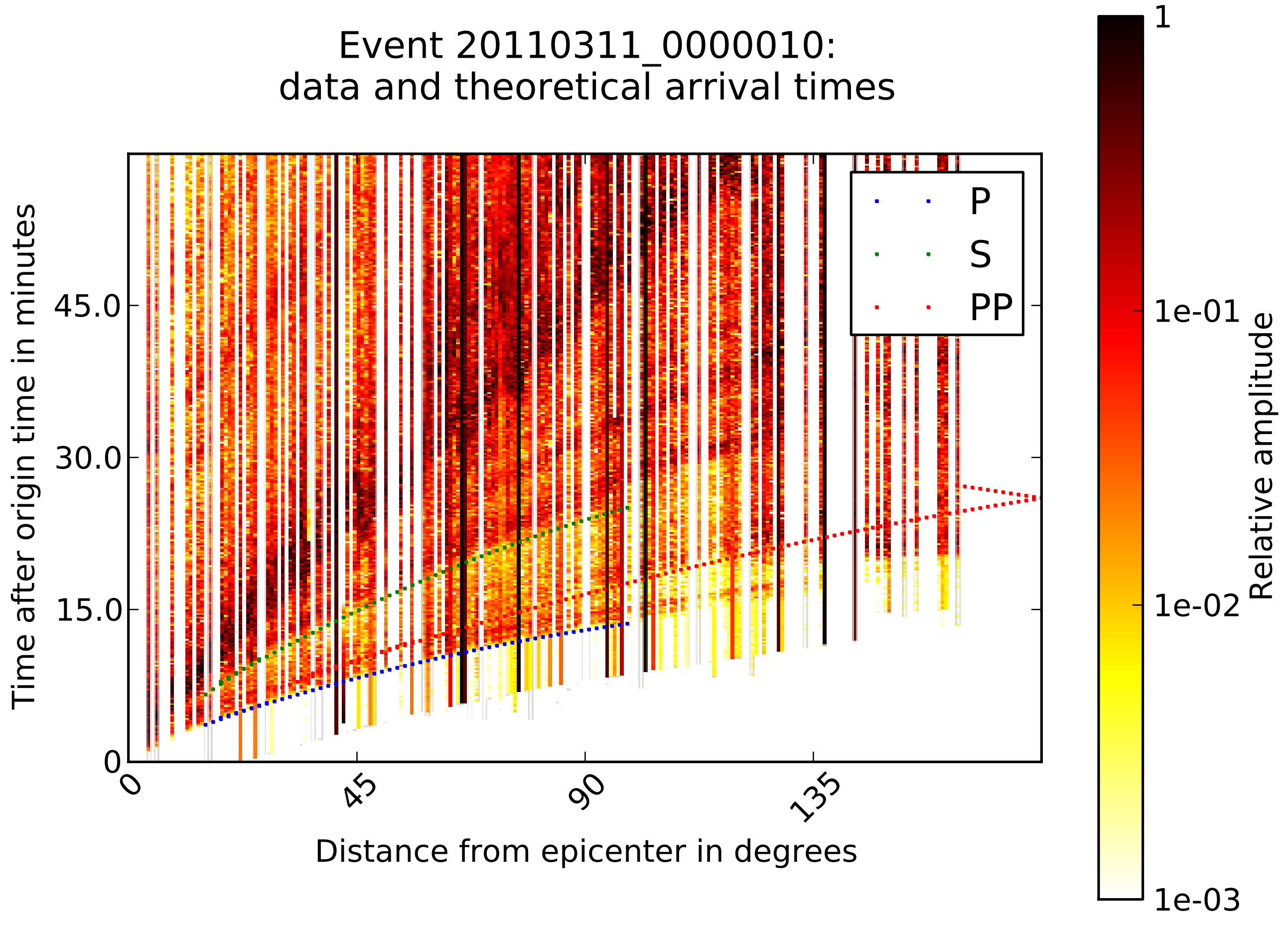

Figure 4: Stacked waveform data of 202 stations and theoretical arrival times for the M_w9.0 earthquake off the coast of Japan on March 11th, 2011.

3.2 Building up a metadata database

It might be an interest of many seismologists to build up a central metadata database. This can be achieved by

```
$ obspyload.py -q -P metadata_db -s 1970-001
```

If this download (currently over 100000 files) is interrupted by pressing *"q"*, it can be resumed by running the same command again.
Running the same command again, supplemented by the *-u* (update) option, will add newly registered stations to the database. Unfortunately, it is not possible to update existing files inside a database with solely internal methods of *ObsPyLoad*. A possible stopgap measure might be to always add new database folders for recent time-periods, or to download the whole database again at regular intervals.
To provide an example, the author downloaded instrument response and dataless seed files from only the *KBS* station of the *IU* network (additional command line options -N IU -S KBS). The result can be found inside the *examples* folder on the accompanying CD.

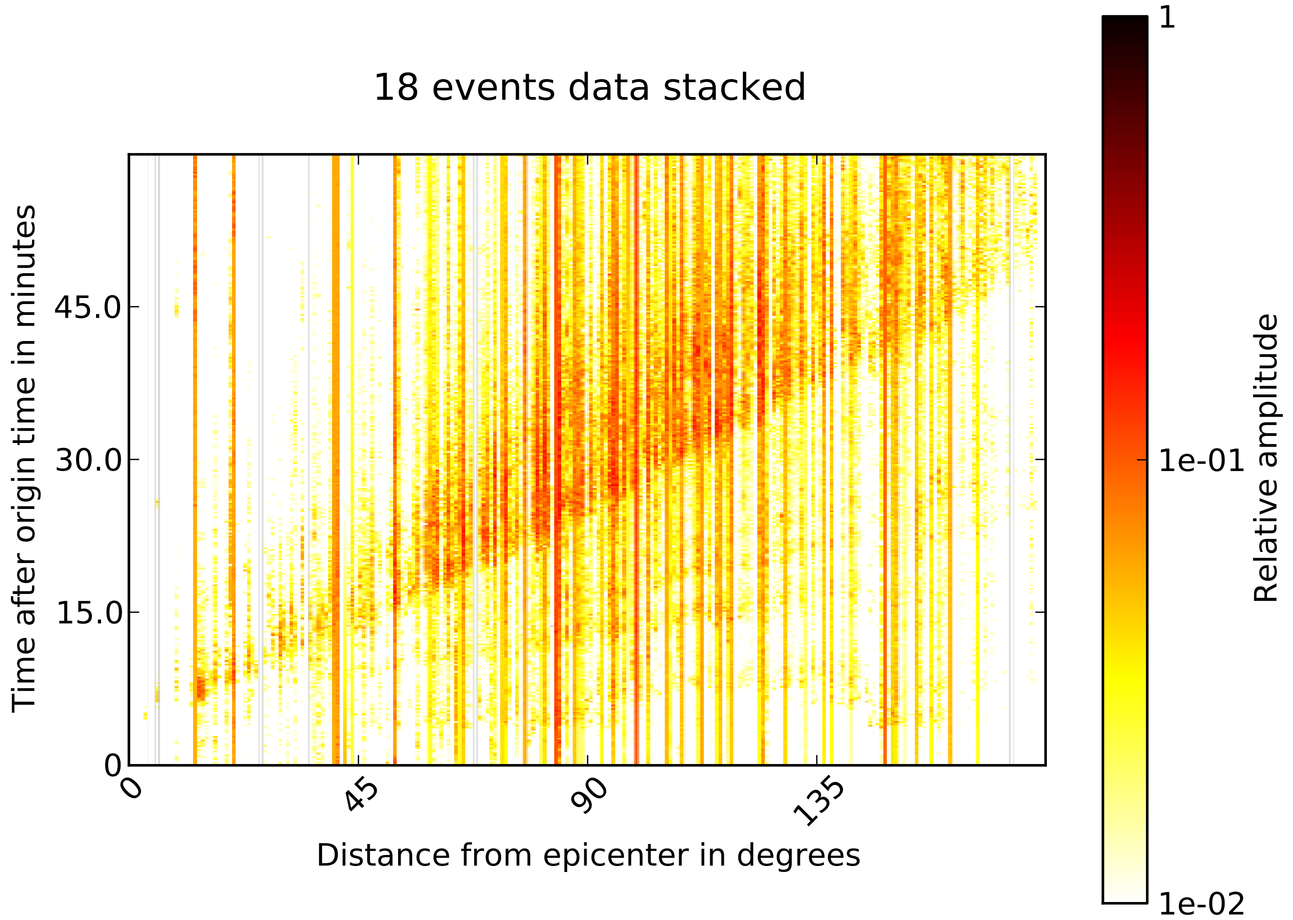

Figure 5: Another example plot filling the whole plotting area (*-F* option) without theoretical arrival times (*-a none* option) for 18 $M_w \geq 7.7$ events stacked (the stacked events plot is saved as file *allevents_waveforms.pdf* in the top level data directory). This is broadband vertical component data from solely the stations for which the device was set to *00*. The plot can be found in PDF format in the examples folder of the accompanying CD. The corresponding data folder is about 750 megabytes large and therefore not included on the CD.

4 Installation of ObsPyLoad

4.1 Dependencies

- Python (`http://www.python.org`)
- NumPy (`http://numpy.scipy.org`)
- SciPy (`http://scipy.org`)
- Matplotlib (`http://matplotlib.sourceforge.net`)
- lxml (`http://lxml.de`)
- ObsPy (`http://www.obspy.org`)

On most modern operating GNU/Linux operating systems, all of these dependencies except *ObsPy* may probably be installed with the package manager. If that is not favored, for most operating systems it should be simple to follow the following procedure.

- Download Python 2.6.x from `http://www.python.org/download/`. Uncompress the archive. For Windows users, an executable installer is provided.
- Run

```
./configure --prefix=$HOME
make
make install
export PATH="$HOME/bin:$PATH"
```

- Download *Easy Install* from `http://peak.telecommunity.com/dist/ez_setup.py`
- Run

```
python ez_setup.py
```

- Now use *easy_install* to install the required dependencies:

```
easy_install numpy
easy_install scipy
easy_install matplotlib
easy_install lxml
```

- Installing *ObsPy* can either also be done using *easy_install*:

```
easy_install -N obspy.core
easy_install -N obspy.mseed
easy_install -N obspy.sac
easy_install -N obspy.gse2
easy_install -N obspy.imaging
easy_install -N obspy.signal
easy_install -N obspy.arclink
easy_install -N obspy.xseed
easy_install -N obspy.seishub
easy_install -N obspy.seisan
easy_install -N obspy.wav
easy_install -N obspy.fissures
easy_install -N obspy.sh
easy_install -N obspy.taup
```

- Or using SVN (subversion), which retrieves the latest version:

```
svn checkout https://svn.obspy.org/trunk obspy
cd obspy/misc/scripts/
./develop.sh
```

4.2 ObsPyLoad

Finally, *ObsPyLoad* can either be retrieved from the supplemented CD, or using SVN:

```
svn checkout https://svn.obspy.org/branches/scheingraber obspyload
```

It may be convenient to use the tool without the full path, e.g. by creating a symlink

```
        ln -s /path/to/obspyload.py /usr/local/bin/obspyload.py
```

or an alias, e.g. for bash:

```
        echo "alias obspyload.py=\"/path/to/obspyload.py\"" >> ~/.bashrc
```

5 OptionParser help message

```
Usage: obspyload.py [options]

Options:
  -h, --help            show this help message and exit
  -H, --more-help       Show explanatory help and exit.
  -q, --query-metadata  Instead of downloading seismic data, download
                        metadata: resp instrument and dataless seed files.
  -P DATAPATH, --datapath=DATAPATH
                        The path where ObsPyLoad will store the data (default
                        is ./ObsPyLoad-data for the data download mode and
                        ./ObsPyLoad-metadata for metadata download mode).
  -u, --update          Update the event database when ObsPyLoad runs on the
                        same directory a second time in order to continue data
                        downloading.
  -R, --reset           If the datapath is found, do not resume previous
                        downloads as is the default behaviour, but redownload
                        everything. Same as deleting the datapath before
                        running ObsPyLoad.
  -s START, --starttime=START
                        Start time. Default: 3 months ago.
  -e END, --endtime=END
                        End time. Default: now.
  -t TIME, --time=TIME  Start and End Time delimited by a slash.
  -v MODEL, --velocity-model=MODEL
                        Velocity model for arrival time calculation used to
                        crop the data, either 'iasp91' or 'ak135'. Default:
                        'iasp91'.
  -p PRESET, --preset=PRESET
                        Time parameter in seconds which determines how close
                        the event data will be cropped before the calculated
                        arrival time. Default: 5 minutes.
  -o OFFSET, --offset=OFFSET
                        Time parameter in seconds which determines how close
                        the event data will be cropped after the calculated
                        arrival time. Default: 80 minutes.
  -m MAGMIN, --magmin=MAGMIN
                        Minimum magnitude. Default: 3
  -M MAGMAX, --magmax=MAGMAX
                        Maximum magnitude.
  -r RECT, --rect=RECT  Provide rectangle with GMT syntax:
                        <west>/<east>/<south>/<north> (alternative to -x -X -y
                        -Y).
  -x SOUTH, --latmin=SOUTH
                        Minimum latitude.
  -X NORTH, --latmax=NORTH
                        Maximum latitude.
  -y WEST, --lonmin=WEST
                        Minimum longitude.
  -Y EAST, --lonmax=EAST
```

```
                      Maximum longitude.
-i IDENTITY, --identity=IDENTITY
                      Identity code restriction, syntax: nw.st.l.ch
                      (alternative to -N -S -L -C).
-N NW, --network=NW   Network restriction.
-S ST, --station=ST   Station restriction.
-L LO, --location=LO  Location restriction.
-C CH, --channel=CH   Channel restriction.
-n, --no-temporary    Do not request all networks (default), but only
                      permanent ones.
-f, --force           Skip working directory warning.
-E, --exceptions      Instead entering the normal download procedure, read
                      the file exceptions.txt in the datapath, in which all
                      errors ObsPyLoad encountered while downloading are
                      saved. This mode will try to download the data from
                      every station that returned an error other than 'no
                      data available' last time.
-I PLT, --plot=PLT    For each event, create one plot with the data from all
                      stations together with theoretical arrival times. You
                      may provide the internal plotting resolution: e.g. -I
                      900x600x5. This gives you a resolution of 900x600, and
                      5 units broad station columns. If -I d, or -I default,
                      the default of 1200x800x1 will be used. If this
                      parameter is not passed to ObsPyLoad at all, no plots
                      will be created. You may additionally specify the
                      timespan of the plot after event origin time in
                      minutes: e.g. for timespan lasting 30 minutes: -I
                      1200x800x1/30 (or -I d/30). The default timespan is
                      100 minutes. The final output file will be in pdf
                      format.
-F, --fill-plot       When creating the plot, download all the data needed
                      to fill the rectangular area of the plot. Note:
                      depending on your options, this will approximately
                      double the data download volume (but you'll end up
                      with nicer plots ;-)).
-a PHASES, --phases=PHASES
                      Specify phases for which the theoretical arrival times
                      should be plotted on top if creating the data plot(see
                      above, -I option). Usage: -a phase1,phase2,(...).
                      Default: -a P,S. See the long help for available
                      phases. To plot all available phases, use -a all. If
                      you just want to plot the data and no phases, use -a
                      none.
-d, --debug           Show debugging information.
```