



ObspyDMT: a Python toolbox for retrieving and processing large seismological data sets

Kasra Hosseini, Karin Sigloch

► To cite this version:

Kasra Hosseini, Karin Sigloch. ObspyDMT: a Python toolbox for retrieving and processing large seismological data sets. Solid Earth, 2017, 8 (5), pp.1047-1070. 10.5194/se-8-1047-2017 . hal-03425560

HAL Id: hal-03425560

<https://hal.science/hal-03425560>

Submitted on 30 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ObspyDMT: a Python toolbox for retrieving and processing large seismological data sets

Kasra Hosseini^{1,2} and Karin Sigloch¹

¹Dept. of Earth Sciences, University of Oxford, South Parks Road, Oxford, OX1 3AN, UK

²Dept. of Earth Sciences, Ludwig-Maximilians-Universität München, Theresienstrasse 41, 80333 Munich, Germany

Correspondence to: Kasra Hosseini (kasra.hosseinizad@earth.ox.ac.uk)

Received: 2 May 2017 – Discussion started: 9 May 2017

Revised: 25 July 2017 – Accepted: 26 July 2017 – Published: 12 October 2017

Abstract. We present obspyDMT, a free, open-source software toolbox for the query, retrieval, processing and management of seismological data sets, including very large, heterogeneous and/or dynamically growing ones. ObspyDMT simplifies and speeds up user interaction with data centers, in more versatile ways than existing tools. The user is shielded from the complexities of interacting with different data centers and data exchange protocols and is provided with powerful diagnostic and plotting tools to check the retrieved data and metadata. While primarily a productivity tool for research seismologists and observatories, easy-to-use syntax and plotting functionality also make obspyDMT an effective teaching aid. Written in the Python programming language, it can be used as a stand-alone command-line tool (requiring no knowledge of Python) or can be integrated as a module with other Python codes. It facilitates data archiving, preprocessing, instrument correction and quality control – routine but nontrivial tasks that can consume much user time. We describe obspyDMT’s functionality, design and technical implementation, accompanied by an overview of its use cases. As an example of a typical problem encountered in seismogram preprocessing, we show how to check for inconsistencies in response files of two example stations. We also demonstrate the fully automated request, remote computation and retrieval of synthetic seismograms from the Synthetics Engine (Syngine) web service of the Data Management Center (DMC) at the Incorporated Research Institutions for Seismology (IRIS).

1 Introduction

Seismology is a data-rich science, and since the advent of global digital networks in the 1990s, the growth of seismological waveform data holdings at international data centers has constantly accelerated. The data avalanche is a blessing, but also poses challenges to the scientist who needs to find and process these waveforms. Which data are available at the various international data centers? How can subsets of interest be selected, downloaded, organized, preprocessed, instrument-corrected and quality-controlled in a manageable amount of user time? Quality control and instrument corrections are nontrivial tasks, requiring tools that provide adequate diagnostics to verify data integrity. Almost every data-driven workflow in seismology begins with these considerations. As a project progresses, local data holdings often need to be updated, repaired, or extended, including the troubleshooting of earlier failed requests, adding waveforms made available since initial retrieval, adding (meta-)data from other data centers and downloading corrected metadata files. Surgical tasks of this kind can easily require more human supervision than the initial retrieval.

For a sense of data volumes, consider the example of Fig. 1, which arose in our work on global waveform tomography (Hosseini et al., 2014; Hosseini, 2016). Using the obspyDMT software, we queried the Incorporated Research Institutions for Seismology (IRIS) Data Management Center (DMC) about hour-long, broadband waveform segments containing earthquakes exceeding a magnitude of 5. Figure 1a plots the data center’s response: since 1990, IRIS’ event catalog lists 1000–3000 such events per year, visual-

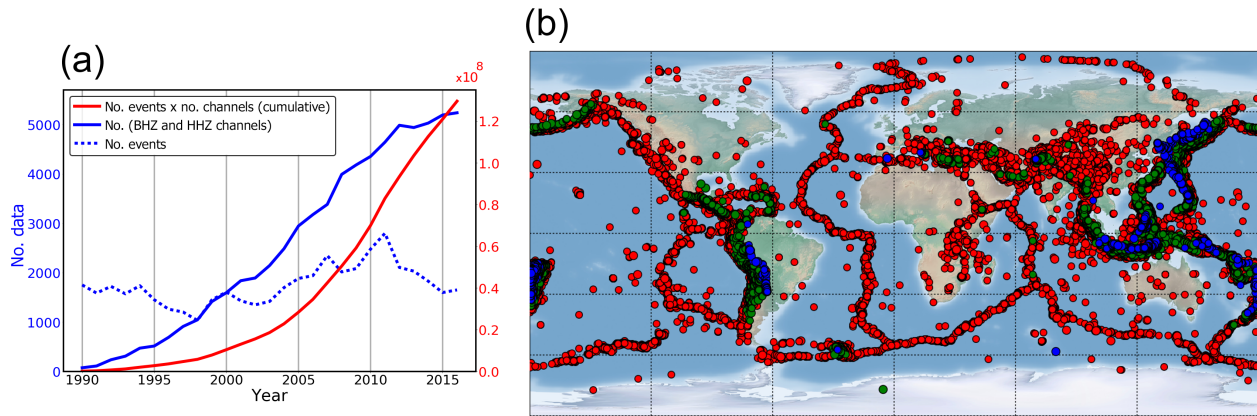


Figure 1. `obsipyDMT --datapath iris_events_dir --min_date 1990-01-01 --max_date 2017-01-01 --min_mag 5.0 --event_info --plot_seismicity`

Rapid growth of seismological waveform data holdings at international data centers since 1990. Using the obsipyDMT command above, we queried the IRIS DMC for hour-long, vertical, broadband (BHZ and HHZ) waveform segments containing earthquakes exceeding a magnitude of 5.0. **(a)** The data center's response. Red line shows cumulative sum of available event-based waveforms for this request; $\sum_{y=1990}^{\text{year}} [\text{num_events}(y) \times \text{num_channels}(y)]$. Number of events and seismograms in each year are shown by dotted and solid blue lines, respectively. **(b)** Global seismicity map of earthquakes in panel **(a)** colored by depth. Red: 0–70 km; green: 70–300 km; blue: ≥ 300 km. The generation of this map is triggered by the `--plot_seismicity` flag. Upon startup of the plotting module, the user can select the map style, “Shadedrelief” in this example.

ized in obsipyDMT's automatically generated map of Fig. 1b. The number of archived broadband channels has grown to almost 5200 in 2016, and we are offered more than 10^8 waveforms, corresponding more than 20 terabytes of data (and very long download times). Most applications would call for the selection of desirable subset of data before launching an actual request.

Besides large volumes, the hallmark of seismological data is heterogeneity. A culture of data sharing from permanent networks and temporary experiments means that waveforms get archived at many different data centers around the world in different waveform and metadata formats and documented and quality-controlled to varying degrees. Archives receive continuous inflows of data from telemetered stations, but also batchwise contributions from temporary experiments. Many experiments make metadata available immediately but restrict access to actual waveforms for several years. No general mechanism exists for broadcasting updates about data center holdings, which instead need to be actively and repeatedly queried by interested users. Data access mechanisms tend to be specific to each center. Downloading time-continuous or very long seismograms may be less supported than downloading short segments around earthquake occurrences.

obsipyDMT is free, open-source community software that strives to address these access challenges in a more comprehensive, integrated and time-saving manner than existing software, which includes WILBER, WebDC, BREQ_FAST, NetDC, EMERALD (West and Fouch, 2012), IGeoS (Morozov and Pavlis, 2011a, b), SOD (Owens et al., 2004) and

ObsPyLoad (Scheingraber et al., 2013). It is an easy-to-use command-line tool for the query, retrieval and management of seismograms. The user is shielded from the complexities of interacting with different data centers and provided with powerful diagnostic tools to check the retrieved data and metadata and to execute most routine preprocessing tasks, including instrument corrections. ObsipyDMT is written in the Python programming language and runs on Linux, Mac OS and Windows platforms.

Section 2 gives a high-level overview of obsipyDMT's functionality in comparison to existing seismogram retrieval and management tools. Section 3 is a concise but near-complete tour that aims to turn the reader into a productive obsipyDMT user very quickly while also listing all usage options. Section 4 discusses implementation and performance of features that set obsipyDMT apart from existing tools, specifically its communication with data centers, its robustness and its diagnostics for instrument corrections.

All graphics in this paper were generated by obsipyDMT. The caption of each figure gives the generating command(s) that handled the data and produced the plot.

2 Overview of software functionality

obsipyDMT is a stand-alone tool for data retrieval and management that is not associated with any one seismological data center, data exchange protocol, or data format. In a style similar to Unix shells, it issues a single, one-line command

obsypyDMT

which produces a default behavior and can be customized with many different options flags. There are no required options, and the omission of an option flag will trigger default behavior. This makes obsypyDMT robust to run and easy to learn. The possibilities for customization are extensive, as will be discussed in Sect. 3. To give an idea, the command

```
obsypyDMT --datapath iris_events_dir
--min_date 1990-01-01 --max_date
2017-01-01 --min_mag 5.0
--event_info --plot_seismicity
```

downloaded a global seismicity catalog from the IRIS DMC, saved the metadata in a predefined directory structure and generated Fig. 1 as a diagnostic display of the result. Invoking obsypyDMT without any flags would have requested from the IRIS event catalog metadata for all events since 1970 that exceeded a magnitude of 3.0.

obsypyDMT is part of the ObsPy ecosystem (Beyreuther et al., 2010; Megies et al., 2011; Krischer et al., 2015), an open-source community project that develops Python software for seismological observatories under the GNU Lesser General Public License, hosted by the Ludwig-Maximilians-Universität Munich. ObspyDMT uses many of ObsPy's utility functions, as well as functions from Python's numpy, scipy and matplotlib libraries (Hunter, 2007), combining them into a more specialized piece of software. While no knowledge of Python is required to use obsypyDMT, a software developer may seamlessly integrate it with other Python code. Python also makes it easy to wrap source codes written in other programming languages. For example, ObsPy wraps evalresp, IRIS' maturely developed software for instrument response corrections. ObspyDMT's functionality can be summarized as follows.

- Query of station metadata: by absolute time or relative to earthquake occurrences; by geographic area (rectangles or circles); by channel or instrument type; wildcarding (*) is supported; simultaneous queries of different data centers.
- Query of earthquake source metadata: from different catalog providers (currently from NEIC, GCMT (Global Centroid Moment Tensor), IRIS DMC, NCEDC, USGS, INGV and ISC); event origin information or full-moment tensors; by time window, region, event magnitude and/or event depth.
- Diagnostic plots to visualize metadata; plots are generated simply by appending an option flag to the data-handling command.

- Retrieval of actual waveform data (seismograms) according to the results of metadata queries. Support for different data exchange protocols (International Federation of Digital Seismograph Networks (FDSN) web services, ArcLink).
- Retrieval of time-continuous series of arbitrary length; generation of diagnostic log files.
- Parallelized retrieval of waveform data from a data center for increased speed. Simultaneous retrieval from different data centers.
- Update mode: identical or modified queries can be relaunched; only new, modified, or previously failed data will be retrieved from the data center(s).
- Tolerant of retrieval errors and missing data (includes diagnostic logs).
- Automatic organization of data, metadata and log files into standardized directory trees. (At present no tie to a database system.)
- Processing of retrieved data sets using default or user-defined instructions. ObsPy, SAC (George Helffrich and Bastow, 2013) or any other processing tool can be used to customize the processing unit on the waveform level. Supports processing immediately upon waveform retrieval or later, batch-type processing. Support for parallel processing.
- Application of instrument responses. Support for various instrument formats (e.g., StationXML and dataless SEED). Diagnostic plots of analog and digital “filter stages”. Option of parallelized instrument correction, taking advantage of multi-core architectures now common even on desktop processors.
- Automated retrieval of synthetic seismograms from IRIS' data services products (Hutko et al., 2017) for comparison to real data.

Various community software packages exist for achieving these tasks, but to our knowledge no other freely available package achieves them all. Table 1 compares the features of popular seismological community software to those of obsypyDMT. We consider only tools that include functionality for data retrieval.

All data centers offer such tools, but each is limited to retrieving data from that specific center. For example, both the IRIS DMC in the US and ORFEUS Data Center (ODC) in Europe implement the web-form-based WILBER service for retrieving event-based waveforms, as well as the email-based BREQ_FAST service for time-continuous waveforms. If a user requires data from both centers, they need to be contacted separately. If event-based as well as continuous data

Table 1. Comparison of seismological data retrieval and management tools. Abbreviations: E – event-based; C – continuous time series; U – update mode. ObspyDMT is the only tool to provide access to both FDSN and ArcLink (in a single command), to retrieve both event-based and time-continuous waveform data, and to offer an “update” mode for waveforms, response files and/or metadata information. Few other tools provide for the management of data download and archiving, instrument correction, or diagnostics plots. EIDA: European Integrated Data Archive.

Tool	Data access			Data management		
	Method	Data sources/interfaces	Retrieval modes	Archiving	Instrument correction	Plots
WILBER	web portal	IRIS DMC or ODC/EIDA	E	×	×	×
WebDC	web portal	ODC/EIDA	E	×	×	×
BREQ_FAST	email	IRIS DMC or ODC/EIDA	C	×	×	×
NetDC	email	NCEDC	C	×	×	×
EMERALD	direct	IRIS DMC	E	✓	×	✓
IGeoS	direct	IRIS DMC	E	×	×	✓
SOD	direct	FDSN	E	×	✓ (gain correction)	✓
obsypyDMT	direct	FDSN and ArcLink	C, E, U	✓	✓	✓

are required, any given center needs to be contacted twice, using two different tools.

obsypyDMT is the only tool among those in Table 1 that provides access to several data centers (in a single command) and to both types of waveform data (in two separate commands). The demand for continuous time series, often in large quantities, has surged with the rapid rise in cross-correlation methods based on ambient noise (Shapiro and Campillo, 2004). ObspyDMT provides more convenient access than the email-based tools BREQ_FAST or NetDC.

obsypyDMT is also the only tool to offer an “update” mode for waveforms, response files and/or metadata information: relaunching a previous request will identify and retrieve only data that could not be retrieved earlier. Like obsypyDMT, the SOD, IGeoS and EMERALD tools are stand-alone software that runs on the user’s computer rather than a data center server. All four communicate with data centers via the relatively new web services interfaces defined by the FDSN. Queries are formulated as URL strings (uniform resource locators) that point to physical data resources over the internet. We refer to this access method as “direct”. Compared to older access methods, it can save much human intervention time by freeing the user from the need to click through web pages (WILBER, WebDC) or manage emails (BREQ_FAST, NetDC). SOD, IGeoS and EMERALD retrieve event-based waveforms only, i.e., queries are based on earthquake occurrences.

The stand-alone tools obsypyDMT and EMERALD additionally manage the data download and archiving to a local computer, thus relieving users of additional tedious and time-consuming steps. Both include certain plotting options (more extensively in obsypyDMT).

obsypyDMT also offers full instrument correction based on RESP or StationXML station metadata, combined with diagnostic plots of transfer functions for individual filter stages. SOD is the only other tool to offer instrument correction, but

this includes gain correction only, and it offers no diagnostic plots.

obsypyDMT is the only tool to provide an automated update functionality for a user’s existing, local data holdings.

3 Guided tour of use cases

The purpose of this section is to turn the reader into a proficient user of obsypyDMT in the short space of a few pages. We demonstrate the most common use cases for the query, selection, retrieval and management of seismograms, metadata and synthetic waveform. We list obsypyDMT’s full set of options in Table 2, which should be consulted as a cross-reference during the various stops of this guided tour.

We will

1. query event metadata from different earthquake catalogs
2. query station metadata from different data centers
3. request waveform data for a subset of events (“event-based mode”), from several different data centers
4. demonstrate how to update a local data set (“update mode”)
5. query and download continuous time series in arbitrary, user-provided time windows (“continuous mode”)
6. speeding up data retrieval by parallelization and bulk requests
7. demonstrate obsypyDMT’s plotting capabilities as we go
8. apply instrument corrections to waveform data
9. retrieve synthetic seismograms from Syngine (Synthetics Engine) web service (Krischer et al., 2017), to match observed seismograms.

obsipyDMT is a command-line tool that consists of a single command

```
obsipyDMT
```

usually followed by option flags to modify the default behavior. Table 2 lists all available flag options, with explanations.

3.1 Querying earthquake metadata

First, we request event information from one of several supported seismicity catalogs, without downloading any waveforms yet.

```
obsipyDMT --datapath neic_event_dir
--min_date 1990-01-01 --max_date
2017-01-01 --min_mag 5.0 --event_catalog
NEIC_USGS --event_info --plot_seismicity
```

This obsipyDMT command with seven option flags queries the NEIC catalog (`--event_catalog NEIC_USGS`) for all events exceeding a magnitude of 5.0 (`--min_mag`) that happened between 1990 and 2016 (`--min_date`, `--max_date`). `--plot_seismicity` triggers the generation of the global seismicity map plot of Fig. 2. `--event_info` switches off the retrieval of actual seismograms so that only metadata are downloaded to a local directory named `neic_event_dir/` (argument of `--data_path`). This directory is created if necessary, and it is populated with the following subdirectory and files:

```
neic_event_dir
├── EVENTS-INFO
│   ├── catalog.txt
│   ├── catalog.ml
│   ├── catalog_table.txt
│   ├── event_list.pickle
│   └── logger_command.txt
```

Geographical restrictions for event (or station) queries are supported in rectangular or circular areas. For example, to extract only earthquake metadata for Indonesia, specify `lonmin/lonmax/latmin/latmax` as

```
--event_rect 80/135/-15/35
```

Appended to the earlier command, this generates the map inset of Fig. 2b. Note the rendering of colored beach balls (deepest seismicity in the foreground). The global map of Fig. 2 also plots beach balls rather than simple black dots, but they do not become apparent at this zoom level.

3.2 Query of station metadata

Let's say we plan to investigate earthquakes exceeding a magnitude of 6.0 that occurred in this Indonesian rectangle at depths above 100 km. We want to know which seismometers in the Global Seismic Network (GSN) were operational to record them from 1 February to 1 December 2014. We issue the following query:

```
obsipyDMT --datapath event_based_dir
--min_date 2014-02-01 --max_date
2014-12-01 --min_mag 6.0 --max_depth
100 --event_rect 80/135/-15/35
--event_catalog NEIC_USGS --net _GSN
--cha BHZ --meta_data
```

The NEIC event catalog returns 16 matching earthquakes, metadata for which are stored in 16 separate subdirectories of a local directory called `event_based_dir`. Each of the 16 event subdirectories holds a subdirectory called `availability.txt` to which metadata were written describing the GSN seismometers that were operational during the event. (Refer to Appendix A and Fig. A1 for a graphic depicting the full directory structure created by obsipyDMT.) Only station metadata are requested, as specified by the mode flag `--meta_data`. We want StationXML files for (all) stations in the GSN network (`--net _GSN`), but only for the broadband, high-gain, vertical components of these stations, as specified by channel flag `--cha BHZ`. A subset of stations could be specified by the `--sta` flag, which supports wildcarding `*`, like many obsipyDMT options. Since the option is absent here, it defaults to `--sta *`, i.e., all stations in the `_GSN` network. (See Table 2 for defaults for all options.) The underscore in `--net _GSN` marks this as a virtual network, whereas the two regular networks IU and II would be queried by `--net "IU, II"`.

3.3 Requesting and retrieving waveform data in event-based mode

Next, we retrieve the actual BHZ seismograms from the GSN network that were recorded during the 16 Indonesian earthquakes identified in Sect. 3.2. In our earlier obsipyDMT command, only a few option flags need to be changed:

```
obsipyDMT --datapath event_based_dir
--min_date 2014-02-01 --max_date
2014-12-01 --min_mag 6.0 --max_depth
100 --event_rect 80/135/-15/35
--event_catalog NEIC_USGS --net _GSN
--cha BHZ --preset 300 --offset 3600
--instrument_correction --data_source
IRIS
```

Table 2. Complete list of option flags to customize the default behavior of the obspyDMT command.

Group	Options	Description	Example
Check installation	--help --tour --check --version	Show this help message and exit. Run a quick tour. Check all basic dependencies and their installed versions on the local machine and exit. Show the obspyDMT version and exit.	
Local path specification	--datapath <PATH> --reset	Path where obspyDMT will store/process/plot data (default: “./obspydmtdata”). If the datapath is found, delete it before running obspyDMT.	“/desired/path”
Data retrieval modes	--event_based --continuous --meta_data --local	Event-based request mode (default). Continuous time series request mode. Metadata request mode. Local mode for processing/plotting (no data retrieval).	
General options (all modes)	--data_source <SOURCE>	Data source(s) for retrieving waveform/response/metadata (default: “IRIS”).	“IRIS” or “IRIS,ORFEUS” or “all”
	--print_data_sources	Print-supported data centers that can be passed as arguments to --data_source.	
	--print_event_catalogs	Print-supported earthquake catalogs that can be passed as arguments to --event_catalog.	
	--waveform <True/False> --force_waveform	Retrieve waveform(s) (default: true). Retrieve waveform(s), force override of any preexisting waveforms in local datapath directory.	False
	--response <True/False> --force_response	Retrieve response file(s) (default: true). Retrieve response file(s), force override of any preexisting response files in local datapath directory.	False
	--dir_select <DirNames>	Selects a subset of data directories for which to update/process/plot the contents (default False, i.e., all subdirectories will be considered).	“dir1,dir2”
	--min_epi <in deg> --max_epi <in deg> --min_azimuth <in deg> --max_azimuth <in deg> --list_stas <PATH>	Retrieve/plot all stations with epicentral distance \geq min_epi. Retrieve/plot all stations with epicentral distance \leq max_epi. Retrieve/plot all stations with azimuth \geq min_azimuth. Retrieve/plot all stations with azimuth \leq max_azimuth. User-provided station list instead of querying availability with a data center (default: false).	“30” “90” “10” “120” “/path/list-stations”
Time window, waveform format and sampling rate (all modes)	--min_date <DATE>	Start time, syntax: “YYYY-MM-DD-HH-MM-SS” or “YYYY-MM-DD” (default: “1970-01-01”).	“2010-09-24”
	--max_date <DATE>	End time, syntax: “YYYY-MM-DD-HH-MM-SS” or “YYYY-MM-DD” (default: today).	“2015-01-01”
	--preset <in sec>	Time interval in seconds to add to the retrieved time series <i>before</i> its reference time. In event_based mode, the reference time is the earthquake origin time by default but can be modified by --cut_time_phase. In continuous mode, the reference time(s) is (are) specified by --interval option, and --preset prepends the specified lead to <i>each</i> interval (default: 0).	“300”
	--offset <in sec>	Time interval in seconds to include to the retrieved time series <i>after</i> the time reference. In event_based mode, the reference time is the earthquake origin time by default but can be modified by --cut_time_phase. In continuous mode, the reference time(s) are specified by --interval option, and --offset appends the specified offset to <i>each</i> interval (default: 1800).	“3600”
	--cut_time_phase	In event_based mode, use the first-arriving phase as reference time (i.e., P, Pdiff or PKIKP, determined automatically). Overrides the use of origin time as default reference time.	
	--waveform_format <mseed/sac>	Format of retrieved waveforms. Default is miniSEED (“mseed”), alternative option is “sac”. This fills in some basic header information as well.	“sac”
	--sampling_rate <in Hz>	Desired sampling rate (in hertz). If not specified, the sampling rate of the waveforms will not be changed.	“10”
	--resample_method <lanczos/decimate>	Resampling method: “decimate” or “lanczos”. Both methods use sharp low-pass filters before resampling in order to avoid aliasing. If the desired sampling rate is 5 times lower than the original one, resampling will be done in several stages (default: “lanczos”).	“decimate”

Table 2. Complete list of option flags to customize the default behavior of the obspyDMT command.

Stations (all modes)	--net <NET>	Network code (default: *).	"TA" or "TA.G" or "T*" or "*"
	--sta <STA>	Station code (default: *).	"RR01" or "RR01,RR02" or "R*" or "*"
	--loc <LOC>	Location code (default: *).	"00" or "*"
	--cha <CHA>	Channel code (default: *).	"BHZ" or "BHZ,BHE" or "BH*" or "*"
	--identity <NET.STA.LOC.CHA>	Identity code restriction, syntax: net.sta.loc.cha, e.g., IU.*.*.BHZ to search for All BHZ channels in IU network (default: *.*.*).	"IU.*.*.BH*"
	--station_rect <lonmin/lonmax/latmin/latmax>	Include all stations within the defined rectangle, syntax: <lonmin>/<lonmax>/<latmin>/<latmax>. Cannot be combined with circular bounding box (--station_circle) (default: -180.0/+180.0/-90.0/+90.0).	"20/30/-15/35"
	--station_circle <lon/lat/rmin/rmax>	Include all stations within the defined circle, syntax: <lon>/<lat>/<rmin>/<rmax>. Cannot be combined with rectangular bounding box (--station_rect) (default: 0/0/180).	"20/30/10/80"
Speedup options (all modes)	--req_parallel	Enable parallel waveform/response request. Retrieve several waveforms/metadata in parallel.	
	--req_np <num_thread>	Number of thread to be used in --req_parallel (default: 4).	"8"
	--bulk	Send a bulk request to an FDSN data center. Returns multiple seismogram channels in a single request. Can be combined with --req_parallel.	
	--parallel_process	Enable parallel local processing of the waveforms, useful on multicore hardware.	
	--process_np <num_thread>	Number of threads to be used in --parallel_process (default: 4).	"8"
Restricted data	--user <username>	Username for restricted data requests, waveform/response modes (default: none).	"your_username"
	--pass <password>	Password for restricted data requests, waveform/response modes (default: none).	"your_password"
Event-based mode	--event_catalog <CATALOG>	Event catalog, currently supports LOCAL, NEIC_USGS, GCMT_COMBO, IRIS, NCEDC, USGS, INGV, ISC (default: LOCAL). --event_catalog LOCAL searches for an existing event catalog on the user's local machine, in the EVENTS-INFO subdirectory of --datapath <PATH>. This is usually a previously retrieved catalog.	"IRIS"
	--event_info	Retrieve event information (metadata) without downloading actual waveforms.	
	--read_catalog <PATH>	Read in an existing local event catalog and proceed. Currently supported catalog metadata formats: "CSV", "QUAKEML", "NDK", "ZMAP". Format of the plain text CSV (comma-separated values) is explained in the obspyDMT tutorial. Refer to ObsPy documentation for details on QuakeML, NDK and ZMAP formats.	"/path/to/file.ml"
	--min_depth <in km>	Minimum event depth (default: -10.0 (above the surface!)).	"10"
	--max_depth <in km>	Maximum event depth (default: +6000.0).	"100"
	--min_mag <min_mag>	Minimum magnitude (default: 3.0).	"4.0"
	--max_mag <max_mag>	Maximum magnitude (default: 10.0).	"7.0"
	--mag_type <mag_type>	Magnitude type. Common types include "MI" (local/Richter magnitude), "Ms" (surface wave magnitude), "mb" (body wave magnitude), "Mw" (moment magnitude) (default: none, i.e., consider all magnitude types in a given catalog).	"Mw"
	--event_rect <lonmin/lonmax/latmin/latmax>	Include all events within the defined rectangle, syntax: <lonmin>/<lonmax>/<latmin>/<latmax>. Cannot be combined with circular bounding box (--event_circle) (default: -180.0/+180.0/-90.0/+90.0).	"80/135/-15/35"
	--event_circle <lon/lat/rmin/rmax>	Search for all the events within the defined circle, syntax: <lon>/<lat>/<rmin>/<rmax>. Cannot be combined with rectangular bounding box (--event_rect) (default: 0/0/180).	"20/30/10/80"
	--isc_catalog <COMPREHENSIVE/REVIEWED>	Search either the COMPREHENSIVE or the REVIEWED bulletin of the ISC. COMPREHENSIVE: all events collected by the ISC, including most recent events that are awaiting review. REVIEWED: includes only events that have been relocated by ISC analysts. (default: COMPREHENSIVE).	"REVIEWED"

Table 2. Complete list of option flags to customize the default behavior of the obspyDMT command.

Continuous time series mode	--interval <in sec>	Specify time interval for subdividing long continuous time series (default: 86400 s).	"3600"
Local processing	--pre_process <name_process_unit> --force_process	Process retrieved data based on processing instructions in the selected processing unit (default: "process_unit"). Forces running of the processing unit on the local/retrieved data, overwriting any previously processed data in local datapath directory.	"process_unit_sac"
	--instrument_correction --corr_unit <DIS/VEL/ACC>	Apply instrument correction in the process unit. Correct the raw waveforms for displacement in <i>m</i> (DIS), velocity in <i>m/s</i> (VEL) or acceleration in <i>m/s²</i> (ACC) (default: DIS).	"VEL"
	--pre_filt (f1,f2,f3,f4)	Apply a bandpass filter to the seismograms before deconvolution, syntax: "none" or "(f1,f2,f3,f4)" which are the four corner frequencies of a co-sine taper, default: "(0.008, 0.012, 3.0, 4.0)".	"(0.008, 0.012, 3.0, 4.0)"
	--water_level <in dB>	Water level in dB for instrument response deconvolution (default: 600.0).	"300"
	--syngine --syngine_bg_model <MODEL>	Retrieve synthetic waveforms using IRIS/syngine webservice. Syngine background model (default: "iasp91_2s").	"iasp91_2s" or "prem_a_2s"
Synthetic seismograms	--print_syngine_models	Print-supported syngine models that can be passed as arguments to --syngine_bg_model.	
	--syngine_geocentric_lat <True/False>	Requesting synthetic seismograms based on geocentric latitudes of events/stations (default: true).	False
	--plot --plot_sta --plot_availability	Activates plotting functionality. Plot all stations found in the specified directory (--datapath). Plot all availabilities (potential seismometers) found in the specified directory (--datapath).	
	--plot_ev --plot_focal --plot_ray	Plot all events found in the specified directory (--datapath). Plot beachballs instead of dots for event locations. Plot the ray coverage for all station–event pairs found in the specified directory (--datapath).	
	--create_kml --create_event_vtk --plot_seismicity --depth_bins_seismicity <in km>	Create a KML file for event/station/ray. KML format is readable by Google Earth. Create a VTK file for event(s). VTK format is readable by Paraview. Create a seismicity map and some basic statistics on the results. Depth bins for plotting the seismicity histogram (default: 10 km).	"5"
Plotting	--plot_waveform --plot_dir_name <raw/processed/...>	Plot waveforms arranged by epicentral distance. Directory name that contains the waveforms for --plot_waveform option flag, e.g., --plot_waveform processed (default: raw).	"raw"
	--plot_save <PATH>	Path where plots will be stored (default: ".", i.e., the current directory).	"."
	--plot_format <png/jpeg/pdf/...>	Image format of plots (default: "png").	"png"
	--plot_lon0 <lon0>	Central meridian (<i>x</i> axis origin) for projection (default: 180).	"160"
	--plot_stationxml	Plot the contents of stationXML file(s), i.e., transfer function of filter stages, specified by --datapath.	
Explore instrument responses (stationXML files)	--plotxml_date <DATE>	Date and time to be used for plotting the transfer function, syntax: "YYYY-MM-DD-HH-MM-SS" or "YYYY-MM-DD". If not specified, the starting date of the last channel in the stationXML will be used.	"2010-01-01"
	--plotxml_output <DIS/VEL/ACC>	Type of transfer function to plot: DIS/VEL/ACC (default: VEL).	"DIS"
	--plotxml_allstages --plotxml_paz --plotxml_plotstage12	Plot all filter stages specified in response file. Plot only poles and zeros (PAZs) of the response file, i.e., the analog stage. Plot only stages 1 and 2 of full response file.	
	--plotxml_start_stage <stage>	First stage in response file to be considered for plotting the transfer function (default: 1).	"1"
	--plotxml_end_stage <stage>	Final stage in response file to be considered for plotting the transfer function, (default: last stage given in response file or the 100th stage, whichever number is smaller).	"3"
Others	--plotxml_min_freq <in Hz> --plotxml_map_compare	Minimum frequency in Hz to be used in transfer function plots (default: 0.01). Plot all stations for which instrument responses have been compared (PAZ against full response).	"0.001"
	--plotxml_percentage <percent>	Percentage of the phase transfer function's frequency range to be used for checking the difference between methods. "100" will compare transfer functions across their entire spectral range, i.e., from min_freq (set by --plotxml_min_freq) to Nyquist frequency; "80" compares from min_freq to 0.8 times Nyquist frequency (default: 80).	"100"
	--email <email address>	Send an email to the specified address after completing the job (default: false).	"email_address"
	--arc_avai_timeout <in sec>	Timeout (in seconds) for sending a data availability query via ArcLink (default: 40).	"60"
	--arc_wave_timeout <in sec>	Timeout (in seconds) for sending a waveform data or metadata request via ArcLink (default: 2).	"60"

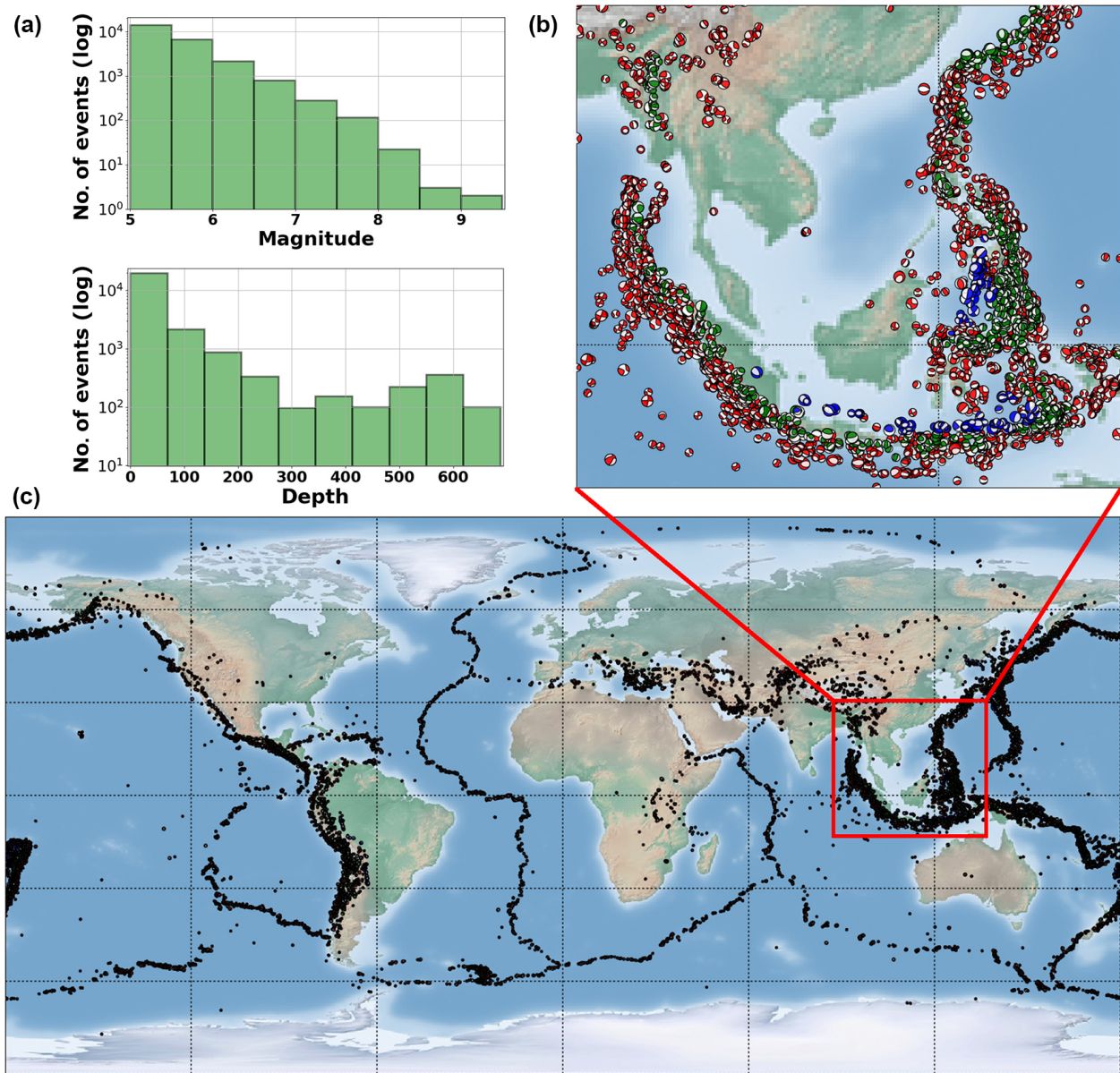


Figure 2. `obsipyDMT --datapath neic_event_dir --min_date 1990-01-01 --max_date 2017-01-01 --min_mag 5.0 --event_catalog NEIC_USGS --event_info --plot_seismicity`
Global seismicity map of archived earthquakes in NEIC catalog of a magnitude of more than 5.0 that occurred between 1990 and 2016. One command queried the NEIC catalog, stored and organized the retrieved information and generated the seismicity map. (No actual waveform data were queried in this example.) The results of some basic statistics (magnitude and depth histograms) are also generated and plotted automatically (a). Note the rendering of colored beach balls in the map inset (deepest seismicity in the foreground). The global map also contains beach balls rather than just simple black dots, but they do not become apparent at this zoom level.

`--data_source` specifies explicitly that the IRIS DMC should be contacted, although this would also be the default if the flag were omitted. If the user is unsure, it is best to specify `--data_source all`, which prompts obsipyDMT to contact all 20 supported data centers listed in Table 3 and probably more in the future. (The list can be inspected by invoking `obsipyDMT --print_data_sources`.)

`--preset 300` and `--offset 3600` specify the retrieval of waveform time windows of 300 s before to 3600 s after the reference time. Since we are downloading in event-based mode, i.e., centered around earthquake occurrences, the reference time defaults to the event origin time. This could be changed to the time of *P*-wave arrival by invoking `--cut_time_phase` (see Table 2), in which case each seismogram would have a different abso-

Table 3. List of international data centers that can be currently accessed via FDSN and ArcLink interfaces of obspyDMT. This list is growing as more and more data centers can be accessed directly (as opposed to FTP or email-based methods). `obsipyDMT --print_data_sources` lists all available data centers, and `--print_event_catalogs` lists all available event catalogs.

Interface	Data source	URL
FDSN	BGR	http://eida.bgr.de
	EMSC	http://www.seismicportal.eu
	ETH	http://eida.ethz.ch
	GEONET	http://service.geonet.org.nz
	GFZ	http://geofon.gfz-potsdam.de
	INGV	http://webservices.rm.ingv.it
	IPGP	http://eida.ipgp.fr
	IRIS	http://service.iris.edu
	ISC	http://isc-mirror.iris.washington.edu
	KOERI	http://eida.koeri.boun.edu.tr
	LMU	http://erde.geophysik.uni-muenchen.de
	NCEDC	http://service.ncedc.org
	NIEP	http://eida-sc3.infp.ro
	NOA	http://eida.gein.noa.gr
	ODC	http://www.orfeus-eu.org
	ORFEUS	http://www.orfeus-eu.org
	RESIF	http://ws.resif.fr
	SCEDC	http://service.scedc.caltech.edu
	USGS	http://earthquake.usgs.gov
	USP	http://sismo.iag.usp.br
ArcLink	Many European data centers	

lute start time. ObspyDMT knows that it is downloading in event-based mode because this is its default mode; adding the flag `--event_based` would have made this explicit. (`--meta_data` mode was introduced in Sect. 3.2; the alternative modes of `--continuous` and `--local` will be demonstrated shortly.)

Issuing this single-line command is the only requirement on user time; everything else is done automatically. Specifically, obspyDMT will do the following:

1. Request event information from the NEIC event catalog `--event_catalog NEIC_USGS`.
2. In the `--datapath event_based_dir`, create a subdirectory `EVENTS-INFO/` containing a local catalog of metadata for the 16 matching events. Also in `--datapath`, create 16 event subdirectories, each containing a subdirectory tree (`info/`, `resp/`, `raw/`, `processed/`) as in Appendix A, Fig. A1.
3. Retrieve station metadata for all GSN stations for the 16 events in StationXML format from the IRIS data center and save these to subdirectories `resp/`.
4. Retrieve BHZ waveforms of 3900 s duration from all matching GSN stations in miniSEED format and save to subdirectories `raw/`.
5. Run default preprocessing operations on the waveforms, consisting of removing means and trends, tapering, fil-

tering, and deconvolving the instrument response (all customizable). The processed seismograms are save to subdirectories `processed/`.

6. Save additional log files on query success to subdirectories `info/`.

Note how user time remains limited to issuing a single command no matter how many earthquakes, stations, or waveforms are being requested. Our tests required no human intervention even for very large requests that took weeks to download and encountered various time-outs or missing data issues at the data centers (cf. Sect. 4.2).

3.4 Update of existing waveform data sets

In the course of working with a waveform data set, it often becomes necessary to update it. This could mean requesting the same data again (because part of the earlier request failed for some reason) or expanding the number of earthquakes, stations or seismograms. ObspyDMT aims to be smart about these various cases and not to retrieve duplicates unless the users explicitly wants it to. We demonstrate typical use cases. They have in common that the local `--datapath` directory must remain identical to that of any earlier request.

If an earlier query encountered problems (e.g., connection down, time-outs) or if the user has reason to expect that the data centers have added more seismograms since (e.g., the

embargo period of a temporal network has ended), then it suffices to relaunch the exact same request (which was saved in log file `EVENTS-INFO/logger_command.txt`):

```
obsipyDMT --datapath event_based_dir
--min_date 2014-02-01 --max_date
2014-12-01 --min_mag 6.0 --max_depth
100 --event_rect 80/135/-15/35
--event_catalog NEIC_USGS --net _GSN
--cha BHZ --preset 300 --offset 3600
--instrument_correction --data_source
IRIS
```

obsipyDMT compares the newly obtained event and station metadata to their local versions and downloads only holdings that differ.

If the user wants to update only certain events, then `--min_date`, `--max_date`, `--min_mag`, `--max_mag` and/or `--event_rect` can be adjusted (see Table 2 for other options). Similarly, if the new date–time window is not contained within the old one, then additional events might fit the criteria and their waveforms would be added in new event directories.

If all 16 preexisting event directories are to be updated, an alternative to the above command is to remove all event criteria because obsipyDMT will then default to the local, preexisting event catalog in `EVENTS-INFO/` for earthquake metadata.

```
obsipyDMT --datapath event_based_dir
--net _GSN --cha BHZ --preset 300
--offset 3600 --instrument_correction
--data_source IRIS
```

If the user decides they need seismograms for all BHE channels (in addition to BHZ), the update command would be

```
obsipyDMT --datapath event_based_dir
--net _GSN --cha BHE --preset 300
--offset 3600 --instrument_correction
--data_source IRIS
```

Augmenting the existing 16 events with seismograms from additional data centers is also an update operation because the waveform holdings of data centers often overlap to some extent. Again obsipyDMT will automatically compare metadata in order to avoid downloading duplicates. To update the data set with all vertical broadband channels of the GFZ and ORFEUS data centers, we would request

```
obsipyDMT --datapath event_based_dir
--cha BHZ --preset 300 --offset 3600
--instrument_correction --data_source
"GFZ,ORFEUS"
```

`--datapath event_based_dir` is identical to what we defined in the previous command line that specifies the name of the top directory.

3.5 Retrieval of waveform data in time-continuous mode (`--continuous`)

In contrast to the examples thus far, some usage cases require waveforms that are not relative to or centered on specific earthquake occurrences. We refer to this usage mode as “time continuous” (`--continuous`). For example, studies that cross-correlate ambient noise often require long time series from many stations, often divided into segments of shorter duration (i.e., 1 day). ObspyDMT makes the handling of continuous time series easy, even if the data sets are voluminous.

```
obsipyDMT --continuous --datapath
yv_continuous_dir --min_date 2012-12-15
--max_date 2013-01-15 --net YV
--sta "RR0*,RR1*,RR2*" --cha BHZ
--sampling_rate 10 --data_source
RESIF --user your_username --pass
your_password
```

This command queries the French RESIF data center for time series from 15 December 2012 to 15 January 2013 recorded by the temporary ocean-bottom seismometer network of the RHUM-RUM (Réunion Hotspot and Upper Mantle – Réunions Unterer Mantel) experiment (network code YV) (Barruol and Sigloch, 2013; Stähler et al., 2016). The wildcard “*” is used to specify multiple station names. Since the data are embargoed until the end of 2017, a username and password needed to be passed to the data center (`--user`, `--pass`). Here we were interested in noise levels on the ocean floor during the passage of tropical storm Dumile and therefore requested waveforms for the storm period, highlighted by the yellow box in Fig. 3. The storm was clearly recorded by elevated noise levels, whose variable onset times track the storm’s diachronous passage across the 1500 km × 1500 km wide network (Davy et al., 2014).

Long time series often need to be downsampled for ease of storage and handling, in this case to 10 Hz from originally 50 Hz (`--sampling_rate 10`). ObspyDMT uses ObsPy functionality for resampling to any rate; if the frequency ratio is large, antialiasing and downsampling are automatically done in multiple stages.

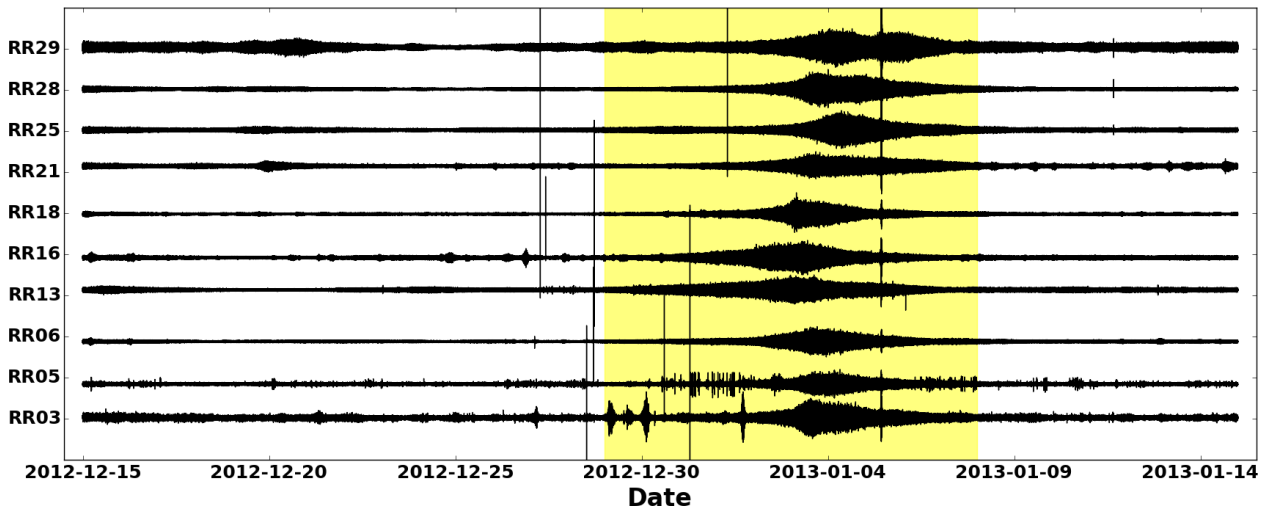


Figure 3. `obsipyDMT --continuous --datapath yv_continuous_dir --min_date 2012-12-15 --max_date 2013-01-15 --net YV --sta "RR0*,RR1*,RR2*" --cha BHZ --sampling_rate 10 --data_source RESIF --user your_username --pass your_password`
Retrieval of continuous time series of arbitrary length, here for 30 days in 2012/2013. Data are from the temporary ocean-bottom network RHUM-RUM (network YV, station names RR*) and are currently still password-protected at the RESIF data center (`--user`, `--pass`). The command specifies downsampling to 10 Hz immediately upon retrieval. The passage of the tropical storm Dumile is highlighted by the yellow box.

3.6 Speeding up data retrieval by parallelization

obsipyDMT uses ObsPy clients to retrieve metadata and actual waveforms from the data centers. Every request consists of three basic steps: (1) connect and send the data request to the data center; (2) download the data; (3) disconnect. By default, obsipyDMT executes these steps for every metadata or waveform request separately, e.g., 3×1000 steps if 1000 waveforms are requested. For large requests, this can become a serious bottleneck. To increase the efficiency in such cases, a functionality for parallelized data retrieval can be enabled as follows:

```
--req_parallel --req_np 4
```

The first flag changes the data retrieval mode from serial (default) to parallelized, and the second flag specifies the number of parallel requests.

The parallelization in obsipyDMT is implemented on two levels: data center and waveforms. As an example of the former, if waveform data from both ORFEUS and IRIS are requested, obsipyDMT sends parallel requests to these data centers.

The other parallelization is at waveform level: if several waveforms are requested from one data center, they are retrieved by `--req_np` parallel processes. (A good choice for `np` is the number of CPUs on the retrieving computer, i.e., 4 to 16 for many current laptops or desktops.) The number of requested waveforms or metadata files will be divided into

the number of specified processes. Each process then sends and retrieves its set of requests serially, but all processes organize their data into the same `--datapath` directory.

Further speeding up can be achieved by specifying a bulk request (`--bulk` flag). Instead of requesting individual items, this will send a list of items (time series or metadata) to the data center, which reduces the number of (dis-)connections. We have, however, noticed occasional instabilities (for very large requests, fewer waveforms are retrieved than in serial mode); hence, serial is set as the conservative default.

3.7 Plotting tools

obsipyDMT offers various plotting tools for visualizing data sets. Figure 2 demonstrates the plotting of seismic sources (beach balls) on a map, via the `--plot_seismicity` option.

Figure 4 demonstrates a map plot of ray paths between sources and receivers for the Indonesian example data set of Sect. 3.1 to 3.4 in Google Earth:

```
obsipyDMT --datapath event_based_dir
--local --plot_ev --plot_focal
--plot_sta --plot_ray --create_kml
```

Triggered by the plotting options, obsipyDMT plots the contents of data directory “event_based_dir”, specifically the 16 event locations (`--plot_ev`) including focal mech-

anisms (`--plot_focal`), stations (`--plot_sta`), and ray paths (`--plot_ray`). One file in KML format is created (`--create_kml`), which can be displayed by Google Earth. If `--create_kml` is omitted, obspyDMT plots the contents of the data set in maps similar to Figs. 2 or 5 (refer to Sect. 3.9). The flag `--local` explicitly tells obspyDMT to operate on preexisting content in the local data path directory, rather than making new contact with a data center.

3.8 Processing and instrument correction

obspyDMT can process the waveforms directly after retrieving the data, or it can process an existing data set in a separate step (local mode). By default, obspyDMT follows processing instructions described in the `process_unit.py` file located in the `/path/to/my/obsipyDMT/obsipyDMT` directory. This scripting file can be freely edited by the user and may include calls to external waveform processing programs such as ObsPy or SAC. This vastly expands the possibilities for waveform processing and lets users easily adapt and integrate functionality from earlier, non-obsipyDMT workflows. Instructions in this file are written at the waveform level, and obsipyDMT applies them to all waveforms in the entire data set (in serial or in parallel mode). The default file included in the current distribution, `/path/to/my/obsipyDMT/obsipyDMT/process_unit.py`, can perform routine processing steps such as resampling, data format conversion and instrument correction. These steps can be accessed via dedicated option flags, each of which results in the execution of only the appropriate part of processing script `process_unit.py` (see `--pre_process` option flag). Hence, a user requiring only these routine operations need not create or modify a processing script file. The operations include

1. resampling time series, for example, downsampling for ease of storage and handling (refer to Sect. 3.5 and `--sampling_rate` option flag)
2. converting the format of retrieved waveforms to SAC and filling in some headers by the simple inclusion of the `--waveform_format sac` option flag
3. instrument correction which includes removing means and trends, tapering, prefiltering (customizable by `--pre_filt` option flag) and deconvolving the instrument response to displacement, velocity or acceleration (all customizable).

As an example, to correct the waveforms for instrument response directly after retrieving the data (similar to the example of Sect. 3.3)

```
obsipyDMT --datapath event_based_dir
--min_date 2014-02-01 --max_date
2014-12-01 --min_mag 6.0 --max_depth
100 --event_rect 80/135/-15/35
--event_catalog NEIC_USGS --net _GSN
--cha BHZ --preset 300 --offset 3600
--instrument_correction --data_source
IRIS --corr_unit VEL
```

`--corr_unit VEL` specifies the physical unit of the processing output, in this case ground velocity in meters per second. The same data set can be corrected for displacement in a separate step (not directly after retrieving the data):

```
obsipyDMT --datapath event_based_dir
--local --force_process
--instrument_correction --corr_unit DIS
```

Since obsipyDMT stores processed waveforms in the processed directory (Fig. A1), good practice is to rename all processed directories before launching the above command line; otherwise, previously processed waveforms will be overwritten (`--force_process`).

The user can also modify the `process_unit.py` or write a new script with new processing instructions. Currently, these files need to be located in the `/path/to/my/obsipyDMT/obsipyDMT` directory and can be accessed via `--pre_process my_proc_unit` option flag, replacing `my_proc_unit` with the name of the Python script. The instructions are written at the waveform level, and obsipyDMT automatically applies them to all archived waveforms. The main advantage of this design choice is its flexibility. The user can customize the processing instructions using available tools in ObsPy; moreover, other processing tools can be used or combined to write these instructions. As an example, the following command line calls a processing instruction `process_unit_sac.py`; this file is located in `/path/to/my/obsipyDMT/obsipyDMT`:

```
obsipyDMT --datapath event_based_dir
--local --force_process --pre_process
process_unit_sac
```

Here, SAC (instead of ObsPy) is used to remove the mean, apply a Hanning window, compute the FFT (fast Fourier transform), plot the amplitude spectrum of each waveform on a log-log plot and save the images as PDF files in the processed directory.

3.9 Requesting synthetic seismograms

obsipyDMT facilitates the generation of synthetic waveforms matching the real data in two ways by (1) retrieving synthetic

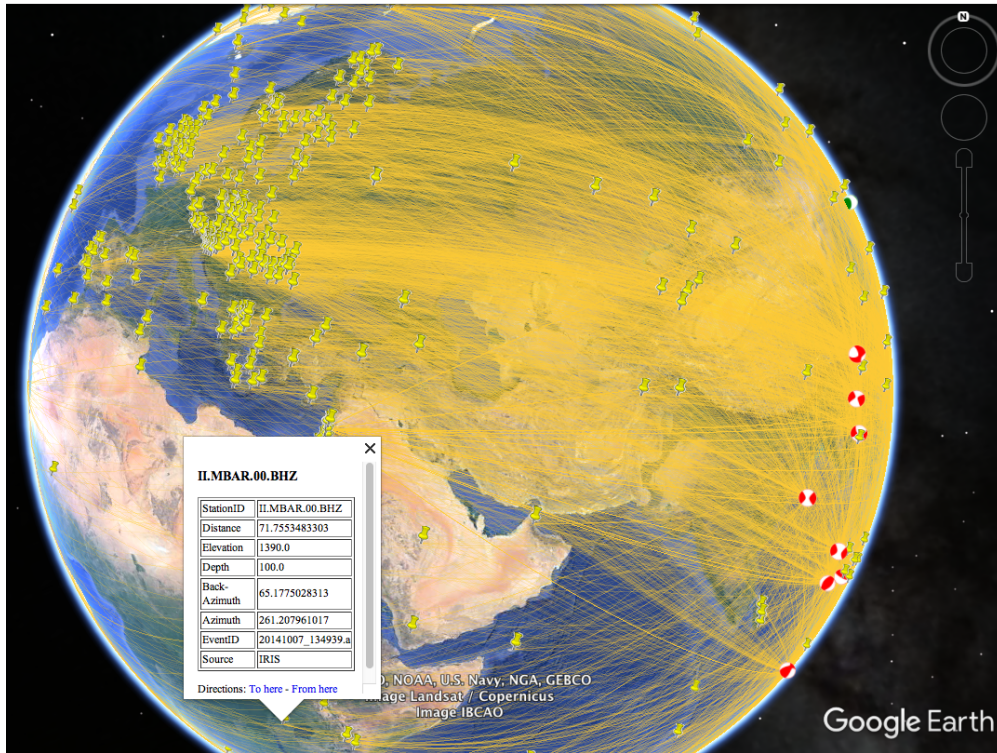


Figure 4. `obsipyDMT --datapath event_based_dir --local --plot_ev --plot_focal --plot_sta --plot_ray --create_kml`

Plot of the contents of the `--datapath event_based_dir` that contains the Indonesian example data set generated in Sects. 3.1 to 3.4. `--local` specifies that the existing, local waveform holdings should be plotted, rather than contacting the data centers anew. Sixteen earthquake locations are plotted as beach balls; stations featuring BHZ channels are indicated by yellow markers. Waveforms were retrieved from three data centers (IRIS, ORFEUS, GFZ).

waveforms from a new IRIS web service: Syngine (Krischer et al., 2017) and (2) providing required metadata for calculating synthetic waveforms using external tools.

Syngine delivers fully numerical seismic waveforms computed on common spherically symmetric Earth models (PREM – Preliminary Reference Earth Model; ak135-f; IASP91). The following example command retrieves not only observed waveforms but also their synthetic counterparts, computed on a PREM (Dziewonski and Anderson, 1981) anisotropic background model:

```
obsipyDMT --datapath data_fiji_island
--min_mag 6.8 --min_date 2014-07-21
--max_date 2014-07-22 --event_catalog
NEIC_USGS --data_source IRIS --min_azi
50 --max_azi 55 --min_epi 94 --max_epi
100 --cha BHZ --instrument_correction
--syngine --syngine_bg_model prem_a_2s
```

The two option flags that triggered the synthetic waveform retrieval are `--syngine` and `--syngine_bg_model prem_a_2s`. The option flags `--min_azi`,

`--max_azi`, `--min_epi` and `--max_epi` specify minimum azimuth, maximum azimuth, minimum distance and maximum distance for station search, respectively. The synthetic waveforms are stored in the `syngine_prem_a_2s` directory, the contents of which can be plotted by obsipyDMT plotting tools (refer to Fig. 5).

Changing the argument of `--syngine_bg_model` to `iasp91_2s`, synthetic seismograms based on the IASP91 (Kennett and Engdahl, 1991) background model can be retrieved (Fig. 5):

```
obsipyDMT --datapath data_fiji_island
--min_mag 6.8 --min_date 2014-07-21
--max_date 2014-07-22 --event_catalog
NEIC_USGS --data_source IRIS --min_azi
50 --max_azi 55 --min_epi 94 --max_epi
100 --cha BHZ --instrument_correction
--syngine --syngine_bg_model iasp91_2s
```

All earth reference models currently supported by Syngine can be listed by invoking

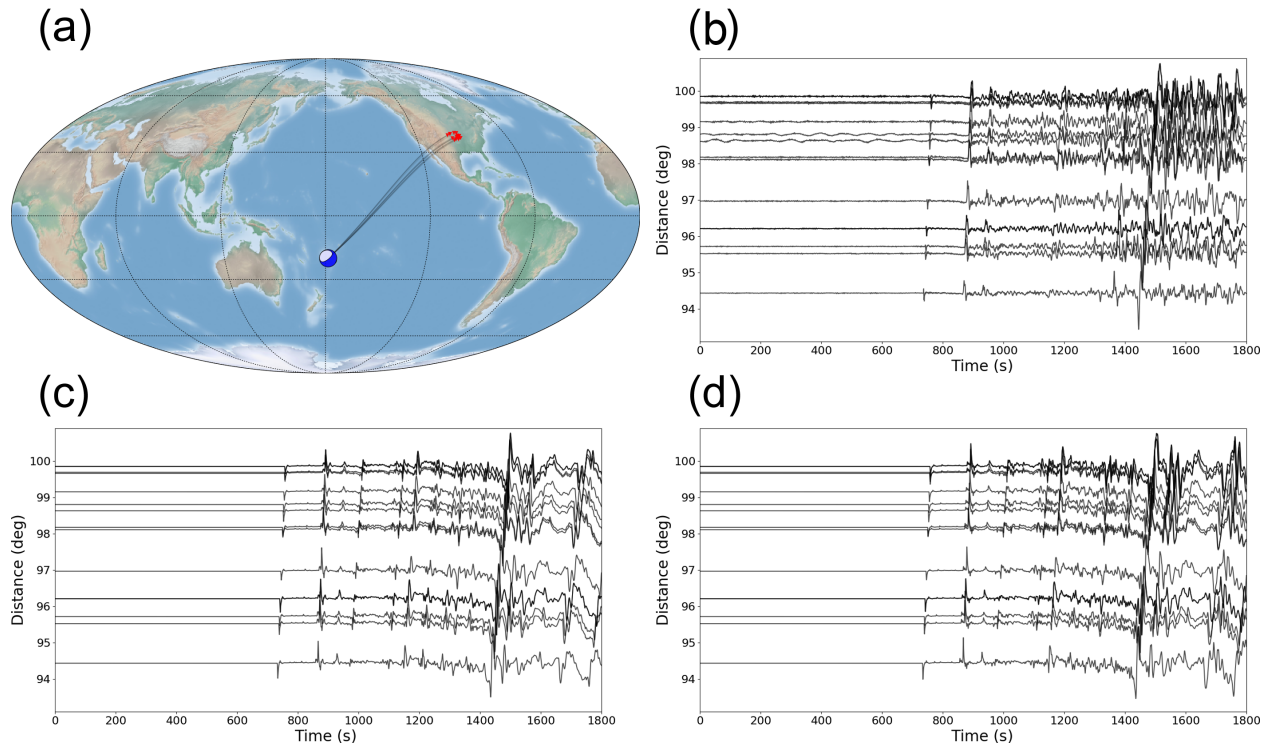


Figure 5. Observed versus modeled broadband seismograms for an earthquake of a magnitude of 6.9 M_w in the Fiji Islands region (21 July 2014, 14:54:41, at 19.802° S, 178.4° W; 615 km depth). **(a)** Source and receiver distribution plotted by `obsipyDMT --datapath data_fiji_island --local --plot_ev --plot_focal --plot_sta --plot_ray`. Note the distribution of stations with respect to the event. The options flags `--min_azi`, `--max_azi`, `--min_epi` and `--max_epi` specified minimum azimuth, maximum azimuth, minimum distance and maximum distance for station search, respectively. **(b)** Observed broadband waveforms plotted by `obsipyDMT --datapath data_fiji_island --local --plot_waveform --plot_dir processed`. **(c)** Synthetic seismograms retrieved from the Syngine web service for the PREM anisotropic background model. The stored waveforms are plotted by `obsipyDMT --datapath data_fiji_island --local --plot_waveform --plot_dir syngine_prem_a_2s`. Panel **(d)** is similar to **(c)** except for the IASP91 background model. Plotted by `obsipyDMT --datapath data_fiji_island --local --plot_waveform --plot_dir syngine_iasp91_2s`.

```
obsipyDMT --print_syngine_models
```

Alternatively, metadata information and log files generated and organized by `obsipyDMT` can be used to link an archived data set to other software for the generation of synthetic seismograms. A practical example of this is multiple-frequency tomography. In this method, frequency-dependent observables (phase shifts or amplitudes) are measured by cross-correlating the recorded waveforms with the corresponding synthetic seismograms in multiple frequency bands (Sigloch, 2008; Zaroli et al., 2015; Hosseini and Sigloch, 2015). Synthetic seismograms need to be computed for exactly the same sources and receivers in the data set. This includes source characteristics (epicenter, depth, moment tensor and source time function) and receiver specifications (latitude, longitude, elevation and burial).

`obsipyDMT` stores station information in one ASCII file per event and in the SAC headers (if this waveform format is selected). It automatically updates metadata information and

log files of a local data archive if stations are added/removed. Event information is written in QuakeML and ASCII formats. Although basic source and receiver information can be retrieved from most data centers, moment tensor solutions are available only in certain seismicity catalogs, among them the NEIC and GCMT catalogs, which are both supported by `obsipyDMT` (refer to moment tensor retrieval as demonstrated by Fig. 2).

In summary, `obsipyDMT` retrieves, organizes and stores all meta-information required to compute synthetic seismograms using arbitrary forward-modeling tools. Users only need to provide scripts that connect this metadata input to their desired computational engine (other than Syngine), for example, AxiSEM (Nissen-Meyer et al., 2014) or Instaseis (van Driel et al., 2015).

4 Discussion

Here we discuss implementation and performance issues, specifically obspyDMT's communication with data centers, its robustness in the case of large and heterogeneous requests, and the usefulness of the instrument correction diagnostics. All three features set obspyDMT apart from existing tools.

4.1 Communication with data centers

obspyDMT can retrieve data from a multitude of international data centers (Table 3; a list that is growing). The user is shielded from having to know communication specifics for each data center. Under the hood, the software implements ObsPy clients for two different kinds of data exchange protocols: FDSN web services and ArcLink.

In 2013, the FDSN defined common web service interfaces (<http://www.fdsn.org/webservices/>), allowing data request tools to work with any of the growing number of FDSN data centers that implement these interfaces (<http://www.fdsn.org/webservices/datacenters/>). These centers currently include the IRIS DMC, BGR, EMSC, ETH, GEONET, GFZ, INGV, IPGP, ISC, KOERI, LMU, NCEDC, NIEP, NOAA, ODC, ORFEUS, RESIF, SCEDC, USGS and USP. Three service interfaces are specified by the FDSN and supported by ObsPy: `fdsnws-station` for accessing station metadata in StationXML format, `fdsnws-dataset` for accessing time series in miniSEED format, and `fdsnws-event` for accessing earthquake parameters in QuakeML format. ObspyDMT offers conversion to other formats, e.g., SAC for waveforms `--waveform_format sac`. Requests are sent via the HTTP internet protocol for individual requests and via HTTP-POST for lists of requests, so that data can be requested from any web browser by generating URLs.

ArcLink is an older data request protocol that arose in Europe in order to virtually consolidate distributed seismological data holdings across various European countries. It is a distributed request protocol developed by the German WebDC initiative of GEOFON and BGR (Bundesanstalt für Geowissenschaften und Rohstoffe) as a continuation of the NetDC concept originally developed by the IRIS DMC. ArcLink communicates via TCP/IP rather than via supervision-intensive email or FTP requests required by other access mechanisms at the time. It accesses waveform data in miniSEED or SEED format and associated meta-information as dataless SEED files. At the time we developed ObsPyLoad, a pre-cursor of obspyDMT (Scheingraber et al., 2013), only a few data centers were implementing FDSN web services. Hence, ArcLink clients greatly expanded the reach of ObsPyLoad, to include most European data centers. ObsPyLoad contacts the ORFEUS DMC via ArcLink, which in turn “forwards” ArcLink requests to other data centers across Europe. This ArcLink functionality is retained in obspyDMT, but if a data center implements both interfaces, then obspyDMT accesses it via web services (default), which

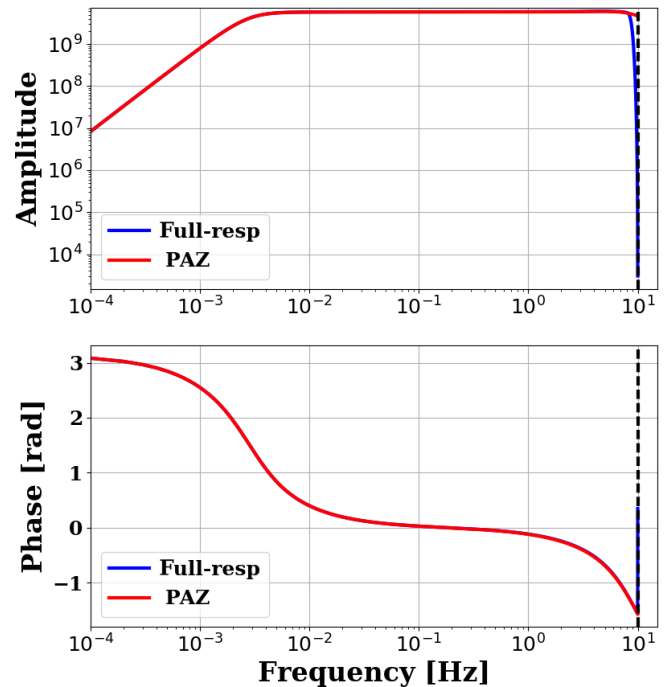


Figure 6. obspyDMT `--plot_stationxml`
`--plotxml_paz` `--plotxml_min_freq 0.0001`
`--datapath /path/to/STXML.IC.XAN.00.BHZ`
 Transfer function spectra (amplitude and phase) of a Streckeisen STS-1VBB w/E300 station (IC.XAN) in China. Blue lines show the transfer function components computed for all filter stages in a StationXML file; red lines are for the analog part. The two functions match very well in all frequencies except for the amplitude spectra close to the Nyquist frequency (dashed line).

now includes the European data centers. It seems likely that web services will completely supersede ArcLink.

4.2 Robustness of data retrieval

In our research we have used obspyDMT extensively, in order to retrieve several voluminous, event-based data sets for global-scale tomography, from different combinations of data centers. We have also requested large volumes of time-continuous data (“ambient noise”) for cross-correlation studies. In all cases, we observed obspyDMT to work stably, i.e., requiring no user intervention despite the fact that many individual waveform requests encounter errors from the data centers, for various reasons. ObspyDMT caught all exceptions and continued undeterred.

In a demanding test that expanded the scope of the example of Sect. 3.3, we retrieved all BHZ channels from all supported data sources, in event-based mode, requesting earthquakes exceeding a magnitude of 6.0 that occurred during 2 years. The idea was to test the most challenging request mode, which includes station and event metadata, and to

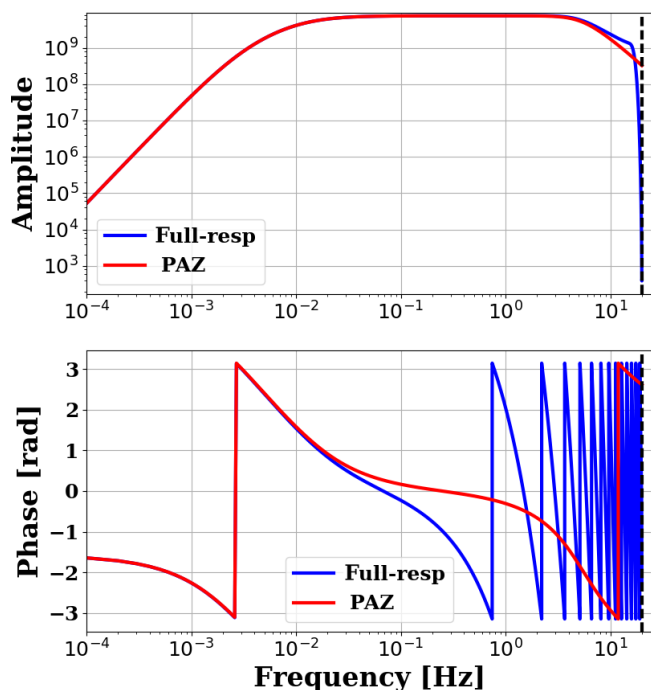


Figure 7. `obspyDMT --plot_stationxml --plotxml_paz --plotxml_min_freq 0.0001 --datapath /path/to/STXML.GT.LBTB.00.BHZ` Transfer function spectra (amplitude and phase) of a Geotech KS-54000 borehole seismometer (GT.LBTB) in Botswana. Blue lines show transfer function components computed for all filter stages in the StationXML file; red lines are for the analog part. A large discrepancy exists between the phase spectra of the two transfer functions. The deviation emerges at frequencies around 10^{-2} Hz and increases up to the Nyquist frequency. Fig. 8 shows that this difference is caused by one of the digital stages in the instrument response.

communicate with all data centers, including some that implemented web services very recently.

```
obspyDMT --datapath 2014_2015_dataset
--min_date 2014-01-01 --max_date
2016-01-01 --min_mag 6.0 --event_catalog
NEIC_USGS --cha BHZ --data_source
all --preset 300 --offset 3600
--req_parallel --req_np 8 --pre_process
False
```

The retrieval took 2 days and 10 h on a standard desktop with 4 CPUs. The retrieved data set was 145 GB in size, containing 293 events and 685 388 waveforms. No user intervention was required at any stage.

This finding is consistent with the performance of obspyDMT's predecessor ObsPyLoad (Scheingraber et al., 2013). With an event-based request similar to the one above to all data centers available at the time (in 2012 this was IRIS and the European centers via ORFEUS/ArcLink), we retrieved

162 GB of waveform data, consisting of 690 503 miniSEED files for three components (BHZ, BHE and BHN) for 154 events. The retrieval took 45 days because the job slowed down considerably after the first 73 GB (but continued at the old speed after relaunching, i.e., requesting the remaining 89 GB through update mode). The fraction of successfully retrieved waveforms varied strongly between data centers and ranged from 99.8 to 34.8 % (availabilities were verified by spot checks in manual retrieval attempts). The exact reasons for the slowdown remained unclear, but aside from the decision to relaunch, no user intervention was required at either download stage.

For the current test in 2017, no such slowdown was observed, and the retrieval of a comparable data volume (145 GB) took only a 1/20 of the time (2.5 days), despite being routed to many more data centers. We conclude that obspyDMT works robustly with all supported data centers, even for large and heterogeneous data and metadata requests.

4.3 Instrument correction

If station metadata could be routinely trusted, correcting for instrument responses would amount to a simple series of deconvolutions of a number of impulse responses (analog and digital filter stages from raw waveforms). Unfortunately, it is not uncommon for filter information in station metadata files to be erroneous. Some of the resulting artifacts in the displacement or velocity seismograms are large enough to potentially cause serious geoscientific misinterpretation, such as pronounced travel time delays under an isolated island station where in reality there are none.

Problems with the contents of StationXML or SEED/RESP files may or may not be straightforward to identify, as discussed below. A full visual representation of filter impulse responses can greatly facilitate trouble shooting. ObspyDMT implements several plotting options for this purpose, as demonstrated in Sect. 3.8 and Figs. 6–8.

An instrument response typically consists of a first, analog stage (a.k.a. “poles and zeros”, or PAZ stage), which describes the transfer function of the sensor, and several digital stages, which describe the A/D conversion, antialiasing and downsampling inside the data logger. The PAZ stage is rarely problematic, whereas specifications of the digital stages are error-prone. Our discussion of neuralgic points and their possible diagnosis follows the PhD thesis of Groos (2010).

Coefficients of asymmetric FIR filters are sometimes given in reverse order from that expected by the SEED convention, which can cause erroneous time delays of up to 1 s in the “corrected” waveforms. This issue may not be easy to detect as it requires knowledge of the correct order of filter coefficients, e.g., by comparing it to a trusted StationXML file describing the same data logger in a different location.

A typical, unproblematic response resembles Fig. 6, with PAZ and full response coinciding everywhere except near the Nyquist frequency. By contrast, a plot like Fig. 7 can flag up

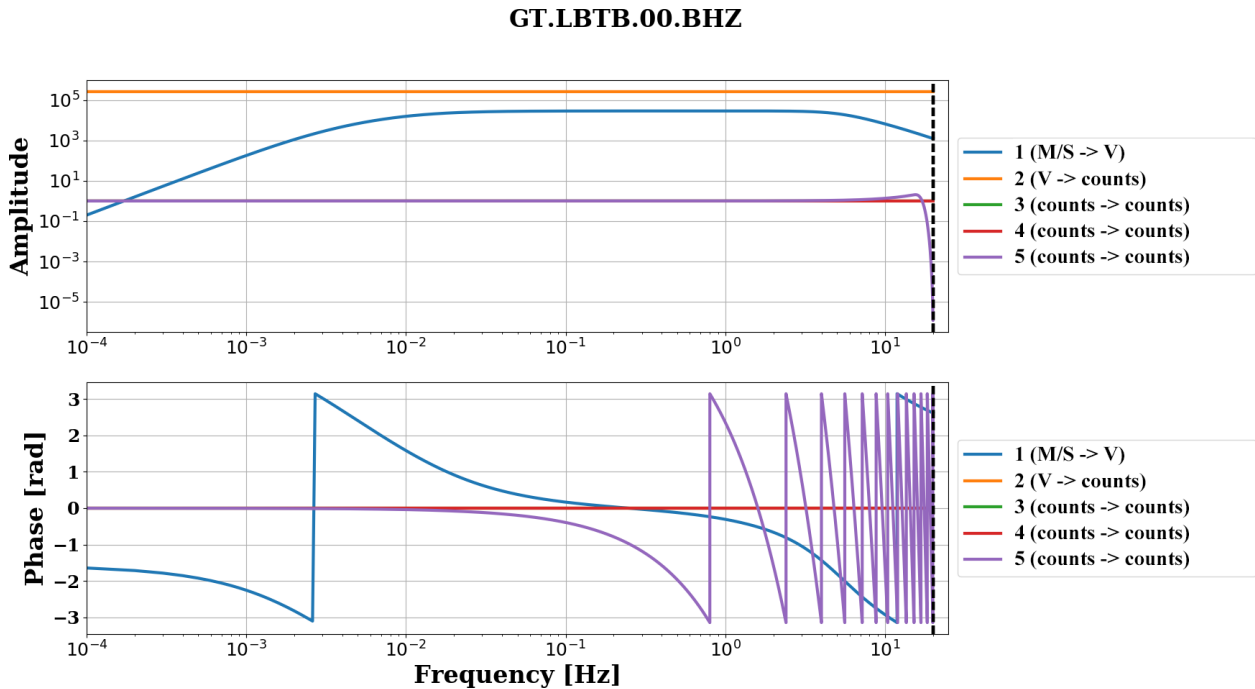


Figure 8. `obspyDMT --datapath /path/to/STXML.GT.LBTB.00.BHZ --plot_stationxml --plotxml_min_freq 0.0001 --plotxml_allstages`
 Transfer function spectra (amplitude and phase) of each stage in the StationXML file of a Geotech KS-54000 borehole seismometer (GT.LBTB) in Botswana. In the phase response, two stages (1 and 5) have non-zero values. Both stages contribute to the phase spectrum of the complete instrument response (“full-resp”) of Fig. 7. However, the effects of Stage 5 on amplitude and phase spectra are not considered in PAZ (analog).

a potential problem. The very different phase responses of PAZ-only versus full response indicate that the digital stages introduce a significant delay (and possibly distortion) of the corrected time series. The user can then question whether this behavior is expected from the data logger. ObspyDMT automatically creates diagnostic reports for stations where PAZ and full response differ significantly. Figure 8 further zooms in on the issue, by indicating that among the digital stages, only Stage 5 has a non-zero phases response, identifying it as the questionable one. If the user decides that the digital stage specifications are suspect, they can choose to apply PAZ-only correction rather than full response – this should give a decent result, except for frequencies very close to Nyquist. Alternatively, if the user is working with low-frequency data only (below 0.01 Hz), they can conclude that no problem would ever arise because even Stage 5 is almost 0 in that spectral range.

Another recurring problem concerns delay time values specified for the FIR filter stages. According to the SEED manual, corrected filter delay times have to be positive; and yet, negative or 0 values are sometimes encountered in retrieved metadata files which can result in erroneous time shifts of 1 to 2 s in corrected waveforms. This problem is easily spotted, but 7 years after the report by (Groos, 2010), we still encounter such response files delivered by data centers.

obspyDMT also checks for inconsistencies in the “estimated delay” and the “correction applied” of the digital filter stages. In modern data loggers, these two values are usually similar because delay times are removed from the waveforms internally. However, discrepancies have been observed, such as negative or 0 values for the corrected delay time. In the example of Fig. 7, the estimated delay is reported as 0.63 s, and the applied correction is 0.0 s. ObspyDMT collects this information and automatically generates one diagnostic report for the results of all consistency checks.

5 Conclusions

We presented obspyDMT, a new software for the query, retrieval, processing and management of large seismological data sets. Its functionality, design and technical implementation were described and compared with existing seismological data retrieval and management tools. Through examples we demonstrated its main functionalities, such as a query of station and earthquake source metadata (full-moment tensor and event origin), the retrieval of event-based or time-continuous waveform data from various data centers in one command line, an update mode, a customizable processing unit, and the automatic organization of (meta-)data and log

files into standardized directory trees. The user is provided with powerful diagnostic and plotting tools to check the retrieved data and metadata. For large seismological data sets, data retrieval and processing can be parallelized on multi-core architectures by the simple inclusion of an option flag. Using obspyDMT's diagnostic plots of analog and digital filter stages, we checked the spectra (amplitude and phase) of instrument response files. Synthetic seismograms matching an example data set were retrieved from IRIS Syngine.

In all these use cases, issuing a single-line command is the only requirement for the user, everything else is done automatically.

Refer to Appendix C for instructions on how to download and install obspyDMT.

Data availability. The code is available on GitHub (<https://github.com/kasra-hosseini/obsipyDMT> or <http://kasra-hosseini.github.io/obsipyDMT/>)

Appendix A: Directory structure

ObspyDMT organizes retrieved seismograms and metadata in a standardized directory structure, as shown in Fig. A1.

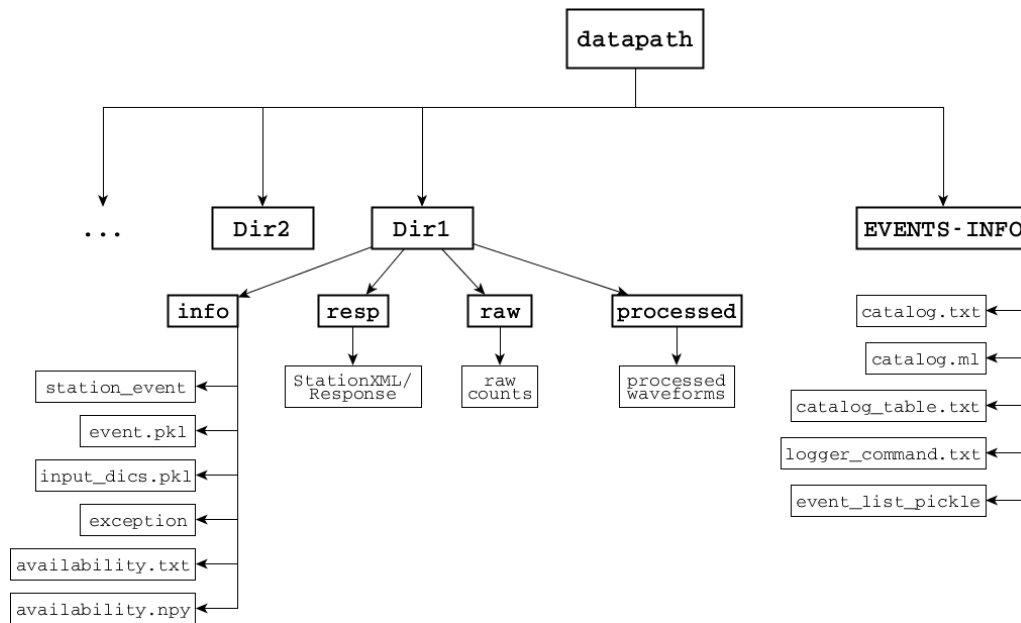


Figure A1. For each request, obspyDMT creates the depicted directory tree inside the user-specified directory `datapath/` and arranges the retrieved data either in event subdirectories (for event-based requests) or in chronologically named subdirectories (for continuous requests). It also creates a subdirectory `EVENTS-INFO/` in which a catalog of all requested events or time spans is stored. Earthquake metadata (date and time, latitude, longitude, depth, magnitude, moment tensor, source time function) are stored in CSV and QuakeML formats (files `catalog.txt`, `catalog.ml`). The file `catalog_table.txt` organizes basic event information (latitude, longitude, depth, date and time, magnitude) in a table. Raw waveforms, StationXML/response files and corrected waveforms are collected in subdirectories. During the data retrieval process, obspyDMT also creates metadata log files about retrieved station and event files, stored in the `info/` subdirectory of each event directory.

Appendix B: Instrument correction

Seismograms recorded by digital broadband seismometers are stored as digitized voltage signals called “raw counts”. The relation between this signal and ground motion (e.g., displacement) depends on the response functions of the seismometer and data logger components (sensor, amplifiers, A/D converters, digital filters). Each component is referred to as a “stage” characterized by a transfer function, and the entire system can be described by the cumulative transfer function, i.e., a product in the frequency domain of the individual stage transfer functions. The instrument sensor, i.e., the analog measurement apparatus before A/D conversion, is referred to as the analog stage or the poles-and-zeros stage. Following the nomenclature of the *SEED Manual* (Ahern et al., 2012), its frequency response G can be written as

$$G(j\omega) = S_d A_0 \frac{\prod_{n=1}^N (j\omega - r_n)}{\prod_{m=1}^M (j\omega - p_m)}. \quad (\text{B1})$$

r and p stand for zeros and poles of a system. N and M are the number of zeros and poles, respectively. S_d is the stage gain. A_0 is the normalization factor, which scales the amplitude of the poles-and-zeros polynomial to unity at a reference frequency (usually 1 Hz):

$$A_0 \left| \frac{\prod_{n=1}^N (j\omega_{\text{ref}} - r_n)}{\prod_{m=1}^M (j\omega_{\text{ref}} - p_m)} \right| = 1. \quad (\text{B2})$$

G relates the ground motion V (input signal) to recorded raw counts R by (Scherbaum, 1996):

$$R(j\omega) = G(j\omega) \times V(j\omega), \quad (\text{B3})$$

in which, $R(j\omega)$ and $V(j\omega)$ are the Fourier transforms of raw counts and ground motion, respectively. Instrument response correction can be carried out by transforming the raw seismogram $R(t)$ to the spectral domain, dividing $R(j\omega)$ by $G(j\omega)$ (deconvolution in time) and transforming the result back into the time domain, in order to obtain $V(t)$ in the physical units of displacement, velocity or acceleration.

Instrument responses are provided by data centers in different formats. An older format called SEED describes transfer functions of all analog and digital stages in a seismometer and is hence sufficient to calculate the frequency response function of a seismic channel ($G(j\omega)$ in Eq. B3). In practice, this format is usually converted to human readable ASCII files called SEED RESP that can be read by other instrument correction software such as evalresp. Recently, FDSN defined a new format FDSN StationXML which contains the most important and commonly used structures of SEED metadata in XML representation (FDSN, 2015). Compared to SEED, StationXML simplifies and adds clarification to station metadata. All data centers that support FDSN web services deliver instrument responses in this format. ObspyDMT can read and interpret both StationXML and SEED.

Appendix C: Installation and system requirements

C1 ObsPy

ObsPy (Beyreuther et al., 2010; Megies et al., 2011; Krischer et al., 2015) is currently running and tested on Linux (32 and 64 bit), Windows (32 and 64 bit) and Mac OS X. Please refer to the ObsPy web page for complete notes regarding ObsPy installation on different platforms.

In addition to Python and ObsPy tools, obspyDMT builds on NumPy, an extension for performing numerical calculations on large arrays and matrices (van der Walt et al., 2011); matplotlib, a popular plotting package (Hunter, 2007); matplotlib basemap toolkit (Whitaker, 2015) to project the data on a map; and SciPy (Jones et al., 2001), a library for advanced math, signal processing or statistics. Most of these libraries are prerequisites for installing ObsPy and are used in obspyDMT.

C2 obspyDMT

Once working Python and ObsPy environments are available, obspyDMT can be installed in different ways:

1. Install obspyDMT package locally (using PyPi). This tends to be the most user-friendly option:

```
pip install obspyDMT
```

2. Install obspyDMT from the source code. The latest version of obspyDMT is available on GitHub. After installing git:

```
git clone https://github.com/kasra-hosseini/obsipyDMT.git /path/to/my/obsipyDMT

cd /path/to/my/obsipyDMT
```

obsipyDMT can be installed by:

```
pip install -e .
```

or

```
python setup.py install
```

obsipyDMT can be used from a system shell without explicitly calling the Python interpreter. The following command checks the dependencies required for running the code properly:

```
obsipyDMT --check
```

obsipyDMT contains various option flags for customizing the request. Each option has a reasonable default value, which the user can change to adjust obsipyDMT option flags to a specific request. The following command displays all available options with their default values:

```
obsipyDMT --help
```

The options are grouped by topics. To display only a list of these topic headings, use

```
obsipyDMT --options
```

To see the full help text for only one topic (e.g., group 2), use

```
obsipyDMT --list_option 2
```

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. We thank Piero Poli, Riccardo Zaccarelli, Frédéric Dubois and editor Charlotte Krawczyk for their careful and constructive reviews. We are grateful to Joachim Wassermann for detailed discussions on instrument correction. We thank Piero Poli and Lion Krischer for valuable ideas and discussions on obspyDMT functionalities. All waveform data used for the examples came from the IRIS, ORFEUS, GFZ and RESIF data management centers. Kasra Hosseini was funded by Deutsche Forschungsgemeinschaft (DFG) grants made to Karin Sigloch (grant numbers SI 1538/1-1, in Priority Programme SAMPLE, and SI 1538/2-1, project RHUM-RUM). The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement no. PCIG14-GA-2013-631104 RHUM-RUM, and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 639003 DEEP TIME). We acknowledge discussions within TIDES COST Action ES1401.

Edited by: CharLotte Krawczyk

Reviewed by: Piero Poli and Riccardo Zaccarelli

References

- Ahern, T., Casey, R., Barnes, D., Benson, R., and Knight, T.: SEED Reference Manual—Standard for the Exchange of Earthquake Data, International Federation of Digital Seismograph Networks Incorporated Research Institutions for Seismology United States Geological Survey, version 2.4 Edn., 2012.
- Barruol, G. and Sigloch, K.: Investigating La Réunion hot spot from crust to core, *Eos, Transactions American Geophysical Union*, 94, 205–207, 2013.
- Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., and Wassermann, J.: ObsPy: A Python Toolbox for Seismology, *Seismol. Res. Lett.*, 81, 530–533, 2010.
- Davy, C., Barruol, G., Fontaine, F. R., Sigloch, K., and Stutzmann, E.: Tracking major storms from microseismic and hydroacoustic observations on the seafloor, *Geophys. Res. Lett.*, 41, 8825–8831, 2014.
- Dziewonski, A. and Anderson, D.: Preliminary reference Earth model, *Phys. Earth Planet. In.*, 25, 297–356, 1981.
- FDSN: FDSN StationXML Schema, available at: <http://www.fdsn.org/xml/station/>, last access: 27 March 2015.
- George Helffrich, J. W. and Bastow, I.: *The Seismic Analysis Code*, Cambridge University Press, 2013.
- Groos, J.: Broadband seismic noise: classification and Green's function estimation, PhD thesis, Karlsruhe Institute for Technology, 2010.
- Hosseini, K.: Global multiple-frequency seismic tomography using teleseismic and core-diffracted body waves, PhD thesis, LMU, 2016.
- Hosseini, K. and Sigloch, K.: Multifrequency measurements of core-diffracted P waves (Pdiff) for global waveform tomography, *Geophys. J. Int.*, 203, 506–521, <https://doi.org/10.1093/gji/ggv298>, 2015.
- Hosseini, K., Sigloch, K., and Stähler, S.: Finite Frequency Measurements of Conventional and Core-diffracted P-waves (P and Pdiff), *AGU Fall Meeting Abstracts*, 1, 4450, 2014.
- Hunter, J. D.: Matplotlib: A 2D graphics environment, *Comput. Sci. Eng.*, 9, 90–95, 2007.
- Hutko, A. R., Bahavar, M., Trabant, C., Weekly, R. T., Van Fossen, M., and Ahern, T.: Data Products at the IRIS-DMC: Growth and Usage, *Seismol. Res. Lett.*, 88, 892–903, 2017.
- Jones, E., Oliphant, T., and Peterson, P.: SciPy: Open source scientific tools for Python, available at: <http://www.scipy.org/> (last access: 1 April 2017), 2001.
- Kennett, B. and Engdahl, E.: Traveltimes for global earthquake location and phase identification, *Geophys. J. Int.*, 105, 429–465, 1991.
- Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., and Wassermann, J.: ObsPy: A Bridge for Seismology into the Scientific Python Ecosystem, *Computational Science & Discovery*, 8, 014003, <https://doi.org/10.1088/1749-4699/8/1/014003>, 2015.
- Krischer, L., Hutko, A. R., van Driel, M., Stähler, S., Bahavar, M., Trabant, C., and Nissen-Meyer, T.: On-demand custom broadband synthetic seismograms, *Seismol. Res. Lett.*, <https://doi.org/10.1785/0220160210>, online first, 2017.
- Megies, T., Beyreuther, M., Barsch, R., Krischer, L., and Wassermann, J.: ObsPy – What Can It Do for Data Centers and Observatories?, *Anna. Geophys.*, 54, 47–58, <https://doi.org/10.4401/ag-4838>, 2011.
- Morozov, I. B. and Pavlis, G. L.: Management of large seismic datasets: I. Automated building and updating using BREQ_FAST and NetDC, *Seismol. Res. Lett.*, 82, 211–221, 2011a.
- Morozov, I. B. and Pavlis, G. L.: Management of large seismic datasets: II. Data center-type operation, *Seismol. Res. Lett.*, 82, 222–226, 2011b.
- Nissen-Meyer, T., van Driel, M., Stähler, S. C., Hosseini, K., Hempel, S., Auer, L., Colombi, A., and Fournier, A.: AxisEM: broadband 3-D seismic wavefields in axisymmetric media, *Solid Earth*, 5, 425–445, <https://doi.org/10.5194/se-5-425-2014>, 2014.
- Scheingraber, C., Hosseini, K., Barsch, R., and Sigloch, K.: ObsPyLoad: A Tool for Fully Automated Retrieval of Seismological Waveform Data, *Seismol. Res. Lett.*, 84, 525–531, <https://doi.org/10.1785/0220120103>, 2013.
- Scherbaum, F.: *Of poles and zeros*, vol. 15, Springer Science & Business Media, 1996.
- Shapiro, N. M. and Campillo, M.: Emergence of broadband Rayleigh waves from correlations of the ambient seismic noise, *Geophys. Res. Lett.*, 31, 107614, <https://doi.org/10.1029/2004GL019491>, 2004.
- Sigloch, K.: Multiple-frequency body-wave tomography, PhD thesis, Princeton University, 2008.
- Stähler, S. C., Sigloch, K., Hosseini, K., Crawford, W. C., Barruol, G., Schmidt-Aursch, M. C., Tsekhmistrenko, M., Scholz, J.-R., Mazzullo, A., and Deen, M.: Performance report of the RHUM-RUM ocean bottom seismometer network around La Réunion, western Indian Ocean, *Adv. Geosci.*, 41, 43–63, <https://doi.org/10.5194/adgeo-41-43-2016>, 2016.

- Owens, T. J., Crotwell, H. P., and Oliver-Paul, P.: SOD: Standing Order for Data, *Seismol. Res. Lett.*, 75, 515–520, 2004.
- van der Walt, S., Colbert, S., and Varoquaux, G.: The NumPy Array: A Structure for Efficient Numerical Computation, *Comput. Sci. Eng.*, 13, 22–30, <https://doi.org/10.1109/MCSE.2011.37>, 2011.
- van Driel, M., Krischer, L., Stähler, S. C., Hosseini, K., and Nissen-Meyer, T.: Instaseis: instant global seismograms based on a broadband waveform database, *Solid Earth*, 6, 701–717, <https://doi.org/10.5194/se-6-701-2015>, 2015.
- West, J. D. and Fouch, M. J.: EMERALD: A web application for seismic event data processing, *Seismol. Res. Lett.*, 83, 1061–1067, 2012.
- Whitaker, J.: Installing; Basemap Matplotlib Toolkit 1.0.8 documentation, available at: <http://matplotlib.org/basemap/users/installing.html>, last access: 27 March 2015.
- Zaroli, C., Lambotte, S., and Lévêque, J.-J.: Joint inversion of normal-mode and finite-frequency S-wave data using an irregular tomographic grid, *Geophys. J. Int.*, 203, 1665–1681, 2015.