DELIVERABLE

D24.1 Open-source toolbox for seismicity analysis

Work package	WP24, JRA2: Characterizing the activity rates of induced and natural earth-quakes
Lead	GFZ-Potsdam
Authors	Danijel Schorlemmer, GFZ-Potsdam; Fabrice Cotton, GFZ-Potsdam; Stefan Wiemer, ETH Zurich
Reviewers	-
Approval	Management Board
Status	Final
Dissemination level	Public
Delivery deadline	30.04.2019
Submission date	30.04.2019
Intranet path	DOCUMENTS/DELIVERABLES/ SERA_D24.1_Toolbox_for_seismicity_analysis

Table of Contents

Summary

Two seismicity toolboxes, ZMAP 7.0 and PyMap, were developed. ZMAP 7.0 is based on recent MATLAB versions while PyMap is a Python library. ZMAP 7.0 resembles the functionality of previous ZMAP versions and is updated to run with newer MATLAB versions. PyMap is a new development and its framework include abstraction layers for coordinate systems (earthquakes, mining events, laboratory acoustic emissions) and for data grids (Cartesian, triangular and hexagonal spherical tessellation). The framework is designed such that all codes using the abstraction layer can be agnostic about the coordinate system and the grid. The framework and first basic functions are accompanied by a full 4D-viewer that allows local to global visualizations of earthquake catalogs, earthquake data, coastlines, topography, and data grids. PyMap development is continued in collaboration with the Collaboratory for the Study of Earthquake Predictability and the pyrocko open-source toolbox.

1 ZMAP 7.0

We developed a new version of the ZMAP toolbox published many years ago (Wiemer, 2001). This new Version 7 is compatible with versions of MATLAB from 2018 onward. Current ZMAP developments facilitate the implementation and transfer of existing modules to PyMap and allow crosschecking of PyMap and ZMAP tools.

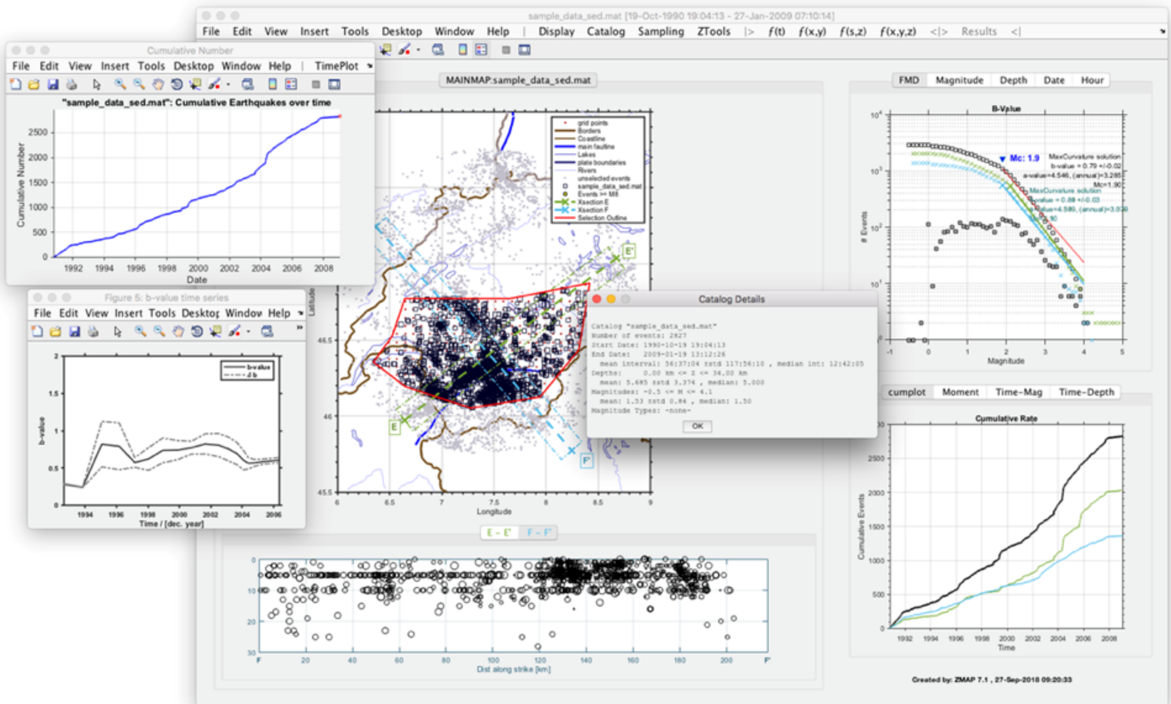

Figure 1. Sample ZMAP 7 interface

Earthquake catalogs are probably the most fundamental products of seismology and remain arguably the most useful for tectonic studies. Modern seismograph networks can locate upwards of a hundred thousand earthquakes annually, providing a continuous and sometime overwhelming stream of earthquake locations. ZMAP is a set of tools driven by a graphical user interface (GUI), designed to help seismologists analyze catalog data. ZMAP is primarily a research tool suited to the evaluation of catalog quality and to addressing specific hypotheses; however, it can also be useful in routine network operations. ZMAP was first published in 1994, with the last major release, version 6.0, in 2001.

ZMAP 7 depends upon MathWorks MATLAB® R2018a or higher, and will work on Windows, MacOSX and Linux operating systems. Additionally, the following MATLAB toolboxes must be installed:

- Mapping Toolbox
- Statistics and Machine Learning Toolbox
- Parallel Computing Toolbox [optional, enables parallel computing]

ZMAP is currently hosted on GitHub at: https://github.com/CelsoReyes/zmap7, with links to the download available from the main SED website http://www.seismo.ethz.ch/en/research-and-teaching/products-software/software/ZMAP/.

ZMAP 7 represents a major reworking of ZMAP 6.0. Every aspect of ZMAP has been modified—from the user interface through the data representations within the program—according the following goals:

- Make ZMAP compatible with modern MATLAB installations. MATLAB has evolved far beyond the version for which ZMAP 6.0 was designed and in several cases broke backwards compatibility. In the intervening years, new, robust techniques for performing object oriented design have driven changes to the graphics system underpinning the user interface, as well as to the language itself.
- Make it easier to add additional functionality. By leveraging functions and classes, future users inherit a consistent interface that allows the easy addition powerful analysis routines with very little code duplication.
- Make the user interfaces more consistent and interactive. Frequently recurring user-interfaces (e.g. dialog boxes) were once generated at a low level in each routine and differed wildly between routines. Now, the most common of these have been consolidated and a method for consistently generating them has been added.
- Make code more robust. Originally, ZMAP code consisted of a large selection of scripts that operated on global variables and made assumptions about the GUI's state. Callbacks, a primary component of GUIs, were string-based scripts that were "invisible" to the MATLAB syntax checker. Now, scripts have been extracted into individual functions, allowing for better code reuse and allowing the languages validation tools to efficiently function.
- Make existing code more readable and maintainable. This has involved reducing code duplication through the use of consolidated helper functions and classes. Home-grown functionality has been removed in favor of standard toolbox functions, further reducing the need for maintenance. All entities (classes, functions, and variables) are being renamed to reduce the cognitive burden of maintainers to follow.
- Simplify access to event catalogues. ZMAP can load catalog data directly from FDSN Event Webservice web sites in addition to a variety of file formats or local variables.

While optimizing for speed was not a goal in itself, the previous points all contribute to the ease of finding and fixing inefficiencies resulting from naïve algorithm choice or inefficient porting from other languages (esp. FORTRAN). Additionally, several algorithms may take advantage of parallel processing capabilities.

ZMAP 7 is currently in alpha release stage while its functionality continues to evolve. However, the basic functionality is in place and already allows one to easily explore earthquake catalogs. When users encounter bugs or user-unfriendly behavior, they are encouraged to report them to the ZMAP developer(s) via the GitHub issue reporting system, conveniently accessible from within the ZMAP help menu. These issues are visible by both the program maintainer(s) and the community and becomes a touchstone for understanding which aspects of ZMAP are important to the community.

2 Jupyter Notebooks

PyMap code development has started with the development of Python Jupyter Notebooks, focusing on the statistical seismology computational chain from input data (earthquake catalogs) to data quality analysis to computation of earthquake-catalog statistical parameters and results visualization. Tools to compute statistical parameters such as seismicity rates and b-values were developed as a Python library by Aiken et al. (2018). The corresponding Jupyter notebooks, which are freely available, provide tutorials for statistical seismicity computations intended for use by seismology students. Figure 2 provides a list of the notebooks and their associated learning goals, which focus on data analysis using the Pandas library and visualization of earthquake statistics such as spatiotemporal b-value variations. The notebooks were successfully applied in a Master's level PSHA course at the University of Potsdam, providing exposure to map-making and statistical analysis using Python to students with little to no programming experience.

Table 1
Learning Goals and Their Associated Jupyter Notebooks

	Learning Goal	Notebook Title
1	Python data types	Introduction to Python
		Introduction to reading data and plotting using Pandas
2	Import and export data	Introduction to reading data and plotting using Pandas
		Introduction to plotting data as a heat map
3	Basic figure making	Introduction to reading data and plotting using Pandas
		Introduction to plotting data as a heat map
		Introduction to scatter plots and histograms
4	Histograms	Introduction to scatter plots and histograms
5	Map making	Creating a map in cartopy and plotting data on it
		Plotting focal mechanisms on a cartopy map
		Plotting a pretty map using cartopy
		Plotting heat map data on a map using cartopy
		Maps for unshaped data using cartopy
6	Exploratory data analysis	Introduction to plotting data as a heat map
		Introduction to scatter plots and histograms
		Creating a map in cartopy and plotting data on it
		Plotting heat map data on a map using cartopy
7	Jupyter Notebooks	All notebooks

Instead of having a linear progression (notebook 1, notebook 2, notebook 3, etc.), we assign Notebooks to learning goals to aid in the modularity of the materials.

Figure 2: Table 1 from Aiken et al. (2018), sumarizing the main learning goals and associated Jupyter notebooks of their Python library for earthquake statistics computation and data visualization.

3 PyMap

Because PyMap is designed as a computational toolbox developed in Python for statistical seismology, we found the approach using only Jupyter notebooks to not be sufficient for the projected tasks of PyMap. In particular, we did not want to implement the different coordinate systems and the abstract grid backend into a distribution of only notebooks. Therefore, we decided to first develop a pure toolbox implementation with higher abstraction to be later used in Jupyter notebooks (as a black box).

PyMap's default input dataset are earthquake catalogs and it comprises tools for data quality analysis, statistical parameters of earthquake catalogs and provides a viewer for data and result visualization. Except for the viewer, PyMap deliberately does not provide any user-interface tools as it is meant to be a toolbox (or function library) to be used in Python scripts, Jupyter notebooks, or included in own software projects. The development in Python ensures largest possible flexibility regarding its use on different operating systems. While PyMap is developed under Linux, the toolbox functionality will work without modifications under MacOS and Windows. Because Python itself, the Python libraries (e.g. numpy, matplotlib, and PyMap are all released under an open-source license, no license fee or permission is necessary for using and further developing PyMap.

PyMap addresses a common difficulty of earthquake statistics tools (e.g. ZMAP): correctly representing all key values over a large range of orders of magnitude and operating on different coordinate systems. Earthquakes or similar events are in general represented in three different coordinate systems:

1. Earthquake hypocenter locations are in most cases represented in latitude, longitude, and depth. This classical system uses in most cases degrees for latitude and longitude and kilometers for depth. This representation is compatible with the standardized representation in QuakeML (Schorlemmer et al., 2004).
2. Small-earthquake hypocenters in mines or induced-seismicity events hypocenters in injection sites are often represented with their northing, easting, and depth relative to a fixed location.
3. Acoustic-emission locations in laboratory experiments are represented purely in relative coordinates to a fixed point of the sample (itself a relative coordinate).

PyMap unifies these there coordinate systems into a single one and provides the necessary library function to operate with the new coordinate system such that all further functions on PyMap are agnostic of the coordinate system. This approach has been chosen to avoid the common rewriting of functions to adapt to a different coordinate system. Any function written using PyMap's abstraction of the coordinate system will work on data from any of the three coordinate systems or even combinations thereof.

The newly introduced coordinate system of PyMap is called 5C (for five-element coordinates). 5C contains the five elements latitude, longitude, depth, northing offset, easting offset. The aforementioned coordinate systems are represented as

1. Earthquake hypocenters are given as latitude, longitude, depth with the offsets set to zero.
2. Events defined in relation to a fixed point use the latitude and longitude of the fixed point and store the lateral offset in the offset component. The depth represented the offset depth combined with the depth of the fixed point.
3. Acoustic-emission hypocenters use any value (always the same for each sample) for the reference latitude and longitude and store the relative location in depth and the offsets.

PyMap defines various distance measures that operate on the 5C system so that all functions of the toolbox can use this abstraction layer to ensure full operational capabilities independent of the coordinate system used in the input data. Already implemented are an Euclidian distance and the distance along a great circle. Moreover, this system allows to combine regular earthquake catalogs with catalogs of induced seismicity (according to the second type) and process and visualize them together.

1. Likewise, PyMap uses a large-range floating point variable for the focal time of the events to allow to store simultaneously thousands of years for (historic) earthquake catalogs as well as microseconds for laboratory experiments.

One of the standard features of such toolboxes like PyMap or ZMAP is the computation of the spatially varying parameters of earthquake catalogs. For simplicity, these are usually represented in a Cartesian grid. For regional analyses, the Cartesian grids are very convenient and simple to use. However, the larger the region, the greater the range of area covered by each grid cell; the closer a cell to the poles, the (relatively) smaller the cell (if defined in fractions of degrees). To accommodate the need for equal-sized grid cells on a regional to global scale, PyMap offers grids based on triangular and hexagonal spherical tessellation. Similar to the solution for the different coordinate systems, PyMap provides a grid abstraction layer that allows any function using any type of grid to be grid-agnostic. This layer allows to query whether or not a point is within a cell, on a cell boundary and if so, to which cell the boundary belongs.

For importing earthquake catalogs into PyMap, we developed an interface to the QuakePy package. QuakePy is the reference implementation for QuakeML and provides import and export filter for various earthquake catalog formats. QuakePy simplifies complex earthquake catalogs for the use in PyMap.

PyMap at the current stage of development provides basic statistics tools (e.g. earthquake activity rates, b-values, etc.). The framework is designed in such a way that further statistical functions can be easily implemented given the abstraction layers for coordinate systems and grids.

As mentioned previously, PyMap comes with one graphical user interface tool. This is the 4D-viewer Sparrow. Sparrow operates by default on the globe and allows to zoom into any area of interest. It can display earthquake catalogs color-coded) in full 3D, see Figures 3 & 4.

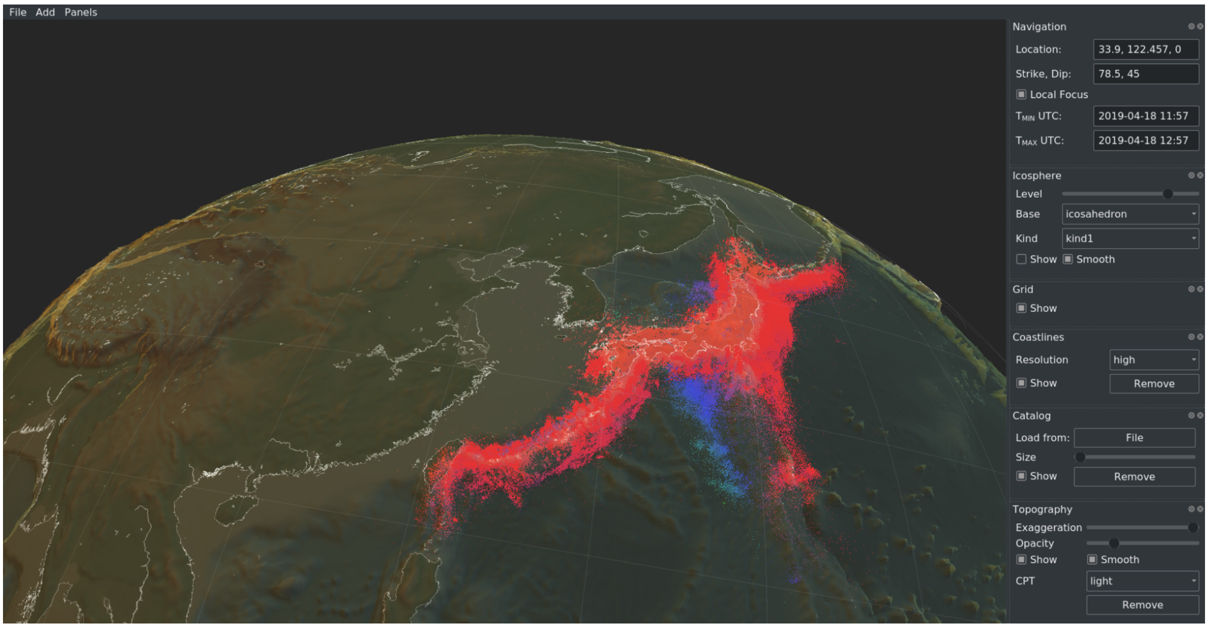

Figure 3: Sparrow display of the earthquake catalog of the Japan Meteorological Agency (JMA) together with coastlines, topography, and bathymetry.

The display can be augmented with coastlines, 3D-topography, and with seismological data, e.g. faults and source models of larger earthquakes, see Figure 5. Sparrow offers the user to find locations by latitude/longitude or by name searches (e.g. “Tokyo”).

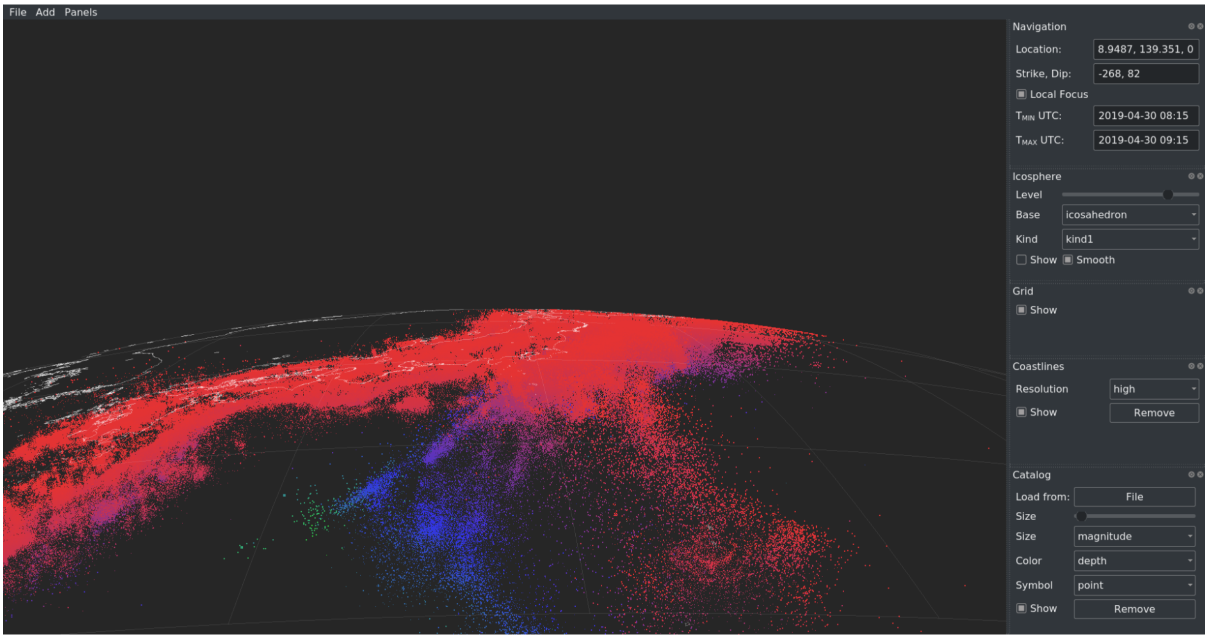

Figure 4: Details of the 3D view of the catalog of the Japan Meteorological Agency in Japan. Events are color-coded according to depths. The subducting slab is nicely visible.

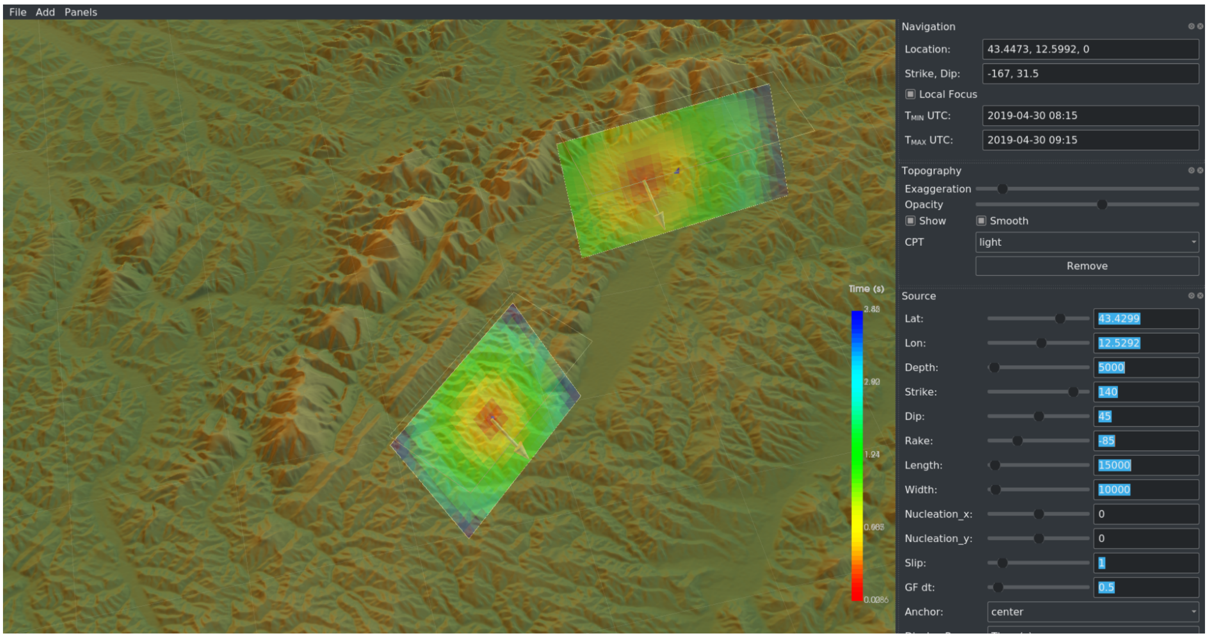

Figure 5: Sparrow display of two scenario earthquake sources. Color-coded is the rupture time . The sources are embedded into a topography display.

The fourth dimension in Sparrow is the time. Sparrow can be used to dispay the evolution of seismicity as an animation to help the user better identify features of interest.

Any computed value represented in any of the supported grid types can be displayed in Sparrow, see Figure 6. This feature, as often used in ZMAP on Cartesian grids, is one key element of PyMap. The new grid abstraction layer allows for the different suported grid types to be used for computations and to be displayed without the user having to develop the rather complex visualization procedure for the complex grids.

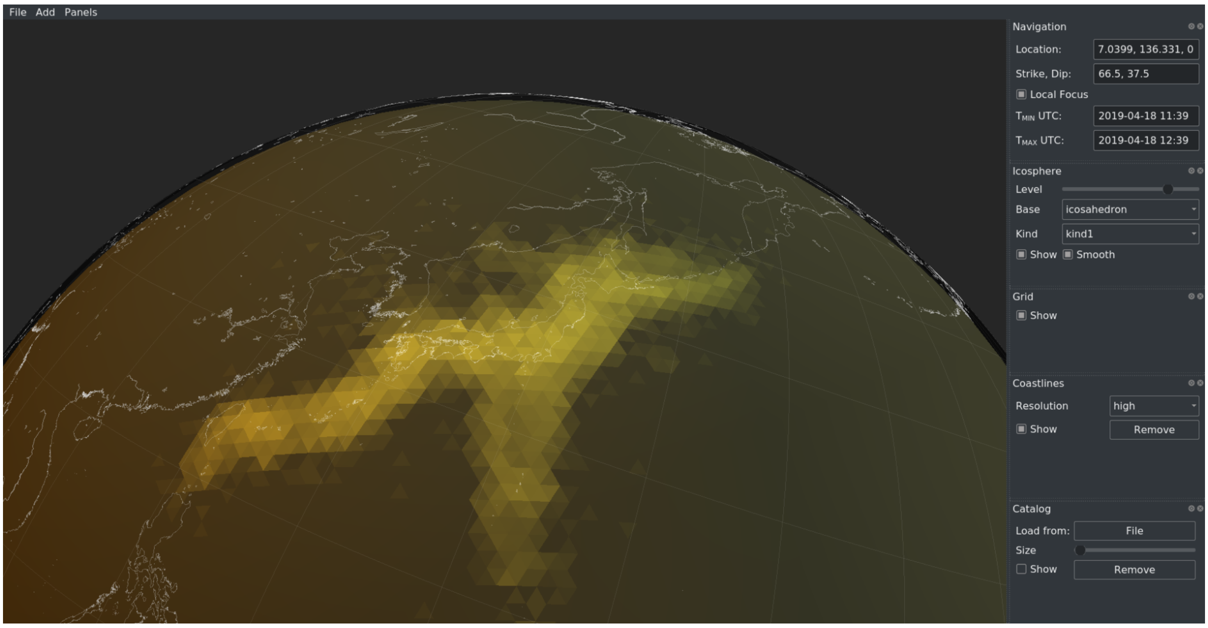

Figure 6: Sparrow display of the earthquake density of the earthquake catalog of the Japan Meteorological Agency (JMA) on a triangular grid over Japan.

Overall, the development of PyMap has not finished. The main task to develop the framework for PyMap is completed. In the near future, we will continue to fill PyMap with useful functions that will allow first users to quickly assemble statistical analyses of seismicity.

One particular development is very important. The Collaboratory for the Study of Earthquake Predictability is redesigning their earthquake forecast testing center software (Schorlemmer & Gerstenberger, 2007; Zechar et al., 2010) and, given the overlap in personnel, the PyMap and CSEP developments will be merged for all parts that are related to earthquake catalog manipulation and statistical seismology, i.e. the core part of PyMap. CSEP's system is also developed in Python and thus integration of PyMap tools will be relatively easy.

Further synergies stem from the collaboration with Pyrocko (Heimann et al., 2017), an open-source seismology toolbox and library, written in Python. Pyrocko is designed for performing a variety of geophysical tasks, like seismological data processing and analysis, modelling of InSAR, GPS data and dynamic waveforms, or for seismic source characterization. PyMap is available from the pyrocko repository. Please refer to pyrocko.org for further information about downloading pyrocko and PyMap.

4 References

Aiken, J. M., C. Aiken, and F. Cotton: A Python library for teaching computation to seismology students, Seismological Research Letters, Vol. 89, No. 3, p. 1165–1171, 2018.

Heimann, S., M. Kriegerowski, M. Isken, S. Cesca, S. Daout, F. Grigoli, C. Juretzek, T. Megies, N. Nooshiri, A. Steinberg, H. Sudhaus, H. Vasyura-Bathke, T. Willey and T. Dahm: Pyrocko - An open-source seismology toolbox and library. V. 0.3., GFZ Data Services, 2017, 10.5880/GFZ.2.1.2017.001

Schorlemmer, D., A. Wyss, S. Maraini, S. Wiemer and M. Baer: QuakeML–An XML schema for seismology. ORFEUS Newsletter, 6(2), 9, 2004.

Schorlemmer, D. and M. Gerstenberger: RELM Testing Center. Seismol. Res. Lett., 78(1), 30-36, 2007.

Wiemer, S.: A software package to analyze seismicity: ZMAP. Seismological Research Letters, 72(3), pp.373-382, 2001, 10.1785/gssrl.72.3.373

Zechar, J. D., D. Schorlemmer, M. Liukis, J. Yu, F. Euchner, P. J. Maechling and T. H. Jordan: The Collaboratory for the Study of Earthquake Predictability – perspective on computational earthquake science. Concurrency and Computation: Practice and Experience, 22(12), 1836-1847, 2010,10.1002/cpe.1519

Contact

Project lead	ETH Zürich
Project coordinator	Prof. Dr. Domenico Giardini
Project manager	Dr. Kauzar Saleh
Project office	ETH Department of Earth Sciences
	Sonneggstrasse 5, NO H62, CH-8092 Zürich
	sera_office@erdw.ethz.ch
	+41 44 632 9690
Project website	www.sera-eu.org

Liability claim

The European Commission is not responsible for any use that may be made of the information contained in this document. Also, responsibility for the information and views expressed in this document lies entirely with the author(s).