Statistics-based counterfactual climate data toolbox

D2.4

30th September 2025

Title	Statistics-based counterfactual climate data toolbox
Lead Beneficiary	Potsdam Institute for Climate Impact Research
Lead Author(s)	Matthias Mengel (Potsdam Institute for Climate Impact Research), Sven Willner (akator.de), Robert Gieseke (freelance)
Contributors	Simon Treu (Potsdam Institute for Climate Impact Research)
Deliverable number	2.4
Work Package	WP2: Attribution frameworks and physical drivers of compound extremes
Submission date	30/09/2025

Dissemination Level

PU	Public — fully open (automatically posted online)	X
SEN	Sensitive — limited under the conditions of the Grant Agreement	
CI	EU classified —Restreint-UE/EU-Restricted, Confidentiel-UE/EU-Confidential, Secret-UE/EU-Secret under Decision 2015/444	

Version History

Date	Version	Contributors	Comments
31/08/2025	0.1	Matthias Mengel	First version for internal review
21/09/2025	0.2	Pawel Terefenko	Internal review
29/09/2025	0.3	Matthias Mengel, Pawel Terefenko, Anaïs Couasnon	Final review and first finished version
30/09/2025	1.0	Anaïs Couasnon	Final version

Citation

Mengel, M., Willner, S., Gieseke, R., Treu, S., Terefenko, P., Couasnon, A. (2025): Statistics-based counterfactual climate data toolbox. Horizon Europe project COMPASS. Deliverable D2.4.

Funded by the European Union

The COMPASS project has received funding from the European Union's HORIZON Research and Innovation Actions Programme under Grant Agreement No. 101135481

Executive summary

Deliverable 2.4 "Statistics-based counterfactual climate data toolbox" is a software deliverable. It documents the release of the ATTRICI toolbox, a Python package for generating counterfactual climate data based on the statistical relationship between global mean temperature (GMT) and the local distribution of climate variables. The toolbox enables users to remove GMT-related trends from observational or reanalysis datasets, thereby producing counterfactual forcing data for attribution analyses with impact models. This functionality accommodates multiple climate variables and is designed to support attribution studies across a range of spatial and temporal resolutions.

The original ATTRICI method (Mengel et al., 2021) was developed to produce counterfactual climate datasets for attribution studies within the Intersectoral Impact Model Intercomparison (ISIMIP) project. It builds on a quantile-mapping framework where global mean temperature change, rather than time, is used as the predictor for local climate variable distributions. This approach removes the long-term warming signal while preserving observed variability. In its initial implementation, the seasonal cycle was represented by Fourier harmonics, and parameter estimation was carried out with Bayesian inference (PyMC3 Python package, Salvatier et al. 2016). However, the original code was limited in portability as it was tied to the ISIMIP global data resolution and closely linked to the Potsdam Institute high-performance computing environment.

The toolbox has been prepared for usability across computing environments from personal computers to high-performance clusters, with a focus on reproducibility, validation, and modularity. New developments include an alternative SciPy-based solver (Virtanen et al. 2020) for parameter estimation, an improved treatment of the annual cycle inspired by ISIMIP practices, and a bootstrap-based validation framework to assess the robustness of fitted trends. Full technical documentation is available online.

The code is openly developed and maintained in the ISIMIP GitHub repository at https://github.com/ISI-MIP/attrici. This link has been added to the HORIZON-COMPASS GitHub organization that contains all code outputs of the COMPASS project. The versioned release associated with D2.4 is archived on Zenodo to ensure long-term availability and citability. It is available at https://zenodo.org/records/16992690 and refers to release v2.0.0 available here: https://github.com/ISI-MIP/attrici/releases/tag/v2.0.0.

Table of Contents

1. Achieved milestones

M1 – Improvements in usability and reproducibility

The toolbox was refactored into a pip-installable Python package, with clearly defined dependencies and reproducible environments. Preprocessing and postprocessing workflows were streamlined, relying on the xarray Python package (Hoyer and Hamman, 2017) for efficient handling of NetCDF data. Reproducibility was further ensured by storing experiment metadata, including command-line arguments, code version, and package versions. Deprecated settings were removed, and an argument parser replaced the earlier configuration file.

M2 – Better model validation

A validation framework was introduced to safeguard against problematic input data and unreliable parameter fits. Automated tests aim to capture issues like unreasonable outputs. To assess the robustness of estimated trends, a moving-block bootstrap was implemented. This provides uncertainty estimates and highlights regions where model fits may be weak or unstable. The results are returned in a spatially explicit form, enabling users to identify problematic areas in their datasets.

M3 – Redesign and code modularization

The entire codebase was redesigned with a modular class-based architecture. The new structure separates concerns into independent, reusable modules with clearer interfaces, making the toolbox easier to maintain and extend. Precipitation, as the most complex variable, was used as the initial test case, ensuring that the framework is sufficiently general to accommodate all other variables used in ISIMIP.

M4 – Additional PyMC5 and Scipy solver

While the PyMC-based implementation for parameter estimation was retained, an alternative solver based on SciPy (Virtanen et al. 2020) was implemented. This provides users with a lightweight maximum-likelihood option alongside the Bayesian inference framework, thereby increasing flexibility and reducing dependency risks. The PyMC-based implementation was updated from PyMC3 (Salvatier et al. 2016) to PyMC5 (PyMC-Devs, 2025).

M5 – Online documentation and test suite added

Comprehensive documentation was prepared, including installation instructions, example workflows, and Jupyter notebooks. Automated testing can help with checking core functionality, and the toolbox now raises informative warnings and errors when inputs do not meet requirements (e.g., temporal alignment, unit consistency, or grid mismatches). The complete documentation is publicly available at https://isi-mip.github.io/attrici/attrici.html .

M6 – Alternative treatment of the annual cycle (ISIMIP-style)

Within this milestone, the toolbox adopted an alternative treatment of the annual cycle inspired by the ISIMIP bias adjustment framework. Instead of fitting Fourier modes to the seasonal cycle as in the original ATTRICI implementation, the approach models each calendar day separately. In practice, 365 models are fitted—one for each day of the year—using pooled data from a ±15-day window around the target day. This provides a more flexible representation of intra-annual variability and is more consistent with established ISIMIP practices.

2. Using, maintaining and updating the toolbox

The toolbox includes small example files for getting accustomed with it. The main usage commands are documented in the Getting Started section of the online documentation.

The toolbox can detrend the climate variables that are used within the ISIMIP project (Frieler et al. 2024) as climate input, see Table 1.

Table 1: Climate variables implemented for detrending in the attrici toolbox (copy from Table 1 of Mengel et al. 2021)

Variable	Short name	Unit	Statistical distribution	Link function
Daily mean near-surface air temperature	tas	K	Gaussian with mean value $\mu(T,t)$ and standard deviation $\sigma(t)$	$g(\mu)=\mu$
Daily near-surface temperature range	tasrange	K	Gamma with mean value $\mu(T,t)$ and shape $k(t)$	$g(\mu)=\ln(\mu)$
Daily near-surface temperature skewness	tasskew	1	Gaussian with mean value $\mu(T,t)$ and standard deviation $\sigma(t)$	$g(\mu)=\mu$
Precipitation	pr	$\mathrm{kg\,m^{-2}\,s^{-1}}$	For wet or dry day: Bernoulli with dry-day probability $p(T,t)$ For intensity of precipitation on wet days: gamma with mean value $\mu(T,t)$ and shape $k(t)$	$g(p)=\ln(p/(1-p))$ $g(\mu)=\ln(\mu)$
Surface downwelling shortwave radiation	rsds	$\mathrm{W\,m^{-2}}$	Gaussian with mean value $\mu(T,t)$ and standard deviation $\sigma(t)$	$g(\mu)=\mu$
Surface downwelling longwave radiation	rlds	$\mathrm{W\,m^{-2}}$	Gaussian with mean value $\mu(T,t)$ and standard deviation $\sigma(t)$	$g(\mu)=\mu$
Surface air pressure	ps	Pa	Gaussian with mean value $\mu(T,t)$ and standard deviation $\sigma(t)$	$g(\mu)=\mu$
Near-surface wind speed	sfcwind	$\mathrm{m\,s^{-1}}$	Weibull with shape $\alpha(t)$ and scale $\beta(T,t)$	$g(\beta)=\ln(\beta)$
Near-surface relative humidity	hurs	%	Beta with mean value $\mu(T,t)$ and dispersion $\phi(t)$	$g(\mu)=\ln(\mu/(1-\mu))$
Near-surface specific humidity	huss	$\mathrm{kg\,kg^{-1}}$	Derived from hurs, ps and tas	
Daily minimum near-surface air temperature	tasmin	K	Derived from tas, tasrange and tasskew	
Daily maximum near-surface air temperature	tasmax	K	Derived from tas, tasrange and tasskew	

The input file should be gridded netcdf files. The basic structure of a compliant input file is shown below.

```
dimensions:
    time = UNLIMITED ; // (44925 currently)
    lon  = 2 ;
    lat  = 2 ;
```

```
variables:
    double time(time) ;
        time:standard_name = "time" ;
        time:long_name     = "time" ;
        time:units         = "days since 1900-1-1 00:00:00" ;
        time:calendar      = "proleptic_gregorian" ;
        time:axis          = "T" ;

    double lon(lon) ;
        lon:standard_name  = "longitude" ;
        lon:long_name      = "Longitude" ;
        lon:units          = "degrees_east" ;
        lon:axis           = "X" ;

    double lat(lat) ;
        lat:standard_name  = "latitude" ;
        lat:long_name      = "Latitude" ;
        lat:units          = "degrees_north" ;
        lat:axis           = "Y" ;

    float hurs(time, lat, lon) ;
        hurs:standard_name = "relative_humidity" ;
        hurs:long_name     = "Near-Surface Relative Humidity" ;
        hurs:units         = "%" ;
        hurs:_FillValue    = 1.e+20f ;
        hurs:missing_value = 1.e+20f ;

    float pr(time, lat, lon) ;
        pr:standard_name   = "precipitation_flux" ;
        pr:long_name       = "Precipitation" ;
        pr:units           = "kg m-2 s-1" ;
        pr:_FillValue      = 1.e+20f ;
```

The structure shown is for near-surface humidity (hurs) and precipitation (pr) as examples. The netcdf files can hold from 1 to all 9 variables (huss, tasmin and tasmax are derived variables, see Table 1). Variables are processed individually. The variables huss, tasmin, tasmax need non-standard processing.

In addition, the toolbox needs a file with a global mean temperature timeseries, which covers the period of the input file. For more detailed specifications, it is recommended to have a look to the included example files for global mean temperature and gridded input.

We also provide a notebook that explains the relations between equations in the original paper (Mengel et al. 2021) and the code.

We encourage users to report bugs in the code by opening an issue on Github (https://github.com/ISI-MIP/attrici/issues). We also welcome information on any problems or errors in the toolbox or documentation. Such issues, and other inquiries, can be reported directly to the main author (matthias.mengel@pik-potsdam.de).

3. References

Frieler, K., Volkholz, J., Lange, S., Schewe, J., Mengel, M., del Rocío Rivas López, M., ... & Bechtold, M. (2024). Scenario setup and forcing data for impact model evaluation and impact attribution within the third round of the Inter-Sectoral Impact Model Intercomparison Project (ISIMIP3a). *Geoscientific Model Development*, *17*(1), 1-51. https://doi.org/10.5194/gmd-17-1-2024

Hoyer, S. & Hamman, J., (2017). xarray: N-D labeled Arrays and Datasets in Python. Journal of Open Research Software. 5(1), p.10. DOI: https://doi.org/10.5334/jors.148

Mengel, M., Treu, S., Lange, S., & Frieler, K. (2021). ATTRICI v1.0: Counterfactual climate for impact attribution. Geoscientific Model Development, 14(8), 5269–5284. https://doi.org/10.5194/gmd-14-5269-2021

PyMC-Devs. (2025). PyMC (v5.25.1). Zenodo. https://doi.org/10.5281/zenodo.16411014

Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55. https://doi.org/10.7717/peerj-cs.55

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., & SciPy 1.0 Contributors. (2020). *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*. *Nature Methods*, 17(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2