1. Introduction

The Match-up Data Base (MDB) approach developed in the framework of the HYPERNETS project aimed at implementing validation analysis of satellite water products using HYPSTAR® data as reference.

The approach is based on the concept of MDB file that was first introduced by EUMETSAT, i.e., a NetCDF file including all the potential match-ups between satellite and in situ data within a time window.

Moreover, it uses a set of open-source tools developed in Python to implement the validation analysis working with the MDB files. These MDB tools are divided into three modules: SAT_EXTRACT, MDB_builder and MDB_reader. All the modules are included in the *hypernets_val* repository available in GitHUB (**https://github.com/HYPERNETS/hypernets_val**).

This remainder of this document is organized as follows: 1) description of the MDB file structure (section 2); and 2) description of the MDB tools, with usage and examples (sections 3-5).

2. MDB file structure

MDB files are based on the open-source NetCDF-4 (Network Common Data Form) file format (https://www.unidata.ucar.edu/software/netcdf), which is built on top of the Hierarchical Data Format version 5 (HDF5) (https://hdfgroup.github.io/hdf5). It supports large, complex and heterogenous data by using a directory-like structure to organize the data within the file. Data in MDB files are stored in variables, *i.e.*, multi-dimensional arrays of values of the same type.

MDB files are built in different stages, adding new data:

a) Satellite extract files. They are generated using the SAT_extract module. They only include satellite data (section 3).

b) MDB files: They are produced using the MDB_builder module. The include satellite and in situ data (section 4).

c) MDB result files (MDBr): They are produced using the MBD_reader module (mode GENERATEMU) and include the match-ups between the satellite and situ data after implementing the quality control (section 5.1).

d) MDB concatenated files (MDBrc): They are generated using the MDB_reader module (mode CONCATENATE). The put together MDB results files adding a set of *flag* variables.

2.1. Satellite extract files

Satellite extract files contain a extract (by default 25 x 25 pixels) of satellite data centered in the in situ location.

These files are generated using the SAT_extract module (section 3). Dimensions, variables and global attributes are summarized in Tables 1-3.

Table 1: Dimensions of satellite extract files

Dimension	**Description**	**Length**
satellite_id	Satellite measurement	Unlimited (1 for each satellite extract file)
satellite_bands	Satellite bands	Depends on sensor/processor
rows	y spatial coordinate	Default: 25
columns	x spatial coordinate	Default: 25

Table 2: Variables included in the satellite extract files.

Variable	Description	Dimensions
satellite_bands	Band wavelengths (nm)	*satellite_bands*
satellite_time	Overpass time	*satellite_id*
satellite_Rrs	Satellite-derived *Rrs*	*satellite_id, satellite_bands, rows, columns*
satellite_latitude	Latitude	*satellite_id, rows, columns*
satellite_longitude	Longitude	
satellite_AOT_0865p50	Aerosol Optical Thickness	
satellite_flag	Flags Data Set	
satellite_OAA	Observation Azimuth Angle	
satellite_OZA	Observation Zenith Angle	
satellite_SAA	Sun Azimuth Angle	
satellite_SZA	Sun Zenith Angle	

Table 3: Global attributes included in the satellite extract files.

Attribute	Description
creation_time	Creation time (YYYY-mm-ddTHH:MM:SSZ)
satellite	Satellite identifier (e.g. S3, S2)
platform	Satellite platform (e.g. A or B for S3 or S2)
sensor	Satellite sensor (e.g. OLCI, MSI)
description	Extract description
resolution	Data resolution (e.g. WFR, 20M)
satellite_aco_processor	Atmospheric correction processor (e.g. STANDARD, C2RCC
satellite_proc_version	Atmospheric correction version
insitu_site_name	In situ site name (e.g. VEIT, BEFR)
insitu_lat	In situ latitude
insitu_lon	In situ longitude

2.2. MDB files

MDB files are generated using the MDB_builder module (section 4) combining satellite data from the extract files (section 3) and in situ data from HYPSTAR® L2 files for the same site and within a time window. Satellite data are directly inherited from the satellite extract files (section 2.1), whereas two new dimensions are required to describe the in situ data (*insitu_id* and *insitu_original_bands*). Note that the current length of *satellite_id* dimension (defined as unlimited) is the number of satellite measurements (i.e. satellite extracts) included in the MDB file, whereas the length of *insitu_id* dimension would be equal to the maximum number of *in situ* measurements (*Rrs* spectra) that could be associated with a specific satellite measurement (by default is set to 40).

Dimensions and variables are summarized in Table 4 and Table 5. Global attributes are inherited from the satellite extract files (Table 3), with an update of the *creation_time* and *description* attributes.

Table 4: Dimensions of the MDB files

Dimension	**Description**	**Length**
satellite_id	Satellite measurements	Unlimited
satellite_bands	Satellite bands	Depends on sensor/processor
rows	y spatial coordinate	Default: 25
columns	x spatial coordinate	Default: 25
insitu_id	In situ measurements	Default: 40
insitu_original_bands	In situ bands	1600 (for HYPSTAR®)

Table 5: Variables included in the MDB files

Variable	Description	Dimensions
	*satellite_* variables (see Table 2)	
insitu_original_bands	Instrument wavelenghts (nm)	*insitu_original_bands*
insitu_time	Measurement time	*satellite_id, insitu_id*
insitu_Rrs	In situ *Rrs*	*satellite_id, insitu_original_bands, insitu_id*
insitu_Rrs_nosc	In situ *Rrs* without correction for the NIR similarity spectrum	
insitu_quality_flag	Quality Flag Dataset	*satellite_id, insitu_id*
insitu_site_flag	Site Flag Dataset	
insitu_viewing_azimuth_angle	Observation Azimuth Angle	
insitu_viewing_zenith_angle	Observation Zenith Angle	
insitu_solar_azimuth_angle	Sun Azimuth Angle	
insitu_solar_zenith_angle	Sun Zenith Angle	
time_difference	Time difference	*satellite_id*

2.3. MDB results (MDBr) files

The MDB result (MDBr) files are generated using the GENERATEMU module of the MDB_reader module (section 5.1). In addition to the satellite and in situ data, it includes the match-ups for each specific wavelength after implementing the quality control.

Dimensions and variables are summarized in Table 6 and Table 7. Global attributes are inherited from the MDB file (Table 3), with an update of the *creation_time* and *description* attributes.

Table 6: Dimensions of the MDBr files

Dimension	**Description**	**Length**
satellite_id	Satellite measurements	Unlimited
satellite_bands	Satellite bands	Depends on sensor/processor
rows	y spatial coordinate	Default: 25
columns	x spatial coordinate	Default: 25
insitu_id	In situ measurements	Default: 40
insitu_original_bands	In situ bands	1600 (for HYPSTAR®)
mu_id	Match-up at wavelength	Unlimited

Match-up data require a new dimension (*mu_id*) defined as unlimited, with a current length equal to the number of satellite measurements (current length of *satellite_id*) by the number of satellite bands (maximum equal to the length of *satellite_bands*) included in the analysis.

Table 7: Variables included in the MDBr files

Variable	Description	Dimensions
satellite_ variables (see Table 2)		
in_situ_ variables (see Table 5)		
mu_ins_rrs	Match-up in situ *Rrs*	*mu_id*
mu_sat_rrs	Match-up satellite *Rrs*	
mu_wavelength	Match-up wavelength	
mu_satellite_id	Match-up *satellite_id*	
mu_valid	Match-up validity	*satellite_id*
mu_insitu_id	Match-up *insitu_id*	
mu_ins_time	Match-up in situ time	
mu_sat_time	Match-up satellite time	
mu_time_diff	Match-up time difference	
time_difference	Time difference	*satellite_id*

2.4. MDB concatenated results (MDBrc) files

The MDB concatenated results (MDBrc) are generated using the mode CONCATENATE of the MDB_reader module (section 5.2.). They combine results from different MDBr files, adding a set of *flag_* variables to identify each match-up. It uses the same dimensions as MDBr files (Table 6). Variables are summarized in Table 8.

Table 8: Variables included in the MDBrc files

Variable	Description	Dimensions
satellite_ variables (see Table 2)		
in_situ_ variables (see Table 5)		
mu_ variables (see Table 7)		
flag_ac	Atmospheric correction	*satellite_id*
flag_site	Site	
flag_satellite	Satellite mission	
flag_sensor	Satellite sensor	
time_difference	Time difference	*satellite_id*

The new flag variables use the following flags included in the global attributes of the MDBr files:

flag_ac: *satellite_aco_processor*

flag_site: *insitu_site_name*

flag_satellite: *satellite + platform*

flag_sensor: *sensor*

A value is assigned to each flag using 2^n with *n* being consecutive numbers staring from 0 (i.e. flag values would be 1, 2, 4, 8, etc).

Global attributes are inherited from the MDBr files (Table 3). In case of attributes used for flagging (*satellite_aco_processor, insitu_site_name, satellite, platform, sensor*), as well as *insitu_lat* and *insitu_lon*, attributed values are updated using a list (comma-separated values) with all the values inherited of the MRDr files included in the concatenation. *creation_time* and *description* attributes are also updated.

3. SAT_extract

The satellite data are provided as satellite extracts created using the tools available in the SAT_EXTRACT module. Different Python extracts tools were created for working with different satellite sensors and/or processors (Table 9).

Table 9: Satellite extract tools implemented in *hypernets_val* repository.

SAT_EXTRACT tool	Satellite	Processor
sat_extract_OLCI	S3A, S3B	STANDARD (WFR)
sat_extract_CMEMS	CMEMS L3 and L4 products	CMEMS
sat_extract_POLYMER	S3A, S3B, S2A, S2B	POLYMER
sat_extract_ACOLITE	S3A, S3B, S2A, S2B	ACOLITE
sat_extract_C2RCC	S3A, S3B, S2A, S2B	C2RCC
sat_extract_NASA	MODIS, VIIRS, VIIRSJ	STANDARD (L2)

The format of the required source files is NetCDF for all the extracts tools except for sat_extract_OLCI, which is based on the Sentinel-3 SAFE format.

All the extract tools are run using a script passing as argument a configuration file with all the parameters and options:

```
$ python sat_extract_OLCI.py -c extract_config.ini -v
```

The tool creates an extract file for each available source image covering the specified in situ location within a temporal range, which can be defined using the start and stop dates or with a date list. Source files can be optionally filtered using a wild card expression.

Output extract files are NetCDF files containing satellite data from a specific product and for an extraction window (by default 25 x 25 pixels) centred on the specified in situ location. It includes a set of attributes defining the site (*insitu_site_name, insitu_lat, insitu_lon*) and satellite product (*satellite, platform, sensor, resolution, satellite_aco_processor, satellite_proc_version*).

Although satellite attributes are retrieved from the satellite source files, values (except for *satellite_aco_processor*) can optionally be established using the configuration file.

Configurations files are organized in three sections: **file_path**, **Time_and_sites_selection** and **satellite_options.**

[file_path]

sat_source_dir: Path to the directory including the satellite source files with a specific format depending on the extract tool. Required.

sat_source_dir_organization: Structure of the source directory in case of files organized in sub-folders indicating the date. It uses YYYY for year, mm for month, dd for day of the month and jjj for the Julian date. For instance, YYYY/jjj or YYYY/mm/dd. Optional.

output_dir: Output folder for the satellite extract files. Required.

tmp_dir: Temporary folder to decompress source files in compressed formats (i.e., zip or tar). Decompressed files are deleted after creating the extract. Optional (*sat_source_dir* is used to decompress source files if this option is not available).

[Time_and_sites_selection]

time_start*:* First date for analysis. Format: YYYY-mm-dd. It used in combination with *time_stop*.

time_stop**:** Last date for analysis. Format: YYYY-mm-dd. It used in combination with *time_start*.

time_list_file**:** Path to a text file including a date list in format YYYY-mm-dd. If this option is given, *time_start* and *time_stop* parameters are not used.

site: Site name. The coordinates of the following WATERHYPERNET sites are already included in the tool: VEIT, GAIT, BEFR, MAFR, M1BE, LPAR. For other sites, please provide latitude and longitude using *insitu_lat* and *insitu_lon*.

insitu_lat*:* Latitude of the site (required if site is not a WATERHYPERNET site)

insitu_lon**:** Longitude of the site (required if site is not a WATERHYPERNET site)

[satellite_options]

extract_size: Size of the extraction window. Optional. Default: 25 pixels.

wce: Wild card expression to filter the source files based on their file name. For instance, S3A* limits the extracts to Sentinel-3A. Optional. Default: None.

BRDF: If *True*, it applies BRDF correction (only for sat_extract_OLCI). Boolean. Default: False. Optional.

satellite: Satellite attribute. Optional.

platform: Platform attribute. Optional.

sensor: Sensor attribute. Optional.

resolution: Resolution attribute. Optional.

satellite_proc_version: Atmospheric correction version. Optional.

Example of satellite extract configuration file:

```
## config file for creating satellite extracts
[file_path]
sat_source_dir: /store3/SAT_EXTRACTS/OLCI/source
output_dir: /store3/SAT_EXTRACTS/OLCI/extracts
tmp_dir: /store3/SAT_EXTRACTS/tmp
sat_source_dir_organization: YYYY/jjj

[Time_and_sites_selection]
time_start: 2019-01-01
time_stop: 2023-03-31
site: BEFR
time_list_file: /store3/SAT_EXTRACTS/configFiles/date_lists/list_BEFR.txt

[satellite_options]
extract_size: 25
BRDF: F
wce:  S3A*
```

4. MDB_builder

The MDB_builder tool assembles in a single MDB file: 1) the satellite data derived from the extract files, and 2) the corresponding HYPSTAR® L2 data available from the specified site within a time window (by default 3 hours) from the satellite acquisition time. The MDB_builder tool was implemented to work with quality assured Level 2 data provided by the HYPERNETS instruments available from the WATERHYPERNET network.

The tool is run using a script passing as argument a configuration file with all the parameters and options:

```
$ python MDB_builder.py -c mdb_builder_config.ini -v
```

A single MDB file must be created for each site, satellite, platform, sensor, resolution and atmospheric correction processor. All these parameters are already defined as global attributes in the satellite extract files.

Therefore, it is recommended to define the values for these attributes in the configuration file, so that the MDB_builder tool can select the correct extract files (extracts with a different value for one of these attributes are skipped). Attribute values which are not defined in the configuration file are extracted from the first analysed extract file. As with satellite source files in the extraction tool, extract files can also be optionally filtered using a wild card expression

In situ HYPSTAR® L2 data are stored as NetCDF files. Each file corresponds to a specific sequence and it contains the reflectance spectra processed using data collected at a specific time interval. By default, two spectra are available (with and without applying the near-infrared correction) and files are organized using folders structured as YYYY/mm/dd/SEQYYYYmmddTHHMMSS.

The configuration file is organized in the following sections: **file_path**, **Time_and_sites_selection**, **insitu_options** and **satellite_options.**

[file_path]

sat_extract_dir: Path to the directory including the satellite extract files. Required.

ins_source_dir: Path to the directory including the in situ files (HYPSTAR). Required.

ins_source_dir_organization: Structure of the in situ directory in case of files organized in sub-folders indicating the date and time (YYYY: year, mm: month; dd: day of the month; jjj Julian date; HH: hour 0-24, MM: minutes; SS: seconds). Default: YYYY/mm/dd/SEQYYYYmmddTHHMMSS.

output_dir: Path to the directory to save the final MDB file. Required.

[Time_and_sites_selection]

time_start: First date for analysis. Format: YYYY-mm-dd. It used in combination with *time_stop*. Default: start and stop times are derived from satellite extract list. Optional.

time_stop: Last date for analysis. Format: YYYY-mm-dd. It used in combination with *time_start*. Default: start and stop times are derived from satellite extract list. Optional.

site: Site attribute. Optional.

[insitu_options]

insitu_type: Type of in situ data. Only *HYPERNETS* is implemented. Required.

level: Level of source files. Only *L2A* is implemented. Required.

n_insitu_id: Maximum number of spectra to be associated with an extract. Default: 50. Optional.

time_max: Maximum time difference between satellite and in situ acquisition, in minutes. Default: 180. Optional.

insitu_bad_spectra_file_list: List of invalid spectral previously defined by the site manager. Each bad spectrum is identified with the date/time using the following format: *site_yyyymmddTHHMMSS*. Bad spectra are flagged as INVALID in the variable *insitu_site_flag*. Optional.

[satellite_options]

satellite: Satellite attribute. Optional.

***platform*:** Platform attribute. Optional.

***sensor*:** Sensor attribute. Optional.

resolution: Resolution attribute. Optional

ac: Atmospheric correction processor attribute (*satellite_aco_processor*). Optional.

wce: Wild card expression to filter the extract files based on their file name. For instance, S3A* limits the extracts to Sentinel-3A. Optional. Default: None.

Output MDB files are saved to the output directory (***output_dir*** option) following the next name convention:

MDB_{satellite}{platform}_{sensor}_{resolution}_{ac}_{start_date}_{stop_date}_{ins_sensor}_{site}.nc

For instance:

MDB_S3A_OLCI_WFR_STANDARD_20210101T000000_20230331T235959_HYPSTAR_BEFR.nc

Example of MDB_builder configuration file:

```
[file_path]
sat_extract_dir: /store3/SAT_EXTRACTS/OLCI/extracts
ins_source_dir:  /store3/HYPERNETS/INSITU_HYPSTAR/BEFR
output_dir:  /store3/MDBs

[Time_and_sites_selection]
time_start: 2021-01-01
time_stop: 2023-03-31
site: BEFR

[insitu_options]
insitu_type: HYPERNETS
level: L2A
n_insitu_id: 50
time_max: 180
insitu_bad_spectra_file_list: /store3/HYPERNETS/INSITU_HYPSTAR/BEFR /bad_spectra.txt
[satellite_options]
satellite: S3
platform: A
sensor: OLCI
resolution: WFR
ac: STANDARD
wce: S3A*
```

5. MDB_reader

The MDB_reader module includes a set of tools for performing the validation analysis starting from the MDB files produced in the step 4. The approach consists of three steps: 1) Match-up generation; 2) Optional concatenation 3) Plot production and metric computation.

5.1. Match-up generation

This step is run using the GENERATEMU mode of MDB_reader tool and a specific MDB file as input. It implements the validation protocols to produce pairs of satellite and in situ *Rrs* for the later plot production and metric computations.

Output data are stored in new variables identified with the *mu_* prefix and are saved in an extended copy of the MDB file. By default, the new file uses the same file name but starting with MDBr instead of MDB. Note that a new dimension *mu_id* is added to identify matchups with a specific wavelength.

The tool is run using a script passing as arguments the input MDB file and a configuration file with the specific quality control options.

```
python MDB_readerV2.py -m GENERATEMU -c config_qc.ini  -i MDB_file.nc -v
```

The quality control options are organized in the configuration file in two sections, **QC_SAT** and **QC_INS**, for implementing the satellite and in situ quality control, respectively.

[QC_SAT]

wllist: List of satellite wavelengths bands. Default: all the satellite bands.

window_size**:** Size *n* of the measurement window (*n* x *n* with *n* being an uneven number), in pixels. Default: 3

min_valid_pixels**:** Minimum number of valid pixels in the measurement window. Default: all (e.g. 9 for 3x3 measurement window).

use_Bailey_Werdell**:** If *True*, the minimum number of valid pixels is set a 50%+1 (excluding land and inland water pixels if they are defined as flags). Boolean. Default: *False*.

stat_value: Quantity to be computed from the measurement window: average (*avg*) or median *(median)*. Default: *avg*.

apply_outliers: If *True*, outliers are excluded before extracting the average or median value. Boolean. Default: *True*.

outliers_info.central_stat**:** Central statistic to compute the outliers: average (*avg*), median (*median*) or percentiles (*percentiles*). Default: *avg*.

outliers_info.dispersion_stat**:** Dispersion statistic to compute the outliers: standard deviation (*std*), median absolute deviation (*mad*) or interquartile range (*iqr*). Default: *std*

outlier_info.factor**:** If ***central_stat*** is *avg* or *median*, factor is used to compute the outliers thresholds as ***central_stat* ± (*factor* x *dispersion_stat*)**. Default: 1.5.

If ***central_stat*** is *percentiles*, factor should be set as a range of percentiles using two comma-separated values (e.g.: 5-95) defining the minimum and maximum thresholds for the outliers.

Masks based on satellite flag variables:

info_flag_x, with ***x*** taking consecutive values starting from 0. It masks all the pixels flagged with one of the flags given in a list (*or* condition). Each flag mask used the following options:

info_flag_x.name*:* Name of the flag band. Required.

info_flag_x.list*:* Flag list (indicating pixels to be flagges. Required

info_flag_x.land: Flag identifying land pixels. Only used if ***use_Bailey_Werdell*** is *True*. Optional.

info_flag_x.inlandwater*:* Flag identifying inland water pixels. Only used if ***use_Bailey_Werdell*** is *True*. Optional.

Masks based on satellite *Rrs* thresholds:

rrs_th_x, with ***x*** taking consecutive values starting from 0. It defines a mask based on a given threshold for specific satellite band(s) selected using a wavelength range. Each threshold mask requires the following options:

rrs_th_x.wl_min: Minimum wavelength to select the satellite band(s).

rrs_th_x.wl_max: Maximum wavelength to select the satellite band(s).

rrs_th_x.th_value*:* Threshold value.

rrs_th_x.th_type**:** Mask type: *lower* (mask pixels with values lower than the threshold) or *greater* (mask pixels with values greater that the threshold).

Macropixel filters based on the satellite *Rrs* variable:

macropixel_filter_rrs_x, with ***x*** taking consecutive values starting from 0. It defines a filter setting as invalid match-ups with *Rrs* statistics computed in the measurement window (micropixel) greater or lower than a given threshold. Each filter requires the following options:

macropixel_filter_rrs_x.wl*:* Wavelength defining the satellite band

macropixel_filter_rrs_x.stat*:* Metric to be computed: *n_values* (number of valid pixels); *avg* (average); *std* (stardard deviation); *median* (median); *min* (minimum); *max* (maximum): or *CV* (variation coefficient).

macropixel_filter_rrs_x.withoutliers*:* Remove outliers in the metric computation (outliers are defined using *outliers_info* options). Boolean.

macropixel_filter_rrs_x.th_value*:* Threshold value.

macropixel_filter_rrs_x.th_type*:* Filter type: *lower* (filter match-ups with metrics lower than the threshold) or *greater* (filter match-ups with metrics greater that the threshold).

Macropixel filter based on other satellite variables:

macropixel_filter_band_x, with ***x*** taking consecutive values starting from 0. It works as ***macropixel_filter_rrs_x***, but using other satellite variables instead of *Rrs* bands (e.g. geometry bands):

macropixel_filter_band_x.band*:* Name of the satellite variable.

macropixel_filter_band_x.stat*:* Same as *macropixel_filter_rrs_x.stat.*

macropixel_filter_band_x.withoutliers*:* Same as *macropixel_filter_rrs_x.withoutliers.*

macropixel_filter_band_x.th_value*:* Same as *macropixel_filter_rrs_x.th_value.*

macropixel_filter_band_x.th_type*:* Same as *macropixel_filter_rrs_x.th_type.*

[QC_INS]

time_diff_max: Maximum difference between the satellite and in situ time measurements in minutes. Default: 120

***apply_nir_correction*:** If *False*, it uses *insitu_Rrs_nosc* instead of *insitu_Rrs* as in situ *Rrs* variable. Boolean. Default: *True*.

Filters based on *Rrs* thresholds:

filter_th_x, with ***x*** taking consecutive values starting from 0. In situ spectra with *Rrs* values lower or greater than the given minimum and maximum thresholds for a specific wavelength range are excluded from the match-up analysis.

filter_th_x.wlmin: Minimum wavelength.

filter_th_x.wlmax: Maximum wavelength.

filter_th_x.thmin: Minimum threshold (-999 for not using it).

filter_th_x.thmax: Maximum threshold (-990 for not using it).

Filters based on in situ flag variables:

info_flag_x with ***x*** taking consecutive values starting from 0. It removes (or keeps) in situ spectra flagged with one of the flags given in a list (*or* condition) in the match-up analysis.

info_flag_x.name_band*:* Name of the in situ flag variable.

info_flag_x.flag_list*:* Flag list (comma-separated values). ALL applies all the flags.

info_flag_x.remove_spectra*:* The filter removes (*True*,) or keeps (*False*) the flagged spectra. Boolean.

Filters based on ranges of other in situ variables:

band_th_x with ***x*** taking consecutive values starting from 0. It removes (or keeps) in situ spectra with a specified values range for a given in situ variable.

***band_th_x.name_band*:** Name of the in situ variable.

***band_th_x.th_type*:** Filter type*: keep* (keep the in situ spectra) or *remove* (remote the in situ spectra).

***band_th_x.th_min*:** Minimum threshold defining the values range.

***band_th_x.th_max*:** Maximum threshold defining the values range.

band_th_x.isangle*:** If *True*, and ***th_min>th_max, two ranges are actually applied: ***th_min*** to 360 and 0 to ***th_max***.

Example of satellite quality control implemented in a configuration file:

```
[QC_SAT]
wllist: 400, 412.5, 442.5, 490, 510, 560, 620, 665, 673.8, 681.3, 708.8, 753.8, 778.8, 865.0, 885.0

window_size: 3

min_valid_pixels: 9

use_Bailey_Werdell: False

stat_value: avg

apply_outliers: True

outliers_info.central_stat: avg
outliers_info.dispersion_stat: std
outliers_info.factor: 1.5

#Flag mask
info_flag_0.name: satellite_WQSF
info_flag_0.flag_list:
LAND,COASTLINE,CLOUD,CLOUD_AMBIGUOUS,CLOUD_MARGIN,INVALID,COSME
TIC,SATURATED,SUSPECT,HISOLZEN,HIGHGLINT,SNOW_ICE,AC_FAIL,WHITECAPS,
RWNEG_O2,RWNEG_O3,RWNEG_O4,RWNEG_O5,RWNEG_O6,RWNEG_O7,RWNEG_O
8
info_flag_0.flag_land: LAND
info_flag_0.flag_inlandwater: INLAND_WATER

#Mask negative values at 400, 442.5 and 442. nm
rrs_th_0.wl_min: 398
rrs_th_0.wl_max: 450
rrs_th_0.th_value: 0
rrs_th_0.th_type: lower

#Spatial homogeneity test: remove mathc-ups with CV>20% at 560 nm
macropixel_filter_rrs_0.wl: 560
macropixel_filter_rrs_0.stat: CV
macropixel_filter_rrs_0.withoutliers: True
macropixel_filter_rrs_0.th_value: 20
macropixel_filter_rrs_0.th_type: greater

#Geometry thresholds: OZA>60, SZA>70
macropixel_filter_band_0.band: satellite_OZA
macropixel_filter_band_0.stat: avg
macropixel_filter_band_0.th_value: 60
macropixel_filter_band_0.th_type: greater

macropixel_filter_band_1.band: satellite_SZA
macropixel_filter_band_1.stat: avg
macropixel_filter_band_1.th_value: 70
macropixel_filter_band_1.th_type: greater
```

Example of satellite quality control implemented in a configuration file:

```
 [QC_INS]
time_diff_max: 120
apply_nir_correction: True

#filter spectra with negative values
filter_th_0.wlmin: 350
filter_th_0.wlmax: 1100
filter_th_0.thmin: 0
filter_th_0.thmax: -999

#flag filter
info_flag_0.name_band: insitu_quality_flag
info_flag_0.flag_list: ALL
info_flag_0.remove_spectra: True

#geometry filter
band_th_0.name_band: insitu_viewing_azimuth_angle
band_th_0.th_type: keep
band_th_0.th_min: 0
band_th_0.th_max: 135
band_th_0.isangle: True
```

5.2. Concatenation

The CONCATENATE mode of the MDB_reader tool creates a new MDB file, putting together results from different MDBr files, i.e. results based on different satellite or sensors, processed with different atmospheric correction algorithms or based on in situ data from different sites.

The match-ups are correctly identified using four new variables (defined using the *flag_* prefix): *flag_ac* (for the processor), *flag_satellite*, *flag_sensor* and *flag_site*, all of them using *satellite_id* as dimension.

The tool is run using a script passing as arguments the input folder containing the MDBr files to be concatenated and the name of the final MDB file:

```
python MDB_reader.py -m CONCATENATE -i folder_with_MDBr  -o  MDBrc_file.nc -v
```

5.3. Metrics and plots

The PLOT mode of the MDB_reader tools implements plot and metric table production using as input MDRr (section 5.1) or MDBrc (section 5.2) files. Options for the different plots are defined using a configuration file, as follows:

```
python MDB_reader.py -m PLOT -i MDBrc_file.nc  -c  config_plot.ini -v
```

In addition to a section with some general options named [GLOBAL_OPTIONS], each section in the configuration file correspond to a plot including all the options for producing it. The type of plot is

defined with the option *type*. Next, we summarize the global options and the options required for each plot.

[GLOBAL_OPTIONS]

output_path: Path in which figures are saved by default as ***output_path/name_section.fig_extension***. Optional, working directory is used if not given.

***fig_extension*:** Figure extension (*png*, *tif* or *jpg*). Default: *tif*.

***fig_resolution*:** Figure resolution. Default: 300.

***mu_valid_variable*:** Variable indicating the valid match-ups: *mu_valid* or *mu_valid_common*. Note that *mu_valid_common* is only relevant with more than one atmospheric correction algorithms and needs to be added previously (section 5.2). Default: *mu_valid*. Optional.

Common option for all the plots:

comment: User comment. Optional.

apply: If *False*, this plot is skipped and not produced. Boolean. Required.

output_file: Output file name. It overrides the output default file name set as: ***output_path/name_section.fig_extension,*** being ***output_path*** and ***fig_extension*** defined in the GLOBAL_OPTIONS section.

Styles

Line styles are implemented using five comma-separated values: *color, marker, markersize, linestyle, linewidth.*

Fill styles are implemented using two comma-separated values: *color, alpha*

5.3.1. Heatmaps with temporal distribution of match-ups (type: *temporalheatmap*)

It produces a heatmap showing the temporal distribution of total or valid match-ups. If a specific flag (e.g. *flag_site*) is not given, it uses the year in the *x* axis and the month in the *y* axis. If a flag in given, year-month is used in the *x* axis and the flag values in the *y* axis.

General options: ***comment, apply***, ***output_file***

***type*:** *temporalheatmap*. Required.

output_type: *total* (all the match-ups); *valid* (only valid match-ups); *valid_common* (only common valid match-ups). More than one option is allowed using comma-separated values (multiple plot is created). Required.

***flag*:** Flag to be used in the *y* axis (*flag_ac*, *flag_satellite*, *flag_sensor* or *flag_site*). Optional.

***flag_list*:** Flag values. Only relevant if *flag* is set. Default: all the available flag values are used. Optional.

***vmin*:** Minimum value. Default: minimum and maximum values are defined automatically. Optional.

***vmax*:** Maximum value. Default: minimum and maximum values are defined automatically. Optional.

If more than one ***output_type*** is defined, options defining multiple plots are also required: ***multiple_plot, xfigsize, yfigsize, widthspace, heightspace*** (see **Multiple plots** section below).

Example of configuration file for a heatmap showing the temporal distribution of flags:

```
[Match-ups_Temporal_Distribution]
comment: temporal distribution of Sentinel-3 match-ups
apply: true
type: temporalheatmap
output_type: total
flag: flag_site
flag_list: BEFR, VEIT, LPAR, MAFR, GAIT, M1BE
```

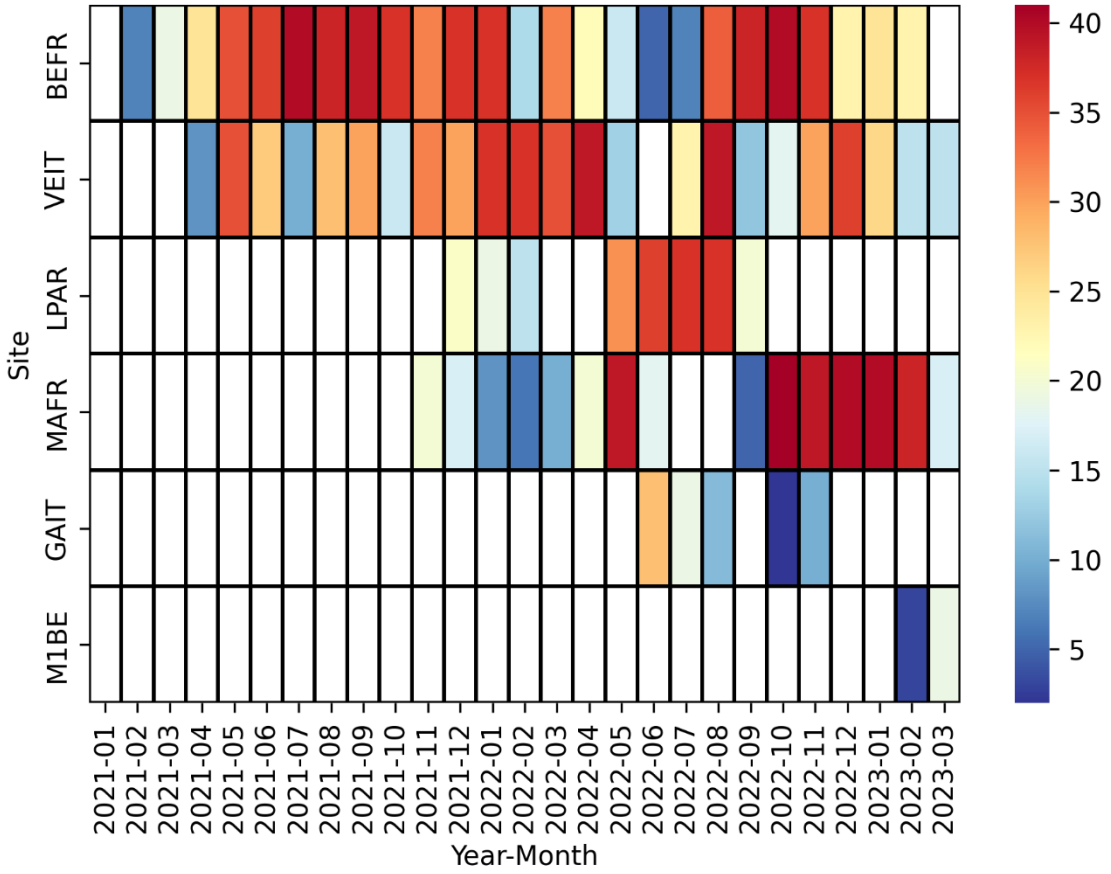

Figure 1: Temporal heatmap showing the temporal distribution of match-ups.

5.3.2. Bar plot with match-ups distribution (type: *validityplot*)

General options: ***comment, apply***, ***output_file***

It produces a bar plot showing the total and/or valid number of match-ups grouped by a specific flag.

type**:** *validityplot*. Required.

output_type**:** *total* (total number of match-ups); *valid* (valid number of match-ups); *total, valid* (both options). Required.

flag**:** Flag to be used to group the number of match-ups (*flag_ac*, *flag_satellite*, *flag_sensor* or *flag_site*). Required.

***flag_list*:** Flag values. Only relevant if *flag* is set. Default: all the available flag values are used. Optional.

series_color*:** Colours (comma-separated values) corresponding to each ***output_type (only used if both ***total*** and ***valid*** are selected). Optional.

series_flag: Names used in the legend if both output types are selected. By default, it uses *Total* and *Valid*. Optional.

show_validity_rates: If *True*, validity rates are included in the *valid* bars (only if *valid* is selected as ***output_type***). Boolean. Default: *True*. Optional.

Example of configuration file for a match-up distribution plot:

```
[Match-up_Distribution]
comment: total and valid number of match-ups per site and validity percent.
apply: true
type: validityplot
output_type: valid, total
flag: flag_site
flag_list: M1BE, GAIT, MAFR, LPAR, VEIT, BEFR
series_color: green, blue
series_flag: Valid,Total
show_validity_rates: True
```

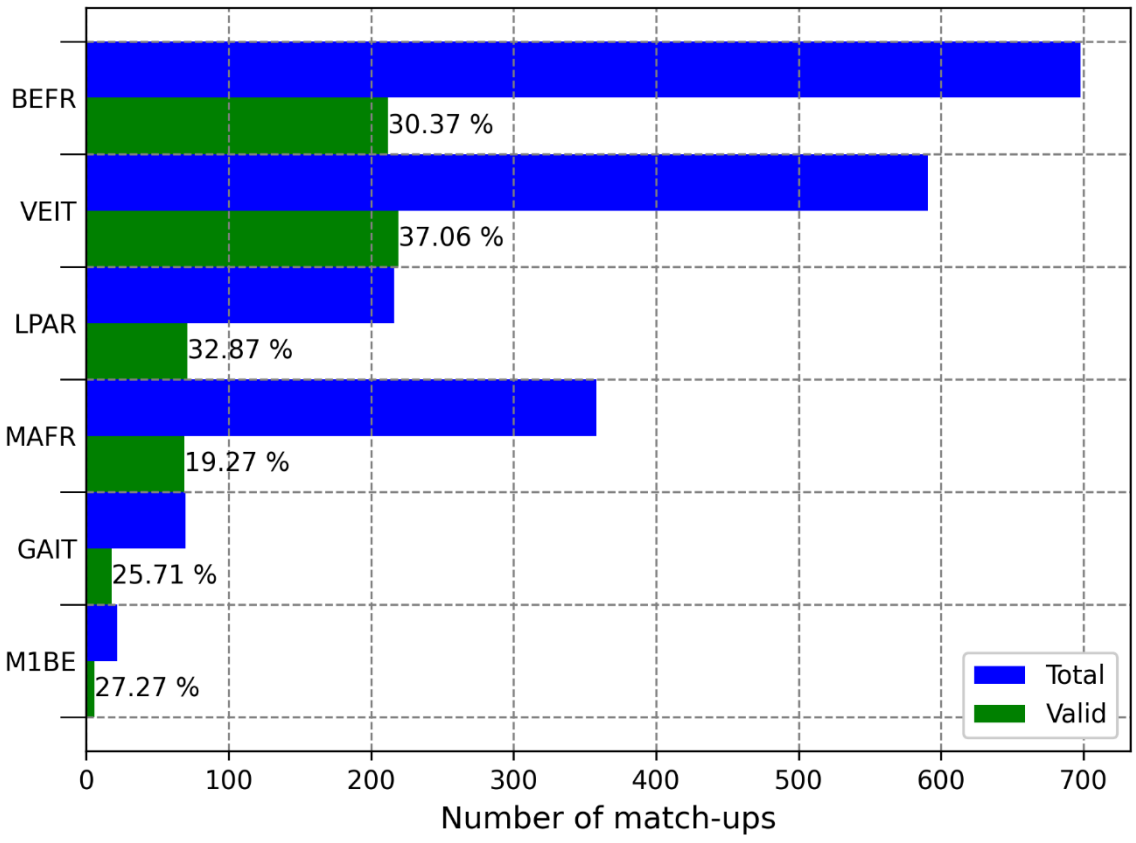

Figure 2: Distribution of total and valid number of match-ups by site, showing the validity rates.

5.3.3. Flag plots (type: *flagplot*)

It produces bar plots showing data related to different satellite flags. Data for each flag could be also grouped in series based on MDBr flags (i.e *flag_ac, flag_satellite, flag_sensor* or *flag_site*).

General options: ***comment, apply, output_file***

***type*:** *flagplot.* Required.

***window_size*:** Size of the window (micropixel) used for the flag analysis. Default: 3. Optional.

***plot_param*:** Parameter to be plotted: *nmacro* (number of flagged macropixels); *pmacro* (percentage of flagged macropixels); *ntotal* (number of flagged pixels); *ptotal* (percentage of flagged pixels). Required.

***n_pixels_macro*:** Minimum number of flagged pixels to consider a measurement window (micropixel) as flagged (and hence plotted using *nmacro* o *pmacro*). Default: 1. Optional.

***n_series*:** Number of series (bars) to be defined for each flag. Default: 1. Optional.

series_names*:** Series names, to be used in the legend. Required if ***n_series>1.

***only_bigger_than_zero*:** If *True*, if the param is equal to zero, the flag is not shown in the plot. Boolean. Default: *False*. Optional.

flag_option_x with ***x*** taking consecutive values starting from 0. It defines the flags shown in the plot using the following options:

flag_option_x.flag_var_name*:* Name of the satellite flag variable. Required.

The specific flag list is required and it can be defined using two options: ***flag_list_and*** or ***flag_list_or***:

flag_option_x.flag_list_and*: and* flag list. It defines a different flag for each flag in the list.

flag_option_x.flag_list_or: *or* flag list, defines a unique flag for all the flag in the list.

flag_option_x.plot_output: Output name of the flag to be shown in the plot. If ***flag_list_and*** is used, an output name must be defined for each flag in the list using comma-separated values (or names in the flag list are directly used in this option is not given). If ***flag_list_or*** is used, this option is required.

flag_option_x.var_group_name: Name of the flag variable (i.e *flag_ac, flag_satellite, flag_sensor* or *flag_site*) used to select the match-ups included in the analysis. Required if ***nseries****>1*.

flag_option_x.var_group_flags: Flag list used to select the math-ups included in the analysis. Required if ***var_group_name*** is set.

flag_option_x.seriesid: Series are defined using integer numbers from 1 to ***n_series***. It could be a single value (flag is applied to the specified series) or a list (comma-separated values) so that each flag defined in ***var_group_flags*** is used for each series (lengths must be the equal).

Example of configuration file for a flag plot:

```
[Flag_WFR_Plot]
comment: flagging analysis. Values organized from bottom to top
apply: true
type: flagplot

flag_option_0.flag_var_name: satellite_WQSF
flag_option_0.flag_list_or: HISOLZEN
flag_option_0.var_group_name: flag_site
flag_option_0.var_group_flags: BEFR, VEIT, LPAR, MAFR, GAIT, M1BE
flag_option_0.seriesid: 1,2,3,4,5,6
flag_option_0.plotoutput: HISOLZEN

flag_option_1.flag_var_name: satellite_WQSF
flag_option_1.flag_list_or: HIGHGLINT
flag_option_1.var_group_name: flag_site
flag_option_1.var_group_flags: BEFR, VEIT, LPAR, MAFR, GAIT, M1BE
flag_option_1.seriesid: 1,2,3,4,5,6
flag_option_1.plotoutput: HIGHGLINT

flag_option_2.flag_var_name: satellite_WQSF
flag_option_2.flag_list_or: INVALID,SUSPECT,AC_FAIL
flag_option_2.var_group_name: flag_site
flag_option_2.var_group_flags: BEFR, VEIT, LPAR, MAFR, GAIT, M1BE
flag_option_2.seriesid: 1,2,3,4,5,6
flag_option_2.plotoutput: INVALID

flag_option_3.flag_var_name: satellite_WQSF
flag_option_3.flag_list_or:
RWNEG_O2,RWNEG_O3,RWNEG_O4,RWNEG_O5,RWNEG_O6,RWNEG_O7,RWNEG_O
8
flag_option_3.var_group_name: flag_site
flag_option_3.var_group_flags: BEFR, VEIT, LPAR, MAFR, GAIT, M1BE
flag_option_3.seriesid: 1,2,3,4,5,6
flag_option_3.plotoutput: RWNEG

flag_option_4.flag_var_name: satellite_WQSF
flag_option_4.flag_list_or: CLOUD,CLOUD_AMBIGUOUS,CLOUD_MARGIN
flag_option_4.var_group_name: flag_site
flag_option_4.var_group_flags: BEFR, VEIT, LPAR, MAFR, GAIT, M1BE
flag_option_4.seriesid: 1,2,3,4,5,6
flag_option_4.plotoutput: CLOUD

nseries: 6
series_names: BEFR, VEIT, LPAR, MAFR, GAIT, M1BE
series_color: blue,red,green,cyan,magenta,orange
window_size: 3
n_pixels_macro: 1
only_bigger_than_zero: False
plot_param: nmacro
```

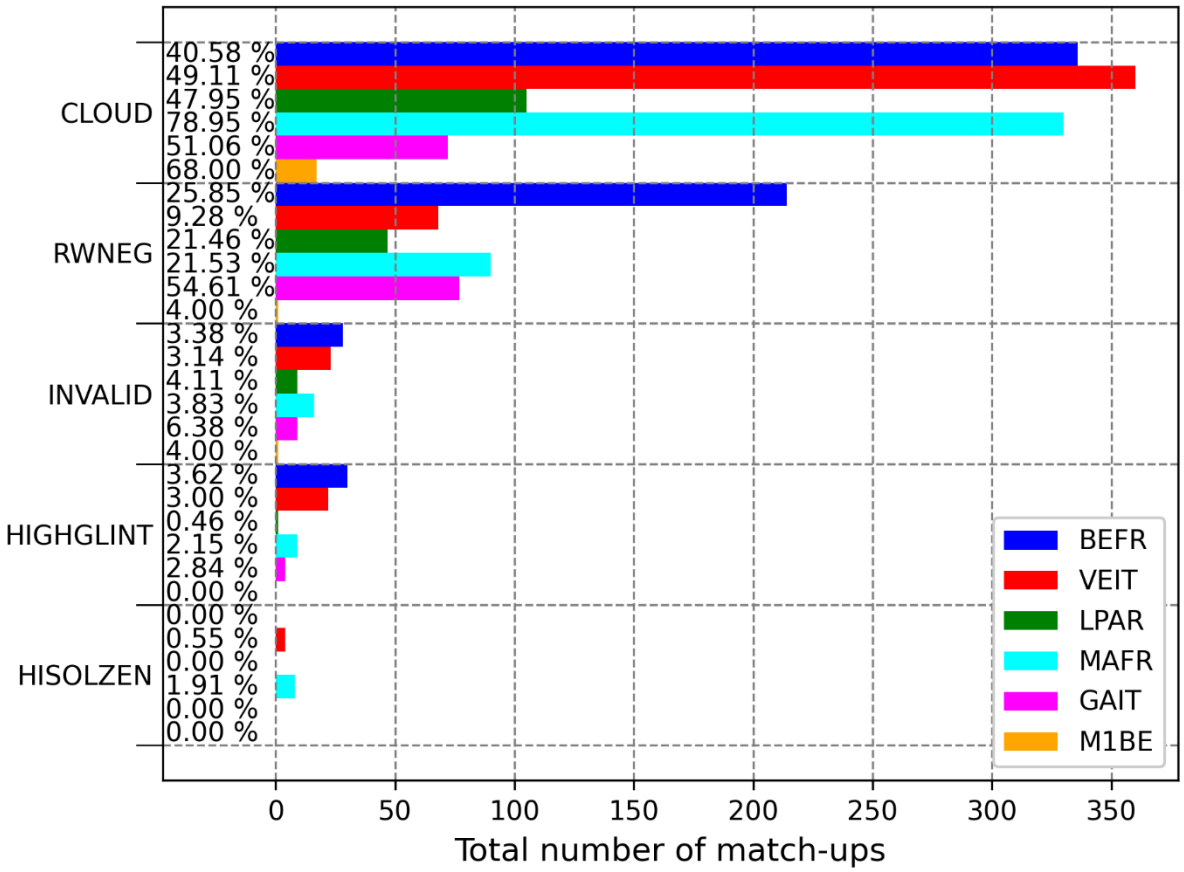

Figure 3: Flag analysis plot

5.3.4. Spectra plots (type: *spectraplot*)

Different types of spectra plots defined with the option ***type_rrs***. Common options include:

***type*:** *spectraplot*

***type_rrs*:** *ins, sat, mu_ins, mu_comparison, comparison_sat_insitu, comparison_sat*

- *ins:* In situ spectra.
- *sat:* Satellite spectra.
- *mu_ins:* In situ spectra for individual match-ups.
- *mu_comparison:* Comparison of in situ and satellite spectra for individual match-ups.
- *comparison_sat_insitu:* comparison of satellite and situ spectra using spectral statistics (central tendency and dispersion).
- *comparsion_sat:* comparison of satellite spectra for different flags *(*i.e., *flag_ac, flag_satellite, flag_sensor* or *flag_site)* using spectral statistics (central tendency and dispersion).

***wl_min*:** Minimum wavelength (in nm). Default: absolute minimum wavelength. Optional.

wl_max: Maximum wavelength (in nm). Default: absolute maximum wavelength. Optional.

***y_min*:** Minimum *Rrs* or *Rhow* (considering scale factor). Default: absolute minimum *Rrs*. Optional.

***y_max*:** Maximum *Rrs* or *Rhow* (considering scale factor). Default: absolute maximum *Rrs*. Optional.

***scale_factor*:** Scale factor to be applied to the data (only multiple of 10). Default: 1000 for *Rrs*, 1 for ρ_w (use_rhow: T). Optional.

***use_rhow*:** If True (T) *Rrs* is converted to *Rhow* (*Rhow* = *Rrs* * PI). Default: False (F). Optional.

***xlabel*:** *x* axis title. Default: Wavelength (nm). Opcional.

***ylabel*:** *y* axis title. Defaults: In situ *Rrs* (sr^{-1}) if *type_rrs* is *ins* or *mu_ins*; Satellite *Rrs* (sr^{-1}) if *type_rrs* is *sat* or *mu_sat*; *Rrs* (sr^{-1}) otherwise. If use_rhow is True, *Rrs*(sr-1) is replaced by ρ_w. Scale factor is also included in the default labels. Optional.

***title*:** Figure title. Default: None. Optional.

In situ spectra plots (type: *spectraplot*, type_rrs: *ins*)

Plots showing in situ spectra for the whole dataset.

General options: ***comment, apply, output_file***

type: *spectraplot*

***type_rrs*:** *ins*

Common *spectraplot* options: ***wl_min, wl_max, y_min, y_max, scale_factor, use_rhow, xlabel, ylabel, title***

***plot_spectra*:** *none, all* (all the spectra), *valid* (only valid spectra), *invalid* (only invalid spectra), *selected* (only selected spectra for valid match-ups). Options can be used together (as comma-separated values) as they are plotted using different line style.

***plot_stats*:** If True, central and dispersion spectra computed using *stat_plot_method* are also plot. Default: True. Boolean. Optional.

***stat_plot_method*:** *iqr* (median ± interquartile range); *std, factor* (average ± (factor x standard deviation). Default: *iqr*. It is only relevant if *plot_stats* is True. Statistics are always computed using valid spectra.

all_line_style: Line style for all the spectra. Default: *black, none, 0, solid,1*. Optional.

valid_line_style: Line style the valid spectra. Default: *green, none, 0, solid,1*. Optional.

invalid_line_style: Line style for the invalid spectra. Default: *red, none, 0, solid,1*. Optional.

selected_line_style: Line style for the selected. Default: *blue, none, 0, solid,1*. Optional.

central_style: Line style of the central spectrum (average or median, depending on *stat_plot_method*). Default: *black, o, 5, solid,1.5.* Optional.

dispersion_style: Line style of the dispersion spectra (average or median, depending on *stat_plot_method*). Default: *black, None, 0, dashed,0*

fill_style: Fill style for the dispersion area. Default: *grey, 0.5*

Satellite spectra plots (type: *spectraplot*, type_rrs: *sat*)

Plots showing satellite spectra for the whole dataset.

General options: ***comment, apply, output_file***

type: *spectraplot*

***type_rrs*:** *sat*

Common *spectraplot* options: ***wl_min, wl_max, y_min, y_max, scale_factor, use_rhow, xlabel, ylabel, title***

***plot_spectra*:** If True, spectra are plotted. Default: True. Boolean. Optional.

***plot_stats*:** If True, central and dispersion spectra computed using *stat_plot_method* are also plotted. Default: True. Boolean. Optional.

***stat_plot_method*:** *iqr* (median ± interquartile range); *std, factor* (average ± (factor x standard deviation). Default: *iqr*. It is only relevant if *plot_stats* is True. Statistics are always computed using valid spectra.

line_style: Line style for all the spectra. Default: *black, none, 0, solid,1*. Optional.

central_style: Line style of the central spectrum (average or median, depending on *stat_plot_method*). Default: *black, o, 5, solid,1.5.* Optional.

dispersion_style: Line style of the dispersion spectra (average or median, depending on *stat_plot_method*). Default: *black, None, 0, dashed,0*

fill_style: Fill style for the dispersion area. Default: *grey, 0.5*

In situ spectra plots for individual match-ups (type: *spectraplot*, type_rrs: *mu_ins*)

Single plot for each match-up including in situ spectra. It uses the same options as ***type_rrs***: *ins*.

General options: ***comment, apply, output_file***

type: *spectraplot*

***type_rrs*:** *mu_ins*

Common *spectraplot* options: ***wl_min, wl_max, y_min, y_max, scale_factor, use_rhow, xlabel, ylabel, title***

ins options: ***plot_spectra***, ***plot_stats***, ***stats_plot_method***, ***all_line_style, valid_line_style, invalid_line_style, selected_line_style***

mu_range: *mu_start, mu_stop.* Match-ups indexes range limiting the plots. Optional.

mu_list: Match-up indexes list (comma-separated values) limiting the plots. Optional.

Example of configuration file for plotting in situ spectra for a specific match-up:

```
[Insitu_Spectra_MU_Plot]
comment: in situ spectra for a match-up
apply: true
type: spectraplot
type_rrs: mu_ins
scale_factor: 1000
ymin: -4
ymax: 10
mu_list: 5
```

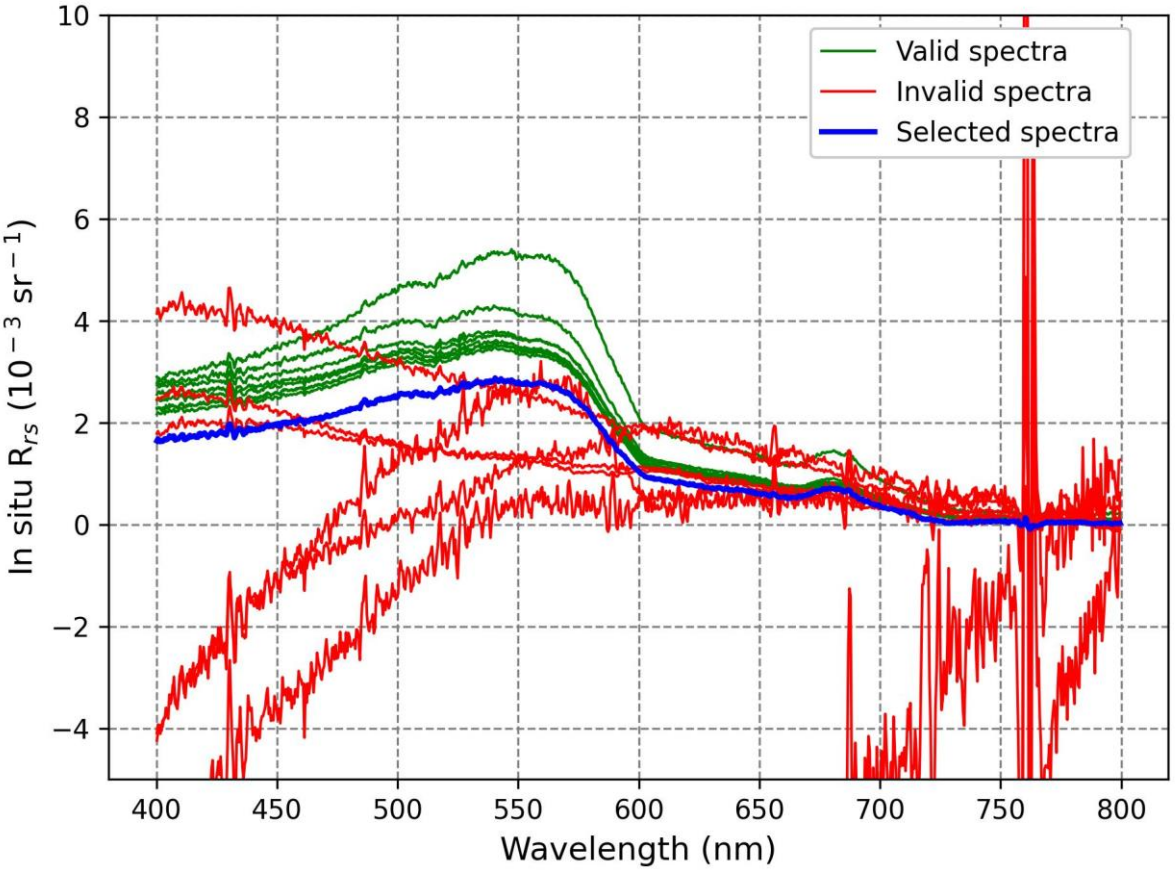

Figure 4: In situ spectra for a specific match-up.

Comparison of satellite and in situ spectra for individual match-ups (type: *spectraplot*, type_rrs: *mu_comparison*).

Single plot for each match-up showing the in situ and satellite spectra.

General options: ***comment, apply, output_file***

type: *spectraplot*

type_rrs**:** *mu_comparison*

Common *spectraplot* options: ***wl_min, wl_max, y_min, y_max, scale_factor, use_rhow, xlabel, ylabel, title***

insitu_line_style: Line style for in situ spectrum. Default: *red, ., 10, solid,1*. Optional.

sat_line_style: Line style for the satellite spectrum. Default: *blue, ., 10, solid,1*. Optional

mu_range: *mu_start, mu_stop*. Match-ups indexes range limiting the plots. Optional.

mu_list: Match-up indexes list (comma-separated values) limiting the plots. Optional.

Example of configuration file for plotting the satellite and in situ spectra for a specific match-up:

```
[Sat_Insitu_Mu_Comparison]
comment: comparison of satellite and in situ spectra for specific match-ups
apply: True
type: spectraplot
type_rrs: mu_comparison
mu_list: 5
```

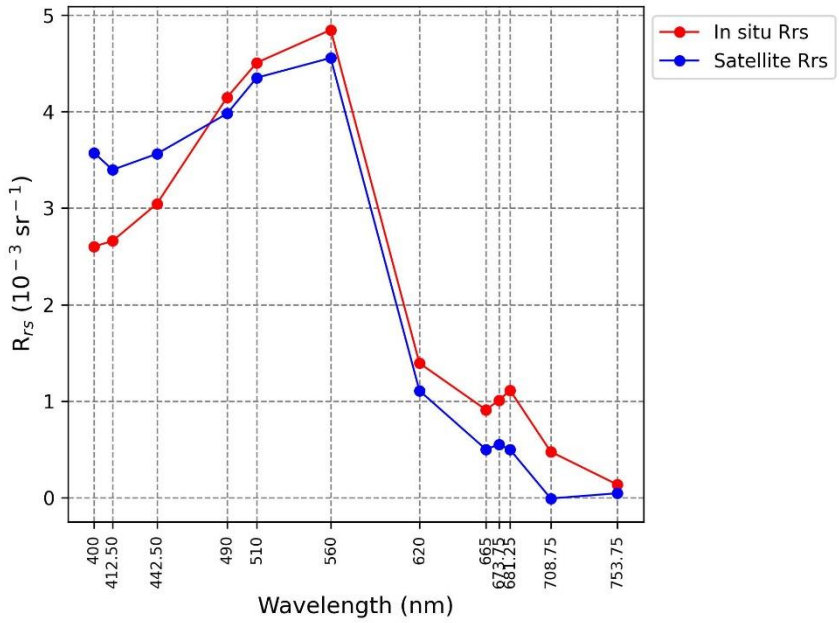

Figure 5: Comparison of satellite and in situ spectra for a specific match-up

Satellite and in situ spectra stats comparison (*type*: *spectraplot, type_rrs*: *comparison_sat_insitu*).

It produces a plot comparing the in situ and satellite spectral statistics (central tendency and dispersion). The analysis could include all the spectra in the MDBrc file or only the spectra associated with a specific flag (producing a plot for each flag) using ***selectBy***. Single plots can be included in a parent plot using the multiple plot options.

General options: ***comment, apply, output_file***

***type*:** *spectraplot*

***type_rrs*:** *comparison_sat_insitu*

Common *spectraplot* options: ***wl_min, wl_max, y_min, y_max, scale_factor, use_rhow, xlabel, ylabel, title***

***stat_plot_method*:** *iqr* (median ± interquartile range); *std*, *factor* (average ± (factor x standard deviation). Required.

***insitu_central_style*:** Line style of the central in situ spectrum (average or median, depending on *stat_plot_method*). Default: *red, o, 5, solid,1*. Optional.

***sat_central_style*:** Line style of the central satellite spectrum (average or median, depending on *stat_plot_method*), Default: *blue, o, 5, solid,1*. Optional.

***insitu_dispersion_style*:** Line style of the in situ dispersion spectra (interquartile range or standard deviation, depending on *stat_plot_method*). Default: *red, o, 0, dashed,0*. Optional.

***sat_dispersion_style*:** Line style of the satellite dispersion spectra (interquartile range or standard deviation, depending on *stat_plot_method*). Default: *blue, o, 0, dashed,0*. Optional.

insitu_fill_style: Fill style for the in situ dispersion area. Default: *red, 0.5*. Optional.

sat_fill_style: Fill style for the satellite dispersion area. Default: *blue, 0.5*. Optional.

***selectBy*:** Name of the flag variable (i.e *flag_ac, flag_satellite, flag_sensor* or *flag_site*) used to select the match-ups included in the analysis, creating a different plot for each flag. Optional.

***selectValues*:** Flag list (comma-separated values) used with *selectBy*. Default: All the flags. Optional.

Multiple plots: In case of using ***selectBy***, each individual plot is saved to a single file. To produce a parent plot including all these individual plots, options defining multiple plots are required: ***multiple_plot, xfigsize, yfigsize, widthspace, heightspace*** (see **Multiple plots** section below).

Example of configuration file for plotting a comparison of the satellite and in situ spectral characteristics

```
[Sat_Insitu_Stats_Comparison]
comment: spectra comparison satellite-in and situ
apply: true
type: spectraplot
type_rrs: comparison_sat_insitu
stat_plot_method: iqr
selectBy: flag_site
selectValues: BEFR
```

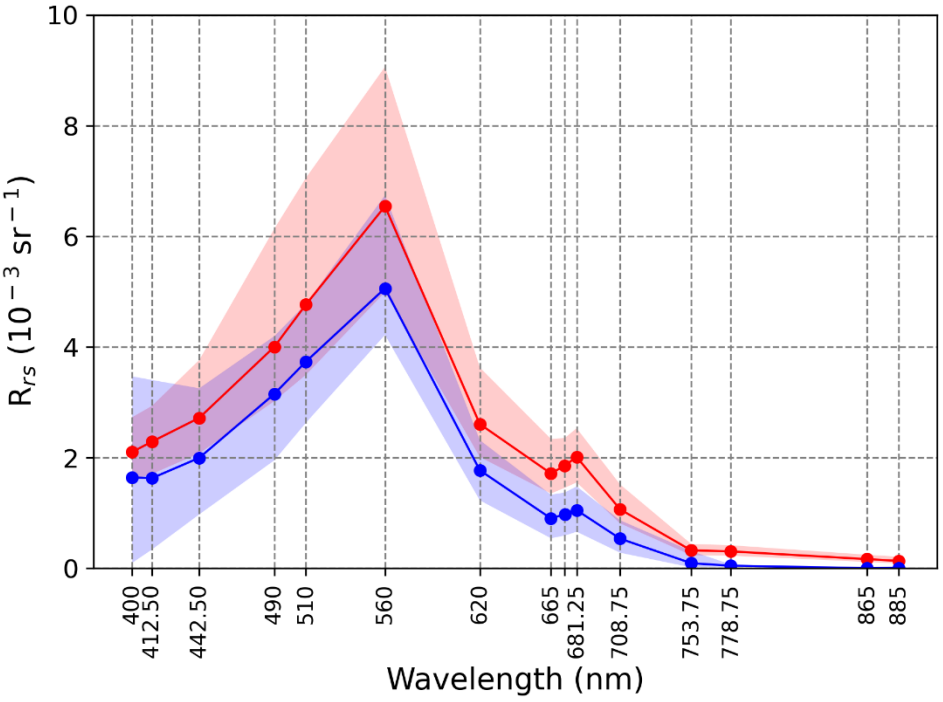

Figure 6: Comparison of satellite and in situ spectral statistics

Satellite spectra stats comparison (*type*: *spectraplot, type_rrs*: *comparison_sat*).

It produces a plot comparing the satellite spectral statistics (central tendency and dispersion) for the flags (based on *flag_ac, flag_satellite, flag_sensor* or *flag_site*) specified using ***groupBy***.

General options: ***comment, apply, output_file***

***type*:** *spectraplot*

***type_rrs*:** *comparison_sat*

Common *spectraplot* options: ***wl_min, wl_max, y_min, y_max, scale_factor, use_rhow, xlabel, ylabel, title***

***stat_plot_method*:** *iqr* (median ± interquartile range); *std, factor* (average ± (factor x standard deviation). Required.

groupBy**:* Name of the flag variable (i.e *flag_ac, flag_satellite, flag_sensor* or *flag_site*) used to group the satellite spectra. Statistical spectra from different groups are plotted with different colours (defined in ***series_colors) using the given line styles (***central_style*** and ***dispersion_style*)**. Required.

groupValues: Flag list (comma-separated values) used with *groupBy*. Default: All the flags. Optional.

***series_colors**:* Comma-separated list of colours, indicating the colours used to plot each *flag* (overriding the colour defined in **central_style** and **dispersion_style**). Default: flag default colours. Optional.

central_style*:** Line style of the central spectra (average or median, depending on *stat_plot_method*). Colours are overridden using ***series_colors. Default: *black, o, 7, solid,2*. Optional.

dispersion_style*:** Line style of the dispersion spectra (interquartile range or standard deviation, depending on ***stat_plot_method), Default: *black, none, 0, dashed, 1*. Optional.

plot_insitu: *flag or ALL*. If this option is given, in situ spectral statistics are also plotted using data only for the given *flag* or the complete datasets (option *ALL*). Style is defined using ***insitu_central_style, insitu_dispersion_style and insitu_fill_style***. Optional.

***insitu_central_style*:** Line style of the central in situ spectrum (average or median, depending on *stat_plot_method*). Default: *black, o, 5, solid,1*. Optional.

***insitu_dispersion_style*:** Line style of the in situ dispersion spectra (interquartile range or standard deviation, depending on *stat_plot_method*). Default: *black, none, 0, dashed,0*. Optional.

insitu_fill_style: Fill style for the in situ dispersion area. Default: *black, 0.5*. Optional.

***selectBy*:** Name of the flag variable (i.e *flag_ac*, *flag_satellite*, *flag_sensor* or *flag_site*) used to select the match-ups included in the analysis, creating a different plot for each flag. Optional.

***selectValues*:** Flag list (comma-separated values) used with *selectBy*. Default: All the flags. Optional.

Multiple plots: In case of using ***selectBy***, each individual plot is saved to a single file. To produce a parent plot including all these individual plots, options defining multiple plots are required: ***multiple_plot, xfigsize, yfigsize, widthspace, heightspace*** (see **Multiple plots** section below).

Example of configuration file for plotting a comparison of satellite spectral characteristics for different flags:

```
[sat_comparison]
commment: Comparison ACOLITE + C2RCC sat spectra
apply: true
type: spectraplot
type_rrs: comparison_sat
groupBy: flag_ac
groupValues: ACOLITE, C2RCC
color: blue,red
selectBy: flag_site
selectValues: BEFR
stat_plot_method: iqr
plot_insitu: ACOLITE
```

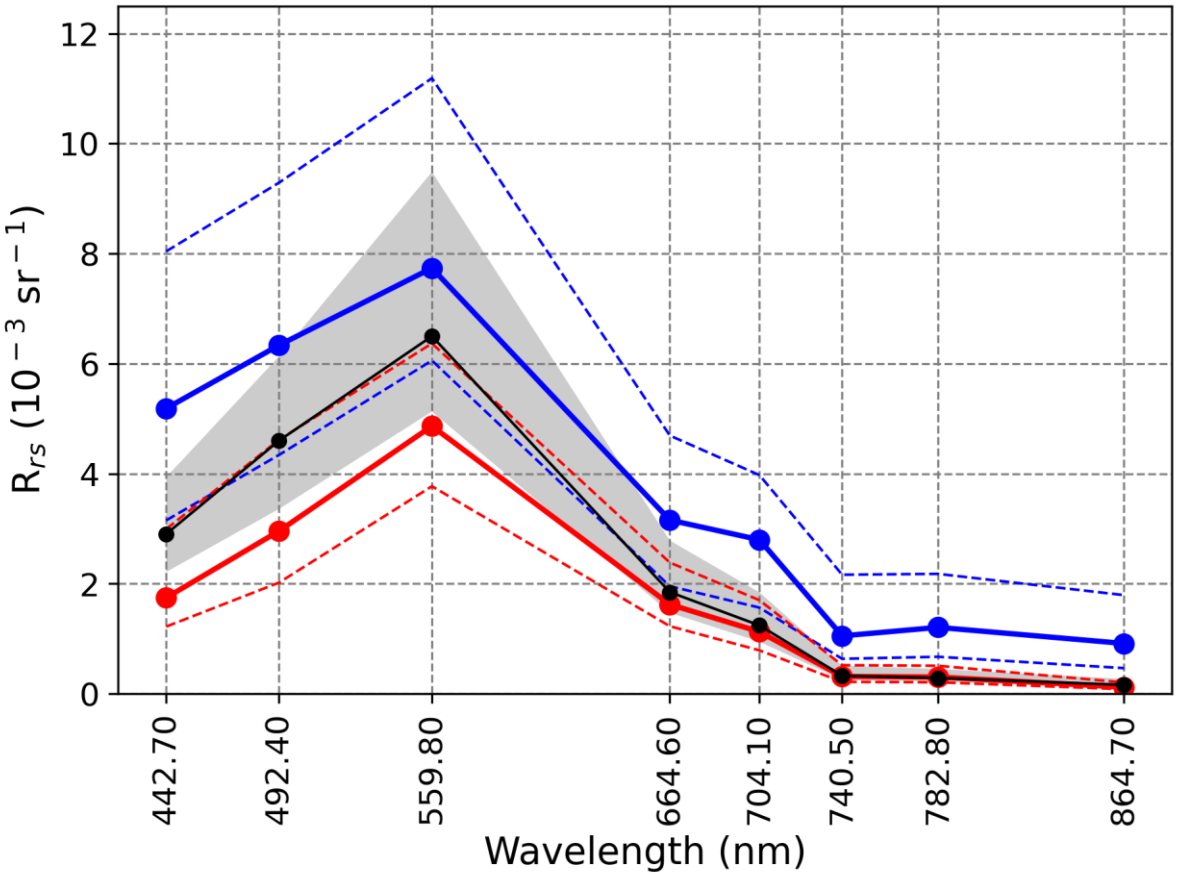

Figure 7: Comparison of satellite spectral statistics.

5.3.5. Scatter plots (type: scatterplot)

Scatter plots of satellite *Rrs* (y axis) versus in situ *Rrs* (x axis). Common options:

***type*:** *scatterplot*

***use_rhow**:* If True (T) *Rrs* is converted to *Rhow* (*Rhow* = *Rrs* * PI). Default: False (F). Optional.

log_scale: If True (T), values are log10-transformed. Default: False(F). Boolean. Optional.

***scale_factor*:** Scale factor to be applied to the data (only multiple of 10). Default: 1000 for *Rrs*, 1 for ρ_w (use_rhow: T). Optional.

title: Figure title. Default: None. Optional.

legend: If True(T), legend is shown in the figure (only applicable with ***groupBy*** or plots including all the wavelengths). Default: True. Optional.

Data points style:

marker: Marker. Default: o. Optional.

markersize: Marker size. Default: 20. Optional.

color: Marker color. Default: black. Note that option is override if data points are coloured by flag (with option ***groupBy***), wavelengths (if ***apply_wavelenght_color*** is True) o density (if ***apply_density*** is True).

***edgecolor*:** Marker edge colour (if applicable to the marker). Default: grey. Optional.

linewidth: Marker edge line width (if applicable to the marker). Default: 0.1. Optional.

Axis style:

min_xy*:* Minimum *Rrs* or *Rhow* (considering ***scale_factor***). Default: it is automatically set based on data. Optional.

max_xy*:* Maximum *Rrs* or *Rhow* (considering ***scale_factor***). Default: it is automatically set based on data. Optional.

ticks: Tick lists (comma-separated values), considering ***scale_factor***. Default: ticks are automatically set based on data. Optional.

xlabel: *x* axis title. Default: In situ *Rrs* (sr^{-1}). If ***use_rhow*** is True, *Rrs*(sr^{-1}) is replaced by ρ_w. Scale factor is also included in the default label. Optional.

ylabel: *y* axis title. Default: Satellite *Rrs* (sr^{-1}). If ***use_rhow*** is True, *Rrs*(sr^{-1}) is replaced by ρ_w. Scale factor is also included in the default label. Optional.

Metrics:

include_stats: If True(T), metrics are included in the plot. Default: False. Boolean. Optinal.

stat_list: Statistics list. Required if ***include_stats*** option is True. Values: N, NMATCH-UPS, WL, BIAS, RMSD, RPD, APD, r2.

stats_xpos: Relative x position of the stats text box. Only used if ***include_stats*** option is True. Default: 0.05. Optional.

stats_ypos: Relative x position of the stats text box. ***include_stats*** option is True. Default: 0.75. Optional.

units: Units to be added to RMSD or BIAS metrics. Default: none. Optional.

Identity line:

identity_line: If True(T), identity (y=x) line is plotted. Default. True. Boolean. Optional.

identity_line_stye: Style of the identity line. Default: black, none,0, dashed, 0.75. Optional.

Regression line:

regression_line: If True(T), regression line based on all the data points is plotted. Default. True. Boolean. Optional.

regression_line_style: Style of the regression line. Default: black, none, 0, solid, 1. Optional.

Data selection:

The scatter plots can include all the match-ups in the MDBrc file or only data associated with a specific flag (producing a plot for each flag) using ***selectBy*.** Analysis can also be limited to a set of wavelengths using ***wlvalues***.

***selectBy*:** Name of the flag variable (i.e *flag_ac*, *flag_satellite*, *flag_sensor* or *flag_site*) used to select the match-ups included in the analysis, creating a different plot for each flag. Optional.

***selectValues*:** Flag list (comma-separated values) used with *selectBy*. Default: All the flags. Optional.

wlvalues: Wavelength list (comma-separated values) to be included in the scatter plot. Default: all the available wavelengths. Optional.

Different types of scatter plots are produced using the options ***selectByWavelength*** and ***groupBy***.

Global scatterplots including all the wavelengths without groups (*selectByWavelength*: False, without *groupBy*)

Global scatter plot including all the wavelengths. Data points can be coloured in three ways by setting the values for the options ***apply_wavelenght_color*** and ***apply_density***: using a colour for each wavelength**,** using density or using the default style**.** Note that data can optionally be limited using ***wlvalues*** and ***selectBy*** (creating a plot for each flag).

General options: ***comment, apply, output_file***

Scatter plot options: ***use_rhow, log_scale, scale_factor, title, marker, markersize, color, edgecolor, linewidth, min_xy, max_xy, ticks, xlabel, ylabel, include_stats, stat_list, stat_xpos, stat_ypos, units, legend, identity_line, identity_line_style, regression_line, regression_line_style, selectBy, selectValues, wlvalues.***

type: *scatterplot*

selectByWavelength: *False*

apply_wavelenght_color: If True(T), data points are coloured by wavelength. Legend could be added using ***legend:*** *True*. Note that legend labels are set automatically and cannot be overridden using ***legend_values***. Default: True (T). Optional.

apply_density*:** If True, data points are coloured by density. If False, the default marker colour is used. Only used if ***apply_wavelenght_color is False**.** Default: True. Optional.

Wavelength scatterplots without groups (*selectByWavelength*: True, without *groupBy*)

A scatterplot is produced for each wavelength (wavelengths can be limited using ***wlvalues***) and saved to an individual file. To produce a parent plot including all these individual scatter plot, options defining multiple plots are required: ***multiple_plot, xfigsize, yfigsize, widthspace, heightspace*** (see **Multiple plots** section below). Colours applied to the data points are managed using ***apply_wavelenght_color*** and ***apply_density*** options**.**

General options: ***comment, apply, output_file***

Scatter plot options: ***use_rhow, log_scale, scale_factor, title, marker, markersize, color, edgecolor, linewidth, min_xy, max_xy, ticks, xlabel, ylabel, include_stats, stat_list, stat_xpos, stat_ypos, units, identity_line, identity_line_style, regression_line, regression_line_style, selectBy, selectValues, wlvalues.***

type: *scatterplot*

selectByWavelength: *False*

apply_wavelenght_color: If True(T), data points are coloured by wavelength. Default: False. Optional.

apply_density*:** If True, data points are coloured by density. If False, the default marker colour is used. Only used if ***apply_wavelenght_color is False**.** Default: True. Optional.

individual_axis: If True, individual axis labels and ticks are applied to each singe plot. If False, *x* and *y* axis labels and ticks are only shown in the last row and first column, respectively. It is only applied when multiple plots are used. Boolean. Default: True. Optional.

Global scatterplots including all the wavelengths with groups (selectByWavelength: False, with groupBy)

Global scatter plot including all the wavelengths with the data points coloured by groups. Groups are established using ***groupBy*** and ***groupValues*** (optional)**.** Note that data can optionally be limited using ***wlvalues*** and ***selectBy*** (creating a plot for each flag).

General options: ***comment, apply, output_file***

Scatter plot options: ***use_rhow, log_scale, scale_factor, title, marker, markersize, color, edgecolor, linewidth, min_xy, max_xy, ticks, xlabel, ylabel, include_stats, stat_list, stat_xpos, stat_ypos, units, legend, identity_line, identity_line_style, regression_line, regression_line_style, selectBy, selectValues, wlvalues, legend, legend_values.***

type: *scatterplot*

selectByWavelength: *False*

groupBy*:* Name of the flag variable (i.e *flag_ac*, *flag_satellite*, *flag_sensor* or *flag_site*) used to group the match-ups. Data points from different groups are plotted with a different colour/style. Required to group data.

groupValues: Flag list (comma-separated values) used with *groupBy*. Default: All the flags. Optional.

color: Marker colours for each group, defined as a comma-separated list. Default: flag colour list. Optional.

regression_line_groups: If True(T), a regression line based on the data points for each flag is plotted.

regression_line_groups_style: Style of the regression lines for each group. Colours are overridden by the option ***color*** using a different colour for each group. Default: black, none, 0, solid, 1**.** Optional.

In addition to ***color***, a specific style for each group can be optionally defined using comma-separated values for the following options: ***marker, markersize, edgecolor*** and ***linewidth***.

legend_values*:** Legend labels (comma-separated list). Only used if ***legend is True**.** Default: legend labels are set automatically equal to ***groupValues***. Optional.

Example of configuration file for plotting global scatter plots:

```
[Global_Scatter_Plot_By_Mission]
comment: scatterplot combining S3A and S3B
apply: false
type: scatterplot
selectByWavelength: false
groupBy: flag_satellite
groupValues: S3A, S3B
color: blue, red
min_xy: 0
max_xy: 0.20
ticks: 0.05, 0.1, 0.15, 0.20
markersize: 25
include_stats: true
stat_list: NMATCH-UPS, RMSD, APD, RPD, r2, BIAS
legend: false
wlvalues: 400, 412.5, 442.5, 490, 510, 560, 620, 665, 673.8, 681.3, 708.8, 753.8, 778.8, 865.0, 885.0
xlabel: $\rho$$_w$ HYPSTAR$R$
ylabel: $\rho$$_w$ OLCI
use_rhow: True
```

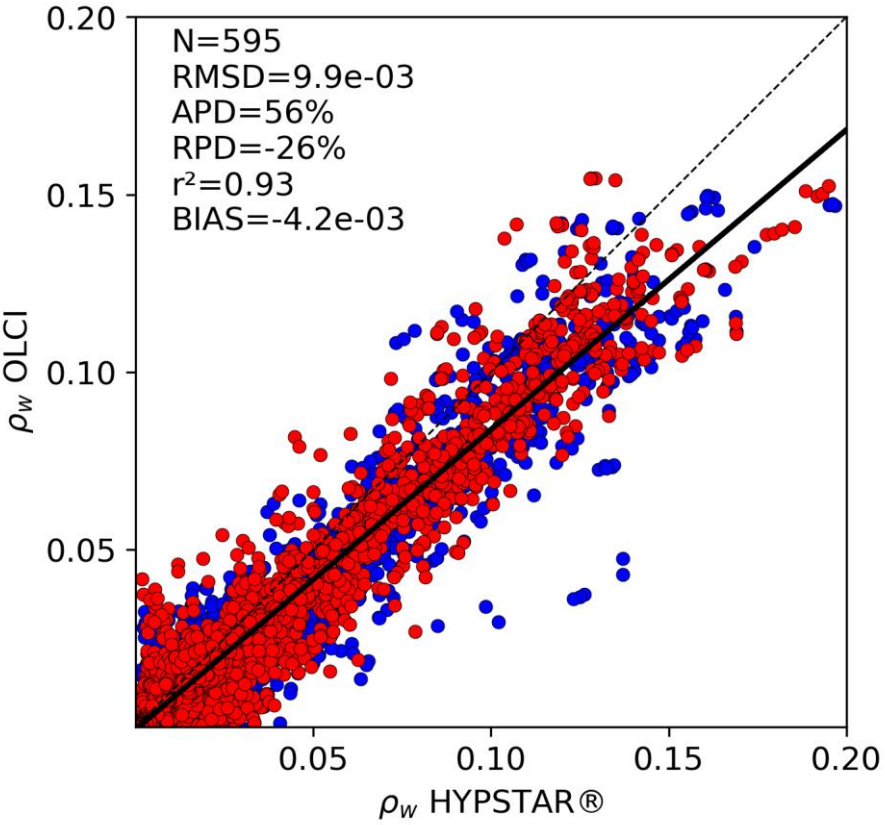

Figure 8: Global scatter plot comparing data points from Sentinel-3A and B.

Wavelength scatterplots with groups (*selectByWavelength*: True, with *groupBy*)

A scatterplot is produced for each wavelength (wavelengths can be limited using ***wl_values***) and saved to an individual file. To produce a parent plot including all these individual scatter plots, options defining multiple plots are required: ***multiple_plot, xfigsize, yfigsize, widthspace, heightspace*** (see **Multiple plots** section below). Data points are coloured by groups. Groups are established using ***groupBy*** and ***groupValues*** (optional)**.** Note that data can optionally be limited using ***selectBy*** (creating a different plot for each flag).

General options: ***comment, apply, output_file***

Scatter plot options: ***use_rhow, log_scale, scale_factor, title, marker, markersize, color, edgecolor, linewidth, min_xy, max_xy, ticks, xlabel, ylabel, include_stats, stat_list, stat_xpos, stat_ypos, units, legend, identity_line, identity_line_style, regression_line, regression_line_style, selectBy, selectValues, wlvalues, legend, legend_values.***

type: *scatterplot*

selectByWavelength: *True*

groupBy*:* Name of the flag variable (i.e *flag_ac, flag_satellite, flag_sensor* or *flag_site*) used to group the match-ups. Data points from different groups are plotted with a different colour/style. Required to group data.

groupValues: Flag list (comma-separated values) used with *groupBy*. Default: All the flags. Optional.

color: Marker colours for each group, defined as a comma-separated list. Default: flag colour list. Optional.

regression_line_groups: If True(T), a regression line based on the data points for each flag is plotted.

regression_line_groups_style: Style of the regression lines for each group. Colours are overridden by the option ***color*** using a different colour for each group. Default: black, none, 0, solid, 1**.** Optional.

In addition to ***color***, a specific style for each group can be optionally defined using comma-separated values for the following options: ***marker, markersize, edgecolor*** and ***linewidth***.

legend_values*:** Legend labels (comma-separated list). Only used if ***legend is True**.** Default: legend labels are set automatically equal to ***groupValues***. Optional.

Example of configuration file for plotting multiple scatter plots (one for each wavelength)

```
[Multiple_Scatter_Pot_By_Site]
comment: multiple scatter plots grouped by site
apply: false
type: scatterplot
selectByWavelength: true
groupBy: flag_site
groupValues: BEFR, VEIT, LPAR, MAFR, GAIT, M1BE
include_stats: true
stat_list: WL, BIAS, r2
stats_xpos: 0.05
stats_ypos: 0.75
fontsizestats: 10
regression_line: true
regression_line_groups: false
marker: o
markersize: 20
linewidth: 0.1
multiple_plot: 4,4
individual_axis: false
min_xy: 0
max_xy: 50
ticks: 0, 10, 20, 30, 40, 50
fontsizelabels: 10
fontsizeaxis: 10
wlvalues: 400, 412.5, 442.5, 490, 510, 560, 620, 665, 673.8, 681.3, 708.8, 753.8, 778.8, 865.0, 885.0
xfigsize: 10
yfigsize: 10
widthspace: 0.1
heightspace: 0.1
color: blue,red,green,cyan,magenta,orange
```

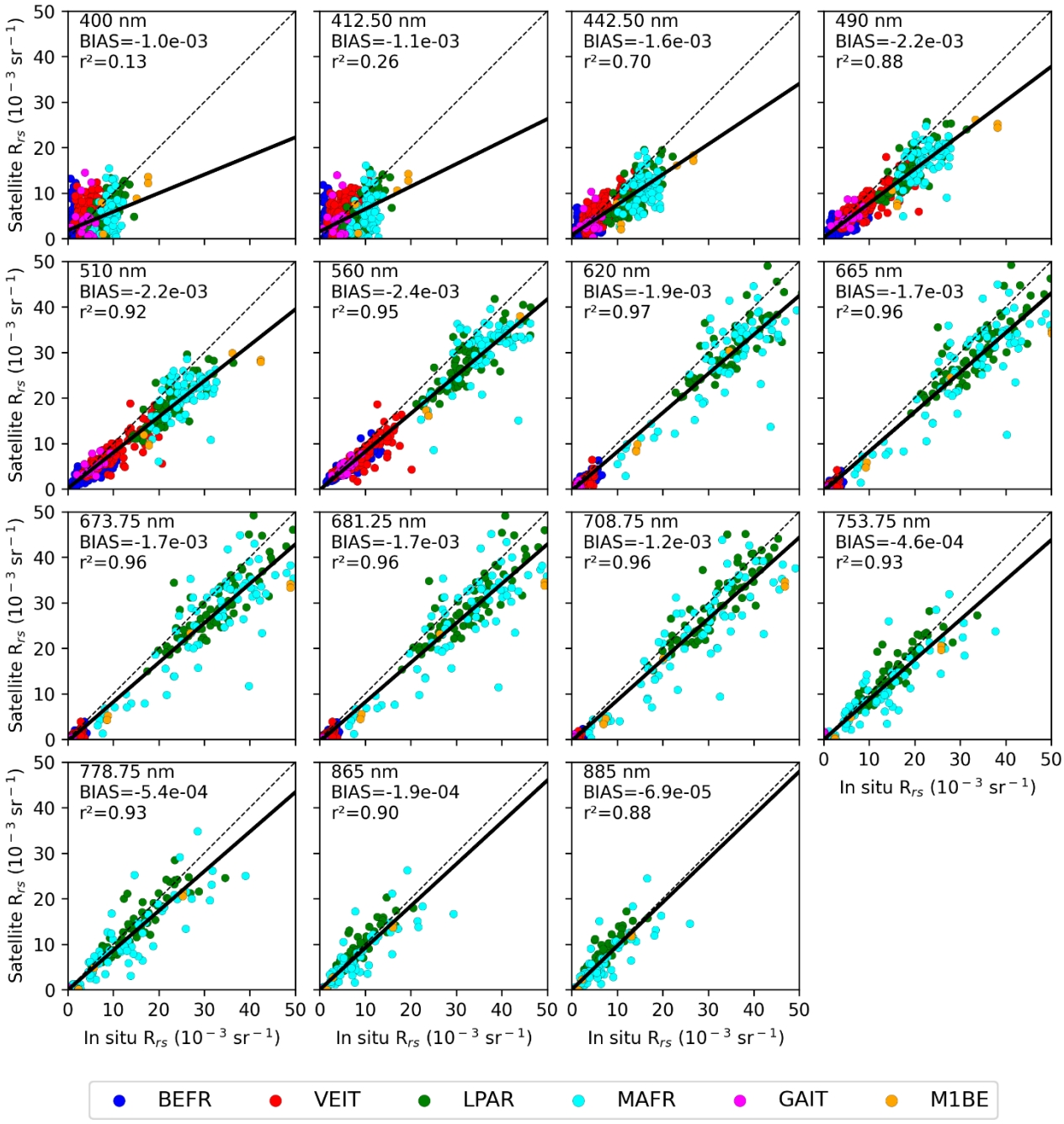

Figure 9: Multiple scatter plot with data points grouped by site

5.3.7. Spectral metrics plots (type: *multipleplot*)

Plots showing the spectral variation of metrics, which are defined using the option ***params***. If more than one metric is selected, a plot file is created for each individual metric. Single plots for each param can also be included in a parent plot using the multiple plot options (***multiple_plot, xfigsize, yfigsize, widthspace, heightspace,*** see section 5.3.8).

Metrics are computed for each wavelength using all the match-ups (single line) or match-ups groups (multiple lines). Groups are established using ***groupBy*** and ***groupValues*** (optional). Wavelengths included in the analysis can also be set using ***wlvalues***.

Note that data can optionally be limited using ***selectBy*** (creating a different plot for each flag).

General options: ***comment, apply, output_file***

type: *statswlplot*

params: Metric list (comma-separated values). Potential values: RMSD, DETER(r2), APD, RPD, BIAS. Required.

wlvalues: Wavelength list (comma-separated values). Default: all the available wavelengths. Optional.

groupBy*:* Name of the flag variable (i.e *flag_ac*, *flag_satellite*, *flag_sensor* or *flag_site*) used to group the match-ups. Metrics from different groups are plotted with a different line style. Optional.

groupValues: Flag list (comma-separated values) used with ***groupBy***. Default: All the flags. Optional.

selectBy*:* Name of the flag variable (i.e *flag_ac*, *flag_satellite*, *flag_sensor* or *flag_site*) used to select data included in the analysis. Optional.

selectValues: Flag list (comma-separated values) used with ***selectBy***. Default: All the flags. Optional.

line_style**:** Line style. A different style could be applied to each group by using a semi-colon separated list of line styles (see section 5.3, Styles). Default: *black, o, 5, solid,1*. Optional.

line_color: Colour list with the colours to be applied to each group. Only used with ***groupBy***. It overrides the colour indicated in ***line_style***. Default: flag colour list. Optional.

multiple_ymin: Minimum metric values (comma-separated values). Default: minimum and maximum values are automatically set depending on data. Optional.

multiple_ymax: Maximum metric values (comma-separated values). Default: minimum and maximum values are automatically set depending on data. Optional.

Example of configuration file for plotting statistical metrics for each wavelength:

```
[Statistical_Metric_Spectra]
comment: plot with spectra for statistical metric
apply: false
type: statswlplot
selectBy: flag_site
selectValues: BEFR, VEIT, LPAR, MAFR, GAIT, GLOBAL
line_color: blue,red,green,cyan,magenta,black
line_width: 2,2,2,2,2,1
marker_size: 15,15,15,15,15,10
params: RMSD
wlvalues: 400, 412.5, 442.5, 490, 510, 560, 620, 665, 673.8, 681.3, 708.8, 753.8, 778.8, 865.0, 885.0
```

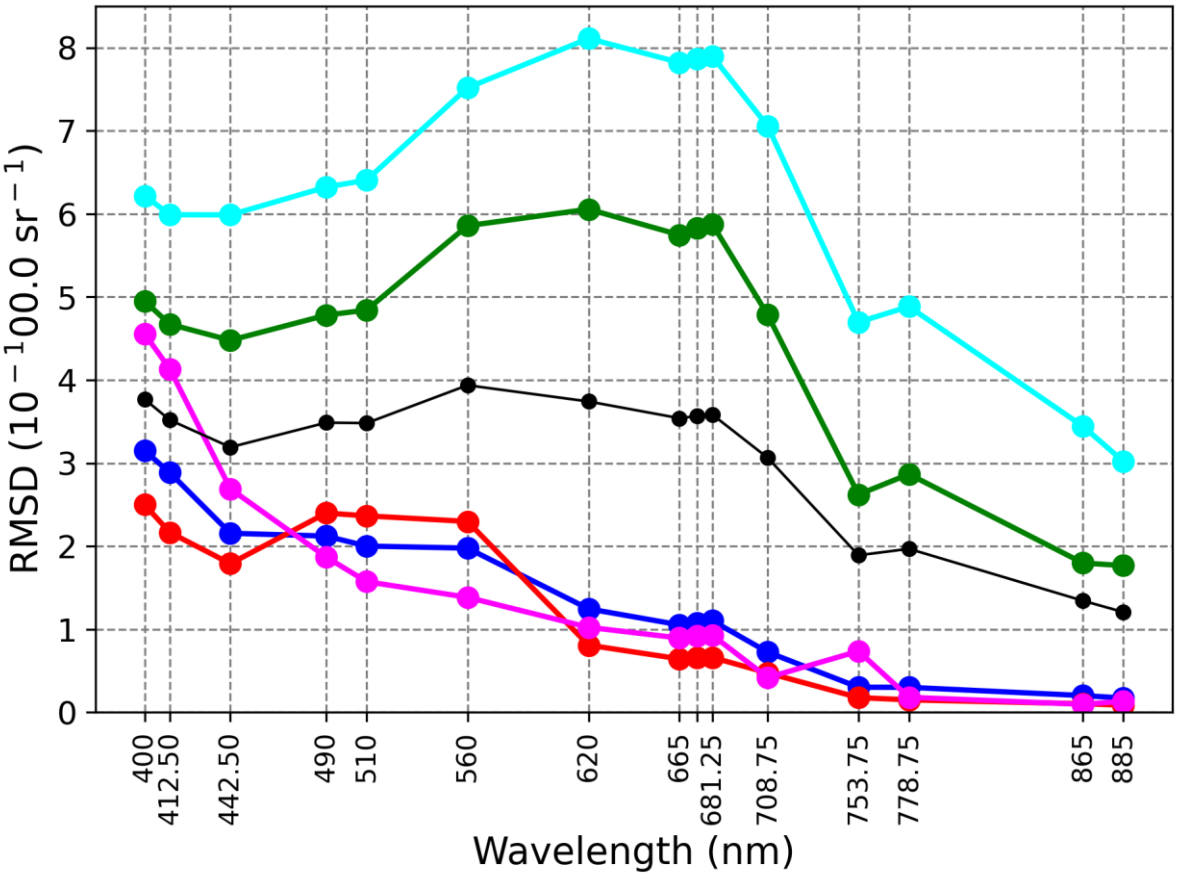

Figure 10: Spectral variation of RMSD grouped by site.

5.3.8. Multiple plots (type: *multipleplot*)

Multiple plots put several plot (files) together into a single plot file.

General options: ***comment, apply, output_file***

type: *multipleplot*

multiple_plot: *nrows, ncols*. Number of rows and columns of the final plot as two comma-separated values. Required.

multiple_files: Path (comma-separated values) of the individual plot files (number must be equal to *nrows* x *ncols*) to be included in the final plot. Paths can be completed or only the file name if files are stored in ***output_path*** (see GLOBAL_OPTIONS). Required.

xfigsize: Size of the final figure **(**in inches). Required.

yfigsize: Size of the final figure (in inches). Required.

widthspace: Width space between plots. Required.

heightspace: Height space between plots. Required.

Moreover, multiple plots can be implemented by adding these options (except for ***multiple_files***) in other plot types:

- Heatmaps with temporal distribution of match-ups (***type***: *temporalheatmap*) (section 5.3.1).

- Satellite and in situ spectra stats comparison (***type***: *spectraplot*, ***type_rrs***: *comparison_sat_insitu*) (Section 5.3.4).

- Satellite spectra stats comparison (***type***: *spectraplot*, ***type_rrs***: *comparison_sat*) (Section 5.3.4).

- Wavelength scatterplots without groups (***selectByWavelength***: *True*, without ***groupBy***) (Section 5.3.5).

- Wavelength scatterplots without groups (***selectByWavelength***: *True*, without ***groupBy***) (Section 5.3.5).

5.3.9. Metrics (type: *statstable*)

It produces a CSV file with metrics values.

General options: ***comment, apply, output_file***

type: *statstable.* Required.

***use_rhow*:** If True (T) *Rrs* is converted to *Rhow* (*Rhow = Rrs* * PI). Default: False (F). Optional.

selectBy*:* Name of the flag variable (i.e *flag_ac, flag_satellite, flag_sensor* or *flag_site*) used to select data included in the analysis. Optional.

selectValues: Flag list (comma-separated values) used with ***selectBy***. Default: All the flags. Optional.

params: Metric list (comma-separated values). Potential values: RMSD, DETER(r2), APD, RPD, BIAS. Required.

wlvalues: Wavelength list (comma-separated values). Default: all the available wavelengths. Optional.

formatted: If True, values are formatted.