pyCSEP: A Python Toolkit for Earthquake Forecast Developers

William H. Savran[1], José A. Bayona[2], Pablo Iturrieta[3], Khawaja M. Asim[3], Han Bao[4], Kirsty Bayliss[5], Marcus Herrmann[6], Danijel Schorlemmer[3], Philip J. Maechling[1], and Maximilian J. Werner[2]

[1]University of Southern California

[2]University of Bristol

[3]GFZ Potsdam

[4]University of California, Los Angeles

[5]University of Edinburgh

[6]University of Naples ‘Federico II’, Italy

Corresponding Author

William H. Savran, Zumberge Hall of Science (ZHS), 3651 Trousdale Pkwy, Los Angeles, CA 90089-0740, wsavran@usc.edu

Declaration of Competing Interests

The authors acknowledge there are no conflicts of interest recorded.

ABSTRACT

The Collaboratory for the Study of Earthquake Predictability (CSEP) is an open and global community whose mission is to accelerate earthquake predictability research through rigorous testing of probabilistic earthquake forecast models and prediction algorithms. pyCSEP supports this mission by providing open-source implementations of useful tools for evaluating earthquake forecasts. pyCSEP is a Python package that contains the following modules: (1) earthquake catalog access and processing, (2) representations of probabilistic earthquake forecasts, (3) statistical tests for evaluating earthquake forecasts, and (4) visualization routines and various other utilities. Most significantly, pyCSEP contains several statistical tests needed to evaluate earthquake forecasts, which can be forecasts expressed as expected earthquake rat es in space-magnitude bins or specified as large sets of simulated catalogs (which includes candidate models for governmental operational earthquake forecasting). To showcase how pyCSEP can be used to evaluate earthquake forecasts, we have provided a reproducibility package that contains all the components required to recreate the figures published in this article. We recommend that interested readers work through the reproducibility package alongside this manuscript. By providing useful tools to earthquake forecast modelers and facilitating an open-source software community, we hope to broaden the impact of the Collaboratory for the Study of Earthquake Predictability (CSEP) and further promote earthquake forecasting research.

INTRODUCTION

The Collaboratory for the Study of Earthquake Predictability

CSEP emerged from the need to place the research field on more robust methodological footing to help overcome the negative sentiment surrounding earthquake prediction efforts (e.g., Geller, 1997). CSEP formed as a collaboration to assess earthquake predictability and provide users of earthquake forecasts with confidence about forecast skill and performance (e.g., government agencies that issue operational earthquake forecasts; Jordan and Jones, 2010; Jordan et al., 2011; Marzocchi et al., 2014). Past efforts were stymied by a range of problems that resulted in a lack of both reproducibility (the inability to regenerate previously issued forecasts, predictions, or test results) and replicability (the inability to reach the same conclusion about a model's predictive skill from different data; Stodden et al., 2018; National Academies of Sciences, Engineering, and Medicine and others, 2019). The peer-review process was frequently insufficient to ensure these necessary standards, an experience mirrored in other empirical research fields (Baker, 2016). Meaningful prospective evaluations require sufficient data, which may take several decades or more to collect in certain regions, especially for large earthquakes. CSEP's multi-region approach and global experiments, which would not be possible without its international collaboration, help alleviate this limitation (e.g., Bird et al., 2015). Although progress in forecast testing may be limited by time, even a few years of data help scientists to falsify certain hypotheses that are inconsistent with observations (Dekel and Feinberg, 2006).

The main pillar of CSEP's approach is the prospective testing of forecasts (i.e., against future observations) in reproducible and transparent forecast experiments carefully designed by the community. Prospective evaluations require that forecasts are unambiguously testable, with all model parameters, forecast specifications, and qualifying target data sources specified in advance, preferably before testing observations were made (Schorlemmer and Gerstenberger, 2007; Schorlemmer et al., 2018). This ensures a zero-degree-of-freedom, independent test of a model's or algorithm's performance.

Starting in 2007, CSEP has managed testing centers that autonomously run prospective forecast experiments (Schorlemmer and Gerstenberger, 2007). In these, automated dispatchers run forecast models to generate forecasts and evaluate them against prospective data (Zechar et al., 2010). Testing centers existed in California, New Zealand, Italy, Japan, and China, and together hosted over 400 models and model versions in a variety of tectonic settings and at a global scale (e.g., Field, 2007; Marzocchi et al., 2014; Tsuruoka et al., 2012; Zechar et al., 2013; Taroni et al., 2018; Strader et al., 2018; Rhoades et al., 2018; Eberhard et al., 2012; Bayona et al., 2021). Through this major community effort, CSEP has provided new insights into the predictability of earthquakes, provided independent assessments of the predictive skills of a range of scientific hypotheses of seismogenesis, galvanised model improvements and motivated new research into evaluation methods (Schorlemmer et al., 2018).

After a decade of operating the CSEP testing centers, it became apparent that the monolithic software design was too strongly entangled with the system architecture and data bookkeeping to support the new types of forecast experiments that the CSEP community would like to conduct (Schorlemmer et al., 2018). CSEP software has always been open-source and accessible; however, in practice, the code was difficult to use by individual researchers. Specifically, the testing center software coupled the evaluation routines with the system architecture making it difficult to use them outside of the testing center context. We developed pyCSEP as the first of many steps to modernize CSEP testing centers and experiments. Modern testing centers should use pyCSEP as a library, decoupling the testing center architcture from the evaluation routines. Additionally, they should follow modern open-science principles to ensure that experiment results are versioned and openly available to the public (e.g., Wilkinson et al., 2016). Testing centers are crucial for addressing the replicability of experimental results, because long-standing prospective experiments are required to capture the time-scales needed for model improvements and updates.

pyCSEP: A Python Toolkit for Earthquake Forecast Developers

Strengthening the collaborative aspects of the CSEP community and increasing the sustainability of CSEP activities, requires a new and collaborative mode of software development with the goal of a flexible, open-source, and community-based processing toolkit. Building sustainable research software requires a community that bridges software engineers and scientists (Anzt et al., 2021). This open-source approach is ideal for research software, as it allows for transparent, extendable code development by the research community that is using the software. It allows practitioners of the code to implement new features and identify potential issues in the software, and become engaged with the development process creating a net benefit for all involved members. We conceived of the open-source pyCSEP toolkit to address this limitation and to create a software community to promote earthquake forecasting research.

At its core, pyCSEP re-implements software running in CSEP testing centers as an open-source Python package, but is already rapidly expanding beyond this. pyCSEP is designed so researchers can evaluate earthquake forecasting models with minimal effort using a beginner-friendly, object-oriented interface. pyCSEP's modular structure allows for easy extensibility (Fig. 1). We encourage researchers to contribute code directly to the toolkit. To enable reproducible research, we strive for collaboratively developed code that is readable, well documented, and, most importantly, vetted. The source code can be found in the GitHub repository for this project (see link in *Data and Resources* section). Savran et al. (2022) provides a brief overview of the motivation for developing pyCSEP. The review process for that article focuses on software development best-practices, and examines the software repository and documentation. This article complements the software focused publication by providing more thorough explanations of the functionality of the software and providing the accompanying reproducibility package.

Software Development Principles

We incorporate several best-practices used by many open-source software projects (e.g., Hunter, 2007; McKinney, 2010; pandas Development Team, 2020) into our development process. In the code repository, we use continuous integration (CI) tools to ensure all new code contributions build successfully and pass unit tests. CI tools trigger workflows in the software repository to run development tasks automatically. The CI tools also build and publish the online documentation (link in *Data and Resources* section). These workflows trigger automatically when changes are made to the 'main' branch of the repository, or when new contributions have been submitted as pull requests in GitHub. We follow the SemVer (https://semver.org) guidelines for software versioning. New releases are made available on `PyPI` and `conda-forge` and can be installed using the package managers pip or conda. Additionally, pyCSEP strives to meet the target best practices as proposed by the Computational Infrastructure for Geodynamics (link in *Data and Resources* section).

Reproducibility of Forecasting Experiments

In CSEP testing centers, experiment components (e.g., model software, input data, forecasts, target data, and test results) were stored on CSEP servers with no external access (Schorlemmer and Gerstenberger, 2007). This approach ensured the integrity and reproducibility of the experiments, but required substantial data management and systems administration resources. The controlled environment of CSEP testing centers also made it difficult to share experimental results. The recent proliferation of freely available online data storage and management tools provide an effective alternative for storing experiment data and code. We encourage the use of these tools to create reproducibility packages (Krafczyk et al., 2021) for publications of earthquake forecasting experiments. A reproducibility package contains the software, data, and other experiment artifacts required to exactly reproduce published results. To illustrate this idea and provide an introduction to pyCSEP, we provide an example reproducibility package for this article (link in *Data and Resources* section).

PYCSEP SOFTWARE

pyCSEP provides an open-source implementation of several peer-reviewed statistical tests for evaluating probabilistic earthquake forecasts (Schorlemmer et al., 2007; Zechar et al., 2010; Rhoades et al., 2011; Werner et al., 2011; Savran et al., 2020). The design includes core classes that represent earthquake forecasts, catalogs, and spatial regions (Fig. 1). Higher-level functions using these classes are implemented to provide a simple interface to analyze forecasting models. Overall, the software design is modular to accommodate new forecast representations and evaluation types. Where possible, we integrate popular Python libraries such as numpy (Harris et al., 2020), matplotlib (Hunter, 2007), and pandas (pandas Development Team, 2020; McKinney, 2010) to allow users to easily include pyCSEP in existing scripts and workflows. pyCSEP also contains routines for working with and visualizing earthquake forecasts and catalogs. Also, general users of earthquake catalogs and gridded data sets may find useful utilities in the package.

Getting started with pyCSEP

The most straightforward way to install pyCSEP is using the `conda` package manager, and installing the most recent release from `conda-forge`. Users can obtain `conda` through the `Anaconda` or `miniconda` distributions. pyCSEP issues regular releases to `PyPI` and `conda-forge`. The latest release can be installed using:

```
conda install --channel conda-forge pycsep
```

The online documentation provides detailed installation instructions and examples that assist new users through tasks such as evaluating grid- and catalog-based earthquake forecasts, working with catalogs, and various plotting tasks (link in *Data and Resources* section).

Core Classes

The following subsections present a more technical introduction to the core classes in pyCSEP (see Fig. 1). Fig. 1 indicates how important methods and classes are related in the code, and can be used as a reference. We recommend interested readers to get started with pyCSEP by following the examples in the online documentation, and working through the reproducibility package (see Section *Reproducibility Packages* for this manuscript; link in *Data and Resources* section).

Regions

Regions are used to define the spatial cells of an earthquake forecast. In practice, they are used to bin, or discretize, an earthquake catalog into these spatial cells. Regions are fundamental in defining earthquake forecasts and preparing observed catalogs for evaluation (Fig. 1). In practice, a region represents a mapping between a list of spatial cells and spatial points. This mapping associates each point with its corresponding cell in the spatial region. There is a many-to-one relationship between points and spatial cells. Each point can only be associated with a single cell; however, a cell can contain many points. pyCSEP defines a standard region on a regular Cartesian grid whose cells have dimensions of $0.1° \times 0.1°$ in latitude and longitude. However, the dimensions of the cells are configurable within pyCSEP. To allow for easy interoperability with previous experiments, pyCSEP currently provides predefined regions for California (Fig. 2a), Italy (Fig. 2b) and the global testing region (not pictured). These can be accessed via simple function calls (e.g., *`california_relm_region()`*, see Fig. 1).

pyCSEP provides a class named *`CartesianGrid2D`* to represent the standard region used in CSEP experiments (e.g., Schorlemmer and Gerstenberger, 2007; Taroni et al., 2018). *`CartesianGrid2D`* implements the mapping so points can be correctly associated with the corresponding Cartesian spatial cells. The class provides flexibility for creating different regions by supplying a list containing the lower-left origin of each cell by calling the *`from_origins()`* class method. The cells are defined such that the lower and left-most edges are inclusive. Functionality for non-regular grids is not currently implemented in the toolkit; however, the object-oriented implementation of the region class allows for non-regular grids to be easily accommodated in the future.

Magnitude ranges are defined using a list containing the left bin edges, and require no additional classes. The magnitude bin edges should be made accessible through the magnitude member of the region classes. The *`regions.create_space_magnitude_region`* function provides a method to associate a discretized magnitude range with a particular spatial region.

Forecasts

Currently, pyCSEP supports two types of probabilistic earthquake forecasts (see Fig. 1). First, we support grid-based forecasts that express expected rates of earthquakes within discrete space-time-magnitude bins (e.g., Schorlemmer and Gerstenberger, 2007). A grid-based forecast is defined by the `GriddedForecast` class. This class is composed of two main data attributes: 1) a 2D numpy array that stores expected rates in space-magnitude bins, and 2) a pyCSEP region class that defines the space-magnitude cells of the forecast. Standard CSEP gridded forecasts use the `CartesianGrid2D` to define this mapping. Each forecast is considered to span a discrete time period, where the expected rate is based on this period. Thus, time-dependent forecasts with multiple periods require individual instances of the `GriddedForecast` class. Additional methods such as `target_event_rates()` are provided by the forecast class, and allow the users to retrieve event rates as defined by the forecast. Grid-based forecasts can be loaded from disk using the `load_gridded_forecast()` function defined in the top-level package.

The `CatalogForecast` class defines the second supported forecast type: catalog-based forecasts. This class represents forecasts that are defined by a list of earthquake catalogs (e.g., `CSEPCatalog` or `UCERF3Catalog`) and a region (e.g., `CartesianGrid2D`). The class provides an iterator implementing a user-defined set of catalog filters that apply automatically to each catalog in the forecast. Also, this implementation allows for working with large UCERF3-ETAS (or other) forecasts by loading the catalogs on demand. This is known as 'lazy' loading. `CatalogForecast` objects can be loaded from disk using the `load_catalog_forecast()` function defined in the top-level package. Fig. 3 shows an example of an UCERF3-ETAS forecast made during the 2019 Ridgecrest sequence. The reader will find examples of working with grid-based and catalog-based forecasts in the *Tutorials* section of the online documentation and the reproducibility package.

Evaluations

CSEP has lead research efforts into developing forecast evaluation methods, tests, and performance measures of probabilistic earthquake forecasts (e.g., Schorlemmer et al., 2007; Werner and Sornette, 2008; Zechar et al., 2010; Zechar and Jordan, 2010; Zechar and Zhuang, 2010; Rhoades et al., 2011; Werner et al., 2011; Marzocchi et al., 2012; Schneider et al., 2014; Gordon et al., 2015; Molchan et al., 2017; Savran et al., 2020, and many others). Different tests are used to address various hypotheses underlying the forecasts they are evaluating. pyCSEP currently contains a selection of consistency tests (comparing forecasts with data) and comparative tests (comparing models against each other on the basis of the data) for both grid-based forecasts and catalog-based forecasts. Different forecast formats require different evaluation methods. Grid-based forecasts use a set of evaluations that based on the Poisson likelihood function (Schorlemmer et al., 2007; Zechar et al., 2010), whereas catalog-based forecasts build empirical distributions to sample the uncertainty contained within the forecast (Nandan et al., 2019; Savran et al., 2020). The Poisson assumption has been widely criticized (Lombardi and Marzocchi, 2010a; Werner and Sornette, 2008) and pyCSEP was designed to accommodate evaluation with different likelihood functions (Bayona et al., 2022). We explain the evaluation methods implemented in pyCSEP below. Evaluations for grid-based forecasts are implemented in the module `poisson_evaluations`, and for catalog-based forecasts in the `catalog_evaluations` module (Fig. 1). Examples on how to evaluate grid- and catalog-based forecasts are shown in the *Tutorial* section of the online documentation. We provide an in-depth explanation of the evaluations along with working code examples in the Electronic Supplement to this article.

For grid-based forecasts, CSEP tests assess the consistency between the observed and the expected number, spatial, magnitude, and likelihood distributions of earthquakes, assuming that seismicity in space-magnitude cells is independent and Poisson-distributed (Zechar et al., 2010; Werner et al., 2011; Rhoades et al., 2011). In the following paragraphs, we provide a high-level overview of the test methods available for grid-based forecasts followed by a brief description of the consistency tests for catalog-based forecasts.

Number test The number (N) test (Schorlemmer et al., 2007; Zechar et al., 2010) evaluates if the total number of observed earthquakes (N_{obs}) falls within the 95% predictive distribution of the forecast distribution, with the expected rate, N_{fore}, equal to the sum of forecasted rates in each space-magnitude bin. Fig. 4 shows the N-test result for time-independent forecasts from the Regional Earthquake Likelihood Model (RELM) experiment that were originally published by Zechar et al. (2013).

Spatial test The spatial (S) test (Zechar et al., 2010) evaluates how well a forecast explains the spatial distribution of earthquakes. One first sums the expected rates in each spatial cell over the magnitude bins to isolate the spatial component of the forecast, and normalizes the resulting spatial rates to the total number of target observations. Next, one computes the (spatial) joint log-likelihood in each cell by evaluating the Poisson likelihood function in each cell, and summing the spatial log-likelihoods over the entire testing region. To assess whether this observed log-likelihood score could have been generated by the forecast, we obtain the distribution of spatial log-likelihood scores consistent with the forecast through simulation. In this and the following two tests, the number of simulated earthquakes is fixed to N_{obs} to remove the dependency on the forecasted rate. To assess the consistency between the observed locations and the spatial forecast, we examine where the observed value falls within the distribution of simulated values. This quantile score is equivalent to the p-value of a one-sided statistical test. In previous CSEP experiments, critical values of $\alpha = 0.01$ or $\alpha = 0.05$ were commonly chosen to reject the null hypothesis that the forecast could have generated the observed locations. However, in practice, we use the consistency tests as diagnostic tools to indicate a degree of (dis)agreement between a forecast and observations during the testing period (e.g., Bayona et al., 2022). Fig. 4b shows the S-test evaluation for time-independent Italian forecasts (originally published by Taroni et al., 2018).

Magnitude test The magnitude (M) test assesses the null hypothesis that the observed magnitude distribution is consistent with that of the forecast. Similarly to the S-test, the M-test (Zechar et al., 2010) first sums rates in each magnitdue bin over spatial cells and normalizes the forecast so that N_{fore} matches N_{obs}, thus isolating the magnitude distribution of the forecast. As with the S-test, the M-test then determines the quantile of the observed (magnitude) joint log-likelihood score in the distribution of joint log-likelihood scores simulated from the forecast. Observed scores in the tail of the model distribution indicate discrepancies between the forecast and data that might be scientifically interesting.

Conditional likelihood test The conditional likelihood (cL) test (Werner et al., 2010, 2011) null hypothesis states that the observed locations and magnitudes are consistent with the forecast conditional on the number of observed earthquakes, i.e. the test checks the joint space-magnitude distribution against the forecast. First, one computes the observed joint log-likelihood score by summing bin-wise log-likelihood scores over all space-magnitude bins. In this evaluation, the forecast rates are not normalized to match the observed rate. Again, we assess where this score falls in the critical range of the simulated distribution of joint log-likelihood scores. Small quantile scores again indicate discrepancies. Effectively, the CL test represents a combination of the S and M tests.

Comparative testing pyCSEP also provides comparative T- and W-tests (Rhoades et al., 2011) to evaluate the relative performance of two models, based on information gain scores per earthquake:

$$IGPE = \frac{1}{N}\sum_{i=1}^{N}[X_i - Y_i] - \frac{N_A - N_B}{N}, \tag{1}$$

where N is the number of observed earthquakes, and $X_i = \ln A(k_i)$ and $\ln Y_i = B(k_i)$ are the log-likelihood scores obtained by model A and model B in the bin k in which earthquake i occurred, and N_A and N_B are the expected number of earthquakes according to forecast A and B, respectively. The T-test assesses whether the $IGPE$ is statistically different from zero. Following Rhoades et al. (2011), one applies the Student's t-test to the $IGPE$ score of forecast A over forecast B. We consider forecast A to be significantly more skillful than forecast B if the $IGPE$ is positive and the confidence interval based on the Student's t-distribution does not include zero. Conversely, if the $IGPE$ is negative and the confidence interval does not include zero, forecast B is significantly more informative than model A. If the confidence interval includes zero, we consider differences in the score to be statistically insignificant. Fig. 5 shows T-test results for Californian and Italian forecasts, which were originally published by Zechar et al. (2013) and Taroni et al. (2018), respectively.

Testing catalog-based forecasts For catalog-based forecasts, pyCSEP provides (1) a number (N) test that compares the (non-Poissonian) number distribution from the forecasts against the observed number of earthquakes; (2) a magnitude (M) test that compares the sum of bin-wise differences in the incremental magnitude distribution; (3) a spatial (S) test that compares the geometric mean of the target event rates; and 4) a pseudo-likelihood test based on a statistic that resembles the likelihood of a continuous point-process (Savran et al., 2020). These tests are essentially analogues of the aforementioned consistency tests, but they relax the Poissonian assumption. For a full description of these evaluations and their application to UCERF3-ETAS forecasts made during the 2019 Ridgecrest earthquake sequence in California, see Savran et al. (2020). In Fig. 6, we show an example of the N-test and S-test for a single seven-day UCERF3-ETAS forecast made immediately after the occurrence of the **M**7.1 mainshock of the Ridgecrest sequence. The catalog-based evaluations are available in the `catalog_evaluations` module in pyCSEP.

Plotting and Other Utilities

Along with the routines for statistical tests, pyCSEP provides a thin wrapper around the matplotlib (Hunter, 2007) and cartopy (Met Office, 2015) plotting libraries to provide functions that visualize test results, catalogs, and spatial forecast maps (Fig. 7). We aim to keep the plotting capabilities both easily accessible for early users (i.e., by calling simple methods within most of pyCSEP core classes) and customizable enough to provide journal-quality figures, including: text formatting, legend and colormap editing, spatial grids, and preparing multi-panel figures. The implementation provides access to cartopy's projection capabilities as well as basic maps, along with various (or user-defined) web-service tiled maps. We intend to keep the plotting functions modular, so that multiple outputs can be combined in single figures, and to preserve the plots if the user requires post-processing of the data or results (as shown in Fig. 7).

REPRODUCIBILITY PACKAGES

CSEP forecasting experiments have run in testing centers, which provide a controlled environment that prevents any access and modification of ongoing experiments. Because pyCSEP now provides the ability to configure bespoke earthquake forecasting experiments, we anticipate that researchers will be interested in using these methods to evaluate their own forecasts. We encourage researchers that use pyCSEP in their publications to follow the approach outlined by Krafczyk et al. (2021) and provide a reproducibility package for their publication.

A reproducibility package is a structured set of code, data, and other files that are required to recreate all figures and tables within a manuscript. To illustrate this principle, we provide a reproducibility package for this manuscript. The entry point of the reproducibility package is a script with the following responsibilities: (1) retrieve and verify data artifacts from Zenodo; (2) create a Docker image with the version of pyCSEP, and its dependencies, used for this publication; and (3) run a program to reproduce the figures from this article. Once the reader obtains the reproducibility package, there is a single command to reproduce all of the figures from this paper. We encourage users to try and run the reproducibility package for this manuscript (link for the reproducibility package in the *Data and Resources* section).

PYCSEP COMMUNITY

The pyCSEP efforts aim to strengthen the community of earthquake scientists with an interest in forecasting. We intend to unite researchers interested in all aspects of earthquake forecasting from model development to testing and evaluation to make the process of forecast testing as transparent and accessible as possible. Fig. 8 is a screenshot from our first community workshop, held (virtually) in March 2021 for modelers involved in the project RISE (‘*Real-time earthquake rIsk reduction for a reSilient Europe*’, financed by the European Commission’s *Horizon 2020* program. The workshop introduced forecast developers to the pyCSEP toolkit, helped to identify where improvements and extensions could be made, and invited modelers to contribute. It was held over three sessions, with the first introducing pyCSEP testing, the second allowing modelers to present their current forecasting work, and the third focusing on a hands-on tutorial session. The workshop brought together modelers and model testers to understand the needs of both groups and familiarize all participants with the testing and visualization options currently available in the toolkit. This was later followed by a workshop on contributing to the pyCSEP project through GitHub to familiarize interested users with open-source community software development.

Two tutorials were created for the workshop to demonstrate the process of model testing with pyCSEP for grid-based and catalog-based forecasts. The tutorials are in the form of interactive Jupyter notebooks (Kluyver et al., 2016) that provide a template for the key steps of model testing with pyCSEP. Both tutorials use real forecasts and catalog data similar to the examples in this paper. The tutorials are available on the pyCSEP online documentation (link in Data and Resources section), which also includes an installation guide, and a detailed user guide that covers the core concepts to details need to extend pyCSEP functionality.

Open Call for Developers

The workshops highlighted that pyCSEP greatly benefits from an active engagement of its community. Sustaining the development is a community effort and new contributions are essential to extend and improve pyCSEP’s utility. In this regard—and to leverage the open-source development approach—we welcome researchers and developers to join our community and to contribute new ideas and methods (e.g., advanced evaluation capabilities, more robust tests, more efficient testing, etc.). Within the GitHub repository, these contributions can be introduced in the form of 'pull requests' (i.e., suggested code changes, improved documentation), or 'issues' (e.g., comments or suggestions about technical and scientific approaches). The contributions are transparent and the community can discuss them together. The pyCSEP community additionally meets in regular (developer) calls to coordinate contributions more interactively (e.g., by reviewing source code and new ideas).

CONCLUSION

pyCSEP is an open-source Python package that provides routines for evaluating probabilistic earthquake forecasting models that are expressed as earthquake rates in discrete space-magnitude cells and simulation-based forecasts consisting of synthetic earthquake catalogs. pyCSEP also includes utilities for visualizing forecasts and earthquake catalogs, and configuring earthquake forecasting experiments. The implementation follows best-practices for open-source software development including documentation and continuous integration to build and test new code contributions. In CSEP, we are adopting a software development process that encourages contributions from researchers. To date we have received contributions that have added new evaluation methods and improved plotting capabilities. We advocate that publications involving pyCSEP are accompanied by reproducibility packages. Additionally, we have started a workshop series to train researchers on using pyCSEP and collaborating in open-source development. In 2021, we hosted two workshops teaching users how to use pyCSEP and to work collaboratively in GitHub. We encourage all interested users to visit the online documentation and the code repository to learn more about pyCSEP.

DATA AND RESOURCES

The pyCSEP software can be found on GitHub at `https://github.com/SCECCode/pycsep` and the documentation can be found at `https://docs.cseptesting.org`. The reproducibility package for this manuscript can be found at `https://doi.org/10.5281/zenodo.6626265` and the data can be found at `https://doi.org/10.5281/zenodo.5777992`. Best-practices from Computational Infrastructure for Geodynamics (CIG) can be found at `https://geodynamics.org/software/software-bp`. The link to GitHub actions documentation can be found at `https://docs.github.com/en/actions`. The RISE project website can be found at `http://www.rise-eu.org`. Map-tiles for plotting maps can be found at `https://maps.stamen.com`. All websites were last accessed on 24 June 2022.

ACKNOWLEDGMENTS

We would like to thank the editors and two anonymous reviewers for their comments on this article. This research was supported by the Southern California Earthquake Center (Contribution No. 11740). SCEC is funded by NSF Cooperative Agreement EAR-1600087 & USGS Cooperative Agreement G17AC00047. This project has received funding from the European Union's *Horizon 2020 research and innovation program* under Grant Agreement Number 821115, *Real-Time Earthquake Risk Reduction for a Resilient Europe* (RISE).

REFERENCES

Anzt, H., Bach, F., Druskat, S., Löffler, F., Loewe, A., Renard, B. Y., Seemann, G., Struck, A., Achhammer, E., Aggarwal, P., Appel, F., Bader, M., Brusch, L., Busse, C., Chourdakis, G., Dabrowski, P. W., Ebert, P., Flemisch, B., Friedl, S., Fritzsch, B., Funk, M. D., Gast, V., Goth, F., Grad, J.-N., Hegewald, J., Hermann, S., Hohmann, F., Janosch, S., Kutra, D., Linxweiler, J., Muth, T., Peters-Kottig, W., Rack, F., Raters, F. H., Rave, S., Reina, G., Reißig, M., Ropinski, T., Schaarschmidt, J., Seibold, H., Thiele, J. P., Uekermann, B., Unger, S., and Weeber, R. (2021). An environment for sustainable research software in germany and beyond: current state, open challenges, and call for action. *F1000Research*, 9:295.

Baker, M. (2016). Reproducibility crisis. *Nature*, 78(26):353–66.

Bayona, J., Savran, W., Strader, A., Hainzl, S., Cotton, F., and Schorlemmer, D. (2021). Two global ensemble seismicity models obtained from the combination of interseismic strain measurements and earthquake-catalogue information. *Geophysical Journal International*, 224(3):1945–1955.

Bayona, J. A., Savran, W. H., Rhoades, D. A., and Werner, M. J. (2022). Prospective evaluation of multiplicative hybrid earthquake forecasting models in California. *Geophysical Journal International*. ggac018.

Bird, P., Jackson, D. D., Kagan, Y. Y., Kreemer, C., and Stein, R. S. (2015). Gear1: A global earthquake activity rate model constructed from geodetic strain rates and smoothed seismicity. *Bulletin of the Seismological Society of America*, 105(5):2538–2554.

Bird, P. and Liu, Z. (2007). Seismic hazard inferred from tectonics: California. *Seismological Research Letters*, 78(1):37–48.

Dekel, E. and Feinberg, Y. (2006). Non-bayesian testing of a stochastic prediction. *The Review of Economic Studies*, 73(4):893–906.

Ebel, J. E., Chambers, D. W., Kafka, A. L., and Baglivo, J. A. (2007). Non-poissonian earthquake clustering and the hidden markov model as bases for earthquake forecasting in california. *Seismological Research Letters*, 78(1):57–65.

Eberhard, D. A. J., Zechar, J. D., and Wiemer, S. (2012). A prospective earthquake forecast experiment in the western pacific. *Geophysical Journal International*, 190(3):1579–1592.

Field, E. H. (2007). Overview of the working group for the development of regional earthquake likelihood models (relm). *Seismological Research Letters*, pages 1–10.

Geller, R. J. (1997). Earthquake prediction: a critical review. *Geophysical Journal International*, 131(3):425–450.

Gordon, J. S., Clements, R. A., Schoenberg, F. P., and Schorlemmer, D. (2015). Voronoi residuals and other residual analyses applied to csep earthquake forecasts. *Spatial Statistics*, 14:133–150.

Group, M. W. (2004). Redazione della mappa di pericolosità sismica prevista dall'ordinanza pc del 20 marzo 2003, n. 3274, all. 1 rapporto conclusivo.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Helmstetter, A., Kagan, Y. Y., and Jackson, D. D. (2007). High-resolution time-independent grid-based forecast for m /*geq* 5 earthquakes in california. *Seismological Research Letters*, 78(1):78–86.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Jordan, T. H., Chen, Y. T., Gasparini, P., Madariaga, R., Main, I., Marzocchi, W., and Papadopoulos, G. (2011). Operational earthquake forecasting. state of knowledge and guidelines for utilization. *Annals of Geophysics*.

Jordan, T. H. and Jones, L. M. (2010). Operational earthquake forecasting: Some thoughts on why and how. *Seismological Research Letters*, 81(4):571–574.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., and Willing, C. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. In Loizides, F. and Schmidt, B., editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press.

Krafczyk, M. S., Shi, A., Bhaskar, A., Marinov, D., and Stodden, V. (2021). Learning from reproducing computational results: introducing three principles and the reproduction package. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2197).

Lombardi, A. and Marzocchi, W. (2010a). The assumption of poisson seismic-rate variability in csep/relm experiments. *Bulletin of the Seismological Society of America*, 100(5A):2293–2300.

Lombardi, A. M. and Marzocchi, W. (2010b). The etas model for daily forecasting of italian seismicity in the csep experiment. *Annals of Geophysics*.

Marzocchi, W., Garcia-Aristizabal, A., Gasparini, P., Mastellone, M. L., and Di Ruocco, A. (2012). Basic principles of multi-risk assessment: a case study in italy. *Natural hazards*, 62(2):551–573.

Marzocchi, W., Lombardi, A. M., and Casarotti, E. (2014). The establishment of an operational earthquake forecasting system in italy. *Seismological Research Letters*, 85(5):961–969.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and

Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Met Office (2010 - 2015). *Cartopy: a cartographic python library with a Matplotlib interface*. Exeter, Devon.

Molchan, G., Romashkova, L., and Peresan, A. (2017). On some methods for assessing earthquake predictions. *Geophysical Journal International*, 210(3):1474–1480.

Nandan, S., Ouillon, G., Sornette, D., and Wiemer, S. (2019). Forecasting the full distribution of earthquake numbers is fair, robust, and better. *Seismological Research Letters*, 90(4):1650–1659.

National Academies of Sciences, Engineering, and Medicine and others (2019). *Reproducibility and replicability in science*. National Academies Press.

pandas Development Team, T. (2020). pandas-dev/pandas: Pandas.

pyCSEP Developers (2021). pycsep online documentation.

Rhoades, D. A., Christophersen, A., Gerstenberger, M. C., Liukis, M., Silva, F., Marzocchi, W., Werner, M. J., and Jordan, T. H. (2018). Highlights from the first ten years of the new zealand earthquake forecast testing center. *Seismological Research Letters*, 89(4):1229–1237.

Rhoades, D. A., Schorlemmer, D., Gerstenberger, M. C., Christophersen, A., Zechar, J. D., and Imoto, M. (2011). Efficient testing of earthquake forecasting models. *Acta Geophysica*, 59(4):728–747.

Savran, W. H., J., W. M., D., S., and J., M. P. (2022). pycsep: A python toolkit for earthquake forecast developers. *Journal of Open Source Software*.

Savran, W. H., Werner, M. J., Marzocchi, W., Rhoades, D. A., Jackson, D. D., Milner, K., Field, E., and Michael, A. (2020). Pseudoprospective evaluation of ucerf3-etas forecasts during the 2019 ridgecrest sequence. *Bulletin of the Seismological Society of America*, 110(4):1799–1817.

Schneider, M., Clements, R., Rhoades, D., and Schorlemmer, D. (2014). Likelihood-and residual-based evaluation of medium-term earthquake forecast models for california. *Geophysical Journal International*, 198(3):1307–1318.

Schorlemmer, D., Gerstenberger, M., Wiemer, S., Jackson, D. D., and Rhoades, D. A. (2007). Earthquake likelihood model testing. *Seismological Research Letters*, 78(1):17–29.

Schorlemmer, D. and Gerstenberger, M. C. (2007). Relm testing center. *Seismological Research Letters*, 78(1):30–36.

Schorlemmer, D., Werner, M. J., Marzocchi, W., Jordan, T. H., Ogata, Y., Jackson, D. D., Mak, S., Rhoades, D. A., Gerstenberger, M. C., Hirata, N., Liukis, M., Maechling, P. J., Strader, A., Taroni, M., Wiemer, S., Zechar, J. D., and Zhuang, J. C. (2018). The collaboratory for the study of earthquake predictability: Achievements and priorities. *Seismological Research Letters*, 89(4):1305–1313.

Stodden, V., Seiler, J., and Ma, Z. (2018). An empirical analysis of journal policy effectiveness for computational reproducibility. *Proceedings of the National Academy of Sciences*, 115(11):2584–2589.

Strader, A., Werner, M., Bayona, J., Maechling, P., Silva, F., Liukis, M., and Schorlemmer, D. (2018). Prospective evaluation of global earthquake forecast models: 2 yrs of observations provide preliminary support for merging smoothed seismicity with geodetic strain rates. *Seismological Research Letters*.

Taroni, M., Marzocchi, W., Schorlemmer, D., Werner, M. J., Wiemer, S., Zechar, J. D., Heiniger, L., and Euchner, F. (2018). Prospective csep evaluation of 1-day, 3-month, and 5-yr earthquake forecasts for italy. *Seismological Research Letters*, 89(4):1251–1261.

Tsuruoka, H., Hirata, N., Schorlemmer, D., Euchner, F., Nanjo, K. Z., and Jordan, T. H. (2012). Csep testing center and the first results of the earthquake forecast testing experiment in japan. *Earth, Planets and Space*, 64(8):661–671.

Werner, M. J., Helmstetter, A., Jackson, D. D., and Kagan, Y. Y. (2011). High-resolution long-term and short-term earthquake forecasts for california. *Bulletin of the Seismological Society of America*, 101(4):1630–1648.

Werner, M. J., Helmstetter, A., Jackson, D. D., Kagan, Y. Y., and Wiemer, S. (2010). Adaptively smoothed seismicity earthquake forecasts for italy. *arXiv preprint arXiv:1003.4374*.

Werner, M. J. and Sornette, D. (2008). Magnitude uncertainties impact seismic rate estimates, forecasts, and predictability experiments. *Journal of Geophysical Research: Solid Earth*, 113(B8).

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J. G., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 't Hoen, P. A. C., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., and Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018.

Zechar, J. D., Gerstenberger, M. C., and Rhoades, D. A. (2010). Likelihood-based tests for evaluating space-rate-magnitude earthquake forecasts. *Bulletin of the Seismological Society of America*, 100(3):1184–1195.

Zechar, J. D. and Jordan, T. H. (2010). Simple smoothed seismicity earthquake forecasts for italy. *Annals of Geophysics*, 53(3):99–105.

Zechar, J. D., Schorlemmer, D., Werner, M. J., Gerstenberger, M. C., Rhoades, D. A., and Jordan, T. H. (2013). Regional earthquake likelihood models i: First-order results. *Bulletin of the Seismological*

Society of America, 103(2A):787–798.

Zechar, J. D. and Zhuang, J. (2010). Risk and return: evaluating reverse tracing of precursors earthquake predictions. *Geophysical Journal International*, 182(3):1319–1326.

AUTHOR LIST

William Savran, Zumberge Hall of Science (ZHS), 3651 Trousdale Pkwy, Los Angeles, CA 90089-0740

José A. Bayona, School of Earth Sciences, University of Bristol, Wills Memorial Building, Queens Road, Bristol BS8 1RJ, United Kingdom

Pablo Iturrieta, Helmholtzstraße 7, 14467 Potsdam, Germany

Khawaja M. Asim, Helmholtzstraße 7, 14467 Potsdam, Germany

Han Bao, Department of Earth, Planetary, and Space Sciences, University of California, Los Angeles, 595 Charles Young Drive East, Box 951567, Los Angeles, CA 90095-1567

Kirsty Bayliss, The King's Buildings, James Hutton Road, Edinburgh EH9 3FE

Marcuss Hermann, Corso Umberto I, 40, 80138 Napoli NA, Italy

Danijel Schorlemmer, Helmholtzstraße 7, 14467 Potsdam, Germany

Philip J. Maechling, Zumberge Hall of Science (ZHS), 3651 Trousdale Pkwy, Los Angeles, CA 90089-0740

Maximilian J. Werner, School of Earth Sciences, University of Bristol, Wills Memorial Building, Queens Road, Bristol BS8 1RJ, United Kingdom

List of Figures

Figure 4: (a) N-test results for the HELMSTETTER (Helmstetter et al., 2007), BIRD_LIU (Bird and Liu, 2007), and EBEL (Ebel et al., 2007) earthquake forecasts for California. The markers depict the number of **M**4.95+ earthquakes during the 2006–2010 RELM evaluation period. The green square indicates that this number falls within the 95% range of the forecast number distribution (solid bar), whereas red circles indicate inconsistencies between the forecast and observations. Thus, the observed number of earthquakes is consistent with the HELMSTETTER model, whereas the BIRD_LIU and EBEL models overestimate seismicity in the region. (b) Results of the S-test for the LOMBARDI (Lombardi and Marzocchi, 2010b), MELETTI (Group, 2004) and WERNER-M1 (Werner et al., 2010) forecasts for Italy. The markers represent the spatial joint log-likelihood of each model. The green square indicates that the spatial distribution forecasted by the LOMBARDI model is consistent with the spatial distribution of observed seismicity at a 0.01 significance level. Red circles indicate that the observed locations are inconsistent with the spatial forecasts by the MELETTI and WERNER models. In both panels, significance levels of the test are chosen from the original publication of these results (i.e., Taroni et al., 2018; Zechar et al., 2013).

Figure 5: (a) T-test results comparing the BIRD_LIU and EBEL forecasts with the benchmark HELMSTETTER forecast (horizontal dashed line) in California. Red circles indicate information gains of the forecasts (here both negative) and vertical red bars show 95% confidence intervals. These results indicate that HELMSTETTER is significantly more informative than BIRD_LIU and EBEL during the evaluation period. (b) Results of the T-test for the LOMBARDI and WERNER forecasts with respect to the MELETTI reference forecast (horizontal dashed line), in Italy. White circles display information gains of the forecasts and vertical gray bars represent 95% confidence intervals. The horizontal dashed line falls within both confidence intervals, indicating that these models are as statistically informative as the MELETTI benchmark model.

Figure 6: (a) Expected rate of **M**2.5+ events within $0.025^\circ \times 0.025^\circ$ spatial cells from a week-long UCERF3-ETAS forecast. The expected rates are computed by averaging over the simulated catalogs. The red circles show the locations of the 827 events observed during the forecast period. (b) N-test result for this UCERF3-ETAS forecast. The histogram shows the forecast number distribution with the two-sided 5% critical region highlighted red. The number of observed events, ω, is depicted as the dashed line, which falls near the mode of the forecast distribution, indicating consistency. δ_1 equals the fraction of catalogs in the forecast that predict at least as many earthquakes as the observations, and δ_2 denotes the fraction of catalogs that contain at most the number of observed earthquakes. (c) S-test for the UCERF3-ETAS forecast. The histogram shows the distribution of simulated scores, computed by assigning each earthquake an expected rate based on the bin-wise values in (a) and computing the geometric mean over each catalog in the forecast. The two-sided 5% critical region is highlighted red. We compute the same statistic from the observed catalog, and show the value, ω, using the dashed line. The score, γ, shows the quantile where the observed value falls in the forecast distribution.

Figure 7: Example of the spatial plotting capabilities of pyCSEP through a thin-wrapper over cartopy. (a) A quick basemap can be obtained from the default plotting arguments that uses map tiles by Stamen Design (link in Data and Resources section). (b) On top of the basemap, two post-processed forecasts (the ratio between them over a given range of magnitudes). (c) The observed catalog within the same magnitude range, with auto-scaled symbols according to their magnitudes. These functions are intended to be used with pyCSEP classes and provide a simple way of visualizing spatial earthquake forecasts and catalogs.

Figure 8: A screenshot of the participants at the virtual pyCSEP training workshop in March 2021 hosted by the RISE project.

FIGURES

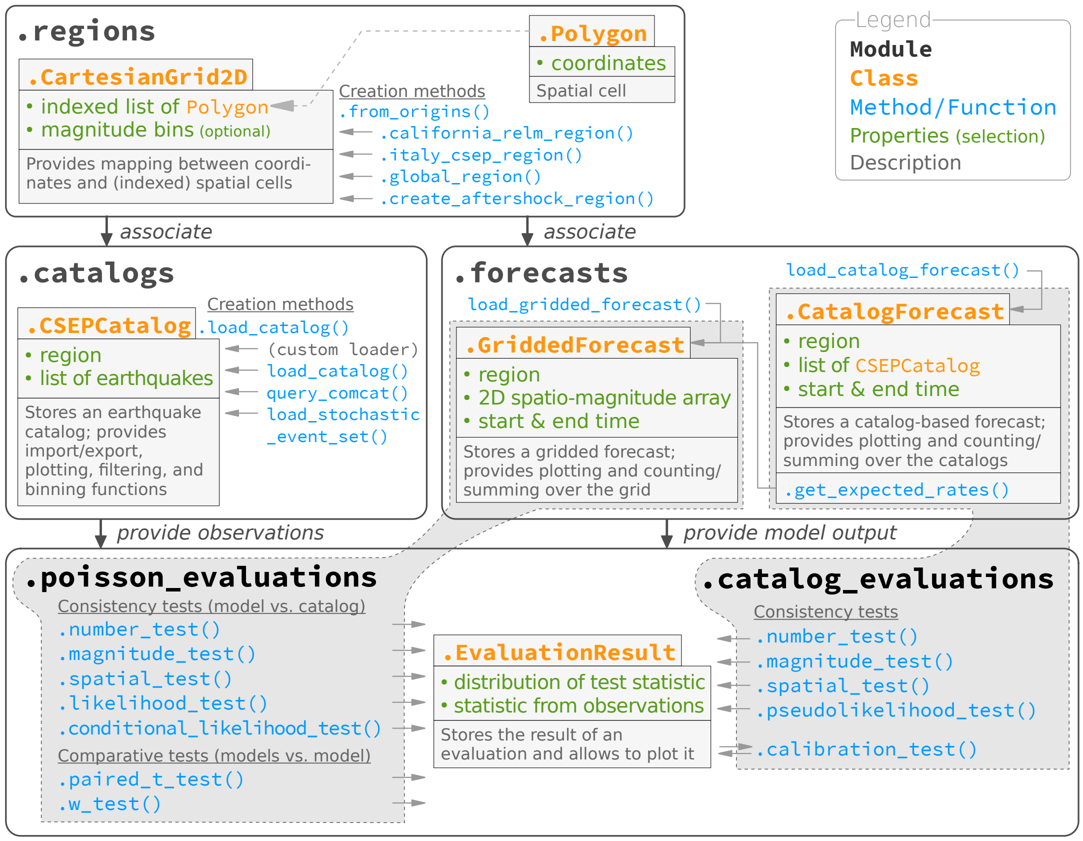

Figure 1. Schematic of the pyCSEP classes and code structure showing the core classes of pyCSEP. For a complete description of the software, please see the online documentation (pyCSEP Developers, 2021).

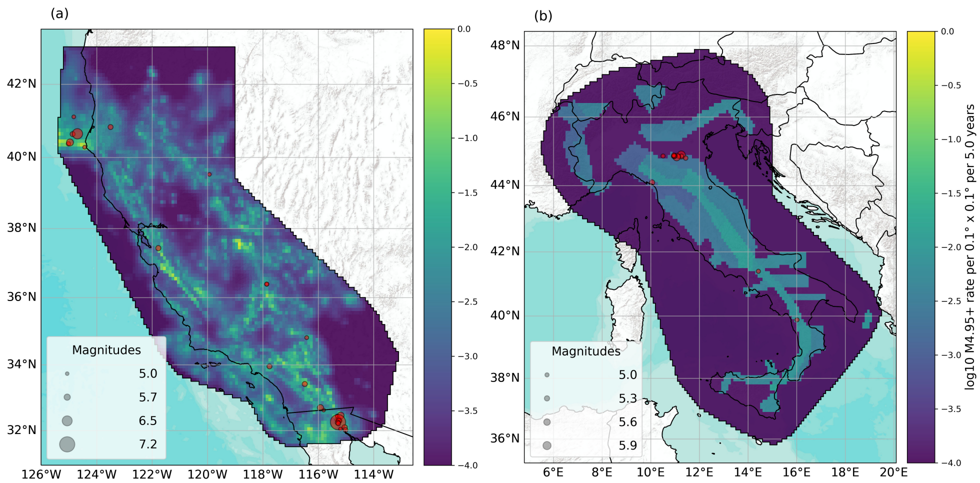

Figure 2. (a) Time-independent grid-based forecasts for the HELMSTETTER model for California (Helmstetter et al., 2007), and (b) the MELETTI model for Italy (Group, 2004). The colormap reflects the logarithm of the expected rate of **M**4.5+ earthquakes in $0.1^\circ \times 0.1^\circ$ spatial bins over a five year period. The red circles depict locations of observed earthquakes during the five-year evaluation period. Earthquakes are shown atop the HELMSTETTER forecast from 01 January 2006 through 01 January 2011, and atop the MELETTI forecast from 01 January 2010 through 01 January 2015.

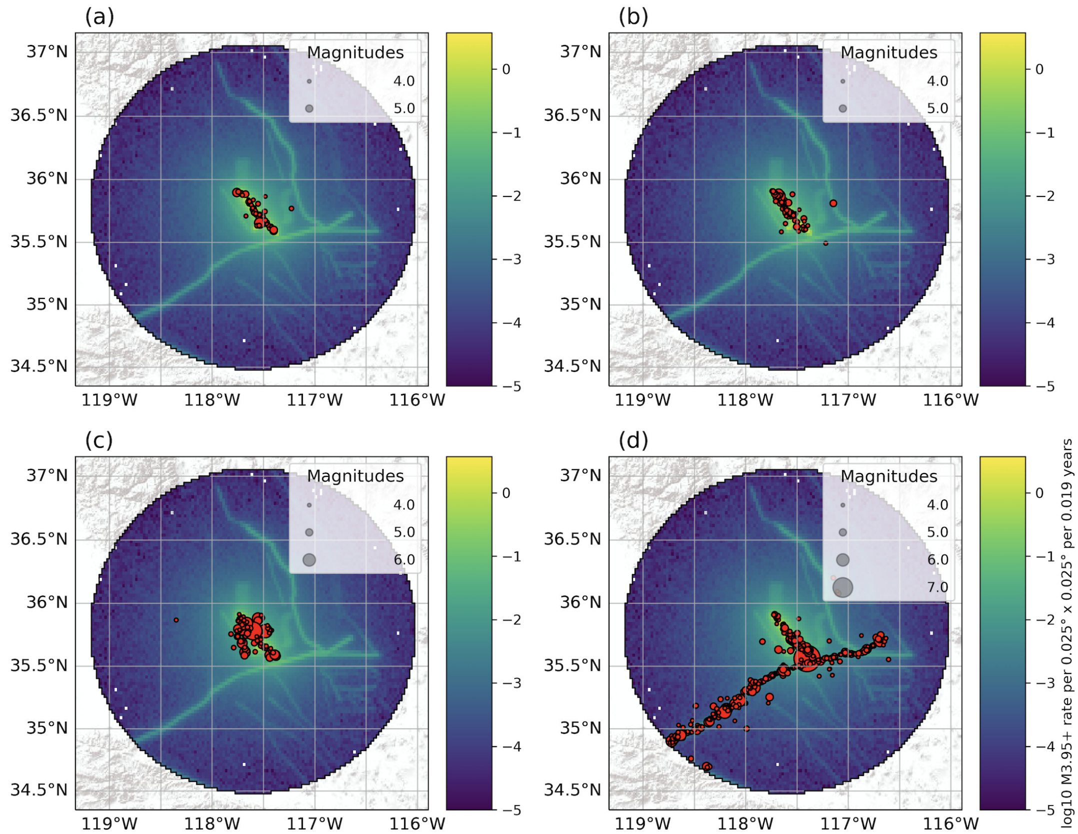

Figure 3. Select realizations (synthetic catalogs) from a week-long UCERF3-ETAS forecast generated during the 2019 Ridgecrest, California, sequence. The forecast starts immediately following the **M**7.1 mainshock. Catalogs are chosen based on their percentile in the forecasted number distribution: (a) shows the 5th percentile, (b) shows the median, (c) shows the 75th percentile, and (d) shows the 99.9th percentile catalog. Individual earthquakes are represented by red circles, and the background image shows the expected rate of **M**2.5+ earthquakes aggregated in $0.025^{\circ} \times 0.025^{\circ}$ spatial bins (i.e. the ensemble average over the simulated catalogs).

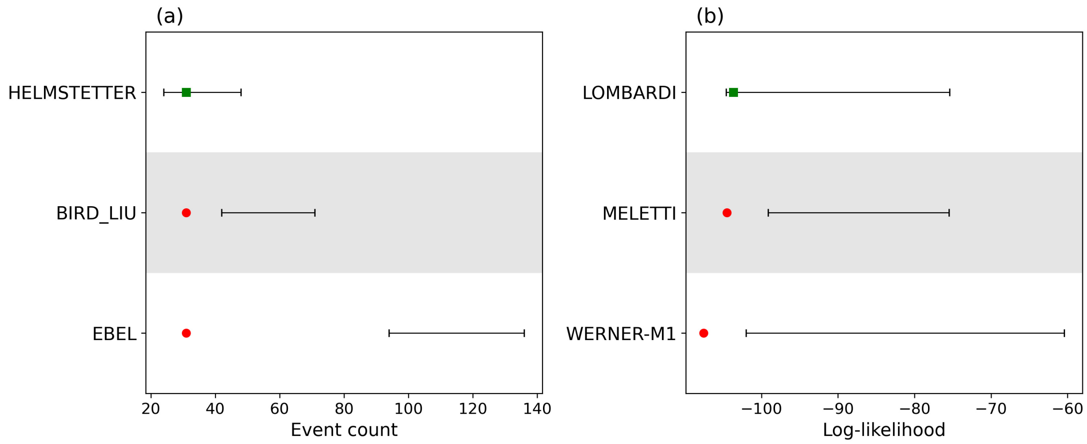

Figure 4. (a) N-test results for the HELMSTETTER (Helmstetter et al., 2007), BIRD_LIU (Bird and Liu, 2007), and EBEL (Ebel et al., 2007) earthquake forecasts for California. The markers depict the number of **M**4.95+ earthquakes during the 2006–2010 RELM evaluation period. The green square indicates that this number falls within the 95% range of the forecast number distribution (solid bar), whereas red circles indicate inconsistencies between the forecast and observations. Thus, the observed number of earthquakes is consistent with the HELMSTETTER model, whereas the BIRD_LIU and EBEL models overestimate seismicity in the region. (b) Results of the S-test for the LOMBARDI (Lombardi and Marzocchi, 2010b), MELETTI (Group, 2004) and WERNER-M1 (Werner et al., 2010) forecasts for Italy. The markers represent the spatial joint log-likelihood of each model. The green square indicates that the spatial distribution forecasted by the LOMBARDI model is consistent with the spatial distribution of observed seismicity at a 0.01 significance level. Red circles indicate that the observed locations are inconsistent with the spatial forecasts by the MELETTI and WERNER models. In both panels, significance levels of the test are chosen from the original publication of these results (i.e., Taroni et al., 2018; Zechar et al., 2013).

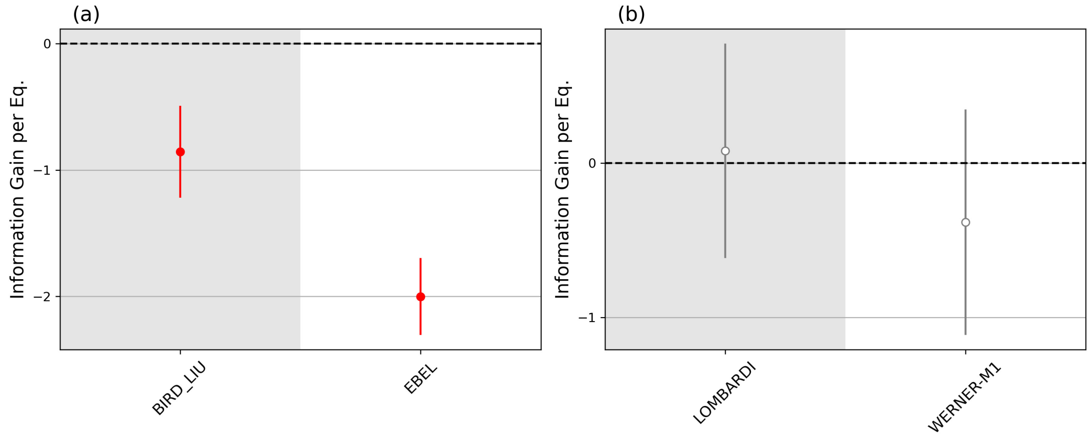

Figure 5. (a) T-test results comparing the BIRD_LIU and EBEL forecasts with the benchmark HELMSTETTER forecast (horizontal dashed line) in California. Red circles indicate information gains of the forecasts (here both negative) and vertical red bars show 95% confidence intervals. These results indicate that HELMSTETTER is significantly more informative than BIRD_LIU and EBEL during the evaluation period. (b) Results of the T-test for the LOMBARDI and WERNER forecasts with respect to the MELETTI reference forecast (horizontal dashed line), in Italy. White circles display information gains of the forecasts and vertical gray bars represent 95% confidence intervals. The horizontal dashed line falls within both confidence intervals, indicating that these models are as statistically informative as the MELETTI benchmark model.

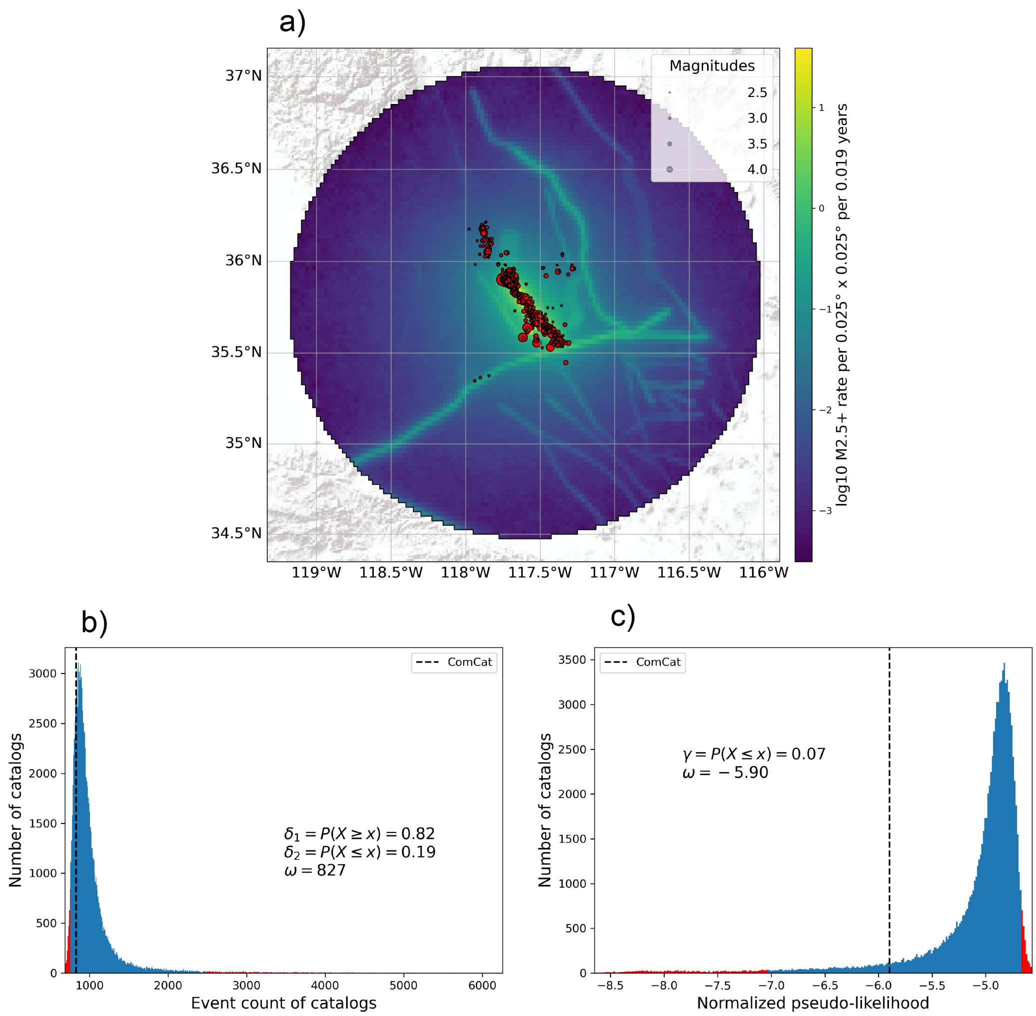

Figure 6. (a) Expected rate of **M**2.5+ events within $0.025° \times 0.025°$ spatial cells from a week-long UCERF3-ETAS forecast. The expected rates are computed by averaging over the simulated catalogs. The red circles show the locations of the 827 events observed during the forecast period. (b) N-test result for this UCERF3-ETAS forecast. The histogram shows the forecast number distribution with the two-sided 5% critical region highlighted red. The number of observed events, ω, is depicted as the dashed line, which falls near the mode of the forecast distribution, indicating consistency. δ_1 equals the fraction of catalogs in the forecast that predict at least as many earthquakes as the observations, and δ_2 denotes the fraction of catalogs that contain at most the number of observed earthquakes. (c) S-test for the UCERF3-ETAS forecast. The histogram shows the distribution of simulated scores, computed by assigning each earthquake an expected rate based on the bin-wise values in (a) and computing the geometric mean over each catalog in the forecast. The two-sided 5% critical region is highlighted red. We compute the same statistic from the observed catalog, and show the value, ω, using the dashed line. The score, γ, shows the quantile where the observed value falls in the forecast distribution.

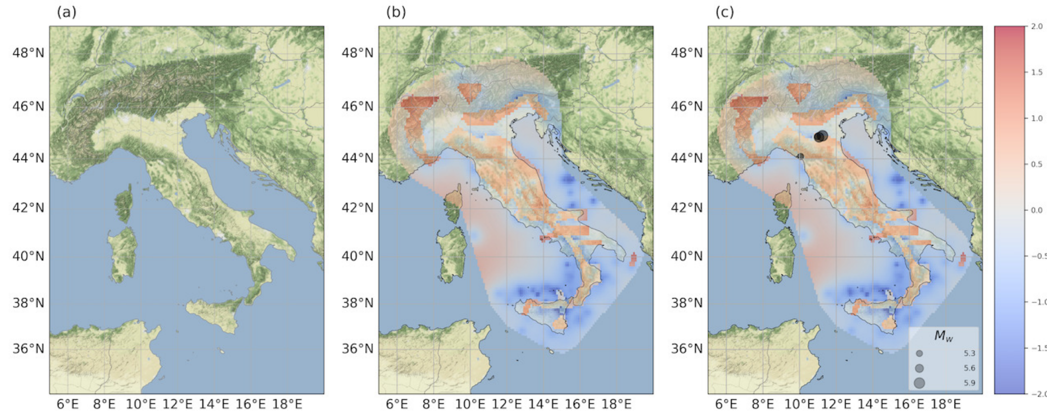

Figure 7. Example of the spatial plotting capabilities of pyCSEP through a thin-wrapper over cartopy. (a) A quick basemap can be obtained from the default plotting arguments that uses map tiles by Stamen Design (link in Data and Resources section). (b) On top of the basemap, two post-processed forecasts (the ratio between them over a given range of magnitudes). (c) The observed catalog within the same magnitude range, with auto-scaled symbols according to their magnitudes. These functions are intended to be used with pyCSEP classes and provide a simple way of visualizing spatial earthquake forecasts and catalogs.

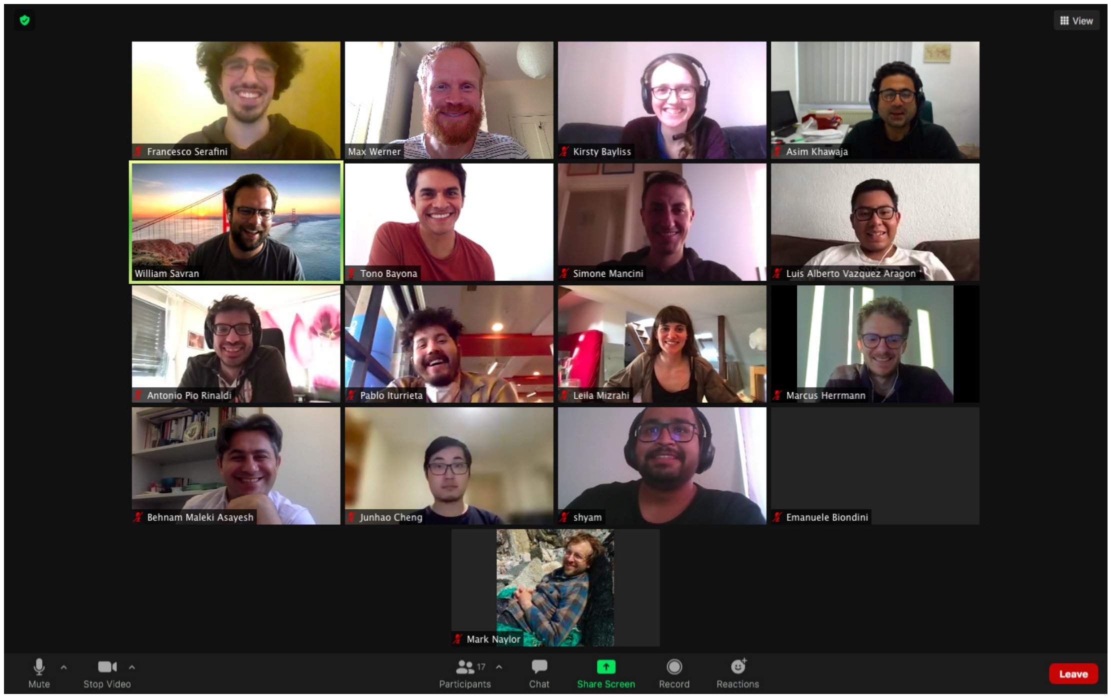

Figure 8. A screenshot of the participants at the virtual pyCSEP training workshop in March 2021 hosted by the RISE project.