

Providentia

An evaluation software package for the atmospheric modelling community

Dene Bowdalo¹

Alba Vilanova¹

Paula Serrano Sierra¹

Amalia Vradi¹

Francesco Benincasa¹

Oriol Jorba¹

Carlos Pérez García-Pando^{1,2}

¹Barcelona Supercomputing Center, Barcelona, Spain

²Catalan Institution for Research and Advanced Studies, Barcelona, Spain

About

Providentia is an evaluation software package designed for the **in-depth analysis of in-situ surface observations and colocated model output**, tailored specifically for the atmospheric modelling community.

It offers a variety of use modes:

- A download mode that automatically retrieves and formats observational and model datasets.
- An interpolation mode that spatially colocates model output with observational stations.
- An interactive dashboard for quick-look visualisations.
- A report mode designed for more exhaustive evaluations.
- A library mode that enables the Providentia backend to be used in Python scripts and notebooks, for example for reading, filtering, or plotting data.

REPOSITORY



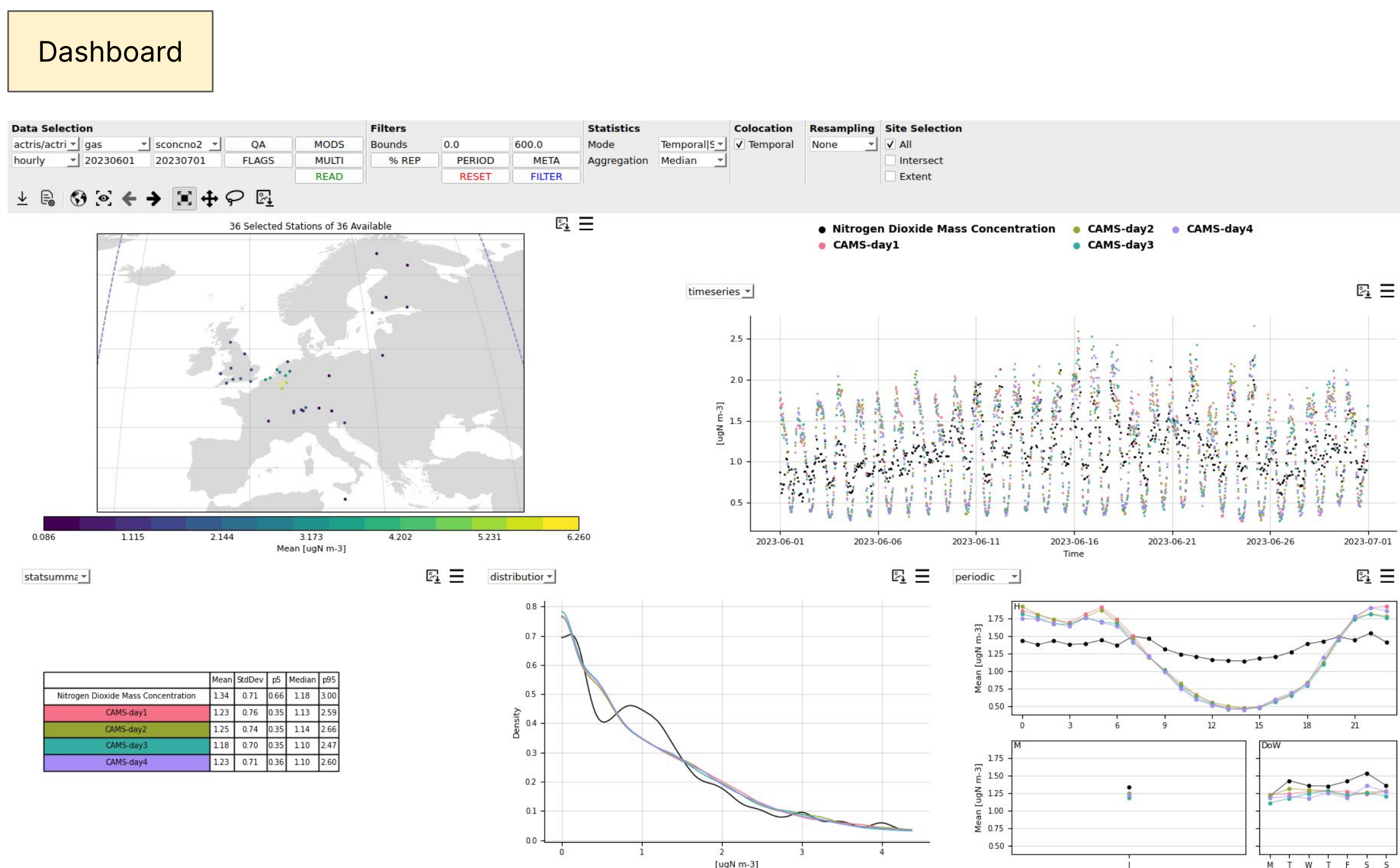
github.com/bsc-es/providentia

DOCUMENTATION



providentia.readthedocs.io

Analysis modes



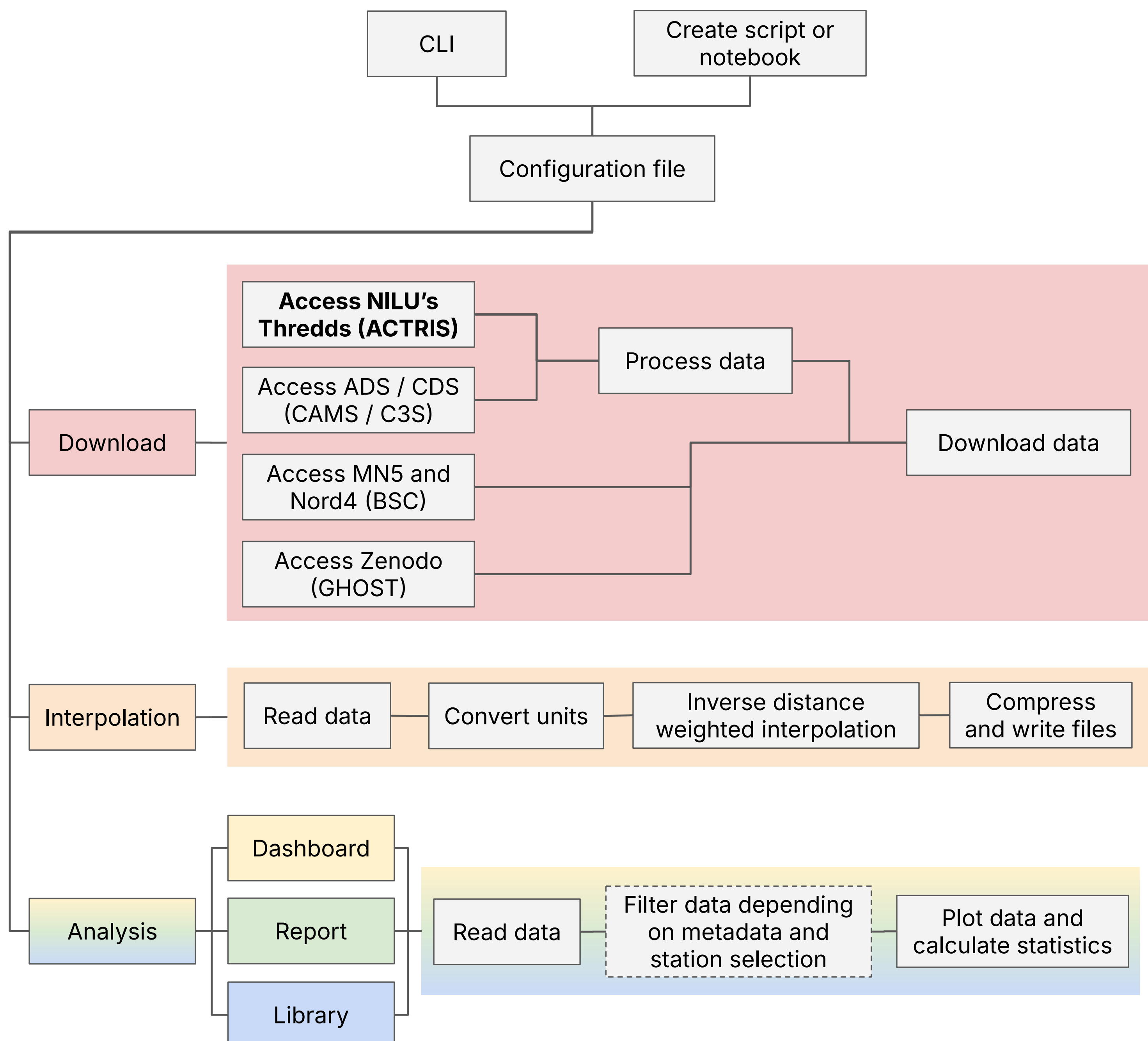
Report

With a configuration file with multiple subsections (e.g. to filter the data by country) we can create subsection plots in a PDF using the report mode.

In relation to ACTRIS

It has been designed to natively handle ACTRIS data. Using a single configuration file, users can **download, format, and analyse ACTRIS observations**, either to inspect the measurements themselves or to compare them with model output. Providentia's capabilities are also highlighted in the ACTRIS VRE, where a dedicated notebook demonstrates its features using ACTRIS data (<https://dc.actris.nilu.no/vre>).

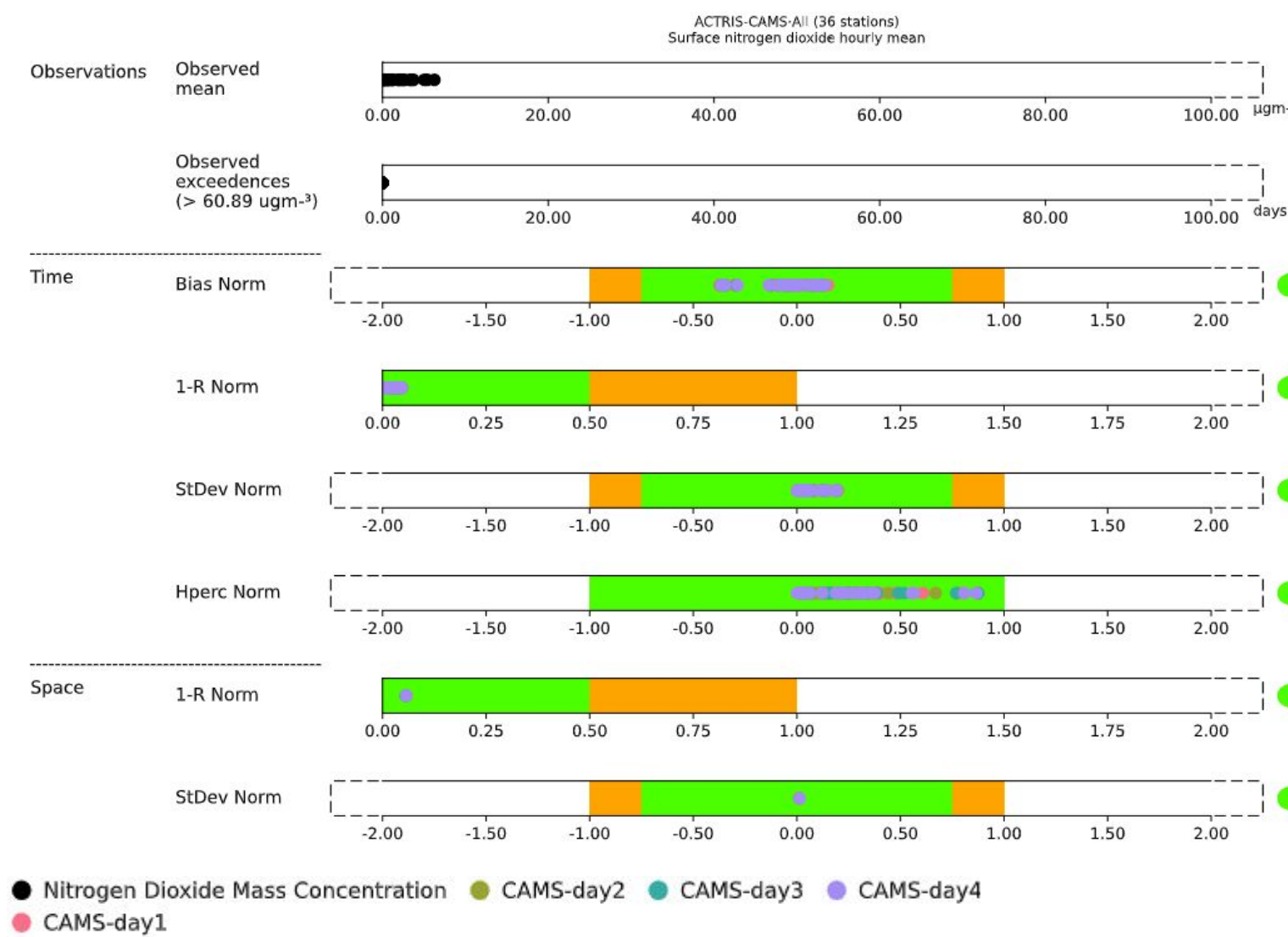
Workflow



STATISTICS SUMMARY

		Mean	StdDev	p5	Median	p95
All	Nitrogen Dioxide Mass Concentration	1.34	0.71	0.66	1.18	3.00
	CAMS-day1	1.23	0.76	0.35	1.13	2.59
	CAMS-day2	1.25	0.74	0.35	1.14	2.66
	CAMS-day3	1.18	0.70	0.35	1.10	2.47
	CAMS-day4	1.23	0.71	0.36	1.10	2.60
Germany	Nitrogen Dioxide Mass Concentration	0.17	0.09	0.06	0.15	0.35
	CAMS-day1	1.08	0.62	0.31	1.01	2.05
	CAMS-day2	1.07	0.58	0.30	1.03	2.00
	CAMS-day3	1.05	0.56	0.30	1.05	1.91
	CAMS-day4	1.10	0.62	0.28	1.07	2.24
UK	Nitrogen Dioxide Mass Concentration	1.59	0.77	0.75	1.37	3.07
	CAMS-day1	1.35	0.83	0.42	1.16	3.06
	CAMS-day2	1.36	0.78	0.41	1.20	2.90
	CAMS-day3	1.31	0.76	0.40	1.19	2.80
	CAMS-day4	1.34	0.75	0.41	1.21	2.86
Switzerland	Nitrogen Dioxide Mass Concentration	1.18	0.40	0.56	1.13	1.86
	CAMS-day1	1.23	0.72	0.36	1.14	2.43
	CAMS-day2	1.24	0.74	0.36	1.12	2.48
	CAMS-day3	1.20	0.70	0.37	1.13	2.33
	CAMS-day4	1.21	0.70	0.37	1.15	2.36

FAIRMODE STATISTICS SUMMARY



Library

```

import providentia as prv

Load configuration file

We are going to load hourly data for nitrogen dioxide mass concentration from the ACTRIS framework and CAMS forecast data for the year June 2023.

prv = prv.Providentia('actris_vre_cams.conf')

prv.print_config()

[ACTRIS-CAMS]
framework = actris/actris
species = nitrogen dioxide mass concentration
resolution = hourly
model = cams_forecast_ensemble (CAMS)
domain = regional
start_date = 20230601
end_date = 20230701
report_type = standard
report_summary = True
report_stations = False
report_filename = PROVIDENTIA_Report
report_title = Report
temporal_colocation = True
observations_data_label = Nitrogen Dioxide Mass Concentration
forecast = day

Download data

Now we format and download the data from NILU Thredds server
(https://thredds.nilu.no/thredds/catalog/ebas_doi/catalog.html) into the path defined in settings/data_paths.json for local machines.

prv.download()

Starting Providentia download...

Download ACTRIS framework data from EBAS DOI...

- /home/pserrano/data/providentia/obs/honghost/actris/actris/hourly/sconcn02, source: https://prod-actris-md.nilu.no/netdata/content/nitrogen_dioxide_mass_concentration
Processing station data: 100%
40/40

Interpolate data

In order to see the model data in the observations space, we need to interpolate it. Inside this function we do a nearest-neighbor interpolation.

prv.interpolate()

```

Plot

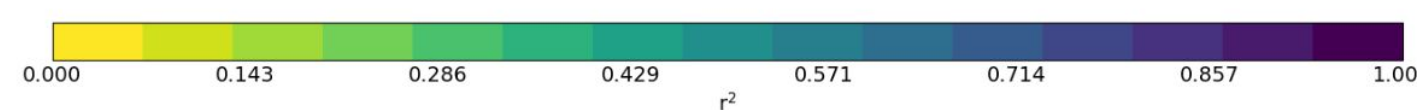
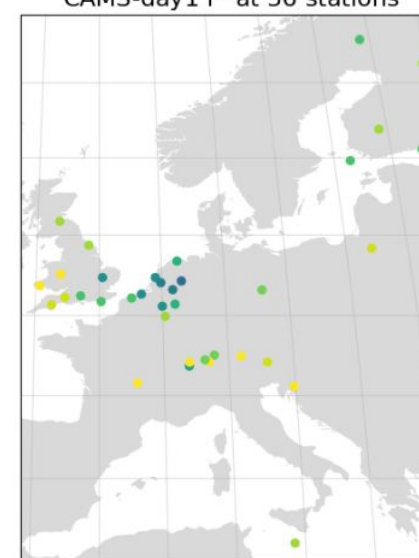
The statistics that can be plotted are: Mean, Median, StdDev, Var, Min, Max, NData, Data%, NStations, Exceedances, p1, p5, p10, p25, p75, p90, p95, p99, NDA8, MB, NMB, ME, NME, MME, MFE, MFE, RMSE, NRHSE, COE, FAC2, IOA, r, r2 and UPA.

```

prv.plot('map-r2', label='Nitrogen Dioxide Mass Concentration', labelb='CAMS-day1')

Warning: Width and/or height have not been passed. The default values will be set.
CAMS-day1 r² at 36 stations

```



Save to netCDF

```

prv.save(format='nc')

Data saved to /home/pserrano/providentia/saved_data/PRV_20260119_1638.nc

```

Save to Numpy

```

prv.save(format='np')

Data saved to /home/pserrano/providentia/saved_data/PRV_20260119_1638.npz

```

TUTORIALS

github.com/BSC-ES/providentia/tree/master/tutorials

