THE METEOIO PRE-PROCESSING LIBRARY FOR OPERATIONAL APPLICATIONS

Mathias Bavay[1,*], Joel Fiddes[2], Charles Fierz[1], Michael Lehning[1], Fabiano Monti[3], Thomas Egger[4]

[1] *Institute for Snow and Avalanche Research SLF, Davos, Switzerland*
[2] *World Meteorological Organization, Geneva, Switzerland*
[3] *Alpsolut, Livigno, Italy*
[4] *Egger Consulting, Vienna, Austria*

ABSTRACT: Originally developed to fulfill the needs of applications consuming meteorological data, the Open Source MeteoIO library offers many valuable features for operational systems. It has been from the very start developed with a focus on robustness and efficiency for applications that must run unattended. Its use has been expanded to cover a wide variety of operational tasks from monitoring to data standardization to data hub and from time series at single locations to time series of gridded data. It is now used by several operational services as well as several companies.

Keywords: meteorological data, data processing, quality control

1. INTRODUCTION

1.1. Context

The key issues for the users of meteorological data are 1) the vast diversity of data formats, 2) the necessity to correct the data for all kind of known measurement errors, 3) the variety of sampling rates and the mismatch between the measured sampling rate and the application's sampling rate of choice. This leads to the very time consuming task of preparing the data for the foreseen usage[1]. The MeteoIO library (Bavay and Egger. (2014)) was iniated in 2008 to address these issues both in a research context (where flexibility and traceability are keys) and in an operational context (where robustness and performance are keys). As an Open Source C++ library, it can easily be integrated into other applications, such as numerical models, dashboards or simple data converters.

1.2. MeteoIO's principle of operation

MeteoIO goes through several steps for preparing the data (Fig. 1), aiming to offer within a single package all the tools that are required to bring raw data to a data consumer: first, the data is read by one of the more than twenty available plugins, matching that many different formats or protocols (such as CSV files, NetCDF files, databases or web services). Then some basic data editing can be performed (such as removing sensors that are known to be faulty, merging stations that are next to each other and would benefit from being combined, renaming sensors). The data can then be filtered, by applying a stack of user-selected generic filters. These filters can either remove invalid data (such as despiking, low or high pass filters, removing precipitation data from unheated rain gauges in winter) or correct the data (such as precipitation undercatch correction, debiasing, correcting for unventilated temperature sensors). Once this is done, the data is resampled to the requested time steps by various temporal interpolations methods (linear interpolation, precipitation reaccumulation, solar radiation by interpolating the atmospheric properties and applying them to the potential solar radiation for this place and time). If there are still missing data points at the requested time steps, it is possible to rely on *data generators* to produce some data out of either parametrizations (such as all-sky incoming long wave parametrization) or very basic strategies

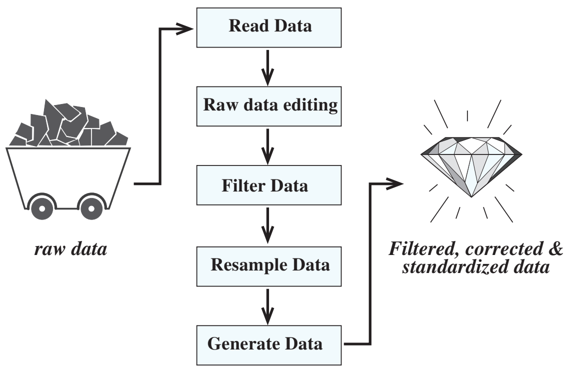

Figure 1: MeteoIO general processing work flow

*Corresponding author address:
Mathias Bavay, Institute for Snow and Avalanche Research SLF, Flüelastr. 11,
CH-7260 Davos Dorf;
tel: +41 81 417 02 65
email: bavay@slf.ch

[1] *Conventional wisdom suggest that data preparation takes about 60 to 80% of the time involved in data mining exercise,* Jermyn et al. (1999)

(such as generating null precipitation to fill gaps).

Another step (not represented here) can take place after the data generators, to compute spatially distributed data over a provided Digital Elevation Model. A wide variety of methods are offered to control how this spatialization takes place.

It is important to keep in mind that during this whole process, MeteoIO works with any sampling rate, including variable sampling rate and can resample to any point in time. Moreover, every step of the process is fully user-configured through an ini file (some tools such as inishell[2] can offer a graphical interface to produce this configuration file). This means that the whole data preparation from raw data to an usable data set is fully described by this ini file that can be used to document what has been done (as in the research context) or to exchange a data preparation workflow between an experimental setup and an operational system.

2. METEOIO'S APPLICATIONS OVERVIEW

Although MeteoIO was originally written to provide data to numerical models, it has steadily been used in broader contexts. Some of these applications are presented in more details below.

2.1. Providing forcings to numerical models

This is the most basic and the original use case: a numerical model needs high quality, gap free data to run properly. By relying on MeteoIO instead of adhoc implementation, the model (or any other application) gains independence from the source of the data and can therefore use forcing data in different formats, for example forcing data in the native format of another model or different data source between research runs, operational nowcasting runs and forecast runs. The application also does not need to worry about quality control of the data and is therefore more focused on its core tasks. There are two ways to use MeteoIO in an application: either by integrating the library directly into it (as is done by Snowpack (Lehning et al. (2002), Fierz et al. (2013)), Alpine3D (Bavera et al. (2014)) or Geotop (Endrizzi et al. (2014))) or by running MeteoIO first (through a very simple application that calls MeteoIO to get the data and write it out) and run the application on the produced data files in a second step (Raleigh and Small. (2017)).

2.2. Virtual stations

When no meteorological parameters are available at the point where the data would be necessary, MeteoIO offers several possibilities to still provide meteorological time series: it is possible to spatially interpolate several time series at the point of interest or extract time series from gridded data sets (such as reanalysis data). This is fully transparent to the application getting its data from MeteoIO that sees the data as if it was coming from any regular data source. This is used for example to extend the temporal validity of measured snow profiles by further simulating them with Snowpack at their measured location (Monti et al. (2016)). This is also used to evaluate snow loads at places where no consistent meteorological records could be available by extracting data out of reanalysis gridded data sets (Nikolov et al. (2018)). Of course, it is still possible to filter the extracted data like from any other source, for example to apply debiasing (such has been done for example to produce forcings at the Urumqi glacier number 1 in China).

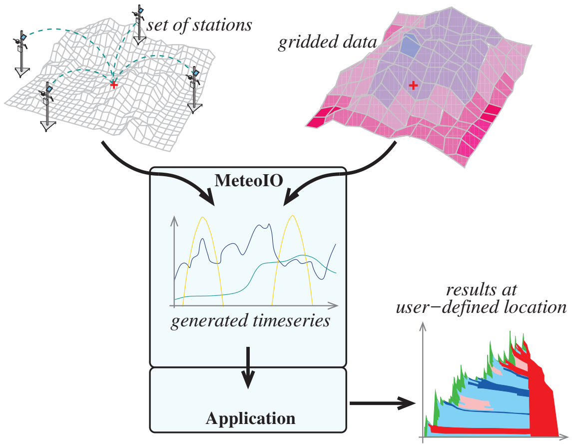

Figure 2: Virtual stations: either by spatially interpolating point data or extracting time series from gridded data

2.3. Chaining models

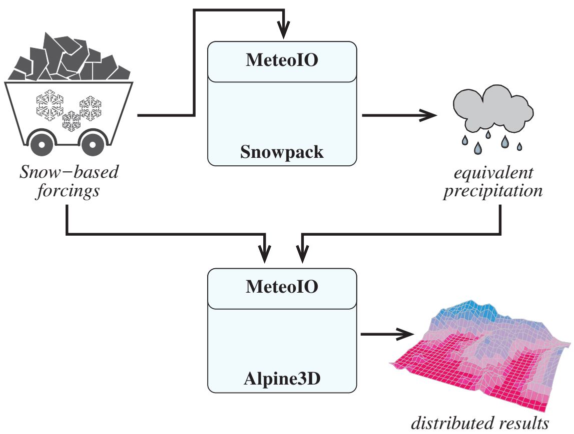

Figure 3: Chaining of models: the output of one model is the input of the next

The integration of all the necessary steps for

[2]https://models.slf.ch/p/inishell/

data preparation into MeteoIO significantly ease the chaining of models (using one model's output as the input of another model). This is specially important when some measured parameters are hard to statistically spatialize (such as the measured snow height, that depends on the cumulative effects of the radiation fields and of the wind fields or the reflected short wave radiation that depends on the surface albedo at each grid cell) and must therefore first be converted into other parameters before spatialization (such as equivalent precipitation or the incoming short wave radiation). MeteoIO allows reading different data formats, excluding some meteorological parameters and merging together data sets into common stations (for example to only take the equivalent precipitation from the first simulation while retaining the original surface temperature as measured) and reading the produced data set again into the second model as spatialized forcings. Fig. 3 shows how this is used at the Vallée de la Sionne test site in order to simulate the snow cover distribution with Alpine3D using Snowpack as first-stage model (Köhler et al. (2018)). Thanks to MeteoIO, this is achieved simply by calling Snowpack, then Alpine3D without the need to tweak any of the data in between.

2.4. Data quality monitoring

Data Quality Tool

Station	PSUM	RSWR	TA	TSG	SUM
BER3	128	128	128	128	512
SCH2	1			336	337
OBW3	47				47
DIA2					0
CAM3	...			...	0
ATT2					0

Figure 4: Data quality information extracted from MeteoIO's logs

MeteoIO has a special mode of operation where it logs every alteration to the raw data set, and provides the station ID, the exact time step, the parameter name and the algorithm that altered the data. This could be a filter that removed a point, a filter that corrected a point or a resampling algorithm that interpolated a missing data point. It is possible to get a good overview of the health of the measurement network by running on all the available stations (128 in the case of the deployment at the Institute for Snow and Avalanche Research SLF (Switzerland) that has been running for the last two winters) over a given time period. An aggregation period of one week has been chosen as a good compromise between reactivity and enough memory of past events so it is not mandatory to look at the extracted results too often. Then another tool aggregates and presents these results to the user. In the SLF test setup, generating the logs only takes 10 to 40 seconds (depending on the database load).

Figure 4 is a screenshot of the SLF setup. Only a very small subset of the stations is shown and some measured parameters have been removed for clarity. One can already identify several issues with the network: the station BER3 shows a relatively high number of errors on almost all parameters, this might be linked to transmission failures. On the contrary, station SCH2 only shows a high number of missing values for one parameter, so here it might be a sensor problem. Stations DIA2 and ATT2 show no missing values while station CAM3 shows the "..." symbol for some parameters, meaning that the said parameters have never had any values (so most probably, these parameters are not measured at this location).

2.5. Data standardization and sharing

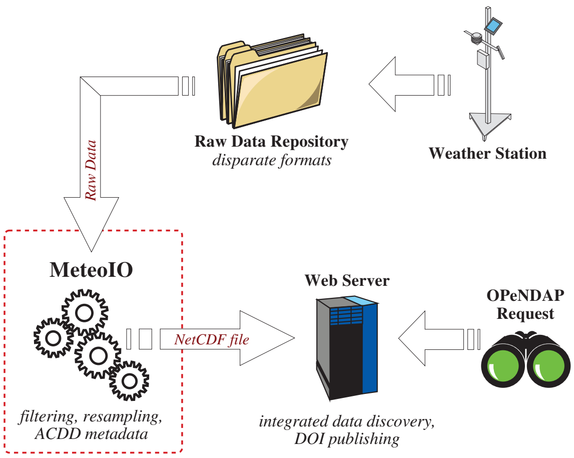

Figure 5: Global Cryosphere Watch data flow: from disparate to standardized and queryable datasets

As MeteoIO can read a variety of data formats (as well as lots of variation within a given format, as is found with CSV or NetCDF), it can serve as a converter from disparate data formats to a unified format. This is the role that it fulfills for the World Meteorological Organization's Global Cryosphere Watch (GCW). GCW is an international mechanism for supporting all key cryospheric in-situ and remote sensing observations. To this effect, it builds up a global network of so called GCW CryoNet stations that report metadata to the GCW data-portal and provide means to access data using a decentralized model. A majority of the measuring stations are operated by universities and research organizations, which are not familiar with the WMO metadata and data exchange mechanisms and do not have the resources to adopt them. Thus MeteoIO offers a way to prepare the data for publication on the GCW data-portal by reading the measurements in their native format as they are generated by the organization in charge of the station, performing some data filtering, filling potential data gaps (all according to the configuration performed by the local organization) and writing out the data in a standardized format and schema, according to WMO recommendations and based on the NetCDF common data model and the CF convention with the ACDD data discovery attributes (Bavay at al. (2018), see Fig. 5).

MeteoIO is also used in this *data hub* role within the PROSNOW EU Horizon project (Morin and Dubois. (2018), Hanzer et al. (2018)) where it converts forcing data from various origins and formats (mostly CSV based) into the native formats of the various models that are used in the project as well as filters, resamples and fills the potential gaps. By adding metadata with very little effort from the user, it can make the prepared data sets easier to identify and distribute and safer to archive.

2.6. Maps generation

Since MeteoIO is also able to spatially interpolate meteorological data in order to deliver spatial field to models that require it, it can also be used to generate maps of various parameters. The user can choose from twenty different algorithms, from simple averaging to kriging, wind fields parametrizations, lidar snow depth distribution weighting of the precipitation distribution or topography and atmospheric-aware solar radiation interpolations. The interpolations fully benefit from C++'s performances and are therefore well suited to batch generation of gridded data. Moreover, several output plugins make it also well suited to web services usages, such as the PNG outputs or the NetCDF outputs with discovery metadata. MeteoIO has been used within an operational context for such tasks in projects such as Sensorscope (Ingelrest et al. (2010)), Switzerland's Gemeinsame Informationsplattform Naturgefahren (GIN, Romang et al. (2009)) and MySnowMaps[3].

3. CONCLUSIONS

The MeteoIO preprocessing library has grown into a robust and flexible swiss army knife to work with meteorological data, both in the research and operational context. Several applications as well as several commercial companies rely on it for their daily operations and the range of its applications is still expanding. Current developments are directed toward a more flexible and refined support of time series of gridded data in order to allow the same kind of filtering and interpolating abilities on gridded data as on point data.

The MeteoIO library is available under the GNU Lesser General Public License v3.0 (LGPL v3) on https://models.slf.ch/p/meteoio/ both as source code (from the source version control system or as packages) or as precompiled binaries for various platforms (Linux™, Apple macOS™, Microsoft Windows™).

[3] http://www.mysnowmaps.com/

ACKNOWLEDGEMENT

This work has been supported by The Swiss Kommission für Technologie und Innovation, the Swiss National Fund (grant IZK0Z2_151680/1), the Swiss Experiment project (CCES grant), The EU Hydrosys project, the AAA/SWITCH funded Swiss Multi Science Computing Grid project as well as the European Union's Horizon 2020 research and innovation pro-gramme under grant agreement No 730203.

REFERENCES

Bavay, M. and Egger, T. J. (2014). MeteoIO 2.4.2: a preprocessing library for meteorological data *Geoscientific Model Development*, 7, pp3135-3151.

Bavay, M., Fiddes, J., Fierz, C., Godøy, Ø. and Nitu, R. (2018). Enhancing data quality and data publication with the MeteoIO library - The example of the Global Cryosphere Watch *European Geosciences Union General Assembly 2018*

Bavera, D., Bavay, M., Jonas, T., Lehning, M. and De Michele, C. (2014). A comparison between two statistical and a physically-based model in snow water equivalent mapping *Advances in water resources*, 63, pp167-178.

Endrizzi, S., Gruber, S., Dall'Amico, M. and 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, pp2831-2857.

Fierz, C., Bavay, M., Wever, N. and Lehning, M. (2013) SNOWPACK: Where do we stand today? *International snow science workshop proceedings 2013*, pp166-169

Hanzer, F., Bavay, M., Carmagnola, C., Ebner, P. P., Lafaysse, M., Lehning, M., Strasser, U., Morin, S.(2018) Simulating snow conditions in ski resorts with the physically based snowpack models AMUNDSEN, Crocus, and SNOWPACK/Alpine3D *International snow science workshop proceedings 2018*

Ingelrest, F., Barrenetxea, G., Schaefer, G., Vetterli, M., Couach, O. and Parlange, M. (2010). Sensorscope: Application-specific sensor network for environmental monitoring *ACM Transactions on Sensor Networks*, 6(2), 17.

Jermyn, P., Dixon, M. and Read, B. J. (1999). Preparing Clean Views of Data for Data Mining *ERCIM Work. on Database Res*, pp1-15.

Köhler, A., Fischer, J.-T., Scandroglio, R., Bavay, M., McElwaine, J. and Sovilla, B. (2018, in review). Cold-to-warm flow regime transition in snow avalanches *The Cryosphere Discussions*, pp1-22.

Lehning, M., Bartelt, P., Brown, B., Fierz, C. and Satyawali, P. (2002). A physical SNOWPACK model for the Swiss avalanche warning: Part II. Snow microstructure *Cold regions science and technology*, 35(3), pp147-167.

Monti, F., Mitterer, C., Steinkogler, W., Bavay, M. and Pozzi, A. (2016) Combining snowpack models and observations for better avalanche danger assessments *International snow science workshop proceedings 2016*, pp343-348

Morin, S., Dubois, G., The PROSNOW Consortium (2018) PROSNOW – Provision of a prediction system allowing for management and optimization of snow in alpine ski resorts *International snow science workshop proceedings 2018*

Nikolov, D., Bavay, M., Lehning, M. and Fierz, C. (2018) Snowload in the Bulgarian mountains: assessment from snow-cover modelling *European Geosciences Union General Assembly 2018*

Raleigh, M. S. and Small, E. E. (2017). Snowpack density modeling is the primary source of uncertainty when mapping basinwide SWE with lidar *Geophysical Research Letters*, 44(8), pp3700-3709.

Romang, H., Binder, P., Spreafico, M. and Petzold, I. (2009). Common Information Platform for Natural Hazards in Switzerland *EGU General Assembly Conference Abstracts*, 11, p2513.