The Journal of Open Source Software

# ncompare: A Python package for comparing netCDF structures

**Daniel E. Kaufman** [1,2] **and Walter E. Baskin** [1,3]

**1** NASA Langley Research Center, Atmospheric Science Data Center, Hampton, VA, USA **2** Booz Allen Hamilton, Inc., McLean, VA, USA **3** Adnet Systems, Inc., Bethesda, MD, USA

**DOI:** 10.xxxxxx/draft

**Software**

- Review
- Repository
- Archive

**Editor:** Arfon Smith

**Reviewers:**

- @cmtso
- @cmarmo

**Submitted:** 12 March 2024
**Published:** unpublished



## Summary

ncompare compares the structure of two Network Common Data Form (netCDF) files at the command line, thus providing rapid and human-readable evaluation of netCDF pairs. The essential inputs to ncompare are the filepaths of two netCDF datasets, and the output is a report that automatically aligns and highlights differences between the matching and non-matching groups, variables, and associated metadata (e.g., dimension lengths, attributes, chunking). The user is provided the option to colorize the terminal output for ease of viewing, to save comparison reports in text, comma-separated value (CSV), and/or Microsoft Excel formats, and to compare values for a particular variable of interest. ncompare is written using the Python programming language (Python Core Team, 2015; Van Rossum & Drake, 2009). To use ncompare, it can be given input from a command line interface (CLI), or its functions can be called directly from within a running Python kernel. The order of operations proceeds through these steps: comparing root-level dimensions, groups, structure and values of an optional user-specified group/variable, and finally all the variables in each group.

## Statement of need

The Network Common Data Form (netCDF) file format enables the storage and use of multidimensional data (Brown et al., 1993; Rew & Davis, 1990). It is widely applied to research problems throughout the Earth sciences — e.g., to store and compare output from climate models, to store and prepare oceanographic or atmospheric reanalyses, and to store and analyze observational data. When creating or modifying netCDF files, there is often a need to evaluate the differences between an original unmodified file and a new modified file, especially for regression testing. Despite the availability of tools (such as `ncmpidiff` or `nccmp`) that compare the *values* of variables, there was not a readily available, Python-based tool for rapid visual comparisons of group and variable *structures*, *attributes*, and *chunking*. ncompare was developed to avoid the inefficient process of manually opening two netCDF files and inspecting their contents to determine whether there are differences in the structure and shapes of groups and variables.

ncompare has been used by the National Aeronautics and Space Administration (NASA) Atmospheric Science Data Center (ASDC) to examine preliminary science data products in preparation for ingesting, archiving, and distributing satellite-based instrument retrievals. For example, to prepare for new data streams from the recently launched Tropospheric Emissions Monitoring of Pollution (TEMPO) instrument (NASA/LARC/SD/ASDC, 2019; Zoogman et al., 2017) — which collects measurements of major air pollutants, including ozone, nitrogen dioxide, and formaldehyde — the ASDC used ncompare to identify data structure changes, or the lack thereof, in a variety of settings. For instance, as data and metadata requirements were being established and refined, ncompare was used to assess changes from one version

of data files to another. The ncompare package was used to confirm whether NASA's data transformation services, including those that perform data subsetting and concatenation, modified dataset variables and attributes appropriately. By allowing data scientists at ASDC to quickly identify any and all changes in netCDF structures, ncompare sped up and enhanced the process of validating data integrity, critical to ensuring the discoverability and usability of TEMPO air quality observations for air quality monitoring, research, and forecasting.

The ncompare package fills a gap in the currently available range of netCDF evaluation tools. The cdo (climate data operators) library (Schulzweida, 2022) does not support NetCDF4 groups. The ncdiff function in the nco (netCDF Operators) library (Zender, 2008) computes value differences, but — as far as the authors are aware — does not have a simple function to show structural differences between netCDF version 4 (netCDF4) datasets. h5diff, provided in the HDF5 (Hierarchical Data Format) software (The HDF Group, 1997-2023), can be used to compare netCDF4 files; however, there are notable differences. In contrast to h5diff, ncompare is written and runnable in Python; ncompare provides an *aligned* and *colorized* difference report for more efficient and intuitive assessments of groups, variable names, types, shapes, and attributes; and can generate report files formatted for other applications. However, note that h5diff provides comparison of "hidden" hdf5 properties, such as _Netcdf4Dimid or _Netcdf4Coordinates, which are not currently assessed by ncompare.

## Development Notes

ncompare is developed as an open-source package on GitHub; contributions and feature suggestions are welcome. Continuous Integration using GitHub Actions ensures code linting (via ruff), formatting (via black), version updating, and testing (via pytest) is routinely performed. ncompare is available on PyPI (The Python Package Index) and can be installed using pip. It is released under the NASA Open Source Agreement, and its source code is available at https://github.com/nasa/ncompare.

## Acknowledgements

This project was supported as part of the STARSS and RSES contracts to the NASA Langley Research Center and Atmospheric Science Data Center. ncompare makes use of numpy (Harris et al., 2020), netCDF4, xarray (Hoyer & Hamman, 2017), colorama, and openpyxl.

## References

Brown, S. A., Folk, M., Goucher, G., Rew, R., & Dubois, P. F. (1993). Software for Portable Scientific Data Management. *Computer in Physics, American Institute of Physics*, *7*(3), 304–308. https://doi.org/10.1063/1.4823180

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

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

NASA/LARC/SD/ASDC. (2019). *TEMPO geolocated earth radiances (BETA)*. NASA Langley Atmospheric Science Data Center DAAC. https://doi.org/10.5067/IS-40e/TEMPO/RAD_L1.002

Python Core Team. (2015). *Python: A dynamic, open source programming language. Python Software Foundation*. https://www.python.org/

Rew, R., & Davis, G. (1990). NetCDF: An interface for scientific data access. *IEEE Computer Graphics and Applications*, *10*(4), 76–82. https://doi.org/10.1109/38.56302

Schulzweida, U. (2022). *CDO user guide* (Version 2.1.0). Zenodo. https://doi.org/10.5281/zenodo.7112925

The HDF Group. (1997-20231997-2023). *Hierarchical Data Format, version 5*.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace. ISBN: 1441412697

Zender, C. S. (2008). Analysis of self-describing gridded geoscience data with netCDF operators (NCO). *Environmental Modelling & Software*, *23*(10), 1338–1342. https://doi.org/https://doi.org/10.1016/j.envsoft.2008.03.004

Zoogman, P., Liu, X., Suleiman, R. M., Pennington, W. F., Flittner, D. E., Al-Saadi, J. A., Hilton, B. B., Nicks, D. K., Newchurch, M. J., Carr, J. L., Janz, S. J., Andraschko, M. R., Arola, A., Baker, B. D., Canova, B. P., Chan Miller, C., Cohen, R. C., Davis, J. E., Dussault, M. E., … Chance, K. (2017). Tropospheric emissions: Monitoring of pollution (TEMPO). *Journal of Quantitative Spectroscopy and Radiative Transfer*, *186*, 17–39. https://doi.org/https://doi.org/10.1016/j.jqsrt.2016.05.008