





TopoPyScale: A Python Package for Hillslope Climate Downscaling

Simon Filhol ^{1*}, Joel Fiddes ^{2*}, and Kristoffer Aalstad ¹

¹ Department of Geosciences, University of Oslo, Norway ² WSL Institute for Snow and Avalanche Research SLF, Switzerland  Corresponding author * These authors contributed equally.

DOI: [10.21105/joss.05059](https://doi.org/10.21105/joss.05059)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Hugo Ledoux](#)  

Reviewers:

- [@dvalters](#)
- [@arbennett](#)

Submitted: 14 December 2022

Published: 23 June 2023

License

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

Summary

Global climate reanalyses and projections are available worldwide for the past and coming century. However, model grids remain too coarse to be directly relevant at the hillslope-scale (Fan et al., 2019), requiring adapted downscaling tools to account for the effects of local topography. Mountain regions are experiencing accelerated warming with the cryosphere rapidly responding to climate change. To understand and study geomorphological, hydrological, and glaciological changes, we need tools to downscale meteorological timeseries for the basic atmospheric variables used to solve surface energy and mass balance in process-based models. Advanced dynamical downscaling methods exist, though they come with a high computational cost and complex technical setup (Kruyt et al., 2022). TopoPyScale uses a pragmatic approach to downscaling by minimizing complexity, reducing computational cost, simplifying interoperability with land surface models, while retaining physical coherence and allowing the primary drivers of land surface-atmosphere interaction to be considered. The toolbox is designed to be flexible in its usage and development.

Statement of need

TopoPyScale is a community supported open-source Python package for performing climate downscaling following the initial work of Fiddes & Gruber (2014) and Fiddes & Gruber (2012). It is designed as a toolbox combining computationally efficient methods to downscale climate reanalysis and projections (Hersbach et al., 2020). At the moment, atmospheric climate models are typically run at a coarse spatial resolution (25 km for ERA5), missing the heterogeneity imposed by the topography of mountain ranges on atmospheric variables. Figure 1 shows an example of a downscaled temperature field for January 1, 2020 at 12:00 computed with TopoPyScale using ERA5 data as input. TopoPyScale allows one to reconstruct the variability of temperatures observed in between valleys and mountain tops within one single ERA5 grid cell. This method is now used in a number of studies to investigate geophysical processes such as geomorphological dynamics of permafrost (Renette et al., 2022), the hydrology of mountain catchments (Fiddes et al., 2019), mountain glaciers (Kronenberg et al., 2022) and downscaling hillslope-scale climate projections (Fiddes et al., 2022). The ease of use as well as the low computational cost help scientists to quickly obtain hillslope-scale atmospheric forcing data that is representative for their study domain. This new implementation of TopoSCALE into TopoPyScale brings a new software architecture facilitating development and usage. It is fully developed within the Python ecosystem of scientific computing libraries, thereby integrating every processing steps into a single workflow.

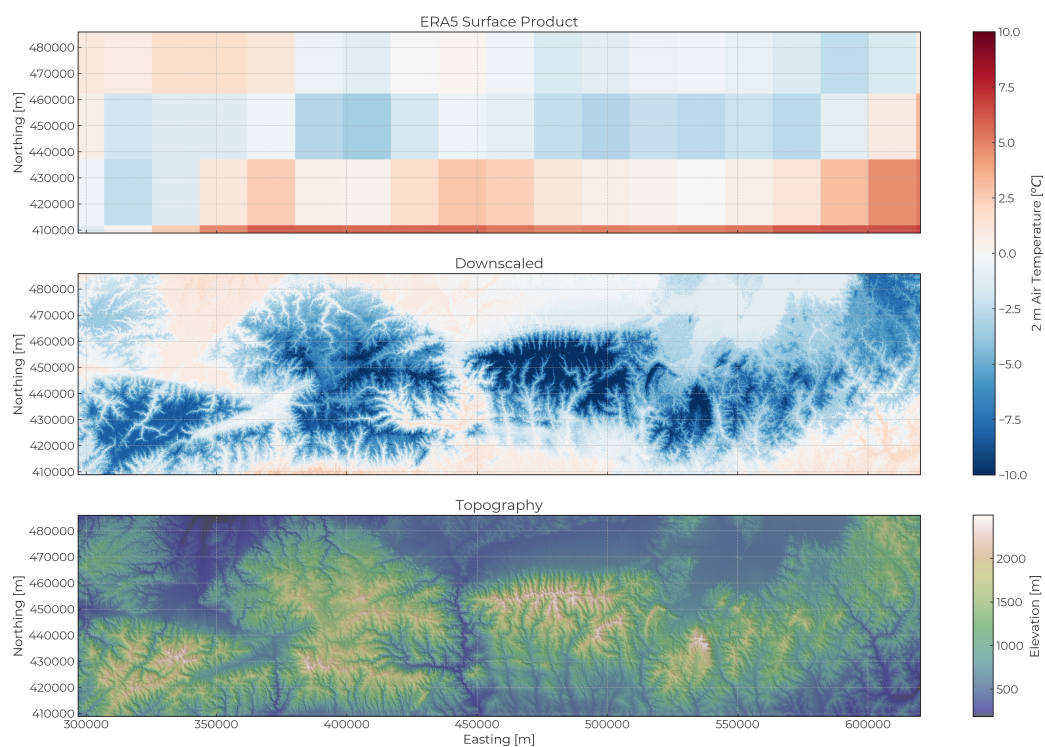


Figure 1: Comparison of the 2 m air temperature on January 1, 2020 at 12:00 UTC over the Southern Carpathians, Romania, between the ERA5 surface product (top) and the downscaled result using TopoPyScale (middle). The DEM (bottom) was segmented using 2000 clusters.

Toolbox methods and structure

TopoPyScale is built on keystone pythonic libraries like [pandas](#) ([The pandas development team, 2020](#)), [xarray](#) ([Hoyer & Hamman, 2017](#)) with parallelization thanks to [dask](#) ([Dask Development Team, 2016](#)) and Python multiprocessing, [topocalc](#) and [pvlib](#) ([Holmgren et al., 2018](#)) for computation of metrics related to topography and solar position, [scikit-learn](#) ([Pedregosa et al., 2011](#)) for the clustering algorithm, [rasterio](#) ([Gillies & others, 2013](#)) and [pyproj](#) for handling geospatial data.

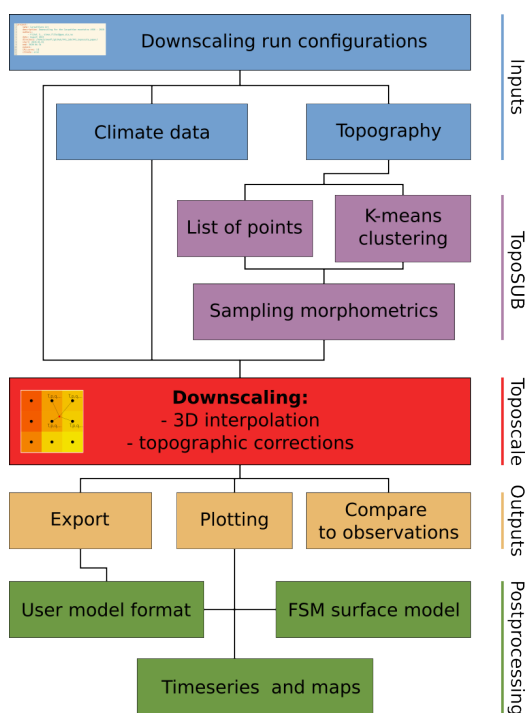


Figure 2: Workflow of TopoPyScale processing pipeline.

TopoPyScale consists of a set of tools to be run in a processing pipeline following the diagram of Figure 2. First, it takes a Digital Elevation Model (DEM), and climate data as inputs. The climate data must include air temperature, pressure, specific humidity, and meridional and zonal wind components from atmospheric pressure levels ranging from mean sea level to above the highest elevation of the DEM, as well as incoming shortwave radiation, incoming longwave radiation, and precipitation from the surface level. The next step is to compute terrain morphometrics such as slope, aspect, and sky view factor. Then, downscaling can be run in two modes, 1) *point*, or 2) *TopoSUB* downscaling. *Point* downscaling is used for a list of specific points (e.g. weather station locations), for which meteorological variables are downscaled only at the coordinates of given points. The *TopoSUB* approach (Fiddes & Gruber, 2012) is instead executed for semi-distributed spatial downscaling (Figure 2). The DEM is clustered based on morphometrics. *Toposcale*, the downscaling routine, is then run only for the centroid of each cluster or the list of given points. This method efficiently abstracts the DEM into a limited number of representative points as opposed to all the DEM grid cells, with potentially several orders of magnitude of computational effort saved. *Toposcale* uses 3D spatial interpolation and geometrical corrections to downscale the meteorological variables (see Table 1). The results can then be exported in a number of formats that can be used as forcing for specialized land surface models for snow, permafrost or hydrology via readily extendable plugins. Currently, this includes the models Cryogrid (Westermann et al., 2023), Crocus (Vionnet et al., 2012), SNOWPACK (Bartelt & Lehning, 2002), Snowmodel (Liston & Elder, 2006), Geotop (Endrizzi et al., 2014), and the data assimilation toolkit MuSA (Alonso-González et al., 2022). Finally, 1D results can be mapped back to the full DEM in order to generate spatially complete results. TopoPyScale also includes a toolbox to perform snow simulations with FSM (Essery, 2015), and data assimilation of fractional snow-covered area (Fiddes et al., 2019).

Table 1: Default output variables of TopoPyScale (based on ERA5).

Name	Variable	Unit	Downscaling type
2 m Air Temperature	t	K	Horizontal and vertical interpolation
2 m Air atmospheric pressure	p	bar	Horizontal and vertical interpolation
10 m Wind speed	ws	m.s-1	Horizontal and vertical interpolation
10 m Wind direction	wd	degree	Horizontal and vertical interpolation
10 m Wind U-component	u	m.s-1	Horizontal and vertical interpolation
10 m Wind V-component	v	m.s-1	Horizontal and vertical interpolation
2 m Specific air humidity	q	kg.kg-1	Horizontal and vertical interpolation
Precipitation rate	tp	mm.hr-1	Horizontal and vertical interpolation (optional: lapse-rate)
Longwave radiation	LW	W.m-2	Geometrical and atmospheric correction
Shortwave radiation	SW	W.m-2	Geometrical and atmospheric correction
Direct shortwave radiation	SW_direct	W.m-2	Geometrical and atmospheric correction

Working examples

The repository [TopoPyScale_examples](#) provides applications of TopoPyScale to three independent regions:

- The site of [Finse](#) in Southern Norway. Located at 1200 m above sea level, Finse is equipped with a wide range of instruments for atmospheric and hydrological studies since 2016. In this example TopoPyScale is run for the weather station location.
- The mountain range of Retezat in the Romanian Carpathian mountains. This is an example for applying the *TopoSUB* subroutine for spatialized downscaling.
- The area of Davos, Switzerland, specifically selected for studies on snow distribution and avalanches. This example shows how TopoPyScale can be combined within the same workflow as the snow model FSM.

Acknowledgements

SF was funded from ClimaLand, an EEA/EU collaboration grant between Norway and Romania 2014-2021, project code RO-NO-2019-0415,1290 contract no. 30/2020. JF was funded by the Swiss National Science Foundation (grant no. 179130). KA was funded by ESA Permafrost_CCI (<https://climate.esa.int/en/projects/permafrost/>) and the Spot-On project (Research Council of Norway #301552), and acknowledges support from the LATICE strategic research area at the University of Oslo.

References

- Alonso-González, E., Aalstad, K., Baba, M. W., Revuelto, J., López-Moreno, J. I., Fiddes, J., Essery, R., & Gascoïn, S. (2022). The Multiple Snow Data Assimilation System (MuSA v1.0). *Geoscientific Model Development*, 15(24), 9127–9155. <https://doi.org/10.5194/gmd-15-9127-2022>
- Bartelt, P., & Lehning, M. (2002). A physical SNOWPACK model for the Swiss avalanche warning: Part i: Numerical model. *Cold Regions Science and Technology*, 35(3), 123–145.

- [https://doi.org/10.1016/S0165-232X\(02\)00074-5](https://doi.org/10.1016/S0165-232X(02)00074-5)
- Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. <https://dask.org>
- Endrizzi, S., Gruber, S., Dall'Amico, M., & Rigon, R. (2014). GEOTop 2.0: Simulating the combined energy and water balance at and below the land surface accounting for soil freezing, snow cover and terrain effects. *Geoscientific Model Development*, 7(6), 2831–2857. <https://doi.org/10.5194/gmd-7-2831-2014>
- Essery, R. (2015). A factorial snowpack model (FSM 1.0). *Geoscientific Model Development*, 8(12), 3867–3876. <https://doi.org/10.5194/gmd-8-3867-2015>
- Fan, Y., Clark, M., Lawrence, D. M., Swenson, S., Band, L. E., Brantley, S. L., Brooks, P. D., Dietrich, W. E., Flores, A., Grant, G., Kirchner, J. W., Mackay, D. S., McDonnell, J. J., Milly, P. C. D., Sullivan, P. L., Tague, C., Ajami, H., Chaney, N., Hartmann, A., ... Yamazaki, D. (2019). Hillslope hydrology in global change research and earth system modeling. *Water Resources Research*, 55(2), 1737–1772. <https://doi.org/10.1029/2018WR023903>
- Fiddes, J., Aalstad, K., & Lehning, M. (2022). TopoCLIM: Rapid topography-based downscaling of regional climate model output in complex terrain v1.1. *Geoscientific Model Development*, 15(4), 1753–1768. <https://doi.org/10.5194/gmd-15-1753-2022>
- Fiddes, J., Aalstad, K., & Westermann, S. (2019). Hyper-resolution ensemble-based snow reanalysis in mountain regions using clustering. *Hydrology and Earth System Sciences*, 23(11), 4717–4736. <https://doi.org/10.5194/hess-23-4717-2019>
- Fiddes, J., & Gruber, S. (2012). TopoSUB: A tool for efficient large area numerical modelling in complex topography at sub-grid scales. *Geoscientific Model Development*, 5(5), 1245–1257. <https://doi.org/10.5194/gmd-5-1245-2012>
- Fiddes, J., & Gruber, S. (2014). TopoSCALE v.1.0: Downscaling gridded climate data in complex terrain. *Geoscientific Model Development*, 7(1), 387–405. <https://doi.org/10.5194/gmd-7-387-2014>
- Gillies, S., & others. (2013). *Rasterio: Geospatial raster i/o for Python programmers*. Mapbox. <https://github.com/rasterio/rasterio>
- Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., ... Thépaut, J.-N. (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730), 1999–2049. <https://doi.org/10.1002/qj.3803>
- Holmgren, W. F., Hansen, C. W., & Mikofski, M. A. (2018). Pvlib python: A python package for modeling solar energy systems. *Journal of Open Source Software*, 3(29), 884. <https://doi.org/10.21105/joss.00884>
- Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, 5(1). <https://doi.org/10.5334/jors.148>
- Kronenberg, M., Pelt, W. van, Machguth, H., Fiddes, J., Hoelzle, M., & Pertziger, F. (2022). Long-term firn and mass balance modelling for Abramov glacier, Pamir Alay. *The Cryosphere Discussions*, 2022, 1–33. <https://doi.org/10.5194/tc-2021-380>
- Kruyt, B., Mott, R., Fiddes, J., Gerber, F., Sharma, V., & Reynolds, D. (2022). A downscaling intercomparison study: The representation of slope- and ridge-scale processes in models of different complexity. *Frontiers in Earth Science*, 10. <https://doi.org/10.3389/feart.2022.789332>
- Liston, G. E., & Elder, K. (2006). A distributed snow-evolution modeling system (SnowModel). *Journal of Hydrometeorology*, 7(6), 1259–1276. <https://doi.org/10.1175/JHM548.1>

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Renette, C., Aalstad, K., Aga, J., Zweigel, R. B., Etzelmüller, B., Lilleøren, K. S., Isaksen, K., & Westermann, S. (2022). Simulating the effect of subsurface drainage on the thermal regime and ground ice in blocky terrain, Norway. *Earth Surface Dynamics Discussions*, 2022, 1–28. <https://doi.org/10.5194/esurf-2022-39>
- The pandas development team. (2020). *Pandas-dev/pandas: pandas* (latest). Zenodo. <https://doi.org/10.5281/zenodo.3509134>
- Vionnet, V., Brun, E., Morin, S., Boone, A., Faroux, S., Le Moigne, P., Martin, E., & Willemet, J.-M. (2012). The detailed snowpack scheme Crocus and its implementation in SURFEX v7.2. *Geoscientific Model Development*, 5(3), 773–791. <https://doi.org/10.5194/gmd-5-773-2012>
- Westermann, S., Ingeman-Nielsen, T., Scheer, J., Aalstad, K., Aga, J., Chaudhary, N., Etzelmüller, B., Filhol, S., Käab, A., Renette, C., Schmidt, L. S., Schuler, T. V., Zweigel, R. B., Martin, L., Morard, S., Ben-Asher, M., Angelopoulos, M., Boike, J., Groenke, B., ... Langer, M. (2023). The CryoGrid community model (version 1.0) – a multi-physics toolbox for climate-driven simulations in the terrestrial cryosphere. *Geoscientific Model Development*, 16(9), 2607–2647. <https://doi.org/10.5194/gmd-16-2607-2023>