SoftwareX 32 (2025) 102339

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

FQSHA: An open-source Python software for fault-based seismic hazard assessment

Nasrin Tavakolizadeh [a,d], [*], Hamzeh Mohammadigheymasi [a,b,c], Nuno Pombo [a,d]

[a] Department of Computer Sciences, University of Beira Interior, Portugal
[b] Department of Earth and Planetary Sciences, Harvard University, United States
[c] Atmosphere and Ocean Research Institute (AORI), The University of Tokyo, Japan
[d] Instituto de Telecomunicações (IT), Portugal

ARTICLE INFO

Dataset link: https://github.com/GeoSignalAnalysis/FQSHA

Keywords:
Graphical user interface
Seismic activity rate
Fault-based seismic hazard assessment

ABSTRACT

The PyQt framework facilitates the development of desktop applications, offering an effective environment for implementing scientific algorithms while leveraging the flexibility of Python. In this paper, we introduce FQSHA, a PyQt5-based application designed to streamline the workflow of fault-based Seismic Activity Rate (SAR) calculation and seismic hazard assessment. The primary aim is to provide a unified and user-friendly interface that makes these processes more accessible. FQSHA enables users to perform hazard calculations and map the results from scratch using minimal input data and requiring only basic knowledge of hazard computation engines. The SAR calculation core (FaultQuake) is seamlessly integrated with the hazard assessment engine (OpenQuake), offering additional flexibility to customize input parameters for hazard analysis. This integration supports an end-to-end workflow within a single software utilizing a user-friendly GUI. We present the architecture of FQSHA and demonstrate its capabilities through a hands-on example, which is publicly available on GitHub.

Code metadata

Current code version	gv01
Permanent link to Reproducible Capsule	https://github.com/GeoSignalAnalysis/FQSHA
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-25-00271
Legal Code License	GNU Affero General Public License v3.0
Code versioning system used	Git
Software code languages, tools, and services used	Python v3.11
Compilation requirements, operating environments & dependencies	Microsoft Windows, Linux
If available Link to developer documentation/manual	https://github.com/GeoSignalAnalysis/FQSHA/blob/main/Documentation.pdf
Support email for questions	n.tavakolizadeh@ubi.pt

1. Motivation and significance

Effective fault-based PSHA requires advanced models to incorporate evolving geometrical and kinematic fault parameters for accurately estimating the faults' activity rates and corresponding uncertainties. On the other hand, with the continuous advancement of active fault databases—through geodetic observations, earthquake detection, and location by deep learning [1–4]—and improved knowledge of slip rates and geometrical attributes of faults, there is a growing need for more efficient fault-based hazard modeling tools to keep pace with ongoing updates to these datasets. In this regard, there has been extensive research in this field, resulting in the development of various frameworks and tools.

Frameworks are being improved to better incorporate fault hazard models into PSHA. In relation to this, OpenSHA [5] was introduced as an extensible framework for seismic hazard analysis and has been pivotal in implementing complex models such as UCERF3 [6], which

[*] Corresponding author at: Department of Computer Sciences, University of Beira Interior, Portugal.
E-mail address: n.tavakolizadeh@ubi.pt (N. Tavakolizadeh).

https://doi.org/10.1016/j.softx.2025.102339
Received 23 April 2025; Received in revised form 25 August 2025; Accepted 27 August 2025
Available online 16 September 2025

integrates geological, geodetic, and seismic data to generate time-independent earthquake rupture forecasts for California. While OpenSHA offers a robust platform for seismic hazard analysis, it has certain limitations: it requires high-quality and extensive input data, limiting its applicability in data-scarce regions; it involves computationally intensive models that demand expert configuration; and its Java-based implementation blackfaces compatibility with modern Python-based workflows.

More recently, the ESHM20 framework presented by [7] introduced the 2020 European Seismic Hazard Model (ESHM20). This model presents a hybrid framework that integrates fault-based sources with smoothed background seismicity across the Euro-Mediterranean region. The model incorporates the European Fault-Source Model 2020 (EFSM20), comprising 1248 crustal faults and four subduction systems, and applies a deterministic approach to estimate seismic activity rates from geological or geodetic slip rates via moment conservation, without statistical fitting. Faults are modeled as unsegmented composite sources capable of generating ruptures anywhere along their surface. Seismicity in areas without mapped faults is represented using adaptive kernel smoothing within tectonic zones, with activity parameters derived from Gutenberg–Richter distributions. The model is implemented using the OpenQuake-engine, leveraging its logic-tree capabilities to propagate epistemic uncertainties. This study highlights the usefulness of OpenQuake's fault modeling for seismic hazard assessment, especially in contexts where input data is available but limited. It also underscores the importance of SAR estimation for each fault source as a critical step in hazard modeling.

Several tools have been developed to facilitate fault-based PSHA by converting geological and kinematic fault parameters into seismic activity rates suitable for hazard modeling. FiSH [8] estimates earthquake recurrence parameters using fault slip rates and scaling relationships, producing time-independent seismicity rates as input for hazard models. SUNFiSH and FRESH [9] extend this approach by incorporating logic-tree structures to explore epistemic uncertainties in fault parameters such as slip rate, maximum magnitude, and recurrence model. These tools allow for multiple recurrence models, including characteristic and truncated Gutenberg–Richter distributions. SHERIFS [10] enables the construction of fault-based earthquake rupture forecasts by defining possible rupture geometries, recurrence parameters, and spatial distributions, with support for time-dependent models. While these tools efficiently generate source models for integration into platforms like OpenQuake, they typically stop at the activity rate or source model generation stage and require separate workflows for hazard computation and visualization.

The most recent tool, FaultQuake [11], incorporates SCC into the SAR calculation process. A significant feature of this software is its ability to operate on a simple dataset including geometrical specifications, slip rates, and observed earthquakes to compute SAR. It also distinguishes between seismic and aseismic deformation, a critical factor in understanding the behavior of active faults [12]. Although this tool performs SAR calculation, it is not a comprehensive solution for hazard calculation or hazard mapping.

On the hazard calculation side, the OpenQuake engine [13], promoted by the Global Earthquake Model (GEM) Foundation, provides a platform that facilitates fault-based hazard modeling and serves as a multi-purpose tool for seismic hazard assessment. However, OpenQuake does not compute seismic activity rates; instead, it requires the user to supply the seismic moment rate or MFD for each fault source.

These platforms often require users to be proficient in compiling at least two different software packages written in various programming languages and handling different input formats (e.g., .txt, XML, .ini files). Additionally, mapping the output PoE and visualizing the hazard distribution along fault traces, as well as comparing results with fault's mapped trace, are final steps that can be challenging and time-consuming—especially for students, educators, and researchers.

Fault-based hazard maps are typically developed in three steps: (1) compiling fault datasets and calculating activity rates, (2) preparing inputs for hazard engines like OpenQuake, and (3) mapping the resulting hazard to evaluate the hazard potential of each fault using tools like GMT, or QGIS. The motivation for developing our integrated workflow, FQSHA, which includes a graphical user interface (GUI), stems from the need for a versatile and open-source tool that consolidates all these steps. FQSHA processes basic input data in JSON format, including fault geometry, slip rates, and historical seismicity, to calculate SAR, generate necessary hazard parameters, integrate them with OpenQuake, and visualize fault-based hazard maps. It is designed to enable rapid testing of alternative fault segmentation and moment-rate scenarios. The GUI provides intuitive access to all features, facilitating rapid exploration and scenario testing for both experienced researchers and new users in seismic hazard analysis.

2. Software description

This section presents the theoretical background and structural design of the FQSHA tool. The significance of the FQSHA tool in the context of current literature lies in its ability to unify the workflow for fault-based seismic hazard analysis. Its main contributions include:

- FQSHA eliminates the need for time-consuming tasks such as preparing and analyzing inputs for various tools and packages by requiring only a single JSON file containing fault specifications, thereby removing the burden of manually handling complex input formats.
- The automated handling of XML and JSON files streamlines data processing and minimizes errors, thereby enhancing the efficiency of fault-based seismic modeling.
- Integrating SAR computation directly within the framework, removing the dependency on a software to calculate seismic activity rates and exporting the Activity Rate (AR)s to be compatible by another software.
- Enhancing accessibility to fault-based hazard analysis by enabling direct computation of seismic hazard from fault models within a single environment.
- Providing built-in functionality to generate fault-specific hazard maps, thereby accelerating the evaluation of multiple hazard scenarios.

2.1. Software functionality

The core functionality of this Python-based graphical interface lies in the integration of fault SAR modeling (based on the methodology introduced by [11]) with the OpenQuake hazard computation engine [13], allowing for the direct generation of fault-based seismic hazard maps (Fig. 1).

The primary objective of this method to realistically incorporate the physical characteristics of active faults into hazard models. This involves leveraging available geological and geophysical data, such as fault geometry (length, dip, seismogenic thickness), slip rates (minimum and maximum), and earthquake history (observed magnitude (M_{obs}) and its associated Standard Deviation (SD), to estimate the likelihood and impact of potential earthquakes. In this context, FQSHA utilizes an advanced SAR calculation engine that also accepts optional inputs such as the Seismic Coupling Coefficient (SCC) to partition the total deformation into seismic and aseismic components (see Table 4). These geometrical and kinematic attributes are used to derive five different Empirical Magnitude (EM) estimates. Through Conflation of Probabilities (CoP) that also incorporates M_{obs}, the maximum expected earthquake magnitude (Maximum Magnitude (M_{max})) and its uncertainty (σ) are computed. The total moment rate is calculated by integrating $M_{\max}$, σ, and the mean recurrence interval T_{mean}, which is

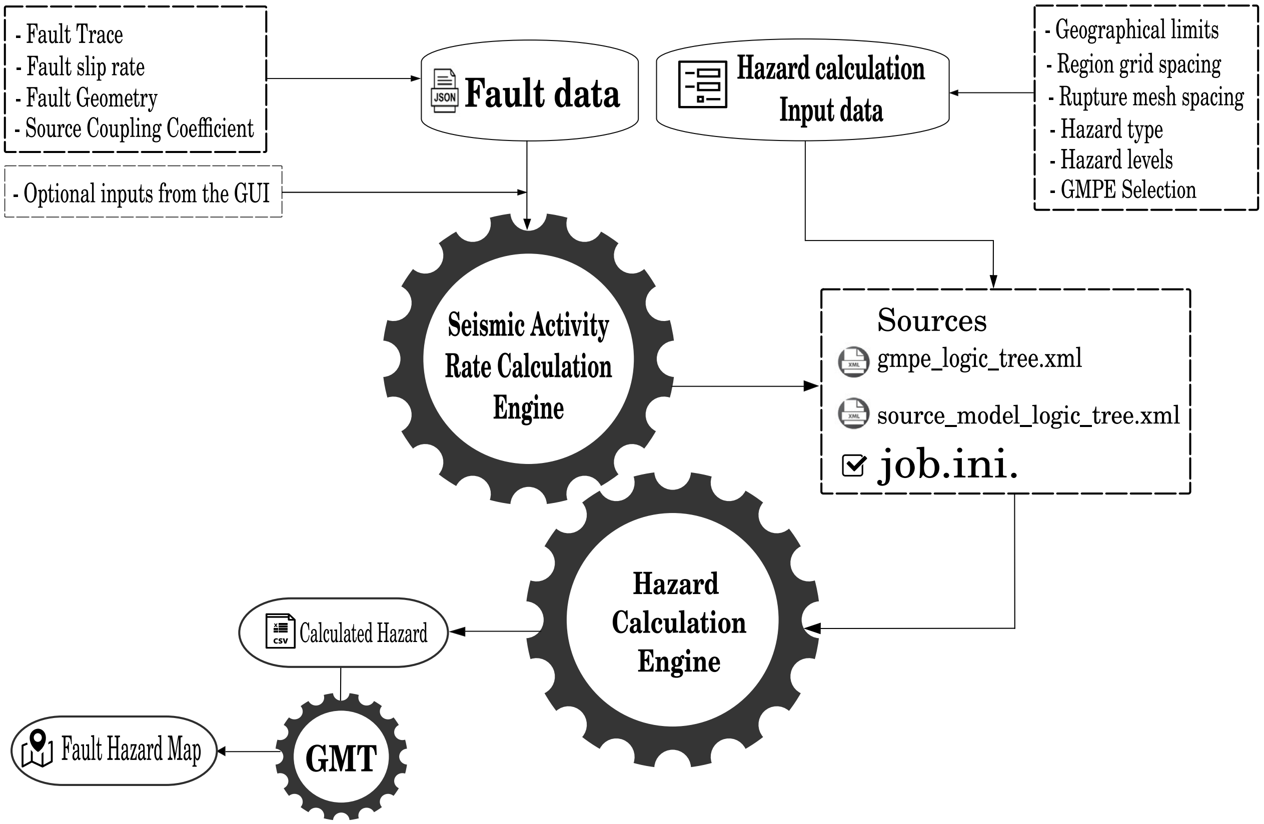


Fig. 1. FQSHA features and functionalities are presented in a schematic display framework. As illustrated, the tool integrates three core engines for Seismic Activity Rate (SAR) and Hazard calculation, and GMT mapping toolkit.

derived using the segment seismic moment conservation principle [14]. This moment rate is then balanced across a range of magnitudes between M_{min} and $M_{\max}$, using relative activity models such as the Truncated Gutenberg–Richter (Truncated Gutenberg-Richter (TGR)) or Characteristic Gaussian Distribution (Characteristic Gaussian Distribution (CGD)) models. The result is a complete SAR distribution for each fault, which forms the basis for subsequent hazard computations. Further methodological details are available in [11].

To perform the seismic hazard analysis, FQSHA interfaces directly with the OpenQuake engine. Fault-based PSHA in OpenQuake is performed by explicitly modeling individual seismogenic fault sources using their geometric, kinematic, and history of earthquakes. Each fault is defined as a complex polygonal surface or a simple planar geometry with associated parameters such as slip rate, rake, dip, and seismogenic depth range. Rather than estimating seismic activity rates from earthquake catalogs, OpenQuake requires the user to provide fault-specific moment rates or magnitude–frequency distributions (MFDs), often derived from geological or geodetic slip rates under the assumption of moment conservation [13,15]. Fault ruptures are generated stochastically across the fault surface according to empirical magnitude-area scaling relationships and user-defined rupture scenarios [16,17]. These ruptures serve as inputs to ground motion prediction equations (GMPEs) within a logic-tree framework that enables systematic propagation of epistemic uncertainties. OpenQuake then computes hazard curves, maps, and uniform hazard spectra by aggregating the PoE at defined site locations.

2.2. Fault and hazard input data

The input data requirements necessary to initiate the software's processing workflow are a primary input file containing a comprehensive set of mandatory parameters, as well as optional inputs that may be provided through the GUI. After computing the AR of the faults, the software automatically generates the essential files required for the hazard calculation step. All input data must be prepared in `JSON` format, with the following attributes specified for each fault: `ScR` (Scale Relationship code; see Table 1), `year_for_calculations`, `Length` (in kilometers), `Dip` (in degrees), `upperSeismoDepth` and `lowerSeismoDepth` (upper and lower seismogenic depths, in kilometers), `SRmin` and `SRmax` (minimum and maximum slip rates from geologic or geodetic sources), `Mobs` (maximum observed magnitude), `sdMobs` (standard deviation of `Mobs`), `Last_eq_time` (year of occurrence of `Mobs`), `SCC` (Seismic Coupling Coefficient), `ShearModulus`, `StrainDrop`, `Mmin` (minimum magnitude threshold for inclusion in hazard calculations), `b-value`, and `fault_trace` (coordinates representing the mapped trace of the fault). The structure of this input format is discussed in Section 3.

Table 1
Scale Relationship Code (ScR) options incorporated and should be specified in the input file.

Scale relationship code	Fault type and Reference
WC94-N	Normal faults [18]
WC94-R	Reverse faults [18]
WC94-S	Strike-slip faults [18]
WC94-A	All kinematics [18]
Le10-D	Dip slip faults [17]
Le10-S	Strike-slip faults [17]
Le10-SCR	Stable continental regions [17]
Volc	All kinematics in volcanic context [19,20]

OpenQuake supports two primary modes for fault-based PSHA: classical and event-based, which can be selected via the hazard calculation panel of the GUI. In the classical mode, the software computes ground motion levels corresponding to specified probabilities of exceedance (PoE) by integrating over all possible earthquake ruptures

Table 2
Recognized magnitude scaling relationships in OpenQuake for fault-based PSHA, as specified via the GUI. `FQSHA` uses `WC1994` as the default scaling relationship.

Scaling relationship code	Meaning
`WC1994`	Global empirical scaling for rupture dimensions [18]
`Leonard2010`	Crustal fault scaling for active regions [17]
`Leonard2014`	Updated scaling for stable continental crust [21]
`Strasser2010`	Subduction interface rupture scaling [22]
`Shaw2002`	Blind thrust fault rupture model [23]
`Thingbaijam2017`	Global scaling from a modern rupture database [24]

Table 3
GMPE options available in the GUI and recognized by OpenQuake, with corresponding descriptions and references.

GMPE code	Meaning and Reference
`AbrahamsonEtAl2014`	NGA-West2 for active crustal regions [25]
`BooreEtAl2014`	NGA-West2, ground motion from shallow crustal events [26]
`CampbellBozorgnia2014`	NGA-West2 for shallow crustal faults [27]
`ChiouYoungs2014`	NGA-West2 for active tectonic regions [28]
`YoungsEtAl1997SSlab`	Subduction slab events GMPE [29]
`Atkinson2015`	Eastern North America GMPE [30]
`AtkinsonBoore2003SInter`	Subduction interface GMPE [31]
`AtkinsonBoore2006`	Crustal events in Eastern North America [32]
`AbrahamsonEtAl2018SInter`	Global subduction interface GMPE [33]
`ToroEtAl1997`	Stable continental regions in Central/Eastern North America [34]
`AkkarBommer2010`	European and Middle Eastern shallow crustal events [35]
`AbrahamsonEtAl2015SSlab`	Subduction slab GMPE [36]
`AkkarEtAlRepi2014`	European GMPE with Repi distance [35]
`BooreAtkinson2008`	NGA model for shallow crustal earthquakes [37]
`CampbellBozorgnia2008`	NGA model, shallow crustal active regions [38]
`ChiouYoungs2008`	NGA model for active tectonic crustal earthquakes [39]
`ZhaoEtAl2016Asc`	Japanese strong motion interface/slab events [40]
`Idriss2014`	NGA-West2, modified for rock sites [41]
`McVerryEtAl2006`	New Zealand-specific crustal and subduction events [42]
`Bradley2010`	New Zealand GMPE for crustal events [43]
`SilvaEtAl2002`	Eastern US crustal events [44]
`AkkarEtAlRjb2014`	European GMPE with Rjb distance [45]

and their associated annual occurrence rates. For fault-based models, these ruptures are derived from fault geometry, slip rates, and magnitude–frequency distributions, with ground motions estimated using the selected Ground Motion Prediction Equations (GMPE). The outputs typically include hazard curves, hazard maps, and uniform hazard spectra. In contrast, the event-based mode employs a Monte Carlo simulation approach to generate thousands of synthetic earthquake catalogs over a user-defined time window. Each event is stochastically sampled from the fault model, and corresponding ground motion fields are calculated at the defined sites of interest. The spatial extent of the hazard calculation region is defined by the user through the hazard panel, where the coordinates of the four bounding corners are specified. This region constrains the faults to be included in the simulation. Alternatively, if left unspecified, the extent is automatically computed based on the traces of the input faults. Additional user-defined parameters include the spacing between grid points (grid resolution), the shear-wave velocity (V_{s30}), and the scaling relationship code used for rupture generation (see Table 2). By default, the software uses the `WC1994` scaling relationship [18] for hazard calculations.

In the OpenQuake Engine hazard workflow, GMPEs are essential components that quantify the relationship between earthquake source characteristics (e.g., magnitude, distance, faulting mechanism) and site conditions (e.g., shear-wave velocity, V_{s30}) with ground motion intensity measures such as PGA and Spectral Acceleration (SA). These models are incorporated into the hazard calculation via logic trees, enabling the inclusion of multiple GMPEs with user-defined weights to account for epistemic uncertainty. Each GMPE is linked to a specific tectonic region type (e.g., active shallow crust, subduction interface), and OpenQuake supports the combination of different models to compute hazard curves, hazard maps, and uniform hazard spectra. The selection and weighting of GMPEs play a critical role in shaping hazard estimates and should be tailored to the regional seismotectonic setting. The tool provides a library of 22 GMPE options (Table 3) that users can configure and weight through the GUI. Based on the selected models and weights, the software automatically generates the required logic tree files for subsequent hazard computations.

2.3. Software architecture

The algorithm in 1, together with the conceptual diagram in Fig. 1, outlines the sequential workflow of the FQSHA software, including its inputs, outputs, and the core functions invoked at each stage. These core computational routines implement widely accepted models in fault-based seismic hazard analysis.

Input data and optional parameters provided via the GUI are used to compute $M_{\max}$ and $\sigma_{M_{\max}}$ and to address epistemic uncertainty in magnitude estimates, the `conflate_pdfs()` function combines multiple probability density functions (PDFs) into a single conflated distribution using a normalized geometric mean approach. T_{mean} is estimated from the seismic moment rate and the expected moment release per event, based on the derived $M_{\max}$ and associated rupture parameters. For the moment budget calculation the `kin2coeff()` and `coeff2mag()` functions retrieve and apply empirical magnitude scaling relationships, based on fault rupture length, area, and aspect ratio [17,18,21].

The algorithm 2 details the implementation of seismic activity rate (SAR) modules within FQSHA. The subsequent step utilizes the `sactivityrate` module to calculate the total moment and redistribute it across the magnitude range between $M_{\min}$ and $M_{\max}$ for SAR computation. For modeling seismic activity rates, the software supports both characteristic and Gutenberg–Richter formulations. The `CHGaussPoiss()` and `CHGaussBPT()` functions implement a characteristic Gaussian magnitude model paired with time-independent (Poisson) and time-dependent (Brownian Passage Time, BPT) recurrence models, respectively [46]. Alternatively, the `TruncatedGR()` function applies a truncated Gutenberg–Richter distribution using moment-balance constraints and fault-specific b values. SARs are then passed to the OpenQuake engine to generate seismic hazard maps corresponding to 10% and 2% PoE. Finally, the results are visualized using the GMT Python toolkit, which produces detailed hazard map plots.

Algorithm 1 FQSHA GUI-Based Seismic Hazard Workflow

```
Input: Fault geometry (JSON), hazard parameters (GUI), user selections
Launch PyQt5 GUI and display fields for model input and configuration
On Browse button click:
    Open file dialog and parse fault JSON
On RUN button click:
    Collect user inputs from GUI fields
    Validate input fields and apply defaults
    Create output directory tree
Prepare Input Files:
    Export parameters to inputs.json
    Export fault traces to fault_traces.json
    Call source_model_logic_tree() to generate source model logic tree XML
    Call generate_job_ini() to create OpenQuake configuration
    Call gmpe_generate_xml() to export selected GMPEs
Moment Budget and Scaling:
    Call momentbudget() from SeismicActivityRate.py
        Uses: kin2coeff(), coeff2mag(), conflate_pdfs()
        Computes: Mmax, $\sigma_{M_{\max}}$, $T_{mean}$, MoRate
    Generates PDFs and plots for fault-specific magnitude models
Seismic Activity Rate Modeling:
    Call sactivityrate() with:
        - Fault behavior (Characteristic Gaussian or Truncated GR)
        - Time window, bin width
        Internally calls:
            CHGaussPoiss() or CHGaussBPT() for Gaussian faults
            TruncatedGR() for GR faults
    Updates fault XML files using export_faults_to_xml()
Run Hazard Calculation:
    Call run_oq_engine() to run OpenQuake engine with job.ini
    Exports CSV outputs to OutPut/
Map Visualization:
    Use find_latest_hazard_map() to locate latest CSV result
if hazard map is found then
    Call create_contour_map_with_faults() to plot hazard contours
end if
Display completion message and highlight output folder
```

Algorithm 2 Seismic Activity Rate Calculation in FQSHA

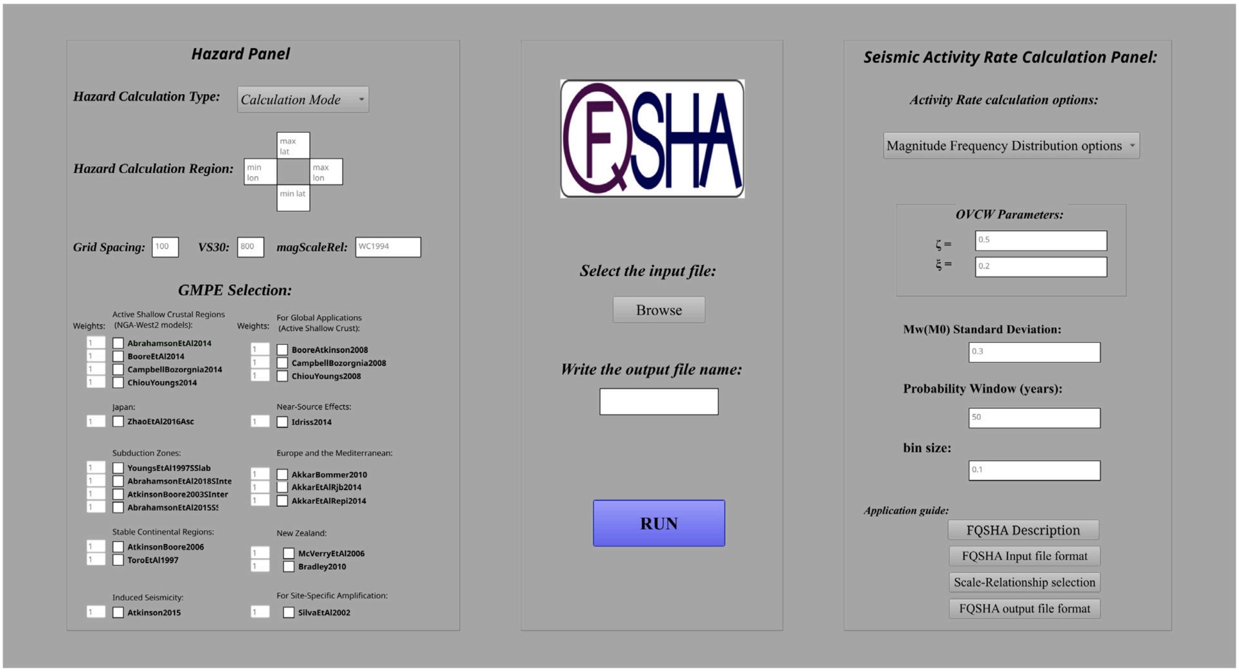
```
Input: Fault dictionary from JSON, parameters: Zeta, Khi, Siggma, Fault behaviour, time window $w$, bin width
Initialize constants $c = 1.5$, $d = 9.1$
for all faults in input do
    Extract and convert fault geometry and physical parameters
    Compute fault area from dip and seismogenic thickness
    Derive multiple magnitude estimates:
    - $M_{\mathrm{Mo}}$: from moment equation
    - $M_{\mathrm{ASP}}$: from aspect ratio
    - $M_{\mathrm{RLD}}, M_{\mathrm{RA}}$: from scaling laws via coeff2mag()
    if observed magnitude $M_{obs}$ exists then
        Conflate $M_{obs}$ with other estimates using conflate_pdfs()
        Adjust SD using optional parameters:
            - $\zeta$ and $\xi$: the factors for $\sigma_{M_{\max}}$ calculation by CoP
    end if
    Compute $M_{\max}$ and $\sigma_{M_{\max}}$ from conflated PDF
    Compute $T_{\mathrm{mean}}$ using moment rate and geometry
    Store $M_{\max}, \sigma_{M_{\max}}, T_{\mathrm{mean}}$ in fault dictionary
    Plot magnitude PDFs and save figures
end for

Activity Rate Modeling (sactivityrate):
for all faults with computed $M_{\max}$ and $T_{\mathrm{mean}}$ do
    Generate magnitude range: $[M_{\max} - \sigma, M_{\max} + \sigma]$
    Compute moment distribution and normalize PDF
    Calculate total moment and scale to match moment rate
    Derive cumulative rate: $T_m = 1/\lambda$
    if Telapsed is available then
        Compute time-dependent probability $H_{BPT}$ using inverse Gaussian
    else
        Use Poissonian probability $H_{Pois}$
    end if
end for

Distribution Handling:
if Fault behavior is "Characteristic Gaussian" then
    if Telapsed exists then
        Call CHGaussBPT() to compute BPT hazard and export XML
    else
        Call CHGaussPoiss() to compute Poisson hazard and export XML
    end if
else if Fault behavior is "Truncated Gutenberg Richter" then
    Call TruncatedGR() for incremental and cumulative rates
else
    Log warning about unknown fault behavior
end if
```

3. Illustrative example

The input selection of the application and the functionality is further demonstrated and evaluated through an operational simple case study. Furthermore, the outputs generated by the software are presented. These illustrations include:

- The necessary information and format for the input.
- Defining the inputs and selecting the options in the GUI.
- The outputs generated by applying the SAR calculation step.
- The generated file structure, necessary for hazard calculation.
- The generated output hazard maps for a sample study area.

The GUI includes intuitive buttons and icons that allow users to configure parameters related to both seismic activity rate calculation and seismic hazard assessment. The right panel allows users to define parameters for seismic activity rate computation, including ζ and ξ (factors used to assess the inclusion of M_{obs} in M_{max} and the calculation of $\sigma_{M_{\max}}$ by the CoP, respectively) as well as SD values and MFD balancing parameters such as the probability window and bin size (see [11] for more details). The left panel is dedicated to seismic hazard assessment inputs. Here, users can specify the hazard calculation type, define the extent of the region of interest for grid-based hazard computation using geographical coordinates, input the shear-wave velocity (V_{s30}), select the magnitude scale relationship, and choose the appropriate GMPE. GMPE are essential components of the OpenQuake hazard assessment workflow. In the OpenQuake engine:


Hazard Panel
Hazard Calculation Type:
Calculation Mode
Hazard Calculation Region:
max lat
min lon
max lon
min lat
GMPE Selection:
Weights:
Active Shallow Crustal Regions (NGA-West2 models):
Weights:
For Global Applications (Active Shallow Crust):
Japan:
Near-Source Effects:
Subduction Zones:
Europe and the Mediterranean:
Stable Continental Regions:
New Zealand:
Induced Seismicity:
For Site-Specific Amplification:
FQSHA
Select the input file:
Browse
Write the output file name:
RUN
Seismic Activity Rate Calculation Panel:
Activity Rate calculation options:
Magnitude Frequency Distribution options
OVCW Parameters:
Application guide:
FQSHA Description
FQSHA Input file format
Scale-Relationship selection
FQSHA output file format

Grid Spacing: 100 VS30: 800 magScaleRel: WC1994

1 □ AbrahamsonEtAl2014
1 □ BooreEtAl2014
1 □ CampbellBozorgnia2014
1 □ ChiouYoungs2014
1 □ BooreAtkinson2008
1 □ CampbellBozorgnia2008
1 □ ChiouYoungs2008
1 □ ZhaoEtAl2016Asc
1 □ Idriss2014
1 □ YoungsEtAl1997SSlab
1 □ AbrahamsonEtAl2018SInte
1 □ AtkinsonBoore2003SInter
1 □ AbrahamsonEtAl2015SS
1 □ AkkarBommer2010
1 □ AkkarEtAlRjb2014
1 □ AkkarEtAlRepi2014
1 □ AtkinsonBoore2006
1 □ ToroEtAl1997
1 □ McVerryEtAl2006
1 □ Bradley2010
1 □ Atkinson2015
1 □ SilvaEtAl2002

ζ = 0.5
ξ = 0.2

Mw(M0) Standard Deviation: 0.3

Probability Window (years): 50

bin size: 0.1

Fig. 2. The GUI of FQSHA, displaying the main menus of the software. The left panel shows the input parameters for hazard calculation, including a section for defining the geographical extent for seismic hazard computation, grid spacing, Vs30 values, and the selection of various GMPEs and introducing their weights to be incorporated into the logic tree model. The middle panel provides options to browse for the JSON input file and specify the output file name for saving the hazard results. The right panel presents the input parameters for the SAR calculation.

- GMPEs are assigned to specific Tectonic Region Types, such as Active Shallow Crust.
- Multiple GMPEs can be combined using a logic tree structure, which allows for incorporating epistemic uncertainty in ground-motion modeling.
- Each GMPE branch is assigned a weight that reflects its relative applicability or expert-assigned credibility. These weights must sum to exactly 1.0 within each branch set.

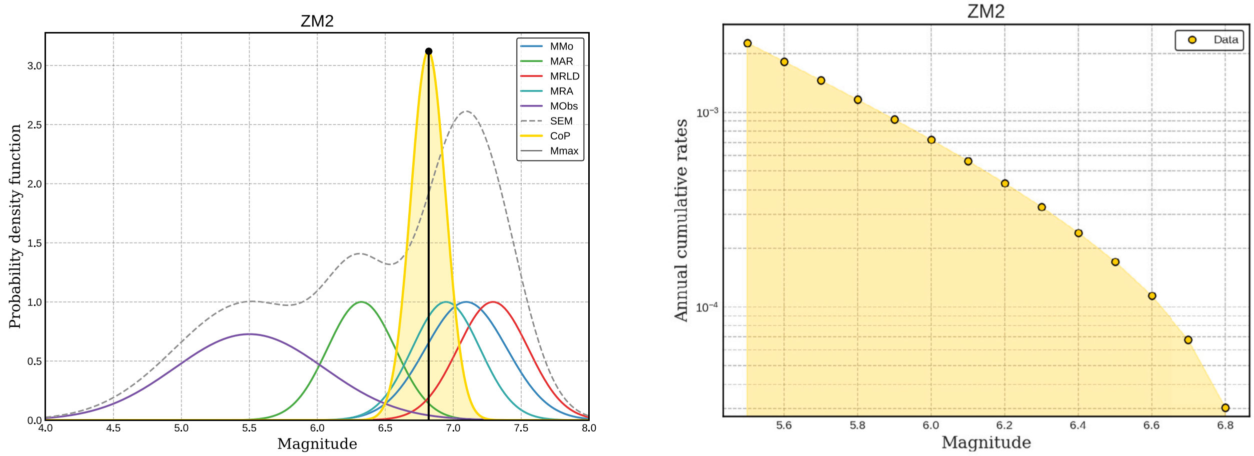
The FQSHA GUI allows users to configure GMPEs for use with the OpenQuake engine as follows:

- Users can select one or more GMPEs using checkboxes within the graphical interface.
- For each selected GMPE, the user may optionally input a custom weight. If no value is provided, a default of 1.0 is assumed.
- The GUI automatically normalizes the weights of all selected GMPEs to ensure the sum equals 1.0, as required by OpenQuake.
- A logic tree file (`gmpe_logic_tree.xml`) is generated, defining the GMPE branch set with the specified models and weights in the NRML format.
- This logic tree file is referenced in the OpenQuake job configuration (`job.ini`) and used during the hazard calculation.

By executing the main code, GUI (Fig. 2) reads the input file (Table. 4) FQSHA prepares first set of output files which are the result of applying SAR analysis: the plot of Empirical Magnitudes (EMs) and the observed magnitude with the calculated M_{max} by CoP (Fig. 3(a)), the SAR values of each fault and their corresponding probability curves plots (Fig. 3(b)), the output folder containing the calculated hazard maps, and the source file containing the OpenQuake formatted fault source XML file (Listing 1).

Listing 1: The sample XML output file that can be used by OpenQuake engin for the hazrd calculation.

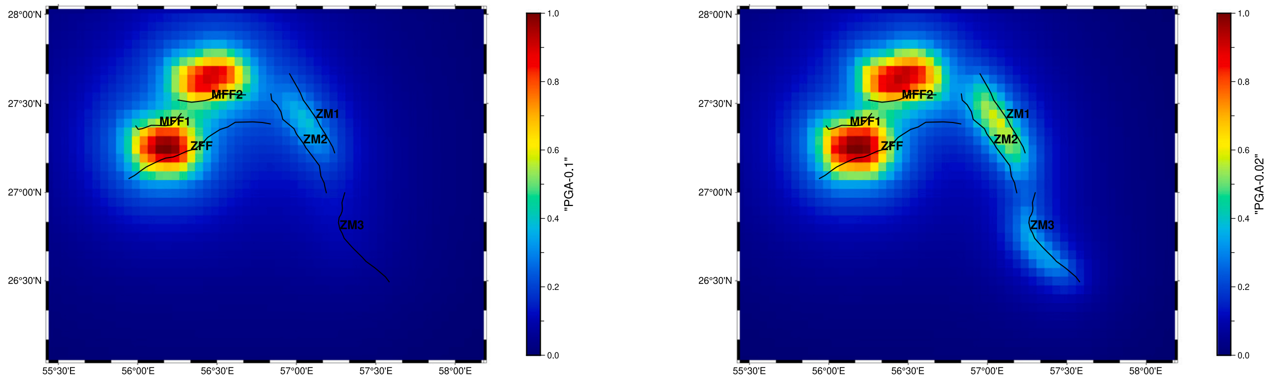
```
<?xml version="1.0" encoding="utf-8"?>
<nrml xmlns="http://openquake.org/xmlns/nrml/0.4"
     xmlns:gml="http://www.opengis.net/gml">
   <sourceModel name="ZM2">
      <simpleFaultSource id="5" name="ZM2Source"
                    tectonicRegion="ActiveShallow
                        Crust">
         <simpleFaultGeometry>
            <gml:LineString>
               <gml:posList>
                  57.1870688929067 26.9933947333503
                  57.1788116557739 27.0243593725982
                  57.1726187279244 27.0522275479213
                  57.1519756350924 27.0904172696603
                  57.1426862433181 27.1492500842313
                  57.0745640369728 27.2307903009173
                  57.0580495627072 27.2493690844661
                  57.0415350884417 27.2751729505059
                  56.9981845934947 27.3278128372273
                  56.9827022738707 27.3577453218336
                  56.9114836036006 27.410385208555
                  56.9001299025431 27.4527035488604
                  56.8732938818616 27.4929575798826
                  56.8423292426137 27.5187614459225
                  56.8361363147642 27.55591901302
               </gml:posList>
            </gml:LineString>
            <dip>70</dip>
            <upperSeismoDepth>10.0</upperSeismoDepth>
            <lowerSeismoDepth>20.0</lowerSeismoDepth>
         </simpleFaultGeometry>
         <magScaleRel>WC1994</magScaleRel>
         <ruptAspectRatio>2.0000000E+00</
             ruptAspectRatio>
         <incrementalMFD minMag="5.5" binWidth="0.1">
            <occurRates>
               4.506316e-04 3.662871e-04 2.977294e-04
               2.420035e-04 1.967078e-04 1.598901e-04
               1.299636e-04 1.056384e-04 8.586608e-05
               6.979457e-05 5.673116e-05 4.611282e-05
```



(a) Illustration of potential maximum magnitude estimations and the final result computed using conflation methods.

(b) Annual cumulative activity rates of the sample fault balanced over the magnitude range between M_{min} and M_{max}.

Fig. 3. Two outputs from the FQSHA software: (a) fault-based maximum magnitude estimation and (b) SAR computation based on fault-specific MFDs.



(a) Output hazard map for sample faults with a 10% Probability of Exceedance.

(b) Output hazard map for sample faults with a 2% Probability of Exceedance.

Fig. 4. The generated PGA hazard maps for the sample faults, corresponding to 10% and 2% PoE, were produced using the FQSHA workflow. The calculations assume an SCC of 25%, and the maps were plotted with the GMT tool integrated into the workflow.

```
                    3.748191e-05 3.046644e-05
                </occurRates>
            </incrementalMFD>
            <rake>80</rake>
        </simpleFaultSource>
    </sourceModel>
</nrml>
```

This is the list of the output files generated after applying the FQSHA workflow:

```
FQSHA_output/
|-- Sources/
|-- OutPut/
|-- fault_traces.json
|-- gmpe_logic_tree.xml
|-- inputs.json
|-- job.ini
|-- source_model_logic_tree.xml
```

The computed hazard maps are rendered within the user-defined geographic extent using the GMT toolkit, illustrating ground motion levels corresponding to 10% (Fig. 4(a)) and 2% (Fig. 4(b)) PoE. In addition, the resulting hazard data are available for visualization using external mapping tools.

4. Software testing and verification

To ensure the correctness and reliability of FQSHA's core functionalities, we utilized Python's built-in `unittest` framework [47] along with the `coverage.py` tool for code coverage analysis. This approach improves software stability and minimizes the risk of errors according to [48] and is effective in validating Python-based scientific software components.

4.1. Automated GUI testing with pytest-qt

For GUI testing in PyQt5 applications, the `pytest-qt` plugin extends `pytest` to support interaction-based testing. This tool enables developers to simulate user actions, verify UI responsiveness, and assess the functionality and stability of the GUI without manual intervention [49]. These tests replicate user interactions — such as parameter selection, button clicks, and computation triggers — to ensure the GUI behaves as expected. Moreover, this testing approach verifies the GUI's ability to invoke core functions and generate outputs, including input data handling, M_{max} computation, seismic activity rate calculation, visualization of the annual cumulative moment budget, EMs plots with M_{max} annotations, and successful integration with OpenQuake. The outcomes are compiled into a comprehensive test report.

Table 4
Input Fault data in JSON format.

```
{
  "ZFF": {
    "ScR": "WC94-R",
    "year_for_calculations": 2024,
    "Length": 109,
    "Dip": 40,
    "upperSeismoDepth": 3,
    "lowerSeismoDepth": 10,
    "SRmin": 1.36,
    "SRmax": 2.04,
    "Mobs": 6.4,
    "sdMobs": 0.05,
    "Last_eq_time": 1497,
    "SCC": 0.205,
    "ShearModulus": 3,
    "StrainDrop": 3,
    "Mmin": 5.5,
    "b-value": 0.9,
    "fault_trace": [
      [56.8364, 27.3840],
      [56.7842, 27.3923],
      ...
    ]
  },
  "ZM1": {
    "ScR": "WC94-R",
    "year_for_calculations": 2024,
    "Length": 60,
    "Dip": 70,
    "upperSeismoDepth": 2,
    "lowerSeismoDepth": 12,
    "SRmin": 2.2,
    "SRmax": 4.3,
    "Mobs": 5.5,
    "sdMobs": 0.05,
    "Last_eq_time": 1950,
    "SCC": 0.205,
    "ShearModulus": 3,
    "StrainDrop": 3,
    "Mmin": 5.5,
    "b-value": 0.9,
    "fault_trace": [
      [56.9531, 27.6688],
      [56.9761, 27.6329],
      ...
    ]
  }
}
```

4.2. Unit testing for core functions

Unit testing was also applied to critical functions within FQSHA, particularly those responsible for SAR calculations and input/output data handling. The `pytest` framework was utilized to automate testing, enabling verification of expected outputs under various input conditions. All test cases are implemented as Python classes inheriting from `unittest.TestCase`, and they can be executed using the `unittest` test runner (e.g., via `python -m unittest tests/test_sample.py -v`).

4.3. Code coverage analysis

To assess the thoroughness of our testing efforts, we used the `coverage.py` tool [50], which measures the percentage of source code executed by the test suite, helping to identify untested code and improve overall robustness. For the GUI test code (`tests/test_gui.py`), out of 76 statements only 3 were missed, resulting in an overall coverage of **96%**. For the core functions and data-handling tests (`tests/test_sactivityrate_xmlExport.py`), 60 statements were evaluated with 4 missed, corresponding to an overall coverage of **93%**. The tool provides detailed reports including statement coverage and highlights untested blocks in an interactive HTML format. All test scripts and configurations are included in the source code repository. The tests can be executed in any Python environment with minimal setup and can be automatically run using the `coverage.py` test runner (e.g., via `coverage run -m unittest tests/test_guiii.py -v`).

5. Impact

The FQSHA tool serves as a comprehensive and streamlined framework for fault-based seismic hazard analysis, transforming raw fault input data into final hazard estimates with minimal user intervention. By integrating the Seismic Activity Rate (SAR) computation engine (FaultQuake) with the widely used hazard calculation platform (OpenQuake), FQSHA provides an environment tailored for Probabilistic Seismic Hazard Assessment (PSHA) directly from fault sources. The tool is automated and designed with usability in mind, making it accessible to both experts and new users to the geoscience community. Given its ease of use and efficiency in addressing a critical aspect of seismic hazard assessment, we expect it to attract a broad range of users and applications. Its utility and user-centered design are expected to encourage frequent use, broad adoption, and noticeable impact within the field.

6. Conclusions

The development of FQSHA addresses the need for a transparent and accessible tool to support research and education in fault-based PSHA. It streamlines the modeling process by enabling users to perform fault-based hazard calculations with minimal manual input. The toolkit integrates a graphical interface for computing fault-based SAR, which is directly coupled with the OpenQuake Engine for hazard computations and the GMT toolkit for visualizing hazard maps. Its modular architecture supports ongoing development and customization. Planned extensions include the incorporation of advanced SAR models and additional hazard computation methodologies, as well as tighter integration with OpenQuake's latest features. Future work will also explore validation using established benchmarks such as GEM datasets. Given its design and flexibility, FQSHA holds potential for application in national seismic hazard studies and educational training, and may be extended toward supporting early warning and decision-support systems.

CRediT authorship contribution statement

Nasrin Tavakolizadeh: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Hamzeh Mohammadigheymasi:** Writing – original draft, Validation, Supervision, Software, Methodology, Investigation, Conceptualization. **Nuno Pombo:** Validation, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is funded by FCT/MCTES through national funds and when applicable co-funded by FEDER—PT2020 partnership agreement under the project UIDB/EEA/50008/2020.

Data availability

I have shared the link to my codes on GitHub: https://github.com/GeoSignalAnalysis/FQSHA.

References

[1] Mohammadigheymasi H, Shi P, Tavakolizadeh N, Xiao Z, Mousavi SM, Matias L, Pourvahab M, Fernandes R. IPIML: A deep-scan earthquake detection and location workflow integrating pair-input deep learning model and migration location method. IEEE Trans Geosci Remote Sens 2023;61:1–9. http://dx.doi.org/10.1109/TGRS.2023.3293914.

[2] Xiao Z, Wang J, Liu C, Li J, Zhao L, Yao Z. Siamese earthquake transformer: A pair-input deep-learning model for earthquake detection and phase picking on a seismic array. J Geophys Res: Solid Earth 2021;126(5). http://dx.doi.org/10.1029/2020JB021444, e2020JB021444.

[3] Mousavi SM, Ellsworth WL, Zhu W, Chuang LY, Beroza GC. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. Nat Commun 2020;11(1):3952. http://dx.doi.org/10.1038/s41467-020-17591-w.

[4] Mohammadigheymasi H, Tavakolizadeh N, Matias L, Mousavi SM, Silveira G, Custódio S, Dias N, Fernandes R, Moradichaloshtori Y. Application of deep learning for seismicity analysis in Ghana. Geosystems Geoenvironment 2023;2(2):100152. http://dx.doi.org/10.1016/j.geogeo.2022.100152.

[5] Field EH, Jordan TH, Cornell CA. OpenSHA: A developing community-modeling environment for seismic hazard analysis. Seismol Res Lett 2003;74(4):406–19.

[6] Field EH, Arrowsmith JR, Biasi GP, Bird P, Dawson TE, Felzer KR, Jackson DD, Johnson KM, Jordan TH, Madden C, Michael AJ, Milner KR, Page MT, Parsons T, Powers PM, Shaw BE, Thatcher WR, Weldon RJ, Zeng Y. Uniform California earthquake rupture forecast, version 3 (UCERF3)—The time-independent model. Bull Seismol Soc Am 2014;104(3):1122–80. http://dx.doi.org/10.1785/0120130164.

[7] Danciu L, Giardini D, Weatherill G, Basili R, Nandan S, Rovida A, Beauval C, Bard P-Y, Pagani M, Reyes CG, Sesetyan K, Vilanova S, Cotton F, Wiemer S. The 2020 European seismic hazard model: overview and results. Nat Hazards Earth Syst Sci 2024;24:3049–73. http://dx.doi.org/10.5194/nhess-24-3049-2024.

[8] Pace B, Visini F, Peruzza L. FiSH: MATLAB tools to turn fault data into seismic-hazard models. Seismol Res Lett 2016;87(2A):374–86. http://dx.doi.org/10.1785/0220150189.

[9] Visini F, Valentini A, Chartier T, Scotti O, Pace B. Computational tools for relaxing the fault segmentation in probabilistic seismic hazard modelling in complex fault systems. Pure Appl Geophys 2020;177(4):1855–77. http://dx.doi.org/10.1007/s00024-019-02350-1.

[10] Chartier T, Scotti O, Lyon-Caen H. SHERIFS: Open-source code for computing earthquake rates in fault systems and constructing hazard models. Seismol Res Lett 2019;90(4):1678–88. http://dx.doi.org/10.1785/0220180332.

[11] Tavakolizadeh N, Mohammadigheymasi H, Visini F, Pombo N. FaultQuake: An open-source python tool for estimating seismic activity rates in faults. Comput Geosci 2024;191:105659. http://dx.doi.org/10.1016/j.cageo.2024.105659.

[12] Tavakolizadeh N, Mohammadigheymasi H, Matias L, Silveira G, Fernandes R, Dolatabadi N. To what extent do slip rates contribute to the seismic activity of faults? In: EGU general assembly conference abstracts. 2022, p. EGU22–12893. http://dx.doi.org/10.5194/egusphere-egu22-12893.

[13] Pagani M, Monelli D, Weatherill G, Danciu L, Crowley H, Silva V, Henshaw P, Butler L, Nastasi M, Panzeri L, Simionato M, Vigano D. The OpenQuake engine: An open hazard (and risk) software for the global earthquake model. Seismol Res Lett 2014;85(3):692–702. http://dx.doi.org/10.1785/0220130087.

[14] Field EH, Jackson DD, Dolan JF. A mutually consistent seismic-hazard source model for southern California. Bull Seismol Soc Am 1999;89(3):559–78.

[15] Danciu L, Giardini D, Woessner J, Monelli D, Crowley H, Cotton F, Grünthal G, Valensise G, Basili R, Slejko D, Meletti C, Rovida A, Sesetyan K, Demircioglu MB, Hiemer S, Zuccolo E, Sandikkaya MA, Sørensen MB, Lindholm C, Stucchi M. The 2020 update of the European seismic hazard model: Mapping seismic hazard across europe. Earthq Spectra 2021;37(1_suppl):67–91. http://dx.doi.org/10.12686/a15.

[16] Youngs RR, Coppersmith KJ. Probabilistic seismic hazard analysis. In: Earthquake engineering handbook. CRC Press; 2003.

[17] Leonard M. Earthquake fault scaling: Self-consistent relating of rupture length, width, average displacement, and moment release. Bull Seismol Soc Am 2010;100(5A):1971–88. http://dx.doi.org/10.1785/0120090189.

[18] Wells DL, Coppersmith KJ. New empirical relationships among magnitude, rupture length, rupture width, rupture area, and surface displacement. Bull Seismol Soc Am 1994;84(4):974–1002. http://dx.doi.org/10.1785/BSSA0840040974.

[19] Azzaro R, D'Amico S, Peruzza L, Tuvè T. Estimating the expected seismicity rates of volcano-tectonic earthquakes at mt. Etna (Italy) by a geometric-kinematic approach. In: Atti del 33° convegno nazionale, gruppo nazionale di geofisica della terra solida, tema 1: geodinamica. 2014, Conference paper; available from GNGTS archive. URL https://gngts.ogs.it/archivio/files/2014/S11/Riassunti/Azzaro.pdf.

[20] Villamor P, Berryman KR. A late quaternary extension rate in the Taupō Volcanic zone, New Zealand, derived from fault slip data. N. Z J Geol Geophys 2001;44(2):243–69. http://dx.doi.org/10.1080/00288306.2001.9514937.

[21] Leonard M. Self-consistent earthquake fault-scaling relations: Update and extension to stable continental strike-slip faults. Bull Seismol Soc Am 2014;104(6):2953–65. http://dx.doi.org/10.1785/0120140087.

[22] Strasser FO, Arango MC, Bommer JJ. Scaling of the source dimensions of subduction zone megathrust earthquakes. Seismol Res Lett 2010;81(6):941–50. http://dx.doi.org/10.1785/gssrl.81.6.941.

[23] Shaw JH, Plesch A, Dolan JF, Pratt TL, Fiore P. Puente hills blind-thrust system, los angeles, California. Bull Seismol Soc Am 2002;92(8):2946–60. http://dx.doi.org/10.1785/0120010291.

[24] Thingbaijam KKS, Mai PM, Goda K. New empirical earthquake source-scaling laws. Bull Seismol Soc Am 2017;107(5):2225–46. http://dx.doi.org/10.1785/0120170017.

[25] Abrahamson NA, Silva WJ, Kamai R. NGA-West2 empirical ground-motion model for the average horizontal component of PGA, PGV, and 5%-damped linear acceleration response spectra. Earthq Spectra 2014;30(3):1025–55. http://dx.doi.org/10.1193/070913EQS198M.

[26] Boore DM, Stewart JP, Seyhan E, Atkinson GM. NGA-West2 equations for predicting PGA, PGV, and 5%-damped PSA for shallow crustal earthquakes. Earthq Spectra 2014;30(3):1057–85. http://dx.doi.org/10.1193/070113EQS184M.

[27] Campbell KW, Bozorgnia Y. NGA-West2 ground motion model for the average horizontal components of PGA, PGV, and 5% damped linear acceleration response spectra. Earthq Spectra 2014;30(3):1087–115. http://dx.doi.org/10.1193/062913eqs175m.

[28] Chiou BS-J, Youngs RR. NGA-West2 ground-motion model for the average horizontal component of peak ground motion and response spectra. Earthq Spectra 2014;30(3):1121–51. http://dx.doi.org/10.1193/072813EQS219M.

[29] Youngs RR, Chiou BS-J, Silva WJ, Humphrey JR. Strong ground motion attenuation relationships for subduction zone earthquakes. Seismol Res Lett 1997;68(1):58–73. http://dx.doi.org/10.1785/gssrl.68.1.58.

[30] Atkinson GM. Ground-motion prediction equation for small-to-moderate events at short hypocentral distances, with application to induced-seismicity hazards. Bull Seismol Soc Am 2015;105(2A):981–92. http://dx.doi.org/10.1785/0120140142.

[31] Atkinson GM, Boore DM. Empirical ground-motion relations for subduction-zone earthquakes and their application to cascadia and other regions. Bull Seismol Soc Am 2003;93(4):1703–29. http://dx.doi.org/10.1785/0120020156.

[32] Atkinson GM, Boore DM. Earthquake ground-motion prediction equations for eastern north America. Bull Seismol Soc Am 2006;96(6):2181–205. http://dx.doi.org/10.1785/0120060059.

[33] Abrahamson N, Gregor N, Addo K. Summary of the ASK14 ground motion relation for subduction interface earthquakes. Techreport PEER 2018/02, Berkeley, CA: Pacific Earthquake Engineering Research Center; 2018, URL https://peer.berkeley.edu/publications/peer-reports/2018-summary-ask14-ground-motion-relation-subduction-interface-earthquakes.

[34] Toro GR, Abrahamson NA, Schneider JF. Model of strong ground motions from earthquakes in central and eastern north America: Best estimates and uncertainties. Seismol Res Lett 1997;68(1):41–57. http://dx.doi.org/10.1785/gssrl.68.1.41.

[35] Akkar S, Bommer JJ. Empirical ground-motion models for point- and extended-source crustal earthquake scenarios in Europe and the middle east. Bull Earthq Eng 2010;8(2):335–66. http://dx.doi.org/10.1007/s10518-013-9461-4.

[36] Abrahamson NA, Gregor N, Addo K. BC hydro ground motion prediction equations for subduction earthquakes. Earthq Spectra 2016;32(1):23–44. http://dx.doi.org/10.1193/051712EQS188MR.

[37] Boore DM, Atkinson GM. Ground-motion prediction equations for the average horizontal component of PGA, PGV, and 5%-damped PSA at spectral periods between 0.01 s and 10.0 s. Earthq Spectra 2008;24(1):99–138. http://dx.doi.org/10.1193/1.2830434.

[38] Campbell KW, Bozorgnia Y. NGA ground motion model for the geometric mean horizontal component of PGA, PGV, PGD and 5%-damped linear elastic response spectra. Earthq Spectra 2008;24(1):139–71. http://dx.doi.org/10.1193/1.2857546.

[39] Chiou BS-J, Youngs RR. An NGA model for the average horizontal component of peak ground motion and response spectra. Earthq Spectra 2008;24(1):173–215. http://dx.doi.org/10.1193/1.2894832.

[40] Zhao JX, Jiang F, Shi P, Xing H, Huang H, Hou R, Zhang Y, Yu P, Lan X, Rhoades DA, Somerville PG, Irikura K, Fukushima Y. Ground-motion prediction equations for subduction slab earthquakes in Japan using site class and simple geometric attenuation functions. Bull Seismol Soc Am 2016;106(4):1535–51. http://dx.doi.org/10.1785/0120150056.

[41] Idriss IM. An NGA-West2 empirical model for estimating the horizontal spectral values generated by shallow crustal earthquakes. Earthq Spectra 2014;30(3):1155–77. http://dx.doi.org/10.1193/070613EQS195M.

[42] McVerry GH, Abrahamson NA, Caldwell J, Zhao JX. New Zealand attenuation relations for acceleration spectra. Bull N. Z Soc Earthq Eng 2006;39(1):1–58. http://dx.doi.org/10.5459/bnzsee.39.1.1-58.

[43] Bradley BA. A New Zealand–specific pseudospectral acceleration ground-motion prediction equation for active shallow crustal earthquakes based on foreign models. Bull Seismol Soc Am 2010;100(3):864–80. http://dx.doi.org/10.1785/0120120021.

[44] Silva WJ, Gregor N, Darragh RB. Development of regional hard rock attenuation relations for central and eastern north America. Tech. rep., El Cerrito, CA: Pacific Engineering and Analysis; 2002, Technical report prepared for Brookhaven National Laboratory, Upton, NY, for U.S. Nuclear Regulatory Commission, Contract No. NRC-04-91-068. URL https://www.ce.memphis.edu/7137/PDFs/attenuations/Silva_Darragh_2002.pdf.

[45] Akkar S, Sandıkkaya MA, Bommer JJ. Empirical ground-motion models for point- and extended-source crustal earthquake scenarios in Europe and the middle east. Bull Earthq Eng 2014;12(1):359–87. http://dx.doi.org/10.1007/s10518-013-9461-4.

[46] Matthews MV, Ellsworth WL, Reasenberg PA. A Brownian model for recurrent earthquakes. Bull Seismol Soc Am 2002;92(6):2233–50. http://dx.doi.org/10.1785/0120010267.

[47] Foundation PS. Unittest — Unit testing framework. 2024, Python Standard Library, Version 3.12. URL https://docs.python.org/3/library/unittest.html.

[48] Gruber M, Lukasczyk S, Kroiß F, Fraser G. An empirical study of automated unit test generation for Python. Empir Softw Eng 2022;27(5). http://dx.doi.org/10.1007/s10664-022-10248-w.

[49] pytest-qt contributors. pytest-qt: Pytest plugin for qt and PySide/PyQt applications. 2025, URL https://pypi.org/project/pytest-qt/. [Accessed 11 April 2025].

[50] Batchelder N. Coverage.py: Code coverage measurement for python. 2024, Version 7.4.0. URL https://coverage.readthedocs.io/.