Reviewer 1 - Pablo Iturrieta

I have reviewed the Software Report article describing the OpenQuake Model Building Toolkit (OQ-MBTK). This is a Python-based tool suite designed to create the input models for Probabilistic Seismic Hazard Analysis (PSHA). It guides users through the end-to-end process of constructing Earthquake Rupture Forecasts—Seismicity Source Model (ERF/SSM)-- and calibrating Ground-Motion Models (GMM), ensuring consistency and reproducibility throughout. The final outputs are OpenQuake Engine-compatible input files to perform traditional hazard calculations. OQ-MBTK is aimed at researchers and hazard practitioners, particularly for those constructing models (e.g., National, Regional PSHA models) with reproducibility requirements, that is, reproducing hazard results from its constituent input data—catalogs, fault databases and strain-rate maps—through consistent computational steps, methods and code.

The OpenQuake ecosystem is widely used by researchers around the world, and efforts like the OQ-MBTK to standardize the construction of hazard are highly valuable. It will facilitate the development and reproducibility of new models, as well as serving as training resource for future engineering seismologists. Therefore, I consider this toolkit to play an important role in advancing the field. It also offers a platform to share, review and develop the implementation of new methods to build ERFs and GMMs. This review assessed the manuscript, documentation, and codebase, while also implementing an own model based on the provided materials.

The article is well written and effectively presents an overview of the toolkit's capabilities. Some aspects, detailed in the *Manuscript Review* section, could be improved—particularly the inclusion in the manuscript of two modules intended for internal GEM use rather than public deployment (see also Technical Review, **ghm** and **man** modules). Overall, the documentation is quite clear, but varies in quality across modules and lacks comprehensive tutorials/use cases that combine instructions, data and configuration files. A complete reference for all available configuration arguments of config-based modules is unfortunately missing.

The codebase is of high quality: modular, well-structured and extensively unit-tested. It follows open-source practices, is released under an open license, and is architected for extensibility. This package clearly builds on years of development and practical insights accumulated within GEM and consolidates numerous methods and practices used in the construction of PSHA models. While the overall design is robust, there are few areas where the user experience and codebase could be improved. The main caveat is the mixed use of declarative (configuration-based) and imperative (step-by-step) approaches in some modules, which can burden users with tracking numerous intermediate files without centralized guidance. Many modules, notebooks and functions in the toolkit are not kept up to date, or are redundant, and could be essentially removed for better maintainability (and re-implemented when they can be effectively used by the user). There are some minor issues with dependencies (described below). Whereas one of the strong points of this manuscript is assisting the hazard models' reproducibility, there is no comment or documentation about computational reproducibility (e.g., system and library dependencies). Nonetheless, this is definitely a publication-worthy piece of software, and I highly recommend its acceptance once the noted issues have been addressed. I also understand that this is an evolving software, and I would be open to some of my recommendations being deferred to a Roadmap when justified.

I suggest the manuscript to be accepted after minor revisions. As a minimum requirement, I recommend the authors ensure that all modules presented in the manuscript includes at least basic documentation and a testable use case. These should involve example data, configuration files, and clear instructions on how to execute the workflow--many of which already exists in notebooks and unit tests scattered across the repository. If data size is a concern for repository cleanness, authors may consider hosting use-case datasets externally and linking them from the documentation—README files, rather than bundling them directly into the repository.

Looking forward to see this manuscript published,

Pablo Iturrieta

Technical Review

Here, I acted both as reviewer and partial beta-tester while attempting to integrate the toolkit into my own workflows. This review is by no means exhaustive (and is presented in the order in which issues emerged rather than by priority) as the toolkit is extensive, and due to the limited documentation in several modules, it was often difficult to distinguish between actively maintained features and legacy components. Nonetheless, my review is not a professional tester effort, but I hope some observations compiled here assist the authors in further polishing the software. Some essential comments herein, particularly related to documentation and demos, should indeed be refined and addressed by the authors.

Installation

- Installation was straightforward using the provided instructions.
- However, dependency issues persist with MBTK and GMT. Although the documentation states that GMT is only required for plotting, GMT becomes a de facto dependency across all modules:
    - **(Details)**: Running any mbtk command (e.g., oqm cat merge) triggers GMT dependencies because:
        - pygmt is always installed via requirements-py*-linux.txt;

        - All submodules in mbi are imported (via engine.openquake.baselib.sap), eventually loading pygmt through openquake.sub.get_profiles_from_slab2pt_geojson.py
        - If pygmt fails o find the shared libgmt.so library, the code breaks due to improper exception handling in get_profiles_from_slab2pt_geojson.py:L41.
- Even with GMT libraries installed via apt, shared libraries (e.g., */usr/lib/libgmt.so*) are not detected by the installed pygmt version due to version incompatibilities. This required manual creation of symbolic links to allow pygmt to recognize them (see: pygmt install issues).
- Geopandas version conflicts: Geopandas now resolves to >=1.0.0, which breaks code relying on deprecated syntax (e.g., gpd.sjoin(..., op='within') should now use predicate='within'). See the GeoPandas 1.0.0 changelog
- *oq-mbtk/notebook/Dockerfile* fails building

Recommendations:

- Avoid loading GMT unless explicitly required; capture exceptions where pygmt may be imported.
- Provide clear installation instructions for GMT/pyGMT, including version requirements.
- Add unit tests that simulate OQ-MBTK CLI usage. Would have the benefit of triggering *oq-engine.sap* library loading all modules, so importing errors can be catch.
- Update Geopandas usage and version requirements accordingly.

cat module

- I was able to follow the documentation and complete an analysis with my own data.
- Tutorials are needed to be included, including example data, configuration files, and instructions (it could be accompanying the existing documentation with the actual used data).
- Documentation is needed to explain (all) the possible configuration files parameters.
- It is not straight-forward for users to import their own catalog formats.
    - No link is provided to the GEM Catalog format; converting to ISF is a heavy overhead.

    - The latest HMTK manual may be outdated (e.g., missing magnitudeType support). An inset in this same OQ-MBTK *cat* documentation containing the accepted format would be excellent.
    - Serialized catalogs from openquake.hmtk.catalogue.Catalogue.write_catalogue() cannot be read back in. I understand that harmonising all Reader/Writers across openquake would be cumbersome, but I suggest to write in each module's documentation what will be the accepted I/O methods.
    - ISC-GEM catalog (from ISC website) cannot be loaded directly, since it is probably not well formatted. I suggest including a parser in the library, because it will be probably the most used proxy catalog by future users.
- Geopandas 1.0+ breaks polygon-based catalog filtering:
    - Example: openquake.cat.hmg.merge.py:L136 uses op='intersects', which should be predicate='intersects'.
- The homogenisation workflow is not consistent: some times I/O arguments are passed through the config files, sometimes through the CLI, which can burden the user with heavy file tracking. This could be harmonised and avoid relying on *.bash* scripts. Same applies for WKT module.
- Switching between TOML configs and manual Python scripting during homogenisation makes workflows harder to reproduce (and require user's familiarity with both interfaces). as future users must manually track configuration files and scripted steps. While I understand iterative analysis is expected in regressions (therefore scripting becomes necessary), the documentation should better reflect how user choices should be placed in the final configuration file to maintain reproducibility.
- Utility code for reading/querying .h5 files that is described in the docs should be part of the cat API.

ghm module

- While the functionalities described in the manuscript are highly relevant for anyone working with multiple hazard models—whether the GEM Mosaic or other model combinations—I'm not in favor of publishing tools that are not meant for general use. In my opinion, it is also in the authors' convenience to create additional repositories for this purpose, so the MBTK repo is kept clean and better maintainable.
- However, this module could have a large impact if it is generalized to support broader use cases, such as having overlapping hazard models, or having global or regional mosaics.

man module

- This module could have essential functionality to evaluate, a-posteriori, the resulting ERF/SSM files of OQ-MBTK. However, the documentation, tutorials and codebase is not at pair with other modules.
- Same as with *ghm*, I don't agree on its inclusion on the manuscript if it is not general use.
- If the package is on early development stages but intended to grow, I recommend to keep it in the manuscript and showcase the existing functionality in tutorials with simple data (already contained in "man.notebooks", albeit currently with no data and hardcoded paths), so users may be able to use its functionality on their workflows. Or move its functionality to *plt*.

mbt module

- The module, as outlined in the manuscript, should be one of the most important modules in the toolkit, but unfortunately currently lacks documentation.
- I was able to adapt some of the notebooks to my own workflow and data, could run some of the routines with a lot of trouble due to old dependencies and breaking code.
- For this manuscript, I would suggest releasing the essential use cases in the documentations. These could be straightforwardily adapted from the existing notebooks (perhaps including them in the docs with the package nbsphinx), once this module has been updated to the requirements of the current MBTK release (i.e., will managing OqtProject/OqtModel be a feature? Otherwise remove) .

wkt module

- I followed the documentation and completed an analysis with my data.
- Documentation is needed to explain (all) the possible configuration files parameters.
- This module overlaps heavily with the intended **mbt** module's purpose. Following the manuscript description, much of the code under *openquake.wkt* should be under *openquake.mbt* (e.g., *compute_gr_params.py*, *analysis_hypocentral_depth.py*, smoothing algorithms, etc).
- I recommend to keep the distinction between config-type coding (declarative) under WKT and move all the functions to MBT (for python scripting). Essentially, **wkt** should work as the config-based wrapper for **mbt** workflow.
- API is hard to follow when tracking all intermediate file locations. I have no easy solution for this, but it would definitely be easier to just have standard-paths defined from the configuration file.
- It is not clear to me the reasoning behind the use of the *PSHAModelBuilder* library when it contains only one method that could easily be included in *OQ-MBTK*.

- I remain unconvinced that the performance benefits gained from using external Julia packages justify the added complexity and dependency overhead. Particularly because the Helmstetter algorithm (in python) is in theory computationally more expensive than Frankel's (in Julia)-- involving distance calculation between catalog-grid and catalog-catalog, rather than just catalog-grid, respectively. This choice detracts from the otherwise streamlined architecture of the toolkit.
- Minor bugs or error in docs:
  - In "Some notes on setup" docs: BIND and BIN1 should point only to: os.path.join('/Users', 'kjohnson', 'GEM', 'oq-mbtk', 'openquake', 'bin').
  - In Julia smoothing script (PSHAModelBuilder.src.seismicity.smoothing.jl:L83-84), when using the example .toml from the docs:
    maxdistkm = model["kernel_maximum_distance"]
    smoothing_σs = model["kernel_smoothing"]
    should be:
    maxdistkm = model["smoothing"]["kernel_maximum_distance"]
  - *smoothing_σs = model["smoothing"]["kernel_smoothing"]*
  - command "oqm wkf create_smoothing_per_zone" fails with Python 3.12 due to strange use of positional args and *openquake.baselib.sap* module. Check CLI entrypoint in *openquake.mbi.wkf.create_smoothing_per_zone.py: L10.*
    Fixed with changing some elements of the signature:
    create_smoothing_per_zone(fname_points, fname_polygons, folder_out, skip, use)
    to kwargs:
    create_smoothing_per_zone(fname_points, fname_polygons, folder_out, skip=skip, use=use)

sub module

- This module is particularly important because there is no standard way for modeling seismic hazard in subduction zones; outcomes depend heavily on user-defined choices such as slab geometry, seismicity classification, whether any segmentation is assumed, coupling assumptions, and a myriad of other parameters and assumptions. This module could help make these choices explicit and supports the standardization of subduction modeling procedures.
- However, the documentation relies heavily on internal data paths, making it impossible to reproduce the examples as written. As with other modules, a minimal working example or notebook with a synthetic case would greatly enhance usability and allow users to explore the workflow independently. Nevertheless, I could implement some of the procedures following the unit-tests and data provided there, but this is at no means optimal for potential users.

smt module

- The inclusion of demo code and data significantly supports user onboarding, and the code itself is succinct, modular, and well unit-tested. This level of coverage and clarity makes the module one of the most mature within the toolkit, perhaps due to its well-defined scope.
- Unfortunately, my knowledge is limited in ground-motion processing and I was unable to test the module with my own data, but it is evident that the *smt* module is both relevant and in good condition. I found no inconsistencies between the manuscript and the documentation.

fnm module

- The FNM module presents a highly interesting and state-of-the-art approach to modeling complex multi-fault ruptures. The code seems well written at first glance, but the absence of documentation and runnable tutorials—notebooks that lack data—makes it currently impossible to follow the intended workflow or evaluate the code's implementation. For a meaningful review in the future, I recommend the inclusion of initial documentation, a working example, and references to the methods from which the implementation is derived.

plt module

- This module has demo data but lacks of documentation. I could run some notebooks, but experienced issues with jupyter failing to load gmt shared libraries.

Manuscript Review

- Line 10-12 *"...(OQ-MBTK), a collection of functions for constructing probabilistic seismic hazard models. This toolkit encompasses a wide array of functions essential for hazard model development".*
    - These sentences are slightly repeated. Please modify.
- Line 16 *"... and carry out thorough sensitivity analyses"*
    - Do the authors refer to sensitivity in the hazard space? I couldn't find this feature in the toolkit. Please specify
- Line 26 *"... entails the collection of various information, its homogenization, pre-processing, and final use"*
    - Perhaps add: *".., pre-processing, composition (or ensembling, combination), and final use"*
- Line 26-28 *"The model building process consists of the application of different processing steps that, from the basic information, prepare parts of a hazard model input"*

    - This phrase is repeated too much in the introduction, and this lines in particular could be removed, since the idea is specified in L30.

- Across the manuscript *"e.g. "*
    - Please replace to "e.g.," for final submission
- Line 64 *"(bringing developments back to a general framework).... Such an approach makes the use of new methods straightforward within other projects."*
    - This sentence is too vague. Perhaps specify the benefits that open frameworks have, from code reusability and validation by the community.
- Line 72 *"FM"*
    - FM could be unclear for the reader. Please explain or provide the link from the company's website.

- Line 75-76 "The disadvantage of this approach included the need to maintain complicated software for running various Jupyter notebooks in a single run and an excessively large data structure which contained all the original information as well as the intermediate and final results of the workflow"
    - Please elaborate further on the need that OQ-MBTK satisfy, perhaps linking these problematics (and others) of the old MBTK to the design/architecture choices and improvements that the current MBTK has.

- Section 2 and 3.
    - Section 3 could be easily merged into Section 2, since both are jointly explaining the initial idea, development, current state and uses of the toolkit. With a good section title, both sections could be merged into one.

- Section 4.x
    - I recommend reordering the sub-sections similar to the logical order presented in Fig1 (e.g., cat, mbt, wkf, sub, fnm, smt, man, ghm, plt).

- Line 143 *"tools for analysing the completeness of a catalogue, deriving MFDs from catalogues or slip rates, and smoothing of seismicity based on past earthquake locations"*
    - I would suggest placing here the references corresponding to the implemented methods (e.g., Weichert 1980; Frankel 1995; Helmstetter et al 2007; and others)

- Section 4.6
    - I recommend to briefly mention the main challenges of modeling subduction zones for PSHA, provide an overview of the process that comes after the classification of evens in MBT (just a couple of lines for user reference), and the resulting output -- e.g., segmented fault surfaces with GR parameters, floating ruptures, etc.

- Section 5
    - It would be good to also mention about the license type of MBTK, that the package is extensively unit-tested, and that fulfills software development QA, such as code versioning and code review between GEM members.

- Line 270 *"reproducibility"*
    - As mentioned in the overall review, it is critical to discuss computational reproducibility strategies for rapidily evolving dependencies (e.g., geopandas breaking code in my review). Particularly critical due to MBTK not supporting *conda* builds, therefore having an additional overhead on managing shared libraries. Similarly, documentation and example could be provided with Docker for the reproducibility of a SHM workflow.

Reviewer 2 – Peter Powers

Review of "**The OpenQuake Model Building Toolkit: A suite of tools for building components of a seismic hazard model**"
Peter Powers
Aug 19, 2025

The authors present a 'Software Report' paper that largely mirrors the documentation available online for the OpenQuake Model Building Toolkit (MBTK). The paper is appropriate for publication in Seismica and will be a useful resource for those wishing to learn about or use OpenQuake tools. If authors and editors wish to proceed with publication of the paper largely in it's current form, that's probably ok, but there are two significant ways in which the paper can be improved.

1) Structure

As a practitioner of PSHA, I had a difficult time understanding the flow of the paper. This may be due in part to lifting text in the order presented from the MBTK online documentation (I don't wish to recommend refactoring the online documentation, but I think that exercise would prove useful for the long-term growth and support of the OQ MBTK). The 'module overview' section of the paper starts with a catalog processing module (4.1), which is one of the most common first steps in creating a PSHA input model. But then the following sections are telling me how to assemble a global mosaic (4.2), and then how to analyze my model (4.3)... I don't even have a model built yet! Then we get to fault and subduction zone modeling, but interspersed with plotting, workflow, and ground motion tools (4.5-4.9). It's pretty confusing.

I'd suggest strengthening the introduction with a concise description of what constitutes a PSHA model and example steps one might take to construct a model (the example doesn't have to be expansive; it could be restricted to an active crust tectonic setting). In walking through those steps, the authors could highlight the relevant modules to use and the order in which a typical user might want to use them. Then details of the modules in order of use follow.

2) Writing

I found much of the writing to be a little awkward or verbose (I give examples from and suggestions for the introduction below). Unless it violates internal policy, I would strongly suggest bring AI tools to bear on revising and tightening up the writing.

Lines 22-29 (possible cleaned up alternative) The former includes a comprehensive description of the position, geometry and properties all earthquake sources and associated epistemic uncertainties. The latter describes the models adopted to compute ground motions at a site and related epistemic uncertainties. The SSC and GMC collectively form the PSHA input model. Preparation of the SSC and GMC components entails the collection, pre-processing and homogenization of earthquake and ground motion information into hazard model inputs. Data frequently considered when preparing the SSC includes earthquake catalogues, geodetic data ...

Line 33 Traditional causes considered include modeling (1) distributed seismicity in the stable and actively deforming continental crust, (2) shallow crustal faults and fault systems, and (3) interface and intraslab sources in the subduction environment.

Line 34 Variants in each of these broad categories are present. (unnecessary)

Line 37 "on the other had," (unnecessary)

Line 47 Reproducibility lends credibility and is a desirable...

Line 60 There are assorted reasons that can explain this lack. (awkward)

Dear reviewers,

We thank you for your time and consideration. Your feedback has helped us to improve this manuscript, and the feedback from Reviewer A has helped us to also improve the mbtk itself.

Broadly, we revised the ordering of sub-sections in the manuscript to better reflect a logical order of use. We also revised the introduction and the writing in general to improve clarity. Below, we respond to each of the specific comments (in blue italics).

—---

Reviewer A

I have reviewed the Software Report article describing the OpenQuake Model Building Toolkit (OQ-MBTK). This is a Python-based tool suite designed to create the input models for Probabilistic Seismic Hazard Analysis (PSHA). It guides users through the end-to-end process of constructing Earthquake Rupture Forecasts—Seismicity Source Model (ERF/SSM)-- and calibrating Ground-Motion Models (GMM), ensuring consistency and reproducibility throughout. The final outputs are OpenQuake Engine-compatible input files to perform traditional hazard calculations. OQ-MBTK is aimed at researchers and hazard practitioners, particularly for those constructing models (e.g., National, Regional PSHA models) with reproducibility requirements, that is, reproducing hazard results from its constituent input data—catalogs, fault databases and strain-rate maps—through consistent computational steps, methods and code.

The OpenQuake ecosystem is widely used by researchers around the world, and efforts like the OQ-MBTK to standardize the construction of hazard are highly valuable. It will facilitate the development and reproducibility of new models, as well as serving as training resource for future engineering seismologists. Therefore, I consider this toolkit to play an important role in advancing the field. It also offers a platform to share, review and develop the implementation of new methods to build ERFs and GMMs. This review assessed the manuscript, documentation, and codebase, while also implementing an own model based on the provided materials.

The article is well written and effectively presents an overview of the toolkit's capabilities. Some aspects, detailed in the *Manuscript Review* section, could be improved—particularly the inclusion in the manuscript of two modules intended for internal GEM use rather than public deployment (see also Technical Review, **ghm** and **man** modules). Overall, the documentation is quite clear, but varies in quality across modules and lacks comprehensive tutorials/use cases that combine instructions, data and configuration files. A complete reference for all available configuration arguments of config-based modules is unfortunately missing.

The codebase is of high quality: modular, well-structured and extensively unit-tested. It follows open-source practices, is released under an open license, and is architected for extensibility. This package clearly builds on years of development and practical insights accumulated within GEM and consolidates numerous methods and practices used in the construction of PSHA models. While the overall design is robust, there are few areas where the user experience and codebase could be improved. The main caveat is the mixed use of declarative (configuration-based) and imperative (step-by-step) approaches in some modules, which can burden users with tracking numerous intermediate files without centralized guidance. Many modules, notebooks and functions in the toolkit are not kept up to date, or are redundant, and could be essentially removed for better maintainability (and re-implemented when they can be effectively used by the user). There are some minor issues with dependencies (described below). Whereas one of the strong points of this manuscript is assisting the hazard models' reproducibility, there is no comment or documentation about computational reproducibility (e.g., system and library dependencies). Nonetheless, this is definitely a publication-worthy piece of software, and I highly recommend its acceptance once the noted issues have been addressed. I also understand that this is an evolving software, and I would be open to some of my recommendations being deferred to a Roadmap when justified.

I suggest the manuscript to be accepted after minor revisions. As a minimum requirement, I recommend the authors ensure that all modules presented in the manuscript includes at least basic documentation and a testable use case. These should involve example data, configuration files, and clear instructions on how to execute the workflow--many of which already exists in notebooks and unit tests scattered across the repository. If data size is a concern for repository cleanness, authors may consider hosting use-case datasets externally and linking them from the documentation—README files, rather than bundling them directly into the repository.

Looking forward to see this manuscript published,

Pablo Iturrieta

Technical Review

Here, I acted both as reviewer and partial beta-tester while attempting to integrate the toolkit into my own workflows. This review is by no means exhaustive (and is presented in the order in which issues emerged rather than by priority) as the toolkit is extensive, and due to the limited documentation in several modules, it was often difficult to distinguish between actively maintained features and legacy components. Nonetheless, my review is not a professional tester effort, but I hope some observations compiled here assist the authors in further polishing the software. Some essential comments herein, particularly related to documentation and demos, should indeed be refined and addressed by the authors.

We very much appreciate this detailed technical review. We feel that this was well beyond what we could ask of a reviewer, and we very much appreciate the time and effort invested. We have made improvements to the mbtk to address these points, in particular, improving documentation, adding notebook examples, and removing older code where it is not currently used. We have included responses below, including links to the appropriate pull requests addressing the issues.

Installation

- Installation was straightforward using the provided instructions.

- However, dependency issues persist with MBTK and GMT. Although the documentation states that GMT is only required for plotting, GMT becomes a de facto dependency across all modules:
    - **(Details)**: Running any mbtk command (e.g., oqm cat merge) triggers GMT dependencies because:
        - pygmt is always installed via requirements-py*-linux.txt;
        - All submodules in mbi are imported (via engine.openquake.baselib.sap), eventually loading pygmt through openquake.sub.get_profiles_from_slab2pt_geojson.py
        - If pygmt fails to find the shared libgmt.so library, the code breaks due to improper exception handling in get_profiles_from_slab2pt_geojson.py:L41.
- Even with GMT libraries installed via apt, shared libraries (e.g., */usr/lib/libgmt.so*) are not detected by the installed pygmt version due to version incompatibilities. This required manual creation of symbolic links to allow pygmt to recognize them (see: pygmt install issues).
- Geopandas version conflicts: Geopandas now resolves to >=1.0.0, which breaks code relying on deprecated syntax (e.g., gpd.sjoin(..., op='within') should now use predicate='within'). See the GeoPandas 1.0.0 changelog
- *oq-mbtk/notebook/Dockerfile* fails building

Recommendations:

- Avoid loading GMT unless explicitly required; capture exceptions where pygmt may be imported.
- Provide clear installation instructions for GMT/pyGMT, including version requirements.
- Add unit tests that simulate OQ-MBTK CLI usage. Would have the benefit of triggering *oq-engine.sap* library loading all modules, so importing errors can be catch.
- Update Geopandas usage and version requirements accordingly.

We improved the documentation for the installation, including explanations of when GMT/pyGMT/Julia might be useful and links to installation guidelines to ensure that these are installed properly if required. We also tidied the exception handling in `get_profiles_from_slab2pt0_geojson.py` to ensure that pygmt remains optional (but perhaps recommended).

The mbtk has recently received updates to upgrade to numpy2, and the requirements for the mbtk were updated accordingly during this procedure, adding version numbers for packages where possible. The dockerfile has also been updated.

PRs: https://github.com/GEMScienceTools/oq-mbtk/pull/515 (documentation); https://github.com/GEMScienceTools/oq-mbtk/pull/522/files (updates to requirements)

cat module

- I was able to follow the documentation and complete an analysis with my own data.
- Tutorials are needed to be included, including example data, configuration files, and instructions (it could be accompanying the existing documentation with the actual used data).
- Documentation is needed to explain (all) the possible configuration files parameters.
- It is not straight-forward for users to import their own catalog formats.
    - No link is provided to the GEM Catalog format; converting to ISF is a heavy overhead.
    - The latest HMTK manual may be outdated (e.g., missing magnitudeType support). An inset in this same OQ-MBTK *cat* documentation containing the accepted format would be excellent.
    - Serialized catalogs from openquake.hmtk.catalogue.Catalogue.write_catalogue() cannot be read back in. I understand that harmonising all Reader/Writers across openquake would be cumbersome, but I suggest to write in each module's documentation what will be the accepted I/O methods.
    - ISC-GEM catalog (from ISC website) cannot be loaded directly, since it is probably not well formatted. I suggest including a parser in the library, because it will be probably the most used proxy catalog by future users.
- Geopandas 1.0+ breaks polygon-based catalog filtering:
    - Example: openquake.cat.hmg.merge.py:L136 uses op='intersects', which should be predicate='intersects'.
- The homogenisation workflow is not consistent: some times I/O arguments are passed through the config files, sometimes through the CLI, which can burden the user with heavy file tracking. This could be harmonised and avoid relying on *.bash* scripts. Same applies for WKT module.
- Switching between TOML configs and manual Python scripting during homogenisation makes workflows harder to reproduce (and require user's familiarity with both interfaces). as future users must manually track configuration files and scripted steps. While I understand iterative analysis is expected in regressions (therefore scripting becomes necessary), the documentation should better reflect how user choices should be placed in the final configuration file to maintain reproducibility.
- Utility code for reading/querying .h5 files that is described in the docs should be part of the cat API.

The documentation for the cat module has been updated to better describe catalogue types and converters between them, for example, to clarify that it is straightforward to read in a user's own catalogues by using specific column names and clarifying compatibility with the older HMTK-format catalogues.

We added a parser for the ISC-GEM catalogue that parses the catalogue to the standard csv format.

We also added some clarifications to the workflow for homogenisation so it is clearer that the regressions are indeed iterative and that the final values should be added to the toml.

The config files are required for the functions; use of the bash script is recommended but not necessary. In our experience, bash scripts help to keep track of changes in the modelling and make updates to the catalogues easier to implement consistently, though we understand this may not be true for all users. As such, we updated the documentation to make clear that the bash script approach is not necessary.

PR: https://github.com/GEMScienceTools/oq-mbtk/pull/515 (documentation)

ghm module

- While the functionalities described in the manuscript are highly relevant for anyone working with multiple hazard models—whether the GEM Mosaic or other model combinations—I'm not in favor of publishing tools that are not meant for general use. In my opinion, it is also in the authors' convenience to create additional repositories for this purpose, so the MBTK repo is kept clean and better maintainable.
- However, this module could have a large impact if it is generalized to support broader use cases, such as having overlapping hazard models, or having global or regional mosaics.

We agree that perhaps these tools are more internally useful, but we also wish to make the process used at GEM for homogenising the hazard results as transparent as possible. For the time being, and especially as we work towards a new release of the global map this year, we keep these tools as they are, though we will revisit them in the near future.

man module

- This module could have essential functionality to evaluate, a-posteriori, the resulting ERF/SSM files of OQ-MBTK. However, the documentation, tutorials and codebase is not at pair with other modules.
- Same as with *ghm*, I don't agree on its inclusion on the manuscript if it is not general use.
- If the package is on early development stages but intended to grow, I recommend to keep it in the manuscript and showcase the existing functionality in tutorials with simple data (already contained in "man.notebooks", albeit currently with no data and hardcoded paths), so users may be able to use its functionality on their workflows. Or move its functionality to *plt*.

There are several very useful general interest functions in here, though there are also a lot of older materials that are no longer widely used. We are in the process of updating these *repositories, but in order to ensure that we don't remove important functionality for existing users this process is a little slow. We hope to resolve this in the near future.*

PR: https://github.com/GEMScienceTools/oq-mbtk/pull/497 (tidying man)

mbt module

- The module, as outlined in the manuscript, should be one of the most important modules in the toolkit, but unfortunately currently lacks documentation.
- I was able to adapt some of the notebooks to my own workflow and data, could run some of the routines with a lot of trouble due to old dependencies and breaking code.
- For this manuscript, I would suggest releasing the essential use cases in the documentations. These could be straightforwardily adapted from the existing notebooks (perhaps including them in the docs with the package
- nbsphinx), once this module has been updated to the requirements of the current MBTK release (i.e., will managing OqtProject/OqtModel be a feature? Otherwise remove) .

This module contains many old notebooks that were the original mbtk. In tidying the repo, we proposed removing the old notebooks to an archived version of the repository, along with the OqtProject and OqtModel features that are no longer widely used by the hazard team. However, these tools are still used by some early adopters of the mbtk and so we decided to retain them and revisit some aspects of these in the future.

Finally, we added documentation for the remaining mbt functions, though there is a significant overlap with the wkf functions below.

PRs:https://github.com/GEMScienceTools/oq-mbtk/pull/496 (tidying mbt)

https://github.com/GEMScienceTools/oq-mbtk/pull/514 (tidying mbt)

https://github.com/GEMScienceTools/oq-mbtk/pull/515 (documentation)

wkt module

- I followed the documentation and completed an analysis with my data.
- Documentation is needed to explain (all) the possible configuration files parameters.
- This module overlaps heavily with the intended **mbt** module's purpose. Following the manuscript description, much of the code under *openquake.wkt* should be under *openquake.mbt* (e.g., *compute_gr_params.py*, *analysis_hypocentral_depth.py*, smoothing algorithms, etc).
- I recommend to keep the distinction between config-type coding (declarative) under WKT and move all the functions to MBT (for python scripting). Essentially, **wkt** should work as the config-based wrapper for **mbt** workflow.
- API is hard to follow when tracking all intermediate file locations. I have no easy solution for this, but it would definitely be easier to just have standard-paths defined from the configuration file.

- It is not clear to me the reasoning behind the use of the *PSHAModelBuilder* library when it contains only one method that could easily be included in *OQ-MBTK*.
- I remain unconvinced that the performance benefits gained from using external Julia packages justify the added complexity and dependency overhead. Particularly because the Helmstetter algorithm (in python) is in theory computationally more expensive than Frankel's (in Julia)-- involving distance calculation between catalog-grid and catalog-catalog, rather than just catalog-grid, respectively. This choice detracts from the otherwise streamlined architecture of the toolkit.
- Minor bugs or error in docs:
    - In "Some notes on setup" docs: BIND and BIN1 should point only to: os.path.join('/Users', 'kjohnson', 'GEM', 'oq-mbtk', 'openquake', 'bin').
    - In Julia smoothing script (PSHAModelBuilder.src.seismicity.smoothing.jl:L83-84), when using the example .toml from the docs:

      maxdistkm = model["kernel_maximum_distance"]

      smoothing_σs = model["kernel_smoothing"]

      should be:

      maxdistkm = model["smoothing"]["kernel_maximum_distance"]
    - *smoothing_σs = model["smoothing"]["kernel_smoothing"]*
    - command "oqm wkf create_smoothing_per_zone" fails with Python 3.12 due to strange use of positional args and *openquake.baselib.sap* module. Check CLI entrypoint in *openquake.mbi.wkf.create_smoothing_per_zone.py: L10.* Fixed with changing some elements of the signature:

      create_smoothing_per_zone(fname_points, fname_polygons, folder_out, skip, use)

      to kwargs:

      create_smoothing_per_zone(fname_points, fname_polygons, folder_out, skip=skip, use=use)

We updated the documentation to remove the noted errors and to include more details where suggested or found necessary.

In general, the intent for the wkf/mbt was as you state, with functions in the mbt and the workflow wrappers in the mbi module, but some steps have instead been added in the wkf module, either because they seem more relevant to a workflow approach rather than an essential feature on their own (i.e., the hypocentral depth analysis) or because of some poor structuring on our part. We have begun work on tidying these, but this is not trivial and will take some time.

We agree that the benefits of the Julia smoothing are perhaps a little redundant with the much slower adaptive option, and it is our intention to add Python versions of both the smoothing and the rate redistribution (currently also uses the PSHA model builder Julia tools). Unfortunately, these are not yet available, so the Julia steps remain. However, we hope that the documentation improvements make these steps less troublesome for users in the meantime.

Our priorities in this preliminary phase were to add the methodologies and workflows needed for model-building. Building on this, our future priorities will aim to improve structure to more easily preserve internal consistency. In this regard, your suggestions are very helpful, and we would look to include them in future improvements to the tools.

PR: https://github.com/GEMScienceTools/oq-mbtk/pull/515 (documentation)

sub module

- This module is particularly important because there is no standard way for modeling seismic hazard in subduction zones; outcomes depend heavily on user-defined choices such as slab geometry, seismicity classification, whether any segmentation is assumed, coupling assumptions, and a myriad of other parameters and assumptions. This module could help make these choices explicit and supports the standardization of subduction modeling procedures.
- However, the documentation relies heavily on internal data paths, making it impossible to reproduce the examples as written. As with other modules, a minimal working example or notebook with a synthetic case would greatly enhance usability and allow users to explore the workflow independently. Nevertheless, I could implement some of the procedures following the unit-tests and data provided there, but this is at no means optimal for potential users.

This module reflects several different approaches to modelling subduction zones, and as with other modules, some of the notebooks included were quite out-dated! As such, we have removed these.

We have also updated the documentation to be more current and may soon add new example notebook(s).

PR: https://github.com/GEMScienceTools/oq-mbtk/pull/515 (documentation)

smt module

- The inclusion of demo code and data significantly supports user onboarding, and the code itself is succinct, modular, and well unit-tested. This level of coverage and clarity makes the module one of the most mature within the toolkit, perhaps due to its well-defined scope.
- Unfortunately, my knowledge is limited in ground-motion processing and I was unable to test the module with my own data, but it is evident that the *smt* module is both relevant and in good condition. I found no inconsistencies between the manuscript and the documentation.

fnm module

- The FNM module presents a highly interesting and state-of-the-art approach to modeling complex multi-fault ruptures. The code seems well written at first glance, but the absence of documentation and runnable tutorials—notebooks that lack data—makes it currently impossible to follow the intended workflow or evaluate the code's implementation. For a meaningful review in the future, I recommend the inclusion of initial documentation, a working example, and references to the methods from which the implementation is derived.

Fermi is a particularly new module, so at the moment the documentation and tutorials are lacking. We have added to the manuscript to clarify that this is still very new, and a separate paper on fermi and the fault-modelling processes is expected in the near future.

plt module

- This module has demo data but lacks of documentation. I could run some notebooks, but experienced issues with jupyter failing to load gmt shared libraries.

After reviewing this module and its current use, we have decided to remove this section from the paper. We will leave the code in the mbtk available for legacy reasons, and may update it in the future.

Manuscript Review

- Line 10-12 *"...(OQ-MBTK), a collection of functions for constructing probabilistic seismic hazard models. This toolkit encompasses a wide array of functions essential for hazard model development".*
    - These sentences are slightly repeated. Please modify.

We updated the text to: "To facilitate this process, we have developed the OpenQuake Model Building Toolkit (OQ-MBTK), a collection of functions for constructing probabilistic seismic hazard models, enabling users to start from catalogue and fault data and sequentially step through the model building process to produce hazard inputs compatible with the OpenQuake (OQ) Engine.

- Line 16 *"... and carry out thorough sensitivity analyses"*
    - Do the authors refer to sensitivity in the hazard space? I couldn't find this feature in the toolkit. Please specify

Though we use the tools in the mbtk in our sensitivity analyses, there are no tools specifically for this currently in the mbtk. We have modified the text as follows: "and carry out thorough sensitivity analyses by easily investigating the consequences of model changes."

- Line 26 *"... entails the collection of various information, its homogenization, pre-processing, and final use"*
    - Perhaps add: *".., pre-processing, composition (or ensembling, combination), and final use"*
- Line 26-28 *"The model building process consists of the application of different processing steps that, from the basic information, prepare parts of a hazard model input"*

- This phrase is repeated too much in the introduction, and this lines in particular could be removed, since the idea is specified in L30.

We updated these lines to: "Preparation of the SSC and GMC components entails collecting, pre-processing and homogenising earthquake and ground motion information into hazard model inputs. For the SSC, the primary data sources are earthquake catalogues, as well as geodetic (e.g., from space geodesy), geological, and paleoseismological information that shed light on active faults and deformation patterns (e.g., fault types, geometries, and slip rates)."

Across the manuscript *"e.g. "*

- Please replace to "e.g.," for final submission

Done

- Line 64 *"(bringing developments back to a general framework).... Such an approach makes the use of new methods straightforward within other projects."*
    - This sentence is too vague. Perhaps specify the benefits that open frameworks have, from code reusability and validation by the community.

Reworded to: "though we think that new improvements should be brought back to a more general framework, making the use of new methods straightforward within other projects and for other modellers."

- Line 72 *"FM"*
    - FM could be unclear for the reader. Please explain or provide the link from the company's website.

We have added a link (https://www.fm.com/) and also a clarification that FM are a long-standing sponsor of GEM

- Line 75-76 "The disadvantage of this approach included the need to maintain complicated software for running various Jupyter notebooks in a single run and an excessively large data structure which contained all the original information as well as the intermediate and final results of the workflow"
    - Please elaborate further on the need that OQ-MBTK satisfy, perhaps linking these problematics (and others) of the old MBTK to the design/architecture choices and improvements that the current MBTK has.

Added "The new OQ-MBTK tools instead provide functions that can be used more flexibly by importing directly in python, in jupyter notebooks, or directly from the command line, as described in section 6.

- Section 2 and 3.

- Section 3 could be easily merged into Section 2, since both are jointly explaining the initial idea, development, current state and uses of the toolkit. With a good section title, both sections could be merged into one.

We merged sections 2 and 3 into a single section, titled "The OQ-MBTK pilot study and applications"

- Section 4.x
  - I recommend reordering the sub-sections similar to the logical order presented in Fig1 (e.g., cat, mbt, wkf, sub, fnm, smt, man, ghm, plt).

We reordered the sub-sections as suggested

- Line 143 *"tools for analysing the completeness of a catalogue, deriving MFDs from catalogues or slip rates, and smoothing of seismicity based on past earthquake locations"*
  - I would suggest placing here the references corresponding to the implemented methods (e.g., Weichert 1980; Frankel 1995; Helmstetter et al 2007; and others)

We added references to the implemented methods where possible

- Section 4.6
  - I recommend to briefly mention the main challenges of modeling subduction zones for PSHA, provide an overview of the process that comes after the classification of evens in MBT (just a couple of lines for user reference), and the resulting output -- e.g., segmented fault surfaces with GR parameters, floating ruptures, etc.

Added: The SUB model offers tools for constructing subduction earthquake sources. The tools are iteratively improved alongside subduction zone science (e.g., allows to impose or exclude segmentation of the subduction interface) and to ensure the rupture geometries produced are compatible with PSHA applications (e.g., risk analysis; scenarios).

Added: "After regionalisation, the events classified as belonging to the slab and interface are used to define suitable MFDs for these sources, with the option of considering tectonic limits for the upper magnitudes. The SUB tools can then be used to generate the subduction sources, including 3D fault surfaces with floating ruptures for the interface, with appropriate MFDs derived from seismicity, tectonics or a hybrid model (i.e., a la Youngs and Coppersmith, 1985), and non-parametric sources (e.g., predefined rupture geometries and their rates) that model seismicity in the slab."

- Section 5
  - It would be good to also mention about the license type of MBTK, that the package is extensively unit-tested, and that fulfills software development QA, such as code versioning and code review between GEM members.

We added the following to Section 5: " The package is extensively unit-tested and all code is subject to review among GEM members. The OQ-MBTK is released under a GNU Affero general public license 3.0 ."

- Line 270 *"reproducibility"*
  - As mentioned in the overall review, it is critical to discuss computational reproducibility strategies for rapidily evolving dependencies (e.g., geopandas breaking code in my review). Particularly critical due to MBTK not supporting *conda* builds, therefore having an additional overhead on managing shared libraries. Similarly, documentation and example could be provided with Docker for the reproducibility of a SHM workflow.

We have updated the requirements to ensure that dependencies have specific version requirements.

Reviewer B

Review of "**The OpenQuake Model Building Toolkit: A suite of tools for building components of a seismic hazard model**"
Peter Powers
Aug 19, 2025

The authors present a 'Software Report' paper that largely mirrors the documentation available online for the OpenQuake Model Building Toolkit (MBTK). The paper is appropriate for publication in Seismica and will be a useful resource for those wishing to learn about or use OpenQuake tools. If authors and editors wish to proceed with publication of the paper largely in it's current form, that's probably ok, but there are two significant ways in which the paper can be improved.

1) Structure

As a practitioner of PSHA, I had a difficult time understanding the flow of the paper. This may be due in part to lifting text in the order presented from the MBTK online documentation (I don't wish to recommend refactoring the online documentation, but I think that exercise would prove useful for the long-term growth and support of the OQ MBTK). The 'module overview' section of the paper starts with a catalog processing module (4.1), which is one of the most common first steps in creating a PSHA input model. But then the following sections are telling me how to assemble a global mosaic (4.2), and then how to analyze my model (4.3)... I don't even have a model built yet! Then we get to fault and subduction zone modeling, but interspersed with plotting, workflow, and ground motion tools (4.5-4.9). It's pretty confusing.

We updated the order of the sub-sections to follow that shown in Figure 1 (i.e., to follow a logical progression for hazard modelling rather than listing the sub-sections alphabetically)

I'd suggest strengthening the introduction with a concise description of what constitutes a PSHA model and example steps one might take to construct a model (the example doesn't have to be expansive; it could be restricted to an active crust tectonic setting). In walking through those steps, the authors could highlight the relevant modules to use and the order in which a typical user might want to use them. Then details of the modules in order of use follow.

2) Writing

I found much of the writing to be a little awkward or verbose (I give examples from and suggestions for the introduction below). Unless it violates internal policy, I would strongly suggest bring AI tools to bear on revising and tightening up the writing.

We implemented the suggested changes below, and reviewed the manuscript again to ensure clarity.

Lines 22-29 (possible cleaned up alternative) The former includes a comprehensive description of the position, geometry and properties all earthquake sources and associated epistemic uncertainties. The latter describes the models adopted to compute ground motions at a site and related epistemic uncertainties. The SSC and GMC collectively form the PSHA input model. Preparation of the SSC and GMC components entails the collection, pre-processing and homogenization of earthquake and ground motion information into hazard model inputs. Data frequently considered when preparing the SSC includes earthquake catalogues, geodetic data …

Updated as per this suggestion (see tracked changes)

Line 33 Traditional causes considered include modeling (1) distributed seismicity in the stable and actively deforming continental crust, (2) shallow crustal faults and fault systems, and (3) interface and intraslab sources in the subduction environment.

Updated as per this suggestion (see tracked changes)

Line 34 Variants in each of these broad categories are present. (unnecessary)

Removed

Line 37 "on the other had," (unnecessary)

Removed

Line 47 Reproducibility lends credibility and is a desirable…

Updated as per this suggestion (see tracked changes)

Line 60 There are assorted reasons that can explain this lack. (awkward)

Updated to: “There are numerous reasons that can explain this, though the most probable is a general preference in the hazard modelling community for developing in-house tools, to address more specific needs emerging in each project.”