Reproducible Earthquake Forecasting Experiments with pyCSEP

William Savran[1] (wsavran@usc.edu)
José Bayona[2], Maximilan Werner[2], Philip J. Maechling[1]

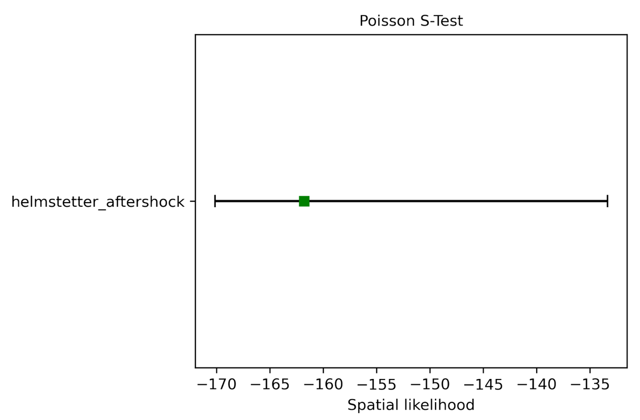
1 SCEC AN NSF+USGS CENTER
2 University of BRISTOL

Overview

An important goal of the Collaboratory for the Study of Earthquake Predictability (CSEP) is to facilitate transparent and reproducible earthquake forecasting experiments. To reproduce an experiment, we must obtain consistent results, at least statistically, when the experiment is re-run using the same data and computational methods. For over a decade, earthquake forecasting experiments were computed within CSEP testing centers, which are dedicated servers and software to compute and evaluate probabilistic earthquake forecasting models against observations. Recently, CSEP has developed pyCSEP, an open-source Python library, to provide forecast evaluation software used in the CSEP testing centers to the research community. pyCSEP enables users to conduct forecasting experiments on their own machines. This flexibility, however, also makes it difficult to maintain a strictly controlled computing environment for these bespoke experiments. CSEP testing centers had maintained such a controlled environment to ensure the reproducibility and unbiased nature of CSEP experiments. User-run experiments do not replace testing centers, but rather supplement them. In this work, we show how transparent and reproducible forecasting experiments can be conducted using freely available tools such as git, Zenodo, and Docker. We demonstrate this approach by reproducing results from the Regional Earthquake Likelihood Model (RELM) forecasting experiment published by Zechar et al. (2013). We reproduce summary statistics of the forecast and CSEP consistency tests for forecasts in the mainshock+aftershock class, as defined by Zechar et al. (2013). By substituting transparency for a controlled environment, we can uphold the reproducibility requirements for CSEP experiments. This new approach has advantages, namely that all research artifacts and code are made publicly available rather than stored on managed CSEP servers. In addition, external entities manage and store the data and meta-data. The focus on reproducibility and data management in this approach should be standard practice for future CSEP experiments, and should guide future CSEP infrastructure developments.

pyCSEP example: Gridded Forecast Evaluation

```python
import csep
from csep.core import poisson_evaluations as poisson
from csep.utils import datasets, time_utils, plots
# Model parameters
start_date = time_utils.strptime_to_utc_datetime('2006-11-12 00:00:00.0')
end_date = time_utils.strptime_to_utc_datetime('2011-11-12 00:00:00.0')
# Load Helmstetter forecast
forecast = csep.load_gridded_forecast(datasets.helmstetter_aftershock_fname,
                                      start_date=start_date,
                                      end_date=end_date,
                                      name='helmstetter_aftershock')
# Plot forecast
ax = forecast.plot(plot_args = {
    'title': forecast.name,
    'grid_labels': True,
    'borders': True,
    'feature_lw': 0.5,
    'region_border': False,
    'basemap': 'ESRI_terrain'
})
# Query Comcat for evaluation catalog
catalog = csep.query_comcat(forecast.start_time, forecast.end_time,
                            min_magnitude=forecast.min_magnitude,
                            name='comcat')
# Region is automatically parsed from forecast file
catalog = catalog.filter_spatial(forecast.region)
# Evaluate Forecast
spatial_test_result = poisson.spatial_test(forecast, catalog)
ax = plots.plot_poisson_consistency_test(spatial_test_result,
                                         plot_args={'xlabel': 'Spatial likelihood'})
```



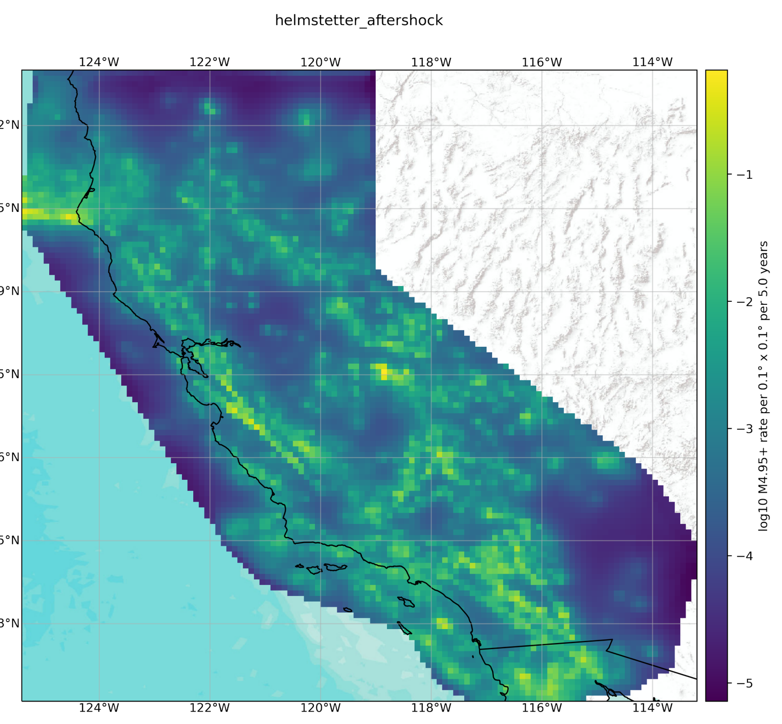


Figure 1 (left). S(patial)-test result for Helmsetter smoothed seismicity forecast. Evaluation period from 12 Nov. 2006 - 12 Nov. 2011 and contained 34 events with **M** ≥ 4.95. This test evaluates the spatial component of the forecast agains data.

Figure 2 (right). Spatial plot of the Helmstetter smoothed seismicity forecast for a 5 year period between 12 Nov. 2006 and 12 Nov. 2011. The yellow areas indicate higher forecasted rates. This model can be considered a benchmark time-independent forecast for the California region.

About pyCSEP

- CSEP has **released** a **software toolkit** through standard Python distribution channels **PyPI** and **conda-forge**

- pyCSEP supports evaluating both **gridded** and **catalog-based** earthquake forecasts

- Includes **new evaluations** for **catalog-based** earthquake forecasts

- **Modular** design allows users to develop **complex workflows** associated with forecast development and testing

- **Reproducibility packages** should accompany publications of earthquake forecasting experiments using **pyCSEP**

Try out the reproducibility package for this poster
https://github.com/wsavran/relm_pycsep_reproducibility

Getting Started with PyCSEP

Try it out: Installing PyCSEP is as simple as: `conda install --channel conda-forge pycsep`

In-depth instructions can be found at https://docs.cseptesting.org/getting_started/installing.html

Documentation: Read the documentation at https://docs.cseptesting.org
Try out some examples: https://docs.cseptesting.org/tutorials/
Report any issues: https://github.com/SCECcode/pycsep/issues
Help contribute to the project: https://github.com/SCECcode/pycsep

CSEP COLLABORATORY FOR THE STUDY OF EARTHQUAKE PREDICTIBILITY

Reproducibility Packages for Earthquake Forecasting Experiments

Reproducibility packages contain the digital artifacts required to reproduce a scientific result (Krafzyk et al., 2021).

Proposed structure Krafzyk et al. (2021)

Project Directory

```
README
LICENSE
run_all.sh
Dockerfile
expected_output
computational_effort.md
```

Suggested tools:

docker

GitHub

zenodo

RELM Experiment using pyCSEP

- We attempt to reproduce the results for the mainshock+aftershock class from Zechar et al. (2013) using data provided in the Electronic Supplement of the manuscript.

- We follow the methods described in the paper, but provide a new implementation of the software.

- We are unable to obtain bitwise reproducibility due to the random nature of the calculations and rounding of results in the manuscript. The scientific conclusions remain unchanged.

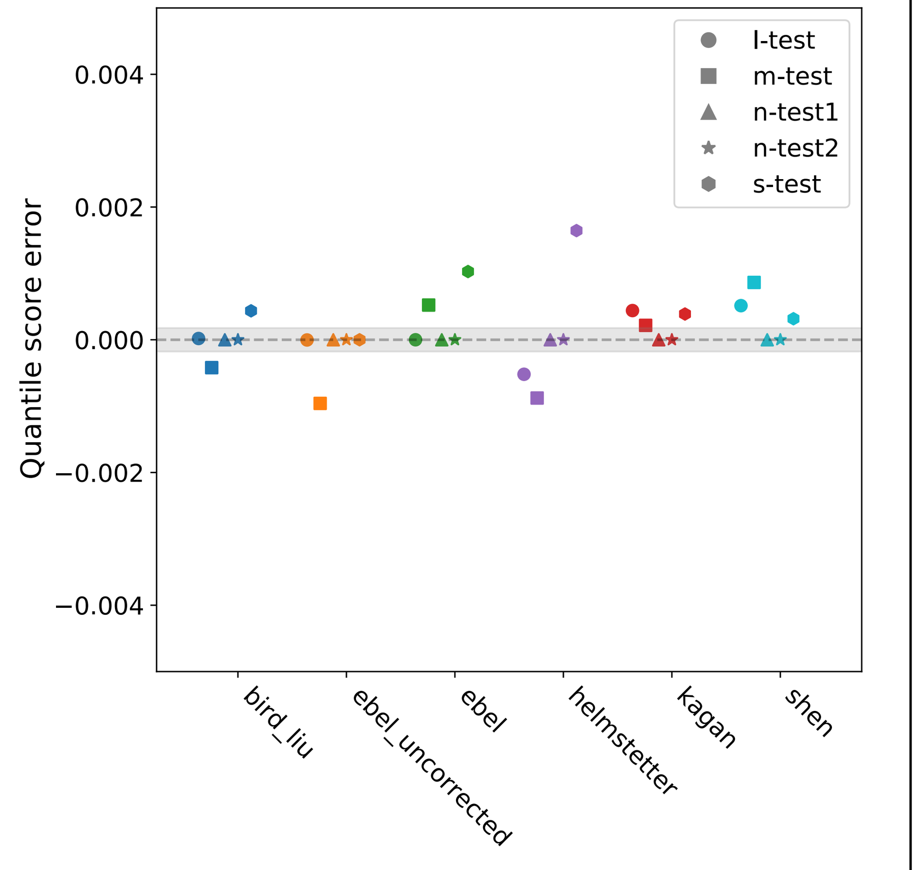


Figure 3. Difference in quantile scores from CSEP consistency test scores computed using pyCSEP and those provided by Zechar et al. (2013). The error is reported as the signed difference between the two scores. We show comparisons for the L-Test; M-Test; N-Test; and S-Test.

References

Helmstetter, A., Y. Y. Kagan, and D. D. Jackson (2007). High-resolution time-independent grid-based forecast for M≥ 5 earthquakes in California, Seismological Research Letters 78 78-86.

Krafczyk, M. S., A. Shi, A. Bhaskar, D. Marinov, and V. Stodden (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.

Zechar, J. D., D. Schorlemmer, M. J. Werner, M. C. Gerstenberger, D. A. Rhoades, and T. H. Jordan (2013). Regional Earthquake Likelihood Models I: First-Order Results, Bulletin of the Seismological Society of America 103 787-798.