**MOOD Project**

[**https://mood-h2020.eu**](https://mood-h2020.eu)

**Description of Tool Supporting Geospatial Data Integration**

Within the MOOD project, the integration of the structured disease data with environmental covariate data involves handling geospatial references according to different representations, since the covariates are usually made available in the form of georeferenced raster datasets (e.g., GeoTIFF files), and the structured disease data is instead represented as tabular data (e.g., CSV or Excel files) where one of the attributes corresponds to place names.

To handle challenges related to geospatial integration, one specific software package was developed within MOOD with the following functionalities:

* Convert placenames into structured geographic information (i.e., geocode place names into GeoJSON geometries).
* Convert point locations or GeoJSON geometries into placenames (i.e., reverse geocoding).
* Compute zonal statistics (i.e., aggregates associated to polygonal boundaries) from placenames and raster data.
* Apply the geocoding, reverse geocoding, and zonal statistical operations to tabular data stored in CSV or Excel files (i.e., adding additional columns with the corresponding placenames, geospatial footprints, or statistics).

This tool is publicly available in a Github repository[[1]](#footnote-1) and it uses data from two public sources, namely the gazetteer data from the Who’s on first project[[2]](#footnote-2), and the global population data from the Gridded Population of the World project[[3]](#footnote-3).

In brief, the Who’s on first gazetteer corresponds to a large list of places covering the entire planet, each with a stable identifier, a geospatial footprint in the form of a polygonal boundary, and some number of descriptive properties about that location. The gazetteer was derived from many public sources, including Geonames and the OpenStreetMap. The entire dataset is stored and indexed within a sqlite database, supporting efficient search (i.e., the software package that was developed offers APIs for geocoding and reverse-geocoding, which are implemented as queries over this sqlite database).

In turn, the Gridded Population of the World project offers consistent population counts for the entire planet, gridded with a resolution of 30 arc-seconds (approximately 1 km at the equator). We used this dataset to complement the information from the Who’s on first gazetteer, computing population counts for all gazetteer entries through the associated polygonal boundaries (i.e., we compute the sum of all raster cells that are contained within the polygonal footprint of each gazetteer entry).

The main functionalities in the *gazetteer-access* tool are as follows:

* The geocoding functionality takes placenames as input and returns polygonal boundaries from the matching gazetteer entries, giving preference to highly populated places in the case of ambiguous matches. This functionality can be accessed through a simple Python interface, or through a command line tool that can take CSV or Excel files as input, in which one of the attributes corresponds to the place names.
* The reverse geocoding functionality uses point-in-polygon or polygon intersection queries over the gazetteer entries. This functionality can also be accessed through a Python interface, or through a command line tool.
* Zonal statistics (e.g., averages or sums over polygonal boundaries) can be computed from raster data with basis on place names. Geocoding (or reverse geocoding) is used to transform the placenames (or latitude and longitude coordinates) into polygonal boundaries, and the rasterstats[[4]](#footnote-4) Python package is used to compute the aggregate statistics associated to the polygonal boundaries. Again, this functionality can be accessed through a simple Python interface, or through a command line tool that takes as input a CSV/Excel file (where one of the columns corresponds to placenames or to geospatial coordinates) and one GeoTIFF file, producing as output a new CSV/Excel file with the added zonal statistics.

The tool was developed with reusability and extensibility in mind, envisioning the support for different types of integration mechanisms and workflows. Until the end of the project, we still plan to make additional developments associated to this tool, specifically envisioning:

* Geocoding complex place references, including references that include spatial qualifies (e.g., *north of Lisbon*) or combinations of multiple names to aid in the disambiguation (e.g., *Paris, Texas* versus *Paris, France*).
* Handling of raster datasets with multiple layers (e.g., layer for different temporal snapshots) in the computation of zonal statistics.
* Considering sets of related place references (e.g., multiple placenames mentioned in a single document) for geocoding, instead of handling each placename separately.

Data integration within the MOOD project also involves handling temporal references, although in this case we only considered simple scenarios related to date normalization. We rely on an existing Python library named dateparser[[5]](#footnote-5) to interpret human readable dates in a variety of different formats, converting the temporal references into the ISO 8601 calendar date format.

1. <https://github.com/bgmartins/gazetteer-access> [↑](#footnote-ref-1)
2. <https://whosonfirst.org> and <https://geocode.earth/data/whosonfirst/combined/> [↑](#footnote-ref-2)
3. <https://sedac.ciesin.columbia.edu/data/collection/gpw-v4> [↑](#footnote-ref-3)
4. <https://pythonhosted.org/rasterstats/> [↑](#footnote-ref-4)
5. <https://dateparser.readthedocs.io/en/latest/> [↑](#footnote-ref-5)