The Journal of Open Source Software

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: 10.21105/joss.07509

Software

- Review
- Repository
- Archive

Editor: Mengqi Zhao

Reviewers:

- @einaraz
- @mikapfl

Submitted: 02 October 2024
Published: 14 January 2025

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

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.

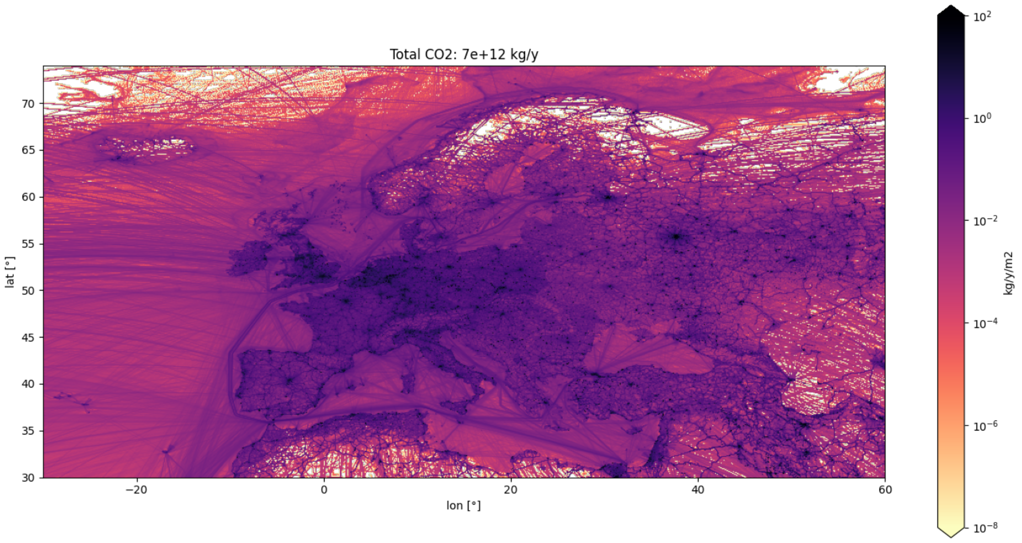

Figure 1: Anthropogenic CO2 emissions of the year 2022 from the EDGARv6 inventory .

Inventories are created in multiple different formats and resolutions, which makes it difficult to compare and use them in atmospheric transport models. 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 (points, lines, polygons) depending on the category of the source. Atmospheric models require emission inventories to be in a specific file format. For example ICON-ART (Schröter et al., 2018) requires emissions on a semi-structured triangular grid and temporal profiles to scale the emissions with hourly, daily and monthly variability.

Statement of need

`emiproc` is a Python package that provides tools for processing emission inventories, harmonizing datasets, and preparing such data as inputs for atmospheric transport models. Designed to be flexible and extendable, `emiproc` allows users to easily add custom functionality, read new inventories or export data to other formats.

When modellers design their simulations, they are often interested in modifying the inventories. For example, they could do the following: scale 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 et al. (2024) conducted urban climate simulation using emissions produced with `emiproc` for cities of Zurich and Basel.
- Ponomarev et al. (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 by 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

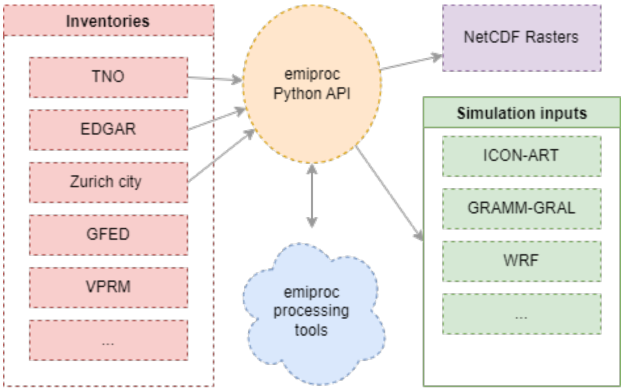

Figure 2: Design of emiproc.

The API of `emiproc` leverages the advantages of object-oriented programming. The main class is `Inventory`. It provides a common data structure to represent and harmonize all the different emission inventories. The `Grid` object represent the spatial distribution of the emission. It is flexible enough to implement all kinds of grids, from regular latitude-longitude grids to the icosahedral grid from the ICON Model (ICON partnership (DWD; MPI-M; DKRZ; KIT; C2SM), 2024). Finally `TemporalProfile` and `VerticalProfile` are used for the distribution of emissions in time and height.

Thanks to the harmonization of the data, functions for additional data processing can easily be applied to the different inventories once loaded into `emiproc`. These functions are listed in the API documentation. Examples of operations on inventory data are among others:

- remapping to different model grids
- aggregating emissions by sector or pollutant
- crop an inventory over a specific area

`emiproc` also allows for generation of additional emission sectors not always present in the inventories. For example, the vegetation emissions can be modelled through VPRM (Vegetation Photosynthesis and Respiration Model) (Mahadevan et al., 2008).

The export of inventories for various atmospheric models is done through custom functions that produce all emission input files required by the model.

For data vizualisation `emiproc` provides custom functions based on `matplotlib` (Hunter, 2007). Figure 1 was created using such a function.

`emiproc` is built on top of `geopandas` (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`.

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, K., Emmenegger, L., & 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

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

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, N., Steiner, M., Koene, E., Constantin, L., Rubli, P., Grange, S., Emmenegger, L., & 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