Geophysical Journal International

Geophys. J. Int. (2016) **207,** 1003–1011
Advance Access publication 2016 August 25
GJI Seismology

doi: 10.1093/gji/ggw319

An Adaptable Seismic Data Format

Lion Krischer,[1] James Smith,[2] Wenjie Lei,[2] Matthieu Lefebvre,[2] Youyi Ruan,[2]
Elliott Sales de Andrade,[3] Norbert Podhorszki,[4] Ebru Bozdağ[5] and Jeroen Tromp[2,6]

[1] *Department of Earth and Environmental Sciences, Ludwig-Maximilians-University, Munich, Germany. E-mail: krischer@geophysik.uni-muenchen.de*
[2] *Department of Geosciences, Princeton University, Princeton, NJ, USA*
[3] *Department of Physics, University of Toronto, Toronto, Canada*
[4] *Oak Ridge National Laboratory, Oak Ridge, TN, USA*
[5] *Géoazur, University of Nice Sophia Antipolis, Valbonne, France*
[6] *Program in Applied & Computational Mathematics, Princeton University, Princeton, NJ, USA*

Accepted 2016 August 22. Received 2016 August 9; in original form 2016 March 31

SUMMARY

We present ASDF, the Adaptable Seismic Data Format, a modern and practical data format for all branches of seismology and beyond. The growing volume of freely available data coupled with ever expanding computational power opens avenues to tackle larger and more complex problems. Current bottlenecks include inefficient resource usage and insufficient data organization. Properly scaling a problem requires the resolution of both these challenges, and existing data formats are no longer up to the task. ASDF stores any number of synthetic, processed or unaltered waveforms in a single file. A key improvement compared to existing formats is the inclusion of comprehensive meta information, such as event or station information, in the same file. Additionally, it is also usable for any non-waveform data, for example, cross-correlations, adjoint sources or receiver functions. Last but not least, full provenance information can be stored alongside each item of data, thereby enhancing reproducibility and accountability. Any data set in our proposed format is self-describing and can be readily exchanged with others, facilitating collaboration. The utilization of the HDF5 container format grants efficient and parallel I/O operations, integrated compression algorithms and check sums to guard against data corruption. To not reinvent the wheel and to build upon past developments, we use existing standards like QuakeML, StationXML, W3C PROV and HDF5 wherever feasible. Usability and tool support are crucial for any new format to gain acceptance. We developed mature C/Fortran and Python based APIs coupling ASDF to the widely used SPECFEM3D_GLOBE and ObsPy toolkits.

Key words: Time-series analysis; Seismic tomography; Computational seismology; Wave propagation.

1 INTRODUCTION

1.1 Motivation

Seismology is, to a large extent, a science driven by observing, modelling, and understanding data. The process of making discoveries from data requires simple, robust, and fast processing and analysis tools, empowering seismologists to focus on actual science. Modern seismological workflows assimilate data on an unprecedented scale, and the need for efficient processing tools is pressing. In this context, the format in which data is stored and exchanged plays a central role. For example, passive seismic data are commonly stored in such a way that each time-series corresponds to a single file on the file system. The amount of I/O required to process and assimilate data stored this way quickly becomes debilitating on modern HPC platforms. As another example, simulated seismograms depend on a large number of input parameters, particular versions of modelling software, and specific run-time execution commands. A modern data format should strive for complete reproducibility by keeping track of such data provenance. The majority of existing seismic data formats were created in a more primitive computing era, when no one could have foreseen the size, complexity, and challenges that seismological data sets must accommodate today. New seismological techniques, such as interferometry and adjoint tomography, require access to very large computers, where I/O poses a major bottleneck and data mining and feature extraction are challenging.

In this article we introduce a new data format—the Adaptable Seismic Data Format (ASDF)—designed to meet these challenges. We are fully aware of the fact that the introduction of yet another seismic data format should ideally be avoided. However, we believe it to be justified because the current state of the art is just not good enough. We further believe that the advantages of the proposed format are significant enough to quickly outweigh the initial difficulties of switching to a new format. We identify five key issues that a new data format must resolve, namely:

(i) **Efficiency:** Data storage is cheap, but data operations are increasingly becoming the limiting factor in modern scientific workflows. More efficient and better performing data processing and analysis tools are badly needed.

(ii) **Data organization:** Different types of data (waveforms, source & receiver information, derived data products such as adjoint sources, receiver functions, and cross-correlations) are needed to perform a variety of tasks. This results in ad hoc data organization and formats that are hard to maintain, integrate, reproduce and exchange.

(iii) **Data exchange:** In order to exchange complex data sets, an open, well-defined and community driven data format must be developed.

(iv) **Reproducibility:** A critical aspect of science is the ability to reproduce results. Modern data formats should facilitate and encourage this.

(v) **Mining, visualization and understanding of data:** As data volumes grow, more complex, new techniques to query and visualize large data sets are needed.

The ultimate goal is to empower seismologists to focus on actual science. This is the time for the community to build an organized, high-performance and reproducible seismic data format for seismological research. In order to facilitate integration of the new format into existing scientific workflows and to demonstrate that this is not just an academic exercise, we developed a Python library hooking ASDF into the ObsPy library (Beyreuther *et al.* 2010), which, as a hugely beneficial side-effect, also takes care of any data format conversion issues, be it to, or from ASDF. A C-based ASDF library features an API for reading and writing ASDF files and includes examples in both C and Fortran. Embedding this library in the widely used spectral-element waveform solver SPECFEM3D_GLOBE (Komatitsch & Tromp 2002a,b) made it gain native support for ASDF-integrated workflows. To engage and educate the community, a wiki provides demonstrations of the format and includes technical and non-technical introductions for both users and developers.

1.2 Scope

The proposed Adaptable Seismic Data Format is designed to be an efficient, self-describing data format for storing, processing, and exchanging seismological data, including full provenance information. It is intended to be used by researchers and analysts working with data, after it has been recorded. In contrast, it is not aspiring to replace the time proven MiniSEED format for data archival, streaming, and low-latency applications. These use cases are contrary to a comprehensive and self-describing data format and both can probably not be achieved simultaneously.

ASDF is applicable to a large number of areas in seismology and related sciences. Its use ranges from classical earthquake seismology to active source data sets, ambient seismic noise studies, and GPS time-series. Furthermore, it is generic enough to accommodate any kind of derived or auxiliary data that might accrue in the course of a research project.

1.3 Benefits

A well-defined format with the previously listed attributes directly results in a number of advantages and applicable use cases. In this section we list a few of these, in no particular order.

(i) Seismological data sets usually contain waveform data as well as associated meta data, such as information about events and stations. All this data needs to be integrated and accessed concurrently, which requires a large amount of bookkeeping as data sets grow. Consequently, many tools are one-off scripts that cannot be reused for subsequent projects. Additionally, data sets become difficult to share with research groups that do not employ the same internal structure and data organization. Over the years, numerous groups have developed customized seismological data formats to work around these limitations. In contrast, ASDF is a well-defined format that can be used to store and exchange full seismological data sets, including all necessary meta information.

(ii) It is oftentimes convenient to locally build up a database of pre-processed waveforms. A common example is storage of instrument corrected and bandpass filtered data. If a project continues for some years, it might ultimately no longer be known how exactly data were processed. The makeup of the team may have changed, or perhaps the processing software had a bug that has been fixed in the meantime, and this may or may not have affected the data. *Provenance*, that is, the tracking and storing of the history of data, solves this particular problem, and ASDF accommodates that. Existing data formats do not (or only in a very limited manner) track the origin of data and what operations were performed on it due to limited and inflexible metadata allowances. ASDF is capable of storing the full provenance graph that resulted in a particular piece of waveform or other data.

(iii) For the first time, ASDF accommodates proper storage and exchange of synthetic seismograms, including information about the numerical solver, earthquake parameters, the Earth model and all other parameters influencing the final result. Waveform simulations at high frequencies and in physically plausible Earth models are extremely expensive computationally, so preserving and carefully documenting such simulations is of tremendous value.

(iv) ASDF greatly reduces the number of files necessary for many tasks, because a single ASDF file can replace tens to hundreds of thousands of single waveform files. Beside raw performance and organizational benefits, this also facilitates workflows that run into hard file count quota limits on supercomputers. Please note that ASDF can store data from very many receivers as well as arbitrarily long time-series from only a single receiver and any combination in between.

(v) Importantly, ASDF offers efficient parallel I/O on modern clusters with the required hardware. This facilitates fully parallel data processing workflows that actually scale.

(vi) ASDF offers optional and automatic lossless data compression, thereby reducing file size.

(vii) Seismograms are certainly not the only type of data used in seismology. Other data types, including various spectral estimations, cross-correlations, adjoint sources, receiver functions, and so on, also benefit from organized and self-describing storage.

ASDF is intended as a container for all the various kinds of data materializing in seismological research, including all required meta information. Additionally, each piece of data should be able to describe itself and what led to it. Having an organized and standard data container will, in the long run, increase the speed and accuracy of seismic research, and provides a medium for effectively communicating research results. The remainder of this article is structured as follows: We first provide an overview of the layout of the format and justify some choices that needed to be made. We then compare the ASDF format to existing data formats in use in seismology, thereby further justifying its development. Finally, we showcase a number of existing implementations, detail several use cases for the ASDF format, and discuss future possibilities. The article is intentionally light on technical details to focus on a high-level view. A technical definition of the ASDF format can be found online.

2 OVERVIEW OF THE FORMAT

ASDF, at its most basic level, organizes its data in a hierarchical structure inside a container—in a simplified manner a container can be pictured as a file system within a file. The contents are roughly arranged in four sections, as follows.

(i) Details about seismic events of any kind (earthquakes, mine blasts, rock falls, etc.) are stored in a QuakeML document.

(ii) Seismic waveforms are sorted in one group per seismic station together with meta information in the form of a StationXML document. Each waveform is stored as an HDF5 array.

(iii) Arbitrary data that cannot be understood as a seismic waveform is stored in the auxiliary data section.

(iv) Data history (provenance) is kept as a number of SEIS-PROV documents (an extension to W3C PROV).

Existing and established data formats and conventions are utilized wherever possible. This keeps large parts of ASDF conceptually simple, and delegates pieces of the development burden to existing efforts. The ASDF structure is summarized in Fig. 1 and is discussed in more detail in the following paragraphs. It is worth noting that almost everything is optional. The amount of stored information can thus be adapted to any given use case.

2.1 Container

Large parts of the ASDF definition are independent of the employed container format. An advantage of this approach is a certain resilience to technological changes as major pieces of ASDF can in theory be adapted to other container formats. Nonetheless, the container format has to be fixed to not severely affect interoperability and ease of data exchange. We evaluated a number of possibilities and chose HDF5 (Hierarchical Data Format version 5; The HDF Group 1997–2015). It is used in a wide variety of scientific projects and has a healthy and active ecosystem of libraries and tools. NetCDF 4 (Rew & Davis 1990) is implemented on top of HDF5 and ASDF does not gain from the additional functionality. While not being as fast as ADIOS (Liu *et al.* 2014) for the most extreme use cases, HDF5 also fulfils our hard requirement of being capable of efficient parallel I/O with MPI (message passing interface; MPI Forum 2009). It can be argued that seismology does not have to deal with the same amount of data as, for example, particle physics or biology, where single data sets can easily attain volumes of multiple petabytes (Bird *et al.* 2014; Stephens *et al.* 2015). At the time of writing, the HDF5 libraries work on more platforms and have more users as well as available tools, which we believe is well worth the minor loss in maximum potential I/O performance. Using HDF5 also grants a number of useful features (other formats also offer some or all of them): First, there is no need to worry about the endianness of data, which historically has been a big issue in seismology. Second, HDF5 has a number of built-in data compression algorithms and data corruption tests in the form of check summing.

2.2 Seismic event information

Information about all kinds of seismic events, including earthquakes, building collapses, fluid injections, and so on, are stored in a single QuakeML (Schorlemmer *et al.* 2004, 2011) file inside the container. QuakeML is an XML (Bray *et al.* 2008) representation intended for different types of seismological meta information, but is in practice mostly used to describe earthquakes.

Note that one QuakeML document can describe an arbitrary number of events in a comprehensive manner. It is the de-facto standard for defining seismic events, adopted as a standard by the International Federation of Digital Seismograph Networks (FDSN, https://www.fdsn.org), and widely available, because it is served by web services of data centres around the world. A crucial capability is that it can specify a number of different hypocentres and focal mechanisms for each individual event, which might be the results from different source inversion algorithms. Each of these is identified by a unique id. ASDF uses these identifiers to, for example, determine the exact moment tensor and event location that was used to simulate an event that resulted in a particular waveform.

Shortcomings of the latest QuakeML version at the time of writing include no proper possibility for storing either finite fault sources or custom source time functions. This might be alleviated in future QuakeML versions, at which point ASDF also gains that functionality. As of now, both could either be stored in custom elements in a QuakeML document in a separate namespace, or as part of the auxiliary data section of ASDF files.

The exploration community employs seismic sources that cannot be appropriately described by the QuakeML standard. Nonetheless the concept of having detailed descriptions of seismic sources naturally translates to the active source case. It is conceivable that a standard for describing these sources might appear in the future at which point it can be incorporated into ASDF. In the use cases section we demonstrate how that could be achieved.

2.3 Waveforms and station meta information

At the heart of ASDF is the waveform data. A single file can store any number, combination and length of waveform data. Waveforms are restricted to single and double precision floating point and signed integer data and are stored as HDF5 native data arrays. These data arrays are logically grouped by using four codes: the network code denotes the operator of a seismological network, the station code denotes a station within that network, the location code denotes a particular instrument at a station, and finally the channel code denotes the recording component. These codes are often called SEED (Standard for the Exchange of Earthquake Data) compatible identifiers (Incorporated Research Institutions for Seismology (IRIS) 2012) and, together with some temporal information, allow the unique identification of seismic instruments and are also used in the QuakeML and StationXML standards.

ASDF organizes waveforms and associated meta information at a station level granularity. Other choices would have been possible, but this provides a certain balance between the necessary nesting and the number of elements per group (like a directory in HDF5 terms). Each station can optionally contain a StationXML document made up of meta information for one or more channels of that station. StationXML is the current FDSN standard for station

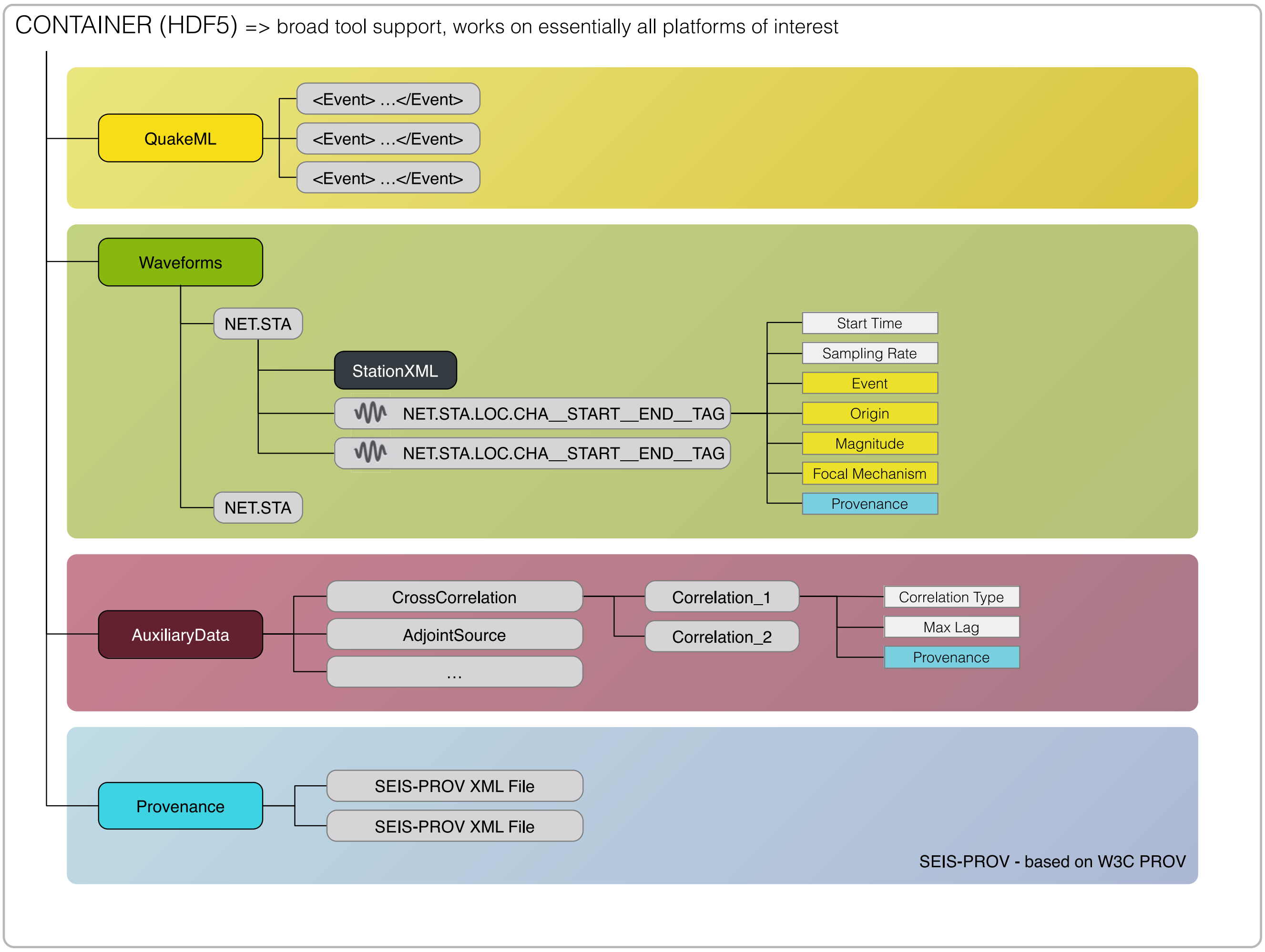

Figure 1. The general structure of an ASDF file in its HDF5 container—it has four distinct parts: (1, yellow) Information about an arbitrary number of earthquakes (or other seismic events) is stored in a single QuakeML document, the most complete earthquake description format currently available. (2, green) Seismic waveforms are stored per station together with the necessary meta information in the form of an FDSN StationXML document. (3, red) Anything that cannot be regarded as a seismic waveform is hierarchically stored in the auxiliary data section. (4, blue) Provenance information is stored as a number of SEIS-PROV documents, an extension to W3C PROV. Background colours in the attributes (rectangular boxes) denote relations to other sections in an ASDF file. Examples of this are relations of a waveform to a certain event or a provenance record for a piece of auxiliary data.

information and the successor of the SEED standard. Roughly speaking, it contains information about who runs a network and deployed the station, about the geographical and geological setting of the station, and the impulse response of each recording channel. This is vital information, and storing it alongside the actual waveform data eases many common undertakings. A StationXML document can contain as much or as little information as appropriate for any given task. A further benefit is that StationXML can also be used to describe non-seismological time-series, such as pressure and temperature curves.

The waveform data are stored as pieces of continuous, well-behaved time-series data. Each piece, in the following called a trace, consists of a start time, a sampling rate, and a data array representing regularly sampled data. The starting time of each trace is internally represented as a nanosecond precision UNIX epoch time. The use of a 64-bit integer grants a temporal range from about the year 1680 to 2260, which is sufficient for all envisioned use cases. Times are always in UTC in accordance with most other seismological data and file formats.

Every station can contain an arbitrary number of traces consisting of data from multiple locations and channels. Each trace is named according to the following scheme:

`NET.STA.LOC.CHA__STARTTIME__ENDTIME__TAG`

`NET`, `STA`, `LOC` and `CHA` are placeholders for the network, station, location and channel codes. `STARTTIME` and `ENDTIME` are string representations of the start and end time of the trace. The final `TAG` part serves as another hierarchical layer. The need for this layer becomes obvious, for example, when attempting to store data from two waveform simulations but with a slightly different Earth model. They need to be given different names—a randomized string would have been possible, but human readable tags seem to be a nicer alternative. Unprocessed data straight from a digitizer are, by convention, given the tag `raw_recording`; other tags will always depend on the use case. Traces may have any length without inhibiting the ability to work with them. Incidentally, HDF5 supports reading portions of an array which enables users to read only portions of very long time-series within an ASDF file.

Real world data is not perfect, and seismic receivers can fail and thus produce gaps or overlaps in data. Many existing file formats have no concept of this and thus require workarounds. In ASDF a gap is represented by one trace before and another trace after the gap and two overlapping traces denote an overlap. This construct has proven itself to work very well in practice and is also employed in the MiniSEED format as well as the ObsPy library.

Last but not least, each trace potentially also carries some more meta information and relations to other places within an ASDF file. These are elaborated upon in a later section.

ASDF's construction is not a perfect fit for active source exploration data, which is mainly a consequence of the chosen nesting structure and StationXML heavily leaning towards passive source and station based seismology. Most branches of seismology, however, work with the concept of sources and receivers. Thus we encourage the exploration community to come up with a general definition of their receivers, at which point it can be integrated into ASDF with only a minor effort.

2.4 Auxiliary data

Seismologists are used to working with waveform data so they oftentimes exploit the same formats for other data. Receiver functions, cross-correlations, and H/V stacks are all examples of this reuse. Header fields of the format are then used to store some limited amount of meta information. This becomes problematic if that data should be archived for future generations of researchers or exchanged with the wider community. Within the ASDF format this type of data is referred to as auxiliary data, and can be anything that is not considered a seismic waveform. Conceptually, each piece of auxiliary data is stored in an arbitrarily nested path in the auxiliary data group and consists of a data array of any dimension and any necessary meta information in a key-value representation.

ASDF does not define auxiliary data in more detail on purpose. On the one hand, many areas of seismology where the concept of auxiliary data is interesting are in a heavy state of flux and are seeing a lot of active research. It is often unclear what to store and keep track of and that view constantly evolves. On the other hand, we are not experts in all areas of seismology, and it would take a long time to agree on what needs to be stored for each type of auxiliary data.

Over time, we hope that conventions for certain types of data, such as cross-correlations, will become established by the wider community. Nonetheless, ASDF allows for arbitrary and descriptive meta information for any type of data to explain what the data actually are. This becomes particularly powerful when combined with the provenance information, which is described next.

2.5 Provenance

Reproducibility is frequently discussed and widely recognized as a critical requirement of scientific results. In practice, it is so difficult and time consuming to achieve that it is frequently just ignored. Provenance is the process of keeping track of and storing all constituents of information that were used to arrive at a certain result or a particular piece of data. This information is then used to judge the quality and trustworthiness of the results. While not being identical to reproducibility, the concept of provenance is a key ingredient towards this goal.

Each piece of waveform and auxiliary data within ASDF can optionally store provenance information in the form of a W3C PROV or SEIS-PROV document. The implications of this are that ASDF can store any piece of observed, processed, derived, or synthetic data with full provenance information. Thus, such a file can be safely archived and exchanged with others, and information that led to a certain piece of it is readily available. It is important to note that SEIS-PROV only documents the processes that led to a certain piece of data. It does not, by default, store the actual data at each intermediate step, although this could also be achieved within the ASDF format.

W3C PROV is a data model to describe provenance, and SEIS-PROV is a domain-specific extension for using W3C PROV in the context of seismological data processing and generation. We quickly introduce SEIS-PROV as it is a critical component of ASDF; the motivation and reasoning behind it will be detailed in a separate publication.

Provenance can be described from different points of view. SEIS-PROV employs a process-centred provenance description that aims to capture all actions taken to arrive at a certain piece of data. That is a natural fit for seismological data processing. In a nutshell, it works by describing things or entities which (in the context of seismology) might be waveform traces or cross-correlation stacks at different stages in a processing chain. These representations are then connected by so called activities that can use existing entities and create new ones. A simple example of an activity is a filter in signal processing that takes an existing waveform trace and produces a new, filtered one. Additionally, all entities and actions can be assigned to agents that are responsible for it. Agents are usually persons or software programs. Fig. 2 illustrates these concepts with a simplistic example.

The goal of the provenance descriptions in ASDF is that scientists looking at data described by it should be able to tell what steps were taken to generate that particular piece of data.

ASDF only takes cares of the storage of the provenance information. In practice, provenance will only be generated and used if it is captured and stored in a fully automatic fashion and is thus strongly dependent on the software used to generate and process data.

2.6 Data relations

Data always needs to be regarded and interpreted in a wider context. This ranges from information about the origin of the data, which is dealt with in the previous section, to relations to other pieces of data. Classical relations in seismology are waveform data and information about the recording site and instrument, as well as the sources of the recorded wavefield.

Any time different pieces of data are required that are stored in varying places, formats, and files, the required bookkeeping to make workflows run can be substantial. ASDF greatly eases that pain by storing everything in a well-defined place within the same file. The need to find and assemble the different pieces can thus be performed by software, thereby requiring less mental work from scientists. ASDF, as shown in the previous sections, can store waveforms, events, station meta information, provenance, and auxiliary data all in the same file. Additionally, it permits relations between these items. For example, each waveform trace can be associated with a certain event, or a certain event origin or focal mechanism. Relations for each block of data to its provenance record are also retained.

All in all this allows for fully self-explanatory, complete data sets preserving complex internal relations. This is something that is constantly required in scientific and data driven applications. Today, people usually deal with this by using project-specific

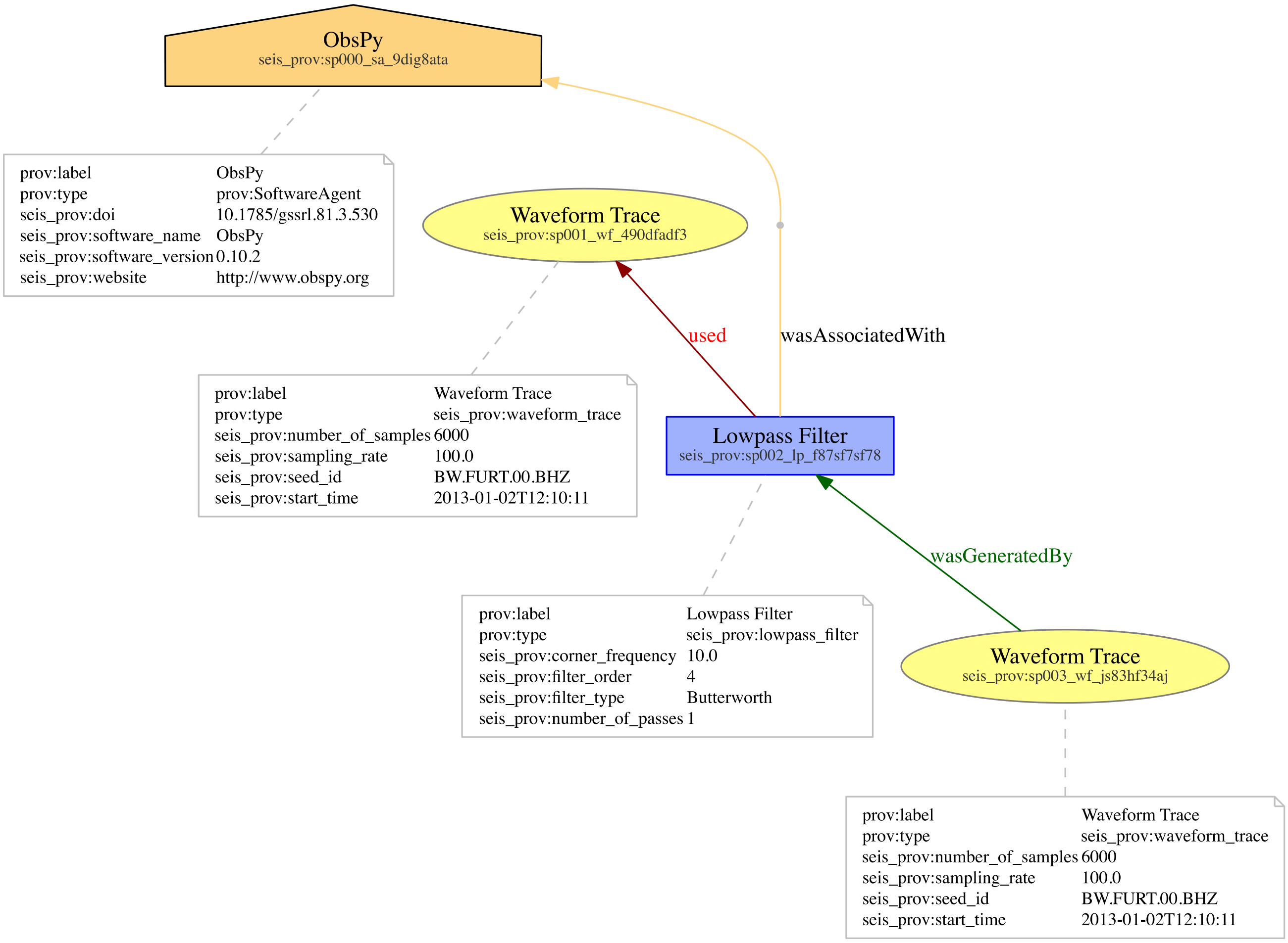
ObsPy
seis_prov:sp000_sa_9dig8ata
prov:label ObsPy
prov:type prov:SoftwareAgent
seis_prov:doi 10.1785/gssrl.81.3.530
seis_prov:software_name ObsPy
seis_prov:software_version 0.10.2
seis_prov:website http://www.obspy.org
Waveform Trace
seis_prov:sp001_wf_490dfadf3
used
wasAssociatedWith
prov:label Waveform Trace
prov:type seis_prov:waveform_trace
seis_prov:number_of_samples 6000
seis_prov:sampling_rate 100.0
seis_prov:seed_id BW.FURT.00.BHZ
seis_prov:start_time 2013-01-02T12:10:11
Lowpass Filter
seis_prov:sp002_lp_f87sf7sf78
wasGeneratedBy
prov:label Lowpass Filter
prov:type seis_prov:lowpass_filter
seis_prov:corner_frequency 10.0
seis_prov:filter_order 4
seis_prov:filter_type Butterworth
seis_prov:number_of_passes 1
Waveform Trace
seis_prov:sp003_wf_js83hf34aj
prov:label Waveform Trace
prov:type seis_prov:waveform_trace
seis_prov:number_of_samples 6000
seis_prov:sampling_rate 100.0
seis_prov:seed_id BW.FURT.00.BHZ
seis_prov:start_time 2013-01-02T12:10:11

Figure 2. Simple example to illustrate the key concepts of storing provenance information with SEIS-PROV and W3C PROV. It describes a single waveform trace that has been low-pass filtered to create a filtered waveform trace. The arrows in this graphical representation mostly point backwards in the process towards the origin of something. The yellow ellipses are called entities, and here they represent a waveform trace at two different points in time. The blue rectangle is an activity that can use and generate entities. It denotes a low-pass filter and uses the first waveform trace to generate a new, filtered waveform trace. The orange house shape symbolizes an agent who is responsible for something. In this case, it stands for the software that performed the filtering operation. Finally, the white rectangles are attributes with more details about any node. Please note that this figure shows only one possible graphical representation of the underlying data model and more or less detailed ones can be employed as appropriate.

directory structures that cannot be exchanged nor properly archived, and ASDF clearly improves that system on all fronts.

3 COMPARISON TO EXISTING FORMATS

Having yet another format induces more complexity and, potentially, noise into the community using that type of data and the landscape of software able to deal with it. 'Do we really need a new format?' is thus a natural and understandable question. This sections addresses why no single existing data format in seismology is able to satisfy our needs and thus justifies the introduction of the ASDF format.

We limit ourselves to detailing alternative waveform formats as we directly incorporate the StationXML and QuakeML formats and no true alternative to storing derived data or provenance is currently in existence. A wide variety of different seismological data formats is used by researchers world wide. We will discuss the most widely used ones, namely, (Mini)SEED, SAC, and SEG Y/PH5. Please see Bormann (2012) and Havskov (2010) for additional information and descriptions of more formats.

3.1 MiniSEED

SEED was developed in the late 1980s and at least the data-only part (MiniSEED) continues to be in wide use today, and will likely continue to be the dominant data streaming and archival format for the foreseeable future. The ASDF format does not attempt to replace it. Some of MiniSEED's features, such as the ability to build up large data volumes by concatenating small and short pieces, are very well suited for their use in data archives, where data is constantly streamed in. While the full SEED format can in theory store waveforms as well as station meta information, the complexity of the format hinders that. It furthermore can only properly store raw waveform recordings and no event information. Additionally, the dataless part of SEED, e.g., the part with the station information, sees declining usage nowadays with that responsibility being taken over by StationXML. MiniSEED, on the other hand, is more than capable of storing arbitrarily large waveform volumes, but the file then contains no index of what is in it, so one must always read the entire file to figure that out, making large data volumes fairly impractical. Additionally, the amount of meta information in MiniSEED files is strongly limited, so one always needs additional files to work with it.

Summing up, MiniSEED is a good data archival format for data centres, streaming and low-latency applications, but it is not well suited for the later research and processing stages, where ASDF has significant advantages.

3.2 SAC

The Seismic Analysis Code (SAC, Helffrich *et al.* 2013) introduced a new format named after its parent program, and is still in widespread use today. This is likely due to two reasons: the popularity of the SAC program itself and the relative simplicity of the format with a number of header fields that can be adapted to different purposes.

The SAC format is well suited for many tasks, but ASDF offers a number of advantages. The most obvious ones are the ability to store multiple components—including gaps and overlaps—in a single file without awkward workarounds, as well as the potential to create full data sets incorporating all necessary meta information. ASDF is, for large workflows, also more efficient, facilitates the storage of different data types—integers as well as floats—and, with the help of HDF5 offers file compression and check summing.

The combination of these factors results in ASDF being more suitable and convenient for many workflows. Some, for example experiments with millions of waveform files, are almost impossible without a more advanced seismological data format. In fact, part of the motivation for developing ASDF stems from the fact that reading and writing SAC files for a large tomographic inversion practically brings a huge parallel file system to its knees due to the very large number of involved files.

3.3 SEG Y and PH5

The SEG Y Data Exchange Format (SEG Technical Standards Committee 2002) is one of many in the family of data formats introduced and defined by the Society of Exploration Geophysicists (SEG) Technical Standards Committee. Among these, it is probably the most widely known and used. The more modern PH5 (IRIS/PASSCAL Data Group 2012) format has a data model similar to SEG Y, but stores its data in an HDF5 container. This eliminates some limitations of the SEG Y format and facilitates more extensive meta information. It has been developed as an archiving format for active source seismic experiments. Typical workflows extract data from PH5 and save it as SEG Y, which is used in the further stages.

Both on- and off-shore active source data is very structured, meaning that all receivers generally have the same response and record for the same time span with the same sampling rate. Receivers are placed in lines and geo-referenced by relative coordinates. In contrast, passive source seismology is frequently very unstructured, with different receiver types scattered across a geographical region, and the meta information is fairly rich and detailed.

SEG Y and PH5 are well suited for active source experiments, but it is difficult to adapt these formats for passive source seismologists to suit their purposes. Historically, SEG Y is essentially not used in passive source seismology, and there is no reason to expect this will change with PH5. The inverse is true as well, in that passive source seismology tools are rarely used in active studies. A consequence is that the current iteration of the ASDF format is not fully suitable for exploration studies as it relies on certain formats and conventions. In the use cases section we will show an example of how it can still be done.

The concept of seismic sources and receivers nonetheless holds true in both active and passive source seismology. We have the hope that, in the future, ASDF will be used as a standard for both. Active source seismology currently lacks community accepted standards for sources and receivers as is common in passive source seismology with formats like QuakeML and StationXML. Methods, ideas, and techniques are frequently exchanged between these communities, and we encourage the development of these missing standards. A common data format would enable greater sharing of tools, whole workflows, and most importantly human knowledge and skill, greatly benefiting both sides. The ASDF format is ready to incorporate these aforementioned definitions.

4 IMPLEMENTATIONS

We developed three usable implementations of the ASDF format and expect more to follow:

(i) A C library with Fortran bindings to read and write ASDF files. This is for example used in the SPECFEM3D_GLOBE (Komatitsch & Tromp 2002a,b) wave propagation solver.

(ii) A Python library to read, write, and convert ASDF files to a large number of other formats backed by the ObsPy library (Megies *et al.* 2011; Krischer *et al.* 2015).

(iii) A graphical user interface to visually and interactively explore the contents of ASDF files.

Technological advances often make existing codes and tools obsolete in a matter of just a few years, and we anticipate that these implementations will continue to undergo rapid development and expansion.

5 DEMONSTRATIONS AND USE CASES

The proposed ASDF format can be used in a number of different branches in seismology and its success will revolve around its adoption by the seismological community. This section shows some practical applications and benefits of the format.

5.1 Data set building

A data set is the collection of all data necessary for a particular purpose. Examples include waveform data for a number of stations for a particular earthquake, all waveforms from a single array, or data from an active source study. A complete data set also includes information about seismic receivers and sources and therefore contains everything that is needed for a certain task. All of this can be stored in a single ASDF file. Thus, one no longer needs to deal with complicated and custom directory structures. Tools and scripts written to work on larger data sets can work on a defined structure and be exchanged and adapted to new uses more easily.

Aside from facilitating data management this greatly decreases the number of files one has to deal with. Transfer and copy times can be prohibitive when dealing with a large number of small files. One million waveform files can easily take an hour to transfer from a cluster to a personal workstation. Storing all these waveform in about 50 ASDF files reduces the total transfer time to about a minute. Additionally, many clusters and operating systems impose hard file count limits.

HDF5 furthermore grants access to a number of different lossless compression algorithms reducing the size of ASDF files. Fig. 3

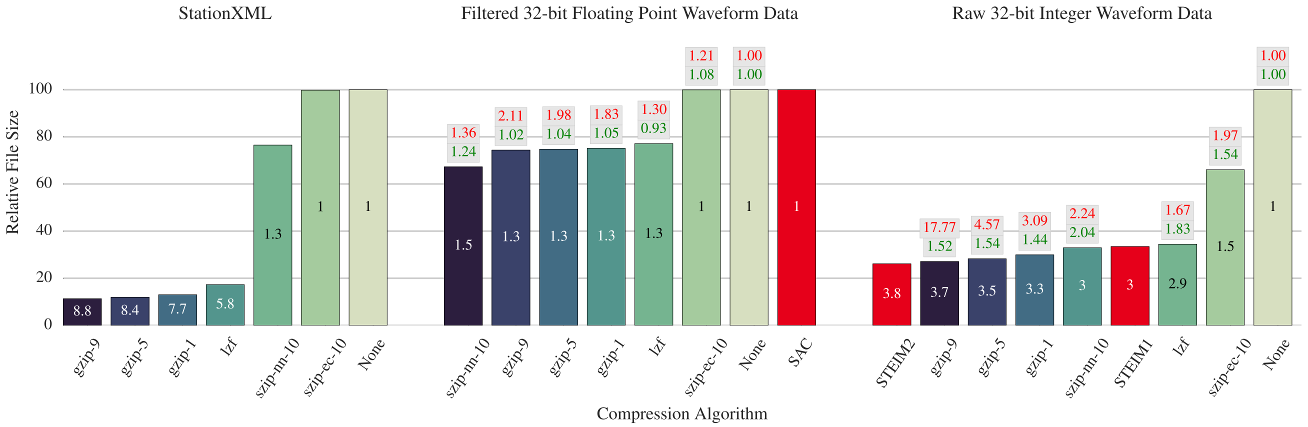

Figure 3. Compression efficiency of the ASDF format using algorithms available in HDF5 for a number of typical seismological data sets. Please keep in mind that the efficiency and I/O speed of these algorithms are heavily dependent on the actual data and hardware, and thus your mileage may vary. The columns represent the file size relative to the uncompressed case, the numbers inside are the achieved compression ratios. The small boxes above the columns denote the relative writing duration in red and the relative reading duration in green compared to the uncompressed case. The left plot shows the efficiency for a data set containing 500 StationXML documents adding up to 120 MiB. I/O speed differences are irrelevant as the cost for parsing and generation of the XML documents is constant and dominates the total run time. The middle plot shows the compression efficiency for a bandpass filtered waveform data set stored as 32-bit floating point numbers. It consists of 3466 waveform traces taking up 282 MiB on disk. The red bar compares it to the uncompressed SAC format. The rightmost plot shows the efficiency for storing 3346 raw waveform files stored as 32-bit integers taking up 2340 MiB. The red bars here show the efficiency for the same data set of the STEIM1 and STEIM2 special purpose compression algorithms defined for the SEED format measured by writing them as MiniSEED files.

shows the efficiency and computational cost of these for a number of typical seismological data sets.

5.2 Storage, exchange and archival of processed or synthetic waveforms

Synthetic seismograms can be very expensive to compute, especially in 3-D media with realistic rheologies. The same is true for some processing chains to, for example instrument correct, and filter data, and it is thus oftentimes worthwhile to preserve these pieces of data.

Their proper long-term storage, exchange and archival are only possible if all processes that went into their creation are documented and stored alongside the data. This includes, for example, the precise version of the used software, and details about all processing steps. For synthetic data it includes the used Earth and source model as well as the waveform solver's settings. ASDF, in combination with SEIS-PROV, preserves all this information.

5.3 (Parallel) large-scale data processing

Data volumes are constantly growing, and the community has access to the computing power needed to process and work with it. However, we are at a point where I/O itself, i.e., reading and writing from and to disk, is one of the most expensive parts of many operations. This is especially true for a very large number of typically small files, as previously pointed out. ASDF, with the help of HDF5, supports efficient, parallel I/O on full data sets. Our implementations shown in the previous section make use of this and facility the construction of fully parallel workflows.

Applications for this operational approach are numerous and in the following we illustrate this on an example occurring in large-scale full waveform inversions using adjoint techniques (Tromp *et al.* 2005; Fichtner *et al.* 2006; Tape *et al.* 2010), but the general concepts translate to other types of workflows using large amounts of data. This iterative procedure requires routine comparison of millions of waveform traces. We replaced an implementation based on the SAC package and file format (Helffrich *et al.* 2013) with an ASDF centred implementation. The SPECFEM3D_GLOBE waveform solver (Komatitsch & Tromp 2002a,b) directly produces ASDF files which are then tied into a single cohesive workflow relying on the ObsPy (Beyreuther *et al.* 2010) package. All components integrate with each other and stream data from one unit to the next. I/O only happens at the very beginning and the end.

These changes empower us to increase the scale of our inversions—in terms of frequency content, number of earthquakes and number of stations—and fully exploit modern computational platforms. Additionally, they reduce the complexity of operations and thus stabilize them. Last but not least, provenance information is kept to increase reproducibility and for future reference.

5.4 Active source industry data set

Industry data sets are not the primary focus of the ASDF format, but it is worthwhile proposing how we could adapt the format to that particular case in an effort to bring the active and passive communities closer together. Active source data is more structured and array like, both in sources and receivers. As the industry currently lacks standards to describe these, we utilized the QuakeML and StationXML formats with some extensions to for example share the array configuration and source time functions. Waveforms are grouped by recording instrument—one network corresponds to one receiver layout.

We believe the industry would benefit from adopting ASDF, since the format offers improved data organization, simple but efficient parallel processing, and provenance capabilities all wrapped up in a modern format. Please see Section 3.3 for some more concrete suggestions and requirements.

5.5 Further uses

The extraction of information from recordings of ambient seismic noise is a prime candidate for fully utilizing ASDF as the required data volumes are among the biggest in our science. ASDF enables the storage of arbitrarily long waveform traces in a single file with fine grained access. One example is storing a station's data for several years in one file and only accessing a portion of the data whenever it is needed.

Many more use cases of the ASDF format can be envisioned, and we hope different subgroups within the seismological community will adapt it for their own purposes. Aside from seismological waveforms, ASDF's ability to save auxiliary data, including full provenance, enables it to store a lot of different pieces of data.

Examples include storing time-dependent power spectral densities and combining them into probabilistic power spectral densities on the fly (e.g. McNamara & Buland 2004) or building a database of historical earthquake data. Even non-seismological data, such as GPS time-series and magnetotelluric data, are not out of the question and would benefit from the provenance description and the advanced processing tools developed around ASDF. Some of these examples are already being attempted, and we intend to maintain a collection of use cases on our website.

6 CONCLUSIONS AND FUTURE DIRECTIONS

ASDF has been developed with the broader seismological community in mind, and our hope is that scientists within this community will continually test, offer feedback, and improve the format and its associated tools. Through such a communal effort, we will gracefully meet future data challenges and empower ourselves to make new scientific discoveries.

All components of the format, including its definition, implementation and other tools, are freely available under open source licenses and hosted on GitHub. A central entry point is the http://seismic-data.org website. We welcome any outside comments, criticisms and success stories, and we are committed to maintaining the documentation and implementations for the foreseeable future.

ACKNOWLEDGEMENTS

This research was partially supported by the EU-FP7 VERCE project (number 283543) and US NSF grant 1112906. We are grateful for the QUEST Initial Training Network (Marie Curie Actions, http://www.quest-itn.org) and the Computational Infrastructure for Geodynamics (CIG, https://geodynamics.org/) organization for holding a joint workshop that sparked the creation of the ASDF format.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

The authors also recognize support from the NSERC G8 Research Councils Initiative on Multilateral Research Funding and the Discovery Grant No. 487237.

Additionally, we thank Chad Trabant and Tim Ahern from the Incorporated Research Institutions for Seismology (IRIS) as well as Emiliano Russo, Peter Danecek and Rodolfo Puglia for fruitful discussions and useful tips. We also thank editor Andrea Morelli and two anonymous reviewers for their thoughtful comments which helped improve the manuscript. Finally, we gratefully acknowledge conversations with HDF5 Director of Earth Science Ted Habermann and help from Mohamad Chaarawi via the HDF5 User's Forum.

REFERENCES

Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y. & Wassermann, J., 2010. ObsPy: a Python toolbox for seismology, *Seismol. Res. Lett.,* **81**(3), 530–533.

Bird, I. *et al.*, 2014. Update of the Computing Models of the WLCG and the LHC Experiments, Tech. Rep. CERN-LHCC-2014-014 / LCG-TDR-002, CERN, Geneva.

Bormann, P., 2012. *New Manual of Seismological Observatory Practice (NMSOP-2), IASPEI*, GFZ German Research Centre for Geosciences, http://nmsop.gfz-potsdam.de/.

Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E. & Yergeau, F., 2008. 'Extensible Markup Language (XML) 1.0 (5th edn)'. Available at: https://www.w3.org/TR/2008/REC-xml-20081126/, last accessed 7 October 2015.

Fichtner, A., Bunge, H.-P. & Igel, H., 2006. The adjoint method in seismology. I. Theory, *Phys. Earth planet. Inter.,* **157**(1–2), 86–104.

Havskov, J., 2010. *Routine Data Processing in Earthquake Seismology with Sample Data, Exercises and Software,* Springer.

Helffrich, G., Wookey, J. & Bastow, I., 2013. *The Seismic Analysis Code: A Primer and User's Guide,* 1st edn, Cambridge Univ. Press.

Incorporated Research Institutions for Seismology (IRIS), 2012. 'SEED Reference Manual - Standard for the Exchange of Earthquake Data'. https://www.fdsn.org/seed_manual/SEEDManual_V2.4.pdf.

IRIS/PASSCAL Data Group, 2012. *Introduction to Active Source Data Archiving Utilizing PH5 as the Archive Format,* IRIS/PASSCAL Instrument Center, Version: 2012336.

Komatitsch, D. & Tromp, J., 2002a. Spectral-element simulations of global seismic wave propagation—I. Validation, *Geophys. J. Int.,* **149,** 390–412.

Komatitsch, D. & Tromp, J., 2002b. Spectral-element simulations of global seismic wave propagation—II. Three-dimensional models, oceans, rotation and self-gravitation, *Geophys. J. Int.,* **150,** 308–318.

Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C. & Wassermann, J., 2015. ObsPy: a bridge for seismology into the scientific Python ecosystem, *Comput. Sci. Discovery,* **8**(1), 14 003–14 020.

Liu, Q. *et al.*, 2014. Hello ADIOS: the challenges and lessons of developing leadership class I/O frameworks, *Concurrency Comput., Pract. Exp.,* **26**(7), 1453–1473.

McNamara, D.E. & Buland, R.P., 2004. Ambient noise levels in the continental United States, *Bull. seism. Soc. Am.,* **94**(4), 1517–1527.

Megies, T., Beyreuther, M., Barsch, R., Krischer, L. & Wassermann, J., 2011. ObsPy—what can it do for data centers and observatories?, *Ann. Geophys.,* **54**(1), 47–58.

MPI Forum, 2009. 'Message Passing Interface (MPI) Forum Home Page'. Available at: http://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf.

Rew, R. & Davis, G., 1990. NetCDF: an interface for scientific data access, *IEEE Comput. Graph. Appl.,* **10**(4), 76–82.

Schorlemmer, D., Wyss, A., Maraini, S., Wiemer, S. & Baer, M., 2004. 'Orfeus Newsletter 6(2): QuakeML - An XML schema for seismology'. Available at: http://www.orfeus-eu.org/organization/Organization/Newsletter/vol6no2/quakeml.shtml, last accessed 7 October 2015.

Schorlemmer, D., Euchner, F., Kästli, P., Saul, J. & Group, Q.W., 2011. QuakeML: status of the XML-based seismological data exchange format, *Ann. Geophys.,* **54**(1), 59–65.

SEG Technical Standards Committee, 2002. *SEG Y rev 1 Data Exchange Format,* Society of Exploration Geophysicists.

Stephens, Z.D. *et al.*, 2015. Big data: Astronomical or genomical?, *PLoS Biol.,* **13**(7), e1002195, doi:10.1371/journal.pbio.1002195.

Tape, C., Liu, Q., Maggi, A. & Tromp, J., 2010. Seismic tomography of the southern California crust based on spectral-element and adjoint methods, *Geophys. J. Int.,* **180**(1), 433–462.

The HDF Group, 1997–2015. 'Hierarchical Data Format, version 5'. Available at: https://www.hdfgroup.org/HDF5/, last accessed 7 October 2015.

Tromp, J., Tape, C. & Liu, Q., 2005. Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels, *Geophys. J. Int.,* **160**(1), 195–216.