Open-source tools to support standardization and cleaning of ground temperature data

Nick Brown[1,2], Alex Gao[2], and Stephan Gruber[2]

1. NSERC PermafrostNet
2. Carleton University, Ottawa, Canada

Lack of standardization hampers efficient data handling

Permafrost science and engineering rely on ground temperature data obtained from a range of different sources: sensors, permafrost databases or model output. Unfortunately, cleaning, organizing and sharing data continues to be an obstacle [1]. Several challenges limit how efficiently this data can be used in ressearch or in practice (Figure 1).

More than half of cryospheric datasets use 'in-house' text formats to store and distribute data and most researchers rely on ad-hoc scripts to manage data [3]. Consequently, new data sources commonly take a day or more to integrate into workflows. Permafrost ground temperature databases differ in the structure and nomenclature of available data. Similarly, dataloggers can be configured to output text files with differences in date format, separator, and metadata structure, confounding attempts to develop reusable scripts that easily open datalogger files for further processing.

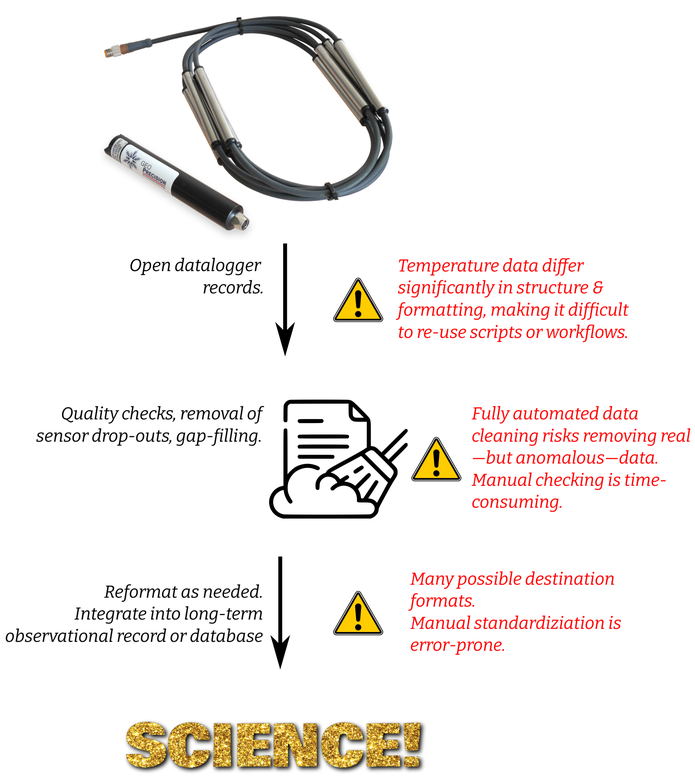

Figure 1: Some of the challenges working with ground temperature data illustrated as part of a typical workflow starting with sensor-based observations.

`tsp`: an engine for ground temperature data

`tsp` ("teaspoon") is a python library designed to simplify ground temperature data handling (Figure 1) in three ways [2]:

1. Increase the ease with which ground temperature time series data from any source can be read into python.
2. Provide reuseable functions to perform the most common visualization and analysis conducted on permafrost data.
3. Improve permafrost data interoperability by providing methods to output data in common text file formats used by the permafrost community and as more standardized, self-documenting formats such as netCDF.

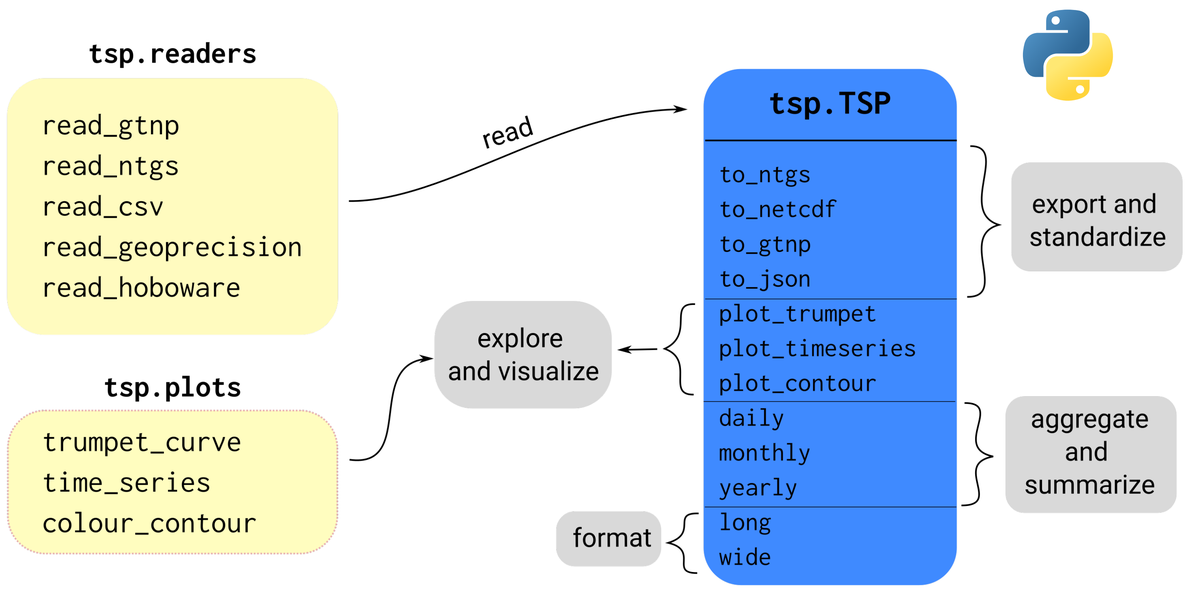

Figure 2: An overview of the tsp package showing the main functions (yellow) and classes (blue) that are included.

Building reuseable workflows with `tsp`

Researchers frequently re-write similar code for visualizing, analysing, or cleaning ground temperature data in successive projects as input data changes slightly. With **`tsp`**, the focus is on bringing data into a unified structure that can be treated the same way regardless of the format of the input files. This also means that techniques can be more easily shared because they rely less on a particular source data format. There are a number of common workflows that can benefit from the capabilities offered by **`tsp`** (Figure 3).

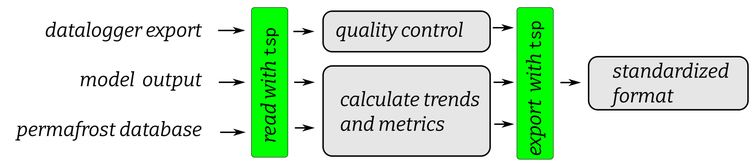

Figure 3: Example of how the **`tsp`** library (green boxes) can be integrated into various generic workflows, allowing them to ingest and export ground temperature data in a variety of formats.

`tempcf`: interactive quality control

As a demonstration of how `tsp` can be used to build reuseable workflows, we present `tempcf`. This software improves the useability of permafrost data by allowing data creators to visually flag, assess, and take action on suspicious measurements. It includes a library of functions for identifying suspicious data, and a visual interface to assist in the cleaning of the data (Figure 2).

By using the tsp library, tempcf can be used to open ground temperature files from a variety of sources including model outputs, datalogger exports, and files from territorial, national, and global permafrost databases.

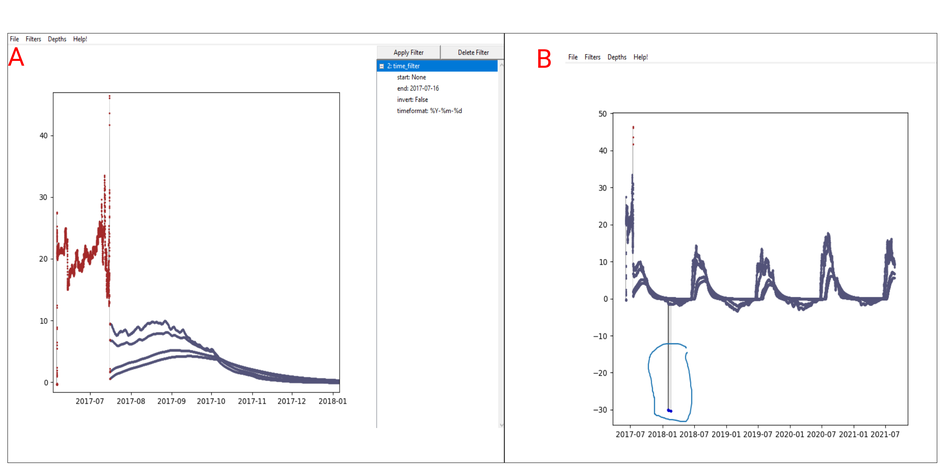

Figure 3: (a) In **`tempcf`**, data can be flagged using one of the available filtering algorithms. Here, data collected before an instrument was deployed is . These data can subsequently be selected and deleted. (b) Data can also be selected manually using a lasso tool. This is valuable for quickly removing sensor drop-outs or erroneous data that are easy to distinguish visually but difficult to distinguish in an automatic way.

Next steps

Development on both **`tsp`** and **`tempcf`** is ongoing. More source and destination file formats will be added to `tsp`, and contributions are encouraged to broaden the utility of this library. As a demonstrator example, tempcf currently has limited filtering algorithms to detect spurious data; further development will involve implementing more robust algorithms as well as adding tools for gap-filling as a way to produce robust datasets for simulation, or analysis.

Discover, comment, collaborate

To contribute to the projects, or to see a tutorial video of these tools being used, scan the QR codes below with your smartphone. Both projects were designed with participation in mind to make the tools useful as a shared community resource.

`gitlab.com/permafrostnet/teaspoon`

`gitlab.com/permafrostnet/tempcf`

`youtu.be/YF-L43Qi5Hc`

References and acknowledgements

[1] Brown, N., Gruber, S., Pulsifer, P., Stewart-Jones, E. (2020) Permafrost Data Workshop Final Report. NSERC PermafrostNet. May 27-29, 2020: Ottawa, Canada. http://doi.org/10.22215/pn/10120001

[2] Brown, N., (2022). tsp ("Teaspoon"): A library for ground temperature data. Journal of Open Source Software, 7(77), 4704, https://doi.org/10.21105/joss.04704

[3] Bavay, M., Fiddes, J. and Godøy, Ø., 2020. Automatic Data Standardization for the Global Cryosphere Watch Data Portal. Data Science Journal, 19(1), p.6.DOI:http://doi.org/10.5334/dsj-2020-006

This project was enabled in part by support provided by Westgrid (westgrid.ca) and the Digital Research Alliance of Canada (alliancecan.ca).

Digital Research Alliance of Canada | Alliance de recherche numérique du Canada

PermafrostNet NSERC | CRSNG

Carleton University