



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



Providentia Training Session 1

25/02/2026

Paula Serrano | Alba Vilanova | Dene Bowdalo

Introduction



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Data Selection
 EEA_AQ_eRi: gas | sconco3 | QA | EXPS
 hourly | 20180101 | 20190101 | FLAGS | MULTI | READ

Filters
 Bounds: 0.0 | 400.0
 % REP | PERIOD | META | RESET | FILTER

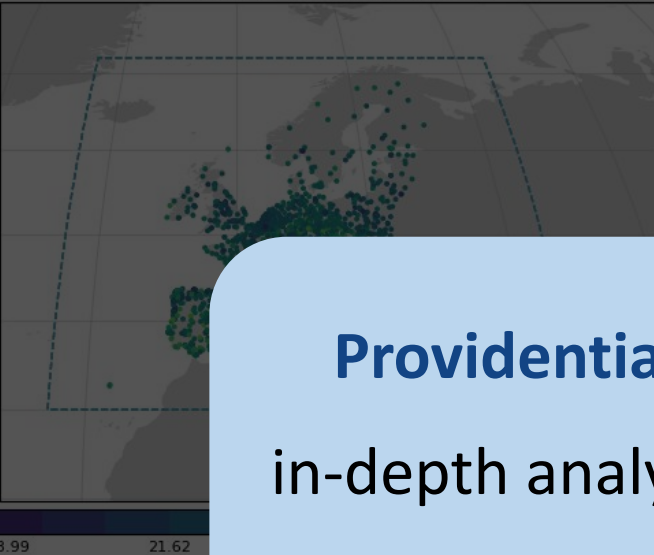
Statistics
 Mode: Temporal | St | Aggregation: Median

Colocation
 Temporal

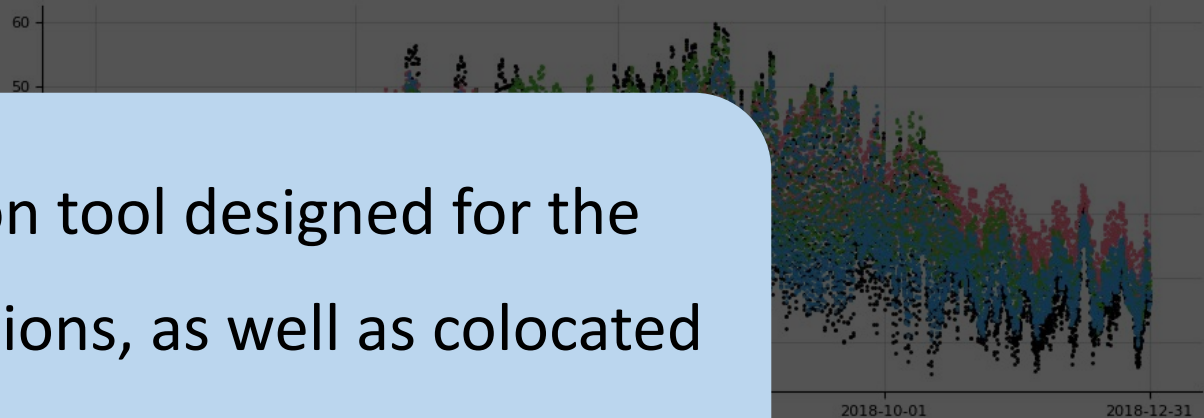
Resampling
 None

Site Selection
 All
 Intersect
 Extent

2241 Selected Stations of 2241 Available



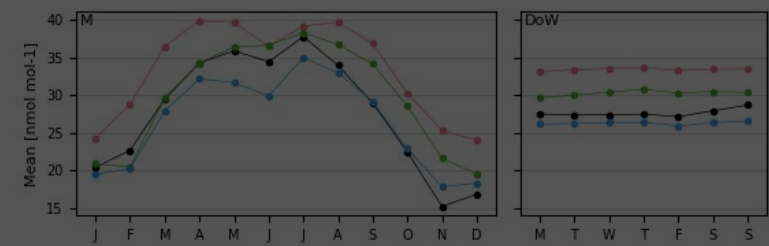
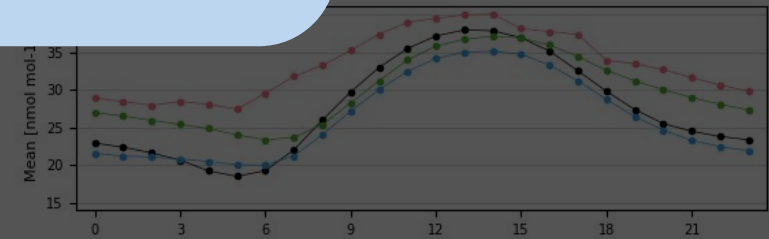
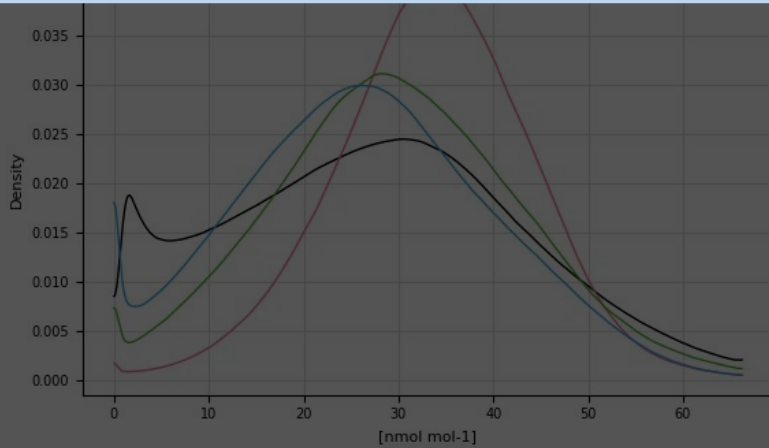
timeseries



Providentia is an evaluation tool designed for the in-depth analysis of observations, as well as colocated model output.

statsummar

	Mean	StdDev	p5	Median	p95
observations	27.68	14.60	4.31	27.03	54.13
EMEP	33.44	9.52	17.35	33.25	49.48
MONARCH	30.27	11.96	11.55	29.21	51.06
SILAM	26.29	12.31	7.66	25.27	49.34



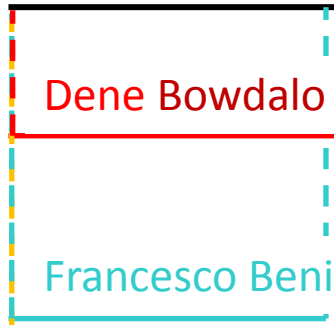
2019 2020 2021 2022 2023 2024 2025 2026



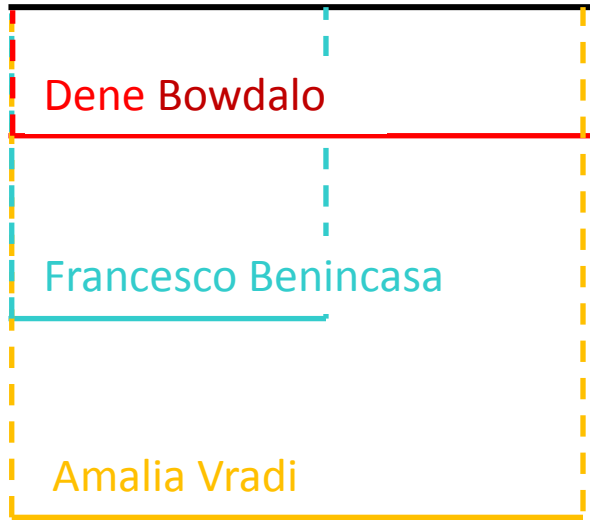
Dene Bowdalo



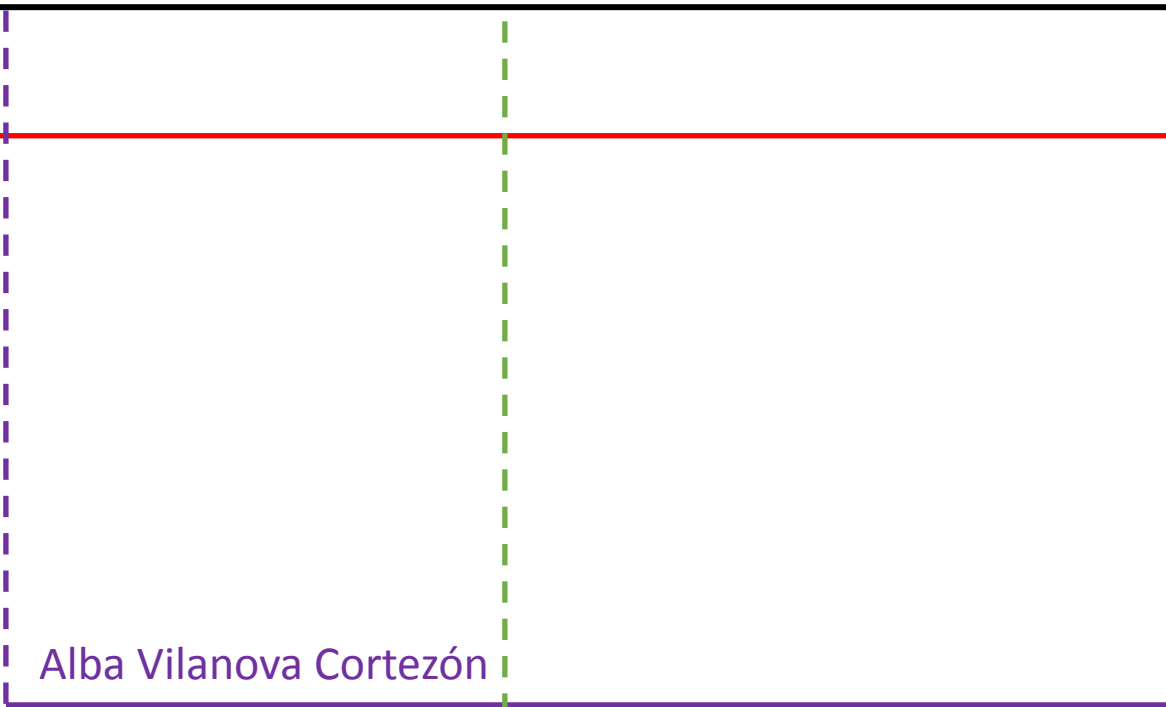
Francesco Benincasa



Amalia Vradi



Alba Vilanova Cortezón



Paula Serrano Sierra



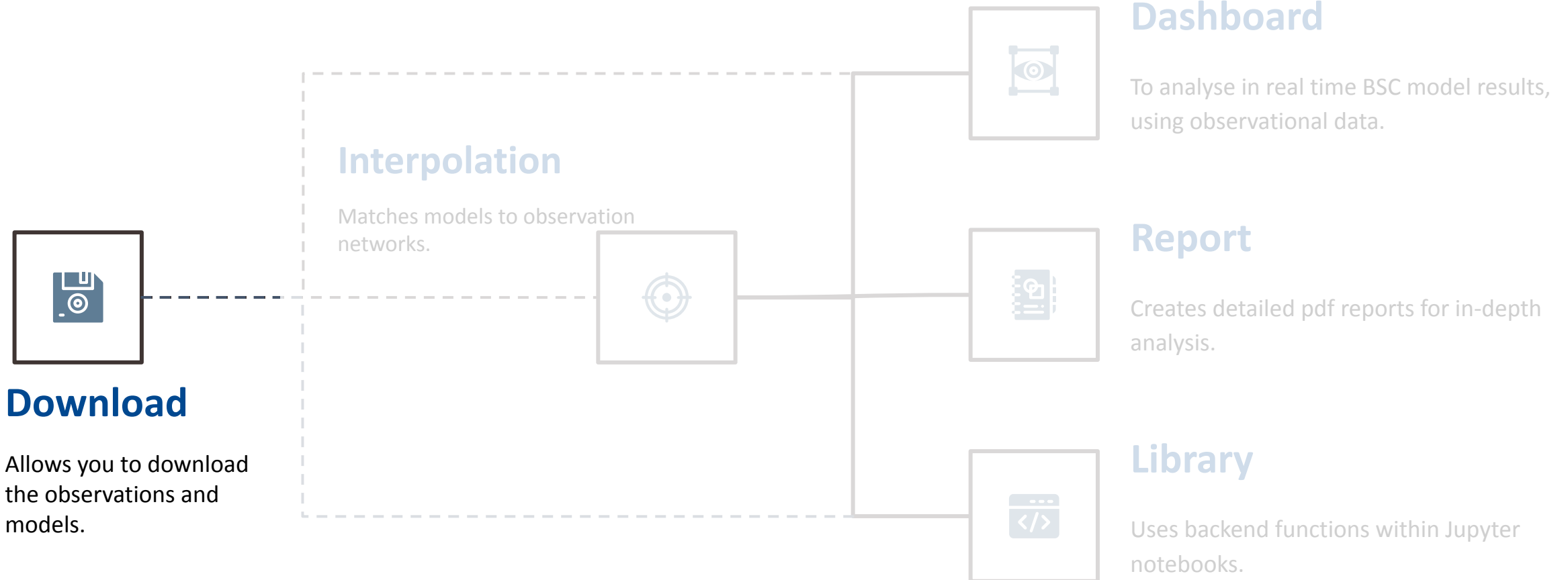
Functionalities



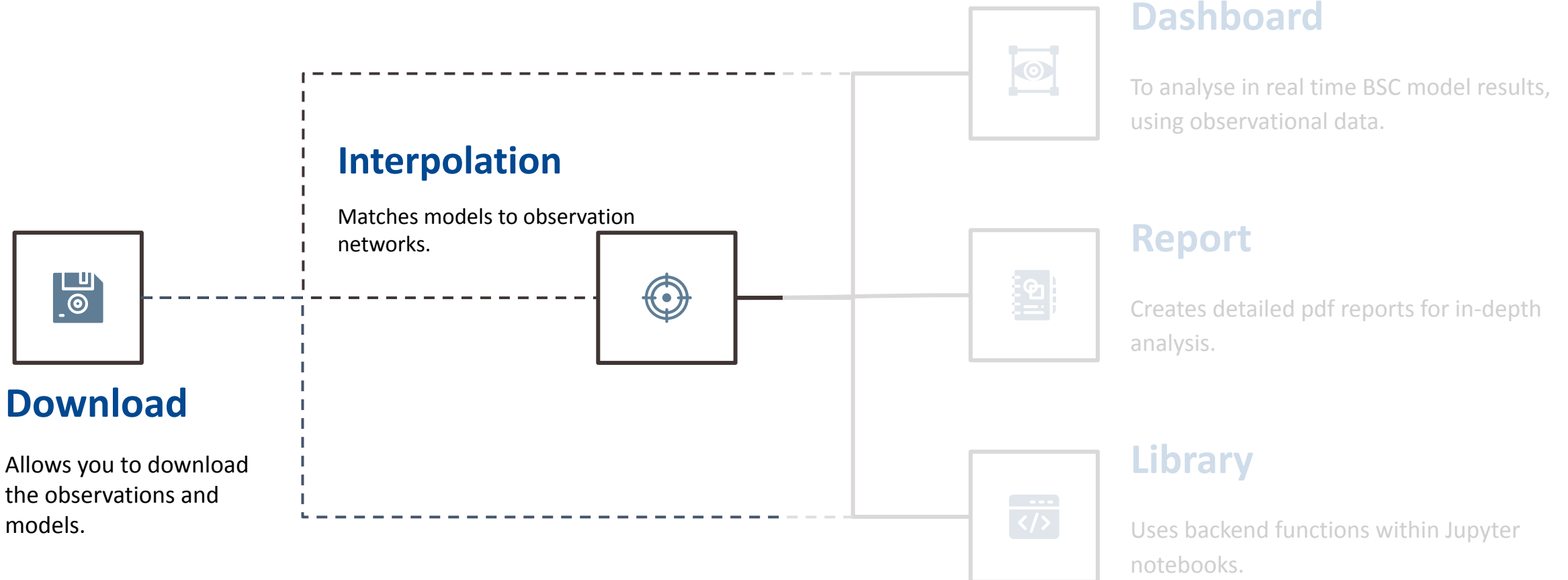
**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

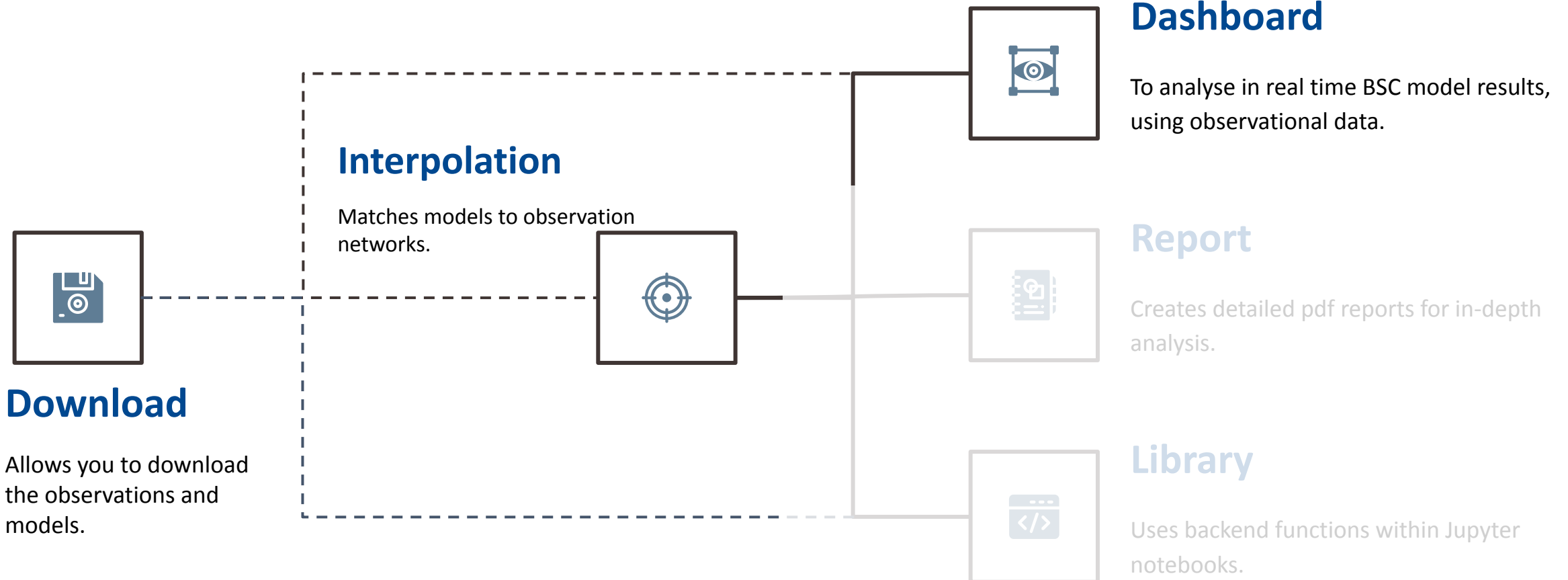
Structure



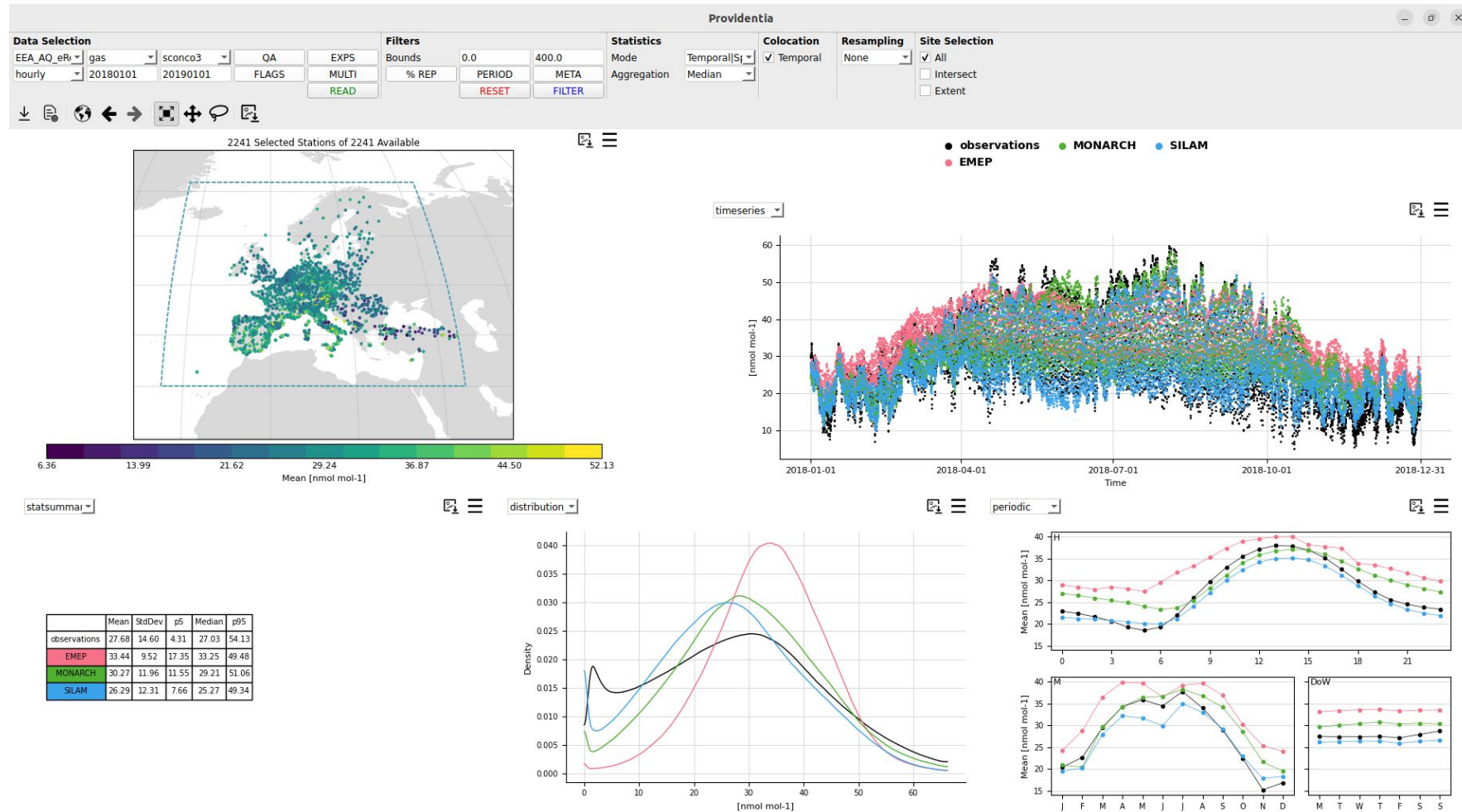
Structure



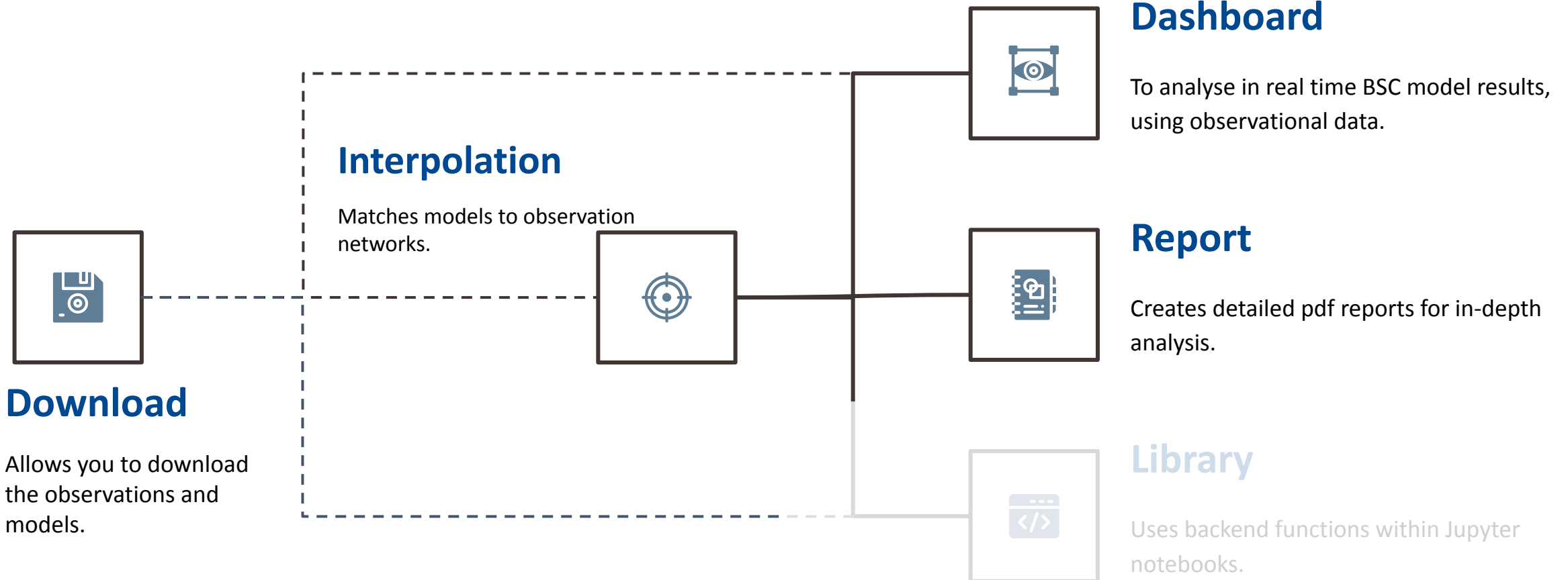
Structure



Dashboard



Structure

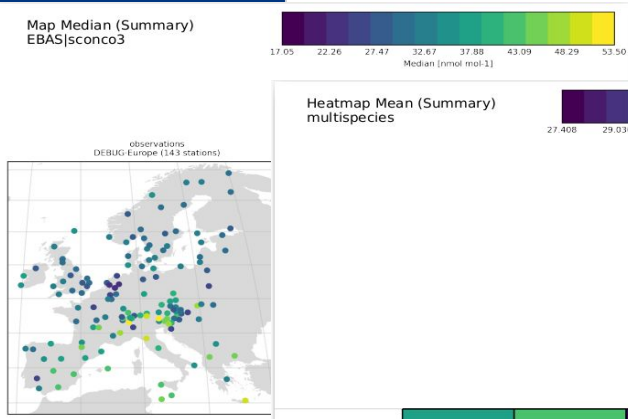


Report

EVALUATION REPORT

Network: EBAS
 Species: O3
 Temporal resolution: Hourly
 Date range: 01/01/2018 - 01/01/2019
 Models: EMEP, MONARCH
 Subsections: DEBUG-Europe, DEBUG-Spain

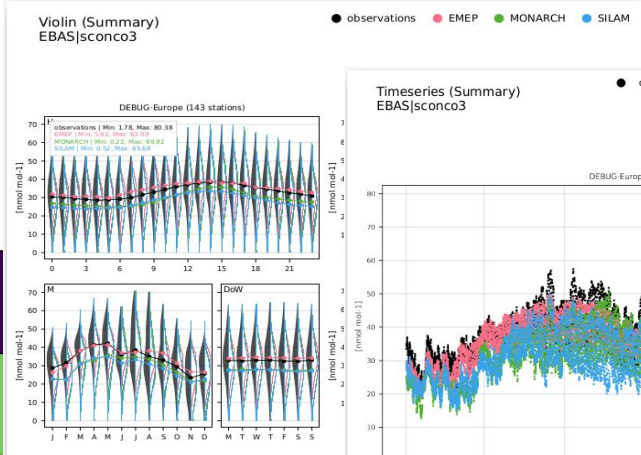
Map Median (Summary)
EBAS|sconco3



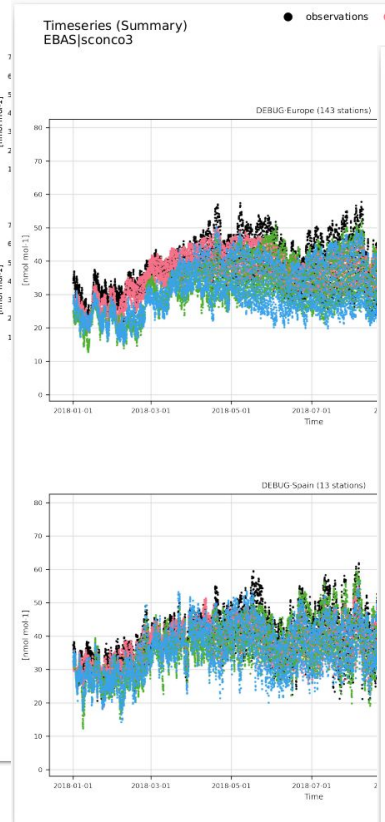
Heatmap Mean (Summary)
multispecies



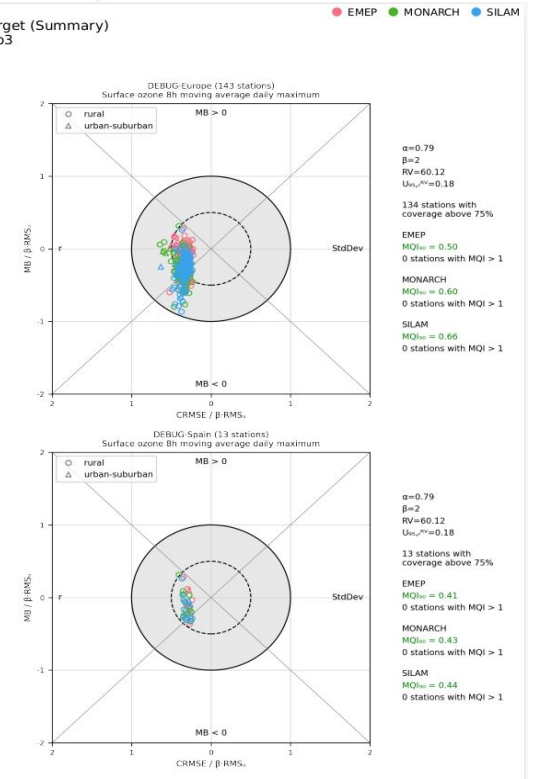
Violin (Summary)
EBAS|sconco3



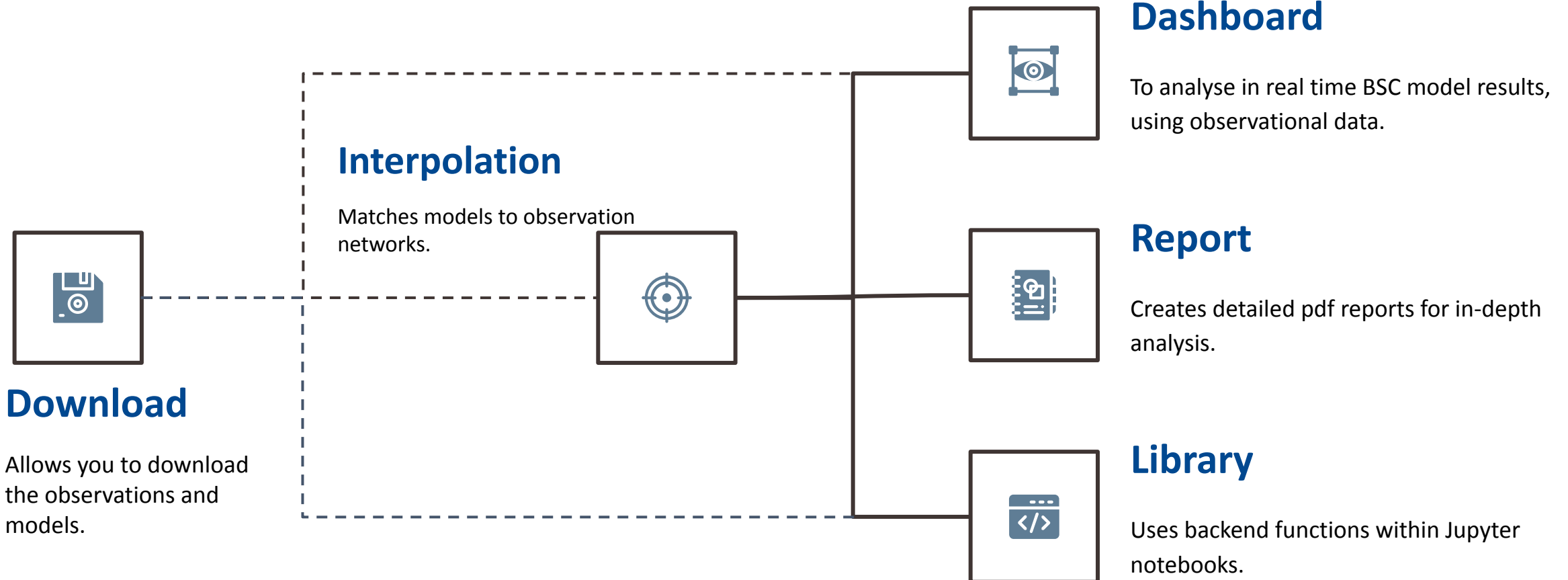
Timeseries (Summary)
EBAS|sconco3



Fairmode target (Summary)
EBAS|sconco3



Structure



Library

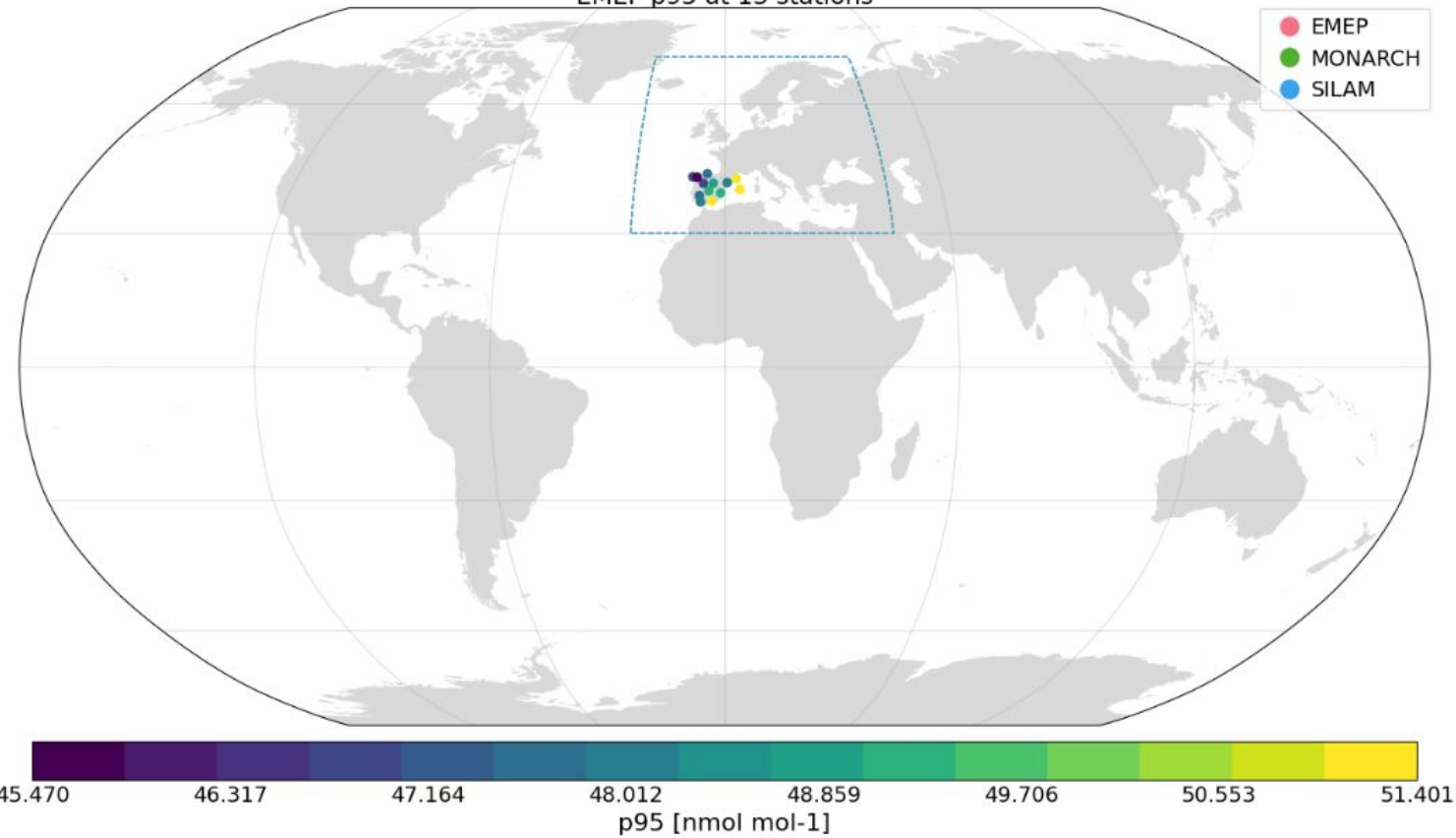
```
import providentia as prv
```

```
provi = prv.load("interactive_template.conf")
```

```
# make a map plot (p95 statistic)  
provi.make_plot('map-p95', labela='EMEP', map_extent=[-180, 180, -90, 90], plot_options=['domain'])
```

Warning: Width and/or height have not been passed. The default values will be set.
Warning: More than 1 network or species defined, can only plot for 1 pair. Taking EBAS|sconco3.

EMEP p95 at 13 stations



Set up



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Prerequisites

Providentia is stable on macOS and Linux, where you need to install [Git](#) and [Conda](#).

The software has not been designed to work on Windows. If you are a Windows user you must first choose one of these options:

- Use Windows Subsystem for Linux (WSL), only for Windows Pro users (**Recommended**). Instructions [here](#).
- Use a virtual machine (e.g. Oracle, VirtualBox). Instructions [here](#).
- Use Git Bash. Instructions [here](#).

BSC HPC users:

- Providentia is available on **MN5** and **Nord4**
- No previous installation is required

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/Getting-started.html#prerequisites>

Connect to HPC machines (BSC HPC users)

To run Providentia on **Nord4** or **MN5**, you need to configure SSH access from your local machine.
On your local computer, open and edit your SSH configuration file:

```
vi ~/.ssh/config
```

Host **mn5**

```
HostName glogin4.bsc.es  
User bsc032XXX  
IdentityFile ~/.ssh/id_rsa  
ForwardX11 yes  
ForwardX11Trusted yes  
ForwardX11Timeout 7d  
Compression yes  
Ciphers aes128-gcm@openssh.com
```

Host **nord4**

```
HostName n4login0.bsc.es  
User bsc032XXX  
IdentityFile ~/.ssh/id_rsa  
ForwardX11 yes  
ForwardX11Trusted yes  
ForwardX11Timeout 7d  
Compression yes  
Ciphers aes128-gcm@openssh.com
```

Connect to HPC machines (BSC HPC users)

To run Providentia on **Nord4** or **MN5**, you need to configure SSH access from your local machine.
On your local computer, open and edit your SSH configuration file:

```
vi ~/.ssh/config
```

Host mn5	Host nord4
HostName glogin4.bsc.es	HostName n4login0.bsc.es
User bsc032XXX	User bsc032XXX
IdentityFile ~/.ssh/id_rsa	IdentityFile ~/.ssh/id_rsa
ForwardX11 yes ForwardX11Trusted yes ForwardX11Timeout 7d	ForwardX11 yes ForwardX11Trusted yes ForwardX11Timeout 7d
Compression yes Ciphers aes128-gcm@openssh.com	Compression yes Ciphers aes128-gcm@openssh.com

Only glogin4 has internet

Recommended for password-free SSH access. A tutorial is available [here](#)

Only n4login0 has internet

Mandatory to run the Dashboard

Necessary to make the Dashboard faster

Clone Providentia

Enter the project's GitHub page (<https://github.com/bsc-es/providentia>) and clone Providentia in your working directory:

```
git clone https://github.com/bsc-es/providentia
```

The latest releases are saved in the production branch, but we suggest users to stay in the **master** branch. In order to stay up-to-date just run this line from time to time:

```
git pull
```

BSC HPC users

- If you are working on an HPC machine, we recommend cloning the directory in **gpfs/** (e.g. **/gpfs/scratch/bsc32/bsc032XXX**) from **MN5 (glogin4)** to have access from **MN5** and **Nord4**.

Internal directories

When cloning the Providentia repository from GitHub, the project is automatically created with a predefined directory structure. These internal directories are required for the correct execution of Providentia:

- |— `configurations/` Contains configuration files required to run all Providentia modes.
- |— `logs/` Stores output files generated by Interpolation.
- |— `notebooks/` Holds template notebooks.
- |— `plots/` Saved plots if Providentia is used as a module.
- |— `reports/` Saved reports generated in Report.
- |— `saved_data/` Stores configuration, NetCDF and NumPy files if Providentia is used as a module.
- |— `settings/` Contains files that configure various aspects of Providentia.

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/Getting-started.html#providentia-internal-directories>

Launching Providentia

After cloning the repository, open a terminal and move to the Providentia directory.

```
cd providentia
```

Then start the dashboard by running:

```
./bin/providentia
```

The first time the software runs in a local machine it will create a conda environment called **providentia-env_v[version]** with all the modules needed.

Using a configuration file

If you want to define which data is loaded in advance, you can use a configuration file.

To specify which configuration file to use, you can specify it from the command line using the **--config** argument.

Configuration file inside the configurations/ folder

If the file is located inside the **configurations/** folder, only the filename is required:

```
./bin/providentia --config=example.conf
```

Configuration file in a different location:

If the file is stored in a different location, you can provide the full path:

```
/bin/providentia --config=/path/to/file/example.conf
```

Main launch options

	Main purpose	Argument
Dashboard	On-the fly analysis	
Report	Detailed analysis in PDFs format	--report, --reports
Notebook	Open a Jupyter notebook	--notebook, --nb
Interpolation	Interpolate models to observational networks	--interp, --interpolate, --interpolation
Download	Download data to local machine	--download, --dl
Clean	Clean logs and new files	--clean
Configuration file	Path or name (if in configurations/ directory)	--conf, --config, --configuration
Section	Specify configuration file section	--section
Logfile	Redirects output to a file.	--logfile

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/Command-line-configuration.html>

Configuration files



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Create a configuration file

Configuration files specify which data to download (download mode), interpolate (interpolation mode), read and plot (dashboard, report and library modes) in the different modes.

By default, Providentia reads these files from the **configurations/** directory:

```
> .github
> assets
> bin
▼ configurations
  ⚙ actris_vre_cams.conf
  ⚙ actris_vre.conf
  ⚙ caliope.conf
  ⚙ cams_dl.conf
  ⚙ cams_global_forecast.conf
  ⚙ cams2_40.conf
  ⚙ cams2_82_NRT_DU_SS_maps.conf
  ⚙ cams2_82_NRT_DU_SS_table.conf
  ⚙ cams2_82_NRT_scoreboards_htap.conf
  ⚙ cams2_82_NRT_table_Global.conf
  ⚙ cams2_82_NRT_table_NAMEEE.conf
  ⚙ cams2_82.conf
  ⚙ debug_nonghost.conf
  ⚙ debug.conf
```

Structure of a configuration file

Configuration files are composed of:

- Sections (mandatory) act as the parent container for all parameters.
- Subsections (optional), contain metadata to filter data.

When reading and filtering data, subsections inherit all values defined in their parent section.

configurations/configuration_name.conf

Section ([name])

```
[PRV_sconco3_CHIMERE]
ghost_version = 1.5
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
model = cams61_chimere_ph2-eu-000 (CHIMERE)
```

Additional notes:

- Configuration fields are not order-sensitive.
- When defining a section-subsection pair in the command line, use:

--section=section·subsection

Subsection ([[name]])

```
[[Barcelona]]
latitude = 39.8, 41.8
longitude = 1.5, 2.5
area_classification = keep: rural || remove: urban-suburban
```

All fields in this section are shared parameters across modes.

For a full list of shared parameters: [Shared fields](#)

Main fields in a configuration file

The following fields are commonly defined in configuration files across all the different modes:

	Description
ghost_version	GHOST version used when a GHOST network is selected. (e.g. 1.5)
network, observation, framework	Observational data. (e.g. EBAS)
model, models, experiments, experiment	Model or experiment IDs. Domain and ensemble information can also be included. (e.g. a4j1-regional-000)
species	Atmospheric components to analyse in BSC nomenclature. (e.g. sconco3)
resolution	Temporal resolution of the observations (e.g. hourly)
start_date	Start date. In YYYYMMDD format or YYYYMM for interpolation. (e.g. 20210101)
end_date	End date. In YYYYMMDD format or YYYYMM for interpolation. (e.g. 20220101)

Species

- Species refer to atmospheric components such as gases and aerosols.
- Providentia uses the BSC nomenclature defined in [GHOST standards](#).

Chemical formula	Description	BSC parameter name
O ₃	Surface concentration of ozone	sconco3
NO	Surface concentration of nitrogen monoxide	sconcno
NO ₂	Surface concentration of nitrogen dioxide	sconcno2
SO ₂	Surface concentration of sulphur dioxide	sconcso2
CO	Surface concentration of carbon monoxide	sconcco
CH ₄	Surface concentration of methane	sconcch4

Model, ensemble and domain

In Providentia, models are defined using three key components:

- **Model:** Model ID (or experiment ID in MONARCH terminology).
Examples: cams61_chimere_ph2, a8g1, osuite.
- **Domain:** Geographical region where the model is run.
Examples: global, eu, cat.
- **Ensemble:** Simulation of the same model with different initial conditions or configurations to estimate uncertainty.
Examples: 000, 001, 002.

You can specify **model, domain, and ensemble** either:

- All in one line.
- Separately in different fields.
- A combination of both.

For more details, check the models section in the official documentation: [Models configuration](#)

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
```

```
model = cams61_chimere_ph2-eu-000 (CHIMERE)
```

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
```

```
model = cams61_chimere_ph2 (CHIMERE)
domain = eu
ensemble = 000
```

Structure of a configuration file (Analysis and visualisation modes)

- Apart from the main shared parameters, each mode has specific configuration fields to control its behavior.
- Analysis and Visualisation modes (Dashboard, Report and Library) share many parameters.
 - Full list of shared fields: [Dashboard, Report and Library Parameters](#)
- Subsections in configuration files affect only visualisation modes because they carry all relevant metadata.
 - Full list of subsection fields: [Metadata Fields](#)

Section ([[name]])

Subsection ([[name]])

configurations/dashboard_configuration.conf

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
model = cams61_chimere_ph2-eu-000 (CHIMERE)
statistic_mode = Flattened
active_dashboard_plots = timeseries, periodic-violin, scatter, distribution
```

```
[[Barcelona]]
latitude = 39.8, 41.8
longitude = 1.5, 2.5
area_classification = keep: rural || remove: urban-suburban
```

Full list of Dashboard-specific parameters: [here](#)

configurations/report_configuration.conf

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
model = cams61_chimere_ph2-eu-000 (CHIMERE)
statistic_mode = Flattened
report_filename = PROVIDENTIA_Report
```

```
[[Barcelona]]
latitude = 39.8, 41.8
longitude = 1.5, 2.5
area_classification = keep: rural || remove: urban-suburban
```

Full list of Report-specific parameters: [here](#)

Structure of a configuration file (Download and interpolation modes)

- Subsections do not work in Download or Interpolation modes.
- Just like in the Analysis and Visualisation modes. They also have specific configuration fields to control its behavior.

Section ([name])

configurations/download_configuration.conf

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
model = cams61_chimere_ph2-eu-000 (CHIMERE)
dl_overwrite = True
```

Full list of Download-specific parameters: [here](#)

configurations/interpolation_configuration.conf

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
model = cams61_chimere_ph2-eu-000 (CHIMERE)
interp_n_neighbours = 3
```

Full list of Interpolation-specific parameters: [here](#)

Multiple sections in a configuration file

When launching the tool multiple sections, the modes behave differently:

- **Dashboard:** a specific section-subsection pair must be selected, it can be specified with the **--section** argument.
- **Library:** The first section in the configuration file is used by default, unless another section is explicitly specified.
- **Report, Download and Interpolation:** All sections (and subsections) in the configuration file are processed.

If only one section needs to be used, it can be specified with the **--section** argument.

Section1 ([name])

Section1-Subsection1 ([[name]])

Section1-Subsection2 ([[name]])

Section2 ([name])

Section2-Subsection1 ([[name]])

Section3 ([name])

configurations/multiple_configuration_name.conf

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180601
models = cams61_chimere_ph2-eu-000 (CHIMERE)

[[Spain]]
latitude = 35, 44
longitude = -9, 2
area_classification = keep: rural || remove: urban-suburban

[[British_Isles]]
latitude = 50, 58
longitude = -12, 2

[PRV_sconco3_EMEP]
network = EBAS
species = sconco2
resolution = hourly
start_date = 20180101
end_date = 20190101
model = cams61_emep_ph2-eu-000 (EMEP)

[[Spain]]
country = keep: Spain ||

[PRV_sconco3_fc]
network = eea/eionet
species = sconco3
resolution = hourly
start_date = 20230101
end_date = 20230201
model = a59g-regional-000 (fc)
```

Data



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Access to data

Providentia currently provides direct access to both observational and model data via a dedicated download mode (more soon).

Observational data

- GHOST (Zenodo and **BSC**)
- non-GHOST (**BSC**)
- ACTRIS (via NILU THREDDS)

Model data

- CAMS regional forecast, analysis, and reanalysis
- CAMS global forecast, and reanalysis

Data formats

Providentia requires observational and model data to be of a defined file format to be able to be read.

Providentia's in-built download mode automatically formats data to be the correct format (more soon), but in case someone wants to read their own data with Providentia, we have substantive guides which aid this process.

For a guide on how to format observational data files to be read by Providentia, see [here](#).

For a guide on how to format model data files to be read by Providentia, see [here](#).

Setting data paths

Providentia knows where data lives via one key file: **settings/data_paths.yaml**

There are multiple headings in the file which set **observational** and **model** data paths on different machines, e.g. **local** and BSC machines (**mn5**, **nord4**, etc.).

Under each heading, the following paths are set:

- **ghost_root** : Observational GHOST data
- **nonghost_root**: Observational non-GHOST data
- **mod_root**: Interpolated model data
- **mod_to_interp_root**: Non-interpolated model data

If you are using Linux inside Windows (through WSL, VirtualBox or Git Bash) you will need to modify the paths to point to your Windows paths.

```
local:  
  ghost_root: ~/data/providentia/obs/ghost  
  mod_root: ~/data/providentia/mod  
  mod_to_interp_root: ~/data/providentia/mod_to_interp  
  nonghost_root: ~/data/providentia/obs/nonghost
```

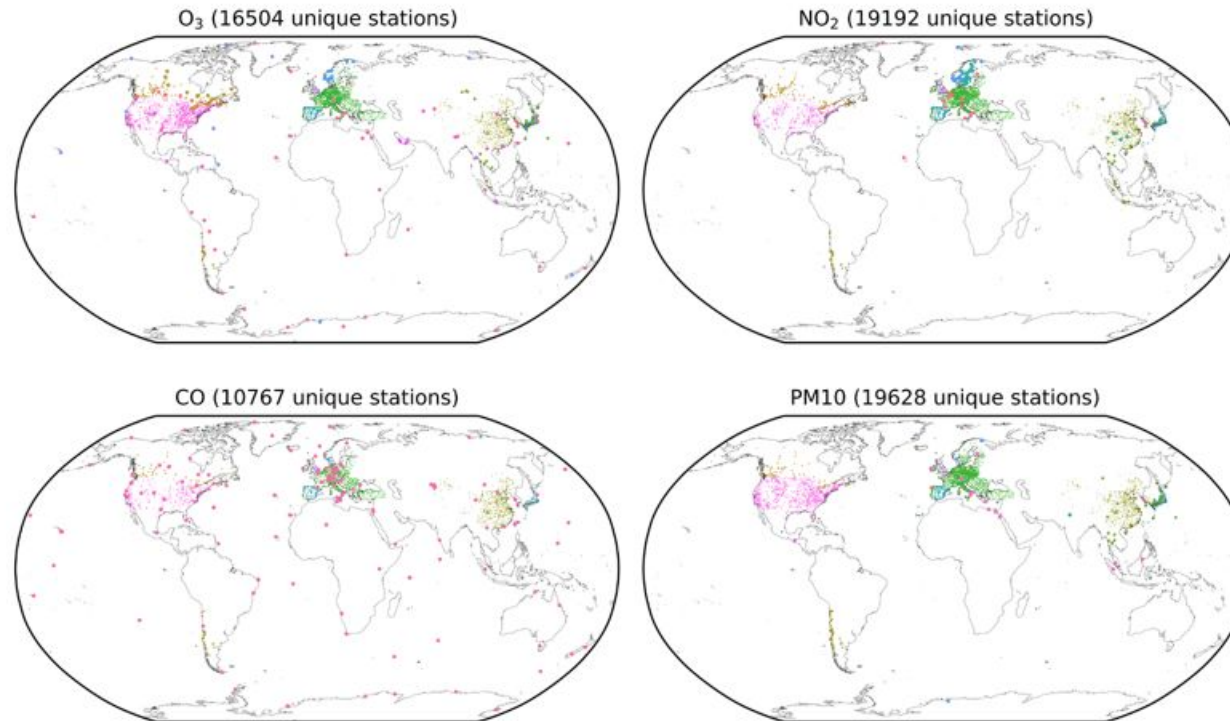
GHOST: Globally Harmonised Observations in Space & Time

N° Processed
Measurements:
~7.3 billion

N° Components:
227

N° Networks:
38

Temporal range:
1970-2023



- | | | | |
|----------------------|------------------------|------------------------|----------------|
| ● ACTRIS | ● AERONET v3 Level 1.5 | ● AERONET v3 Level 2.0 | ● AMAP |
| ● BJMEMC | ● CAMP | ● Canada NAPS | ● CAPMoN |
| ● Chile SINCA | ● CNEMC | ● COLOSSAL | ● EANET |
| ● EEA AirBase | ● EEA AQ e-Reporting | ● EMEP | ● EUCAARI |
| ● EUSAAR | ● HELCOM | ● HTAP | ● IMPACTS |
| ● Independent (EBAS) | ● Japan NIES | ● Mexico CDMX | ● MITECO |
| ● NADP AMNet | ● NADP AMoN | ● NILU | ● NOAA-ESRL |
| ● NOAA-GGGRN | ● OECD | ● UK AIR | ● UK DECC |
| ● US EPA AirNow DOS | ● US EPA AQ5 | ● US EPA CASTNET | ● WMO GAW WDCA |
| ● WMO GAW WDCGG | ● WMO GAW WDCRG | | |

Why GHOST?

1. Fully documented methods

- Paper published in **Earth System Science Data (ESSD):**
<https://doi.org/10.5194/essd-16-4417-2024>
- This outlines all steps taken in harmonising observations.
- The code used to process GHOST is also publicly available:
<https://zenodo.org/records/13859075>



Why GHOST?

2. Data standardisation

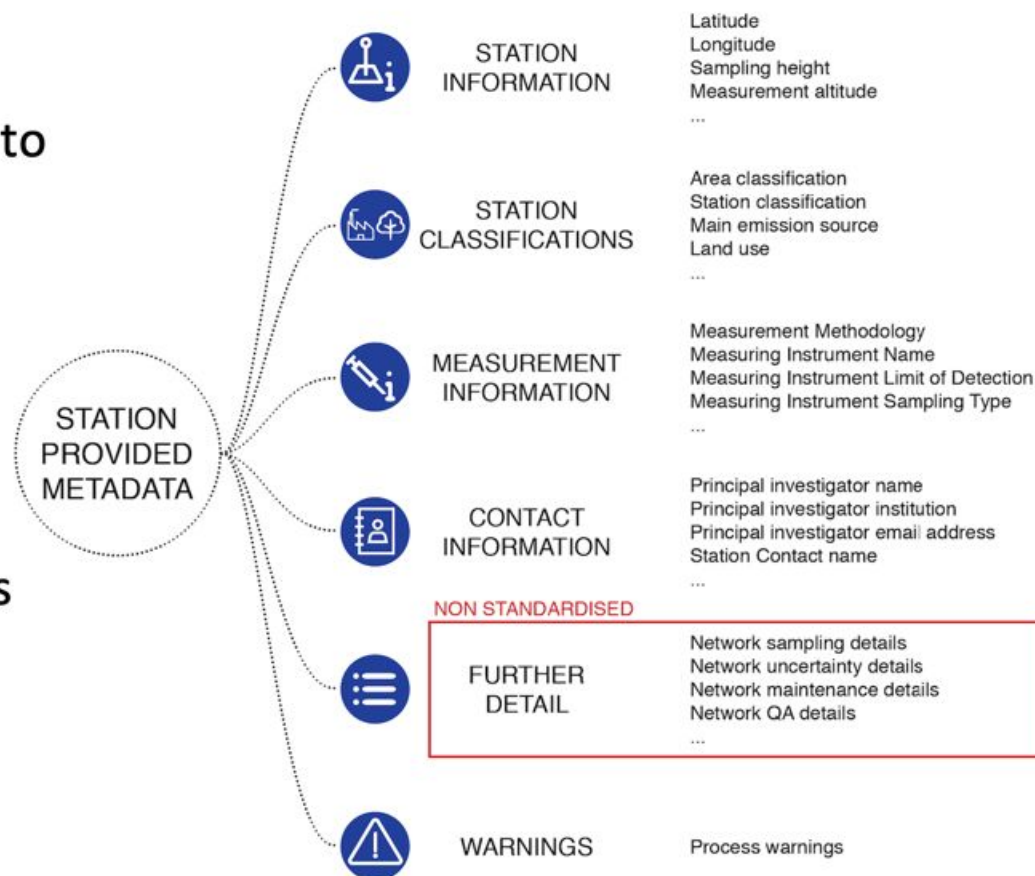
- All components associated with standard units, e.g. nmol mol^{-1} for ozone.
- Where applicable, components are standardised to common temperature and pressure (European standards).
- **No screening** is performed on the data. Rather, data is associated with flags (more shortly).

GHOST component name	Chemical formula	Long component name	Standard unit
gas			
sconco3	O ₃	Ozone	nmol mol^{-1}
sconco	NO	Nitrogen monoxide	nmol mol^{-1}
sconco2	NO ₂	Nitrogen dioxide	nmol mol^{-1}
sconco2	SO ₂	Sulphur dioxide	nmol mol^{-1}
sconco	CO	Carbon monoxide	nmol mol^{-1}
sconch4	CH ₄	Methane	nmol mol^{-1}
sconcc2h4	C ₂ H ₄	Ethene	pmol mol^{-1}
sconcc2h6	C ₂ H ₆	Ethane	pmol mol^{-1}
sconcc3h6	C ₃ H ₆	Propene	pmol mol^{-1}
sconcc3h8	C ₃ H ₈	Propane	pmol mol^{-1}
sconcisop	C ₅ H ₈	Isoprene	pmol mol^{-1}
sconcc6h6	C ₆ H ₆	Benzene	pmol mol^{-1}
sconcc7h8	C ₇ H ₈	Toluene	pmol mol^{-1}
sconcc10h16	C ₁₀ H ₁₆	Monoterpenes	pmol mol^{-1}
sconcnmvoc	–	Total non-methane volatile organic compounds	nmol mol^{-1}
sconcvoc	–	Total volatile organic compounds	nmol mol^{-1}
sconnmhc	–	Total non-methane hydrocarbons	nmol mol^{-1}
sconchc	–	Total hydrocarbons	nmol mol^{-1}
sconcnh3	NH ₃	Ammonia	nmol mol^{-1}
sconchno3	HNO ₃	Nitric acid	nmol mol^{-1}
sconcpn	C ₂ H ₃ NO ₅	Peroxyacetyl nitrate	nmol mol^{-1}
sconchcho	CH ₂ O	Formaldehyde	nmol mol^{-1}
sconchcl	HCl	Hydrochloric acid	nmol mol^{-1}
sconchf	HF	Hydrofluoric acid	nmol mol^{-1}
sconch2s	H ₂ S	Hydrogen sulfide	nmol mol^{-1}

Why GHOST?

3. Metadata standardisation

- Network metadata standardised to 100+ variables.
- Metadata is allowed to change through the time record.
- Extensive efforts made to standardise station classifications and measurement process information.



Why GHOST?

3. Metadata standardisation

- We have created a measurement process library, storing unique measurement methods (~100) and any instrument names associated with each method (~500 in total).
- Moreover, instrument specific information, e.g. flow rate, limits of detection etc. are also in the library. These obtained by a large manual effort, parsing through instrument manuals and specification sheets.

Teledyne API Model 200A NO_x Analyzer Instruction Manual, 02246, Rev. G, DCN 5247

3 SPECIFICATIONS, AGENCY APPROVALS, WARRANTY

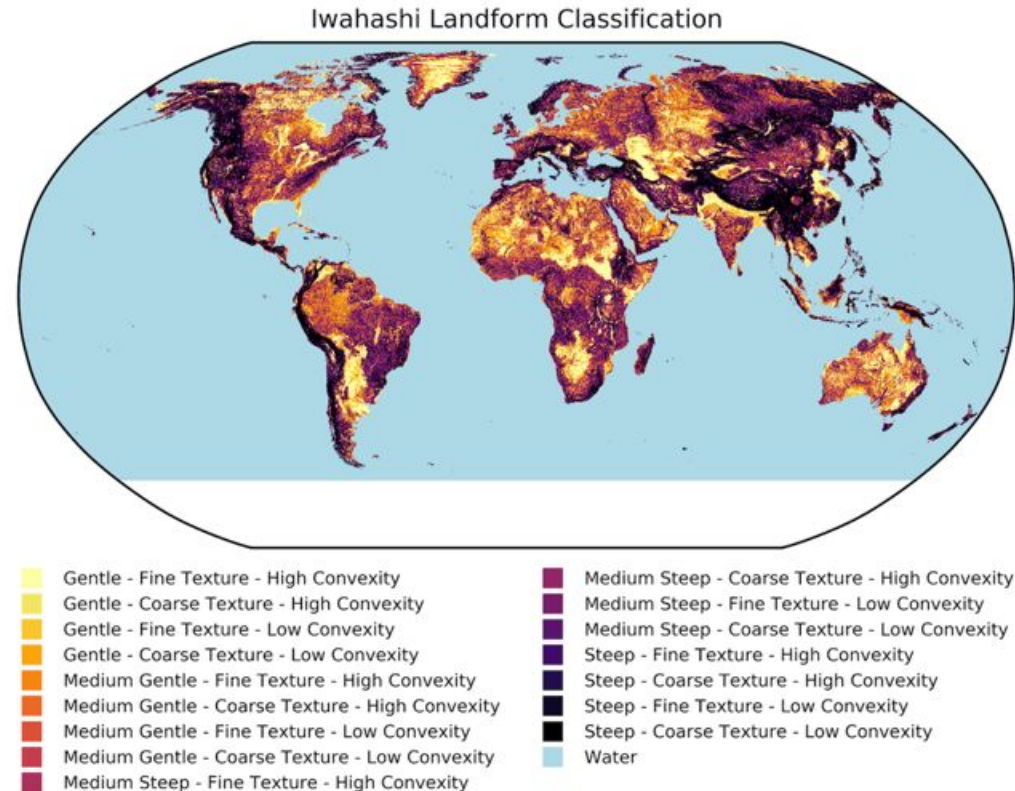
3.1 Specifications

Ranges	In 1ppb increments from 50ppb to 20,000ppb
Range Modes	Single, Independent, AutoRange
Noise at Zero ¹	0.2 ppb RMS
Noise at Span ¹	<0.5% of reading above 50 ppb
Lower Detectable Limit ²	0.4 ppb RMS
Zero Drift ³	<0.5 ppb/24 hours
Zero Drift	1 ppb/7 days
Span Drift	<0.5% FS/7 days
Lag Time	20 sec
Rise Time	95% in <60 sec
Fall Time	95% in <60 sec
Sample Flow Rate	500 cc/min. ± 10%
Linearity	1% of full scale
Precision	0.5% of reading
Temperature Range	5-40°C
Temp Coefficient	< 0.1% per °C
Humidity	0-95% RH non-condensing
Voltage Coefficient	< 0.1% per V
Dimensions HxWxD	7" x 17" x 23.6" (18cm x 43cm x 61cm)
Weight, Analyzer	43 lbs (20 kg) w/external pump
Weight, Analyzer	55 lbs (25 kg) w/internal pump
Weight, Ext Pump Pack	16 lbs (7 kg)
Weight, Internal Pump	5 lbs (2 kg)
Power, Analyzer	100 V ~ 50/60 Hz, 120V~ 60 Hz, 220 V~ 50 Hz, 240 V~ 50 Hz, 125 watts
Power, Analyzer ⁵	230 V ~ 50 Hz, 2.5A peak
Power, Ext Pump	110 V ~ 60 Hz, 220V~ 50 Hz, 240V~ 50Hz, 295 watts
Power, Ext Pump ⁷	230 V ~ 50 Hz, 2.5A peak
Power, Int Pump	110 v/50/60 Hz, 60 watts
Environmental	Installation Category (Over-voltage Category) II Pollution Degree 2
Recorder Output ⁴	0-100 mV, 0-1, 5, 10v
Analog Resolution	1 part in 1024 of selected voltage or current range
Status Option	12 Status Outputs from opto-isolator
Measurement Units	ppb, ppm, ug/m ³ , mg/m ³

Why GHOST?

4. Gridded metadata

- ~30 of the most popular gridded metadata variables used by the atmospheric science community are associated per station (updated every 5 years).
- Examples include products used for determining if a station can be vertically resolved by a model, e.g. the Iwahashi landform classification, informing on the local orography.



Why GHOST?

5. Network QA

- Many networks provide QA flags, informing of various potential issues with data.
- The codes/names associated with these flags are invariably different across networks, but there is a large crossover in the physical meanings associated with these flags.
- GHOST standardises these to 186 standard flags, and to 6 simplified flags.

Flag Code	Flag Name	Flag Code	Flag Name	Flag Code	Flag Name	Flag Code	Flag Name
Basic Flags							
0	Valid Data	1	Preliminary Data	2	Missing Data	3	Invalid Data – Unspecified
4	Un-Flagged Data						
Estimated Flags							
10	Estimated Data – Unspecified	11	Estimated Data – Measured Negative Value	12	Estimated Data – No Value Detected	13	Estimated Data – Value Below Detection Limit
14	Estimated Data – Value Above Detection Limit	15	Estimated Data – Value Substituted from Secondary Monitor	16	Estimated Data – Multiple Parameters Aggregated		
Extreme / Irregular Flags							
20	Extreme / Irregular Data – Unspecified	21	Data Does Not Meet Internal Network Quality Control Criteria	22	High Variability of Data	23	Irregular Data Manually Screened and Accepted
24	Irregular Data Manually Screened and Rejected	25	Negative Value	26	No Value Detected	27	Reconstructed / Recalculated Data
28	Value Close to Detection Limit	29	Value Below Acceptable Range	30	Value Above Acceptable Range	31	Value Below Detection Limit
32	Value Above Detection Limit						
Measurement Issue Flags							
40	Measurement Issue – Unspecified	41	Chemical Issue	42	Erroneous Sampling Operation	43	Extreme Internal Instrument Meteorological Conditions

Some of the standardised network QA flags.

Why GHOST?

6. GHOST QA

- 79 GHOST QA checks.
- These checks can be simplistic, e.g. for missing, negative or infinite data
- Or more advanced, e.g. assessing the consistency of the distribution of measurements in each month with that of surrounding years, and across the time record.

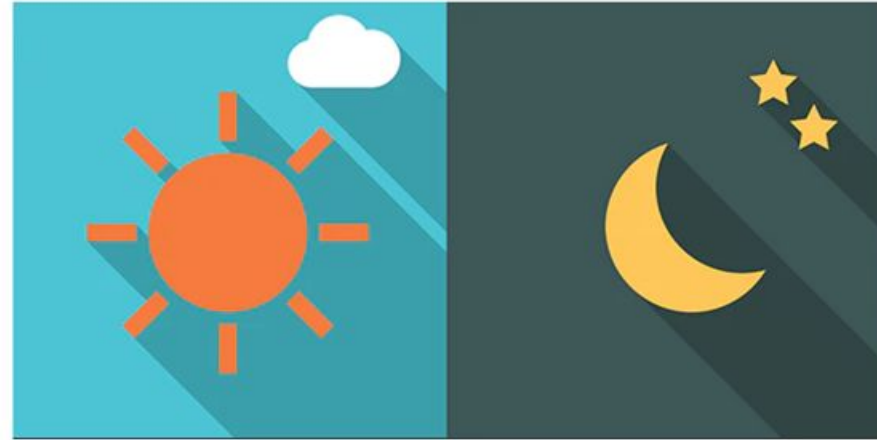
QA Flag	QA Name	Description
Basic Flags		
0	Missing Measurement	Measurement is missing (i.e. NaN), or have network QA flag stating missing measurement.
1	Infinite Value	Measurement is infinite. This happens when values are outside of the range that the float32 data type can handle (-3.4E+38 to +3.4E+38).
2	Negative Measurement	Measurement is negative (i.e. < 0.0), or have network QA flag stating negative measurement.
3	Zero Measurement	Measurement is zero, or have network QA flag stating no value detected.
4	Not Maximum Data Quality Level	Measurement is not of the highest data quality level available from data provider.
5	Preliminary Data	Measurement which is flagged in the network QA as preliminary.
6	Invalid Data Provider Flags – GHOST Decreed	Measurement is associated with network QA flag/s which have been decreed by the GHOST project architects to suggest the measurements are associated with substantial uncertainty / bias.
7	Invalid Data Provider Flags – Network Decreed	Measurement is associated with network QA flag/s which have been decreed by the reporting network to suggest the measurements are associated with substantial uncertainty / bias.
8	No Valid Data to Average	After screening by GHOST QA, no valid data remains to perform temporal average.
Measurement Process Flags		
10	Methodology Not Mapped	The reported measurement methodology has not been able to be mapped to a standard methodology name.
11	Assumed Primary Sampling	A level of assumption has been made in determining the primary sampling type.
12	Assumed Sample Preparation	A level of assumption has been made in determining the sample preparation.
13	Assumed Measurement Methodology	A level of assumption has been made in determining the measurement methodology.
14	Unknown Primary Sampling Type	The specific name of the primary sampling type is unknown.
15	Unknown Primary Sampling Instrument	The specific name of the primary sampling instrument is unknown.

Some of the standard GHOST QA flags.

Why GHOST?

7. Temporal Codes

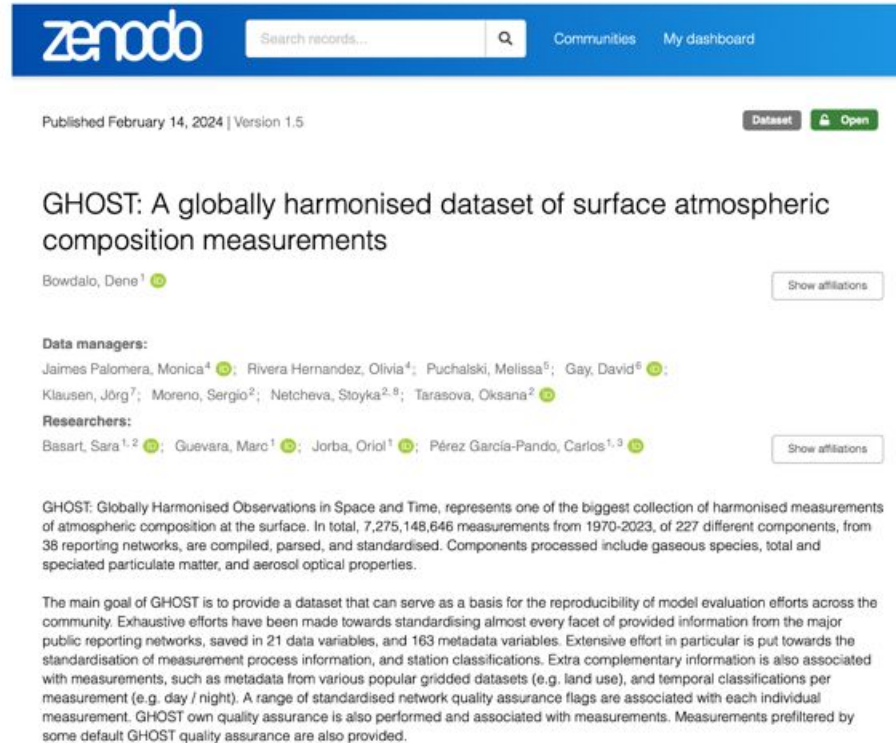
- Temporal classification codes are added per timestep.
- These are specifically: **seasons, day/night, weekday/weekend.**



Why GHOST?

8. Available data

- Data made available via DOI as **YYYYMM** netCDFs, per component, in 3 different temporal resolutions: **hourly, daily, monthly.**
- Data available in 2 forms: **separated per network** and **merged across networks:**
<https://doi.org/10.5281/zenodo.10637449>



The screenshot shows the Zenodo dataset page for "GHOST: A globally harmonised dataset of surface atmospheric composition measurements". The page includes a search bar, navigation links for "Communities" and "My dashboard", and a "Dataset" button. The dataset is published on February 14, 2024, and is version 1.5. The author is listed as Bowdalo, Dene. The data managers are Jaimes Palomera, Monica; Rivera Hernandez, Olivia; Puchalski, Melissa; Gay, David; Klausen, Jörg; Moreno, Sergio; Natcheva, Stoyka; Tarasova, Oksana. The researchers are Basart, Sara; Guevara, Marc; Jorba, Oriol; Pérez García-Pando, Carlos. The description states that GHOST represents one of the biggest collections of harmonised measurements of atmospheric composition at the surface, with 7,275,148,646 measurements from 1970-2023 across 227 components and 38 reporting networks. The main goal is to provide a dataset for model evaluation efforts across the community, with extensive efforts towards standardising information from public reporting networks, 21 data variables, and 163 metadata variables.

non-GHOST

Providentia can read data non-GHOST observational data, the only limitation is the data will be have limited filtering features, specifically:

- No GHOST QA flags
- No network flags
- No native representativity filters
- No period filters
- Limited metadata

On the dashboard this is reflected via greyed out buttons:

Data Selection					Filters		
eea/eionet ▾	gas ▾	sconco3 ▾	QA	MODS	Bounds	0.0	400.0
hourly ▾	20180101	20190101	FLAGS	MULTI	% REP	PERIOD	META
				READ		RESET	FILTER

Download



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Download data

If Providentia is cloned on a local machine, no data is available initially. Data must be downloaded before running any analysis or visualisation.

To download data, add the **--download** or **--dl** option together with a configuration file:

```
./bin/providentia --dl --config=example.conf
```

By default, all sections in the configuration file are processed. If only one section is required, it can be specified with the **--section** argument:

```
./bin/providentia --dl --config=example.conf --section=section1
```

Downloaded data location

- The downloaded data is stored in the directory defined in **settings/data_paths.yaml** under the **local** key.
- Users can modify these paths. The data will be downloaded, interpolated and read from the locations specified by the user.

Path identifiers used in data_paths.yaml

- Observational GHOST data → **ghost_root**
- Observational non-GHOST data → **nonghost_root**
- Non-interpolated model data → **mod_to_interp_root**
- Interpolated model data → **mod_root**

```
local:  
  ghost_root: ~/data/providentia/obs/ghost  
  mod_root: ~/data/providentia/mod  
  mod_to_interp_root: ~/data/providentia/mod_to_interp  
  nonghost_root: ~/data/providentia/obs/nonghost
```

Types of downloads

Providentia supports several types of download:

- [Observational and model data from the Barcelona Supercomputing Center](#) (BSC HPC users)
- [GHOST observational data from Zenodo](#)
- [ACTRIS observational data from NILU Thredds servers](#)
- [CAMS non-interpolated model data from the Atmosphere Data Store](#)

Also a copy mode:

- [Transfer from esarchive/ to gpfs/](#) (BSC users)

BSC HPC Download



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Download from BSC HPC machines (BSC HPC users)

Downloads **GHOST** and **non-GHOST** observations and **interpolated** and **non-interpolated** model output from BSC HPC machines.

A BSC account is required to use this feature.

Downloads from HPC to local machines are done via the **Transfer5** machine (or **MN5** if **Transfer5** is unavailable).

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/BSC-download.html>

Download source paths (BSC HPC users)

- Data is retrieved from the paths specified in the **storage5** or **mn5** key of **settings/data_paths.yaml**.
- Users can modify these paths in the **storage5** or **mn5** key of **settings/data_paths.yaml** so that it points to the other location.

```
mn5:  
  ghost_root: /gpfs/projects/bsc32/AC_cache/obs/ghost  
  mod_root: /gpfs/projects/bsc32/AC_cache/recon/exp_interp  
  mod_to_interp_root: /gpfs/projects/bsc32/AC_cache/exp  
  nonghost_root: /gpfs/projects/bsc32/AC_cache/obs/nonghost  
  
storage5:  
  ghost_root: /gpfs/projects/bsc32/AC_cache/obs/ghost  
  mod_root: /gpfs/projects/bsc32/AC_cache/recon/exp_interp  
  mod_to_interp_root: /gpfs/projects/bsc32/AC_cache/exp  
  nonghost_root: /gpfs/archive/bsc32/esarchive/obs
```

SSH credentials for BSC downloads

When running a download from Barcelona Supercomputing Center for the first time, Proventia will ask for your SSH credentials.

First execution

- You will be prompted to enter your machine SSH user:
 - *Insert BSC storage5 ssh user (bsc032XXX):*
- If passwordless SSH is not configured, you will also be asked for your password:
 - *Insert password:*

Credentials storage

Providentia stores SSH connection information (username and password) in a `.env` file so that credentials only need to be entered once.

- The file is generated automatically the first time the download mode is used, if the user chooses to save preferences.
- Once created, the `.env` file can be accessed and modified manually from the Providentia root directory.
- If any field is removed, it will be requested again and regenerated on the next execution if the user chooses to restore it.

```
PRV_USER=bsc032XXX  
PRV_PWD=example_pwd
```

.env file

PRV_USER

This field stores the username used to connect via SSH to remote machines.

Possible Values:

- **Any valid username** – Ensure you have an account and can successfully connect using `ssh {username}@transfer1.bsc.es` in download mode.

PRV_PWD

This field stores the password for logging into the remote machine.

Note that the password is not required if you have configured the passwordless connection to different servers/machines. Tutorial [here](#).

Possible Values:

- **Any valid password**

Download GHOST observational data from BSC HPC machines (BSC HPC users)

GHOST observational networks are datasets that have already been processed according to [GHOST](#) standards.

Data structure

- To be discovered automatically on the machines, the data must follow the structure below:

`{ghost_root}/{network}/{ghost_version}/{resolution}/{species}/{species_YYYYMM.nc}`

Data locations

- Networks are saved under the **ghost_root** paths specified in the **local** key of **settings/data_paths.yaml**.
- Networks are retrieved from the **ghost_root** paths specified in the **storage5** or **mn5** key of **settings/data_paths.yaml**.

By default, for both **MareNostrum5** and **Storage5**, the path is:

`/gpfs/projects/bsc32/AC_cache/obs/ghost`

Download GHOST observational data from BSC HPC machines (BSC HPC users)

Steps to download GHOST observational data:

1. Include the network in your configuration file

You must include at least one network in your configuration file.

If the configuration file contains only observation(s) and no model(s) there is no need to set the download mode, network data will be downloaded automatically.

2. Set the download mode

- Answer “**both**”/”**obs**” to the prompt:
 - *Which type of data do you want to download? Observational, modelled or both?*
- Or set in the configuration file **dl_mode** to **both** or **obs**.

3. Set the observational data source

- Answer “**y**” to the prompt:
 - *Do you want to download observational data from the BSC remote machine? (Otherwise, GHOST observational data will be retrieved from Zenodo)*
- Or set in the configuration file **dl_ghost_source** to **bsc**.

Configuration file example

```
[BSC-GHOST]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180201
dl_ghost_source = bsc
```

Download non-GHOST observational data from BSC HPC machines (BSC HPC users)

Non-GHOST observational networks are datasets that have not been processed.

Data structure

- In order to find the data in the machines it needs to be in this format:

`{nonghost_root}/{institution}/{network}/{resolution}/{species}/{species_YYYYMM.nc}`

Providentia knows if a network is non-GHOST if it has the format `{institution}/{network}`

Examples:

- eea/eionet
- xvpca/xvpca (sometimes the institution and network are the same)

Data locations

- Networks are saved under the **nonghost_root** paths specified in the **local** key of **settings/data_paths.yaml**.
- Networks are retrieved from the **nonghost_root** paths specified in the **storage5** or **mn5** key of **settings/data_paths.yaml**.

By default, for both **MareNostrum5** and **Storage5**, the path is:

`/gpfs/projects/bsc32/AC_cache/obs/ghost`

Download non-GHOST observational data from BSC HPC machines (BSC HPC users)

Steps to Download non-GHOST observational data:

1. Include the network in your configuration file

You must include at least one network in your configuration file.

If the configuration file contains only observation(s) and no model(s) there is no need to set the download mode, network data will be downloaded automatically.

2. Set the download mode

- Answer “**both**”/”**obs**” to the prompt:
 - *Which type of data do you want to download? Observational, modelled or both?*
- Or set **dl_mode** to **both** or **obs**.

Configuration file example

```
[BSC-NONGHOST]
network = nasa-aeronet/oneill_v3-lev15
species = od500aerocoarse
resolution = 3hourly
start_date = 20240901
end_date = 20241001
```

Download non-interpolated model data from BSC HPC machines (BSC HPC users)

Non-interpolated model data refers to model outputs that are ready to be interpolated against a network using interpolation mode.

Most non-interpolated datasets are stored in **esarchive/**, although some may already exist in **gpfs/**.

Data structure

- To be discovered automatically on the machines, the data must follow the structure below:

- Standard non-interpolated models:

{model_id}/{domain}/{resolution}/{species}/{species_YYYYMM[DD].nc}

- Ensemble-stats species non-interpolated models:

**{model_id}/{domain}/{resolution}/ensemble-stats/{species}_{ensemble_stat}_an/
{species_YYYYMM[DD]_{ensemble_stat}_an.nc}**

Download non-interpolated model data from BSC HPC machines (BSC HPC users)

Data locations

- Models are **saved** under the **mod_to_interp_root** path specified in the **local** key in **settings/data_paths.yaml**.
- Model data is **retrieved** from two possible locations in the following order:
 1. **interp_models.yaml**
First, Providentia attempts to retrieve models from the paths specified in **settings/interp_models.yaml**.
 2. **data_paths.yaml**
If the model is not found in **settings/interp_models.yaml** or in any of the paths defined there, the system retrieves it from the **mod_to_interp_root** path specified under the **storage5** or **mn5** keys in **settings/data_paths.yaml**. By default, for both **mn5** and **Storage5**, the path is **/gpfs/projects/bsc32/AC_cache/exp**.

Download non-interpolated model data from BSC HPC machines (BSC HPC users)

Defining models in interp_models.yaml

The `settings/interp_models.yaml` file contains a dictionary of default relevant models grouped by type, which contains the list of model names and their possible storage paths.

If a model is located in one of the predefined paths (for example `/esarchive/exp/monarch/`), it only needs to be added to the corresponding model list.

If the model is stored elsewhere, you must define:

- The **model type**
- The **model name**
- The **model storage directory**, excluding the model name

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/Interpolation.html#defining-models-in-interp-models-yaml>

```
{
  "monarch": {
    "models": ["a7kg", "a7kh", "a7rs", "alvi", "a6iz"],
    "paths": [
      "/esarchive/exp/monarch/"
    ]
  },
  "example_model_type": {
    "models": ["example_model_name"],
    "paths": [
      "/example/model/path"
    ]
  },
}
```

Download non-interpolated model data from BSC HPC machines (BSC HPC users)

Steps to download non-interpolated model data:

1. Include the model in your configuration file

You must include at least one model in your configuration file.

If the configuration file contains only model(s) and no observation(s) there is no need to set the download mode nor the interpolated mode, non-interpolated model data will be downloaded automatically.

2. Set the download mode

- Answer “**both**”/”**mod**” to the prompt:
 - *Which type of data do you want to download? Observational, modelled or both?*
- Or set **dl_mode** to **both** or **mod**.

3. Indicate that you want non-interpolated data

- Answer “**n**” to the prompt:
 - *Model data was detected in the configuration file. Do you want to download the interpolated version?
(Otherwise, the non-interpolated model data will be downloaded)*
- Or set **dl_interpolated** to **False**.

Configuration file example

```
[BSC-EXPERIMENT]
model = cams61_emep_ph2-eu-000
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180201
```

Download interpolated model data from BSC HPC machines (BSC HPC users)

An interpolated model consists of model output that have already been spatially interpolated to the observations.

Data structure

- To be discovered automatically on the machines, the data must follow the structure below:

`{mod_root}/{ghost_version}/{model_id}-{domain}-{ensemble}/{resolution}/{species}/{network}/
{species_YYYYMM.nc}`

Data locations

- Models are **saved** under the **mod_root** path specified in the **local** key of **settings/data_paths.yaml**.
- Models are **retrieved** from the **mod_root** path specified in the **storage5** or **mn5** key of **settings/data_paths.yaml**. By default, for both **MareNostrum5** and **Storage5**, the path is **/gpfs/projects/bsc32/AC_cache/recon/exp_interp**.

Download interpolated model data from BSC HPC machines (BSC HPC users)

Steps to download interpolated model data:

1. Include the model in your configuration file

You must include at least one model and at least one network in your configuration file.

2. Set the download mode

- Answer “**both**”/”**mod**” to the prompt:
 - *Which type of data do you want to download? Observational, modelled or both?*
- Or set **dl_mode** to **both** or **mod**.

3. Indicate that you want interpolated data

- Answer “**y**” to the prompt:
 - *Model data was detected in the configuration file. Do you want to download the interpolated version?
(Otherwise, the non-interpolated model data will be downloaded)*
- Or set **dl_interpolated** to **True**.

Configuration file example

```
[BSC-GHOST-EXPERIMENT]
network = EBAS
species = sconco3
resolution = hourly
start_date = 20180101
end_date = 20180201
model = cams61_emep_ph2-eu-000
dl_interpolated = True
dl_mode = mod
```

GHOST Zenodo Download



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Download GHOST observational data from Zenodo

GHOST observational networks uploaded to a [Zenodo repositories](#). An account is not required to use this feature.

Currently available versions are [1.5](#) and [1.5.1](#). To see the list of available networks, check the [available networks](#) section in the official documentation.

AERONET_v3_lev1.5.zip md5:291cc6b86d1372cbae3cd99205fd5f44	21.4 GB	Preview	Download
AERONET_v3_lev2.0.zip md5:4590e25a565c42c37eb909e233013cf7	12.7 GB	Preview	Download
CANADA_NAPS_axyat.zip md5:8c8dc0cb77717d12572c553c874fdfa7	1.6 GB	Preview	Download



Data location

- The networks are saved under the **ghost_root** path specified in the **local** key of **settings/data_paths.yaml**.

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/Zenodo-download.html>

Download GHOST observational data from Zenodo

Steps to download GHOST observational data:

1. Include the network in your configuration file

You must include at least one network in your configuration.

If the configuration file contains only observation(s) and no model(s) there is no need to set the download mode, network data will be downloaded automatically.

2. Set the download mode

- Answer **“both”/“obs”** to the prompt:
 - *Which type of data do you want to download? Observational, modelled or both?*
- Or set **dl_mode** to **both** or **obs**.

3. Set the observational data source

- Answer **“n”** to the prompt:
 - *Do you want to download observational data from the BSC remote machine? (Otherwise, GHOST observational data will be retrieved from Zenodo)*
- Or set **dl_ghost_source** to **zenodo**.

Configuration file example

```
[ZENODO]
ghost_version = 1.5
network = EBAS-EUCAARI
species = pm10
resolution = daily
start_date = 200912
end_date = 201001
dl_ghost_source = zenodo
```

ACTRIS Download



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Download of ACTRIS network data from Thredds

Observational data can be downloaded from the ACTRIS framework through [THREDDS](#). An account is not required to use this feature.

Species names are accepted using both the ACTRIS naming convention and the BSC naming convention.

Examples:

- sconco3 → ozone mass concentration
- sconcno3 → aerosol particle nitrate mass concentration

There is a total of 105 available species. Find the whole list [here](#).

Data location

- The framework is saved under the **nonghost_root** path specified in the **local** key of **settings/data_paths.yaml**.

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/ACTRIS-download.html>



Download of ACTRIS network data from Thredds

Steps to download ACTRIS observational data:

1. Include the framework in your configuration file

You must include the **actris/actris** framework in your configuration file.

If the configuration file contains only observation(s) and no model(s) there is no need to set the download mode, framework data will be downloaded automatically.

2. Set the download mode

- Answer **“both”/“obs”** to the prompt:
 - *Which type of data do you want to download? Observational, modelled or both?*
- Or set **dl_mode** to **both** or **obs**.

3. Decide whether to update the Thredds filetrees

- Answer **“y”** or **“n”** to the prompt:
 - *File containing information of the files available in Thredds for {actris_parameter} ({info_path}) already exists. Do you want to update it?*
- Or set **dl_thredds_update** to either **“True”** or **“False”**

Configuration file example

```
[ACTRIS]
framework = actris/actris
species = ozone mass concentration
resolution = hourly
start_date = 20180101
end_date = 20180201
dl_thredds_update = False
```

ADS CAMS Download



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Download of non-interpolated model data from the Atmosphere Data Store (ADS)

Non-interpolated model output data provided by CAMS can be downloaded from the [Atmosphere Data Store](#)

An ECMWF account is required to use this feature. For a tutorial on how to create and set up your account, see the [Account setup in Atmosphere Data Store](#) page.

Data locations

- The framework is saved under the **mod_to_interp_root** path specified in the **local** key of **settings/data_paths.yaml**.

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/CAMS-download.html>



Download of non-interpolated model data from the Atmosphere Data Store (ADS)

Steps to download CAMS model data:

1. Include one of the models in your configuration file

You must specify the model as **cams_analysis**, **cams_forecast** or **cams_reanalysis** in your configuration file.

If the configuration file contains only observation(s) and no model(s) there is no need to set the download mode, network data will be downloaded automatically.

2. Set the download mode

- Answer **“both”/“obs”** to the prompt:
 - *Which type of data do you want to download? Observational, modelled or both?*
- Or set **dl_mode** to **both** or **mod**.

3. Indicate that you want non-interpolated data

- Answer **“n”** to the prompt:
 - *Model data was detected in the configuration file. Do you want to download the interpolated version? (Otherwise, the non-interpolated model data will be downloaded)*
- Or set **dl_interpolated** to **False**.

Configuration file example

```
[CAM5]  
start_date = 20250701  
end_date = 20250702  
species = sconco2  
model = cams_analysis_dehm  
domain = regional  
resolution = hourly
```

esarchive/ to gpfs/ Transfer



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Transfer from **esarchive/** to **gpfs/** (BSC users)

Providentia provides a functionality to automatically copy non-interpolated models from **esarchive/** to **gpfs/**, making them available for interpolation on **MN5**.

Context

When working with non-interpolated models:

- **Nord4** contains most of the non-interpolated data in **esarchive/**, but performing interpolation there is slow.
- **MN5** is faster for interpolation, but **esarchive/** is not mounted, so the data is not directly accessible.

Data location

- The copied model is stored in the directory specified in **settings/data_paths.yaml** under the **exp_to_interp_root** path for the machine.
- The original location of the model in **esarchive/** must be specified in **settings/interp_models.yaml**.

Transfer from esarchive/ to gpfs/ (BSC users)

Steps to transfer non-interpolated model data from esarchive/ to gpfs/:

1. Connect to BSC storage5 or Nord4

Connect via SSH to storage5 or Nord4. Then go to the path where you have your Providentia repository cloned.

2. Include the model in your configuration file

You must include at least one model in your configuration file.

If the configuration file contains only model(s) and no observation(s) there is no need to set the download mode, model data will be downloaded automatically.

3. Set the download mode

- Answer “**both**”/”**mod**” to the prompt:
 - *Which type of data do you want to download? Observational, modelled or both?*
- Or set **dl_mode** to **both** or **mod**.

4. Indicate that you want non-interpolated data

- Answer “**n**” to the prompt:
 - *Model data was detected in the configuration file. Do you want to download the interpolated version? (Otherwise, the non-interpolated model data will be downloaded)*
- Or set **dl_interpolated** to **False**.

Use of wildcards

You can use the * wildcard in the following fields to automatically select all available values:

network, observation, framework
model, models, experiments, experiment
species
resolution
start_date
end_date

Example of a configuration file with wildcards

```
[WILDCARD]
network = EBAS
species = sconco3, sconco2
resolution = hourly
start_date = *
end_date = *
model = cams61_emep_ph2-eu-000
```

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/Download-mode.html#using-wildcards>

New configuration field: model_resolution

Models and observations may have different native temporal resolutions.

In Providentia, interpolation between model data and observations includes:

- Upsampling → duplicates temporal steps
- Downsampling → cuts or aggregates temporal steps

Now the data download and interpolation processes can be performed using the same configuration file.

```
[PRV_sconco3_CHIMERE]
network = EBAS
species = sconco3
resolution = daily
start_date = 20180101
end_date = 20180601
model = cams61_chimere_ph2-eu-000 (CHIMERE)
model_resolution = hourly
```

For more details, see the official documentation:

<https://providentia.readthedocs.io/en/latest/Download-mode.html#model-resolution>

BREAK
15 minutes



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

Download practical



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Download practical (BSC HPC users)

1. Download **hourly ozone** and **nitrogen dioxide** for the following **non-interpolated models** from the **BSC HPC machines** located in `/esarchive/exp/monarch/`, for **January - February 2023**:
 - **a59j-regional-006**
 - **a59g-regional-000**

Use the [GHOST standards file](#) to identify the “`bsc_parameter_name`” equivalent for ozone and nitrogen dioxide.

2. Download **ozone** and **nitrogen dioxide** for the **observational** European **EEA** network in **hourly** and **daily** resolutions from the **BSC HPC machines** using **GHOST version 1.6** for **January - February 2023**.
3. Combine the two configuration files and automate the download so that **questions do not appear**. **Do not overwrite existing files**. Help can be found here:

[Automation of the download](#)

4. Download **observational ozone** again from the European **EEA** network from the **BSC HPC machines** using **GHOST version 1.6** for **January - February 2023** for **all available resolutions**. Use wildcards as described here:

[Using wildcards](#)

Useful configuration fields

ghost_version

network

species

resolution

start_date

end_date

model

model_resolution

dl_overwrite

dl_ghost_source

dl_interpolated

dl_mode

dl_thredds_update

Download practical (Outsiders)

1. Check your data_paths.yaml

- Go to your **settings/data_paths.yaml** file and verify where the local data paths are defined.

2. Download the data

- Go to the [shared folder](#). Click **Download all** to download the complete dataset.

3. Copy the zip file to the correct directory

- Once the download finishes: Move or copy the files into the path **settings/data_paths.yaml** indicates.
- If you did not change the path copy it to your home directory.

4. Unzip the files

- Unzip the downloaded files inside the correct directories:



Nombre	Propietario	Fecha de modificación	Tamaño de e	Ordenar
providentia	Propietario oculto	24 feb	—	

```
local:  
ghost_root: ~/data/providentia/obs/ghost  
mod_root: ~/data/providentia/mod  
mod_to_interp_root: ~/data/providentia/mod_to_interp  
nonghost_root: ~/data/providentia/obs/nonghost
```

Exercise 1

Download **hourly ozone** and **nitrogen dioxide** for the following **non-interpolated models** from the **BSC HPC machines** located in **/esarchive/exp/monarch/**, for **January - February 2023**:

- **a59j-regional-006**
- **a59g-regional-000**

```
[DL1]
model = a59g-regional-000,
a59j-regional-006
species = sconco3, sconco2
resolution = hourly
start_date = 20230101
end_date = 20230201
```

Exercise 2

Download **ozone** and **nitrogen dioxide** for the **observational** European **EEA** network in **hourly** and **daily** resolutions from the **BSC HPC machines** using **GHOST version 1.6** for **January - February 2023**.

```
[DL2]
ghost_version = 1.6
network = EEA
species = sconco3, sconco2
resolution = hourly, daily
start_date = 20230101
end_date = 20230201
```

Exercise 3

Combine the two configuration files and automate the download so that **questions do not appear. Do not overwrite existing files.**

```
[DL3]
ghost_version = 1.6
network = EEA
species = sconco3, sconco2
resolution = hourly, daily
start_date = 20230101
end_date = 20230201
model = a59g-regional-000, a59j-regional-006
dl_mode = both
dl_interpolated = false
dl_ghost_source = bsc
dl_overwrite = true
```

Exercise 4

Download **observational ozone** again from the European **EEA** network from the **BSC HPC machines** using **GHOST version 1.6** for **January - February 2023** for **all available resolutions**. Use wildcards.

```
[DL4]
ghost_version = 1.6
network = EEA
species = sconco3, sconco2
resolution = *
start_date = 20230101
end_date = 20230201
```

Interpolation



**Barcelona
Supercomputing
Center**

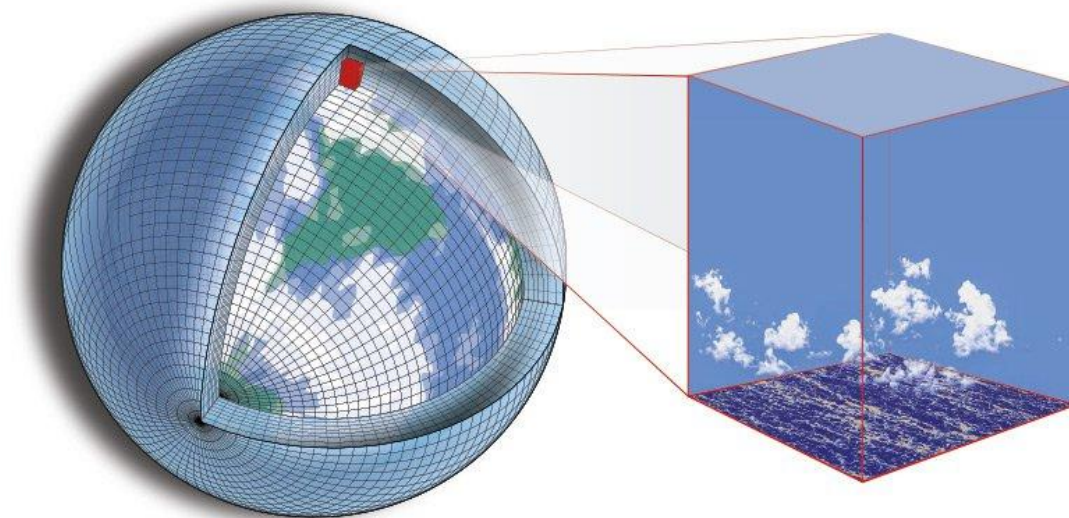
Centro Nacional de Supercomputación

What does interpolation do?

Providentia makes use of in-situ observations from specific measurement stations, at fixed locations.

When you have model data, its output is typically provided on a regular model grid at a fixed resolution.

Interpolation in Providentia refers to the mathematical process of estimating the model variable at the exact location of an observational station.

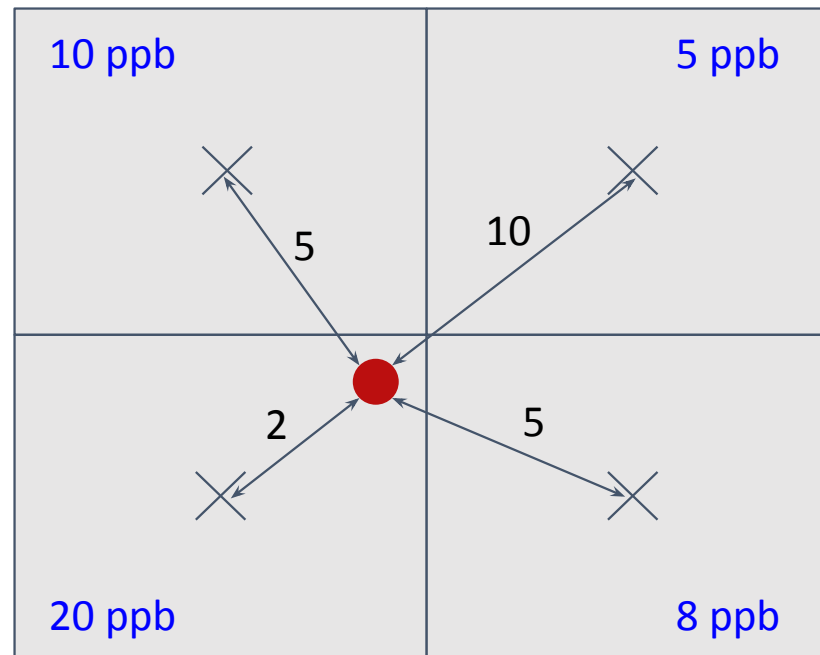


Inverse distance weighted interpolation

Inverse Distance Weighted interpolation estimates the value at an observational station by averaging model data in **N closest neighbour** grid cells, giving higher weight to grid cells that are closer.

N neighbours can be set via **interp_n_neighbours = N** in the configuration file. The default is **4**.

For 4 closest neighbours (default):



Weights

$$\frac{1}{2}, \frac{1}{5}, \frac{1}{5}, \frac{1}{10}$$
$$= 0.5, 0.2, 0.2, 0.1$$

Weighted average = **14.1 ppb**

Nearest neighbour = **20 ppb**

Setting path for model data to interpolate

On local machines the path where the model data used for interpolation is defined in **settings/data_paths.yaml** as **mod_to_interp_root**. As long as this is set, nothing more is needed.

On **BSC HPC** machines, models can be stored in a variety of different locations, therefore a separate file exists which stores information for the different paths where models can be found:

settings/interp_models.yaml. In this file the specific model ID is needed to be set under a specific path grouping, e.g. “monarch”. If the model ID is not set there, the path for **mod_to_interp_root** in **settings/data_paths.yaml** will be secondarily checked. See [here](#) for more information.

settings/data_paths.yaml

```
local:
  mod_to_interp_root:
    ~/data/providentia/mod_to_interp
```

settings/interp_models.yaml

```
"monarch": {
  "models": [ "a005" ],
  "paths": [
    "/esarchive/exp/monarch/"
  ]
}
```

Mismatching observational and model temporal resolutions

Model simulations are sometimes run at temporal resolutions that don't match the resolution of observations. Providentia currently needs matching temporal resolutions for model evaluation, therefore work is needed to be done to match these.

If no information is given about the mismatching resolutions, Providentia attempts to handle this automatically for you. In all cases it is the model resolution that is adjusted to match the observational resolution, which is taken to be fixed via the **resolution** argument.

Providentia goes through the following operations to match the model resolution to the observational resolution:

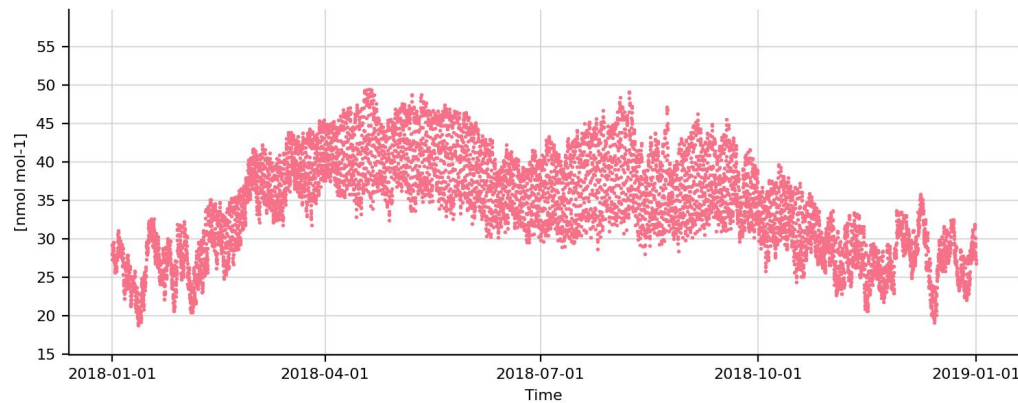
1. Firstly, it is checked if a model resolution matches the observational resolution. If so, model data is taken as is.
2. Next, it is checked if a finer model resolution exists than the observational resolution. Model downsampling is used if so.
3. Finally, it is checked if a coarser model resolution exists than the observational resolution. Model upsampling is used if so.

The **model_resolution** variable can be used to explicitly tell Providentia which resolution model data to use.

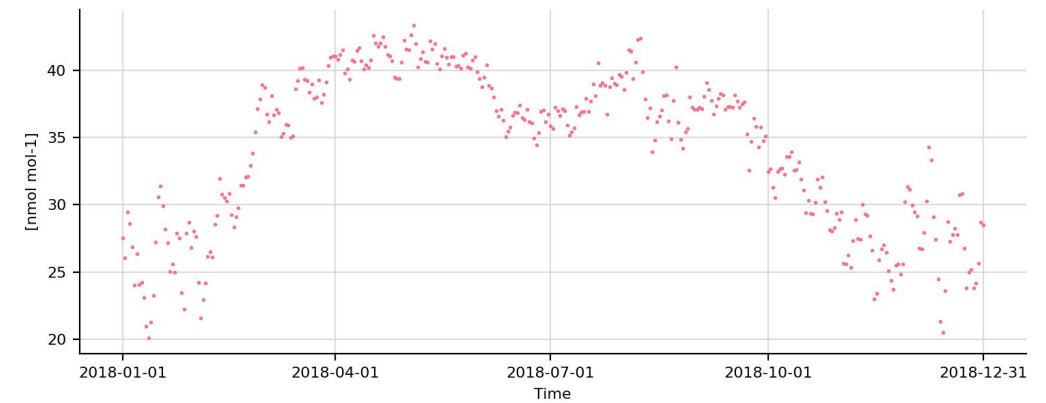
Downsampling model resolution

If we have a model resolution that is finer than our observational resolution, this can be handled via an operation called **downsampling**. This is as simple as averaging the finer model resolution to a coarser resolution.

This can be set via `interp_model_downsampling = [option]` in the configuration file, with the two valid options being **mean** and **median**. The default is **mean**.



→
mean



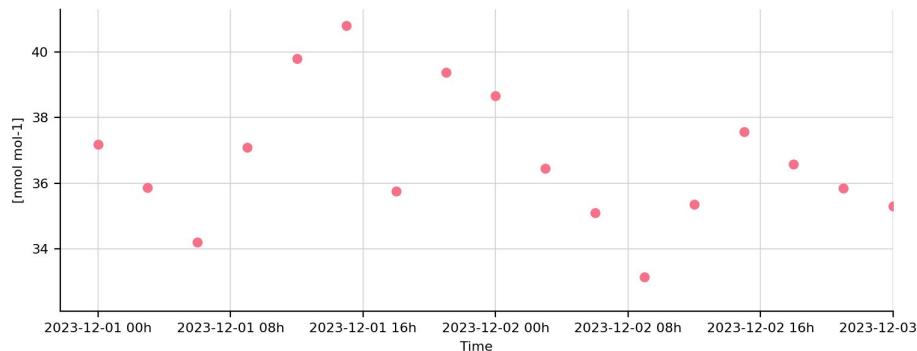
Hourly

Daily

Upsampling model resolution

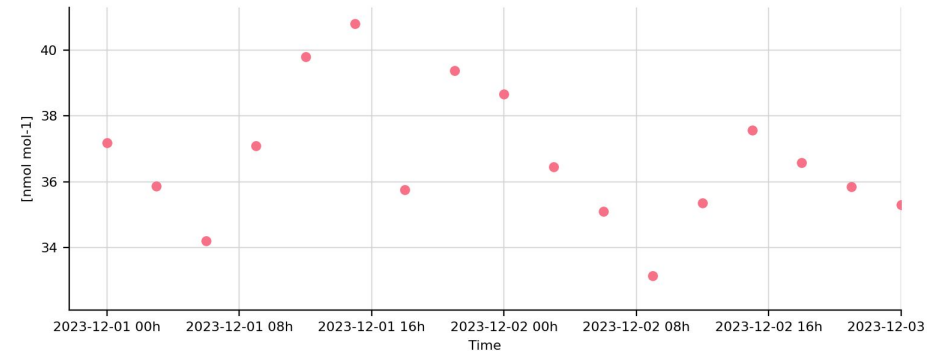
If we have a model resolution that is coarser than our observational resolution, this can be handled via an operation called **upsampling**.

This can be set via `interp_model_upsampling = [option]` in the configuration file, with the two valid options being **fill** and **gaps**. The default is **fill**.

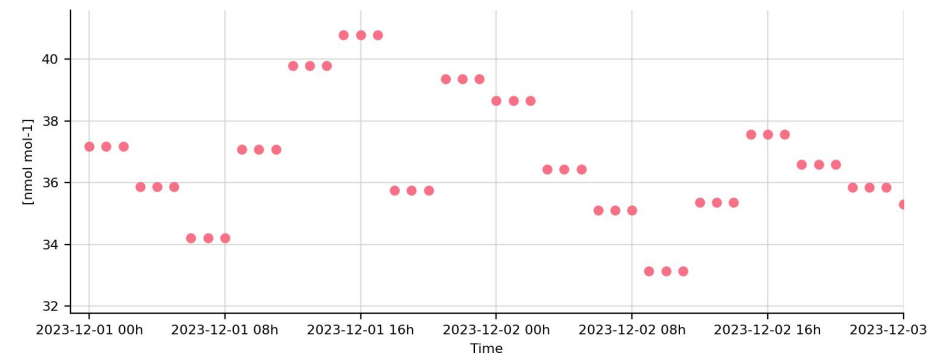


3Hourly

gaps



fill



Hourly

Forecast interpolation

There are various kinds of model data which can be handled by Providentia, including free-running, diagnostic, analysis, reanalysis and forecast output.

Forecast models predict the state of the atmosphere in the near future, most typically on the scale of days. One issue with forecast data is that it comes in a different format to most other kinds of model data, with data per day for the subsequent N days.

For the interpolation to handle forecast data correctly, a valid option for the **forecast** field must be set in the configuration file, these are:

forecast = day, dayN, daily, dailyN, combined

If the forecast field is not set then, then only first day of the forecast is interpolated and is subsequently treated as non-forecast data.

Mapping species

Sometimes the component provided in the observations does not exactly match the component given in a model. For example a model may provide information in 4D (e.g. time, lon, lat, alt), whereas in the observations it may be given in 3D (e.g. time, lon, lat). For these cases Providentia provides some allowance so that data can be mapped together.

This information can be set in **settings/mapping_species.yaml**, with the syntax for doing so being:

[observational component]

- **[model component]**

The typical use cases for mapping are:

1. 3D observational component to 4D model component. This can be either be through an additional vertical dimension in the model, e.g. sconco3 : vmro3 ; or an additional binned dimension in the model, e.g. vconcaerobin1 : loaddu
2. Observational component to fractional model component, e.g. pm10 : pm10du

settings/mapping_species.yaml

```
sconco3
- vmro3
vconcaerobin1
- loaddu
pm10
- pm10du
```

Interpolation parallelisation

When run locally, the interpolation runs in parallel using the Python **multiprocessing** package, with the number of cpus used determined automatically based on system load.

On **BSC machines**, the interpolation runs in parallel by default via the **Greasy** package, designed to handle very large submissions of interpolation jobs.

If you wish to use the multiprocessing package rather than Greasy on **BSC machines**, simply add **interp_multiprocessing=True** to your configuration file.

Other Interpolation configuration options

Variable	Purpose
interp_reverse_vertical_orientation	Reverse vertical order of model levels. Used when vertical grid erroneously reversed.
interp_chunk_size	Minimum number of jobs per interpolation chunk. Used for Greasy jobs.
interp_job_array_limit	Maximum number of chunks in the job array. Used for Greasy Jobs.
interp_multiprocessing	Use multiprocessing parallelisation instead of Greasy on HPC systems.
interp_spinup_timesteps	Number of initial model timesteps skipped to account for model spin-up.
network_type	Determines whether to use all GHOST or all non-GHOST networks when the network field uses the * wildcard.
model_resolution	Specific model resolution to interpolate.

Run interpolation

If **--interp**, **--interpolate** or **--interpolation** is added as a launch option together with the mandatory configuration file on the command line, the interpolation will begin:

```
./bin/providentia --conf=configuration_name.conf --interp
```

configurations/configuration_name.conf

```
[A005]
start_date = 20180101
end_date = 20190101
resolution = hourly
species = sconco3
network = EBAS
models = cams61_chimere_ph2-eu-000 (CHIMERE)
```

Note:

Each interpolation job covers one full month, thus for the given **start_date** and **end_date** only **YYYYMM** is considered, **DD** will be ignored.

Output Logs

When running the interpolation locally, information will be output in realtime to the command line as it goes on. Other logs will be generated in multiple different folders inside the *logs/interpolation* folder:

- |— [management_logs/](#) Informs on the output of the overall interpolation. Go here first.
- |— [interpolation_logs/](#) Stores output information of individual interpolations.
- |— [submission_logs/](#) Gives information on the slurm and Greasy submissions to the HPC machines ([BSC users](#)).

Interpolation practical



**Barcelona
Supercomputing
Center**

Centro Nacional de Supercomputación

Interpolation practical

1. Perform a model interpolation for **hourly ozone** of **a59j-regional-006** model in **January - February 2023**, against the European **EEA** network, using **GHOST version 1.6**.
2. Perform a **forecast** model interpolation for **hourly ozone** and **nitrogen dioxide** of both **a59j-regional-006** and **a59g-regional-000** models in **January - February 2023**, against the European **EEA** network, using **GHOST version 1.6**, using **8 nearest neighbours**.
3. Perform a **forecast** model interpolation for **daily nitrogen dioxide** of the **a59g-regional-000** model in **January - February 2023**, against the European **EEA** network, using **GHOST version 1.6**, using **12 nearest neighbours**, **downsampling** the model using the **median**.

Useful configuration fields

ghost_version

network

species

resolution

start_date

end_date

model

forecast

interp_n_neighbours

Interp_model_downsampling

interp_model_upsampling

model_resolution

interp_spinup_timesteps

Exercise 1

Model interpolation for **hourly ozone** of
a59j-regional-006 model in **January - February 2023**,
against the European **EEA** network, using **GHOST version**
1.6.

```
[INTERP1]
ghost_version = 1.6
network = EEA
species = sconco3
resolution = hourly
start_date = 202301
end_date = 202302
model = a59j-regional-006
```

Exercise 2

Forecast model interpolation for **hourly ozone** and **nitrogen dioxide** of both **a59j-regional-006** and **a59g-regional-000** models in **January - February 2023**, against the European **EEA** network, using **GHOST version 1.6**, using **8 nearest neighbours**.

```
[INTERP2]
ghost_version = 1.6
network = EEA
species = sconco3, sconco2
resolution = hourly
start_date = 20230101
end_date = 20230201
model = a59j-regional-006, a59g-regional-000
forecast = day
interp_n_neighbours = 8
```

Exercise 3

Forecast model interpolation for **daily nitrogen dioxide** of the **a59g-regional-000** model in **January - February 2023**, against the European **EEA** network, using **GHOST version 1.6**, using **12 nearest neighbours**, **downsampling** the model using the **median**.

```
[INTERP3]
ghost_version = 1.6
network = EEA
species = sconcn02
resolution = daily
start_date = 20230101
end_date = 20230201
model = a59g-regional-000
forecast = day
interp_n_neighbours = 12
interp_model_downsampling = median
model_resolution = hourly
```



**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación



Thank you for your attention!

More information at:

<https://github.com/bsc-es/providentia>

Join the #providentia Slack channel!

paula.serrano@bsc.es | alba.vilanova@bsc.es | dene.bowdalo@bsc.es