Electronic Seismologist

easyQuake: Putting Machine Learning to Work for Your Regional Seismic Network or Local Earthquake Study

Jacob I. Walter*[1], Paul Ogwari[1], Andrew Thiel[1], Fernando Ferrer[1], and Isaac Woelfel[1]

Abstract

We developed a Python package—easyQuake—that consists of a flexible set of tools for detecting and locating earthquakes from International Federation of Digital Seismograph Networks-collected or field-collected seismograms. The package leverages a machine-learning driven phase picker, coupled with an associator, to produce a Quake Markup Language (QuakeML) style catalog complete with magnitudes and *P*-wave polarity determinations. We describe how nightly computations on day-long seismograms identify lower-magnitude candidate events that were otherwise missed due to cultural noise and how those events are incorporated into the Oklahoma Geological Survey statewide network upon analyst manual review. We discuss applications for the package, including earthquake detection for regional networks and microseismicity studies in arbitrary user-defined regions. Because the fundamentals of the package are scale invariant, it has wide application to seismological earthquake analysis from regional to local arrays and has great potential for identifying early aftershocks that are otherwise missed. The package is fast and reliable; the computations are relatively efficient across a range of hardware, and we have encountered very few (~1%) false positive event detections for the Oklahoma case study. The utility and novelty of the package is the turnkey earthquake analysis with QuakeML file output, which can be dropped directly into existing real-time earthquake analysis systems. We have designed the functions to be quite modular so that a user could replace the provided picker or associator with one of their choosing. The Python package is open source and development continues.

Cite this article as Walter, J. I., P. Ogwari, A. Thiel, F. Ferrer, and I. Woelfel (2020). easyQuake: Putting Machine Learning to Work for Your Regional Seismic Network or Local Earthquake Study, *Seismol. Res. Lett.* **XX**, 1–9, doi: 10.1785/0220200226.

Motivation

Automatic earthquake identification, phase arrival pick association, and event location can be conducted by myriad different open-source and proprietary algorithms and computer programs. For the earth scientist, there are no or limited barriers to collecting data from open local experiments or archives and producing an earthquake catalog. To identify an earthquake within a continuous seismogram, often the seismologist will utilize a detection algorithm that identifies impulsive pulses; for example, a short-term average to long-term average trigger is sufficient to identify a pick on a single seismograph component at a single station. Picks from multiple stations that cluster in time may identify a possible earthquake. Given a velocity model, travel-time lookup tables are referenced to determine a distance based on relative *S*-*P* times, and an earthquake origin location can be determined or further refined with the relative *P*-wave travel times at different stations. Subsequent analysis might yield a magnitude, relocation of the hypocenter, or other ancillary information (focal mechanism and so on) about the source.

Machine learning in seismology has rapidly evolved with recent enhancements to earthquake detection (Perol *et al.*, 2018; Ross *et al.*, 2018; Kong *et al.*, 2019) and earthquake pick association (Ross *et al.*, 2019), which enhances research efforts that utilize those small earthquakes. Machine-learning catalogs can identify smaller earthquakes than might otherwise be determinable with routine processing, and they have the added advantage that they are not reliant upon pre-existing template

1. Oklahoma Geological Survey, University of Oklahoma, Norman, Oklahoma, U.S.A.
*Corresponding author: jwalter@ou.edu

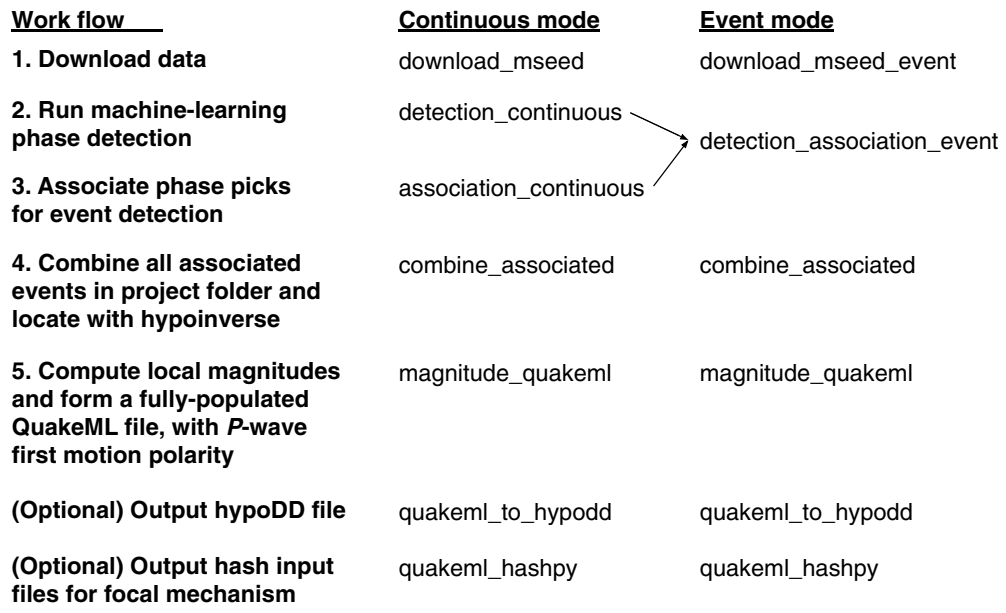


Figure 1. Workflow within the easyQuake package for detection, location, magnitude calculation, and other optional additional analysis steps for both continuous and event modes.

earthquakes in network matched-filter methods (Walter *et al.*, 2017) or repeating signals (e.g., Skoumal *et al.*, 2016).

Often, earthquake catalogs produced for research purposes lack the earthquake association information (pick times, stations utilized, and so on) for identified events, when those studies are published. Meanwhile, earthquake information from earthquake observatories worldwide have gradually shifted to utilizing the Quake Markup Language (QuakeML) for earthquake event metadata, which is an open international definition in the extensible markup language (XML) format. The standard allows the sharing of event data among observatories and is the de-facto standard for event queries to various International Federation of Digital Seismograph Networks (FDSN) webservices. For example, QuakeML is utilized by regional seismic networks (e.g., Walter *et al.*, 2020) within the Advanced National Seismic System (ANSS) to push solutions to the U.S. Geological Survey (USGS) that forms the Comprehensive Catalog (ComCat). Observatory-derived earthquake catalog products in the QuakeML format are rich sources of information, are easily accessible, as well as are reproducible datasets.

It is certainly possible that machine learning will unveil fundamental earthquake behaviors, as the field evolves. The ability to detect smaller and smaller earthquakes could yield surprising discoveries related to earthquake nucleation, the spectrum of fault-slip behaviors, and could provide unique insight that eventually enables short-term forecasting. Many of these advanced seismological tools do not automatically output richly detailed datasets, such as the details found in a QuakeML file. We specifically guided our development of easyQuake to make earthquake regional network monitoring and research studies more reproducible and verifiable, by outputting the detailed QuakeML file. We assembled a Python package using various existing components with some modification and have written several utilities so that the package can serve as a turnkey solution for taking raw seismograms and producing a robust research-grade earthquake catalog. The package we introduce here strives to strike the balance between reproducibility while improving the performance of earthquake identification available through other methods. We describe several potential uses for easyQuake, including an example from a recent event. The main driver of this article includes a case study for how we have implemented it to augment our routine real-time earthquake analysis and cataloging of events in Oklahoma.

Running easyQuake

The easyQuake suite relies on several recent Python packages released in the last 5 yr. First and foremost, ObsPy (Krischer *et al.*, 2015) underlies all of the reading and writing of seismograms and event metadata. The ObsPy package is well documented and broadly utilized among seismologists. Essentially, easyQuake guides all steps of the earthquake catalog generation workload, including, producing candidate phase arrival detection, gathering of picks for earthquake association, locating events during association, and magnitude determination. We also include several utilities that output necessary input files for hypoinverse (Klein, 2002) and hypoDD (Waldhauser and Ellsworth, 2000) input files for earthquake location and relocation, respectively. A more detailed summary of each analysis step follows. Each step of the analysis (Fig. 1) is driven by a separate subprogram of the easyQuake package to provide the utmost modularity in all the analysis steps and provide flexibility for handling larger datasets, which is described later.

Download data—Gather seismograms

We start with gathering seismograms from relevant FDSN data repositories. Optionally, easyQuake can be pointed toward existing data that are organized around a structure in which day-long data reside in folders designating the day (YYYYMMDD), within a project folder. Although seismograms are downloaded

in miniSEED format, existing datasets can be processed if they are in miniSEED or Seismic Analysis Code formats. In general, all the data are organized around a top-level project folder (the user-assigned variable project_folder is the full path for the project directory on the user's computer). Below this folder, directory names that correspond to individual days or assigned event times (variable dirname is the sub directory in which pick, travel time, and 1D associator tables are stored.). If data are gathered from FDSN sources, station metadata, including instrument response, will be downloaded into the day directory. If data are locally gathered, easyQuake is programmed to assume that the metadata also exists in the day directory, though, if it is not there or does not exist, it will attempt to gather it from FDSN webservices during later analysis steps.

Earthquake detection

By default, easyQuake utilizes machine learning for the earthquake phase detection, by incorporating a system call to the Generalized Phase Detection (GPD) picker (Ross *et al.*, 2018). The GPD picker utilizes a convolutional network trained on millions of seismograms from southern California, to predict the pick times and identify the body-wave phase type of potential arrivals at individual stations. The picker leverages machine-learning software that is significantly sped up by the massively parallel chip architecture of a graphics processing unit (GPU). By default, easyQuake is configured to utilize 1 GPU; users with multiple GPUs should identify the file location of the picker script within the easyQuake package and edit accordingly. In addition, although the GPD picker will run on a CPU, it will be incredibly slow without an NVIDIA video card with the Compute Unified Device Architecture (CUDA) parallel computing platform installed on the machine. If hardware limitations prevent utilizing the GPD picker, users can also optionally select a frequency-band (FB) picker, which is based on Lomax *et al.* (2012) and included in the PhasePApy package (Chen and Holland, 2016).

Earthquake association

For earthquake association, which gathers available picks, clusters them in time and attempts to associate events utilizing travel-time lookup tables, we utilize the PhasePApy 1D associator (Chen and Holland, 2016). Despite the availability of deep-learning phase associators (e.g., Ross *et al.*, 2019), we choose a more standard associator for simplicity in that users have the ability to choose the thresholds inherent to earthquake association, for example, epicentral distance, number of associated candidates before declaring an event, and so on. PhasePApy utilizes travel-time lookup tables that are computed using the TauP (Crotwell *et al.*, 1999) submodule within ObsPy and using the IASP91 (Kennett and Engdahl, 1991) global velocity model. The user can provide their own velocity model, if they so choose.

The first step of the association process is to aggregate picks from the same station on different components and to utilize *S-P* times to identify possible paired *P*/*S* picks for the station. In our testing, the GPD picker did not satisfactorily deduce *S* picks with a high degree of confidence, and, thus, we added this conservative step to force the *S-P* travel times to dictate possible candidate events at a single station. Next, the associator determines candidate *S-P* distances from the *S-P* travel times and corresponding travel-time lookup tables. The associator determines an origin time based on the *S-P* distance and backprojecting a corresponding *P*-wave travel time. Because origin times will naturally cluster when verifiable events occur, the associator next clusters those origin times based on a user-defined value for the number of stations.

Locations are determined by minimizing the residuals between the *S*–*P* distances and the epicenter-to-station distance. Further details on the PhasePApy associator and other possible user-defined thresholds for optimization are described in detail in Chen and Holland (2016) and should be consulted when making changes to the association that deviate from the default parameters. Although some of the easyQuake functions directly control the associator, such as the maximum *S-P* distance, others, such as the number of clustered candidate stations, before an event is declared, can be edited by directly editing the PhasePApy code that is distributed within the easyQuake package.

The simplicity of this associator makes the earthquake solutions easier to understand for the novice user. Furthermore, the separation between detection and association within the package allows the user to vary thresholds within both detection and association steps to optimize event detection for the unique geometry of their regional network or earthquake study.

One of the drawbacks of the currently implemented associator is the relative slow speed of association when dealing with a large number of picks derived from the machine-learning detection stage. This issue grows, as the detection area or density of stations incorporated into the analysis grows. Various strategies could be implemented to improve slowdowns associated with earthquake association. First, users could select overlapping subnetworks within the broader regional detection area; this strategy is sometimes implemented in real-time earthquake processing. Second, the user can increase the hyper parameter associated with pick probability in the modified GPD picker script (gpd_predict.py) included within the package, which should speed up association by reducing the number of candidate picks. By default, it is set to 0.994, which in our testing provided a good balance between robust detection and minimizing false positive detections. Last, a user could run the detection step separate from association, and run it serially for several days. In another script, one would want to parallelize the association steps using a computer with several processor cores and sufficient random

access memory (RAM). We provide some example scripts in the Github repository that demonstrate how this could be done. In future releases, we plan to improve the earthquake association component of the package by either modifying the PhasePApy associator further or exploring incorporating a machine-learning associator.

Combine all project associations and format for QuakeML

Once all earthquakes have been associated, the events and stations databases exist as sqlite-formatted database files within individual day or event folders within the project directory. The next step combines all events within those folders into a QuakeML catalog file through the use of ObsPy tools. The user has the further option to split the catalog into individual QuakeML event files. The QuakeML standard has specific prescriptions for how the metadata for an earthquake should be organized in a hierarchical way. The populating of these fields for each event, including, Pick, Arrival, and Origin objects, occurs during this step. During the conversion from sqlite association tables to the ObsPy catalog objects that are QuakeML-compatible, we also create hypoinverse station and event files that are saved to the project top-level directory and can be used for later relocation. The initial hypocenters generated by the associator can be fairly coarse, because the travel-time tables are calculated with TauP on a generic velocity model.

Local magnitude calculation

Once all earthquakes have been associated, and the events are in QuakeML style formats as a catalog object, we can then calculate local magnitudes. We follow recent International Association of Seismology and Physics of the Earth's Interior (IASPEI) guidance (Bormann and Dewey, 2014) that local magnitudes should be calculated with the Hutton and Boore (1987) formulation for earthquake magnitudes for areas with attenuative properties similar to California (Bormann and Dewey, 2014). The formulation for a station magnitude estimate is:

$$M_L = \log_{10}(A) + 1.11\log_{10}(D) + 0.00189D - 2.09, \quad (1)$$

in which A is the horizontal displacement maximum amplitude in nm from a seismograph in which the modern instrument response is deconvolved, and a digital response for the Wood–Anderson seismometer is convolved to simulate a reading from that type of seismometer. We use a 2080 sensitivity and damping constant of 0.7 for the Wood–Anderson response (Uhrhammer and Collins, 1990). Following the IASPEI recommendation, each horizontal displacement is measured independently, so there may be two estimates for magnitude from a given station. We combine all these station-magnitude estimates within a 160 km radius and compute a median value, which is the event magnitude. We utilize a cutoff of 160 km to avoid the critical refraction distance in which *Lg* waves are prominent for ~40 km thick crust. The standard deviation of the station magnitudes is the magnitude uncertainty. The individual station magnitudes, event magnitude, and event magnitude uncertainty are included in the QuakeML catalog file. In practice, if one were to utilize the package in other regions or were conducting a research study in which an accurate magnitude is desired, one would utilize a more generalized form of equation (1) and determine coefficients that correspond to the local attenuative properties of the study area (e.g., Walter *et al.*, 2018).

Further earthquake analysis

We include utilities within easyQuake to reingest the QuakeML files and produce the files necessary to run hypoinverse (Klein, 2002) and hypoDD (Waldhauser and Ellsworth, 2000). Furthermore, during the earthquake magnitude step, polarities for *P* waves are estimated by searching for a local minimum or maximum after the *P*-wavepick time, as long as the absolute value of the pick amplitude is at least five times greater than the standard deviation of the seismic waveform in a 5 s window before the *P*-wavepick time. The polarity pick is added to the relevant QuakeML file. We also include modules for calculating a HASH focal mechanism through the use of the hashpy Python module (see Data and Resources). We plan to further extend the capability of earthquake analysis within easyQuake, to include the automatic production of maps and various statistical properties of the seismicity catalogs within individual projects.

Case Studies

Augmenting the detection capability of a regional seismic network

At the Oklahoma Geological Survey (OGS), we utilize the SeisComP3 (SC3) real-time earthquake processing software to detect, associate, review, and disseminate earthquake events to state stakeholders and through the USGS Product Distribution Layer (Walter *et al.*, 2020). As the authoritative regional network under the ANSS for the state, those solutions are posted by USGS and archived into the ComCat. Thus, cataloged earthquakes should be verified before posting. To reduce the chance of a false positive event, we conduct a manual review of any machine-learning detected earthquake prior to public release.

When the UTC day ends (7 p.m. central daylight time), we run a nightly detection on the data we generated in the past 24 hr. During the next business day, analysts will look through the detections and then manually trigger events within the SC3 system, if they can verify that the M_L-detected event has sufficient coverage to work the event in SC3. Because the event is already in the QuakeML format, it is trivial to load the candidate event information directly into the SC3 internal processing database that aggregates the origin, pick, amplitude, and so

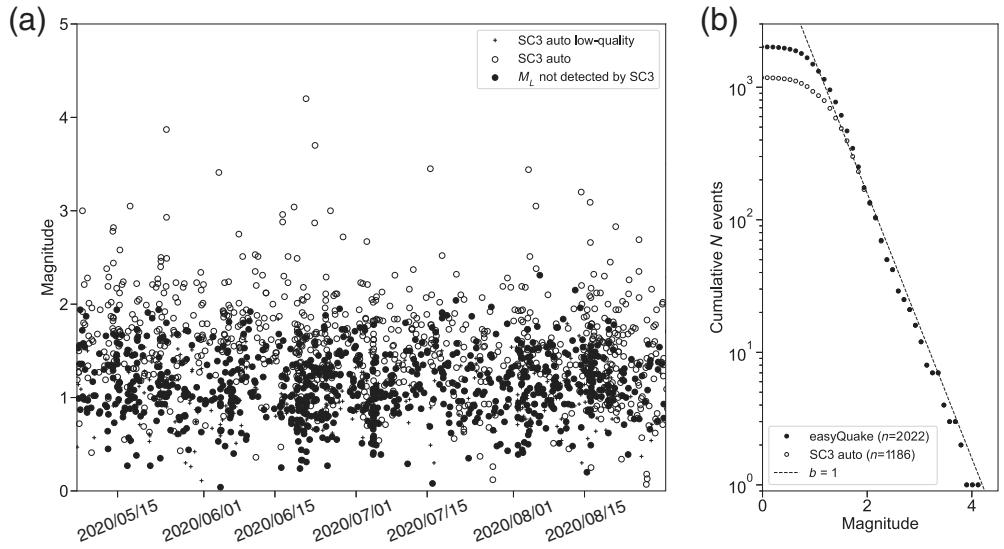


Figure 2. Performance of easyQuake relative to the real-time SeisComP3 (SC3) system at Oklahoma Geological Survey (OGS). (a) Earthquake magnitudes with time, since we started running easyQuake nightly to augment the real-time SC3 detection. Black circles identify earthquakes detected by easyQuake that were otherwise missed entirely by SC3. (b) Gutenberg–Richter style plot of cumulative histogram of detected events within 0.1 magnitude bins, indicating an ~0.7 unit magnitude of completeness improvement. A Gutenberg–Richter *b*-value equal to 1 is plotted as a dashed line.

on. We are able to strip away the event identifier information, which is created within easyQuake automatically, and only pass the pick, arrival, and origin information to SC3, in which the system will associate a new event. In this way, the detected events are indistinguishable from regularly detected SC3 events, and the M_L-detected events are queued for analysis that occurs during each working day. Once the event is reviewed by the analyst, it is released to the public. The script to run the daily detection and add it to the SC3 system is included in the Github repository in the examples folder.

We find that, in general, the easyQuake package performs quite well at augmenting the regional network. Although we have only been running the nightly detection for the last month, we identify a factor of ~2 more earthquakes than the real-time SC3 system (Fig. 2). Often, the earthquakes detected by easyQuake but missed by SC3 tend to be in central locations (where station density is higher) and have small magnitudes, or they are slightly larger magnitude events in areas where the station geometry is less dense (Fig. 3). In addition, easyQuake appears to identify more quarry events than were detectable with just SC3. This is, in part, due to the fact that more quarries are located in eastern Oklahoma, where station density is lower. Overall, easyQuake lowers the magnitude of completeness statewide by ~0.7 magnitude units (Fig. 2).

Evaluating efficacy and future improvements

Machine-learning convolutional neural networks use rather large datasets to train a model that can then be utilized to make decisions on future datasets. As stated previously, we chose to use the GPD picker and the corresponding model that was trained on seismograms from southern California. If we had chosen to train our own model using Oklahoma seismograms, one way to validate the picker would be to test or validate on a portion of the dataset not in the training dataset to evaluate how effectively the model recalls the labeled pick. However, we chose not to train a new Oklahoma-specific model, in part, because we found that the GPD picker suits our initial purposes in reducing the catalog completeness by ~0.7 magnitude units. The central purpose of easyQuake is facilitating the turnkey earthquake detection and location, and we leave it up to future users to determine what parameters are appropriate for their use case. There are several layers of possible variations from phase detection to event association that can be tweaked that it would be difficult to evaluate the effectiveness when the true number of earthquakes that could be detected within the noise is an unknown quantity.

The improvement in detection ability at OGS appears to stem from the GPD picker portion of the easyQuake package. It performs well at identifying earthquakes that would otherwise not have been flagged by the Akaike information criteria (AIC) picker that we currently utilize within SC3. Figure 4 provides a qualitative example of the performance of the GPD picker within easyQuake versus the SC3 AIC picker. Figure 4a indicates that several more picks were identified by easyQuake relative to the SC3 system, though both approaches successfully identified and associated this event. Figure 4b shows an earthquake that had no SC3 picks, whereas easyQuake identified a sufficient number of picks to associate an event. One caveat is that the GPD picker does produce many false-positive phase picks and does not always correctly identify the phase type. As described previously, the PhasePApy associator does not ingest the phase information at this time, so it is not impacted by incorrect phase type identification. Rather, the associator identifies only the pick time when it attempts to cluster picks in time to trigger the next steps in the event association.

As shown in Figure 2, easyQuake augmented routine analysis efforts at OGS during the time period from 7 May 2020 to

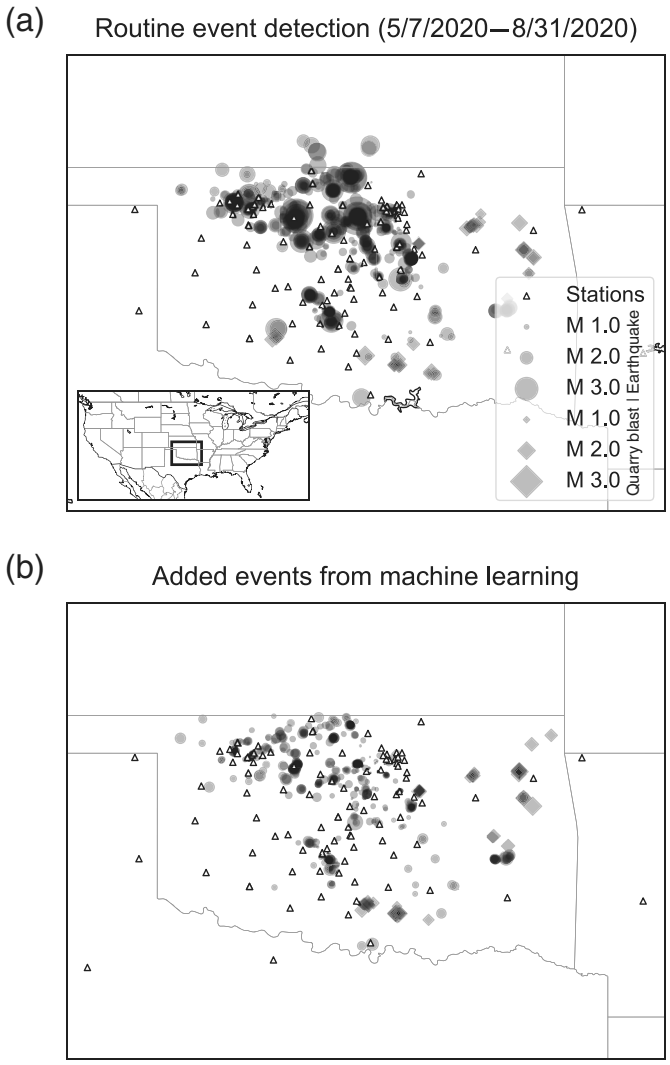


Figure 3. Example earthquake (circles) and quarry event (diamonds) detections from several weeks of routine and easyQuake processing of network data. (a) Map includes only events identified by the SC3 analysis system. Inset shows the North America with the study region highlighted by the black box. (b) Map includes those events that were missed by SC3 but identified by easyQuake, which are entered into the regional catalog.

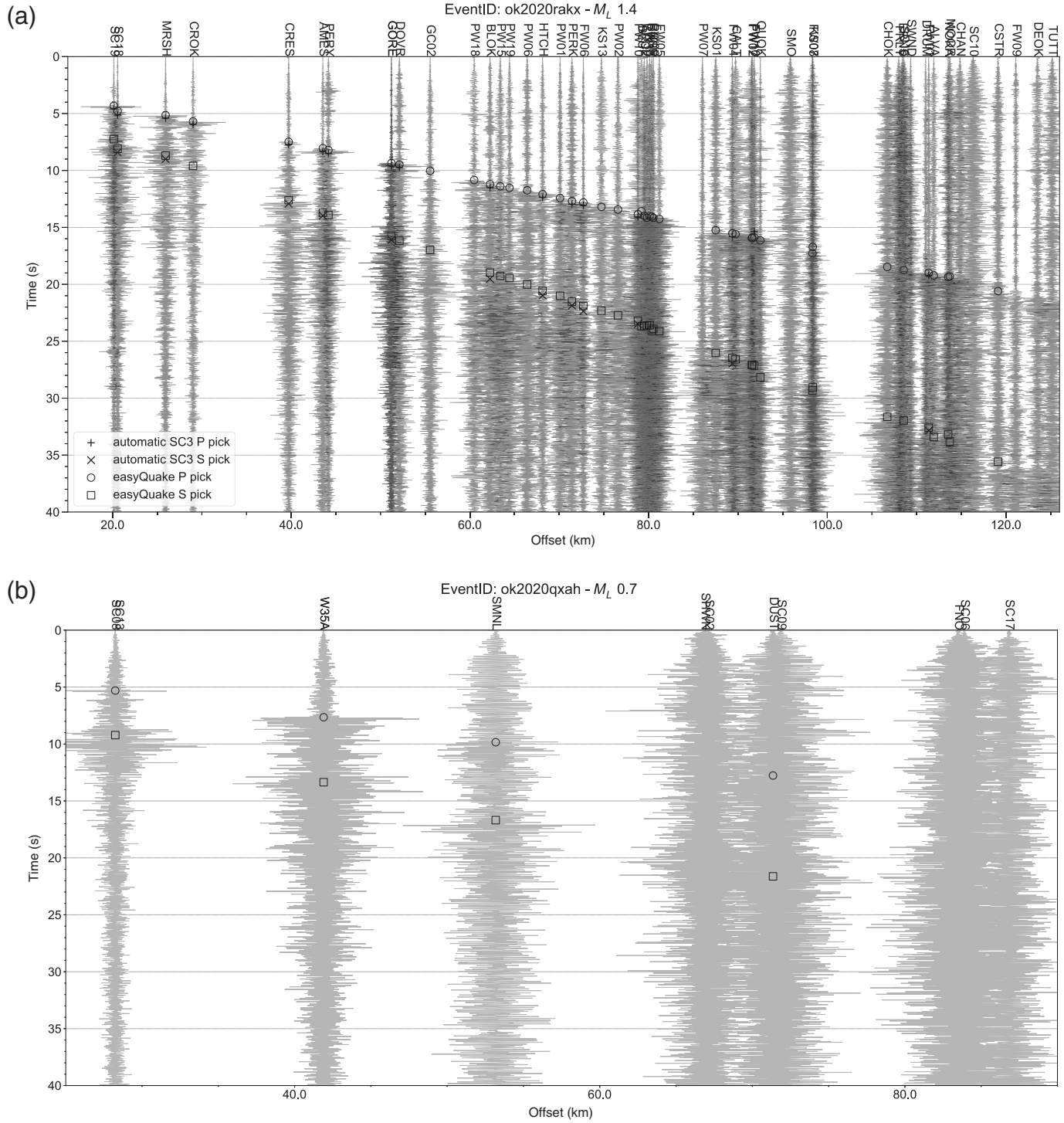
31 August 2020, by detecting an additional 836 events that would have been otherwise undetected, which is a 70% improvement in the number of reported earthquakes (1186 events would have been reported solely with SC3). Those additional events were verified and often repicked by analysts. We identified no instances when easyQuake failed to identify an SC3-detected event and 23 instances in which an analyst could not verify an earthquake that occurred during the origin time identified by easyQuake. The 23 instances are either true false positives, in which random noise picks are associated as a possible event, or an event has occurred but is not immediately identifiable by an experienced analyst. Because OGS functions as a regional network, providing verifiable earthquake information, we simply categorize such unconfirmable events as false positives. During that time period, 23 unconfirmed out of 2022 confirmed events that suggest 1% of events could be false positives, with the default settings in the version of easyQuake distributed in Github.

The 23 candidate events that were unconfirmable had preliminary magnitudes M_L 1.3–1.8, as calculated with equation (1). Those magnitudes and all easyQuake detections tend to have magnitudes that are systematically ~0.4 magnitude units higher than when confirmed within SC3, because we utilize a magnitude scale that is region-specific to Oklahoma (e.g., Walter *et al.*, 2020). This suggests that smaller magnitude earthquakes may have slightly higher chance of associating a false event. To prevent possible false positives, a user could take a number of steps, including increasing the number of stations required to declare an event within the association step or altering parameters within the GPD picker to reduce the chance of noise triggering a pick declaration. Although the total number of detected events would decrease, there would be a smaller chance of a false-positive event declaration.

We plan future improvements to ease the addition of additional pickers or additional associators, which may be interchanged based upon user's preference or computational improvements. For example, as stated previously, the associator is much slower than the picker portion. We have tested the daily OGS calculations on a machine with an NVIDIA Geforce 1050 Ti, with an overall value of ~600 U.S. when it was purchased 4 yr ago. During a test of a daily calculation, it requires several hours to detect and associate a day of data in which we identify 10–15 earthquakes. However, these estimates would vary wildly, depending upon the hardware that is used. This is clearly the end member of minimum system capability that one should utilize with easyQuake; the specified hardware is at the minimum level of CUDA compatibility to run the GPD picker component of easyQuake.

Other applications and early aftershock detection

The flexibility of the easyQuake module is such that it would be useful for a wide range of problems, when the science is advanced by increasing the number of identified earthquakes. One such example would be to rapidly identify earthquakes after large events, when the regional network may be slow to respond or when a regional network does not exist. If there are sufficient data that are FDSN-archived, these data can be rapidly processed. For example, we show an example of the easyQuake package applied to a region surrounding the 31 March 2020 central Idaho earthquake (Fig. 5). We identify an order of magnitude more events (15,367 earthquakes) than available for download through ComCat (1693). The easyQuake-identified events have magnitude values that are systematically ~0.6 units higher magnitude than the ComCat events, likely due to the IASPEI-guided magnitude calculation utilized within easyQuake. If one were pursuing an expanded study



of the Idaho earthquake, one might want to calibrate their own local magnitude distance attenuation function to USGS ComCat waveform-based moment tensor magnitudes (e.g., Walter *et al.*, 2018). For this demonstration of easyQuake to central Idaho, it required only defining the region and time period of analysis (see Github for example code).

The easyQuake package would also be suitable for densifying aftershock catalogs. For example, Rosson *et al.* (2019) identified relatively high Omori aftershock decay *c*-values for

Figure 4. Record section plots for two events in Oklahoma. (a) Event ok2020rakx, an M_L 1.4 earthquake, was detected and located automatically, and both SC3 and easyQuake picks are shown. (b) For event ok2020qxah, an M_L 0.7 earthquake, there was no location and no picks detected by SC3, whereas easyQuake identified several picks and associated an event that was later confirmed by an OGS analyst.

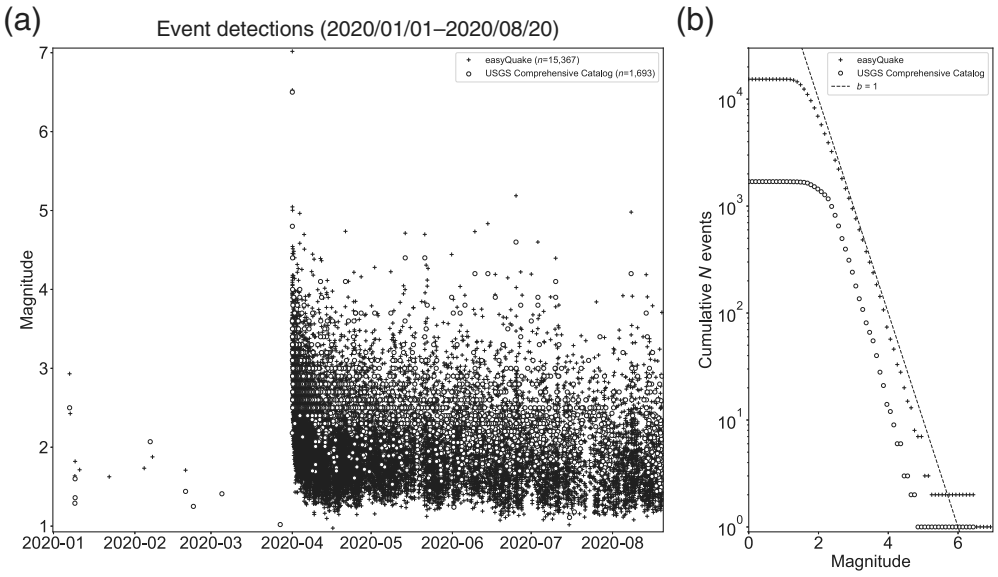


Figure 5. Example use case for identifying aftershocks for the March 2020 M_w 6.5 central Idaho earthquake. (a) Earthquake magnitudes with time in which the white circles identify earthquakes downloaded from Comprehensive Catalog. Black plus signs indicate easyQuake-identified earthquakes. (b) Gutenberg–Richter style plot of cumulative histogram of detected events for both catalogs. A Gutenberg–Richter *b*-value equal to 1 is plotted as a dashed line. The higher cumulative values for easyQuake-identified events indicate systematically higher magnitudes, because those event magnitudes are calculated using a generic rather than a regionally tuned attenuation function for local magnitude.

Oklahoma mainshock–aftershock sequences. Large *c*-values could indicate incomplete earthquake catalogs, as a result of unidentified early aftershocks within the coda or trailing the coda of the larger mainshocks. easyQuake may help in identifying early aftershocks in a systematic way for the myriad large earthquakes that occur in areas with dense instrumentation. Goebel *et al.* (2019) identified a reduced aftershock efficiency for Oklahoma earthquakes, when the state regulatory agency required injection well volume reductions after larger events. The easyQuake package could be used to extend such analysis and test the robustness of various other aftershock-related hypotheses. For example, systematically improved aftershock catalogs may improve the quality of aftershock forecasts after large events (e.g., Michael, 2018).

We demonstrated a workflow whereby we redundantly run easyQuake to identify smaller earthquakes. However, it could be routinely utilized by regional networks to ensure that earthquakes above a network's nominal magnitude of completeness are not missed. In situations where networks are overwhelmed, such as during aftershocks or volcanic eruptions, one might envision a future in which easyQuake or similar approaches could be deployed to augment regional networks in an ad hoc way with cloud-based infrastructure. The cloud offers an environment in which computational resources can elastically increase, based on operational need after large events or eruptions, as long as regional networks are prepared to handle and trust the externally created data products.

Conclusion and Future Work

We showcased the capability of the easyQuake package and demonstrated that it augments the performance of the real-time network at OGS. Because it is written in a flexible manner, it can easily be applied to recent events in which the data are freely available at FDSN archives. We showed that in Oklahoma, easyQuake augments our real-time capabilities and increases the number of detected events by a factor of 2. In one other case study, we demonstrated that applying the package to Central Idaho after the 31 March 2020 earthquake yields a factor of 10 more detected events.

Although machine learning holds tremendous promise to improve routine human analysis of earthquakes and enhance our scientific understanding of the earthquake process, some caution is necessary. The role of regional seismic networks is to deliver timely and reliable earthquake information. Machine learning is at the stage in which regional networks should leverage the efficiencies inherent in identifying events that would be otherwise obscured by noise. However, regional networks are run by trusted organizations that serve various stakeholders, primarily the public. The trusted relationship relies upon remaining a credible information resource and cannot afford the possibility of faulty or false information. When large magnitude or complex events occur, it is critical to retain well-trained and experienced analysts and scientists to process the trusted data and disseminate contextual information to the public. Although the future seems limitless for machine learning's role in seismology, those capabilities should be balanced with concerted workforce support and training.

Data and Resources

All the necessary scripts and some example files are available on Github (https://github.com/jakewalter/easyQuake). The package is installable with the Python package manager, Python Package Index ("pip install easyQuake"). Additional examples and guidance on utilizing easyQuake can be obtained by contacting the authors. We utilize datasets from various regional seismic networks, and those

real-time data are available at the Incorporated Research Institutions for Seismology (IRIS) Data Management Center at ds.iris.edu under Federated Digital Seismic Network codes OK (doi: 10.7914/SN/OK), O2 (doi: 10.7914/SN/O2), RC (doi: 10.7914/SN/RC), IW (doi: 10.7914/SN/IW), IE (doi: 10.7914/SN/IE), UU (doi: 10.7914/SN/UU), US (doi: 10.7914/SN/US), and UW (doi: 10.7914/SN/UW). Within the main article, we mention the use of the hashpy Python module (https://github.com/markcwill/hashpy). The Oklahoma Geological Survey (OGS) catalog may be downloaded from the U.S. Geological Survey (USGS) Comprehensive Catalog (ComCat; https://earthquake.usgs.gov/data/comcat/) or at the OGS webpage (https://ogsweb.ou.edu/eq_catalog/). The Idaho easyQuake catalog is available (https://github.com/jakewalter/easyQuake/blob/master/examples/catalog_idaho.zip) as a zipped file containing a QuakeML-formatted catalog that can be read with the ObsPy Python package. The Idaho ComCat dataset may be downloaded from the USGS ComCat. All websites were last accessed in September 2020.

Acknowledgments

The authors thank Honn Kao and an anonymous reviewer for comments that greatly improved the article. The authors would like to acknowledge funding from the State of Oklahoma through Oklahoma Geological Survey (OGS) core funding as well as supplementary funding through the Office of the Oklahoma Secretary for Energy and the Environment to fund ongoing seismic network operations. Jacob I. Walter was supported partially by National Science Foundation (NSF) Award 1745135.

References

Bormann, P., and J. W. Dewey (2014). The new IASPEI standards for determining magnitudes from digital data and their relation to classical magnitudes, in *New Manual of Seismological Observatory Practice 2 (NMSOP-2)*, P. Bormann (Editor), Deutsches GeoForschungsZentrum GFZ, Potsdam, Germany, 1–44.

Chen, C., and A. A. Holland (2016). PhasePApy: A robust pure Python package for automatic identification of seismic phases, *Seismol. Res. Lett.* **87,** no. 6, 1384–1396, doi: 10.1785/0220160019.

Crotwell, H. P., T. J. Owens, and J. Ritsema (1999). The TauP toolkit: Flexible seismic travel-time and ray-path utilities, *Seismol. Res. Lett.* **70,** 154–160.

Goebel, T., Z. Rosson, E. E. Brodsky, and J. I. Walter (2019). Aftershock deficiency of induced earthquake sequences during rapid mitigation efforts in Oklahoma, *Earth Planet. Sci. Lett.* **522,** 135–143, doi: 10.1016/j.epsl.2019.06.036.

Hutton, L. K., and D. M. Boore (1987). The M_L scale in southern California, *Bull. Seismol. Soc. Am.* **77,** no. 6, 2074–2094.

Kennett, B. L. N., and E. R. Engdahl (1991). Travel times for global earthquake location and phase association, *Geophys. J. Int.* **105,** 429–465, doi: 10.17611/DP/9991809.

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. Disc.* **8,** 014003.

Klein, F. W. (2002). *User's guide to HYPOINVERSE-2000, a Fortran program to solve for earthquake locations and magnitudes*, *U. S. Geol. Surv. Open-File Rept. OF 02-171*.

Kong, Q., D. T. Trugman, Z. E. Ross, M. J. Bianco, B. Meade, and P. Gerstoft (2019). Machine learning in seismology—Turning data into insights, *Seismol. Res. Lett.* **90,** 3–14, doi: 10.1785/0220180259.

Lomax, A., C. Satriano, and M. Vassallo (2012). Automatic picker developments and optimization: FilterPicker: A robust, broadband picker for real-time seismic monitoring and earthquake early warning, *Seismol. Res. Lett.* **83,** no. 3, 531–540, doi: 10.1785/gssrl.83.3.531.

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,** doi: 10.1785/0120180080.

Ross, Z. E., Y. Yue, M.-A. Meier, E. Hauksson, and T. H. Heaton (2019). PhaseLink: A deep learning approach to seismic phase association, *J. Geophys. Res.* **124,** no. 1, 856–869.

Rosson, Z., J. I. Walter, T. Goebel, and X. Chen (2019). Narrow spatial aftershock zones for induced earthquake sequences in Oklahoma, *Geophys. Res. Lett.* **46,** doi: 10.1029/2019GL083562.

Michael, A. J. (2018). *On the potential duration of the aftershock sequence of the 2018 Anchorage earthquake*, *U.S. Geol. Surv. Open-File Rept. 2018-1195*, 6 pp., doi: 10.3133/ofr20181195.

Perol, T., M. Gharbi, and M. Denolle (2018). Convolutional neural network for earthquake detection and location, *Sci. Adv.* **4,** no. 2, e1700578, doi: 10.1126/sciadv.1700578.

Skoumal, R. J., M. R. Brudzinski, and B. S. Currie (2016). An efficient repeating signal detector to investigate earthquake swarms, *J. Geophys. Res.* **121,** 5880–5897, doi: 10.1002/2016JB012981.

Uhrhammer, R. A., and E. R. Collins (1990). Synthesis of Wood–Anderson seismograms from broadband digital records, *Bull. Seismol. Soc. Am.* **80,** 702–716.

Waldhauser, F., and W. L. Ellsworth (2000). A double-difference earthquake location algorithm: Method and application to the northern Hayward fault, *Bull. Seismol. Soc. Am.* **90,** 1353–1368.

Walter, J. I., J. C. Chang, and P. J. Dotray (2017). Foreshock seismicity suggests gradual differential stress increase in the months prior to the 3 September 2016 M_w 5.8 Pawnee earthquake, *Seismol. Res. Lett.* **88,** no. 4, 1032–1039, doi: 10.1785/0220170007.

Walter, J. I., C. Frohlich, and T. Borgfeldt (2018). Natural and induced earthquakes in the Texas and Oklahoma Panhandles, *Seismol. Res. Lett.* **89,** no. 6, 2437–2446, doi: 10.1785/0220180105.

Walter, J. I., P. Ogwari, A. Thiel, F. Ferrer, I. Woelfel, J. C. Chang, A. P. Darold, and A. A. Holland (2020). The Oklahoma Geological Survey statewide seismic network, *Seismol. Res. Lett.* **91,** no. 2A, 611–621, doi: 10.1785/0220190211.

Manuscript received 23 June 2020
Published online 4 November 2020