



**Barcelona  
Supercomputing  
Center**  
*Centro Nacional de Supercomputación*

# Providentia v2.4.0

## User Meeting

28/08/2024

Paula Serrano | Alba Vilanova | Dene Bowdalo

# Major updates



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Major updates

- Make Providentia work in MN5
- Integrate Providentia Interpolation inside Providentia
- Enable Providentia to run locally on Linux and MacOS machines
- Download data locally from esarchive and Zenodo
- Use of conda environments
- Speed optimisation
- Upgrade most modules to latest versions
- Use of CI/CD pipelines for testing
- Changed operation of spatial colocation

# Mode summary



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Mode summary

	Main purpose	Argument	Configuration file	HPC	Local
<b>Dashboard</b>	On-the fly analysis		✓	✓	✓
<b>Offline</b>	Detailed analysis in PDFs format	--offline	✓	✓	✓
<b>Interactive</b>	DIY	--interactive	✓	✓	✓
<b>Interpolation (NEW!)</b>	Interpolate experiments to observation networks	--interp, --interpolate, --interpolation	✓	✓	x
<b>Download (NEW!)</b>	Download data to local machine	--dl, --download	✓	x	✓
<b>Debug</b>	Avoid waiting in queue to relaunch Providentia	--debug	x	✓	x
<b>Clean (NEW!)</b>	Clean logs	--clean	x	✓	✓
<b>Generate file tree (NEW!)</b>	Check for existing directories to make launch faster	--gft, --generate_file_tree	x	✓	✓



# Interpolation mode



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Interpolation mode

Providentia Interpolation has been merged into Providentia. Now, you can obtain the entire repository by simply cloning Providentia, with **no need to use recursive submodules**.

```
git clone https://earth.bsc.es/gitlab/ac/Providentia.git
```

If `--interp`, `--interpolate` or `--interpolation` is added as a launch option together with the mandatory configuration file in the command line, the interpolation will start:

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

OR

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

OR

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

# Log files

This will generate logs in different folders inside the *providentia/interpolation* folder:

- **Submission logs:** To give information on the slurm and Greasy submissions to the HPC machines.
- **Management logs:** To inform about the interpolation overall. Most errors will appear here.
- **Interpolation logs:** To track the details and duration of individual interpolations.

## Different date format

In interpolation, the date format is *YYYYMM*, while in Providentia, it's *YYYYMMDD*.

- If a *YYYYMMDD* date is passed in interpolation, it's automatically shortened to *YYYYMM*.
- If a *YYYYMM* date is passed in Providentia, it defaults to *YYYYMM01*.



# Evolution of configuration files

Thanks to this update. You now can save multiple configurations.

## OLD: configuration.py

```
qos="gp_bscs"
GHOST_version = 'default'
n_neighbours_to_find = 'default'
start_date = '201801'
end_date = '201901'
species to process = ['sconco3']
experiments_to_process = ['cams61_monarch_ph3']
grid_types_to_process = ['eu']
ensemble_options = ['allmembers']
networks_to_interpolate_against = ['EBAS']
temporal_resolutions_to_output = ['hourly']
```

## NEW: any configuration file

```
[NEW]
ghost_version = default
interp_n_neighbours = default
start_date = 201801
end_date = 201901
species = sconco3
experiments = cams61_monarch_ph3-eu-allmembers (Forecast)
network = EBAS
resolution = hourly
statistic_mode= Temporal|Spatial
statistic_aggregation = Median
periodic_statistic_mode = Independent
periodic_statistic_aggregation = Median
timeseries_statistic_aggregation = Median
```

Fields from other modes

# Parameter changes

Old Providentia-Interpolation	Interpolation mode
qos	<b>No longer available</b>
GHOST_version	ghost_version
n_neighbours_to_find	interp_n_neighbours
start_date	start_date <b>(unchanged)</b>
end_date	end_date <b>(unchanged)</b>
species_to_process	species
experiments_to_process	experiments
grid_types_to_process	domain
ensemble_options	ensemble_options <b>(unchanged)</b>
networks_to_interpolate_against	network
temporal_resolutions_to_output	resolution

# Combining ensemble, domain and experiment

Previously, in Providentia-Interpolation, experiments were set up with separately using the experiments, ensemble\_options and domain fields. Now, you can configure them in various ways:

- Use experiment, ensemble member and domain independently

```
experiments = cams61_monarch_ph3  
domain = eu  
ensemble_options = allmembers
```

```
experiments = cams61_monarch_ph3  
ensemble_options = allmembers
```

```
experiments = cams61_monarch_ph3  
domain = eu
```

```
experiments = cams61_monarch_ph3
```

- Combine experiment and domain

```
experiments = cams61_monarch_ph3-eu  
ensemble_options = allmembers
```

```
experiments = cams61_monarch_ph3-eu
```

# Combining ensemble, domain and experiment

- Combine experiment and ensemble

```
experiments = cams61_monarch_ph3-allmembers  
domain = eu
```

```
experiments = cams61_monarch_ph3-allmembers
```

- Use experiment field only

```
experiments = cams61_monarch_ph3-eu-allmembers
```

**\*Aliases only work in this specific way, and also when there is only a single experiment, a single ensemble member, and a single domain.**



**These changes have been applied to all modes.**

# New interp\_experiments.yaml file

This file replaces the old **defined\_experiments.py**. All experiments used in interpolation should be included here. If you need to use an experiment that isn't listed, add it to this file.

The path that gets used is the first one in the list that works on the machine.

## OLD: defined\_experiments.py

```
{ 'a7bn': {'esarchive': '/esarchive/exp/monarch/a7bn/'},  
  'a7bo': {'esarchive': '/esarchive/exp/monarch/a7bo/'},  
  'a7bp': {'esarchive': '/esarchive/exp/monarch/a7bp/'},  
  'a7by': {'esarchive': '/esarchive/exp/monarch/a7by/'},  
  'a7bz': {'esarchive': '/esarchive/exp/monarch/a7bz/'},  
  'a2in': {'esarchive': '/esarchive/exp/monarch/a2in/'},  
  'a4xf': {'esarchive': '/esarchive/exp/monarch/a4xf/'},  
  'a1vi': {'esarchive': '/esarchive/exp/monarch/a1vi/'}}
```

## NEW: interp\_experiments.yaml

```
{"monarch": {  
  "experiments": ["a1vi", "a6iz", "a123", "a3ku", "a1wd", etc.],  
  "paths": [  
    "/gpfs/scratch/bsc32/bsc32359/esarchive_scratch/exp/monarch/",  
    "/esarchive/exp/monarch/"  
  ]  
},  
...  
"example_experiment": {  
  "experiments": ["example_experiment_name"],  
  "paths": [  
    "/example/experiment/path"  
  ]  
}  
}}
```

# Local runs



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*



# Local runs

You might want to run Providentia on your local machine to **avoid waiting for a job allocation** and be able to use the interactive features of the dashboard **without delay**.

To do this, use the tool as usual. The first time you run it a conda environment will be created in your local machine, and all necessary dependencies will be installed. Providentia will get the data by default from the following paths (defined in *settings/data\_paths.yaml*):

```
"ghost_root": "~/data/providentia/obs/ghost"
```

```
"nonghost_root": "~/data/providentia/obs/nonghost"
```

```
"exp_root": "~/data/providentia/exp"
```

# Download mode



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Download mode

Now it is possible to download data directly into your machine outside HPC via the download mode.

If `--dl` or `--download` is added as a launch option together with the mandatory configuration file in the command line, the download will start:

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

OR

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

This will get the data that needs to be downloaded from your configuration file and save it into the directories specified in `settings/data_paths` for local.

# Download mode

The download mode fetches all the content specified in your configuration file across all sections, similar to how it functions in offline mode.

However, if you want to fetch content from only one specific section, you can pass the **section** keyword argument.

All downloads, including GHOST, Non-GHOST networks and experiment files are done via the **Transfer5** machine (or **MN5** if Transfer5 is unavailable) using **SFTP** for secure and efficient file transfer.

For those without access to the BSC transfer machines, GHOST files are also available for download from the GHOST [Zenodo](#) page.



**BSC users are strongly encouraged to use the Transfer5/MN5 option as it offers faster download speeds and access to more networks.**

# Relevant download fields

In this mode, only the following **seven** fields affect the output:

1. network
2. species
3. resolution
4. start\_date
5. end\_date
6. experiments
7. filter\_species

Other fields can be included in the configuration, but they will not impact the results.

Additionally, some fields have been updated in their usage. Fields that were previously mandatory are now optional, and some that used to accept a single value now support multiple values.

# Mandatory fields

Parameter	Download mode	Other modes
network	✓	✓
species	X	✓
resolution	X	✓
start_date	✓	✓
end_date	✓	✓
experiments	X	X
filter_species	X	X

\*It is not mandatory meaning Providentia will not add new species or resolutions if they are not provided in the configuration file.

## Single to Multi-Value

The **Resolution** field, which is restricted to a single value in other modes, now supports multiple values specifically in Download mode.

## Downloading all available options

If the **Species** or **Resolution** fields are left blank or set to their default value, the system will automatically download all possible species and resolutions. To download all the networks you have to use the wildcard asterisk (\*).



# .env file

The .env file is a recent addition designed to **store** specific **user settings**.

It will be **automatically** generated the first time Download mode is used with user input, provided the user opts to save their preferences.

There are four possible fields that can be added, here an example of an .env file with all of them:

```
BSC_DL_CHOICE=y  
OVERWRITE=y  
PRV_USER=bsc000000  
PRV_PWD=example_pwd
```

Once created, the .env file can be accessed and modified manually from the **Providentia root directory**.

Additionally, if a field is removed, it will be generated with the user input on the next execution if the user opts to restore it.

# .env file

## BSC\_DL\_CHOICE

This field indicates the method of downloading GHOSTs networks.

In BSC the most common method should be from the remote machine.

### Possible Values:

- **y** – Enables BSC remote machine SFTP download of GHOST.
- **n** – Enables GHOST download from the Zenodo webpage GHOST publication.

## OVERWRITE

This field determines whether to overwrite an existing file if a download is attempted for a file that is already present.

### Possible Values:

- **y** – Allows overwriting of the file, so it will be downloaded again.
- **n** – Prevents downloading if the file is already present.

# .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** or **ssh {username}@glogin4.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**

# File trees



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Generation of file trees

- When Providentia was previously launched it first created 2 internal file trees, one for all available GHOST observations, and one for all available non-GHOST observations, by iterating through all relevant observational directories. Doing so however costs time (up to 15 seconds on Nord3v2), and in most cases, the filetrees do not change across executions (i.e. no new data).
- In order to counter this, a way of disabling the filetree generation was created, with filetrees being saved to *settings/internal/ghost\_filetree\_version.json* and *settings/internal/nonghost\_filetree.json*, which are then loaded upon execution.
- By default filetree generation is off on all machines, other than Marenostrom5 and locally (as generation is fast in these cases).
- The *--dft* , *--disable\_file\_tree* flags can be used to disable generation of the filetree.
- The *--gft* , *--generate\_file\_tree* flags can be used to enable generation of the filetree.
- New non-GHOST networks can be added in *settings/init\_prov.yaml*

# Customisation features



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*



# Custom colormaps

Now the legend and colorbar colormaps can be customized by adding a new schema in *settings/color\_palettes.yaml*.

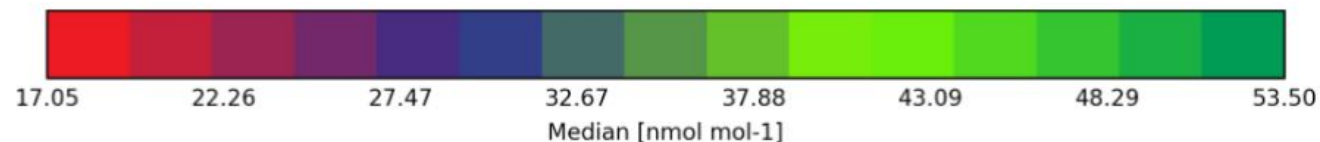
```
"cams": ["#ED1B23", "#2D2F92", "#7CFC00", "#009B55"]
```



To update the legend colors, you will need to set the custom name in *legend\_color\_palette* in the plot characteristics files.

● observations ● EMEP ● MONARCH ● SILAM

For the colorbar, you should define *cmap\_absolute* or *cmap\_bias*, depending on the statistic type. The colors will interpolate depending on *n\_discrete*. Below *n\_discrete* is 15.



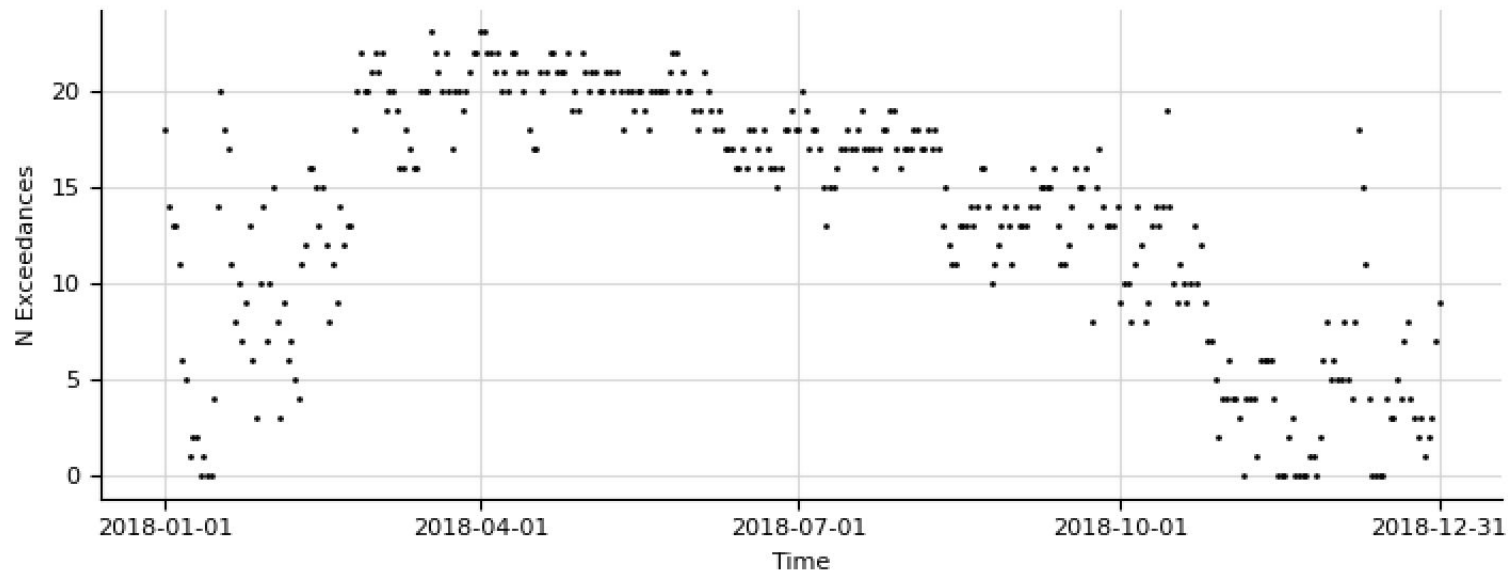
# Custom colorbar style per statistic

Up until now we could only set the colormap per statistic, with this change now you can set the number of ticks and discrete colors individually in *settings/basic\_stats.yaml* and *settings/experiment\_bias\_stats.yaml*.

```
"Median" : {"function": "calculate_median",  
            "order": 9,  
            "label": "Median",  
            "arguments": {},  
            "units": "[measurement_units]",  
            "minimum_bias": [0.0],  
            "vmin_absolute": {},  
            "vmax_absolute": {},  
            "vmin_bias": {},  
            "vmax_bias": {},  
            "cmap_absolute": "cams",  
            "cmap_bias": "cams"}
```

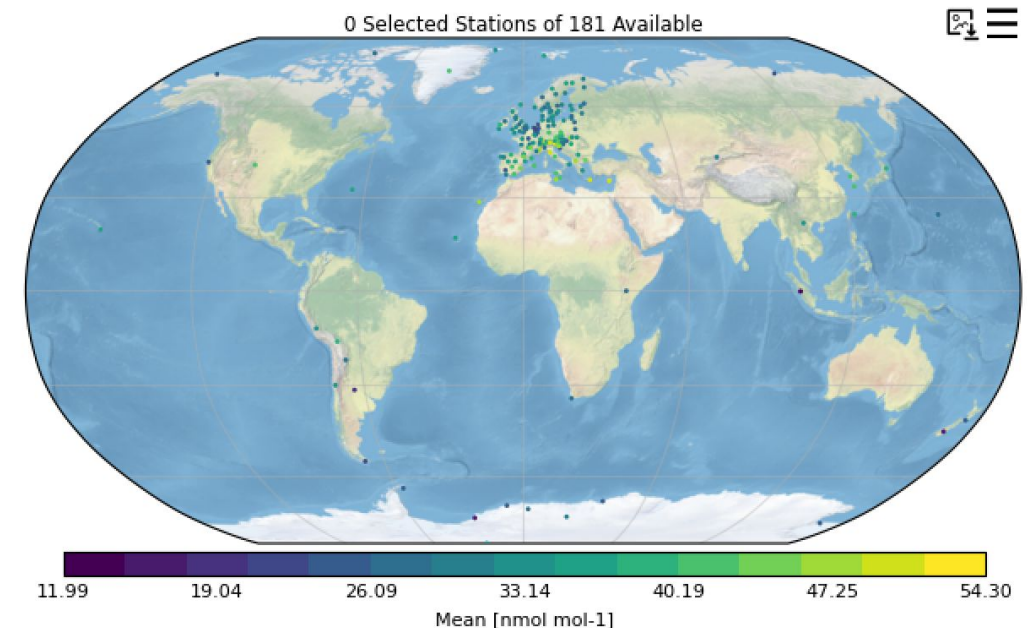
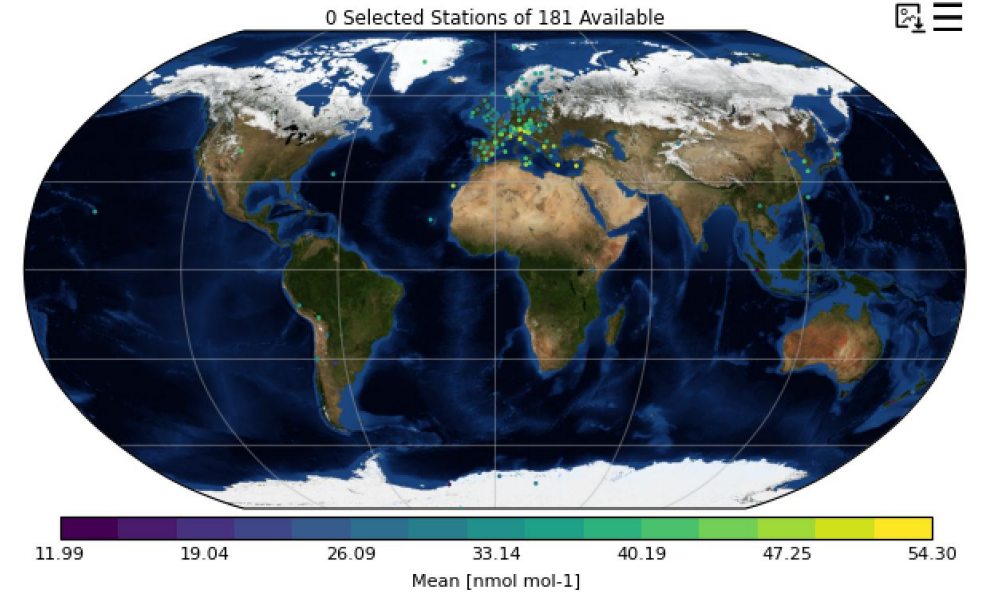
# exceedances.yaml update

- Updated the *settings/exceedances.yaml* to better indicate what can be done with exceedance limits (linked to the **Exceedances** statistic).
- Exceedances can be defined per component, or for a network|component pair.
- The exceedances statistic will simply count the number of instances above these thresholds



# Map background Images

- Users now have the ability to plot a background image on the map axis.
- This can be set by changing the **background** variable under the **map** section in the appropriate plot characteristics file
- There are 3 available standard options:
  1. **providentia** (default)
  2. **blue\_marble** (NASA's blue marble product)
  3. **shaded\_relief** (imagery showing changes in evolution)
- However, any image can be set. Check the Plot Customisation wiki page for all the details.



# Plots types and options



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Plot options



Available and new



Available



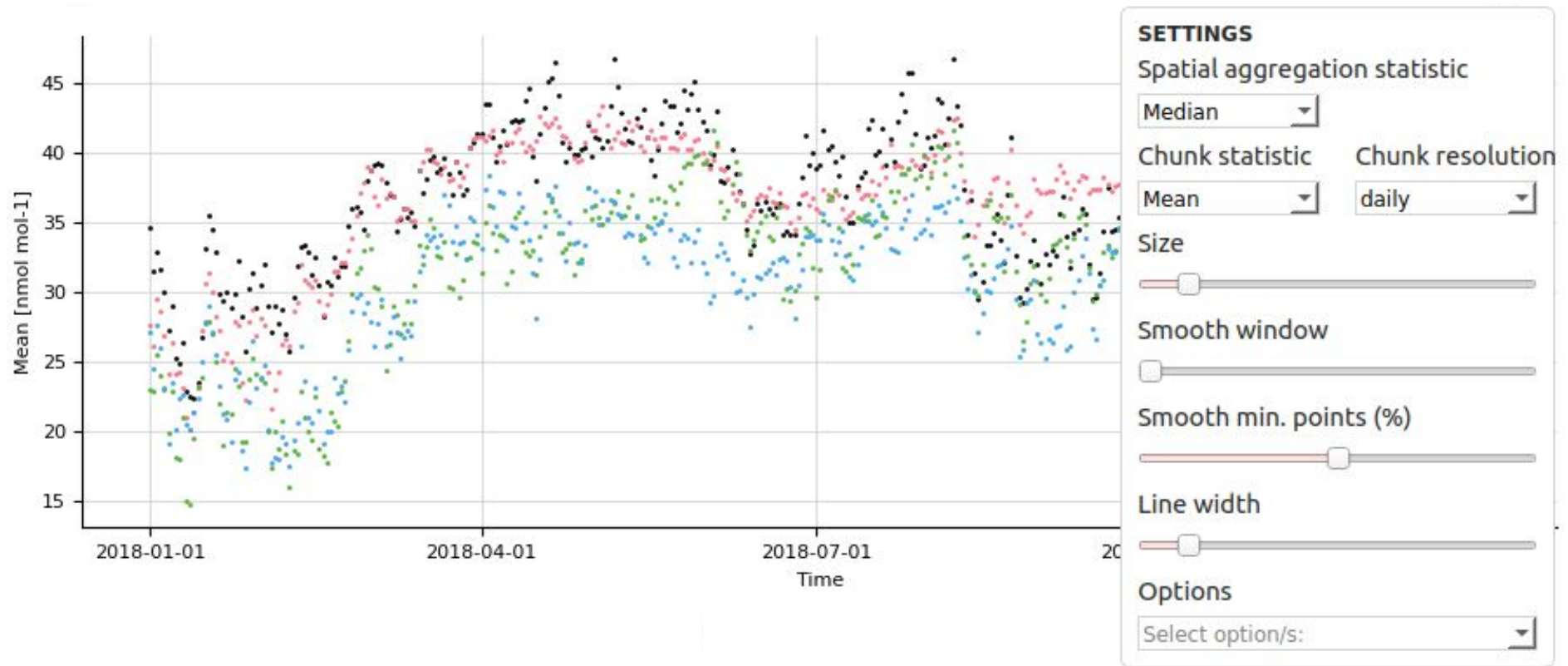
Unavailable

	-[stat]	_bias	_annotate	_regression	_multispecies	_logx	_logy	_smooth	_hidedata	_domain	_threshold
map	Available	Available	Available	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Available	Unavailable
timeseries	Available and new	Available	Available	Unavailable	Unavailable	Unavailable	Available	Available	Available	Unavailable	Available and new
periodic	Available	Available	Available	Unavailable	Unavailable	Unavailable	Available	Unavailable	Unavailable	Unavailable	Available and new
periodic-violin	Unavailable	Unavailable	Available	Unavailable	Unavailable	Unavailable	Available	Unavailable	Unavailable	Unavailable	Available and new
distribution	Unavailable	Available	Available	Unavailable	Unavailable	Available	Available	Unavailable	Unavailable	Unavailable	Available and new
scatter	Unavailable	Unavailable	Available	Available	Unavailable	Available	Available	Unavailable	Available	Unavailable	Available and new
heatmap	Available	Available	Available	Unavailable	Available	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable
table	Available	Available	Unavailable	Unavailable	Available	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable
boxplot	Unavailable	Unavailable	Available	Unavailable	Available	Unavailable	Available	Unavailable	Unavailable	Unavailable	Available and new
statsummary	Unavailable	Available	Unavailable	Unavailable	Available	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable
metadata	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable
taylor	Available and new	Unavailable	Available and new	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable	Unavailable



# Statistical timeseries

The chunk variables in the dashboard will allow you to create statistical timeseries. To do the same in the offline reports and interactive mode, you need to define the plot type as "timeseries-[statistic]\_[resolution]" (example: timeseries-Mean\_daily).

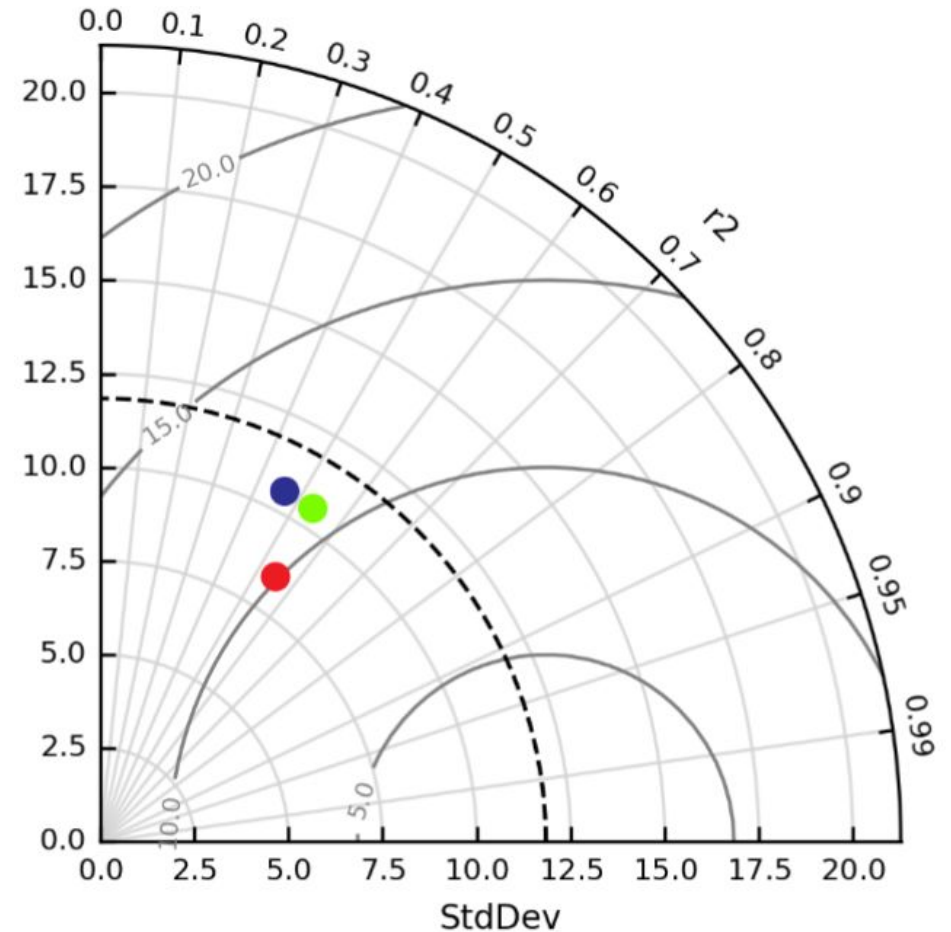


# Taylor diagram

The Taylor diagram provides a quick outlook on how the models compare to a reference value (from the observations) plotting the standard deviation and correlation coefficient.

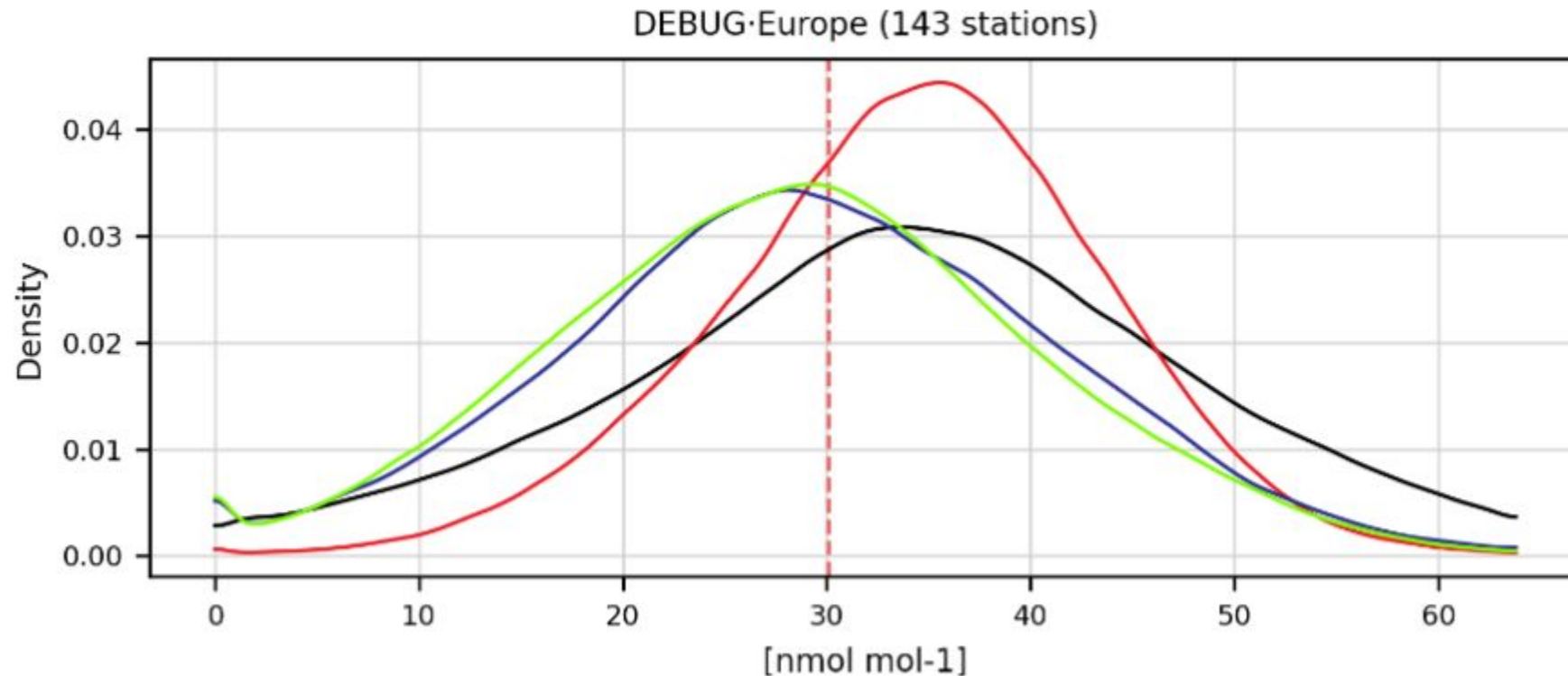
In Providentia, users can choose to see  $r$  or  $r^2$  (taylor-r and taylor-r2 in offline and interactive modes).

This plot is only available in MN5 and locally and cannot be created in Nord3v2 because the modules there are too old.



# Threshold lines

In this new version we can indicate the exceedances by plotting threshold lines in different plot types (timeseries, distribution, boxplot, periodic and scatter). The value to plot can be modified in *settings/exceedances.yaml*. Below we are plotting the default value for ozone, which is  $30.07 \text{ nmol mol}^{-1}$ .



# New statistics

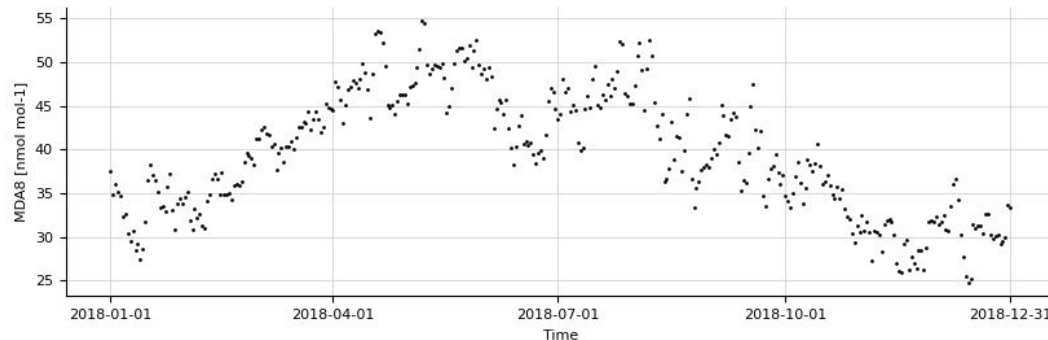


**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

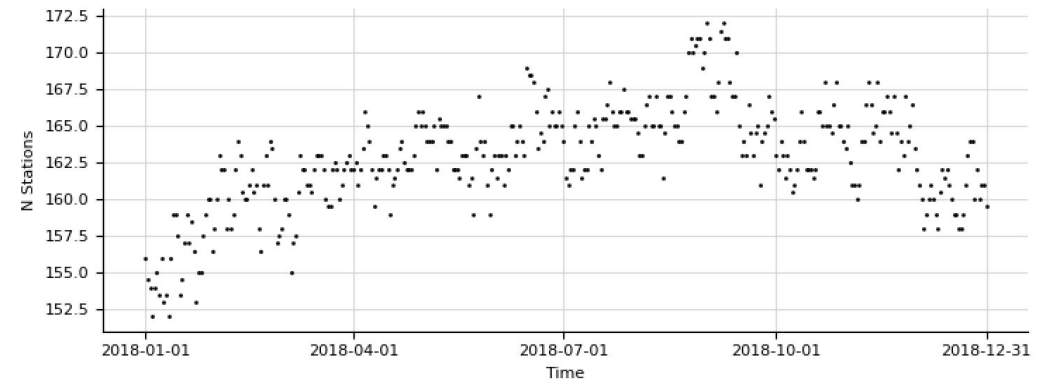
# MDA8

- Daily maximum 8 hour average
- Calculated using all 8 hour windows  
WITHIN the UTC day, i.e. 17 windows from 00:00 - 08:00 to 16:00 - 24:00.
- Only available when resampling hourly to daily resolution.
- Statistic designed for  $O_3$ , but can be used for any component.



# NStations

- Simply the number of stations for which statistical information is calculated at.
- This is only relevant for the **temporal / spatial statistical** mode, as for the other modes, the spatial dimension is removed before statistics are calculated.



# Configuration changes



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*



# Configuration changes

- "resampling" variable is not used anymore and will be deemed as invalid, to activate the resampling you just need to define "resampling\_resolution"
- Now it is possible to comment partial lines using a hashtag. Example:

```
experiments = cams61_emep_ph2-eu-000#, cams61_monarch_ph2-eu-000
```

- As a major change, it is now possible to **update the aggregation statistic of the timeseries and periodic plots** from the configuration files directly, before it was only possible to change in the dashboard. By default, these are set to be Median. The parameters to change them are: "timeseries\_statistic\_aggregation" and "periodic\_statistic\_aggregation".
- The default statistic aggregation and mode are now **Temporal|Spatial and Median**, in the past we set it to be Flattened.

# Configuration changes

- When an invalid field is encountered in the configuration file, the system now generates a warning.
- The **default** keyword can now be used in the configuration file, which automatically inserts the appropriate default value for that setting. This works the same as not passing the parameter.
- Any metadata in non-GHOST files can now be read as long as it is following GHOST standards (e.g. can be classifications or any other advanced metadata).
- Added new variables so that the default QA/flags fields can be added to and subtracted from, without having to rewrite all the individual flags: **add\_qa**, **add\_flags**, **subtract\_qa**, **subtract\_flags**



# Critical bugs



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Critical bugs

- Exporting the data, the data would not be set as NaN when temporal colocation was active.
- When saving 3-hourly data in netCDF format, the times would not correspond to the 3-hourly resolution.
- We remade the netCDF files of the XVPCA network because the coordinates did not match the actual data.
- There was a bug in the interactive mode for which the variables of the last subsection would be kept in other subsections unless they were overwritten. This could be seen after creating statsummary, heatmap or table plots.
- Also in the interactive mode, big plots would cut due to not having set the tight layout.

# Critical bugs

- For the offline reports, passing the section through the command line would not have any influence on which sections were being generated.
- Non-GHOST data could not be read using the function `get_data` in interactive mode with a permission error.
- Providentia dashboard would crash when changing species for any non-GHOST network.

# Optimisations



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# Optimisations

- Reordered plots to have multispecies plots first in offline reports.
- Converted JSON settings files to YAML (to allow for comments).
- Adapted Providentia to work locally on Mac.
- Temporal colocation will now be applied when no experiments are loaded, across multispecies.
- Added ability to control number of decimals of hover annotations. Can be set by modifying `round_decimal_places`, in `marker_annotate_text` for a plot type in the relevant `plot_characteristics` file.

# Improved wiki documentation

- The wiki section is constantly undergoing updates (thanks mostly to Alba!) and therefore we invite you to take a look at the updates we have made:

<https://earth.bsc.es/gitlab/ac/Providentia/-/wikis/home>

- There are new sections for the download and interpolation modes, multispecies filtering, colocation, and an explanation of the representativity filters.

## Download

### Download

Now it is possible to download data waiting for a job to be allocated for

To download data you just need to

```
./bin/providentia --confi
```

This will get the data that needs to be set in `settings/data_paths` for local

The download mode fetches all the

### Fields

Fields work a bit different in this mode

Parameter	Mandatory in
network	✓
species	×
resolution	×
start_date	✓
end_date	✓

## Interpolation

### Interpolation

Starting from the release of v

These are some details that you

- The old configuration file data.
- The file `defined_experiment_ids` need
- The code that runs the in

Now that you know these points

```
./bin/providentia --c
```

This will generate logs in different

- Submission logs: To give
- Management logs: To info
- Interpolation logs: To get

## Multispecies filtering

### Multispecies filtering

Multispecies filtering refers to the ability to filter current network or by an external one.

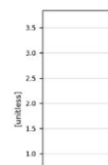
In the reports created to study the dust in the at which values are associated with dust. For instance

If we take a look at the timeseries for one station

**AOD without multispecies filtering**



**AOD with multispecies filtering**



## Colocation

### Colocation

There are two types of colocations

- Temporal colocation
- Spatial colocations

### Temporal colocation

The temporal colocation can be used when turned on, the user has access to

Without temporal colocation:



## Representativity filters

### Representativity filters

One major limitation associated with observations is the amount of gaps often found between measurements.

If these observations are directly compared with model data, this would impose a significant bias upon the comparison.

The representativity filters provides one way to control the temporal robustness of the observations for evaluation.

Providentia has two types of representativity filters. The first is the `representativity_percent` which is used to set the minimum % of representativity needed for a station's measurements to be valid in a specific time period (i.e. daily, monthly, etc.). If for instance we wanted to check the representativity of hourly station data each day, we would use the `daily_representativity_percent` filter. If we set this at 50%, this would mean any daily periods where there are <50% of observations missing, would be set to be NaN. By default, this is always set at 0%.

The second type of representativity filter is the `max_gap_percent`. This is used to set the **maximum** permitted gap allowed for a station's measurements in a specific time period. For example if we wanted to check the max gap of hourly station data each day, we would use the `daily_max_gap_percent` filter. If we set this at 50%, this would mean any daily periods where there is a continuous gap of >50%, would be set to be NaN.

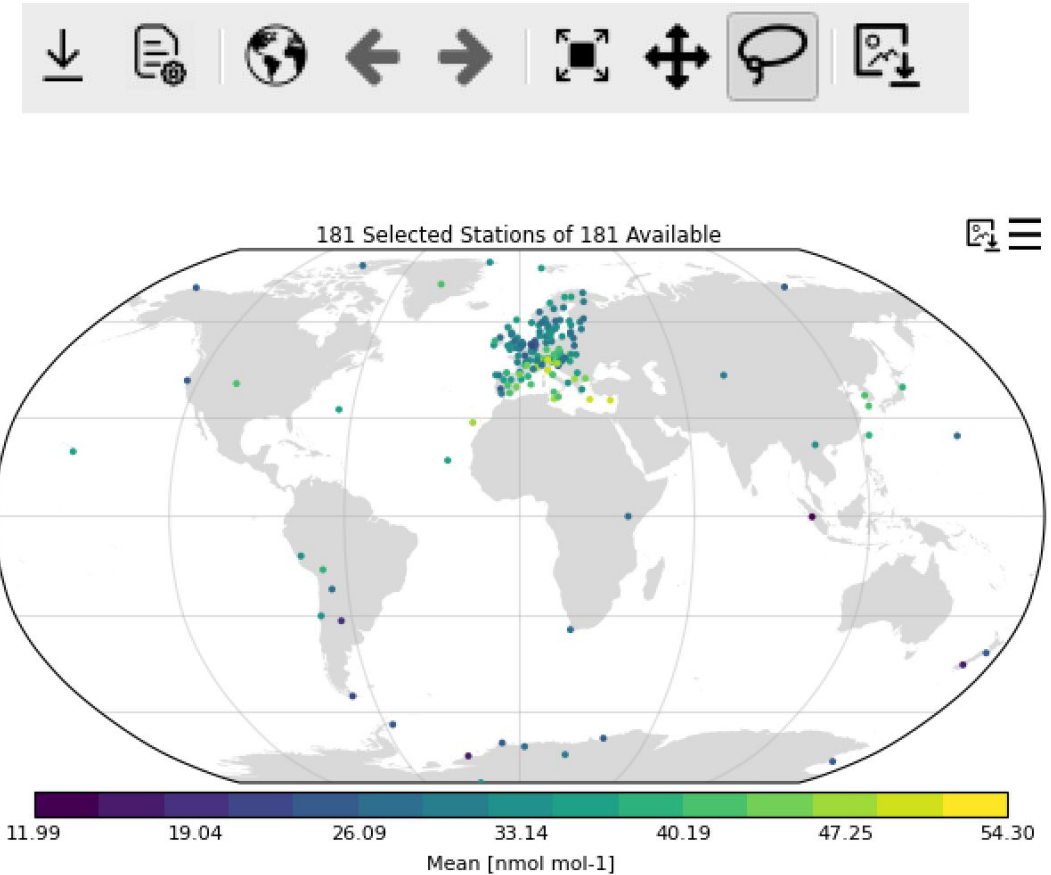
If wanting to apply filters to the entire time period in memory, rather than shorter windows, this can be done using the `all_representativity_percent` and `all_max_gap_percent` filters. Any stations which are completely NaN after filtering are removed from the map.

Whatever the current temporal `resolution` is of the current data in memory, the standard representativity filters will be available for all coarser resolutions, e.g. for hourly resolution data, the standard filters will be available for `daily`, `monthly` and `all` resolutions.

When using GHOST data there are some extra filters available which can be used to assess the representativity of the native resolution data, rather than solely the AC standard averaged temporal resolutions. For example if measurements are natively measured every 10 minutes and these are then averaged to hourly resolution, then the representativity checks will assess the representativity of this native data within the

# Changes to dashboard map lasso

- The map lasso now must be selected on the dashboard toolbar, rather than being enabled by default.
- This significantly improves the speed performance of the map selection as a whole.





# *apply\_filter* method in Interactive mode

- In the previous version of Providentia, in the interactive mode, the only way of filtering by metadata/data variables was to set them in the .conf file and load the .conf.
- There is a new method *apply\_filter* which can now apply specific filters to the loaded data without having to re-load any .conf file.
- Check the Interactive mode wiki page for all the details:

## 🔗 Applying specific filter

If wanting to apply a filter not set in the .conf, this can be done using the following method:

```
provi.apply_filter(field, ...)
```

where `field` is the field to filter by. The fields to filter by can be representativity fields, period fields or metadata fields.

If the field is numeric, lower and upper limits to retain data between can be set as follows:

```
provi.apply_filter(field, lower=28, upper=31)
```

NOTE: it is not mandatory to pass `lower` and `upper` arguments together.

If the field is textual, values to keep and remove associated data with can be set as follows:

```
provi.apply_filter(field, keep='Spain', remove='')
```

If multiple values are wanted to be removed concurrently, the arguments passed should be lists.

```
provi.apply_filter(field, remove=['Spain', 'France'])
```

NOTE: it is not mandatory to pass `keep` and `remove` arguments together.

For the specific case of representativity fields, the argument passed should be `limit`:

```
provi.apply_filter(rep_field, limit=20)
```



# Simplified dashboard representativity filters

- There have been simplifications to the representativity filters on the dashboard to make them more intuitive.
- The maximum gap variables have been removed as they were adding little value.
- **Native:** Representativity of the original data resolution (e.g. it could have been in minutes before averaged to hours)
- **Averaged:** Representativity of the averaged resolution (i.e. how it is stored on esarchive/gpfs)

## Select Minimum Required % Data Representativity

### Native

Hourly

Daily

Monthly

### Averaged

Daily

Monthly

All

# CI/CD pipelines



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*

# CI/CD pipelines

From now on, **every commit** that we do to the master branch will have to pass multiple tests covering **at least 59% of the code statements** and no commit will be approved unless all tests pass. With this, we can catch errors faster and ensure that the results in the reading, filtering and plotting functions do not change with time. We plan to increase the test coverage in the near future and expand it to the other modes (now this only applies to the interactive mode).

Name	Coverage
providentia/__init__.py	100%
providentia/calculate.py	73%
providentia/configuration.py	56%
providentia/fields_menus.py	58%
providentia/filter.py	60%
providentia/interactive.py	66%
providentia/plot.py	52%
providentia/plot_aux.py	57%
providentia/plot_formatting.py	71%
providentia/plot_options.py	61%
providentia/read.py	61%
providentia/read_aux.py	32%
providentia/spatial_colocation.py	43%
providentia/statistics.py	65%
providentia/warnings.py	10%
providentia/writing.py	51%
TOTAL	59%

# Q&A



**Barcelona  
Supercomputing  
Center**

*Centro Nacional de Supercomputación*



**Barcelona  
Supercomputing  
Center**

Centro Nacional de Supercomputación

# Thank you for your attention!

More information at:

<https://earth.bsc.es/gitlab/ac/Providentia>

*Join the #providentia Slack channel!*

[paula.serrano@bsc.es](mailto:paula.serrano@bsc.es) | [alba.vilanova@bsc.es](mailto:alba.vilanova@bsc.es) | [dene.bowdalo@bsc.es](mailto:dene.bowdalo@bsc.es)