Reproducibility review of: Exploring MapSwipe as a Crowdsourcing Tool for (Rapid) Damage Assessment: The Case of the 2021 Haiti Earthquake

Nina Wiedemann

May 29, 2023

This report is part of the reproducibility review at the AGILE conference. For more information see https://reproducible-agile.github.io/. This document is published on OSF at https://doi.org/10.17605/osf.io/m5bhk. To cite the report use

Wiedemann, N. (2023). Reproducibility review of: Exploring MapSwipe as a Crowdsourcing Tool for (Rapid) Damage Assessment: The Case of the 2021 Haiti Earthquake. https://doi.org/10.17605/osf.io/m5bhk

1. Reviewed paper

Groß, S., Herfort, B., Marx, S., and Zipf, A.: Exploring MapSwipe as a Crowdsourcing Tool for (Rapid) Damage Assessment: The Case of the 2021 Haiti Earthquake, AGILE GIScience Ser., 4, 5, https://doi.org/10.5194/agile-giss-4-5-2023, 2023

2. Summary

The paper comes with a GitHub repository that mainly includes a jupyter notebook for reproducing the results. The notebook could be executed right away and is very well-documented. The outputs of the notebook include examplary data as well as all main plots from the paper. On request, the authors further added the raw QGIS files for reproducing map-based visualizations, and deposited the code with a DOI (https://zenodo.org/badge/latestdoi/581154837). Overall, the paper is fully reproducible.

3. Reproducibility reviewer notes

3.1. Repository structure and installation

A Readme file in the code base refers to the paper and describes the main components of the repository. Instructions for installing a virtual environment with conda were added in the context of the reproducibility review via a pull request, as well as the license. Only few Python libraries are necessary to run the code. All necessary data is provided within the repository.

There are two folders, *Maps* and *jupyter*. The *Maps* folder contains the raw QGIS project files for generating the map visualizations in the paper, while the jupyter folder is divided into 1) a notebook for demonstration purposes and 2) the required code in a separate file (`builder.py`). This structure is very suitable since it assures that the user has access to all code, but can nevertheless reproduce the main results in a simple and uncluttered notebook.

3.2. Demonstration notebook

The notebook is well-documented and guides the reader through the code. In the beginning, the data is loaded, which takes a moment as described in the text of the notebook. The data can be inspected and is explained briefly. The next cell creates a confusion matrix between the reference data and the MapSwipe results. This result does not appear in this form in the paper, but can be related to other results shown in the paper. The remaining cells reproduce Figure 6 from the paper which shows different metric by the positive-threshold, as well as Figure 10 from the paper which shows Cohen's Kappa between MapSwipe and CEMS labels. The plots are exactly the same as shown in the paper, and are therefore not included in this report.

The notebook is a bit black-box since it uses all functions from the builder.py file, so the user would need to check that file for more information about the code. The code in builder.py is also well-documented with docstrings for each function.