Emiproc: A Python package for emission inventory processing

Constantin Lionel [1], **Brunner Dominik** [1], **Thanwerdas Joel** [1*], **Keller Corina** [1*], **Steiner Michael** [1*], **and Koene Erik** [1*]

1 Empa, Laboratory for Air Pollution / Environmental Technology, Switzerland * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 02 October 2024
Published: unpublished

Summary

Emission inventories are created by countries and regions to assess and address air quality and climate change targets. An emission inventory is a spatial dataset that reports the yearly amount of pollutants released into the atmosphere, often broken down by specific source sectors or individual sources. Atmospheric modellers use such inventories to simulate the transport of emitted species in order to compute their distribution and assess their potential impact on the environment. The simulations are often compared with measurements to verify if the declared emissions and their trends are consistent with the observed changes in the atmosphere, thereby enhancing confidence in the inventories.

Figure 1 presents an example of a gridded inventory.

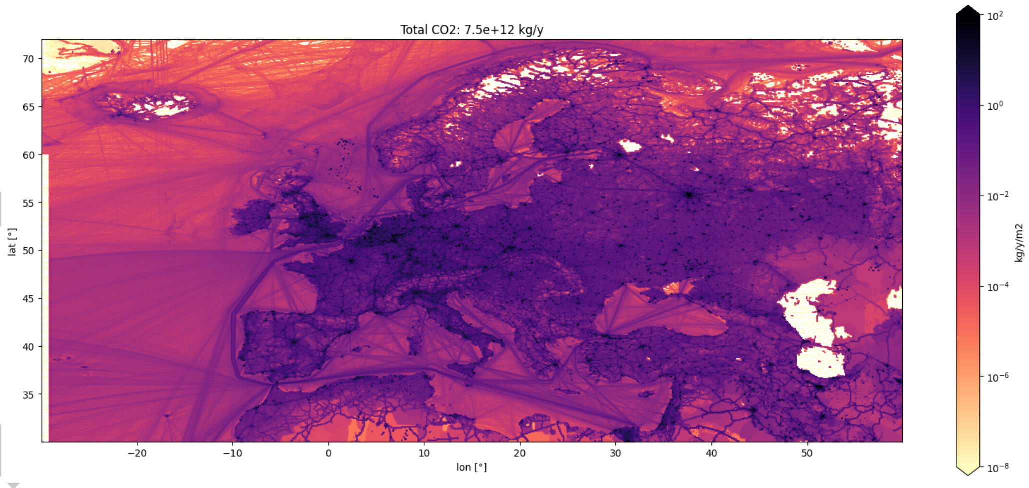

Figure 1: CO2 emissions of the year 2015 from the TNOGHGco2_v6 inventory produced by TNO .

Inventories are created in multiple different formats and resolutions, which makes it difficult to compare and use them in atmospheric transport models. `emiproc` is a Python package that provides tools for processing emission inventories, harmonizing datasets, and preparing such data as inputs for atmospheric transport models. The package provides functions for performing various operations on inventory data, such as remapping to different model grids, aggregating emissions by sector or pollutant, or scaling emissions based on future scenarios.

Emission input files can be generated in regular (e.g. hourly) intervals by applying sector- and country-specific temporal and vertical emission profiles. Alternatively, a small set of input files can be generated, which describe the sectorial gridded emissions and their temporal and vertical profiles. This set of files can then be read by an OEM (online emissions module) described in (Jähn et al., 2020), which applies the temporal and vertical scaling online during the model simulation. OEM is integrated into the atmospheric chemistry and transport models COSMO-ART and ICON-ART.

`emiproc` is designed to be flexible and extendable, allowing users to easily add custom functionality, to read new inventories or export data to other formats.

Statement of need

Emission inventory data can be represented in various formats and resolutions. For example, TNO (Dutch Organization for Applied Scientific Research) provides inventories which contains both, area emissions on a regular grid and point sources at their exact locations. Other inventories, such as one from the city of Zurich, are provided as vector data with various shapes depending on the category of the source. For example:

- Traffic emissions are represented as lines
- Building/industrial emissions are represented as point sources
- Private boats on the lake are represented as polygons

Atmospheric models require emission inventories to be in a specific file format. The atmospheric chemistry transport model ICON-ART (Schröter et al., 2018), for example, requires emissions on a semi-structured triangular grid. As input for OEM it also requires temporal profiles to scale the emissions with hourly, daily and monthly variability. As an other example, the Graz Lagrangian dispersion model (GRAL) can make direct use of emissions represented in vector format as line, point or rectangular sources. GRAL needs additional detailed information such as the height, the temperature or the gas exit velocity of a point source.

When modellers design their simulations, they are often interested in modifying the inventories. For example, they could do the following: scale the emissions based on future scenarios, aggregate emissions by sector or pollutant to simplify their simulations or combine multiple inventories to represent different sources such as anthropogenic and natural emissions. `emiproc` provides this functionality and has already been successfully applied for different use cases:

- (Steiner et al., 2024) produced emission files for ICON-ART-OEM based on the EDGARv6 inventory (Monforti Ferrario et al., 2021) (Emissions Database for Global Atmospheric Research).
- (Dönmez & Brunner, 2024) conducted urban climate simulation using emissions produced with `emiproc` for cities of Zurich and Basel.
- (Ponomarev & Brunner, 2024) used `emiproc` to nest the Zurich city inventory inside the Swiss national inventory and to further nest the Swiss inventory inside the European TNO inventory.

`emiproc` shares some of its functionality with another Python tool, `HERMESv3` (Guevara et al., 2019), which is also designed to process emission data and generate input files for atmospheric transport models. Compared to `HERMESv3`, which relies on specific configuration files, `emiproc` is more flexible, extensible and practical as it can be integrated in existing Python-based workflows.

History

An earlier version of `emiproc` was already published (Jähn et al., 2020), but it was limited to specific models and inventories. Starting in 2022 `emiproc` has been refactored to satisfy the requirements of high flexibility and modularity. This included major changes to code structure, the addition of new capabilities, a major performance increase for the task of spatial regridding, a comprehensive documentation and the addition of test examples.

Since then the package is regularly updated with new features and bug fixes.

Design

In order to use different types of inventories in atmospheric models, it is necessary to harmonize them. This is what the `emiproc` package is designed for.

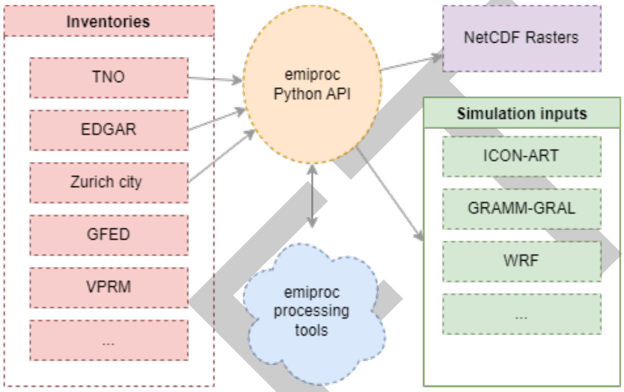

Figure 2: Design emiproc.

Thanks to the harmonization of the data, functions for additional data processing can easily be applied to the different inventories once loaded into `emiproc`. The processed inventories can be exported to any of the formats supported by the package.

The API of `emiproc` leverages the advantages of object-oriented programming. `emiproc` is built on top of the `geopandas` package (Jordahl et al., 2020), which allows storing the geometries of the emission maps and offers many functionalities related to geometric operations. Within emiproc, the emission data of the inventory is stored as a `geopandas.GeoDataFrame`.

Inventory

The main object is the `Inventory` object. Any inventory is a subclass of the base `Inventory`. `emiproc` provides many utility functions to process them. Typically, a processing function takes an inventory and some parameters as input and returns a new inventory with the processed data.

Example: the `group_categories` function regroups an inventory with a given set of source categories to a new inventory with another set of categories, based on a mapping provided by the user. This is useful to reduce the number of sources simulated or to use a standardized set of categories such as the GNFR (Gridded Nomenclature For Reporting) sectors from (European Environment Agency, 2023).

Grid

As inventories and models use different grids, `emiproc` defines a grid and projection through the `Grid` object. Many use cases are covered by the `RegularGrid` child class, which represents a standard latitude-longitude grid. On the other hand, users of models with more complex geometries can define their own grid.

Example: The ICON Model (ICON partnership et al., 2024) uses an icosahedral grid composed of triangular grid cells instead of rectangular ones. For this model, `emiproc` provides an `ICON_grid` object.

Functions in `emiproc` can then perform various operations on these `Grid` objects.

Example: the `remap_inventory` function remaps the emissions to a different grid. It takes an `Inventory` and a `Grid` as input, and returns a new `Inventory` containing emissions remapped to the new grid.

Temporal and vertical profiles

To handle the temporal variation of the emissions, `emiproc` uses the `TemporalProfile` object, which is assigned to the `Inventory` and contains the temporal distribution of the emissions. Profiles can be either defined at specific datetimes or can be cyclic profiles at different temporal resolutions (e.g. daily, weekly, monthly).

The vertical distribution of the emissions is handled in a similar manner by the `VerticalProfile` object.

The temporal and vertical profile objects can be assigned specifically to certain types of emissions. For example, it is possible to assign them to a specific category / pollutant / country / gridcell.

Export functions

`emiproc` contains many functions to export inventories for various atmospheric models. The export produces all emission input files required by the model. These export functions are designed to make life of the modellers as simple as possible.

Some transport models might require additional data not included in the inventories. In this case, `emiproc` provides error messages, which guide the user into adding the missing data.

Emissions generation

Emission inventories do not necessarily contain all data relevant for a simulation. For example, human respiration contributes to emissions of CO2 but is rarely included in an inventory. However, an emission map for this sector can be estimated based on population density. For this purpose, `emiproc` provides a module that helps produce emission maps that rely on a spatial proxy (e.g. population density) and an emission factor.

Another example is CO2 emissions from vegetation. Several different models are available that estimate the exchange of CO2 with vegetation. `emiproc` implements VPRM (Vegetation Photosynthesis and Respiration Model) (Mahadevan et al., 2008), which (by default) makes use of vegetation and soil moisture indices, extracted from satellite observations.

This topic is already well-developed in the bottom-up component of `HERMESv3` (Guevara et al., 2020), as it includes emission modules for a wide range of sectors and pollutants.

Visualizing the data

At the end of the processing, a modeller may want to check the output files to see if the processing was successful. For some of the outputs, `emiproc` provides example plot scripts based on `matplotlib` (Hunter, 2007). Inventories on regular grids can be plotted using the `plot_inventory` function from the `emiproc.plots` module. Figure 1 was created using this function.

Availability

The package is available on GitHub and the documentation is available on readthedocs.

Tutorials are available to guide new users. A good first start is the Edgar processing tutorial which shows how `emiproc` can be used to load, process and export a freely available inventory.

Acknowledgements

We acknowledge all the previous and current contributers of emiproc: Michael Jähn, Gerrit Kuhlmann, Qing Mu, Jean-Matthieu Haussaire, David Ochsner, Katherine Osterried, Valentin Clément, Alessandro Bigi.

We would like to thank Hugo Denier van der Gon and Jeroen Kuenen from TNO for their support for the integration of the latest TNO inventories and Tobias Kugler and Corinne Hörger from the city of Zurich for providing the detailed city inventory.

We also acknowledge C2SM (Center for Climate Systems Modeling) for the development of the first version of `emiproc`.

Finally we would like to thank the developers of the Python packages used by `emiproc` and the whole Python community for providing such a great ecosystem.

References

Dönmez, E., K., & Brunner, D. (2024). Urban climate and CO2 simulations with the new atmospheric model ICON-ART accounting for spatially varying urban morphology and material properties. *EGU General Assembly 2024*. https://doi.org/10.5194/egusphere-egu24-3375

European Environment Agency. (2023). *EMEP/EEA air pollutant emission inventory guidebook 2023 – technical guidance to prepare national emission inventories*. Publications Office of the European Union. https://doi.org/10.2800/795737

Guevara, M., Tena, C., Porquet, M., Jorba, O., & Pérez García-Pando, C. (2019). HERMESv3, a stand-alone multi-scale atmospheric emission modelling framework – part 1: Global and regional module. *Geoscientific Model Development*, *12*(5), 1885–1907. https://doi.org/10.5194/gmd-12-1885-2019

Guevara, M., Tena, C., Porquet, M., Jorba, O., & Pérez García-Pando, C. (2020). HERMESv3, a stand-alone multi-scale atmospheric emission modelling framework – part 2: The bottom–up module. *Geoscientific Model Development*, *13*(3), 873–903. https://doi.org/10.5194/gmd-13-873-2020

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

ICON partnership, DWD, MPI-M, DKRZ, KIT, & C2SM. (2024). *ICON release 2024.01*. World Data Center for Climate (WDCC) at DKRZ. https://doi.org/10.35089/WDCC/IconRelease01

Jähn, M., Kuhlmann, G., Mu, Q., Haussaire, J.-M., Ochsner, D., Osterried, K., Clément, V., & Brunner, D. (2020). An online emission module for atmospheric chemistry transport models: Implementation in COSMO-GHG v5.6a and COSMO-ART v5.1-3.1. *Geoscientific Model Development*, *13*(5), 2379–2392. https://doi.org/10.5194/gmd-13-2379-2020

Jordahl, K., Bossche, J. V. den, Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A. G., Farmer, C., Hjelle, G. A., Snow, A. D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., maxalbert, Bilogur, A., … Leblanc, F. (2020). *Geopandas/geopandas: v0.8.1* (Version v0.8.1). Zenodo. https://doi.org/10.5281/zenodo.3946761

Mahadevan, P., Wofsy, S. C., Matross, D. M., Xiao, X., Dunn, A. L., Lin, J. C., Gerbig, C., Munger, J. W., Chow, V. Y., & Gottlieb, E. W. (2008). A satellite-based biosphere parameterization for net ecosystem CO2 exchange: Vegetation photosynthesis and respiration model (VPRM). *Global Biogeochemical Cycles*, *22*(2). https://doi.org/10.1029/2006GB002735

Monforti Ferrario, F., Crippa, M., Guizzardi, D., Muntean, M., Schaaf, E., Lo Vullo, E., Solazzo, E., Olivier, J., & Vignati, E. (2021). *EDGAR (Emissions Database for Global Atmospheric Research) v6.0 Greenhouse Gas Emissions*. European Commission, Joint Research Centre (JRC) [Dataset]. http://data.europa.eu/89h/97a67d67-c62e-4826-b873-9d972c4f670b

Ponomarev, S., N., & Brunner, D. (2024). Estimation of CO2 fluxes in the city of zurich using the mesoscale atmospheric transport and inversion model ICON-ART-CTDAS. *EGU General Assembly 2024*. https://doi.org/10.5194/egusphere-egu24-7420

Schröter, J., Rieger, D., Stassen, C., Vogel, H., Weimer, M., Werchner, S., Förstner, J., Prill, F., Reinert, D., Zängl, G., Giorgetta, M., Ruhnke, R., Vogel, B., & Braesicke, P. (2018). ICON-ART 2.1: A flexible tracer framework and its application for composition studies in numerical weather forecasting and climate simulations. *Geoscientific Model Development*, *11*(10), 4043–4068. https://doi.org/10.5194/gmd-11-4043-2018

Steiner, M., Peters, W., Luijkx, I., Henne, S., Chen, H., Hammer, S., & Brunner, D. (2024). European CH_4 inversions with ICON-ART coupled to the CarbonTracker data assimilation shell. *Atmospheric Chemistry and Physics*, *24*(4), 2759–2782. https://doi.org/10.5194/acp-24-2759-2024