ROSES: Remote Online Sessions for Emerging Seismologists

Fransiska K. Dannemann Dugick*[1], Suzan van der Lee[2], Germán A. Prieto[3], Sydney N. Dybing[4], Liam Toney[5], and Hank M. Cole[6]

Abstract

In response to a pandemic causing the cancellation of numerous professional development programs for emerging seismologists, we successfully planned, promoted, and executed an 11 week online school for advanced graduate students worldwide during the summer of 2020. Remote Online Sessions for Emerging Seismologists included 11 distinct lessons focused on different topics in seismology. We highlight the course content, structure, technical requirements, and participation statistics. We additionally provide a series of "lessons learned" for those in the community wishing to establish similar programs.

Cite this article as Dannemann Dugick, F. K., S. van der Lee, G. A. Prieto, S. N. Dybing, L. Toney, and H. M. Cole (2021). ROSES: Remote Online Sessions for Emerging Seismologists, *Seismol. Res. Lett.* **XX**, 1–11, doi: 10.1785/0220200421.

Introduction

In the year 2020, a pandemic (Andersen *et al.*, 2020; Lecocq *et al.*, 2020) caused by a contagious and deadly virus (COVID-19) upended professional development summer plans for graduate students worldwide. As an organization with global participation, the American Geophysical Union (AGU) responded by affirming its commitment to innovation and engagement (Bell *et al.*, 2020; McEntee, 2020; Oakes, 2020; Staff, 2020). Under the leadership of Section President Anne Sheehan, the executive committee of AGU's Seismology Section met online in May 2020 and decided to build and host an online seismology summer school during the summer of 2020. The goal was to offer a school for an international target audience of advanced graduate students in seismology. The effort was spearheaded by a junior and senior member of the AGU Seismology executive committee and was promptly supported by two other members, who also taught in the summer school. Instructors were invited, and all but two virtually immediately volunteered to teach one of the units. Building on widespread anecdotal and documented appreciation of the annual summer short course for graduate students "USArray Data Processing for the Next Generation of Seismologists" (EarthScope USArray Data Processing Short Course, 2009–2020), we designed, organized, and delivered Remote Online Sessions for Emerging Seismologists (ROSES) during the summer of 2020.

Inspired by National Science Foundation (NSF) programs that encourage interdisciplinary research training and harnessing the data revolution, such as the NSF Research Traineeship, ROSES offered lectures and experiential learning on topics at the intersection of seismology and data science. Moreover, ROSES offered students who were isolated in their homes because of the pandemic an innovative opportunity to network with peers from 28 different countries and 21 states within the United States (Fig. 1). ROSES used teleworking technologies such as Zoom and Slack to offer 11 topically distinct though loosely threaded learning units on a weekly basis, each with a different instructor active in seismological research. These instructors spanned career stages from graduate students to full professors. The three graduate students enthusiastically provided critical additional support as teaching assistants (TAs) to students and instructors and throughout the course.

Here, we document the ROSES units and course repository and provide a set of lessons learned for future online graduate-level summer schools. The first part of this article provides an overview of the 11 week course content. Next, we outline the lecture and laboratory components of the course followed by the technical requirements of the course. We detail the archived course repository for future use within the seismological community. Finally, we briefly describe participation statistics and student demographics. Each section contains a series of lessons learned developed by the volunteer instructor and coordination team.

1. Geophysical Detection Programs, Sandia National Laboratories, Albuquerque, New Mexico, U.S.A., https://orcid.org/0000-0001-8328-4835 (FKDD); 2. Department of Earth and Planetary Sciences, Northwestern University, Evanston, Illinois, U.S.A., https://orcid.org/0000-0003-1884-1185 (SvdL); 3. Departamento de Geociencias, Universidad Nacional de Columbia, Bogota, Colombia, https://orcid.org/0000-0001-8538-7379 (GAP); 4. Department of Earth Sciences, University of Oregon, Eugene, Oregon, U.S.A., https://orcid.org/0000-0002-9274-6568 (SND); 5. Alaska Volcano Observatory and Wilson Alaska Technical Center, Geophysical Institute, University of Alaska Fairbanks, Fairbanks, Alaska, U.S.A., https://orcid.org/0000-0003-0167-9433 (LT); 6. Geosciences Department, Warner College of Natural Resources, Colorado State University, Fort Collins, Colorado, U.S.A., https://orcid.org/0000-0003-1684-9116 (HMC)

*Corresponding author: fkdanne@sandia.gov

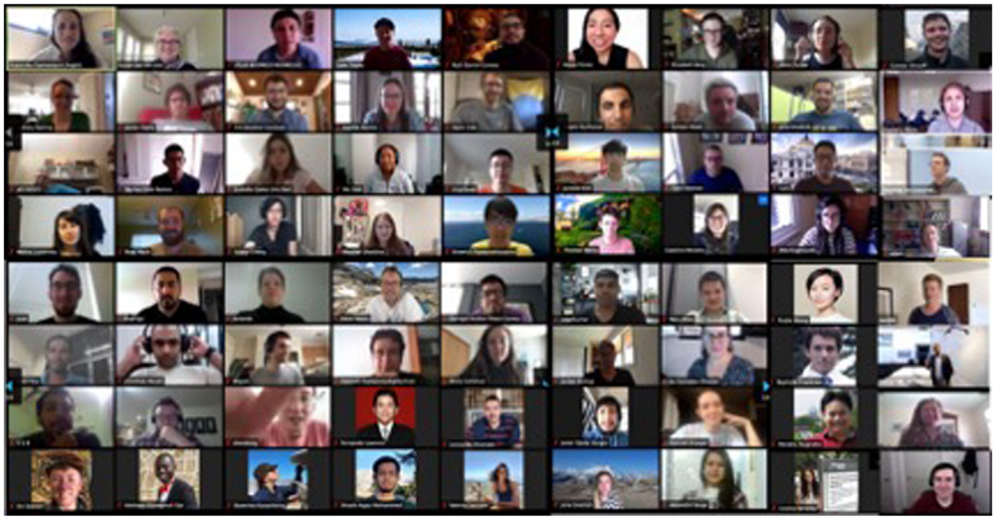


Figure 1. A snapshot of part of the Remote Online Sessions for Emerging Seismologists (ROSES) 2020 cohort. The color version of this figure is available only in the electronic edition.

Course Content

In unit 1, taught by Sydney Dybing (University of Oregon), ROSES students learned to use a Python package called *ObsPy* (Krischer *et al.*, 2015). Unit 1 covered reading seismological waveform data from a file, writing data to a file, downloading data from an online data center directly into the student's Python workspace, an approximate and easy way to remove an instrument response, examples of signal processing such as filtering, and plotting waveform data. As a toolbox for seismological data analysis, *ObsPy* provides parsers for common file formats in seismology, in-code access to data at data centers offering webservices, and seismology-specific waveform-processing methods and functions. An advantage to multigenerational research teams is that these processing routines perform similar to those in the Seismic Analysis Code (SAC), which was the primary toolbox used for seismological data analysis in the late twentieth century. SAC was initially written in Fortran (Tull, 1986), then translated to C (Goldstein *et al.*, 2003), and was recently updated to include in-code access to data at data centers offering webservices (Savage and Snoke, 2020).

In unit 2, taught by Emily Wolin (U.S. Geological Survey), students learned about the reality of seismometer installations, the importance of accurate metadata in the analysis of seismic data, and the underlying theory and pitfalls of accurately removing instrument responses. Unit 2 covered different types of seismometers, digitizers, signal processing routines, and channel naming conventions; different types of noise sources and instrumentation limitations; visualization of instrument response functions and the process of removing them; metadata methods and file formats; the role and effects of prefilters and of replacing a zero instrument response amplitude with a minimal one ("water level"); understanding power spectral density (PSD), and the importance of removing the signal's mean and tapering its beginning and end. Seismology graduate students with an interest in data science may not also be actively involved in acquiring seismic data in the field or operating a seismic network. Despite this, ROSES students will likely revisit and benefit from this material in post-PhD professions that require them to conduct seismic experiments, manage seismic data, interpret noise or low-amplitude signals, or teach instrumentation-based seismology. A research application that quantifies effects of station installation on the PSD can be found in Wolin *et al.* (2015).

In unit 3, taught by Germán Prieto (Universidad Nacional, Colombia), students learned about stationary time series, Fourier analysis, and multitapers. Unit 3 covered statistics of ordered and stationary time series, the theory and application of digital Fourier transforms and PSDs, multiple tapers that reduce spectral leakage, cross correlation, deconvolution, spectral ratios, and an application to earthquake source spectra. As an overview of what could be a semester-long course, this unit provided ROSES students with an information-dense, theory-backed review of what they likely learned as part of their university education but might not have had imminent applications at that time. This changes in and after graduate school, where most seismologists analyze time series (seismograms) on a near-daily basis. Unit 3 connected to material learned in unit 2; however, it would be equally feasible to offer unit 2 after unit 3. A research application of multiple tapers can be found in Prieto *et al.* (2009).

In unit 4, taught by Elizabeth Berg (University of Utah), students learned about waveform cross correlation and associated normalization and equalization in the context of an application of ambient noise tomography (ANT). The unit used knowledge from the preceding units on *ObsPy*, cross correlation, filtering, and calculating PSDs. Unit 4 covered noise cross correlation, different types of temporal normalization and of frequency equalization, deconvolution interferometry and associated "water levels," and an application to waveforms recorded by the 2014 Sweetwater Array in Texas (Woodward, 2014). Several interactive quality inspections are built in at regular intervals in the data-processing workflow. The unit used *cartopy* (for easy, interactive mapping) and spatiotemporal visualization during the workflow and associated code development. The Python code for this unit was translated and modified from the original Fortran package for ANT by Shapiro *et al.* (2005). A research application of ANT is described by Berg *et al.* (2020).

In unit 5, taught by Stephen Arrowsmith (Southern Methodist University), students learned about array seismology through the use of spatiotemporal correlations between seismograms recorded at an array of seismic stations. Unit 5 covered the estimation of slowness vectors (azimuth and ray parameters), beamforming and frequency–wavenumber analysis, and detection methods including the ratio of short-term and long-term averages, backprojection, local similarity methods, network similarity, and an application to infrasound data. The material covered in unit 5 is applied in both fundamental seismology research as well as in international global monitoring collaborations. Examples of research applications in both areas are found in Gibbons and Ringdal (2006) and Arrowsmith *et al.* (2010).

In unit 6, taught by Tolulope Olugboji (University of Rochester), students learned about the fundamental principles of seismic-wave propagation and polarization analysis. Students applied these principles to estimate slowness vectors from single-station seismograms. Unit 6 covered body-wave and surface-wave propagation and polarization using material from the preceding units and applied the theory to determine the orientation of an ocean-bottom seismometer, optimally rotate components in a receiver function analysis, and analyze a split *SKS* wave. The unique polarization of seismic waves provides the opportunity to apply specific analyses that have no equivalent in other disciplines and can provide us with critical information about the interior structure of the Earth. A research application is described by Olugboji *et al.* (2013). Polarization analyses are also used in planetary seismology, such as for locating the epicenters of marsquakes using a single seismic station installed by the InSight mission (Lognonné *et al.*, 2020).

In unit 7, taught by Zachary Ross (California Institute of Technology), students learned about concepts of machine learning (ML) and associated nomenclature. Unit 7 provided an overview of the field of ML and covered features and labels, supervised and unsupervised learning, classification and regression, linear separability and classifiers, different types of optimizers, training and validation, and neural networks and deep learning. Unit 7 used an open-source Python package, *PyTorch*, to linearly separate *P* and *S* waveforms from each other and from background noise using their spectral amplitudes. Like in many science and engineering disciplines, ML is an increasingly popular area of active research in seismology. As for most preceding units, the material touched on in this unit can easily be spread over a semester-long course. In waveform seismology, ML has been successfully applied to produce new results, mostly for signal-detection purposes, including for automated detection of signals from local earthquakes (Ross *et al.*, 2018).

In unit 8, taught by Liam Toney (University of Alaska Fairbanks), we switched from a focus on waveform analyses to a focus on spatial analyses and data visualization. Students learned the basics of *PyGMT*, a recently developed Python interface to the popular Generic Mapping Tools (GMT; Wessel *et al.*, 2019). Unit 8 covered PyGMT installation; basic usage such as specification of map regions and projections, plotting symbols with legends, and displaying gridded data; and how to use GMT modules for which a *PyGMT* wrapper has not yet been released. Seismologists pioneered the open and global sharing of data many decades ago (Smith, 1987) and now embrace Python for similar reasons: it is open source, global, and collaborative. Many seismologists are familiar with command-line-based GMT and have eagerly awaited a Python interface. A major advantage of *PyGMT* (and modern GMT) is the ability to easily retrieve and plot gridded data, such as digital elevation models, from GMT's servers. Although this unit was dedicated to *PyGMT*, other units such as unit 4 used *cartopy* for mapping. The use of multiple packages for ROSES was intentional and highlighted the options available to students for mapping and producing figures using Python.

In unit 9, taught by Steve Myers (Lawrence Livermore National Laboratory), students learned about conditional probabilities, Bayesian statistics, and Markov chain Monte Carlo inversions in a spatial context, applied to the example of inferring the hypocenter of an earthquake and its multidimensional uncertainty. Unit 9 used *Bayesloc*, a package developed by Myers *et al.* (2007) and written in C, for relocating aftershock hypocenters relative to the hypocenter of the mainshock. Bayesian inversion is used in both earthquake and structural seismology, in which its attraction lies in its ability to handle nonlinear inverse problems and to produce uncertainty estimates. These uncertainty estimates are useful for data-rich, overdetermined problems such as inferring earthquake epicenters. A research application is described by Myers *et al.* (2015).

In unit 10, taught by Tony Lowry (Utah State University), students learned about kriging, an application of optimal interpolation in a spatial domain, and more about Bayesian inference. Unit 10 used *PyGMT* knowledge from unit 8 and covered kriging nomenclature, variograms and semivariograms, an application to inferring crustal structure of the United States from USArray waveforms and gravity, and an example of how optimal interpolation can be used to estimate realistic prior probability density functions. Although optimal interpolation is one of the oldest mapping and gridding methods in the Earth sciences, it is experiencing a resurgence of interest in data science and ML curricula, albeit under the name Gaussian processes. The Bayesian nature of the method lends itself well to applications that simultaneously interpret different types of data, including waveform data such as receiver functions. A research application is described by Ma and Lowry (2017).

In unit 11, taught by Suzan van der Lee (Northwestern University), students learned the basics of using inverse methods for mapping irregularly spaced data points onto regularly spaced grid points that support a continuous spatial function. Unit 11 covered a nonoptimal interpolation of the kind that is used in kriging, used *PyGMT* for mapping, introduced basis functions and linear inversion theory, and covered an application to mapping crustal thickness across the United States. This unit shows that the same linear inverse methods used in seismic tomography can also be used for mapping data from one set of points to another or to points, weights, or coefficients that support or describe a continuous function. A research application is available in Chang *et al.* (2010).

Lessons learned

1. Exit survey results indicate that students appreciated the topics and content of ROSES, which combined both new-to-them and familiar topics within seismology.
2. The units proceeded in a logical order, and instructors connected their units to previous units.
3. A logical follow up to unit 11 would be a unit 12 on the concepts of seismic tomography.
4. The computational environment was simple and similar for most units; however, technical problems arose for some units that required additional Python packages or non-Python software. This point is addressed in the following, in which we discuss computational requirements.

Course Format

Each weekly unit was delivered synchronously (live to a student audience) and was divided into three parts. The first section (about 30–60 min) followed a traditional lecture format, in which the instructor used slides to present the introduction of the topic prepared for that week. The slides generally included an introduction to the applications of the topic in seismology, a background of the theory, and Python-based practical application. For some topics, such as the units on *ObsPy* and *PyGMT*, lectures were focused on practical applications of the material and were presented interactively in Python through the use of Jupyter Notebooks. In other cases, lectures were delivered in a more traditional format with a quick overview of the background theory (e.g., Fourier or inverse theory) that would normally be covered over the course of a term in graduate-level study.

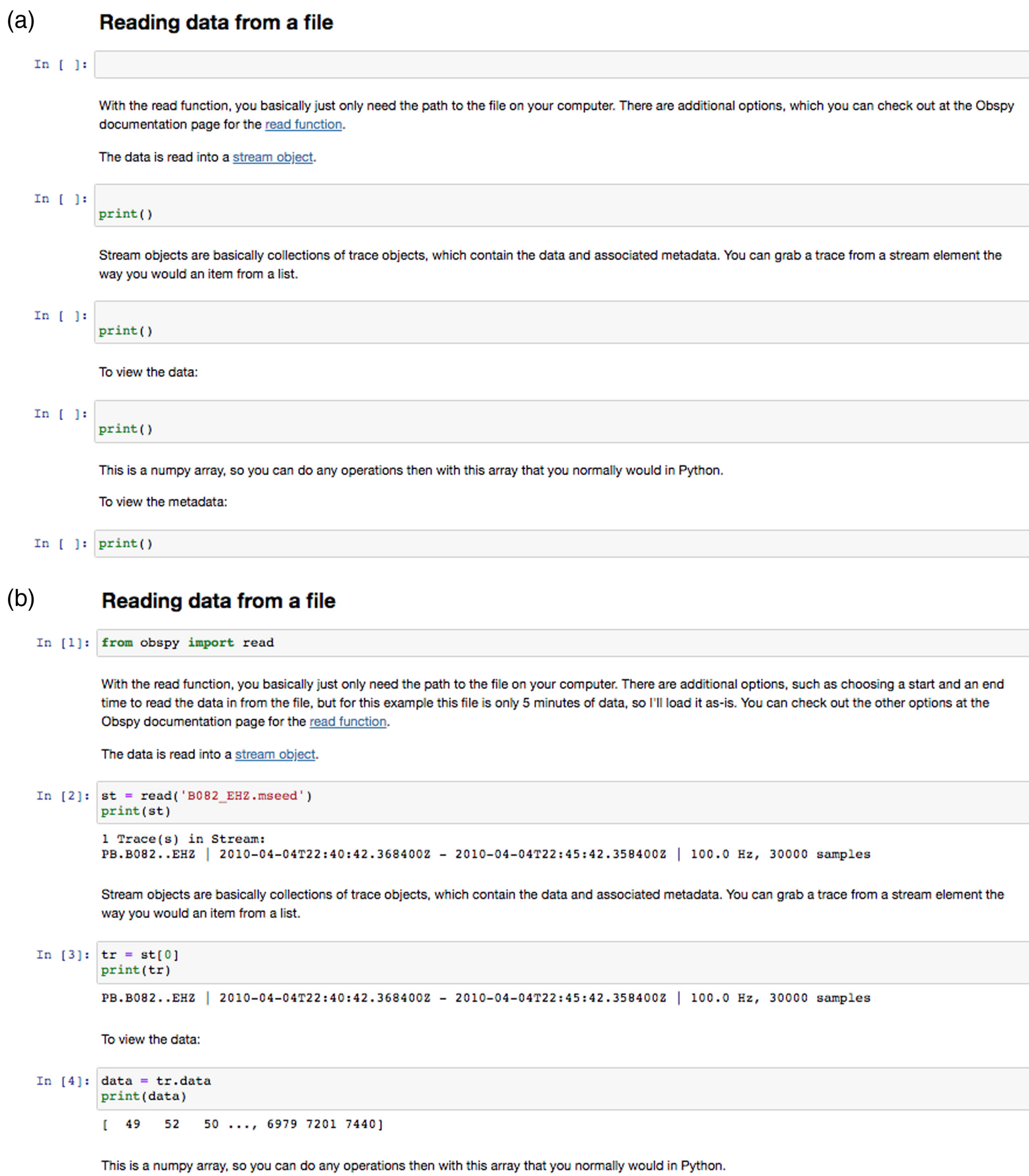
The second part of the unit (30–60 min) used Jupyter Notebooks and Zoom screen-sharing features to deliver a hands-on computing laboratory section. For all units aside from unit 9, labs consisted of Python-based examples related to the lecture topic implemented within a Jupyter Notebook. The labs were prepared in such a way to allow students to run the programs while following along with the instructor. See Figure 2 for an example of a lab notebook. To achieve this, instructors prepared notebooks in which students could follow descriptions of the code and were able to run them on their own computer. In some cases, students were also required to add their own code to the notebooks to obtain a result they could interpret. This allowed the class to change one or more parameters within the notebook to interact with the code and thus better understand how the programs worked. Students were also exposed to some of the limitations, problems, and warnings they should think about when developing their own computer programs.

In the third and final part of the unit (Do-It-Yourself [DIY] lab, 30+ min), students were divided into groups of four to five using the Zoom breakout room feature. Students were provided with a Jupyter Notebook for this section. Instead of coding with the instructor, students were expected to complete a number of tasks beyond what was explained in the laboratory part in collaboration with their breakout room peers. The advantage of such an approach was that students within each room could share their screens, show their codes or approach, and fix bugs together. In addition, the instructor and TAs could visit the breakout rooms to help students if they had difficulty making progress. At the conclusion of the DIY Lab session, we offered an optional do-at-home challenge exercise as a way for students to expand upon concepts addressed in the lecture and laboratory sections. These challenges encouraged student collaboration, application of techniques to their own research projects, and continued instructor–student interactions beyond the live lecture and laboratory session.

Slack communication

All instructors and students joined a ROSES Slack workspace that facilitated group and private communication between participants. Slack is a popular messaging platform available on all operating systems that is designed for organizations to host their own topic-oriented chat rooms called channels. Slack has many social media–like features (e.g., profile pictures, statuses, hashtags, and emojis) that are intended to provide a more casual user experience. The workspace was primarily divided into channels for each unit of the course, but there was also a very active channel dedicated to technical troubleshooting. Throughout the course, the Slack workspace retained more than 120 active users, and >4500 messages were sent. It proved to be an effective platform for facilitating the synchronous and asynchronous components of each unit. On the days of lectures and labs, it was a convenient place to post links and files needed for the course. Throughout the week following each lecture, it was active with people asking follow-up questions and posting their results.

Importantly, Slack features worked in this case as a substitute for the face-to-face interactions that would have occurred if this workshop were hosted in person. We began the course by asking students to post an introduction to themselves and their research in the "Introductions" channel. This activity was appreciated by participants and assisted in facilitating



(a)

Reading data from a file

In [ ]:

With the read function, you basically just only need the path to the file on your computer. There are additional options, which you can check out at the Obspy documentation page for the read function.

The data is read into a stream object.

```
In [ ]: print()
```

Stream objects are basically collections of trace objects, which contain the data and associated metadata. You can grab a trace from a stream element the way you would an item from a list.

```
In [ ]: print()
```

To view the data:

```
In [ ]: print()
```

This is a numpy array, so you can do any operations then with this array that you normally would in Python.

To view the metadata:

```
In [ ]: print()
```

(b)

Reading data from a file

```
In [1]: from obspy import read
```

With the read function, you basically just only need the path to the file on your computer. There are additional options, such as choosing a start and an end time to read the data in from the file, but for this example this file is only 5 minutes of data, so I'll load it as-is. You can check out the other options at the Obspy documentation page for the read function.

The data is read into a stream object.

```
In [2]: st = read('B082_EHZ.mseed')
        print(st)

1 Trace(s) in Stream:
PB.B082..EHZ | 2010-04-04T22:40:42.368400Z - 2010-04-04T22:45:42.358400Z | 100.0 Hz, 30000 samples
```

Stream objects are basically collections of trace objects, which contain the data and associated metadata. You can grab a trace from a stream element the way you would an item from a list.

```
In [3]: tr = st[0]
        print(tr)

PB.B082..EHZ | 2010-04-04T22:40:42.368400Z - 2010-04-04T22:45:42.358400Z | 100.0 Hz, 30000 samples
```

To view the data:

```
In [4]: data = tr.data
        print(data)

[  49   52   50 ..., 6979 7201 7440]
```

This is a numpy array, so you can do any operations then with this array that you normally would in Python.

community building throughout the course. An example of these interactions is shown in Figure 3. Students commonly posted encouraging messages when responding to questions, and a portion of the participants were active in the off-topic social channels.

Figure 2. Example Jupyter Notebook from unit 1: ObsPy. (a) The screenshot detailing the blank notebook provided to students and (b) the screenshot demonstrating an example student input. The color version of this figure is available only in the electronic edition.

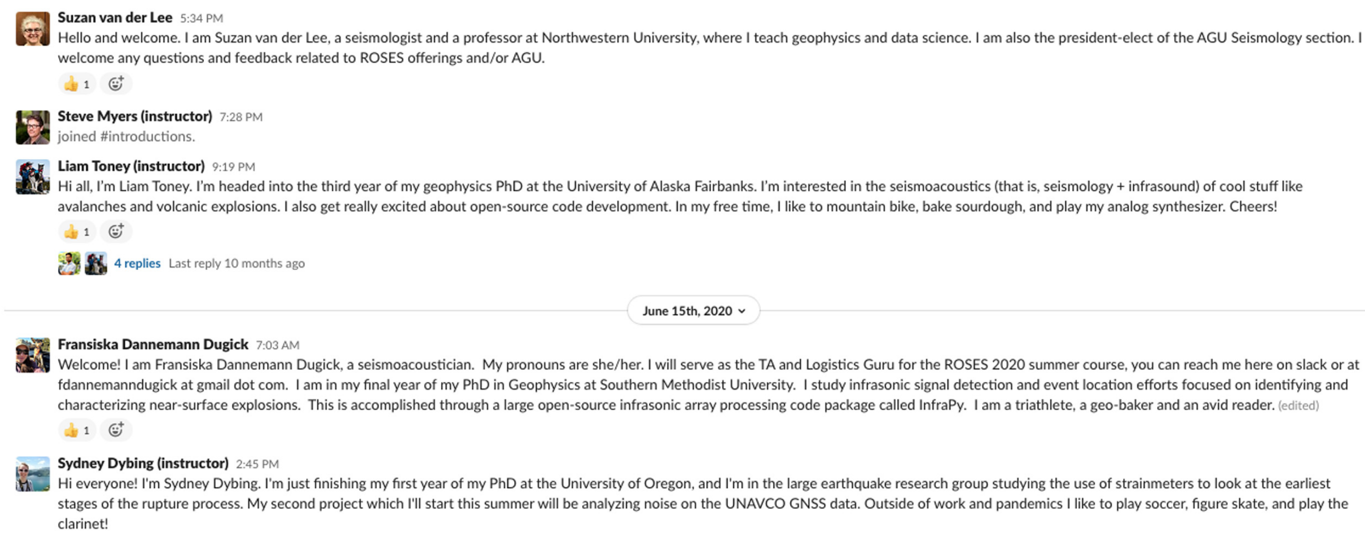


Figure 3. A screenshot from the Slack introductions channel. The color version of this figure is available only in the electronic edition.

Lessons learned

1. Exit survey results indicate that some students felt lectures and labs ran long. This is likely due to a combination of factors, including a lack of information regarding lecture duration in the course advertisement, communication between ROSES organizers and instructors that allowed flexibility regarding lecture timing, and a high volume of detailed student questions within lectures. In addition, this feedback may also be a reflection of an overall virtual summer in which students and instructors alike felt overwhelmed by Zoom fatigue.
2. Instructors were given some flexibility in preparing their lecture and laboratory materials. An even more common format would provide continuity throughout the summer.
3. Breakout room sessions were not well attended. Exit survey results indicated this was due to a combination of timing (labs running long) and students feeling uncomfortable working with strangers. This could be improved by intelligent grouping strategies, additional introductory activities, having TAs facilitate the breakout sessions, or pre-assigning groups to increase student accountability.
4. The use of Slack for communication saved instructors and students from an unmanageable number of emails and also facilitated meaningful networking in a way that would otherwise be difficult in this remote format.

Technical Requirements

To make the cohort size more manageable and enable instructors to prepare material of adequate complexity, we required that students who applied to be a part of ROSES have some existing foundational and technological knowledge. To establish a knowledge baseline for the cohort, we asked that students were preferably in their second, third, or fourth year of graduate school, that they could install a conda Python environment in a Unix-based environment, and that they could add additional packages to this conda environment as necessary as the course progressed.

We established several computing requirements that were relayed to students in advance of their application for ROSES. The instructor team had elected to exclusively use Python in the course because it is open source, free, and collaborative. Because of this and as a result of the experience of the ROSES instructors, we specified that students were required to have access to a Unix-based machine and had installed the conda package manager (via Anaconda or Miniconda) so they could use the environment that the instructor team prepared for the course.

None of the ROSES instructors or organizers used Windows machines, so we emphasized that if anyone chose to participate in the course with a Windows computer and was not emulating a Unix environment on it, we would not be able to provide technical support. Even so, several students did choose to participate in ROSES with Windows computers. The Slack workspace that we had created proved very useful in this regard because students with Windows machines were able to collaborate and solve some of the issues they ran into on the technical channel.

Lessons learned

1. The lack of support provided for Windows users during ROSES may have resulted in the unintentional exclusion of many students from South America, Africa, and Asia. To make the course more inclusive and attract a greater diversity of international students in the future, we would like

to involve instructors and TAs familiar with using Python and conda environments on Windows machines so that these students can participate and receive technical support.
2. Based on exit survey results, we have determined that it would be ideal to loosen the "year in graduate school" prerequisite so that first-year graduate students, as well as postdoctoral scholars, are able to participate provided they have the necessary foundational knowledge.
3. In the exit survey, although many students said the prerequisites were appropriate for the course content, many also said that these prerequisites were unattainable (i.e., students did not have the Python experience required but recognized that Python computing is increasingly common in seismology and the geosciences in general and desired to gain experience with the programs used within laboratory sections). Advertising and registering earlier in the year such that students are able to self-teach ahead of future ROSES iterations may help to mitigate this issue. We include a reference to the official Python tutorial within the GitHub repository as a resource for future students.
4. In the original application, we requested that students be prepared to read or view assigned materials before each lecture. However, because of timing constraints, instructors were not always able to release these materials far ahead of time, and it was often actually requested by ROSES students that this be done to help them prepare for lectures. This is an improvement that we would like to implement for future summer schools. In addition, it would be ideal if instructors provided laboratory materials to students further ahead of the lecture time so that the students could also test them and any problems could be solved over Slack before the live session.
5. During some of the live lecture and lab sections, students had platform-specific problems with the conda environments or the provided Python codes. The Slack workspace and Zoom chat were again both very useful in identifying and working through these problems, but in an ideal situation these types of bugs would be caught before students had access to and were attempting to run the notebooks and codes the instructors provided. Whether this is achievable is unclear, considering there were also issues with operating system versions that may not have been caught even if instructors were able to test all materials on a diversity of operating systems beforehand. In addition, we recognize that dealing with these issues provided students with experience with common debugging tactics that they would at some point likely encounter outside of ROSES, which could be seen as an unintentional benefit of this process.
6. The conda environment provided by the instructor team should, if possible, be established and consistent from the beginning of the summer school. We did build and provide a YAML file with all the dependencies the instructors requested before the first lecture session, but some additional dependencies needed to be added later as instructors continued to prepare their materials. This created confusion for ROSES students and often caused difficulties with installing and running code. As some students specified in the exit survey, this caused them to struggle to follow certain lectures because they fell behind attempting to solve technical issues. This environment will also need to be tested on all operating systems before distribution to students. Some instructors received help from ROSES organizers to port their material to the common computational environment, which further underscores that more professional computational and technical support for more units would have benefitted ROSES students significantly.

Participation Statistics and Student Demographics

Given the short planning timeline associated with ROSES—the course was conceptualized in mid-May and started mid-June—the course registration process necessitated a rapid approach. We drafted and released a course advertisement and registration link two weeks before the planned start date of 11 June 2020. The course was advertised across the AGU Seismology Section Twitter and Facebook page, a variety of university mailing lists and email announcements, and both the AGU Seismology Section mailing list and the AGU Seismology Section Student and Early Career mailing list. Registration was run through a Google Form in which participants self-selected both their availability and compatibility with the prerequisites mentioned. We received 534 registrations within this short two-week window. Figure 4 details the breakdown of how each registrant found out about the course; we were surprised to see that posting on our social media sites brought in so many potential students (26% of applicants).

Of the 534 registrations, we ultimately accepted 181 students into the ROSES 2020 cohort. This limitation on course participants was driven by a combination of our Zoom video capacity limit of 300, adherence to course prerequisites, and a desire to create a global seismology student cohort that would persist after completion of the course. We accepted students who indicated that they were (1) available to attend all lectures "live," (2) within the target population of advanced PhD students, and (3) had the technical requirements outlined previously. The remaining noncohort applicants were invited to view lecture recordings on a delay, as discussed in the following.

Geographic distribution of participating students was determined by a combination of self-identified location within the introductions channel and an exit survey and represents 129 of 181 students within the cohort. Students within the cohort self-identified as being from 28 countries and 21 states within the United States, as shown in Figure 5. The prevalence of U.S.-based students within the cohort (Fig. 5a) is likely due to a combination of the course being offered by U.S.-based

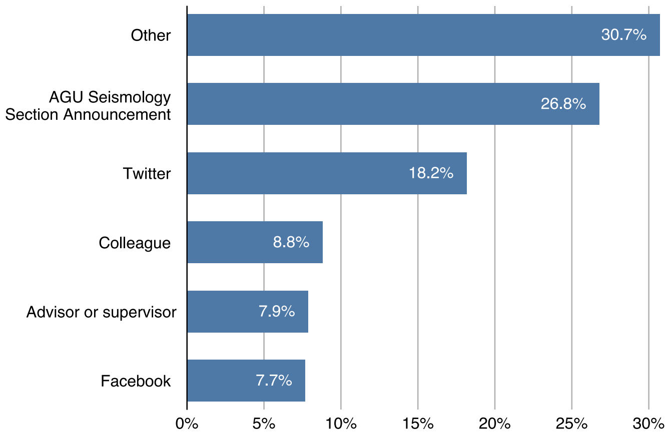


Figure 4. Breakdown of how the 534 students interested in ROSES found out about the course. AGU, American Geophysical Union. The color version of this figure is available only in the electronic edition.

organizers and lectures occurring at 12 p.m. Eastern Daylight Time (EDT).

Of the 181 students accepted into the cohort, 161 regularly participated in lectures and Slack conversations. We used Zoom participation records to document each unit's synchronous attendance as well as track overall course participation. We note that as the summer progressed, Zoom participation declined, corresponding to the beginning of the fall semester (Fig. 6a). We note a similar decline in the number of Slack members active within the community. As shown in Figure 6b, a number of students attended the organizational introductory lecture, in which prerequisites were explained, and then elected not to participate throughout the summer. Similarly, although we requested in registration that students commit to attending nearly all 11 units, only 16 registered students attended all units. Although we recognize that this was an online course and other commitments do come up, in the future, we would like to develop a registration process that ensures students who are accepted commit to full course participation. We offer a few suggestions within the Lessons Learned section to address this point.

Lessons learned

1. Advertising on social media helped reach seismologists who may not be AGU members and do not receive section newsletters or who are AGU members but do not thoroughly read these emails. Broader advertisement and engagement with additional professional societies may increase the geographic distribution of any future ROSES courses.
2. A large number of students registered based on a forwarded email from their advisors. In the future, targeting advisors directly may help in advertising the course to a broader audience. In addition, advisor sponsorship or coregistration may be a way to increase student attendance and accountability throughout the course duration.
3. We only accepted a portion of students who registered for ROSES, and some of these students did not fully participate in the course despite indicating in their registration that they could and would attend all if not most units. This requirement was based on a desire to create continuity within the course, as well as develop a global cohort of seismology students similar to the cohorts that arose from the in-person week-long USArray short courses. There needs to be a better way to hold registered students accountable so

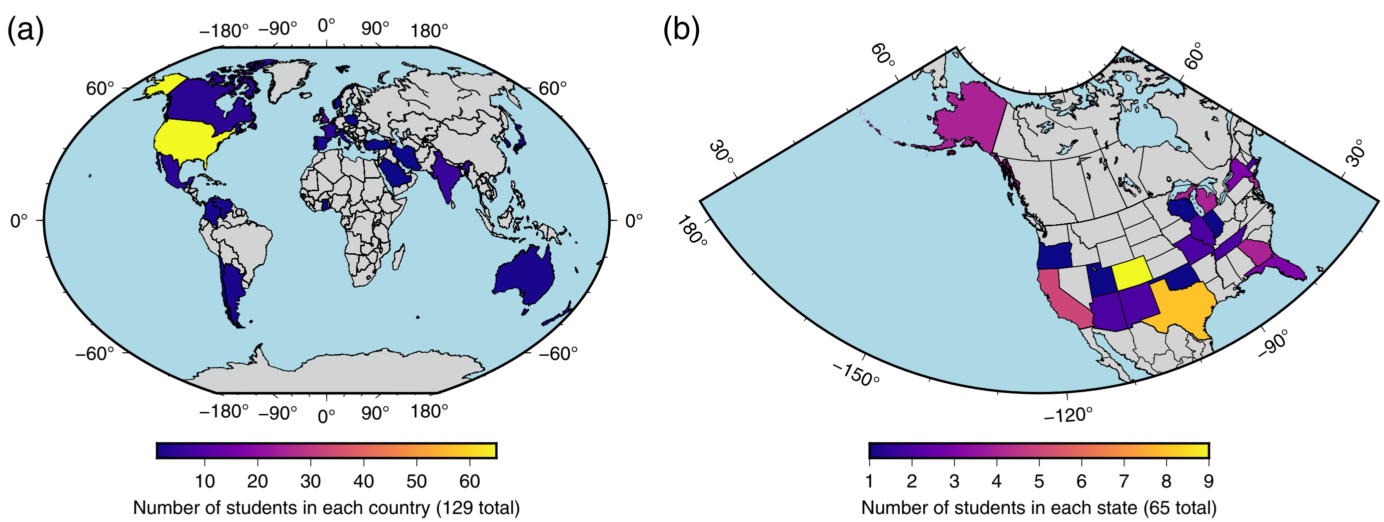


Figure 5. ROSES student locations, as self-identified from the introductions channel and the ROSES exit survey, for (a) the world and (b) the United States. (There were no attendees from Hawaii or any territories of the United States.) The color version of this figure is available only in the electronic edition.

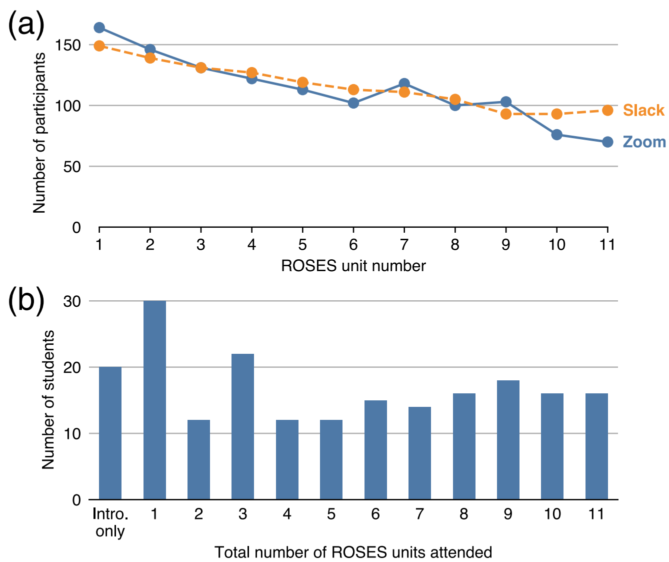


Figure 6. Synchronous ROSES attendance. (a) Number of participants, based on Slack activity and Zoom tallies, for each ROSES unit. (b) Attendance summary for ROSES. The color version of this figure is available only in the electronic edition.

that students who do not participate do not do this at the expense of qualified applicants who were not admitted because of lesser limitations in their availability.

4. An alternative approach to ensuring students' commitment would be to require some sort of investment from enrolled students in exchange for participation. This investment could be monetary, in kind, or educational (e.g., presenting and publishing or posting a final collaborative project).
5. We recognize that a number of factors may have limited global participation, including U.S.-centric live lecture times, inadvertent exclusion of Windows users, the short ramp-up time, and advertising time and advertising venues. We strive to find a more equitable solution for any future ROSES offerings.

Course Repository

Recorded lectures

Because of the overwhelming interest in the course, we elected to record each weekly lecture so that material would be available to a broader audience. Lectures were recorded using the Zoom cloud recording option, which provided the opportunity to produce automatic transcription for closed captioning. The technical content of the lectures was rarely translated effectively, and revising the transcription before posting the lecture was a significant undertaking. Each hour-long lecture required between 2 and 4 hr of transcription review and editing before posting. Recorded lectures are hosted on the Incorporated Research Institutions for Seismology (IRIS) YouTube channel and linked through their website. Posted lectures have been viewed hundreds to thousands of times per unit as of 18 February 2021.

GitHub

The ROSES GitHub repository is a publicly available source to access the code provided to students in lectures and labs. GitHub is an easy-to-use platform that allows anyone to explore the code in browser and download individual files or the entire repository as desired. The repository is organized by unit and includes Jupyter Notebook files and conda environment files if different from the ROSES default conda environment. Raw data used by the Jupyter Notebooks are included in the repository when files are small. Because of GitHub limitations, large files are hosted elsewhere, and instructions to access them are included in the documentation. The main GitHub repository webpage includes instructions for getting started with the code and setting up the default ROSES environment. For each unit, a readme is included that describes the data files, notes the use of special conda environments when appropriate, and provides troubleshooting tips for running the notebooks. The documentation is written such that anyone meeting the course prerequisites can access, run, and troubleshoot all code provided. In combination with the posted lecture videos, this should allow people to use materials to self-teach for any units of interest.

Conclusions

In response to the cancellation of seismology-focused graduate-level professional development activities during the summer of 2020, a team of AGU Seismology Executive Council members and volunteer instructors planned, promoted, and executed an 11 week online school for advanced graduate students within seismology. The course made use of teleworking technologies to offer live lecture and laboratory sessions on a variety of topics within the field. In addition, course materials were preserved and are hosted on the IRIS website (lectures) and GitHub (lab materials). Links are included within Data and Resources. At the conclusion of ROSES 2020, we conducted an exit survey to gauge student reception of the summer school. One hundred eight out of 181 registered participants submitted responses to the exit survey. Based on our experiences and results from our exit survey, we present the following conclusions:

1. There was enormous interest in ROSES (500+ applicants). We had planned for a small cohort of a few dozen. We compromised by accepting 100+ students, which, after some attrition, resulted in an average of 70 students per unit.
2. The willingness of invited instructors to teach a unit testified that we tapped into a positive and collaborative AGU Seismology culture.
3. The topics and content of the units were one of the most appreciated aspects of ROSES, and instructors enthusiastically built connections between their units and the preceding ones.
4. The other most appreciated aspect of ROSES was the emphasis on collaboration and networking within an

international cohort. This was facilitated through the Slack channel, which additionally provided an efficient avenue for communication through the course duration.

5. Because of rapid development and unusual circumstances, some aspects of ROSES could have been implemented better, and they have been detailed throughout this article in the lessons learned.
6. Posting recorded lectures and associated lab materials online reached a wider audience of over 8000 total views in just two months since conclusion of ROSES.
7. Potential future offerings of ROSES should have more planning time and with the help from the learned lessons documented in this article will sure that the course is better positioned to reach an even broader global seismology audience.

Data and Resources

Specific Python packages used in each unit of Remote Online Sessions for Emerging Seismologists (ROSES) are detailed in the unit's overview within the Course Content section. More information about *ObsPy* can be found at https://docs.obspy.org. cartopy can be accessed at https://scitools.org.uk/cartopy/docs/latest/. More information about *PyTorch* can be found at https://pytorch.org/docs/stable/index.html. More information about *PyGMT* can be found at https://www.pygmt.org. All data used within each unit's laboratory section came from published sources listed in the References section. Figure 5 was made using *PyGMT* (Wessel *et al.*, 2019). Recorded ROSES lectures are hosted on the Incorporated Research Institutions for Seismology (IRIS) website at https://www.iris.edu/hq/inclass/course/roses. All ROSES materials can be accessed via GitHub at https://github.com/fdannemanndugick/roses2020. All websites were last accessed in October 2020.

Declaration of Competing Interests

The authors acknowledge that there are no conflicts of interest recorded.

Acknowledgments

The authors thank the team of instructors who all volunteered their time: Sydney Dybing, Emily Wolin, Germán Prieto, Elizabeth Berg, Steven Arrowsmith, Tolulope Olugboji, Zachary Ross, Liam Toney, Steven Myers, Tony Lowry, and Suzan van der Lee. The authors thank Emily Wolin for early discussions about the concept of the summer school and for coining the Remote Online Sessions for Emerging Seismologists (ROSES) acronym. The American Geophysical Union (AGU) Seismology Section President Anne Sheehan called the meeting that led to the creation of ROSES. Danielle Sumy at IRIS aided greatly by hosting ROSES materials on the IRIS website and videos on YouTube. The ROSES logo was generously provided by Mi Vu, a graphic designed based in Windsor, Ontario, Canada. All students and instructors pictured in Figure 1 provided permission for their photos to be included in this publication. All instructors shown in Figure 3 provided permission for their Slack comments and likenesses to be included in this publication.

References

Andersen, K. G., A. Rambaut, W. I. Lipkin, E. C. Holmes, and R. F. Garry (2020). The proximal origin of SARS-CoV-2, *Nat. Med.* **26,** no. 4, 450–452, doi: 10.1038/s41591-020-0820-9.

Arrowsmith, S., J. B. Johnson, D. P. Drob, and M. A. H. Hedlin (2010). The seismoacoustic wavefield: A new paradigm in studying geophysical phenomena, *Rev. Geophys.* **48,** no. 4, doi: 10.1029/2010RG000335.

Bell, R., S. Lozier, and B. Williams (2020). AGU continues its work and commitment for a more diverse and inclusive geoscience community, *Eos*, available at https://fromtheprow.agu.org/agu-continues-its-work-and-commitment-for-a-more-diverse-and-inclusive-geoscience-community/ (last accessed October 2020).

Berg, E. M., F. Lin, A. Allam, V. Schulte-Pelkum, K. M. Ward, and W. Shen (2020). Shear velocity model of Alaska via joint inversion of Rayleigh wave ellipticity, phase velocities, and receiver functions across the Alaska Transportable Array, *J. Geophys. Res.* **125,** no. 2, doi: 10.1029/2019JB018582.

Chang, S. J., S. Van Der Lee, M. P. Flanagan, H. Bedle, F. Marone, E. M. Matzel, M. E. Pasyanos, A. J. Rodgers, B. Romanowicz, and C. Schmid (2010). Joint inversion for three-dimensional *S* velocity mantle structure along the Tethyan margin, *J. Geophys. Res.* **115,** no. 8, doi: 10.1029/2009JB007204.

EarthScope USArray Data Processing Short Course (2009–2020), available at https://www.iris.edu/hq/short-courses/ (last accessed February 2021).

Gibbons, S. J., and F. Ringdal (2006). The detection of low magnitude seismic events using array-based waveform correlation, *Geophys. J. Int.* **165,** no. 1, 149–166, doi: 10.1111/j.1365-246X.2006.02865.x.

Goldstein, P., D. Dodge, M. Firpo, and L. Minner (2003). 85.5 SAC2000: Signal processing and analysis tools for seismologists and engineers, in *International Geophysics*, Vol. 81, Academic Press, 1613–1614, doi: 10.1016/S0074-6142(03)80284-X.

Krischer, L., T. Megies, R. Barsch, M. Beyreuther, T. Lecocq, C. Caudron, and J. Wassermann (2015). ObsPy: A bridge for seismology into the scientific Python ecosystem, *Comput. Sci. Discov.* **8,** no. 1, 014003, doi: 10.1088/1749-4699/8/1/014003.

Lecocq, T., S. P. Hicks, K. van Noten, K. van Wijk, P. Koelemeijer, R. S. M. de Plaen, F. Massin, G. Hillers, R. E. Anthony, M. T. Apoloner, *et al.* (2020). Global quieting of high-frequency seismic noise due to COVID-19 pandemic lockdown measures, *Science* **369,** no. 6509, 1338–1343, doi: 10.1126/science.abd2438.

Lognonné, P., W. B. Banerdt, W. T. Pike, D. Giardini, U. Christensen, R. F. Garcia, T. Kawamura, S. Kedar, B. Knapmeyer-Endrun, L. Margerin, *et al.* (2020). Constraints on the shallow elastic and anelastic structure of Mars from InSight seismic data, *Nature Geosci.* **13,** no. 3, 213–220, doi: 10.1038/s41561-020-0536-y.

Ma, X., and A. R. Lowry (2017). USArray imaging of continental crust in the conterminous United States, *Tectonics* **36,** no. 12, 2882–2902, doi: 10.1002/2017TC004540.

McEntee, C. (2020). The state and future of AGU, *Eos*, available at https://fromtheprow.agu.org/the-state-and-future-of-agu/ (last accessed March 2021).

Myers, S. C., G. Johannesson, and W. Hanley (2007). A Bayesian hierarchical method for multiple-event seismic location, *Geophys. J. Int.* **171,** no. 3, 1049–1063, doi: 10.1111/j.1365-246X.2007.03555.x.

Myers, S. C., N. A. Simmons, G. Johannesson, and E. Matzel (2015). Improved regional and teleseismic *P*-wave travel-time prediction

and event location using a global 3D velocity model, *Bull. Seismol. Soc. Am.* **105,** no. 3, 1642–1660, doi: 10.1785/0120140272.
Oakes, R. (2020). AGU March 2020 council meeting wrap-up, *From the Prow*, available at https://fromtheprow.agu.org/agu-march-2020-council-meeting-wrap-up/ (last accessed March 2021).
Olugboji, T. M., S. Karato, and J. Park (2013). Structures of the oceanic lithosphere–asthenosphere boundary: Mineral-physics modeling and seismological signatures, *Geochem. Geophys. Geosys.* **14,** no. 4, 880–901, doi: 10.1002/ggge.20086.
Prieto, G. A., R. L. Parker, and F. L. Vernon (2009). A Fortran 90 library for multitaper spectrum analysis, *Comput. Geosci.* **35,** no. 8, 1701–1710, doi: 10.1016/j.cageo.2008.06.007.
Ross, Z. E., M. A. Meier, E. Hauksson, and T. H. Heaton (2018). Generalized seismic phase detection with deep learning, *Bull. Seismol. Soc. Am.* **108,** no. 5, 2894–2901, doi: 10.1785/0120180080.
Savage, B., and A. Snoke (2020). Thread: SAC version 102.0 is released, available at http://ftp.iris.edu/message-center/thread/6643/ (last accessed March 2021).
Shapiro, N. M., M. Campillo, L. Stehly, and M. H. Ritzwoller (2005). High-resolution surface-wave tomography from ambient seismic noise, *Science* **307,** no. 5715, 1615–1618, doi: 10.1126/science.1108339.
Smith, S. W. (1987). IRIS—A university consortium for seismology, *Rev. Geophys.* **25,** no. 6, 1203–1207, doi: 10.1029/RG025i006p01203.
Staff, A. (2020). AGU announces new strategic plan, *Eos*, available at https://fromtheprow.agu.org/agu-announces-new-strategic-plan/ (last accessed March 2021).
Tull, J. E. (1986). SAC: A signal processing system for research seismology, *Proc. of the IEEE International Symposium on Circuits and Systems*, available at https://www.osti.gov/biblio/6010224-sac-signal-processing-system-research-seismology (last accessed May 2021).
Wessel, P., J. F. Luis, L. Uieda, R. Scharroo, F. Wobbe, W. H. F. Smith, and D. Tian (2019). The Generic Mapping Tools version 6, *Geochem. Geophys. Geosys.* **20,** no. 11, 5556–5564, doi: 10.1029/2019GC008515.
Wolin, E., S. van der Lee, T. A. Bollmann, D. A. Wiens, J. Revenaugh, F. A. Darbyshire, A. W. Frederiksen, S. Stein, and M. E. Wysession (2015). Seasonal and diurnal variations in long-period noise at SPREE stations: The influence of soil characteristics on shallow stations' performance, *Bull. Seismol. Soc. Am.* **105,** no. 5, 2433–2452, doi: 10.1785/0120150046.
Woodward, R. (2014). *Sweetwater Array*, International Federation of Digital Seismograph Networks, available at http://www.fdsn.org/networks/detail/XB_2014/ (last accessed March 2021).

Manuscript received 9 November 2020
Published online 26 May 2021