Electronic Seismologist

DynTriPy: A Python Package for Detecting Dynamic Earthquake Triggering Signals

Naidan Yun[1], Hongfeng Yang[2], and Shiyong Zhou[*1]

Abstract

Long-term and large-scale observations of dynamic earthquake triggering are urgently needed to understand the mechanism of earthquake interaction and assess seismic hazards. We developed a robust Python package termed DynTriPy to automatically detect dynamic triggering signals by distinguishing anomalous seismicity after the arrival of remote earthquakes. This package is an efficient implementation of the high-frequency power integral ratio algorithm, which is suitable for processing big data independent of earthquake catalogs or subjective judgments and can suppress the influence of noise and variations in the background seismicity. Finally, a confidence level of dynamic triggering (0–1) is statistically yielded. DynTriPy is designed to process data from multiple stations in parallel, taking advantage of rapidly expanding seismic arrays to monitor triggering on a global scale. Various data formats are supported, such as Seismic Analysis Code, mini Standard for Exchange of Earthquake Data (miniSEED), and SEED. To tune parameters more conveniently, we build a function to generate a database that stores power integrals in different time and frequency segments. All calculation functions possess a high-level parallel architecture, thoroughly capitalizing on available computational resources. We output and store the results of each function for continuous operation in the event of an unexpected interruption. The deployment of DynTriPy to data centers for real-time monitoring and investigating the sudden activation of any signal within a certain frequency scope has broad application prospects.

Cite this article as Yun, N., H. Yang, and S. Zhou (2020). DynTriPy: A Python Package for Detecting Dynamic Earthquake Triggering Signals, *Seismol. Res. Lett.* **XX**, 1–12, doi: 10.1785/0220200216.

Supplemental Material

Introduction

It has been widely reported that earthquakes can be triggered by transient stress changes related to the passage of seismic waves; this phenomenon is known as dynamic triggering (e.g., Hill *et al.*, 1993; Gomberg and Johnson, 2005; Velasco *et al.*, 2008; Peng *et al.*, 2010, 2012; Pollitz *et al.*, 2012; van der Elst *et al.*, 2013; Aiken *et al.*, 2018). Systematic investigations of dynamic triggering can help us explore the responses of faults to stress disturbances, and these responses can be leveraged to monitor temporal variations in stress and thus assess earthquake hazards. Moreover, the mechanisms of earthquake interactions are critical for advancing our understanding of earthquake physics (Hill and Prejean, 2007; Brodsky and van der Elst, 2014). In the Coulomb failure model, brittle failure occurs when a dynamic stress perturbation elevates the stress state on a fault plane, exceeding the frictional strength. Therefore, remote earthquakes with larger dynamic stress perturbations should have stronger triggering abilities (Gomberg and Davis, 1996; Gomberg *et al.*, 2001; Wu *et al.*, 2011; Aiken and Peng, 2014; Wang *et al.*, 2015; Miyazawa, 2016). However, a large number of observations have demonstrated that other characteristics of seismic wavefields, such as the back azimuth, dominant frequency, and types of surface and body waves, also affect the dynamic triggering response (e.g., Prejean *et al.*, 2004; Brodsky and Prejean, 2005; West *et al.*, 2005; van der Elst and Brodsky, 2010; Hill, 2012; De Barros *et al.*, 2017). Moreover, the source attribute, such as rupture direction, can influence the wavefield and hence the triggering intensity, although this effect is still debated (e.g., Gomberg *et al.*, 2001; Jiang *et al.*, 2010). At present, it remains difficult to determine a unified mechanism of dynamic triggering or to exactly describe the dominant mechanism in different regions (e.g., Gomberg *et al.*, 1997; Cocco and Rice, 2002; Brodsky *et al.*, 2003; Perfettini *et al.*, 2003; Brodsky and Prejean, 2005; Johnson and Jia, 2005; Gonzalez-Huizar and Velasco, 2011; Shelly *et al.*, 2011;

1. School of Earth and Space Sciences, Peking University, Beijing, China; 2. Earth System Science Programme, Faculty of Science, The Chinese University of Hong Kong, Shatin, Hong Kong, China

*Corresponding author: zsy@pku.edu.cn

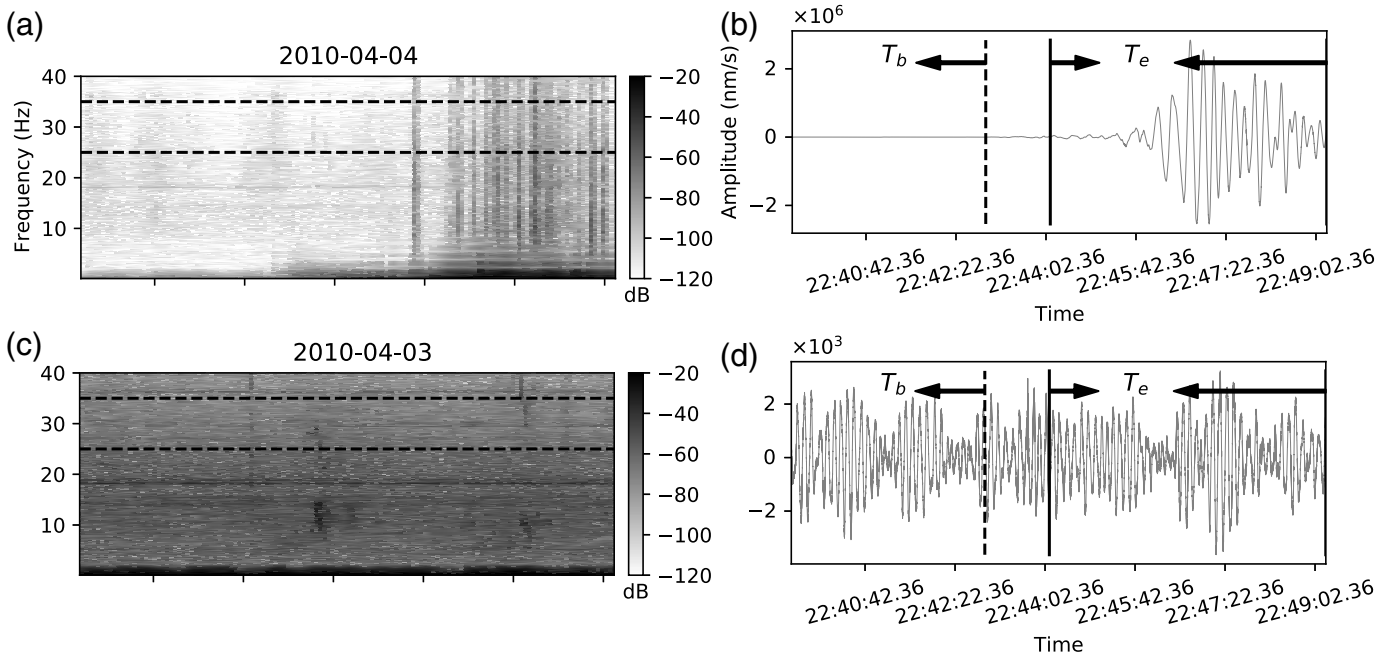

Figure 1. Detection of dynamic triggering for the 4 April 2010 *M* 7.2 Baja California earthquake in the Geysers geothermal area in California. (a,b) Spectrogram and broadband seismogram on 4 April 2010 at station GDXB in the Geysers area. The waveforms are downloaded from the Northern California Earthquake Data Center (NCEDC, 2014; see Data and Resources). Horizontal dashed lines in (a) mark the boundaries of the high-frequency range used to detect local seismicity. Dashed and solid lines in (b) show the arrival times of the *P* wave and a surface wave with a speed of 5 km/s for the Baja California earthquake. T_b represents the 5 hr time window before the *P*-wave arrival, and T_e represents the time window between the arrival times of surface waves with speeds of 5 and 2 km/s. (c,d) Same as (a,b) but on 3 April. T_b and T_e are the same time windows as those on 4 April.

Hill, 2012; Delorey *et al.*, 2015). To address these questions, many investigations of dynamic triggering utilizing rapidly expanding seismic arrays are needed in a variety of regions.

Because of their inception (Hill *et al.*, 1993), many methods have been developed to detect dynamic triggering by identifying triggered earthquakes and then estimating the significance of variations in the rate of seismicity, such as the β statistic (e.g., Matthews and Reasenberg, 1988; Reasenberg and Simpson, 1992). Traditional detection algorithms strongly depend on the accuracy and completeness of earthquake catalogs, which correspond to the network coverage (e.g., Hill *et al.*, 1993). Local earthquakes during teleseismic waveforms are often manually picked, but this process is time-consuming and sometimes subjective (e.g., Peng *et al.*, 2010). Automatic methods for identifying microearthquakes have been applied to observe dynamic triggering (e.g., Miyazawa and Mori, 2005; Delorey *et al.*, 2015; Li *et al.*, 2017; Miyazawa, 2019; Tang *et al.*, 2020); among these methods, the matched filter technique is the most popular (Gibbons and Ringdal, 2006; Peng and Zhao, 2009; Yang *et al.*, 2009), for which an even distribution of template events in the research area is critically important. However, sufficient templates are difficult to obtain in low-seismicity areas, and a large number of templates demand extensive computational times and resources. The convolutional neural network (CNN) has also been widely employed for phase picking and earthquake detection (Perol *et al.*, 2018; Kong *et al.*, 2019; Zhou *et al.*, 2019). The basic idea of the CNN is to optimize the parameters of a neural network model based on a training dataset of earthquakes and then identify events in other waveforms using the trained model. The current development of CNNs is limited by the requirement for a substantial amount of training data, and normally, the network model needs to be retrained when changing the study region (Zhu *et al.*, 2019).

To overcome the dependence on the number of local earthquakes in seismicity rate evaluations, Yun *et al.* (2019) proposed the high-frequency power integral ratio (HiFi) algorithm using the difference in high-frequency energy before and during teleseismic waves (R_E) to detect anomalous seismicity. For example, many studies reported the dynamic triggering of small earthquakes by the 4 April 2010 *M* 7.2 Baja California earthquake in the Geysers geothermal area, California (Aiken and Peng, 2014; Yun *et al.*, 2019). If 5 hr before the *P*-wave arrival is selected as the background time window T_b, the manual detection of local events can become time-consuming. As an alternative, the HiFi method calculates the high-frequency energy directly. The high-frequency (25–35 Hz) energy in the time window T_e aligned with teleseismic waves is richer than that in the time window T_b (Fig. 1a,b). Moreover, to reduce interference due to noise and address disturbances related to background seismicity, high-frequency energy changes among the same time windows on other days (R_B) are also evaluated. For example, the high-frequency energy

TABLE 1
Comparison between Seismic Analysis Code (SAC) and Mini Standard for Exchange of Earthquake Data (miniSEED) Files Recording the Same Continuous Waveform Data at Station ADO with a Length of 24 hr and a Sampling Rate of 40 Hz

Data Format	File Size (MB)	Load Time (s)
SAC	13.8	0.063
miniSEED	7.4	0.045

in the two time windows T_b and T_e on the day before the Baja California earthquake is almost the same (Fig. 1c,d). Finally, a unique confidence level (CL) of detection (0–1) is obtained from the distribution of numerous R_B values on different background days. HiFi was demonstrated to be effective in the Geysers region after a dynamic triggering analysis for hundreds of remote earthquakes (Yun *et al.*, 2019).

Compared with traditional methods, HiFi needs to process only continuous waveforms, a convenient way to comprehensively monitor dynamic triggering in various regions for many stations. However, the original script was made for demonstration using one station in a particular region. We need to implement this algorithm in a generic usage scenario for it to be suitable for drastically expanding seismic arrays. In addition, we initially worked with Seismic Analysis Code (SAC) files, but the mini Standard for Exchange of Earthquake Data (miniSEED) format is commonly used for the storage and transmission of big data. Taking a continuous seismogram covering one day as an example, a miniSEED file requires half the storage and has a faster loading speed than a SAC file (Table 1). Hence, we expect HiFi to handle more compatible data formats.

In this article, we introduce a robust Python package (DynTriPy) to detect dynamic triggering based on the HiFi method. This package is designed for processing data at different stations simultaneously with SAC, miniSEED, and SEED data formats. All calculation functions possess a high-level parallel architecture. To tune parameters conveniently, a database containing high-frequency power integrals is generated first. Moreover, the results of each function are stored for continuous operation in the event of an unexpected interruption. The DynTriPy package is capable to embed in to data centers for real-time monitoring and investigating the sudden activation of any signal within a certain frequency range, such as a triggered tremor. We will first show the details of how we implement the HiFi algorithm and then discuss the computational efficiency and prospects of the DynTriPy package.

Implementation

Python, one of the most popular open-source programming languages, has a global community with millions of developers. In addition, a growing number of mature Python packages can be used directly for scientific computing and geophysics. Therefore, we develop a generic package completely written in Python, termed DynTriPy, to identify the triggering of remote earthquakes utilizing continuous waveforms (Fig. 2).

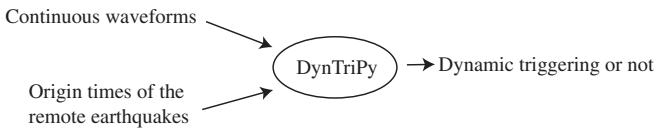

Figure 2. Basic flow of the DynTriPy package.

Four classes of data files are required to run DynTriPy: a catalog of remote earthquakes, continuous waveforms, a list of seismic stations, and instrument response files (Fig. 3). The paths of data sources and parameters make up the input file in JSON format. The users need only to specify the values in a template (Fig. S1, available in the supplemental material to this article), a convenient way to manage the input file. For instance, if we apply DynTriPy to data from the Southern California Seismic Network (CI), we would need to configure the paths in the "data_source" part of the input file. As DynTriPy is designed for the network-based detection of triggering within a large dataset of remote earthquakes, both a list of stations ("station_file," Fig. 3b) and a catalog encompassing the origin times of remote earthquakes ("remote_earthquake_catalog," Fig. 3e) are needed. The time windows T_b and T_e and high-frequency range f_l–f_h are also attributed in the catalog, which can be customized for different events. For example, we define T_b as 5 hr before the *P*-wave arrival and the start and end times of T_e as the arrival times of waves with speeds of 5 and 2 km/s, respectively. The high-frequency range is 5–15 Hz. In addition, records of daily continuous waveforms ("waveform_path," Fig. 3c) and a summary file of instrument responses ("response_file," Fig. 3d) should be prepared because instruments can be upgraded when processing long-term data (e.g., several decades). Hence, we match the daily waveforms and corresponding responses in the summary file.

In DynTriPy, we design a class named *Triggering* with three functions while implementing HiFi to acquire the final CL values to evaluate whether triggering has occurred (Fig. 3a). An instance of the *Triggering* class should be defined first by passing the input file ("input.json," Fig. S1) as the argument, for example "tri=Triggering(input.json)" (Fig. S2). To avoid repeated passing of the same parameters among different functions, we assign all parameters the initial attributes of the instance shared by all functions. By operating the functions of "tri" one by one, we generate the power integral database, logarithm ratios, and CL values. The final and intermediate results of each function are saved to enable work to continue after an unexpected interruption. Next, we will introduce the data used, the details regarding the operation, and the output files of these functions.

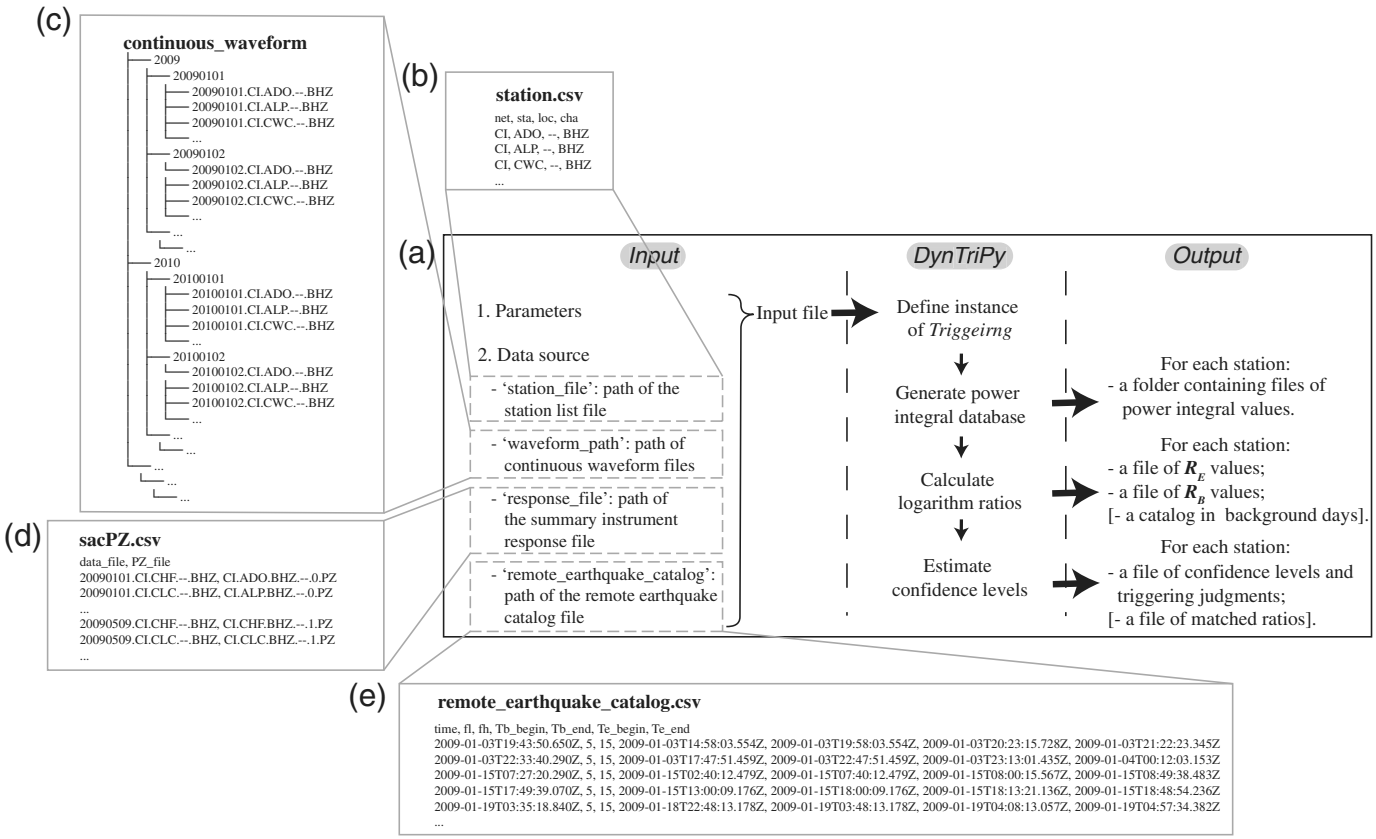

Figure 3. Schematic diagram representing the usage of the DynTriPy package. (a) Summary of the input, workflow of DynTriPy, and output. Strings in single quotation marks are the labels in the "data_source" part of the input file (Fig. S1). (b) Example of the station list file, named "station.csv." The station list contains four columns separated by commas: "net" (network code), "sta" (station code), "loc" (location identifier), and "cha" (channel code). (c) Directory tree structure of an example continuous waveform folder, named "continuous_waveform." Daily data files are stored in date folders, which are archived into folders by year. (d) Example of the summary file of instrument response files, named "sacPZ.csv." Each line of the summary file includes two strings separated by commas: "data_file" (file name of the continuous waveform) and the corresponding "PZ_file" (path of the instrument response). (e) Example of the remote earthquake catalog file named "remote_earthquake_catalog.csv," consisting of "time" (origin time), "fl" (value of f_l), "fh" (value of f_h), "Tb_begin" (begin time of T_b), "Tb_end" (end time of T_b), "Te_begin" (begin time of T_e), and "Te_end" (end time of T_e). PZ, poles and zeroes.

Generating the power integral database: *net_database*

The most time-consuming step in HiFi is computing the power integral ratios in f_l–f_h that correspond to the windows T_b and T_e of a remote earthquake for N background days. The choice of parameters [T_b, T_e, f_l, f_h, N] is necessary and may vary in a considerable way for different regions to ensure reliable detection results (Yun *et al.*, 2019). Therefore, for the efficient implementation of HiFi, redundant calculations during the parameter adjustment phase should be minimized. The first step of HiFi is to select the continuous waveform data in T_b and T_e and then calculate the power spectral density (PSD), recorded as PSD_b and PSD_e, respectively. For the application of HiFi in the Geysers region (Yun *et al.*, 2019), we set T_b to 5 hr before the *P*-wave arrival, which is time-consuming if local events are manually picked. Moreover, our previous study focused on instantaneous triggering, and thus, T_e was set to cover the teleseismic wavetrain (Fig. 1). However, triggering has been reported as delayed by a few hours or even days after the arrival of teleseismic waves (e.g., Brodsky, 2006; Peng *et al.*, 2015; Johnson and Bürgmann, 2016). Therefore, to detect noninstantaneous triggering, the value of T_e may span a large range. Furthermore, PSD_b and PSD_e are integrated over the high-frequency range f_l–f_h to obtain I_b and I_e, respectively, and R_E is calculated by

$$R = \log_{10}\left(\frac{I_e}{I_b}\right). \quad (1)$$

The values of f_l and f_h depend on the local environment and may vary significantly from region to region. Even within the same region, the background noise level in such a frequency range may change over time. In addition, we need to calculate the R_B values by selecting N days before and after the date of the remote earthquake. However, if we implement the earlier operations for each new set of parameters [T_b, T_e, f_l, f_h, N], a vast

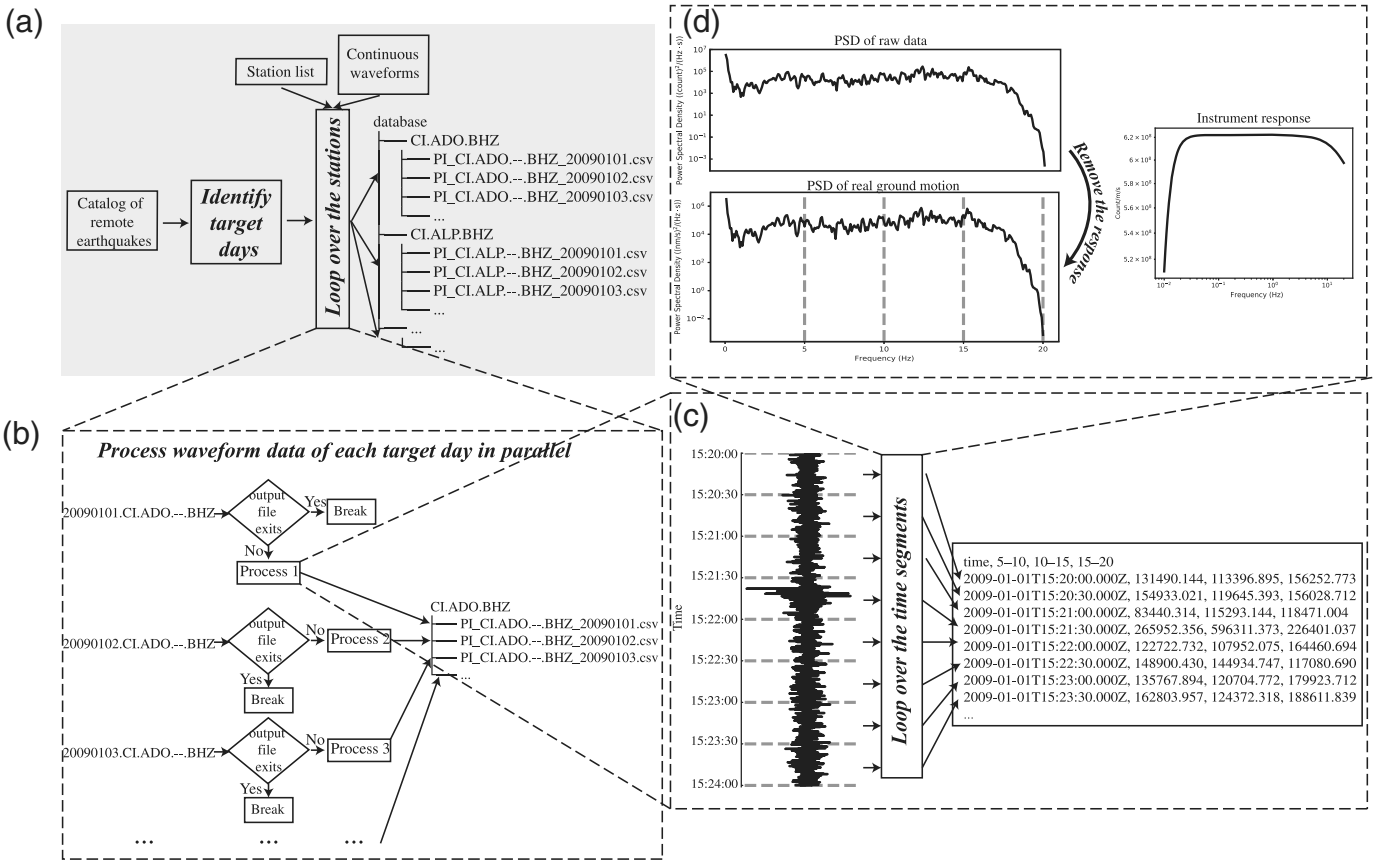

Figure 4. Workflow of the *net_database* function. (a) Basic framework of *net_database*, where "database" is the value of "database_path" set in the parameter file shown in Figure 2. (b) Specific operation for each station, such as station ADO. (c) Data processing for 1 January 2009. The raw data are split by the dashed lines into multiple 30 s segments. We obtain the power integral values in frequency ranges of 5–10, 10–15, and 15–20 Hz for each segment through (d). (d) Power integral calculation for the raw data of the 15:21:30–15:22:00 segment. Dashed lines mark the integral frequency ranges.

number of calculations are repeated. For instance, when enlarging the T_b window from 3 to 5 hr, the power integral within the first 3 hr before the teleseismic arrival will be calculated again. Therefore, we integrate the PSD in different time and frequency segments for continuous seismograms and store the results in a database. For different parameter values, we simply fetch the target power integrals from the database to calculate the R_E and R_B values.

The power integral database is generated by the *net_database* function with a high-level parallel architecture, for example, "tri.net_database(p=5)" (Fig. S2), in which the argument "p" is the number of processes. First, we record a list containing the target dates, namely, M_1 days before and M_2 days after the dates of distant earthquakes according to the catalog. The power integrals of these days must be computed (Fig. 4a). The scope $[M_1, M_2]$ can be modified by the users in the input file ("days," Fig. S1). If the scopes of different earthquakes overlap, the same day is recorded only once. Then, we loop over each station and calculate the power integrals from the continuous waveforms. The results are stored into different subfolders in the output path configured in the input file under "database_path." For each station, the computations for different days are completely independent. Therefore, we parallelize these computations by assigning the tasks of different days to different processors and saving the results into individual files (Fig. 4b). The parallel framework is constructed by Python's built-in multiprocessing package (also adopted for the other parallel computing in our algorithm). Separating the results of different days can avoid excessively large files and prevent conflicts caused by multiple processes writing to the same file. In addition, to reduce unnecessary computing, we check whether the results of that day already exist before assigning the job.

To introduce the details of each process, we show an example of the data at station ADO in the CI network on 1 January 2009, downloaded from the Southern California Earthquake Data Center (SCEDC, 2013; see Data and Resources). The waveforms are loaded by ObsPy (Beyreuther *et al.*, 2010), a commonly used Python package for analyzing seismological data in SAC, miniSEED, SEED and other formats. We remove the mean and trend and split the waveforms evenly into time segments with a length of 30 s (Fig. 4c). The time segment length can be adjusted in the input file ("time_segment").

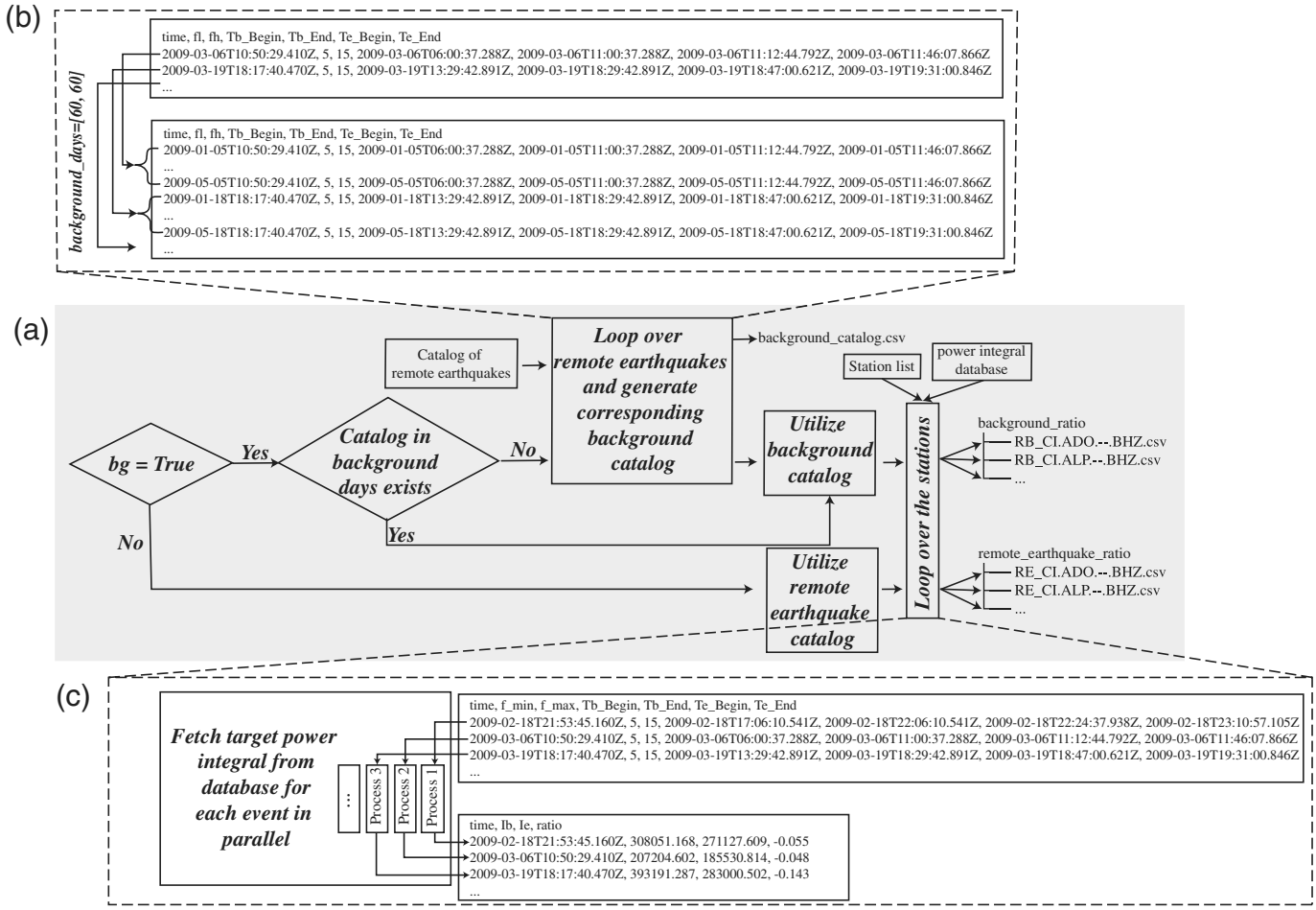

Figure 5. Workflow of the *net_ratio* function. (a) Basic framework of *net_ratio*, where "remote_earthquake_ratio" and "background_ratio" are the values of "RE_path" and "RB_path," respectively, set in the input file (Fig. S1). (b) Correspondence between remote earthquakes and events on background days. (c) Computing the logarithm ratio of each remote earthquake (R_E) at station ADO. The output file consists of four columns: the origin time, which is the same as that in the catalog ("time"), I_b ("Ib"), I_e ("Ie"), and the logarithm ratio between I_b and I_e ("ratio").

We estimate the PSD of each segment using Welch's method by dividing the data into overlapping intervals and averaging the spectrograms of these intervals (Welch, 1967; Fig. 4d). The reciprocal of the time length of the interval determines the frequency resolution of the PSD. We fix the interval as 512 samples in the *net_pi* function, corresponding to a frequency resolution of ~0.1 Hz for sampling rates of ~10–100 Hz. Because the PSDs calculated from raw waveforms and real ground motions may be different, especially when processing long-term data that might be recorded by different instruments, we prefer to remove instrument responses. In our code, the data are already in the frequency domain after obtaining the PSD, so we directly divide the PSD by the square of the response function to acquire the ground-motion PSD. According to the summary response file, the response function is built by loading the values of "sensitivity," "A0," "zeros," and "poles" in the poles and zeroes (PZ) file matched with the waveform data. Next, we integrate the PSD within frequency segments of 5–10, 10–15, and 15–20 Hz, and the results are written into the output file with the start time of the time segment (Fig. 4c). The minimum (*min*), segment length (*step*), and maximum (*max*) of the frequency segments are configured by the users in the input file as the value of "frequency_segment" in the format of [*min*, *step*, *max*].

Calculating logarithm ratios: *net_ratio*

After generating the power integral database, we implement the *net_ratio* function to calculate R_E and R_B values in parallel. The process count "p" can also be set by the user.

It is straightforward to calculate R_E from equation (1) based on a real remote earthquake catalog (Fig. 5c). We provide a utility named *gen_time_windows* to generate T_b and T_e and automatically output a catalog (e.g., Fig. 3e) with the origin time and the start and end times of T_b and T_e (Fig. S3). T_b is a fixed time range before the *P*-wave arrival, and T_e is bounded between two surface waves with specific speeds. The length of T_b, the surface wavespeeds, the raw catalog of remote earthquakes with source locations, a reference point

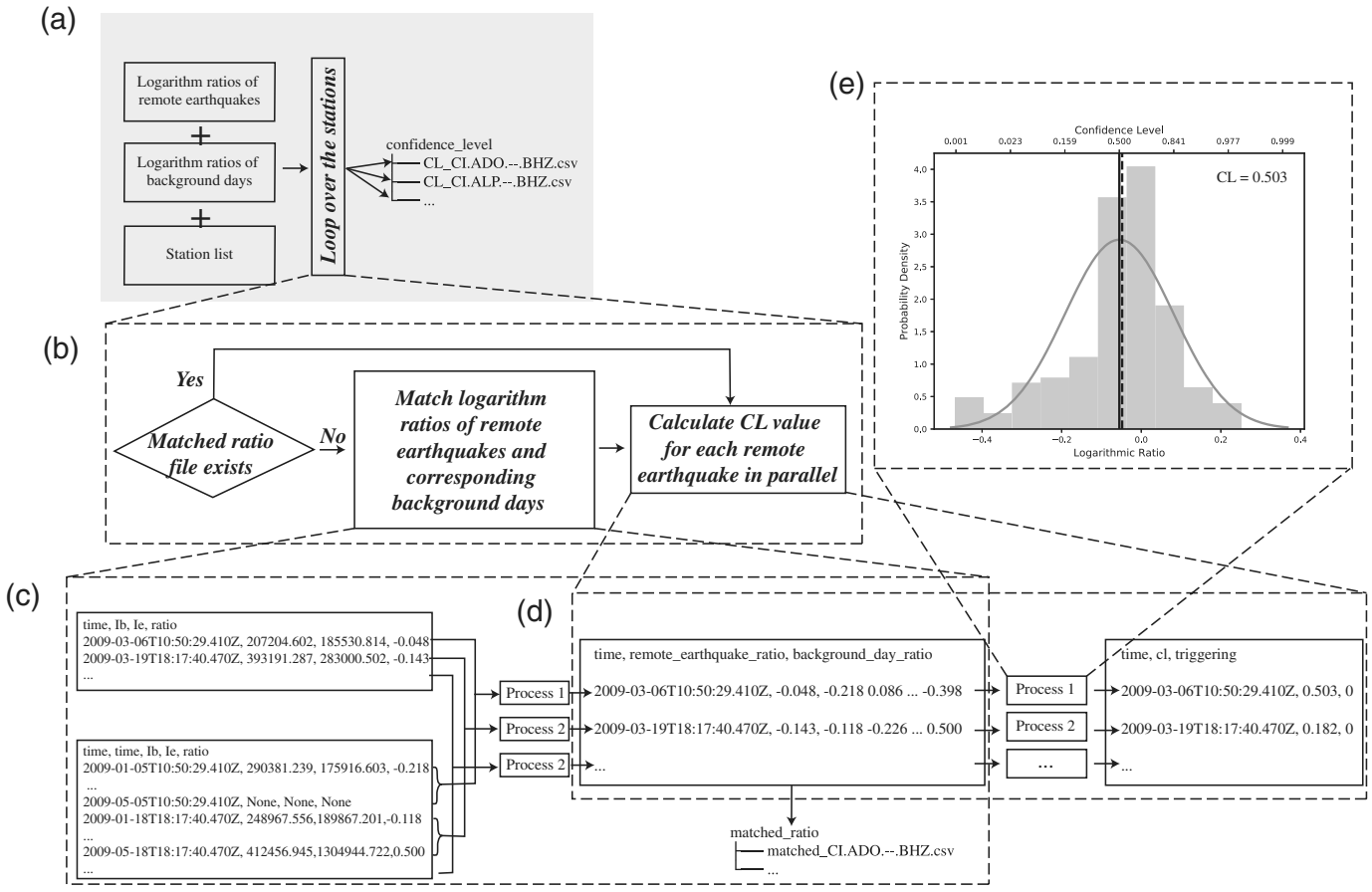

Figure 6. Workflow of the net_cl function. (a) Basic framework of *net_cl*, where "confidence_level" is the value of "cl_path" set in the input file (Fig. S1). (b) Specific processing for each station to obtain the confidence level (CL). (c) Matching of R_E and the logarithm ratios on background days (R_B) at station ADO. In the output file, the columns of "remote_earthquake_ratio" and "background_day_ratio" represent R_E and R_B, respectively. (d) Acquisition of CL values and triggering judgments using the matched logarithm ratios. (e) CL estimation for the 2009-03-06T10:50:29.410Z earthquake. The histogram shows the probability density of R_B values. Fitting the histogram to a normal distribution, we obtain the probability density function (PDF; gray solid line). Black solid and dashed lines mark the mean R_B and R_E values, respectively.

in the research area, and the output file are required as parameters passed to the *gen_time_windows* function.

However, for R_B, we first need to build a virtual earthquake catalog (Fig. 5a). For the 2009-03-06T10:50:29.410 event and the parameter "background_days" in the input file (Fig. S1) assigned the range [60, 60], we generate 60 virtual events both before and after the date of the remote earthquake (Fig. 5b). The background events have the same attributes, such as the origin time, start and end times, and f_l and f_h of the remote event, except for the date. We save the virtual catalog as the "background_catalog" path. We then compute R_B using equation (1) on each background day. If the background catalog file exists, in the event of an accidental interruption, we do not generate it again when rerunning *net_ratio*. Evaluating R_E or R_B depends on whether the argument "bg" is "False" or "True," such as "tri.net_ratio(bg=False, p=5)" to compute R_E (Fig. S2).

Then, we estimate R_E or R_B according to the catalog of remote earthquakes or background days at each station, which are stored into separate files named with the station name (Fig. 5a). The output path is established in the input file, termed "RE_path" or "RB_path." For example, when calculating R_E at station ADO based on equation (1), two power integral groups in f_l–f_h of T_b and T_e are fetched from the database and averaged as I_b and I_e, respectively (Fig. 5c). To speed up this fetching process, we load the target data files in the database as Pandas DataFrame objects (McKinney, 2010; The Pandas Development Team, 2020), which is a 2D size-mutable data structure with labeled axes. The Pandas Python package can analyze large datasets by importing and filtering the data conveniently and efficiently. We filter the row and column names of the DataFrame corresponding to the time and frequency segments, respectively, to acquire the target integral values. The earlier operations for different events are performed in parallel.

Estimating the CL of dynamic triggering: *net_cl*

Finally, we analyze the distribution of R_B and compare it with R_E to obtain the CL of dynamic triggering using the *net_cl*

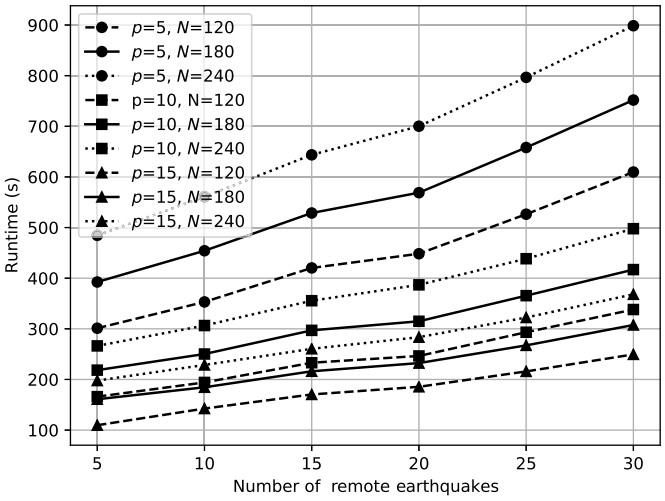

Figure 7. Computational efficiency of the DynTriPy package. Using different numbers of processors (p) to perform the computation in parallel, we test the runtimes of the triggering detection process for different numbers of remote earthquakes and N values.

function, for example, "tri.net_cl(p=5)" (Fig. S2). The CL values for different distant earthquakes at the same station are computed in parallel on "p" different processors and saved as separate files (Fig. 6a).

For each station, we associate the R_E and R_B values, which are stored separately at "RE_path" and "RB_path" (Fig. 6b). The files comprising the R_E and R_B values are loaded as Pandas DataFrame objects, whose row names are the origin times of remote earthquakes and background seismicity, respectively. For each remote earthquake in the DataFrame of R_E, we fetch R_B in parallel on the target background days by filtering the rows of the R_B DataFrame, for example, the 2009-03-06T10:50:29.410Z and 2009-03-19T18:17:40.470Z events at station ADO (Fig. 6c). The associated ratios at different stations are output into an individual subfile in the path "matched_ratio_path" (Fig. S1). The existence of a file is checked before association to avoid duplicate calculations (Fig. 6b). Then, the matched ratios are used to calculate the CL values in parallel (Fig. 6d). We fit the R_B values to a normal distribution, acquiring the probability density function (PDF) at the same time. CL is the integral of the PDF from minus infinity to R_E. For the 2009-03-06T10:50:29.410Z event, the CL result is approximately 0.5 because the R_E value is close to the mean of R_B (Fig. 6e). If we wish to plot and output the figure as in Figure 6e for each remote earthquake, we simply set "figure_path" (Fig. S1) in the input file to store the figures; otherwise, the path is configured as "None." Finally, according to the "threshold" of the CL value set in the input file, we mark the triggering result in the final output file with "1" or "0" corresponding to whether the CL value is greater or less than the threshold, respectively (Fig. 6d).

Discussion

Computational efficiency

We conduct some experiments to demonstrate the performance of our DynTriPy package, especially the improvement in the computational efficiency by utilizing the parallel architecture. The test is performed at station ADO in southern California considering the remote earthquakes that occurred from May 2009 to January 2010. The catalog of remote earthquakes is acquired from the Advanced National Seismic System (U.S. Geological Survey, Earthquake Hazards Program, 2017; see Data and Resources) with magnitudes ≥ 6.5, distances ≥ 1000 km, and depths ≤ 100 km. We download all the continuous BHZ-channel waveform data from 2009 and 2010 for station ADO with a sampling rate of 40 Hz, and we acquire the corresponding PZ instrument response file from SCEDC. The experiments are performed on a workstation with two CPUs (IntelR XeonR E5-2695 v3 at 2.30 GHz) possessing 28 cores in total.

We configure the parameter file (Fig. S1), utilize five processes to generate the power integral database, and activate other functions to obtain the CLs for five remote earthquakes with 120 background days (Fig. S2). To just meet the needs for calculating R_B, we set the "days" of the database to be the same as the "background_days." The average runtime of the whole triggering detection process is approximately 5 min (Fig. 7). As the numbers of remote earthquakes and background days increase, the computing time rises nearly linearly. Processing 30 remote earthquakes with 240 background days requires only approximately 15 min.

In addition, we test the runtime when more processes are used, such as 10 or 15. For 30 distant earthquakes with 120 background days, the computing time is reduced by ∼1/2 (from ∼600 to ∼340 s) as the number of processes increases from 5 to 10. Limited by the input/output (I/O) speed, the computational efficiency improvement slows down as the number of processes increases to 15. However, the computing time is still ∼3/4 of that using 10 processes (from ∼340 to ∼250 s). The parallel architecture significantly improves the computational efficiency.

The runtime with more data can be estimated according to the previous computational efficiency test. If dynamic triggering is analyzed over 10 yr at station ADO with ∼400 remote earthquakes (magnitudes ≥ 6.5), it will take approximately $250 \times 400/30$ s ≈ 0.9 hr when utilizing 15 processes and 120 background days. Among the output files, the power integral database occupies most of the storage space, that is, ∼830 MB with a time interval of 30 s for 10 yr at one station. The time consumption and the output data size increase linearly with the number n of input stations, that is, with complexity $O(n)$.

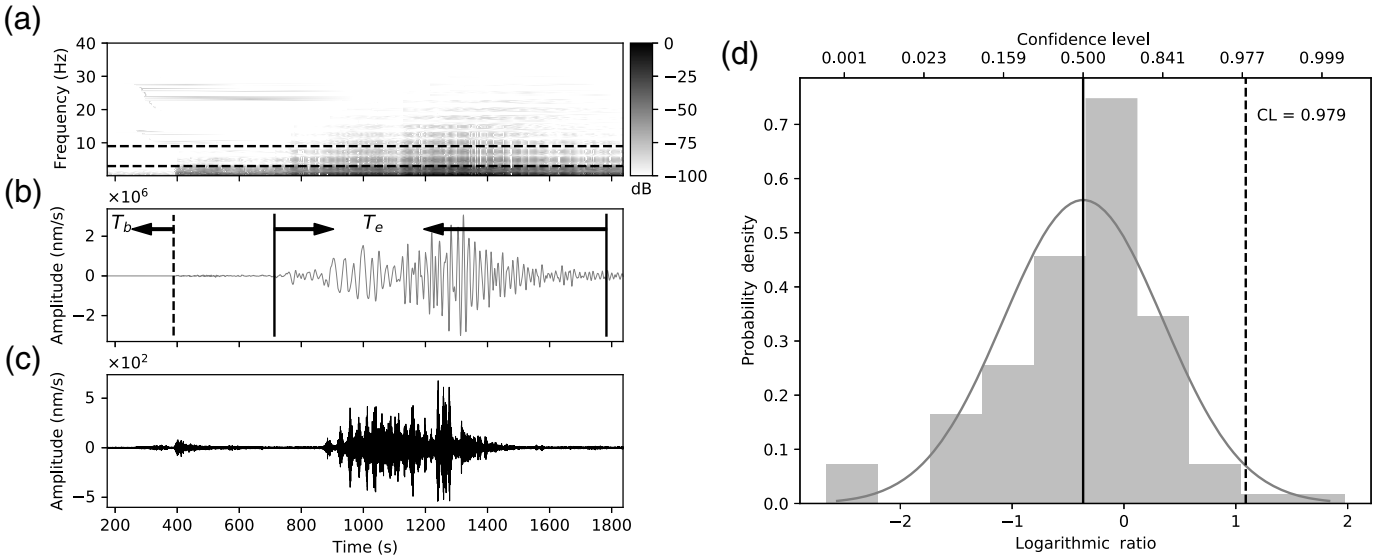

Figure 8. Detection of a triggered tremor for the 3 November 2002 M_w 7.8 Denali fault earthquake near Parkfield, California. (a–c) Spectrogram, raw seismogram, and band-pass-filtered seismogram of the 2002 Denali fault earthquake from station PKD, channel HHN, downloaded from NCEDC. The symbols in (a, b) are the same as those in Figure 1a,b. (d) Same as Figure 6e but for the 2002 Denali fault earthquake. The continuous waveforms used to generate the power integral database are from station PKD, channel BHN, accessed from NCEDC.

Application prospects

Our DynTriPy package allows the detection of dynamic triggering for a large dataset of remote earthquakes based on seismic arrays and enables the long-term investigation and large-scale monitoring of dynamic triggering on a global scale. Sufficient observations are essential to explore the potential factors affecting triggering phenomena to understand the physical mechanism of dynamic triggering. In addition, by automatically supplementing the power integral database regularly every day, the CL value can be evaluated in a very short time after the occurrence of a remote earthquake, realizing the near-real-time monitoring of triggering. As this code supports the miniSEED format, DynTriPy can directly use continuous waveform databases from large data centers, making it possible to implement real-time processing. Long-term and real-time monitoring is helpful for keeping track of the stress state of a study region and thus assessing the seismic hazards therein. For instance, Van der Elst *et al.* (2013) suggested that dynamic triggering is more evident in areas with increased seismicity induced by industrial activities because fault systems reach a critical state due to fluid injection. Therefore, dynamic triggering could provide early warning for earthquake swarms in regions with induced earthquakes. Utilizing our package to examine more remote events, we can obtain a continuous evaluation of the stress state and thus advance our understanding of the influence of fluid injection.

In addition to dynamically triggered earthquakes, our package can be used to detect triggered nonvolcanic tremors within a characteristic frequency range of ~2–8 Hz (e.g., Shelly *et al.*, 2011; Yang and Peng, 2013; Peng *et al.*, 2015). Previous studies suggested that teleseismic waves from the 3 November 2002 M_w 7.8 Denali fault earthquake triggered significant tremors near Parkfield, California (Gomberg *et al.*, 2008; Peng *et al.*, 2008, 2009). Rich energy within ~2–10 Hz and tremor signals are visible at the same time as surface waves during the Denali fault earthquake at station PKD (Fig. 8a,c, Peng *et al.*, 2008). We apply the DynTriPy package to detect these tremor signals. Still taking T_b and T_e as the time windows 5 hr before the *P*-wave arrival and the time between the arrivals of surface waves with speeds of 5 and 2 km/s, respectively, we calculate the R_E and R_B values within 3–9 Hz (Fig. 8a,b). Considering 120 background days, the R_E value (1.089) is obviously larger than the mean of the R_B distribution (Fig. 8d). The final CL value is 0.979, which is larger than the popular threshold of 0.977 (Yun *et al.*, 2019), indicating that these high-frequency signals were triggered by the Denali fault earthquake. Furthermore, we apply our package to the data from stations PHP and PSM, ~7 and ~14 km from station PKD, respectively (Fig. S4). The mean CL value among these three stations is 0.978, suggesting that the high-frequency signals during teleseismic waves are not noise but tectonic signals. In addition, if the frequency integral range is set as 25–35 Hz, the R_E value at station PKD is smaller than zero, indicating the absence of regularly triggered local earthquakes. Hence, we can determine that the 2002 Denali fault earthquake triggered local tectonic tremors near Parkfield. These applications to the Geysers (Yun *et al.*, 2019) and Parkfield areas demonstrate that our DynTriPy package can effectively identify and distinguish dynamic triggered earthquakes and tremors.

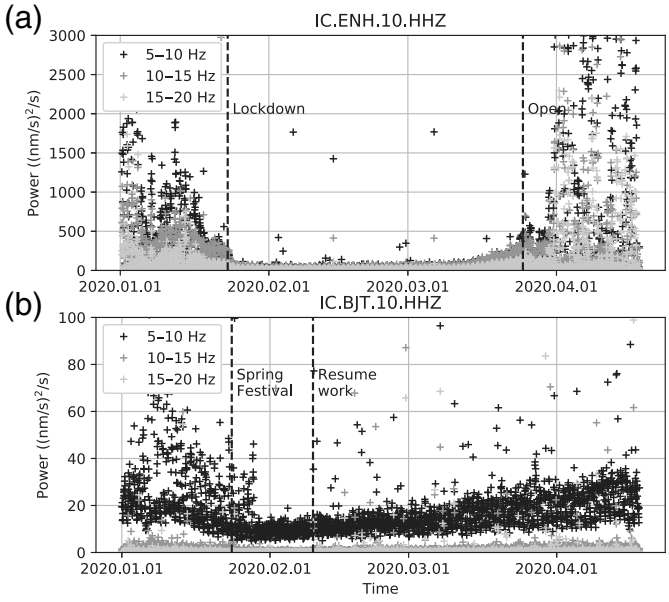

Figure 9. Background noise of seismic data in Hubei and Beijing because of the advent of Coronavirus Disease 2019 (COVID-19). (a) Power integral values at station ENH in Enshi, Hubei, for different frequency ranges. Dashed lines mark the times of the lockdown and reopening of Enshi. (b) Same as (a) but at station BJT in Beijing. The start time of the Spring Festival and resumption of work are denoted by the dashed lines.

Even if the impact of dynamic stress is not considered, the variation in activity within a certain frequency segment during some diurnal period can also be discovered through a comparison with the same period on other days. This general application can be widely used to observe the temporal variations in specific signals of different frequency ranges, including man-made noise. For example, we employ the DynTriPy package to process the continuous waveforms from 1 January to 17 April 2020 recorded at stations in Hubei (ENH) and Beijing (BJT), China, to investigate the changes in noise due to the city lockdowns and holiday extension due to the Coronavirus Disease-2019 pandemic. The output results of the *net_database* function clearly show that the noise levels in the frequency ranges of 5–10, 10–15, and 15–20 Hz were all drastically reduced in the city of Enshi in Hubei, China (Fig. 9a), but the noise levels significantly increased after the lockdown was over. The main frequency range of noise energy at station BJT is 5–15 Hz, probably corresponding to one specific type of human activity.

Conclusion

We designed a Python package named DynTriPy based on the HiFi algorithm to detect dynamic triggering by distinguishing abnormal seismicity. The HiFi method is suitable for processing big data independent of earthquake catalogs or subjective judgments and can suppress the influence of noise and variations in the background seismicity. Finally, a CL of dynamic triggering (0–1) is statistically yielded.

The parameters and data sources (the station name, continuous waveform records, instrument response, and catalog of remote earthquakes) are collected in one input file, which is convenient for users to adjust and manage. Our package consists of one *Triggering* class with three functions. A power integral database of continuous waveforms is generated first utilizing the *net_database* function. Furthermore, the logarithm ratios of high-frequency energy (R_E) for remote earthquakes and background days (R_B) are calculated by the *net_ratio* function. Finally, R_E and the corresponding R_B values are matched to evaluate the CLs using the *net_cl* function.

The DynTriPy package supports the simultaneous processing of data from multiple stations and in different formats, such as SAC, miniSEED, and SEED. Generating a database to store power integrals in different time and frequency segments makes the tuning of parameters more convenient. In addition, repeated computations for different time windows, frequency ranges, and background days are eliminated. The results of each function are output and reserved to enable continuous operation in the event of an unexpected interruption. To take advantage of multiple computational cores, we construct a high-level parallel architecture for all computing functions. When running 15 processes, calculating the CLs for 30 remote earthquakes takes only ∼5 min. In addition to dynamically triggered earthquakes, our package can be used to detect triggered tremors or variations in other signals with characteristic frequency ranges. The intermediate results of generating a power integral database can also be used to monitor the temporal variations in noise energy. We expect to deploy DynTriPy in data centers for the real-time monitoring of dynamic triggering.

Data and Resources

The continuous waveform data and catalog of remote earthquakes are obtained from the Southern California Earthquake Data Center (doi: 10.7909/C3WD3xH1), Northern California Earthquake Data Center (doi: 10.7932/NCEDC), and Advanced National Seismic System of the U.S. Geological Survey (doi: 10.5066/F7MS3QZH). Our DynTriPy package is available at https://github.com/yunndlalala/dynamic_earthquake_triggering (last accessed September 2020). Supplemental material for this article includes Figures S1–S4.

Acknowledgments

The authors thank Jian Piao and Jinping Zi for their constructive feedback after testing the initial version of our package. This work is supported by the National Natural Science Foundation of China (Grant 42074046), the National Key R&D Program of China (Grant 2018YFC1503400), the China Earthquake Science Experiment Project, China Earthquake Administration (CEA) (Grants 2017CESE0103, 2018CSES0102, and 2019CSES0107), the Hong Kong Special Administrative Region (HKSAR) Research Grant Council General Research Fund (GRF) (Grant 14305617), and a

Chinese University of Hong Kong (CUHK) Direct Grant from the Faculty of Science, State Key Lab of Earthquake Dynamics (Grant LED2017B07), Institute of Geology, CEA. The authors are grateful to the anonymous reviewers and Editor-in-Chief Allison Bent and Annastasia Pratt for their valuable thoughts and suggestions.

References

Aiken, C., and Z. Peng (2014). Dynamic triggering of microearthquakes in three geothermal/volcanic regions of California, *J. Geophys. Res.* **119,** 6992–7009, doi: 10.1002/2014JB011218.

Aiken, C., X. Meng, and J. Hardebeck (2018). Testing for the 'predictability' of dynamically triggered earthquakes in The Geysers geothermal field, *Earth Planet. Sci. Lett.* **486,** 129–140, doi: 10.1016/j.epsl.2018.01.015.

Beyreuther, M., R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann (2010). ObsPy: A python toolbox for seismology, *Seismol. Res. Lett.* **81,** no. 3, 530–533, doi: 10.1785/gssrl.81.3.530.

Brodsky, E. E. (2006). Long-range triggered earthquakes that continue after the wave train passes, *Geophys. Res. Lett.* **33,** no. 15, 1–5, doi: 10.1029/2006GL026605.

Brodsky, E. E., and S. G. Prejean (2005). New constraints on mechanisms of remotely triggered seismicity at Long Valley Caldera, *J. Geophys. Res.* **110,** no. 4, 1–14, doi: 10.1029/2004JB003211.

Brodsky, E. E., and N. J. van der Elst (2014). The uses of dynamic earthquake triggering, *Annu. Rev. Earth Planet. Sci.* **42,** no. 1, 317–339, doi: 10.1146/annurev-earth-060313-054648.

Brodsky, E. E., E. Roeloffs, D. Woodcock, I. Gall, and M. Manga (2003). A mechanism for sustained groundwater pressure changes induced by distant earthquakes, *J. Geophys. Res.* **108,** no. B8, 1–10, doi: 10.1029/2002jb002321.

Cocco, M., and J. R. Rice (2002). Pore pressure and poroelasticity effects in Coulomb stress analysis of earthquake interactions, *J. Geophys. Res.* **107,** no. B2, ESE 2-1–ESE 2-17, doi: 10.1029/2000jb000138.

De Barros, L., A. Deschamps, A. Sladen, H. Lyon-Caen, and N. Voulgaris (2017). Investigating dynamic triggering of seismicity by regional earthquakes: The case of the Corinth Rift (Greece), *Geophys. Res. Lett.* **44,** no. 21, 10,921–10,929, doi: 10.1002/2017GL075460.

Delorey, A. A., K. Chao, K. Obara, and P. A. Johnson (2015). Cascading elastic perturbation in Japan due to the 2012 M_w 8.6 Indian Ocean earthquake, *Sci. Adv.* **1,** no. 9, 2–8, doi: 10.1126/sciadv.1500468.

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.

Gomberg, J., and S. Davis (1996). Stress/strain changes and triggered seismicity at The Geysers, California, *J. Geophys. Res.* **101,** no. B1, 733–749, doi: 10.1029/95JB03250.

Gomberg, J., and P. A. Johnson (2005). Dynamic triggering of earthquakes, *Nature* **437,** no. 7060, 830, doi: 10.1038/437830a.

Gomberg, J., M. L. Blanpied, and N. M. Beeler (1997). Transient triggering of near and distant earthquakes, *Bull. Seismol. Soc. Am.* **87,** no. 2, 294–309.

Gomberg, J., P. A. Reasenberg, P. Bodin, and R. A. Harris (2001). Earthquake triggering by seismic waves following the Landers and Hector Mine earthquakes, *Nature* **411,** no. 6836, 462–466, doi: 10.1038/35078053.

Gomberg, J., J. L. Rubinstein, Z. Peng, K. C. Creager, J. E. Vidale, and P. Bodin (2008). Widespread triggering of nonvolcanic tremor in California, *Science* **319,** no. 5860, 173, doi: 10.1126/science.1149164.

Gonzalez-Huizar, H., and A. A. Velasco (2011). Dynamic triggering: Stress modeling and a case study, *J. Geophys. Res.* **116,** no. 2, 1–13, doi: 10.1029/2009JB007000.

Hill, D. P. (2012). Dynamic stresses, Coulomb failure, and remote triggering-corrected, *Bull. Seismol. Soc. Am.* **102,** no. 6, 2313–2336, doi: 10.1785/0120120085.

Hill, D. P., and S. G. Prejean (2007). Dynamic triggering, in *Treatise on Geophysics*, 257–291, doi: 10.1016/B978-044452748-6.00070-5.

Hill, D. P., P. A. Reasenberg, A. Michael, W. J. Arabaz, G. Beroza, D. Brumbaugh, J. N. Brune, R. Castro, S. Davis, D. dePolo, *et al.* (1993). Seismicity remotely triggered by the magnitude 7.3 Landers, California, earthquake, *Science* **260,** no. 5114, 1617–1623, doi: 10.1126/science.260.5114.1617.

Jiang, T., Z. Peng, W. Wang, and Q. F. Chen (2010). Remotely triggered seismicity in Continental China following the 2008 M_w 7.9 Wenchuan earthquake, *Bull. Seismol. Soc. Am.* **100,** no. 5B, 2574–2589, doi: 10.1785/0120090286.

Johnson, C. W., and R. Bürgmann (2016). Delayed dynamic triggering: Local seismicity leading up to three remote M ≥ 6 aftershocks of the 11 April 2012 M8.6 Indian Ocean earthquake, *J. Geophys. Res.* **121,** 134–151, doi: 10.1002/2015JB012243.

Johnson, P. A., and X. Jia (2005). Nonlinear dynamics, granular media and dynamic earthquake triggering, *Nature* **437,** no. 7060, 871–874, doi: 10.1038/nature04015.

Kong, Q., D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, and P. Gerstoft (2019). Machine learning in seismology: Turning data into insights, *Seismol. Res. Lett.* **90,** no. 1, 3–14, doi: 10.1785/0220180259.

Li, L., D. Yao, X. Meng, Z. Peng, and B. Wang (2017). Increasing seismicity in Southern Tibet following the 2015 Mw 7.8 Gorkha, Nepal earthquake, *Tectonophysics* **714/715,** 62–70, doi: 10.1016/j.tecto.2016.08.008.

Matthews, M. V., and P. A. Reasenberg (1988). Statistical methods for investigating quiescence and other temporal seismicity patterns, *Pure Appl. Geophys.* **126,** nos. 2/4, 357–372, doi: 10.1007/bf00879003.

McKinney, W. (2010). Data structures for statistical computing in Python, in S. van der Walt and J. Millman (Editors), *Proc. of the 9th Python in Science Conf.*, 56–61, doi: 10.25080/Majora-92bf1922-00a.

Miyazawa, M. (2016). An investigation into the remote triggering of the Oita earthquake by the 2016 Mw 7.0 Kumamoto earthquake using full wavefield simulation, *Earth Planets Space* **68,** no. 205, doi: 10.1186/s40623-016-0585-z.

Miyazawa, M. (2019). Bayesian approach for detecting dynamically triggered very low-frequency earthquakes in the Nankai subduction zone and application to the 2016 Mw 5.9 off-Kii Peninsula earthquake, Japan, *Geophys. J. Int.* **217,** no. 2, 1123–1140, doi: 10.1093/gji/ggz073.

Miyazawa, M., and J. Mori (2005). Detection of triggered deep low-frequency events from the 2003 Tokachi-oki earthquake, *Geophys. Res. Lett.* **32,** no. 10, doi: 10.1029/2005GL022539.

Northern California Earthquake Data Center (NCEDC) (2014). Dataset, Northern California Earthquake Data Center, UC Berkeley Seismological Laboratory, California, doi: 10.7932/NCEDC.

Peng, Y., S. Zhou, J. Zhuang, and J. Shi (2012). An approach to detect the abnormal seismicity increase in Southwestern China triggered co-seismically by 2004 Sumatra Mw 9.2 earthquake, *Geophys. J. Int.* **189,** no. 3, 1734–1740, doi: 10.1111/j.1365-246X.2012.05456.x.

Peng, Z., and P. Zhao (2009). Migration of early aftershocks following the 2004 Parkfield earthquake, *Nature Geosci.* **2,** no. 12, 877–881, doi: 10.1038/ngeo697.

Peng, Z., D. R. Shelly, and W. L. Ellsworth (2015). Delayed dynamic triggering of deep tremor along the Parkfield-Cholame section of the San Andreas Fault following the 2014 M6.0 South Napa earthquake, *Geophys. Res. Lett.* **42,** no. 19, 7916–7922, doi: 10.1002/2015GL065277.

Peng, Z., J. E. Vidale, K. C. Creager, J. L. Rubinstein, J. Gomberg, and P. Bodin (2008). Strong tremor near Parkfield, CA, excited by the 2002 Denali fault earthquake, *Geophys. Res. Lett.* **35,** no. 23, 1–5, doi: 10.1029/2008GL036080.

Peng, Z., J. E. Vidale, A. G. Wech, R. M. Nadeau, and K. C. Creager (2009). Remote triggering of tremor along the San Andreas Fault in central California, *J. Geophys. Res.* **114,** doi: 10.1029/2008JB006049.

Peng, Z., W. Wang, Q. F. Chen, and T. Jiang (2010). Remotely triggered seismicity in north China following the 2008 Mw 7.9 Wenchuan earthquake, *Earth Planets Space* **62,** 893–898, doi: 10.5047/eps.2009.03.006.

Perfettini, H., J. Schmittbuhl, and A. Cochard (2003). Shear and normal load perturbations on a two-dimensional continuous fault: 2. Dynamic triggering, *J. Geophys. Res.* **108,** no. B9, 2409, doi: 10.1029/2002JB001805.

Perol, T., M. Gharbi, and M. Denolle (2018). Convolutional neural network for earthquake detection and location, *Sci. Adv.* **4,** no. 2, 2–10, doi: 10.1126/sciadv.1700578.

Pollitz, F. F., R. S. Stein, V. Sevilgen, and R. Bürgmann (2012). The 11 April 2012 east Indian Ocean earthquake triggered large aftershocks worldwide, *Nature* **490,** no. 7419, 250–253, doi: 10.1038/nature11504.

Prejean, S. G., D. P. Hill, E. E. Brodsky, S. E. Hough, M. J. S. Johnston, S. D. Malone, D. H. Oppenheimer, A. M. Pitt, and K. B. Richards-Dinger (2004). Remotely triggered seismicity on the United States west coast following the Mw 7.9 Denali fault earthquake, *Bull. Seismol. Soc. Am.* **94,** no. 6B, S348–S359, doi: 10.1785/0120040610.

Reasenberg, P. A., and R. W. Simpson (1992). Response of regional seismicity to the static stress change produced by the Loma Prieta earthquake, *Science* **255,** no. 5052, 1687–1690, doi: 10.1126/science.255.5052.1687.

Shelly, D. R., Z. Peng, D. P. Hill, and C. Aiken (2011). Triggered creep as a possible mechanism for delayed dynamic triggering of tremor and earthquakes, *Nature Geosci.* **4,** no. 6, 384–388, doi: 10.1038/ngeo1141.

Southern California Earthquake Center (SCEDC) (2013). Southern California Earthquake Center, Caltech, Dataset, doi: 10.7909/C3WD3xH1.

Tang, V., P. Seetharaman, K. Chao, B. A. Pardo, and S. van der Lee (2020). Automating the detection of dynamically triggered earthquakes via a deep metric learning algorithm, *Seismol. Res. Lett.* **91,** doi: 10.1785/0220190165.

The Pandas Development Team (2020). *pandas-dev/pandas: Pandas*, Zenodo, doi: 10.5281/zenodo.3509134.

U.S. Geological Survey, Earthquake Hazards Program (2017). Advanced National Seismic System (ANSS) Comprehensive Catalog of Earthquake Events and Products: Various, doi: 10.5066/F7MS3QZH.

van der Elst, N. J., and E. E. Brodsky (2010). Connecting near-field and far-field earthquake triggering to dynamic strain, *J. Geophys. Res.* **115,** no. B7, doi: 10.1029/2009JB006681.

van der Elst, N. J., H. M. Savage, K. M. Keranen, and G. A. Abers (2013). Enhanced remote earthquake triggering at fluid-injection sites in the midwestern United States, *Science* **341,** no. 6142, 164–167, doi: 10.1126/science.1238948.

Velasco, A. A., S. Hernandez, T. Parsons, and K. Pankow (2008). Global ubiquity of dynamic earthquake triggering, *Nature Geosci.* **1,** no. 6, 375–379, doi: 10.1038/ngeo204.

Wang, B., R. M. Harrington, Y. Liu, H. Yu, A. Carey, and N. J. van der Elst (2015). Isolated cases of remote dynamic triggering in Canada detected using cataloged earthquakes combined with a matched-filter approach, *Geophys. Res. Lett.* **42,** no. 13, 5187–5196, doi: 10.1002/2015GL064377.

Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms, *IEEE Trans. Audio Electroacoust.* **15,** no. 2, 70–73, doi: 10.1109/TAU.1967.1161901.

West, M., J. J. Sánchez, and S. R. McNutt (2005). Periodically triggered seismicity at Mount Wrangell, Alaska, after the Sumatra earthquake, *Science* **308,** no. 5725, 1144–1146, doi: 10.1126/science.1112462.

Wu, C., Z. Peng, W. Wang, and Q.-F. Chen (2011). Dynamic triggering of shallow earthquakes near Beijing, China, *Geophys. J. Int.* **185,** no. 3, 1321–1334, doi: 10.1111/j.1365-246X.2011.05002.x.

Yang, H., and Z. Peng (2013). Lack of additional triggered tectonic tremor around the Simi Valley and the San Gabriel Mountain in Southern California, *Bull. Seismol. Soc. Am.* **103,** no. 6, doi: 10.1785/0120130117.

Yang, H., L. Zhu, and R. Chu (2009). Fault-plane determination of the 18 April 2008 mount Carmel, Illinois, earthquake by detecting and relocating aftershocks, *Bull. Seismol. Soc. Am.* **99,** no. 6, 3413–3420, doi: 10.1785/0120090038.

Yun, N., S. Zhou, H. Yang, H. Yue, and L. Zhao (2019). Automated detection of dynamic earthquake triggering by the high-frequency power integral ratio, *Geophys. Res. Lett.* **46,** no. 22, 12,977–12,985, doi: 10.1029/2019GL083913.

Zhou, Y., H. Yue, S. Zhou, and Q. Kong (2019). Hybrid event detection and phase-picking algorithm using convolutional and recurrent neural networks, *Seismol. Res. Lett.* **90,** no. 3, 1079–1087, doi: 10.1785/0220180319.

Zhu, L., Z. Peng, J. McClellan, C. Li, D. Yao, Z. Li, and L. Fang (2019). Deep learning for seismic phase detection and picking in the aftershock zone of 2008 Mw 7.9 Wenchuan Earthquake, *Phys. Earth Planet. In.* **293,** 106261, doi: 10.1016/j.pepi.2019.05.004.

Manuscript received 12 June 2020
Published online 21 October 2020