

ObsPyLoad: A Tool for Fully Automated Retrieval of Seismological Waveform Data

by Chris Scheingraber, Kasra Hosseini, Robert Barsch, and Karin Sigloch

Online Material: Installation instructions, additional figures.

INTRODUCTION

We confront the data avalanche: the amount of waveform data available from seismological data centers has been growing enormously over the past few years. This is a highly welcome development from a scientific point of view, but the time and effort spent on identification, retrieval, and quality control of subsets of these data may quickly exceed tolerable limits for an individual researcher.

Data from different data centers may not be available through the same interfaces or may arrive in different formats, which have tended to change over time. This often results in time-consuming homogenization efforts. The situation has improved in that certain quasistandards have been adopted for data formats; for example SEED, the Standard for the Exchange of Earthquake Data (IRIS Consortium, 1993), has been in use for nearly 20 years. Also, a few Internet data exchange protocols are in wide use now; for example ArcLink, developed within the WebDC project of GFZ and SZGRF (<http://www.gfz-potsdam.de>, <http://webdc.eu>), or DHI, a framework for accessing data and metadata from the IRIS DMC using a DHI-supporting client program (Ahern, 2001).

ObsPyLoad is a software tool that fully automatically queries the metadata holdings of seismological data centers and retrieves all metadata and seismograms of matching attributes via the Internet. Waveforms and metadata can be retrieved from multiple data centers, which is a major advantage over the download tools offered by individual data centers. We currently support downloads from IRIS (via a webservice client), and from ORFEUS (via ArcLink). We expect to add more choices as they become available in the ObsPy framework (see The Choice of Python). Earthquake metadata may currently be retrieved from the Seismic Portal (<http://www.seismicportal.eu>).

ObsPyLoad can either run in standalone mode from a command-line interface, or be integrated into other Python code as a module. A simple call to `obsypyload.py` without any parameters will download event-based data for all events that occurred over the past 30 days. Command-line options

allow for extensive customization (e.g., selection of certain geographical source or receiver regions; time windowing; thresholding by event magnitude; retrieval of metadata only; attempts to update an existing data set or reattempt a download for which problems occurred earlier).

Waveforms may be retrieved either by specifying a time-window, which is similar to BREQ_FAST (http://www.iris.edu/manuals/breq_fast.htm), a formatted mail-based service of the IRIS DMC, or by specifying earthquake characteristics “event-based,” similar to SOD (Owens *et al.*, 2004) or WILBER (<http://www.iris.edu/dms/wilber.htm>). In the latter case, ObsPyLoad first queries the earthquake catalog of the seismic portal, selects the matching event(s), calculates the appropriate time windows, and finally downloads the seismograms that recorded the event(s).

The main inspiration for ObsPyLoad has been the “Standing Order for Data” (SOD) software by Owens *et al.* (2004). Like SOD, we strive for a maximum of automation in order to minimize user intervention time. While SOD and other existing data retrieval tools are standalone applications, ObsPyLoad is written as a module of the high level, general purpose programming language Python. As such, it may either be used as a standalone application from the command line, or it may be seamlessly integrated into existing Python code via a function call. Here we primarily describe usage of ObsPyLoad as a standalone program, which likely appeals to a larger user group and requires no knowledge of Python. The © electronic supplement briefly describes usage by function call, which invokes parameters analogous to command-line options, the use case relevant to Python programmers.

In either case, the user (or calling function) only interacts with ObsPyLoad at call time, after which all metadata and waveforms are retrieved fully automatically via the ArcLink protocol or the IRIS DMC Web Services. Waveforms are sorted into a local directory structure on the user’s machine and are saved in MiniSEED format. Metadata and instrument responses are saved as plain ASCII text files. Extensive diagnostic and plotting output may be generated. There is no further workload on the user through e-mail messages, unlike BREQ_FAST or IGeoS (Morozov and Smithson, 1997; Chubak and Morozov, 2006). ObsPyLoad also eliminates FTP downloads, unlike WILBER or BREQ_FAST. Unlike FetchResp (<http://www.iris.edu/manuals/fetchresp.htm>) no configuration file needs to be edited. ObsPyLoad does not yet provide a graphical user interface (GUI) for map-based, graphical queries, as do VASE and JWEED (<http://www.iris.edu/manuals/vase/> and <http://www.iris.edu/manuals/jweed/>). However, both of these

tools require more user interaction and do not support the retrieval of ORFEUS data via the ArcLink protocol, which is one of ObsPyLoad's major advantages.

ObsPyLoad embraces another central design goal of SOD, which is that the update of local data holdings should be easy and automated. For a given earthquake, seismograms from certain stations may only become available months or years after an initial retrieval (e.g., embargoed PASSCAL experiments). A new call to ObsPyLoad will identify and retrieve only those newly available seismograms, plus any seismograms that previously resulted in download errors, such as those due to network failures. In short, ObsPyLoad wants to be an autonomous and persistent helper for retrieving passive seismological data from multiple data centers in a highly customizable manner.

Quick Tour section demonstrates the most important functionality of ObsPyLoad in a small example. Software Functionality section describes the complete software functionality, including input, output, and plotting options. Field Test: Retrieving a Voluminous Data Set from IRIS and ORFEUS section reports the results of a demanding field test. We downloaded a voluminous set of waveforms from the IRIS and ORFEUS data centers, which include all broadband seismograms for the first six months of 2010, for events of magnitude $M_w \geq 5.8$. We discuss performance in terms of download time, problems encountered, and how the software handled them. The ObsPyLoad software is part of the  electronic supplement and its most up-to-date version may be downloaded at obspy.org.

QUICK TOUR

Example: Downloading Seismograms of a Regional Earthquake Cluster

On 11 March 2011, the great Tohoku-Oki M_w 9.0 earthquake occurred offshore northeastern Japan. Within days before and after the mainshock, several more strong earthquakes of $M_w \geq 7$ occurred in the same area. The following call to ObsPyLoad will query the IRIS and ORFEUS data centers for earthquakes that fit the above description, download seismograms for all matching events, and generate a summary waveform plot for each event, as in Figure 2, which can be written as

```
$ obspyload.py -m 7.0 -t 2011-02-27/2011-04-16
-r 140/146/30/42 -i *.*.00.BHZ
-l 1200x800x5/60 -a P,S,PP -P tohoku_events.
```

Specification of Event and Receiver Characteristics

This example demonstrates a standalone call from the command line. The parameter `-m 7.0` requests a minimum event magnitude M_w of 7.0. The time window of relevant event origin times (27 February to 16 April 2011) is specified by `-t 2011-02-27/2011-04-16`, while the geographical source region is specified by `-r 140/146/30/42`, a rectangle from longitude 140° E to 146° E, and latitude 30° N to 42° N. Receivers

are characterized by a SEED identifier (format *net.sta.loc.cha*), here `-i *.*.00.BHZ`. The wildcards in the network and station fields result in no restrictions being applied, whereas the location ("hole") is restricted to `00`, and the channel is restricted to `BHZ` (broadband, vertical).

The parameter `-P` specifies the name of the top-level folder, here `tohoku_events`, to which all data and diagnostic output will be saved. The last two parameters activate plotting functionalities. The plot resolution (1200x800), seismogram line width (5 pixels), and timespan of the plot (60 min) are specified by `-l 1200x800x5/60`, and the phases for which to superimpose theoretical arrival-time estimates are given by `-a P,S,PP`. None of the above specifications are mandatory as ObsPyLoad provides reasonable default values for each (ObsPyLoad section).

Output

The downloaded data is automatically sorted into a folder structure as described in this paragraph. For a graphical representation, see  supplementary Figure S2. Inside the top-level directory, the file `events.txt` lists all events downloaded by this job, and the file `exceptions.txt` contains a list of all problems encountered. For each matching event, a subfolder is created, named after the respective *event_identifier* assigned by the Seismic Portal. This folder contains all seismograms in MiniSEED format, a waveform summary plot `waveforms.pdf` (Fig. 2), and a receiver metadata file `stations.txt`, listing all downloaded stations, as well as some quality control information. File `quakeml.xml` contains event metadata, such as the moment tensor, according to the *QuakeML* specification (Wyss *et al.*, 2004).

The user may also want to restrict station locations geographically. This can be done, for example, by specifying a small circle, using the `-l lon/lat/rmax` option. If we wanted only European recordings of the Japanese event cluster, we might specify a 20-degree radius around Munich at 48° N, 11° E as

```
$ obspyload.py -m 7.0 -t 2011-02-27/2011-04-16
-l 11/48/20
-r 140/146/30/42 -i *.*.00.BHZ
-l 1200x800x5/60 -a P,S,PP -P tohoku_events.
```

Update Mode

When new data become available or new events are added to the earthquake catalog, it is possible to quickly update an existing download folder by simply running the same command again, supplemented by the `-u` flag as

```
$ obspyload.py -m 7.0 -t 2011-02-27/2011-04-16
-u -l 11/48/20
-r 140/146/30/42 -i *.*.00.BHZ
-l 1200x800x5/60 -a P,S,PP -P tohoku_events.
```

It is not mandatory to specify the same restrictions as for the original download. By providing different parameters (e.g.,

loosening the station or event restriction criteria) it is possible to add data from events or stations that had previously not been considered.

SOFTWARE FUNCTIONALITY

ObsPyLoad provides an unified and automated access mechanism to various data providers. At this time, it can download data and metadata from the IRIS DMC and from the ORFEUS network as well as earthquake catalog information from the Seismic Portal.

To communicate with the ORFEUS data centers, we use the ArcLink distributed-data request protocol. It was founded by the WebDC initiative of the German GEOFON and Bundesanstalt für Geowissenschaften und Rohstoffe (BGR, Hanka and Kind, 1994; WebDC, 2011) and is suitable for downloading MiniSEED, Dataless SEED, and Full SEED files. Python and ObsPy provide high-level support for this because the ObsPy framework contains a module *obspy.arclink* that implements data download via the ArcLink protocol, and a module *obspy.iris*, which implements a client to the IRIS DMC Web Services. We download data in SEED format (Full SEED or MiniSEED), which is the most complete standardized format for exchanging seismograms because it can hold not just header information and the time series themselves, but also quality diagnostics for every time sample. If the user desires a different format, it is possible to convert the SEED volumes locally (e.g., using ObsPy).

The Choice of Python

ObsPyLoad is written in Python, a free and open-source object-orientated programming language (<http://www.python.org>). Its module-extensible structure, combined with dynamic typing and classes, results in a very natural and elegant syntax. One advantage of using Python for this project lies in powerful and readily available open-source libraries. ObsPyLoad builds on the following:

ObsPy (<http://obspy.org>), a framework for processing seismological data, is a free and open-source project (Beyreuther *et al.*, 2010). It provides a software standard sufficient for complete preprocessing workflows in seismology (Megies *et al.*, 2011, 53–55). Extensive documentation and tutorials are available on the project home page.

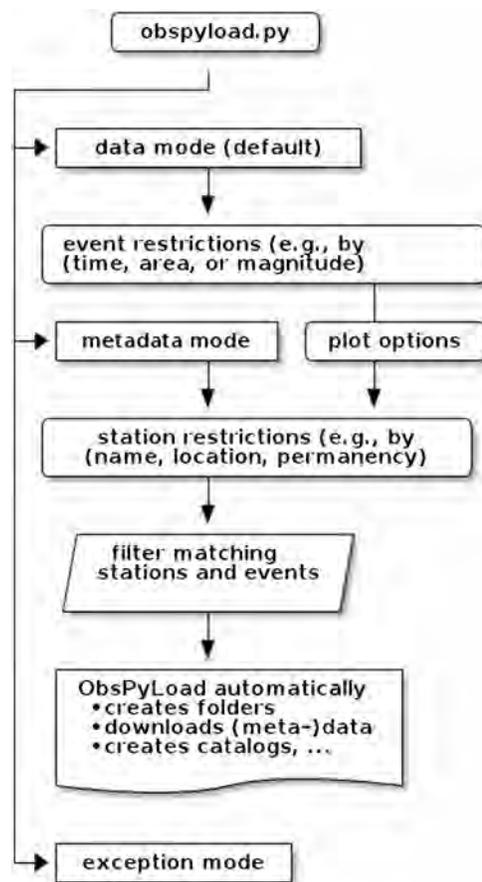
NumPy (<http://www.numpy.org/>) is an extension for performing numerical calculations on large arrays and matrices (Oliphant, 2007). It is mostly written in C and thus very fast.

SciPy (<http://www.scipy.org/>) is a library for advanced math, signal processing, or statistics. It relies on NumPy.

Matplotlib is a popular plotting package (Hunter, 2007).

Installation and System Requirements

Since ObsPyLoad relies on Python and ObsPy (including dependencies resulting from ObsPy modules), these environments need to be installed first. The ObsPy project homepage features easy and straightforward solutions. Instructions also are described in the ☺ electronic supplement.



▲ **Figure 1.** Schematic program flow of obspyload.py. A more detailed version, including all available command line options, is given by ☺ supplementary Figure S1.

ObsPyLoad

Here we discuss the functionality of ObsPyLoad (input parameters, work modes, and output options). Figure 1 gives a schematic overview of program flow; ☺ supplementary Figure S1 is more detailed, explaining available command-line options.

Input Parameters

ObsPyLoad can be used from a system shell without explicitly calling the Python interpreter first. For integration with other software, it may also be imported as a Python module.

The design inclination was to not require the user to specify options, and to provide reasonable defaults for all. When no options are specified, the program prints an explanatory message asking the user to confirm downloading data for all events that happened globally during the past 30 days with a minimum magnitude of M_w 5.8. Seismograms from all available stations are then retrieved in time windows running from 5 minutes before to 80 minutes after estimated P-wave arrival. The *LASP91* velocity model (Kennett and Engdahl, 1991) will be used for travel-time calculations and the program will not generate a plot.

All defaults may be overridden separately. Most options can be specified in two different flavors: lower and upper bounds (for latitude, longitude, magnitude) may either be combined

into one option, separated by slashes or dots, or they may be specified separately. Consider these following alternatives for specifying a geographical rectangle:

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

The more verbose of the two available help functions gives full details on all options as follows

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

The output of this full help may be found in Section D of the  electronic supplement. Additionally, a listing of all possible options is generated and can be accessed via

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

The wide variety of available options add capabilities to restrict event magnitudes (minimum magnitude with `-m` and maximum magnitude with `-M`), restrict stations and networks (`-i`, `-N`, `-S`, `-L`, `-C`), and add a plot of all station data and theoretical arrival time for each event (`-I`, `-F`, `-a`).

Options exist to restrict lower- and upper-event magnitudes (`-m` and `-M`), stations and networks (`-i`, `-N`, `-S`, `-L`, `-C`) and to generate waveform plots, including theoretical arrival times (`-I`, `-F`, `-a`). Options may be combined arbitrarily. For instance, the following command retrieves waveform data from all $m_b \geq 6.0$ events that occurred since 01 January 2012 from all broadband channels at the Geophysical Observatory in Fürstfeldbruck, Germany. Note that ordinal date is used and that it suffices to provide the start time of the time window as

```
$ obspyload.py -m 6.0 -i GR.FUR.*.BH* -s 2012-001.
```

If one does not wish to obtain event-based data but rather download continuous streams recorded within a given time-frame, the `-w` option can be used. All other options may be given as usual as follows:

```
$ obspyload.py -w -s 2012-001 -e 2012-031 -N BW
```

This example downloads everything recorded by all stations in the BW network in January 2012.

Work modes

ObsPyLoad offers three basic work modes: data download mode (default, no option flag), metadata download mode (`-q`), and exception mode (`-E`).

Data download mode retrieves MiniSEED waveform data for all events and stations matching the given criteria. The data are stored locally into a predefined folder structure. Prior to download, ObsPyLoad informs about the number of matching events, stations, and channels for each data provider. Metadata download mode informs about the number of matching stations then retrieves the corresponding instrument response

files (*RESP* files). By default, they are saved to a folder *obspyload-metadata*, or a folder specified by `-P`.

Errors that occur in data or metadata mode are recorded in a log file (*exceptions.txt*). Exception mode reads this file and relaunches previously unsuccessful requests for waveforms or metadata.

Output and Plotting Options

Several options are available for creating plots. It is possible to plot the retrieved waveform data event by event. If data for more than one event are downloaded, a cumulative plot of all waveforms will be generated in the top-level directory of the project. All plots are saved in color, as pdf files. Default behavior of the plotting functionality is activated by `-I d` or `-I default`, resulting in a plot resolution of 1200×800 pixels, and seismograms of line width one pixel. These parameters may be customized as

```
$ obspyload.py -l 900×600×5
```

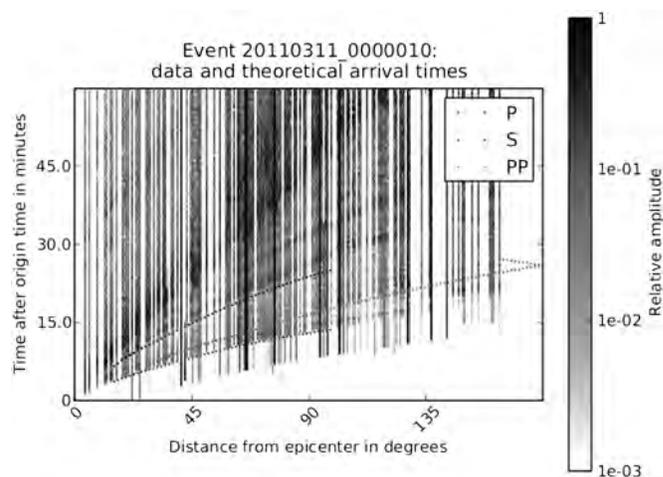
for a total plot area of 900×600 pixels and seismogram line width of five pixels.

The default time window for plotting is 80 minutes long, starting at event origin time. The following two calls change this interval to 60 minutes, as follows:

```
$ obspyload.py -l d/60
$ obspyload.py -l 900×600×5/60.
```

Theoretical arrival-time estimates may be added to the plots, using the `-a` option (Fig. 2), and written as,

```
$ obspyload.py -l d -a P,S,PP.
```



▲ **Figure 2.** A plot as automatically generated by ObsPyLoad, for the example call discussed in Quick Tour section. Each vertical line represents one of the 202 broadband seismograms that could be retrieved from IRIS or ORFEUS for the M_w 9.0 Tohoku-Oki mainshock on 11 March 2011. Theoretical arrival-time estimates for the P, PP, and S phases are overlain as dotted lines.

By default, theoretical arrival times of the P and S phases are overlain. To suppress this, specify `-a none`. If all possible phases should be plotted, use `-a all`. For a list of all available phases, refer to the [Ⓔ](#) electronic supplement or `obspyload.py -H`.

In order to generate pretty plots without white space, ObsPyLoad may be instructed to download longer time series than specified by the preset and offset options for the data job itself. This option is activated by `-F` and the result is shown in [Ⓔ](#) supplementary Figure S4. The command that was used to download the necessary data and generate this plot is shown below. The temporal and geographical restrictions are the same as in the first example given in Quick Tour section as,

```
$ obspyload.py -F -a none -m 7.7 -i *.*.00.BHZ
-t 2011-02-27/2011-04-16
-r 140/146/30/42.
```

To add plots to a pre-existing project folder, use the `-c` option as follows:

```
$ obspyload.py -c -l d/60 -a P,S,PP.
```

FIELD TEST: RETRIEVING A VOLUMINOUS DATA SET FROM IRIS AND ORFEUS

The performance of a program like ObsPyLoad can be measured in several ways: download rate, completeness of retrieved data, and minimization of user intervention time. ObsPyLoad delivers superior performance mainly in terms of user intervention time and its download rate and completeness are comparable to existing tools. As a comprehensive test, we ordered all broadband waveforms from the IRIS and ORFEUS data centers that met the following criteria: event time between 1 January and 30 June 2010 and event magnitude exceeding $M_w \geq 5.8$; start of time windows 5 min before estimated P-arrival, end 80 min after (default values).

At first glance, it may seem unnecessary to order a huge amount of data in order to test user interaction time. However, if a program stops in the middle of a job, this lack of robustness creates considerable additional demands on user time for troubleshooting. Hence our goal was to encounter every possible problem. A massive and nonselective order of data yields the most informative test.

Performance in Data Download Mode

Data download was launched by the command below, followed by the first few lines of ObsPyLoad screen output as,

```
$ obspyload.py -m 5.8 -i *.*.BH* -s 2010-01-01
-e 2010-06-30 -P datatest2010
```

```
Initializing new data-vs-time plot...
Keypress capture thread initialized...
Press 'q' at any time to quit.
Downloading NERIES eventlist... done.
```

```
Received 154 events(s) from NERIES.
Downloading ArcLink inventory data... done.
Received 1641 channel(s) from ArcLink.
Downloading IRIS availability data... done.
Parsing IRIS availability xml... done.
Received 1438 station(s) from IRIS.
Received 4674 channel(s) from IRIS.
Downloading quakeml file for event 20100102_0000016...
```

One hundred fifty-four suitable events were found in the earthquake catalogs of the Seismic Portal. Scheduled for download were 4674 recording broadband channels from the IRIS network and 1641 recording broadband channels from the ORFEUS network. The script then took about 45 days to download 162 GB of waveform data. The 690,503 MiniSEED files retrieved were automatically sorted into 154 event directories, named according to the Seismic Portal *event id*. Each directory was about 1.1 GB in size and contained around 4500 broadband waveform files from 1500 distinct seismic stations on average.

The test ran without errors. However, after 73 GB had been retrieved, the script slowed down considerably from a previously almost constant download rate of 60 kb/s to only a few kb/s. We could not exactly pinpoint the problem, but suspect that it may be due to ObsPyLoad reexamining all its previously written log files before deciding whether a seismogram needs to be downloaded or not. The slowed job was stopped and a second job was launched to download the remaining 89 GB, which finished at the normal rate. As expected, the download project resumed seamlessly after the deliberate interruption. Hence a pragmatic fix is to divide huge jobs into several mid-sized pieces from the start, but corrections underway posed no problem aside from the idle time lost. Download rates can be diagnosed by monitoring the plot *foldersize_vs_time.pdf* in the top-level directory of the project, which gets updated after every new event retrieved.

Tables 1 and 2 give download statistics for some major international networks. Of the 699,468 requests sent to IRIS,

Table 1
Download Statistics for Requests to the IRIS Data Management Center for Some Major International Seismic Networks

Network	Number of Requests	Number of Exceptions	% Retrieved
JP	2772	6	99.78
IM	3696	76	97.94
AU	20636	2010	90.26
G	10318	1456	85.89
IU	67144	9795	85.41
II	34342	5041	85.32
TA	236544	38341	83.79
Total	699468	131449	81.21

Table 2
Download Statistics for Requests to the ORFEUS Data Management Center Using the ArcLink Protocol for Some Major European Seismic Networks.*

Network	Number of Requests	Number of Exceptions	% Retrieved
GR	14322	1749	87.79
NO	7854	1233	84.30
GE	36960	6456	82.53
GB	10626	5444	48.77
CH	14784	9633	34.84
Total	251328	105478	58.03

*Exceptions do not imply that ObsPyLoad malfunctioned, but rather that the data could not be delivered by the data center, either because they were not available or because of routing problems.

81.21% were successful. From the ORFEUS data center, 58.03% of the 251,328 requests could be downloaded successfully. The term “exception” does not imply that the ObsPyLoad program malfunctioned. Rather, these exceptions are encountered and returned by the data server. The ones presented in Table 2 were mostly “no data” (51,782), “timeout” (30,285) and “no content” (16,496). Other exceptions occurred less frequently (e.g., “no route,” which was 1329). The exceptions “no data” and “no content” mean that there is no data available to retrieve. Even though we had a stable connection to the data center, the exception “timeout” sometimes arose. We suspect that this is due to internal routing problems. Unfortunately, the data servers do not provide more explicit information than these brief error codes so the exact reasons for failure to retrieve are not fully transparent.

Performance in Exception Mode

The file *exceptions.txt*, written to the top-level directory of each project, provides a log of errors that occurred during data download mode. Those may be irredeemable problem reports from the data centers, such as “no data available,” or connection problems such as timeout errors. Invoking ObsPyLoad in exception mode (`-E option`) launches a renewed attempt to retrieve only these previously problematic waveforms as

```
$ obspyload.py -E.
```

After the exception mode was finished, the project directory contained 690,534 MiniSEED files, compared to 690,503 before. Hence only very few requests that initially failed could be downloaded successfully in the second attempt. This suggests that the respective waveform data are really not present in the data servers.

Performance in Metadata Mode

The `-q` option was invoked to retrieve all instrument response files associated with the seismograms downloaded in Perfor-

mance in Data Download Mode section. The first few lines of ObsPyLoad output are also shown below:

```
$ obspyload.py -q -s 2010-01-01 -i *.*.*.BH*
ObsPyLoad will download resp instrumental files.
Keypress capture thread initialized...
Press 'q' at any time to quit.
Downloading IRIS availability data... done.
Parsing IRIS availability xml... done.
Received 2009 station(s) from IRIS.
Received 6745 channel(s) from IRIS.
Downloading ArcLink inventory data... done.
Received 1885 channel(s) from ArcLink.
Downloading Resp file for RESP.AC.BCI..BHE... done.
Downloading Resp file for RESP.AC.BCI..BHN... done.
Downloading Resp file for RESP.AC.BCI..BHZ...
```

The IRIS DMC reported holding 6745 channel responses (RESP files for BH* channels); for ORFEUS this number was 1885. The *RESP* files were downloaded into the default folder *obspyload-metadata*, because none was specified by the `-P` option. The metadata download finished within 3 h and retrieved 8045 unique channel response files from the IRIS and ORFEUS networks. This number is lower than the sum $6745 + 1885 = 8630$ because RESP files held by both providers were downloaded only once. ArcLink delivers the metadata in Dataless SEED format rather than RESP, which ObsPyLoad converts to RESP. The conversion failed for 63 invalid Dataless SEED files in which case the unconverted Dataless SEED files were retained. The final RESP database included response files for 99.67% of the seismograms retrieved in Performance in Data Download Mode section.

DISCUSSION AND CONCLUSION

Using ObsPyLoad for data retrieval has some advantages over currently available alternatives. Especially for large jobs, ObsPyLoad requires significantly less of a user’s time for launching and monitoring requests than e-mail-based request tools (e.g., BREQ_FAST, NetDC) or Web interfaces (e.g., WILBER). Download speed seems to be comparable (but generally not superior) to existing download tools. The limitations arise from communication overhead with the data centers. In the ArcLink protocol, a new connection must be established for every individual seismogram. Until recently, communication with the IRIS DMC had the same limitation, but now supports batch orders.

However, the user intervention time, often a much more penalizing bottleneck than download speed, is greatly reduced for ObsPyLoad users. Waveforms are downloaded directly to a user’s computer, rather than requiring separate retrieval by FTP. A field test in which we requested almost 700,000 seismograms (154 earthquakes, data volume 162 GB) from IRIS and ORFEUS took 45 days to complete, but it did not require any user interaction once the request, a one-liner,

had been launched. There may be use cases in which the highly customizable SOD software by Owens *et al.* (2004) would be better suited. SOD may be configured very extensively through XML files. For ObsPyLoad, the design emphasis has been on ease of use and versatility (callable either in standalone mode or as a function in a longer Python program).

A major advantage of ObsPyLoad over other tools is that it supports downloads from more than one data center (currently IRIS and ORFEUS, which use different data exchange protocols). A focus will be to build on this strength and to support additional data centers in the future. A GUI could further increase functionality. The user could select geographical restrictions interactively on a map showing events and stations and receive direct feedback about the selected events before submitting the download job. In cases in which a specific region is of interest, an interactive ray-coverage (metadata) plot could be considered in order to modify the request until coverage is sufficient.

However, we have shown that ObsPyLoad's first-order performance specification, which is fully automated seismological data retrieval, either based on time window or on event specifications, has already been robustly achieved by the command-line version presented here. The ObsPyLoad software is part of the  electronic supplement, but preferably it should be downloaded at obspsy.org, where it will be maintained and developed further. 

ACKNOWLEDGMENTS

We thank Lion Krischer for his contributions to the `obspsy.taup` package, which was created for this project. Karin Sigloch thanks Philip Crotwell for his responsiveness to all questions concerning SOD. We thank an anonymous reviewer and the Associate Editor, whose constructive feedback considerably improved the manuscript.

REFERENCES

Ahern, T. (2001). Data Handling Infrastructure at the IRIS DMC, *IRIS DMC Newsl.*, **3**, no. 1, <http://www.iris.edu/dms/newsletter/vol3/no1/data-handling-infrastructure-at-the-iris-dmc/>.

- Beyreuther, M., R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann (2010). ObsPy: A Python toolbox for seismology, *Seismol. Res. Lett.* **81**, 530.
- Chubak, G., and I. Morozov (2006). Integrated software framework for processing of geophysical data, *Comput. Geosci.* **32**, 767–775.
- Hanka, W., and R. Kind (1994). The GEOFON program, *Ann. Geophys.* **37**, 1060–1065.
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment, *Comput. Sci. Eng.* **9**, no. 3, 90–95.
- IRIS Consortium (1993). Standard for the Exchange of Earthquake Data Reference Manual.
- Kennett, B. L. N., and E. R. Engdahl (1991). Traveltimes for global earthquake location and phase identification, *Geophys. J. Int.* **105**, 429–465.
- Megies, T., M. Beyreuther, R. Barsch, L. Krischer, and J. Wassermann (2011). ObsPy—What can it do for data centers and observatories? *Ann. Geophys.* **54**, 47–58.
- Morozov, I. B., and S. B. Smithson (1997). A new system for multicomponent seismic processing, *Comput. Geosci.* **23**, 689–696.
- Oliphant, T. E. (2007). Python for scientific computing, *Comput. Sci. Eng.* **9**, 10–20.
- Owens, T. J., H. P. Crotwell, C. Groves, and P. Oliver-Paul (2004). SOD: Standing order for data, *Seismol. Res. Lett.* **75**, 515.
- WebDC (2011). European Integrated Data Archives—EIDA, www.webdc.eu, 18th July 2011.
- Wyss, A., D. Schorlemmer, S. Maraini, M. Baer, and S. Wiemer (2004). QuakeML—An XML schema for seismology, in *AGU Fall Meeting Abstracts*, Vol. 1, 0272.

Chris Scheingraber

Kasra Hosseini

Karin Sigloch

Department of Earth and Environmental Sciences

Ludwig-Maximilians-Universität München

Theresienstrasse 41

80333 Munich, Germany

chris@scheingraber.net

hosseini@geophysik.uni-muenchen.de

sigloch@geophysik.uni-muenchen.de

Robert Barsch

EGU Executive Office

Luisenstr. 37

80333 Munich, Germany

barsch@egu.eu