https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

AIDA, a software package for atmospheric data analysis, interpretation, and FAIR data production

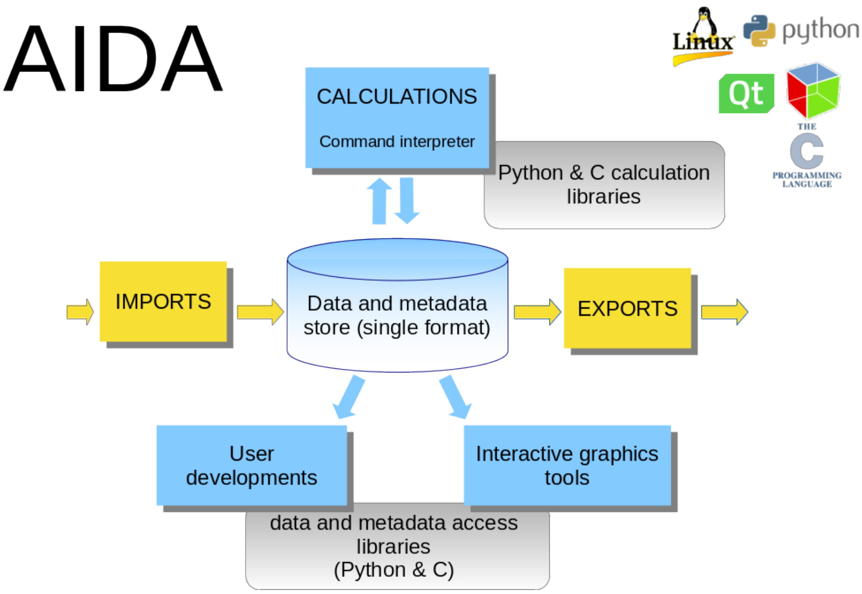
Sylvie BÉNARD[1,*], Bruno PIGUET[1,*], William MAUREL[1,*], Jean-Claude ÉTIENNE[1,*], Jacques COUZINIER[1,*], Pascal RICHARD[1,*], Aurélie RIANDET[1,*], and Arnaud MEQUIGNON[1]

[1]CNRM/GMEI/TRAMM, CNRM, Université de Toulouse, Météo-France, CNRS, Toulouse, France

[*]These authors contributed equally to this work.

Correspondence: Sylvie BÉNARD (sylvie.benard@meteo.fr)

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026



Abstract. AIDA (Analysis and Interpretation of Data from the Atmosphere) is a modular software suite for analysing time series of atmospheric measurements from ground stations, ships, aircraft, drones or other platforms. Users define processing workflows as sequences of mathematical and statistical operators selected from AIDA's library. AIDA records metadata to ensure traceability and produces validated output files (NetCDF, CSV, etc.) that follow international conventions such as CF and ACDD, supporting FAIR principles (Findability, Accessibility, Interoperability, Reuse). Outputs from research projects using AIDA have been published (e.g., (Huynh et al., 2023)). The project stores time series and their metadata together in a self-describing format called the *AIDA Vector*.

Originally written in C, development now uses Python 3, NetCDF4, Udunits2, Qt5/6 and related technologies. This paper describes the *AIDA Vector*, the data organization, the workflow, and the surrounding tools that let users integrate and visualize data at every processing stage. AIDA is suitable for anyone responsible for atmospheric measurements.

1 Introduction

For decades, the CNRM (Centre National de Recherche Météorologique) has used observational resources in atmospheric research to improve weather-forecasting models. The GMEI (Experimental and Instrumental Meteorology Group) implements these observation systems for research and processes the resulting data.

Observation platforms are diverse: conventional weather stations, surface-flux towers, radiosondes, tethered balloons, ground sensors for temperature and humidity, remote sensors (LIDAR, radiometers), aerosol and cloud-particle instruments, and mobile platforms such as aircraft and drones (Dione et al., 2023; Soula et al., 2021; Udina et al., 2020; Calvet et al., 2007; Burnet and Brenguier, 2010). Measurements span fine to mesoscales and sampling rates from 20 Hz down to hourly values.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

AIDA (Analysis and Interpretation of Atmospheric Data) was originally developed (1990–2005) to process research-aircraft data at CAM (Centre d'Aviation Météorologique) and, since 2005, at SAFIRE (Service des Avions Français Instrumentés pour la Recherche en Environnement[1]). From the start, AIDA was designed to be adaptable to many observation systems and future needs. Early experience showed that processing requires end-to-end workflows: acquisition, signal processing, calibration, quality control, validation and visualization, all configurable for different platforms (Piguet et al., 1995). AIDA anticipated tools later developed to meet FAIR principles (Chiriaco et al., 2018; Rautenhaus et al., 2018; Holzinger, 2015; Wang, 2014).

Since its inception, AIDA's source code has been managed under version control systems (initially CVS, then SVN, and now Git).

To standardize processing, AIDA uses a single input format, the *AIDA Vector*, which stores time series together with their metadata. The format has been used for all CNRM campaigns, covering many parameters at various sampling rates. Results from numerous campaigns have been published, including CAPITOUL (Hidalgo et al., 2008; Pigeon, 2007), EUREC4A (Bony et al., 2022; Brilouet et al., 2021), ACROSS (Dewald et al., 2024), MOSAIC P2OA (Zouzoua et al., 2022; Jomé et al., 2023), and Météopole-flux (Carrer et al., 2021; Calvet et al., 2016).

AIDA embeds required metadata at the start of data processing (during raw data ingestion). Consequently, AIDA and its data format ensure traceability of data and metadata and enable reproducible reprocessing years later. Processing methods developed in AIDA are reusable and adaptable.

By default, AIDA processing tools do not overwrite existing data, providing an added layer of safety. AIDA includes an optional file-versioning system that users can enable when needed.

AIDA is fully documented through several volumes: installation manual, getting-started guide, reference guide, user guide, development guide, and operator manual. Each graphical tool includes its own manual.

After a brief graphical overview of AIDA, Section 3 describes the *AIDA Vector* and Section 4 describes the metadata. Section 5 presents the workflow and main processing operators with their tools. Section 6 gives a real post-processing example from a long-term campaign.

2 Quick graphical overview

Figure 1 shows the AIDA data workflow. Data are imported from various formats (CSV, binary, NetCDF, proprietary, etc.) into the *AIDA Vector* format. Calculations and quality checks are performed to validate the data. The resulting FAIR-compliant outputs can then be used by the scientific community.

Via the `aito` graphical interface (see Integrated graphical interface: `aito`), all stages of the workflow — imports, processing, visualization and control, and exports — can be easily monitored.

[1] https://www.safire.fr

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

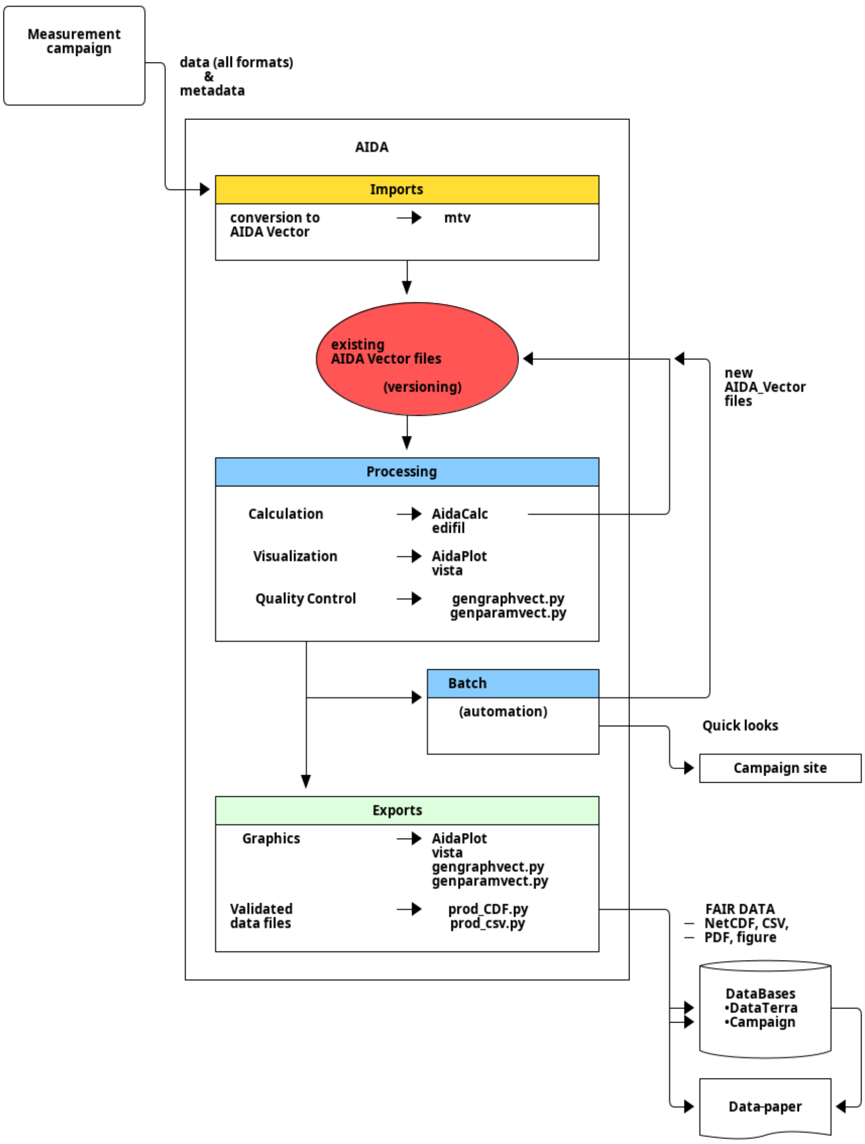


Figure 1. Data flow from acquisition to FAIR data products through AIDA. Boxes represent processing steps and associated AIDA tools; circles indicate *AIDA Vector*. Colors denote stages: yellow = data ingestion into AIDA, red = AIDA Vector, blue = data processing, green = final FAIR products.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

3 The single self-describing file format *AIDA Vector*

Measurements from many sensors and observation systems used during measurement campaigns arrive in diverse formats (CSV, text, proprietary, binary). To process and visualize these data with AIDA, they must first be converted to a single format: the *AIDA Vector*.

An AIDA Vector is a file that stores a time series for a single measured or computed parameter.

It pairs the binary data (raw memory representation) with an ASCII header that describes the data (human- and machine-readable metadata).

This physical coupling ensures metadata always accompanies the data.

Each parameter file has an associated time AIDA Vector giving the timestamp for each sample (the i-th timestamp corresponds to the i-th data value); files that share the same timestamps use the same time AIDA Vector.

To work with data acquired at different times, one of the first steps is to synchronise these data onto a common time base using one of the AIDA library operators via the AidaCalc tool.

Because one parameter = one file, a parameter can be reprocessed independently of an entire archive

Each AIDA Vector includes in its metadata the processing step that produced it. This makes it straightforward to reconstruct a vector's provenance by examining its metadata and that of its ancestors. Such traceability is a key strength of AIDA for producing FAIR data.

The vector format therefore offers great flexibility for data processing and analysis. It allows complete independence from a DBMS and makes the information directly accessible via system commands: the file system effectively acts as AIDA's DBMS.

Figures 2(a–b) show the AIDA Vector structure and examples of multiple AIDA Vectors with different timestamping.

The ASCII header uses INI format. Some fields are mandatory to ensure correct processing in AIDA: the binary data format, the unit (including "count"), the physical dimension, and the data origin (raw or derived by AIDA). Users may add any other useful information.

When converting raw data to an AIDA Vector, it is strongly recommended to include measurement-chain details (equipment serial number, manufacturer, etc.) to ensure traceability of processing and unit tracking.

The names of AIDA Vector files follow a specific nomenclature. They begin with a three-letter family code followed by an underscore (e.g. `tpr_` for temperature, `prs_` for pressure, `rft_` for time reference). Family codes are declared in an INI file. File names use the extension `.vbn`.

Within GMEI the chosen naming convention is:

<family>_<parameter name>_<sensor>_..._<freq>

The `freq` part of the name indicates the acquisition rate: e.g. `20` for 20 Hz, `%60` for one sample every 60 s.

An example of an AIDA Vector header is shown in Section 4.2.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

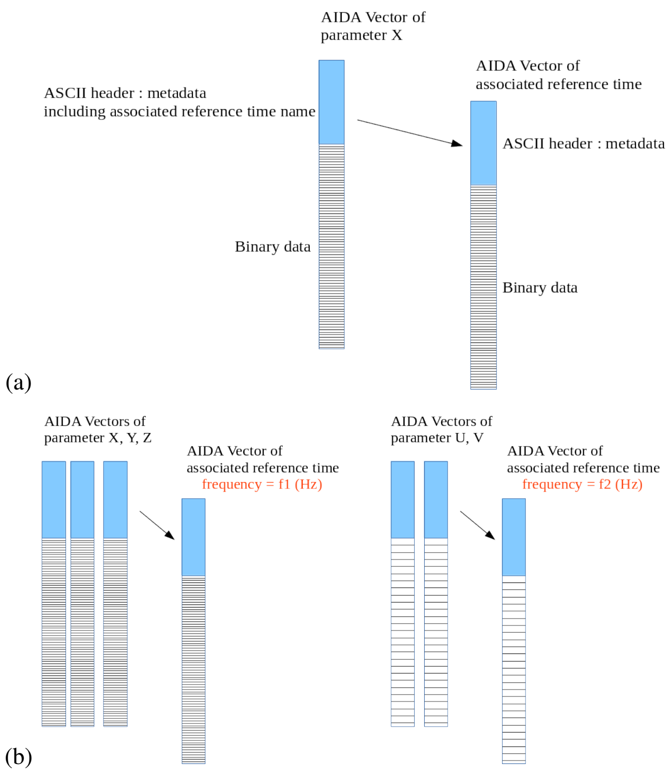


Figure 2. (a) Illustrative representation of the AIDA Vector structure: an ASCII header containing metadata and a binary part that is the computer's memory representation of the data. (b) Illustrative representation of two groups of data acquired over two different time periods.

4 Data and metadata

AIDA manages four levels of metadata:

- at the platform level using a single INI file describing all platforms
- at the parameter level using the self-describing file format *AIDA Vector*, which tightly links parameter data and metadata and its associated time (as described above)
- at the acquisition level using an INI file describing each acquisition
- at the campaign level using an INI file describing each campaign

All these metadata from INI files and AIDA Vectors are stored under two main directories, accessible via two environment variables `AIDA_META` and `AIDA_DATA`. The AIDA libraries (Python and C) provide access to metadata and data files for AIDA tools and user programs.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

A portion of the metadata from these different levels is used by AIDA tools to generate data files that comply with FAIR principles (see an example in "File productions").

4.1 Platform-level metadata

A platform is a coherent set of sensors collocated at the same location (e.g., a drone or a mast).

All platforms are described in a single INI file. Some of the information in this file is:

Parameter	Designation
nom_long	Long description of the platform (one line).
abrev	Platform abbreviation (2 characters).
	Used to build AIDA acquisition IDs (see below).
position	*For a fixed platform:* geographic coordinates (longitude / latitude / altitude in decimal degrees and meters).
	For a mobile platform: the string “mobile”.
nom_serie_mesure	Measurement series identifier (e.g. day, flight, release, release_in_flight).

Table 1. Main fields stored in the platform metadata file (`plate-formes`)

The **abrev** field links parameters to acquisitions and measurement campaigns (similar to a foreign key in DBMS).

Example: part of the description of the platform “lannemezan-mais” is:

```
[lannemezan-mais]
nom_long           = Site Mais MOSAI_P2OA
abrev              = ls
nom_serie_mesure   = Journee
position           = +43.11892 / 0.34714 / 594.0 m
```

4.2 Parameter-level metadata

As shown in Section 3, the header of an AIDA Vector contains all metadata for a parameter.

Example: the header of the AIDA Vector file `hum_sol_10cm_brut_cal_%60.vnb`, which contains the parameter “soil moisture -10 cm” at a frequency of one data point per minute (`%60`), for acquisition `ls230104` (see Section 4.3) of the platform “lannemezan-mais” in the MOSAI-P2OA campaign:

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026


```
[hum_sol_10cm_brut_cal_%60]
name                        = humidité dans le sol à -10cm
english_name                = soil moisture -10cm
vol                         = ls230104
format                      = Ieeefloat_32
dimension                   = 4
dim_array                   = 1
ref_time                    = rft_time_4_%60
misval                      = nan
invval                      = nan
type_ech                    = regulier
freq_ech                    = 0.016666667
unit                        = mV
famille                     = eau_vapeur
origine                     = lecture
position                    = profondeur -0.1m
obs_min                     = 864.0
obs_max                     = 894.0
module_1                    = sensor:Delta T:thetaprobe ML2x:225/070:SHU135:
module_2                    = acquisition_unit:campbell:CR23X:6439:CAC212:
chaine_acquisition          = module_1 module_2
fct_origine                 = mtv.py -F -a ls230104 -c MOSAI_P2OA [...]
endianness                  = little
date_creation               = 21/04/2023 06:29:55
FIN_HDR
```

4.3 Acquisition-level metadata

In AIDA, an acquisition is a set of parameters (stored in AIDA Vectors) collocated during the same period (e.g., a flight or a day) on a single platform. An acquisition is identified by a unique identifier called `acqid`. The `acqid` is an 8-character code in the format `pfyynnnn`, where `pf` is the two-character platform abbreviation, `yy` is the two-digit year of acquisition, and `nnnn` is a four-digit date or serial number for the acquisition.

Metadata for an acquisition are grouped in a single INI file named `acqid.inf`. All `.inf` files are stored together in the same subdirectory.

Example: the file `ls230104.inf` describes the first acquisition of the platform “lannemezan-mais”:

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026


```
[IDENTIFICATION]
acq_id      = ls230104;
campagne    = MOSAI_P2OA;
date        = 14/04/2023;
plate-forme = lannemezan-mais;

[ACQUISITION]
debut_acquisition = 14/04/2023/000100;
fin_acquisition   = 15/04/2023/000000;
```

All parameters of an acquisition are stored in an unique directory whose name depends of the `acqid` of the acquisition.

4.4 Campaign-level metadata

115 A measurement campaign is a coherent set of acquisitions (e.g., all data from the instrumented sites deployed during a field campaign). It is identified by a unique name (e.g., `CAMPAIGN_X`). Its metadata are grouped in a single INI file named `CAMPAIGN_X.camp`.

This file contains all useful information about the campaign, in particular the list of its acquisitions (`acqids`).

All `.camp` files are stored together in the same subdirectory.

120 The following `MOSAI-P2OA.camp` file describes the MOSAI-P2OA campaign, which deployed two platforms: `lannemezan-foret` and `lannemezan-mais`.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026


```
[IDENTIFICATION]
nom                     = MOSAI-P2OA
date_debut              = 17/03/2023
date_fin                = 29/03/2024
projet                  = MOSAI
region_geographique     = Lannemezan, South of France
un_m49_code             = 155
geospatial_lat_min      = 43.11582
geospatial_lat_max      = 43.124228
geospatial_lon_min      = 0.34714
geospatial_lon_max      = 0.363824
description             = MOSAI : Models and Observations for Surface-Atmosphere Interactions.
principal_investigator = XXX : XXX : XXX : XXX ;

[lannemezan-foret]
liste_acqid             = lf230076-lf240078
date_debut              = 17/03/2023
date_fin                = 18/03/2024
principal_investigator = XXX : XXX : XXX : XXX ;

[lannemezan-mais]
liste_acqid             = ls230104-ls240089
date_debut              = 14/04/2023
date_fin                = 29/03/2024
principal_investigator = XXX : XXX : XXX : XXX ;
```

4.5 Metadata relationship diagram

This diagram summarizes the relationships between the platform metadata file, AIDA Vectors, acquisition identifiers (`acqid`), and campaign metadata files. It shows which files reference others (for example, AIDA Vectors include parameter metadata and reference the acquisition `acqid`; the platform file provides `abrev` used in `acqid`; campaign files list their acquisitions). See Figure 3 for a schematic overview.

See an example of the use of these metadata in "File productions".

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

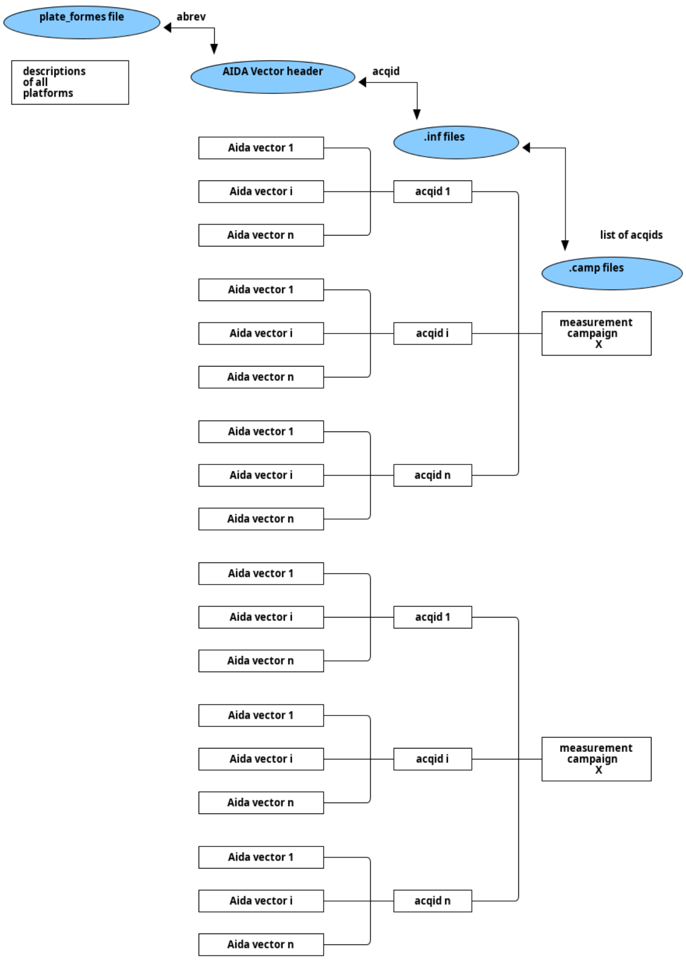


Figure 3. Relationship between plate-formes file, AIDA Vectors, acqids, and measurement campaign data and metadata files

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

5 Workflow and associated tools

5.1 Import: data conversion into AIDA Vector files with `mtv`

As noted above, upstream of AIDA parameters are recorded in various file formats (CSV, binary, NetCDF, proprietary, etc.). Importing data into AIDA requires storing each acquired parameter and its metadata in an AIDA Vector file. This phase is performed using the `mtv` (Miscellaneous To Vectors) tool. `mtv` is an *AIDA Reader module engine*: it generates AIDA Vectors from the information returned by a Reader (see Figure 4).

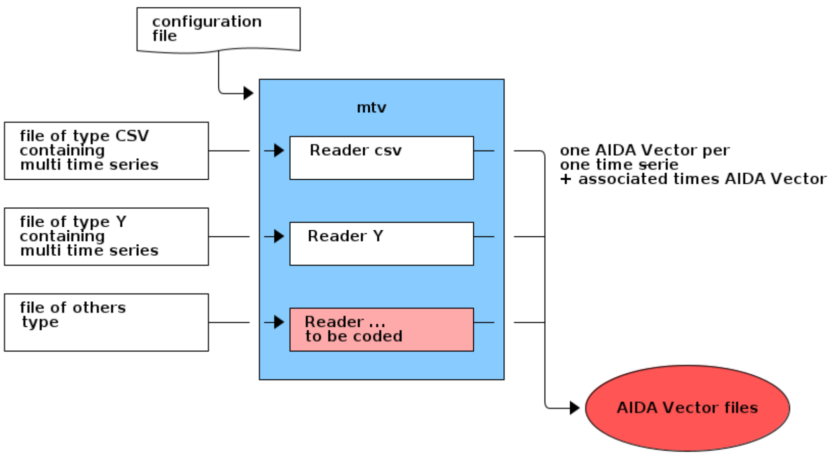


Figure 4. Data and associated times are read by Readers and written by `mtv` into AIDA Vector files. Readers already exist to import data into AIDA from CSV (and its variants), NetCDF, weather model outputs, and proprietary formats. New Readers can be developed if necessary.

A Reader is a Python module that extracts relevant data from a specific format (e.g., CSV) and sends it to `mtv`, which formats and writes the AIDA Vectors. Readers are independent from AIDA and from the AIDA Vector file format (they can be called from any Python program). Programmatically, a Reader is a Python class that inherits from the abstract Reader class and implements, at minimum, two methods: `read_config` and `read_file`. The first method reads and parses the configuration file that describes the data file format. The second method reads the data file contents and, according to the configuration, produces the data to be stored in AIDA Vectors.

Thanks to the modularity of `mtv`, a very wide range of data types can be imported easily as AIDA Vector files.

5.2 Processing: calculations with `AidaCalc`

A *calculation operation* calls an AIDA operator with input and output arguments. Input arguments can be AIDA Vectors or coefficients. Output arguments are AIDA Vectors that contain the calculation results.

Example: Figure 7 shows several real *calculation operations*. One of them is:

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026


```
DensAirsec(pre_air_c1_%60_Int_20, tpr_air_bs_c1_%60_Int_20, tmd_rhoa_c1_20)
```

In this call, `DensAirsec` is the operator name. The operator's documentation (see Section 5.2.3) indicates it takes two input AIDA Vectors (pressure in hPa and air temperature in K or °C) and produces one output AIDA Vector representing the dry-air density in kg/m^3.

Thus the command means: calculate dry-air density from the pressure data `pre_air_c1_%60_Int_20` and the Air temperature data `tpr_air_bs_c1_%60_Int_20` and write the resulting AIDA Vector to `tmd_rhoa_c1_20`.

The `AidaCalc` command interpreter is used to perform *calculation operations* on AIDA Vector files. These operations produce new AIDA Vector files that can serve as input to further calculations. Like all AIDA Vectors, the new files include their own metadata, including the command line used to generate them, so the traceability of data processing can be ensured.

By default, `AidaCalc` never overwrites an AIDA Vector: an existing AIDA Vector is renamed (a version number is appended to its name) before the new AIDA Vector is written, enabling versioning of calculations.

Commands processed by the `AidaCalc` interpreter can be entered interactively or read from ASCII command files. Creating these ASCII command files is made easier by using the `edifil` tool (see Section 5.2.4). Figure 5 shows the operating principle of `AidaCalc`.

5.2.1 Principle of `AidaCalc`

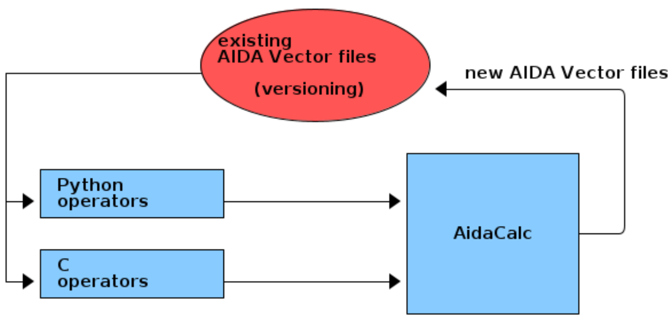


Figure 5. Principle of `AidaCalc` that applies operators on existing AIDA Vectors to generate new AIDA Vectors

In AIDA, an operator has four parts: a calculation function, an interface function between the calculation function and `AidaCalc`, a description of the interface function, and documentation.

The calculation function operates only on data and therefore does not need to know the AIDA Vector. It receives all inputs from the interface function, performs the computation, and returns the results to the interface function. The calculation function may be external to AIDA as long as its inputs and outputs follow this protocol.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

The interface function formats data from `AidaCalc` for the calculation function and then calls it. After the calculation completes, the interface function returns the data and metadata needed by `AidaCalc` to create new AIDA Vector(s).

The interface description allows `AidaCalc` to verify that the command line matches the interface function's expectations. For Python operators, this description is a dictionary. For legacy C operators, it is a function.

The documentation describes the operator, its inputs and outputs, and—when relevant—the calculation formula. It is written in LaTeX and can be pre-generated by an AIDA tool (see Section 5.2.3).

5.2.2 Detailed principle of **`AidaCalc`**

The detailed principle of `AidaCalc`, shown in Figure 6, is as follows:

– 1: using the command line, existing AIDA Vector files, and the interface function description, `AidaCalc` passes data and metadata to the interface function.

– 2: the interface function formats the information for the calculation function.

– 3: the calculation function performs the computations and returns the results to the interface function.

– 4: the interface function formats the received data and prepares the metadata, then returns everything to `AidaCalc`.

– 5: finally, `AidaCalc` generates new AIDA Vector file(s), managing versions.

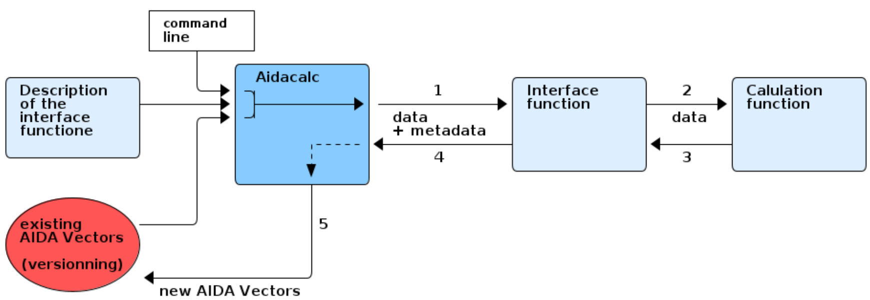


Figure 6. Detailed principle of how `AidaCalc` works.

Thanks to operator parameterization and the syntax replacements performed by `AidaCalc` before calling an operator (see Section 5.2.4 Development of a processing chain with the commands file editor `edifil`), the same command or command file can be applied to several acqids. This is one of `AidaCalc`'s main strengths.

5.2.3 The AIDA Operators catalog

The documentation for each operator is compiled and organized in a single volume: the operators catalog. This PDF is divided into chapters, each documenting a major operator family. Each family groups the documentation for the operators it contains.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

There are eighteen major operator families (see Table 2), for example the *Temporal Transformations* family, which groups operators that synchronise data from different sensors with different timestamps. The *Data Selection* family contains operators for selecting or invalidating data. The *Thermodynamics* family computes the main thermodynamic parameters used in meteorology.

More than four hundred operators have been developed. While the earliest operators were developed in C, they are now implemented in Python; `AidaCalc` can call both C and Python operators.

Basic Mathematical Functions	Spatial Transformations	Dimensional Transformations
Temporal Transformations	Thermodynamics	Precipitation
Microphysics	Radiation	Physical Chemistry
Flow and Turbulence	Geography	Lidar
Detection and Correction of Inhomogeneous Samples	Empirical Estimates	Conversions
Data Selection	Arbitrary Data Modification	Signal Generation

Table 2. The eighteen major operator families

Example: here is the documentation page for the operator `DensAirsec`.

CHAPITRE 5. THERMODYNAMIQUE

5.51 Masse volumique de l'air sec

- **Opérateur :** DensAirsec

- **Objet :** Calcul de la masse volumique («densité») de l'air sec, à partir de la température et de la pression.

Entrées		
P_s	Vecteur, `float`	Pression (hPa)
T_s	Vecteur, `float`	Température de l'air (K ou °C)
Sorties		
$Dens$	Vecteur, `float`	Densité de l'air sec (kg/m^3)

- **Formule**

T_s est converti en Kelvin, si besoin est, puis :

$$Dens = \frac{P_s \times 100}{R_a \times T_s}$$

avec $R_a = 287.05 J.kg^{-1}.K^{-1}$.

- **Commentaires :**

Si on passe une température virtuelle en argument (à la place de T), on obtient (par définition même de Tv) la masse volumique de l'air humide.

- **Auteur :** CNRM/GMEI/TRAMM

- **Références :** Equation d'état d'un gaz parfait, Triplet-Roche [42], page 34.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

5.2.4 Development of a processing chain with the commands file editor `edifil`

A processing chain consists of one or more command files for the `AidaCalc` command interpreter. These files can be written in a text editor, but the graphical AIDA tool `edifil` simplifies creation and fine-tuning.

The main page of `edifil` presents several areas (see Figure 7): an edit area, a search area, a documentation area, an acqid entry area, and a toolbar. The edit area is used to create the command file. Operators and specific command arguments (constants, coefficients, AIDA Vector metadata) can be selected from the search area. A simplified version of the selected operator's documentation is shown in the documentation area.

The acqid entry area selects the acqid on which the user wishes to test or execute the command file, using the toolbar buttons. The `tester` button runs the command file on a copy of the data (dry run), while the `lancer` button runs it on the acqid's real data

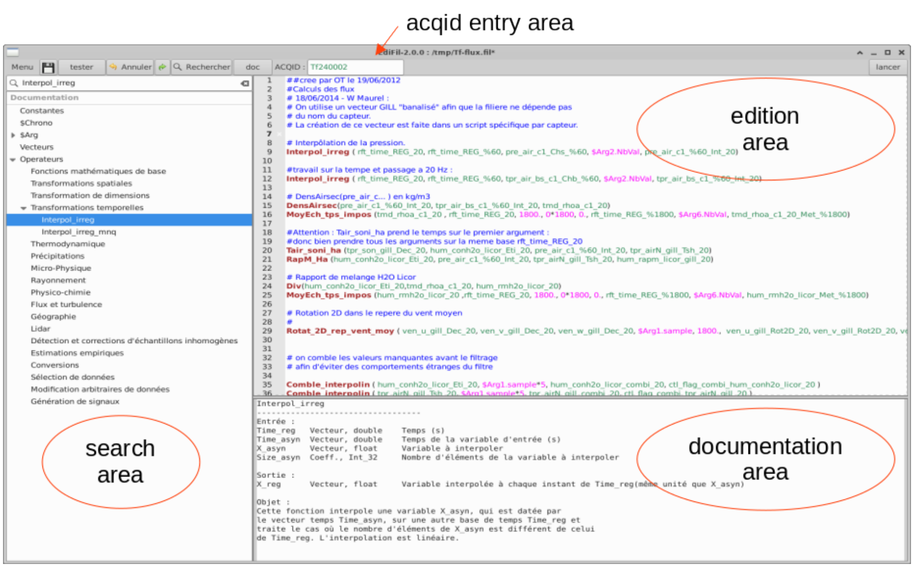


Figure 7. The main page of `edifil` and its different parts

The contents of the edit area are colour-coded: blue for comments, red for operator names, black for operator parameters (as described in the documentation), green for parameter names, and magenta for coefficients.

Input errors are indicated by an icon next to the operator call, with a short description of the error.

While an operator is selected in the search area, `edifil` inserts its standard call syntax into the edit area and displays its documentation in the documentation area.

In Figure 7, at line number nine there is a call to the `Interpol_irreg` operator. As shown in the documentation area, all parameters are inputs except the last one, which is the output parameter. The documentation specifies that the fourth parameter must be an integer coefficient. In the edit area that coefficient appears as `$Arg1.NbVal` (magenta).

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

Before calling `Interpol_irreg`, `AidaCalc` replaces this parameter with "the number of values contained in the first AIDA Vector file of the command line": `rft_time_REG_20`. (`AidaCalc` can perform many similar substitutions.)

As a result, this command line can be executed with any acqid that has the AIDA Vectors `rft_time_REG_20`, `rft_time_REG_%60`, and `pre_air_c1_Chs_%60`, producing the AIDA Vector `pre_air_c1_%60_Int_20`.

5.2.5 Example of a data processing process

We present here an example of data processing based on toulouse-flux platform data from the METEOPOLE-FLUX (*acqid Tf240158*). The AIDA Vector generated from raw wind-speed data measured by the Gill sonic anemometer shows a noisy signal that partly does not correspond to a meteorological phenomenon (see Figure 8(a)).

The processing sequence is:

- detect single-sample jumps above a threshold and replace peaks by the average of the surrounding values (see Figure 8(b)).
- detect and remove (and, when possible, replace) values outside the expected statistical range (see Figure 8(c)).
- apply a digital high-pass filter equivalent to an RC circuit with a -3dB cutoff frequency (see Figure 8(d)).

Each step is implemented by applying an appropriate AIDA operator.

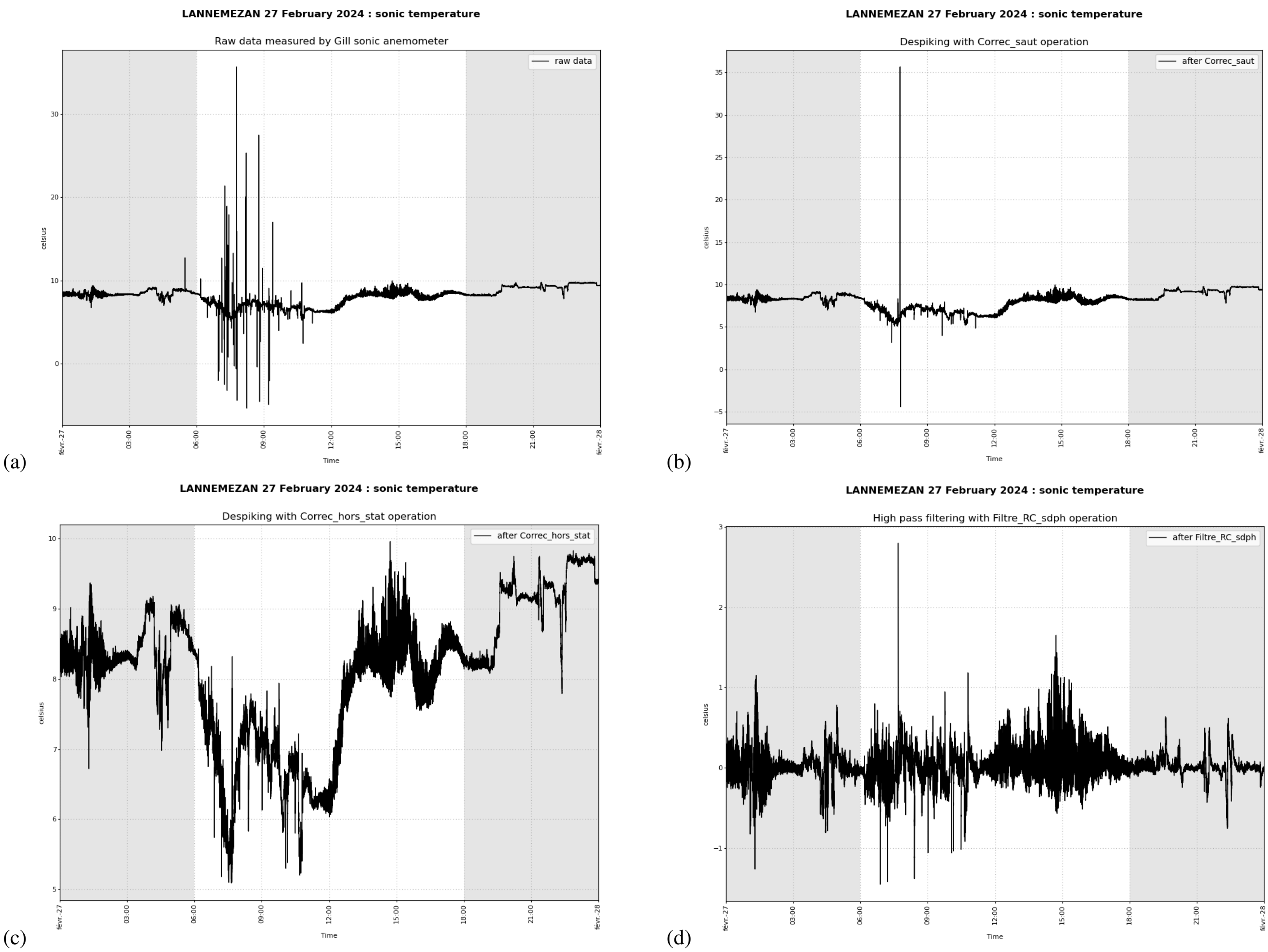


Figure 8. Example of processing wind speed measured by a Gill ultrasonic anemometer to eliminate spikes and to filter the mean wind component for studying turbulence. The successive steps to obtain the final validated data are: (a) raw data, (b) despiking, (c) detection and elimination of values outside a normal statistic, (d) high-pass filter with operator to eliminated mean value.

5.3 Visualizations and Quality Control

AIDA Vectors can be plotted using two complementary tools: `AidaPlot` for AIDA Vector files from a single acqid (which may be high-frequency) and `vista` for viewing AIDA Vector files from multiple acqids (and multiple platforms). Both tools can also generate quicklook files (PDF, PNG, etc.) for scientists; quicklook generation can be automated in batch mode.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

5.3.1 Example of visualization with **AidaPlot**

The `AidaPlot` tool lets you plot data from a single acquisition in various ways. Its geolocation plotting makes it well suited for displaying mobile-platform trajectories. It can also plot the PSD (power spectral density) of an AIDA Vector file.

Plots support zooming and overlaying multiple AIDA Vector files. Titles, legends, and other formatting options are configurable and suitable for batch use.

Figures 9(a–d) show examples of plots produced by `AidaPlot` for different acquisitions.

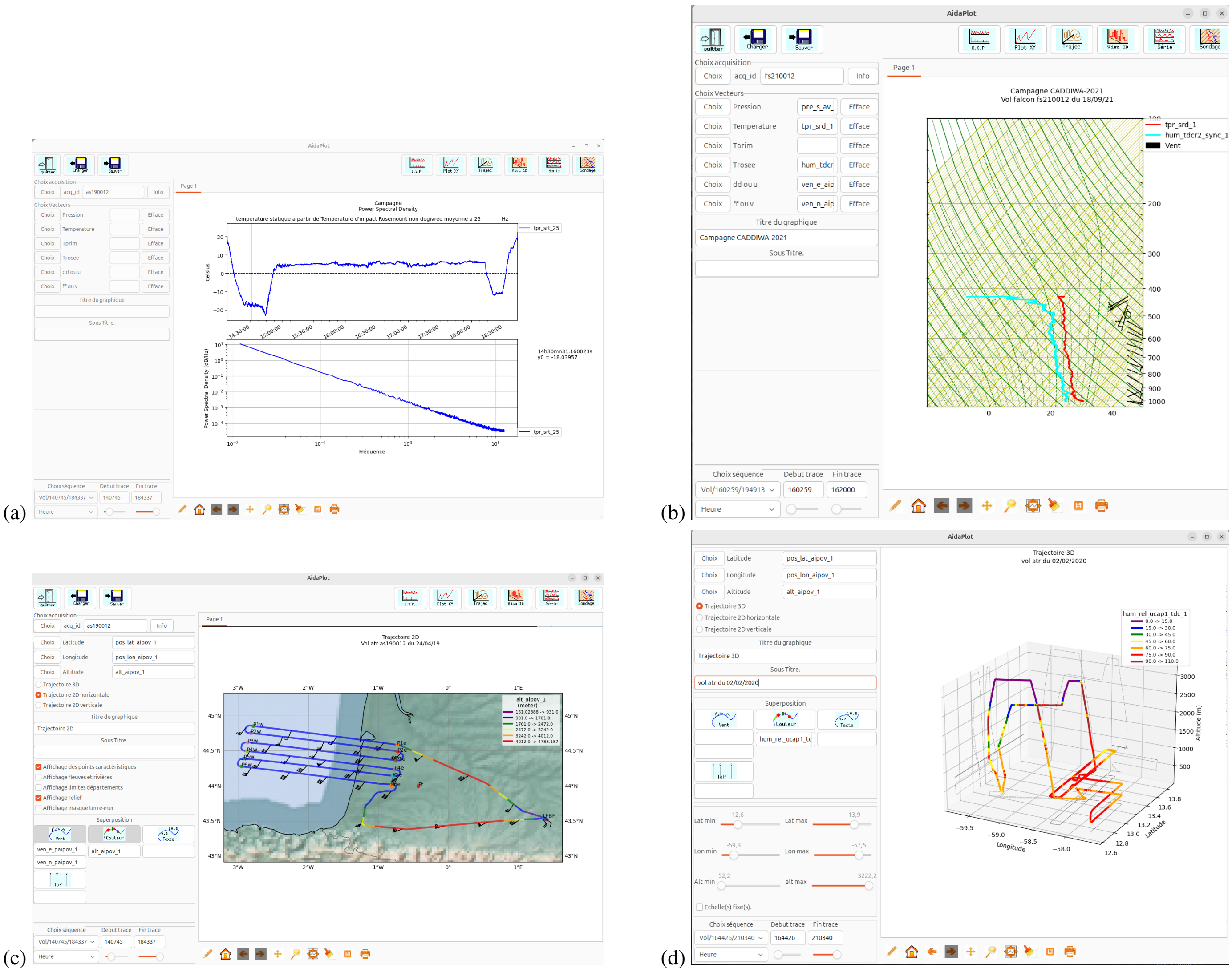


Figure 9. some `AidaPlot` plots : (a) data and its PSD, (b) emagram with wind barbs, (c) 2D trajectory with wind barbs, (d) relative humidity plotted on the 3D trajectory of SAFIRE's ATR42 flight.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

5.3.2 Example of visualization with `vista`

Highly configurable, `vista` lets you create pages containing a wide variety of AIDA Vector plots, for a single acqid or for ranges of acqids or dates. `vista` offers many features, such as data thresholding and simple on-the-fly calculations.

Graphs support zooming and overlaying multiple AIDA Vector files. Titles, legends, and other formatting options are configurable and suitable for batch mode.

`vista`'s ability to plot AIDA Vector files from multiple acqids makes it ideal for tracking parameter variations over several years, which is useful for long measurement campaigns.

Figure 10 shows several graphs for the toulouse-flux platform for March 2024, arranged on a single page.

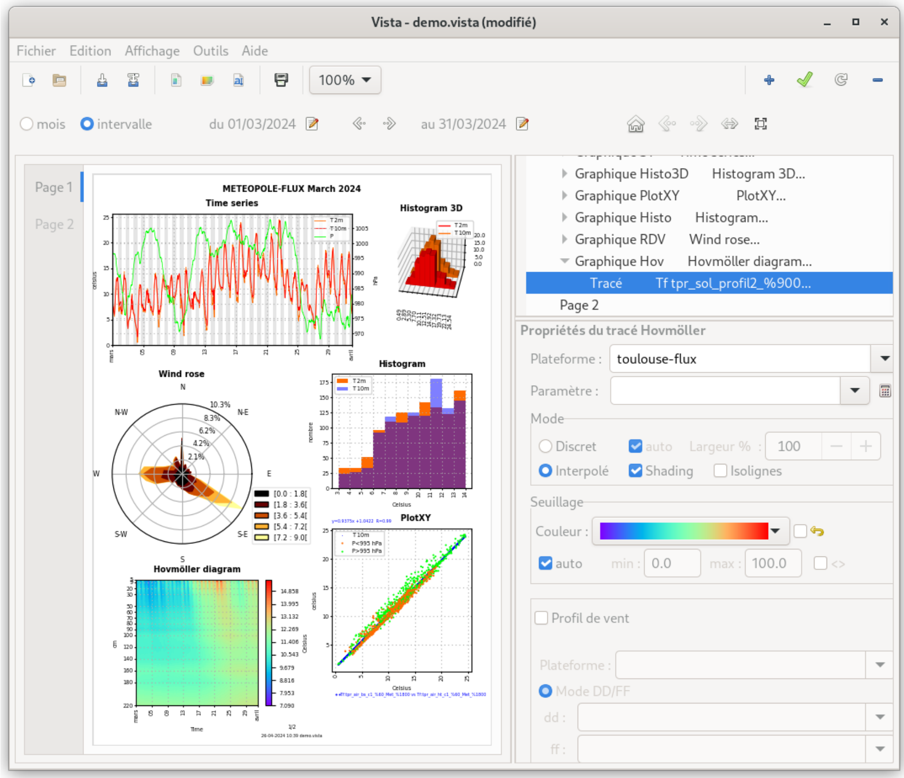


Figure 10. Example of `vista` capabilities

5.4 Traceability and versioning

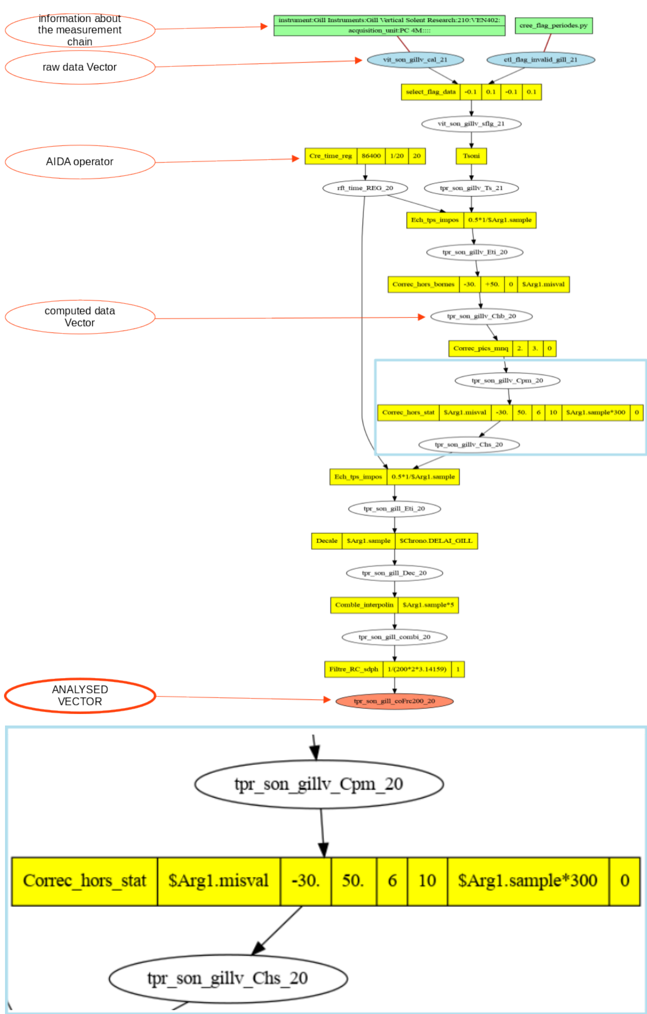
Thanks to the metadata contained in each AIDA Vector file, AIDA ensures traceability of data processing and unit tracking. Versioning of each AIDA Vector file allows old data to be reprocessed. For example, the TRAMM team successfully reworked data over ten years old.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

Using the metadata stored in AIDA Vector files, the script `gengraphvect.py` graphically displays the genealogy of an AIDA Vector file, from raw data to final analysed data. The user can choose the level of detail shown. Measurement-chain information is taken from the AIDA Vector file produced when the raw data were converted.

This capability is useful for reprocessing old data.

Figures 11(a–b) show the genealogy of the AIDA Vector `tpr_son_gill_coFrc200_20` for acqid `Tf23018` (see Figure 8(d)).



(a)

(b)

Figure 11. (a) Genealogy of the `tpr_son_gill_coFrc200_20` AIDA Vector of acqid *Tf23018* . Green, blue, yellow, white and red boxes indicate information about the measurement chain, raw-data AIDA Vector, AIDA operator, calculated AIDA Vector and analysed AIDA Vector respectively, (b) Zoomed view.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

Throughout data processing, AIDA Vector units conform to the International System (SI). Using the same method as above, the script `genparamvect.py` graphically displays unit tracking through the genealogy of an AIDA Vector, from raw data to final analysed data.

This feature helps detect mistakes made during processing.

Figure 12 shows the units genealogy of the AIDA Vector `prp_intensrrmin_dso_Eti_%60` for acqid *Tf230001*.

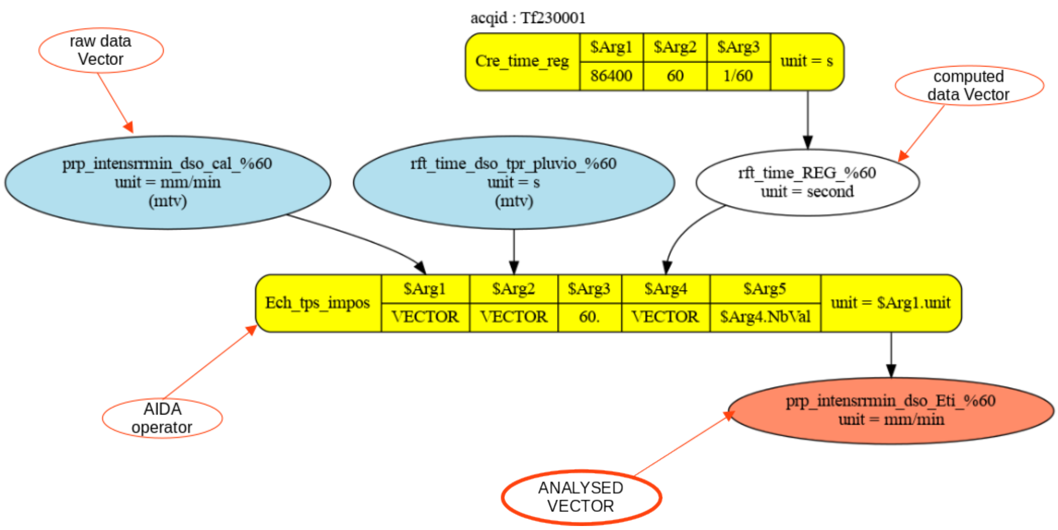


Figure 12. Tracking of units for the AIDA Vector `prp_intensrrmin_dso_Eti_%60` for acqid *Tf230001* (1 January 2023). Blue, yellow, white and red boxes denote raw-data AIDA Vectors, AIDA operators, calculated AIDA Vectors and analysed AIDA Vectors respectively

5.5 Productions

AIDA enables effective participation in open science, in accordance with FAIR principles, by producing validated data files in NetCDF and CSV formats, as well as Quicklooks

5.5.1 File productions

NetCDF files are produced using the `prod_CDF.py` tool. With a single launch, this tool can produce, for a specified period and platform: either a NetCDF file concatenating all acquisitions of the period, or a NetCDF file for each acquisition, or a trajectory-type NetCDF file for each acquisition (for mobile platforms). Inputs come from the command line, configuration files, AIDA Vectors, etc. Outputs are FAIR and comply with the CF and ACDD conventions. In particular, the *standard names* are assigned based on the parameter name and its family.

It is important to note that the period is specified via the launch arguments of `prod_CDF.py`, allowing a single configuration file to produce different NetCDF files. For example:

– `prod_CDF.py ... 01012024 01012024` produces a NetCDF file for 1 January 2024.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

– `prod_CDF.py ... 01012024 31122024` produces a NetCDF file for the whole year 2024.

(the data range can be specified by dates or by acqids).

The configuration file allows adding specific comments to the resulting file.

As shown in the following illustration, `prod_CDF.py` uses metadata from the AIDA Vectors (highlighted in yellow) and the `plate-formes` file (highlighted in green) to generate the NetCDF file. Information highlighted in blue comes from the `prod_CDF.py` configuration file, and information highlighted in orange comes from the command-line arguments. Note that the `source` attribute of a parameter is the result of `prod_CDF.py`'s analysis of the AIDA Vector genealogy for that parameter; it contains descriptions of all sensors that contributed to generating the parameter.

```
netcdf MOSAI_P2OA_MAIZE_CNRM_MTO-FLUX-30MIN_L2_20230414-20240318_V3 {
variables:
    float pa(time) ;
        pa:_FillValue = NaNf ;
        pa:long_name = "air pressure measured at zh_1" ;
        pa:origin = "pre_air_%60_Met_%1800" ;
        pa:standard_name = "air_pressure" ;
        pa:units = "hPa" ;
        pa:source = "Vaisala : PTB210 : B2540003" ;

// global attributes:
        :zh_1 = "0.4 m" ;
        :source_config = "P2OA-MAIZE-30MIN.V0.ccdf" ;
        :acqid = "ls230104-ls240078" ;
        :geospatial_lat_min = 43.11892 ;
        :geospatial_lat_max = 43.11892 ;
        :geospatial_lon_min = 0.34714 ;
        :geospatial_lon_max = 0.34714 ;
        :geospatial_vertical_min = 594. ;
        :geospatial_vertical_max = 594. ;
```

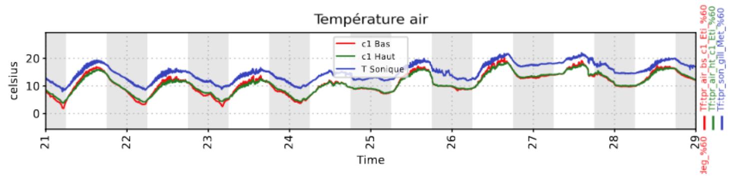
CSV files are produced using the `prod_csv.py` tool. From command-line arguments, a configuration file and AIDA Vectors, `prod_csv.py` generates CSV files. New data can be appended to an existing CSV file.

5.5.2 Graphical productions

Once a processing chain and a batch file for `AidaPlot` and/or `vista` have been created (see "Visualizations and Quality Control"), Quicklook production and delivery to campaign sites can be automated using the standard `cron` scheduler.

Figures 13(a–b) show two Quicklooks automatically generated and delivered from the Météopole-Flux platform (+43.574167 / +1.373889 / 157.0 m) acquisitions.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

(a)


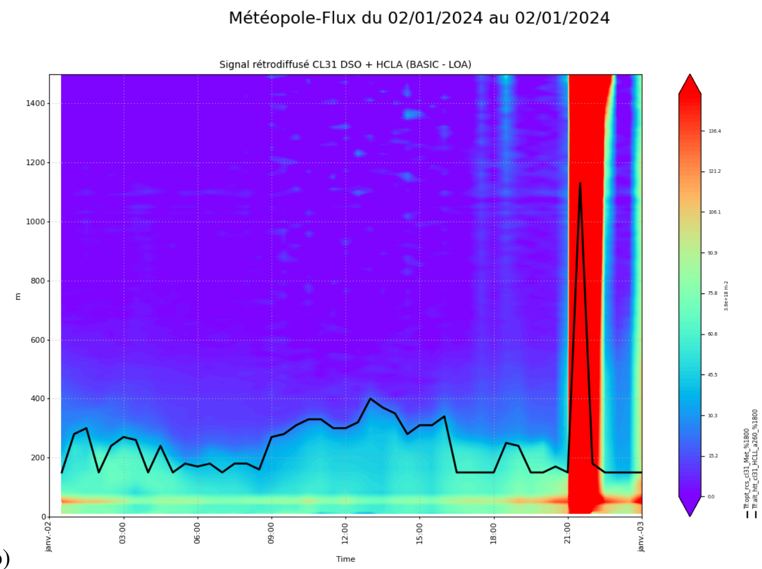
(b)


Figure 13. Examples of Quicklooks (a) Air temperature measured with three probes between 21 and 28 April 2024. (b) Backscattered signal from Vaisala cloud telemeter and atmospheric boundary layer height calculated with LOA (Laboratoire d'Optique Atmosphérique) BASIC software based on Haar Wavelet method (Brooks, 2003) for 2 January 2024.

5.6 Integrated graphical interface: `aito`

The integrated graphical interface `aito` (Aida TOols) provides access to all AIDA data via a user-friendly file navigator. This tool also gives access to all AIDA functionalities.

On the main page, `aito` presents information about data. Data can be searched by acquisition, campaign, date, or by entering an acqid directly. When an acquisition is selected, `aito` displays the AIDA Vector content as graphs. `aito` also allows applying common operators to the data using the `AidaCalc` tool.

The second page allows importing data into AIDA using the `mvt` tool.

The third page allows exporting data from AIDA in NetCDF format using the `prod_CDF.py` tool.

Figures 14(a–c) show several pages of the `aito` graphical user interface (GUI).

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

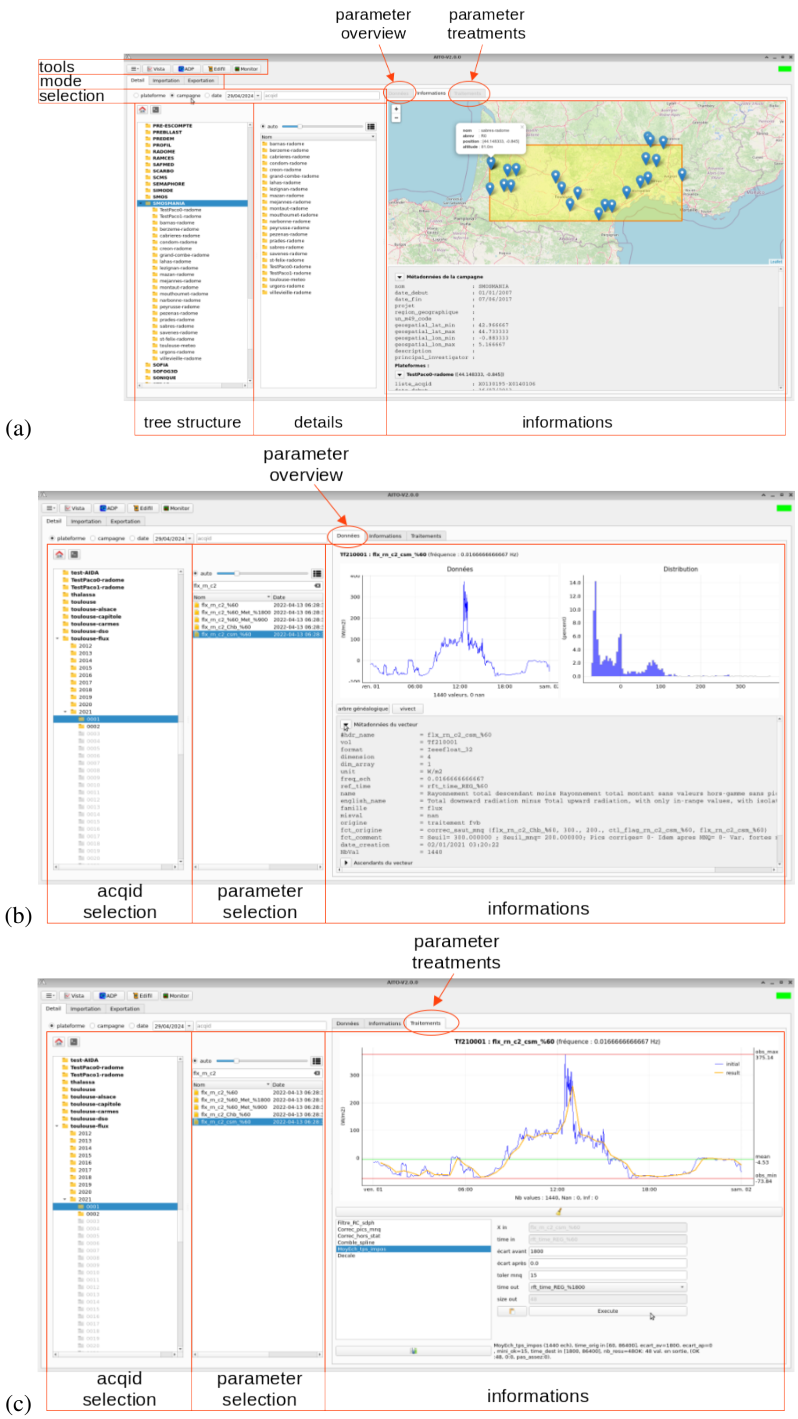


Figure 14. (a) Main page of `aito` for the SMOSMANIA campaign, showing fixed stations on the map and campaign information. (b) Overview of parameter `flx_rn_c2_%60` from acquisition `Tf210001` on 1 January 2021. (c) Data processing of parameter `flx_rn_c2_%60` from acquisition `Tf210001` on 1 January 2021 by the AIDA operator `MoyEch_tps_impos`.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

6 Example of data post-processing on Toulouse Météopole site

AIDA is used to process data from the Météopole observatory located on the Météo-France site in Toulouse. The Météopole instrumented site is a long-term experimental and measurement platform set up in June 2012 for research purposes. It enables long-term monitoring of surface-atmosphere exchanges with the Météopole-Flux project. Its aim is to carry out long-term monitoring of a meadow on the outskirts of an urban area. Its creation is the fruit of the laboratory's long experience in the long-term monitoring of fallow land and grasslands south of Toulouse and in south-western France during the MUREX 1995-1997 (Calvet et al., 1999), SMOSREX 2000-2012 (De Rosnay et al., 2006) and SMOSMANIA projects (Calvet et al., 2007; Zhang et al., 2018). This instrumented platform contributes to the international effort to monitor terrestrial ecosystems, and helps validate surface models developed by CNRM. They are also used to improve and validate Météo-France AROME and ARPEGE weather forecasting models.

The Météopole-Flux station is an associated instrumented site of the ICOS[2] research infrastructure for measurement of greenhouse gases, and as an instrumented site of the ACTRIS-FR[3] infrastructure station network for the measurement of surface fluxes. The Météopole's instrumented site has also been integrated into the MOSAI[4] (Models and Observations for Interactions between the Surface and the Atmosphere) research project, which is studying the modeling of energy exchanges between the surface and the atmosphere as a function of surface heterogeneities. The TRAMM team carries out post-processing and monitoring of the Météopole site with AIDA, and has in particular developed a surface flux processing process. This experience has enabled TRAMM to participate in the evaluation and validation of the configuration of the EddyPro[5] software used by the ACTRIS-FR community to process surface flux stations, in collaboration with the ACTRIS-FR flux working group.

The instrumental set-up consists of a surface flux measurement station built with a sonic anemometer and a Licor fast hygrometer, sonic and conventional wind measurements at 10m, a measurement of the various components of shortwave and longwave radiation, and a pit for measuring temperature and humidity profiles in the soil between 1cm and 220cm (Zhang et al., 2018). The site also benefits from the presence of Météo-France's Direction des Systèmes d'Observations, which provides complementary observations. Meteorological observations and surface flux data have been collected since 2012 and processed with AIDA and available on data center AERIS.

Figures 15 shows an example of some fluxes parameters and meteorological data processed with AIDA over one day.

[2]https://www.icos-cp.eu
[3]https://www.actris.fr
[4]https://mosai.aeris-data.fr/
[5]https://www.licor.com/env/products/eddy-covariance/eddypro

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

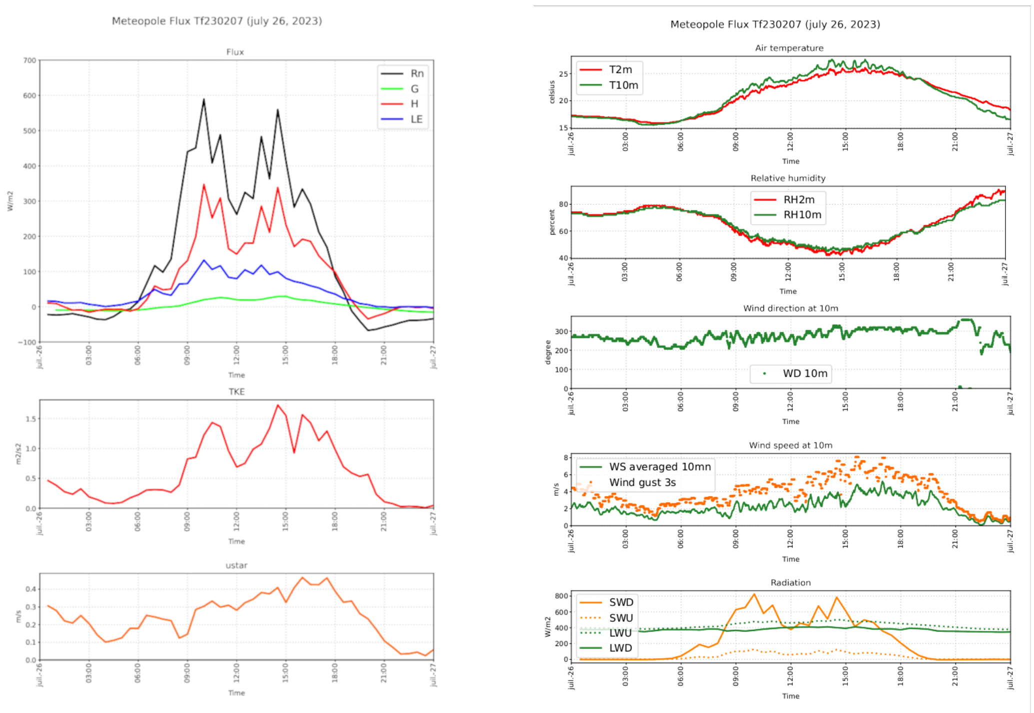


Figure 15. Daily energy balance data and time series of meteo parameters. All parameters are calculated with AIDA script and operator.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

7 Conclusions

AIDA (Analyse et Interprétation de Données Atmosphériques) is a software workshop developed since the 1990's by the TRAMM (Traitement et Analyse de Mesures Météorologiques) team in the CNRM (Centre National de Recherches Météorologiques). Its creation resulted from a request to visualize and process automated data from measurement campaigns that is still relevant. Originally developed for aircraft measurement campaigns, it has been adapted to ground platforms. Thus, since its creation, it has hugely contributed to furnish usable data sets with comprehensible associated metadata in a lot of measurement campaigns. To do so, it is based on its own and unique data format : the AIDA Vector, which guarantees both traceability of metadata and ease to process. The first step in the processing chain with AIDA software workshop is to import data which can come from multiple format. Then, data come through a processing chain peculiar to each parameter of instrument of campaign, based on a succession of operators (calculations that applied to data and could be traced afterwards) that could be freely chosen depending on the need. Finally, processing data could be visualized with AIDA graphical tools such as AidaPlot or vista, tools that could be easily adjustable to control and enhance data.

With its metadata, processing and unit trackings, AIDA could be easily used to fulfill FAIR requirements, increasingly asked by the scientific community, especially when producing NetCDF or CSV files.

After more than 30 years, AIDA is still fully operational and in development. Recently, multi-dimensional data coming from remote sensing instruments (e.g. ceilometer, radiometer) or atmospheric models (e.g. French AROME weather forecast model) have been integrated in AIDA processing chains. The software workshop is also been adapting to new ways of measurements, such as drone profiles.

Code availability. More detailed information and a open version of the AIDA code related to a CeCILL-C license can be obtained at https://github.com/cnrm-tramm/aida_pub.git

Sample availability. Météopole data are available on AERIS/ACTRIS data center with doi = https://doi.org/10.25326/44 for meteorological data and doi = https://doi.org/10.25326/387 for surface fluxes.

Author contributions. Conceptualization : BP, AM; Data curation : all; Formal analysis : all; Project administration : BP, WM, JC-E; Software : all; Validation : all; Visualization : TRAMM team; Writing – original draft preparation : TRAMM team; Writing – review & editing : TRAMM team, BP, WM;

Competing interests. The authors declare no competing interests.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

Acknowledgements. Special thanks to Guylaine CANUT and Quentin LIBOIS for review and support. Thanks to INSU-CNRS (Institut National des Sciences de l'Univers, Centre National de la Recherche Scientifique) and Météo-France for their support of resources and staff.

https://doi.org/10.5194/egusphere-2025-6022
Preprint. Discussion started: 7 January 2026

References

Bony, S., Lothon, M., Delanoë, J., Coutris, P., Etienne, J.-C., Aemisegger, F., Albright, A. L., André, T., Bellec, H., Baron, A., Bourdinot, J.-F., Brilouet, P.-E., Bourdon, A., Canonici, J.-C., Caudoux, C., Chazette, P., Cluzeau, M., Cornet, C., Desbios, J.-P., Duchanoy, D., Flamant, C., Fildier, B., Gourbeyre, C., Guiraud, L., Jiang, T., Lainard, C., Le Gac, C., Lendroit, C., Lernould, J., Perrin, T., Pouvesle, F., Richard, P., Rochetin, N., Salaün, K., Schwarzenboeck, A., Seurat, G., Stevens, B., Totems, J., Touzé-Peiffer, L., Vergez, G., Vial, J., Villiger, L., and Vogel, R.: EUREC4A observations from the SAFIRE ATR42 aircraft, Earth System Science Data, 14, 2021–2064, https://doi.org/10.5194/essd-14-2021-2022, 2022.

Brilouet, P.-E., Lothon, M., Etienne, J.-C., Richard, P., Bony, S., Lernoult, J., Bellec, H., Vergez, G., Perrin, T., Delanoë, J., Jiang, T., Pouvesle, F., Lainard, C., Cluzeau, M., Guiraud, L., Medina, P., and Charoy, T.: The EUREC4A turbulence dataset derived from the SAFIRE ATR 42 aircraft, Earth System Science Data, 13, 3379–3398, https://doi.org/10.5194/essd-13-3379-2021, 2021.

Brooks, I. M.: Finding Boundary Layer Top: Application of a Wavelet Covariance Transform to Lidar Backscatter Profiles, Journal of Atmospheric and Oceanic Technology, 20, 1092–1105, https://doi.org/10.1175/1520-0426(2003)020<1092:FBLTAO>2.0.CO;2, number: 8, 2003.

Burnet, F. and Brenguier, J.: The onset of precipitation in warm cumulus clouds: An observational case-study, Quarterly Journal of the Royal Meteorological Society, 136, 374–381, https://doi.org/10.1002/qj.552, 2010.

Calvet, J.-C., Bessemoulin, P., Noilhan, J., Berne, C., Braud, I., Courault, D., Fritz, N., Gonzalez-Sosa, E., Goutorbe, J.-P., Haverkamp, R., Jaubert, G., Kergoat, L., Lachaud, G., Laurent, J.-P., Mordelet, P., Olioso, A., Péris, P., Roujean, J.-L., Thony, J.-L., Tosca, C., Vauclin, M., and Vignes, D.: MUREX: a land-surface field experiment to study the annual cycle of the energy and water budgets, Annales Geophysicae, 17, 838–854, https://doi.org/10.1007/s00585-999-0838-2, 1999.

Calvet, J.-C., Fritz, N., Froissard, F., Suquia, D., Petitpa, A., and Piguet, B.: In situ soil moisture observations for the CAL/VAL of SMOS: the SMOSMANIA network, in: 2007 IEEE International Geoscience and Remote Sensing Symposium, pp. 1196–1199, IEEE, Barcelona, Spain, ISBN 978-1-4244-1211-2, https://doi.org/10.1109/IGARSS.2007.4423019, 2007.

Calvet, J.-C., Fritz, N., Berne, C., Piguet, B., Maurel, W., and Meurey, C.: Deriving pedotransfer functions for soil quartz fraction in southern France from reverse modeling, SOIL, 2, 615–629, https://doi.org/10.5194/soil-2-615-2016, 2016.

Carrer, D., Meurey, C., Hagolle, O., Bigeard, G., Paci, A., Donier, J.-M., Bergametti, G., Bergot, T., Calvet, J.-C., Goloub, P., Victori, S., and Wang, Z.: Casual Rerouting of AERONET Sun/Sky Photometers: Toward a New Network of Ground Measurements Dedicated to the Monitoring of Surface Properties?, Remote Sensing, 13, 3072, https://doi.org/10.3390/rs13163072, 2021.

Chiriaco, M., Dupont, J.-C., Bastin, S., Badosa, J., Lopez, J., Haeffelin, M., Chepfer, H., and Guzman, R.: ReOBS: a new approach to synthesize long-term multi-variable dataset and application to the SIRTA supersite, Earth System Science Data, 10, 919–940, https://doi.org/10.5194/essd-10-919-2018, 2018.

De Rosnay, P., Calvet, J.-C., Kerr, Y., Wigneron, J.-P., Lemaître, F., Escorihuela, M. J., Sabater, J. M., Saleh, K., Barrié, J., Bouhours, G., Coret, L., Cherel, G., Dedieu, G., Durbe, R., Fritz, N. E. D., Froissard, F., Hoedjes, J., Kruszewski, A., Lavenu, F., Suquia, D., and Waldteufel, P.: SMOSREX: A long term field campaign experiment for soil moisture and land surface processes remote sensing, Remote Sensing of Environment, 102, 377–389, https://doi.org/10.1016/j.rse.2006.02.021, 2006.

Dewald, P., Seubert, T., Andersen, S. T., Türk, G. N. T. E., Schuladen, J., McGillen, M. R., Denjean, C., Etienne, J.-C., Garrouste, O., Jamar, M., Harb, S., Cirtog, M., Michoud, V., Cazaunau, M., Bergé, A., Cantrell, C., Dusanter, S., Picquet-Varrault, B., Kukui, A., Xue, C., Mellouki, A., Lelieveld, J., and Crowley, J. N.: NO $_3$ reactivity during a summer period in a temperate forest below and above the canopy, https://doi.org/10.5194/egusphere-2024-1223, 2024.

Dione, C., Haeffelin, M., Burnet, F., Lac, C., Canut, G., Delanoë, J., Dupont, J.-C., Jorquera, S., Martinet, P., Ribaud, J.-F., and Toledo, F.: Role of thermodynamic and turbulence processes on the fog life cycle during SOFOG3D experiment, Atmospheric Chemistry and Physics, 23, 15 711–15 731, https://doi.org/10.5194/acp-23-15711-2023, 2023.

Hidalgo, J., Pigeon, G., and Masson, V.: Urban-breeze circulation during the CAPITOUL experiment: observational data analysis approach, Meteorology and Atmospheric Physics, 102, 223–241, https://doi.org/10.1007/s00703-008-0329-0, 2008.

Holzinger, R.: PTRwid: A new widget tool for processing PTR-TOF-MS data, Atmospheric Measurement Techniques, 8, 3903–3922, https://doi.org/10.5194/amt-8-3903-2015, 2015.

Huynh, F., Chaljub, E., Quimbert, E., Henry, P., Puissant, A., Ostanciaux, E., Schmidt, S., Payan, S., Biagiotti, I., and Chambodut, A.: Data Terra, the French research e-infrastructure for Earth system and environment, https://doi.org/10.57757/IUGG23-3458, publisher: GFZ German Research Centre for Geosciences, 2023.

Jomé, M., Lohou, F., Lothon, M., Canut, G., Couvreux, F., Brut, A., Derrien, S., Maurel, W., Etienne, J.-C., Vial, A., and Garrouste, O.: Evaluation of the representativity of reference long-term surface flux measurements in an heterogeneous landscape : the Météopole campaign (MOSAI project), https://doi.org/10.5194/ems2023-74, 2023.

Pigeon, G.: Les échanges surface-atmosphere en zone urbaine - projets Clu-escompte et Capitoul, Ph.D. thesis, http://www.theses.fr/2007TOU30243, 2007.

Piguet, B., Brenguier, J.-L., and Rodi, A. R.: A Plan for a Shared Library for Aircraft Processing Software, Bulletin of the American Meteorological Society, 76, 742–744, http://www.jstor.org/stable/26232399, publisher: American Meteorological Society, 1995.

Rautenhaus, M., Bottinger, M., Siemen, S., Hoffman, R., Kirby, R. M., Mirzargar, M., Rober, N., and Westermann, R.: Visualization in Meteorology—A Survey of Techniques and Tools for Data Analysis Tasks, IEEE Transactions on Visualization and Computer Graphics, 24, 3268–3296, https://doi.org/10.1109/TVCG.2017.2779501, 2018.

Soula, S., Pineda, N., Georgis, J.-F., Leroy, A., Vanpoucke, I., Montanyà, J., Casellas, E., Gonzalez, S., and Bech, J.: On the conditions for winter lightning at the Eagle Nest Tower (2537 m asl) during the Cerdanya-2017 field experiment, Atmospheric Research, 247, 105 208, https://doi.org/10.1016/j.atmosres.2020.105208, 2021.

Udina, M., Bech, J., Gonzalez, S., Soler, M. R., Paci, A., Miró, J. R., Trapero, L., Donier, J. M., Douffet, T., Codina, B., and Pineda, N.: Multi-sensor observations of an elevated rotor during a mountain wave event in the Eastern Pyrenees, Atmospheric Research, 234, 104 698, https://doi.org/10.1016/j.atmosres.2019.104698, 2020.

Wang, Y. Q.: MeteoInfo: GIS software for meteorological data visualization and analysis: Meteorological GIS software, Meteorological Applications, 21, 360–368, https://doi.org/10.1002/met.1345, 2014.

Zhang, S., Calvet, J.-C., Darrozes, J., Roussel, N., Frappart, F., and Bouhours, G.: Deriving surface soil moisture from reflected GNSS signal observations from a grassland site in southwestern France, Hydrology and Earth System Sciences, 22, 1931–1946, https://doi.org/10.5194/hess-22-1931-2018, 2018.

Zouzoua, M., Bastin, S., Chiriaco, M., Lothon, M., Lohou, F., Barthès, L., Mallet, C., Derrien, S., Cheruy, F., Bazile, E., Polcher, J., Roehrig, R., and Canut, G.: Using artificial neural network to better evaluate surface turbulent heat fluxes in weather and climate numerical models, https://doi.org/10.5194/ems2022-472, 2022.