OLCI validation_py27

Content

Introduction

The package “Doc OLCI validation_py27” of python 2.7 scripts aims to provide a first version of some basic tools for the evaluation of Earth Observation (EO) data extracted for Swedish waters from Sentinel 3 sensor OLCI (Ocean and Land Colour Instrument) level 2 product Chl_nn, Chlorophyll concentration calculated according to the Case2R algorithm. The evaluated time period is defined by the available data supplied by the user.

The package uses the PYTROLL package Satpy for reading and processing the raw level 2 satellite data. The resolution is 300 meters when transformed to a fixed reference frame (SWEdish REference Frame 1999, Transverse Mercator) based grid called the BAWS (Baltic Algae Watch System) area. The area covers the North Sea, Kattegat and the Baltic Sea. The satpy method to mask clouds and land is used similar to the BAWS surveillance.

The in-situ validation data of Chlorophyll a (bottle data from 0-1 meter depth in physical-chemical columns (with no duplictes)) are extracted from SHARK ("Svenskt HavsARKiv" / "Swedish Ocean Archive").

The basic functioning of the package is divided in two steps

Step 1: Work with the satellite scenes and the entire selected area of downloaded SHARK data. Save EO and SHARK data from matching positions.

1. Read SHARK data from the selected area (from SHARK text file with dot and tab formatted 'utf-8' SHARK data).
2. Read EO data with satpy (from NetCDF file chl_nn.nc from the given scenes).
3. Calculate mean values for EO data (May-Sep, Jun-Aug, May, Jun, Jul, Aug, Sep) and save the mean values and the number of data used for the calculation in each pixel in NetCDF files for later plotting.
4. Select EO data from all EO-points with SHARK data.
5. Plot some basic statistics for the matching data sets from SHARK and EO (Fig.1).

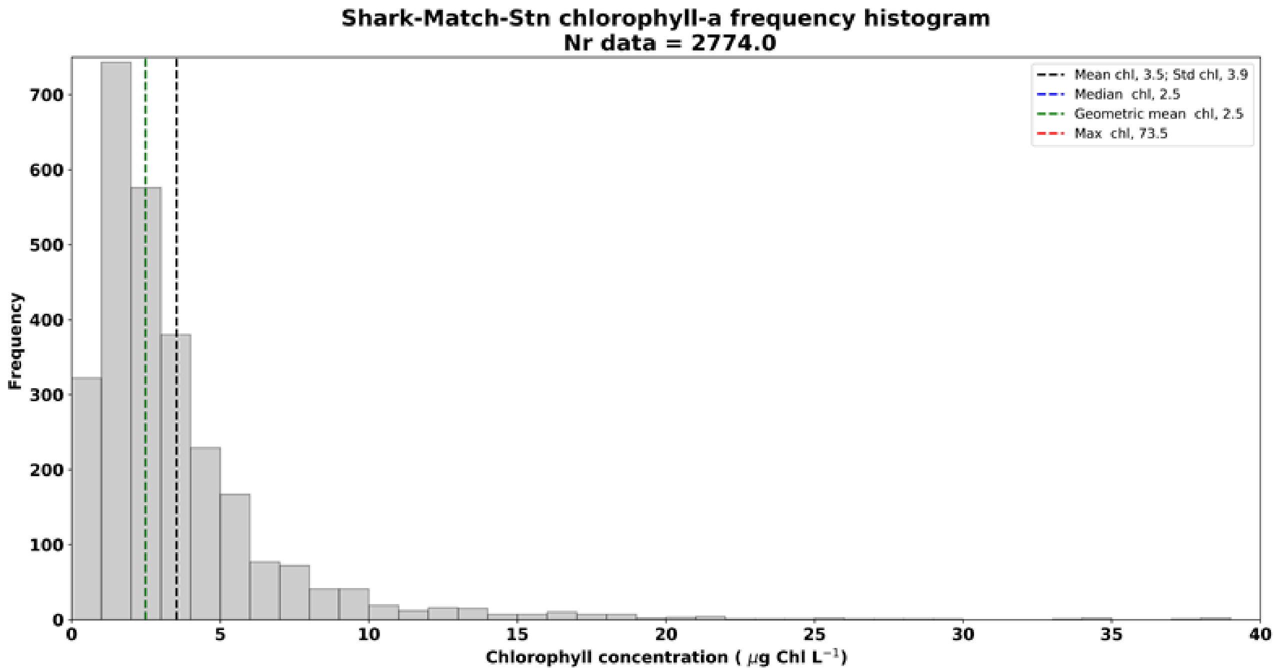

Figure 1. Example histogram of SHARK data for all data found from matching stations. Mean (black), Geometric mean (green), Median (blue, hidden behind the Geometric mean), Max value (red, outside of the selected x-axes range). Similar histograms are produced for matching dates for SHARK and EO as well.

6. Plot maps showing the number of visits at stations both in SHARK and in EO data (Fig.2).

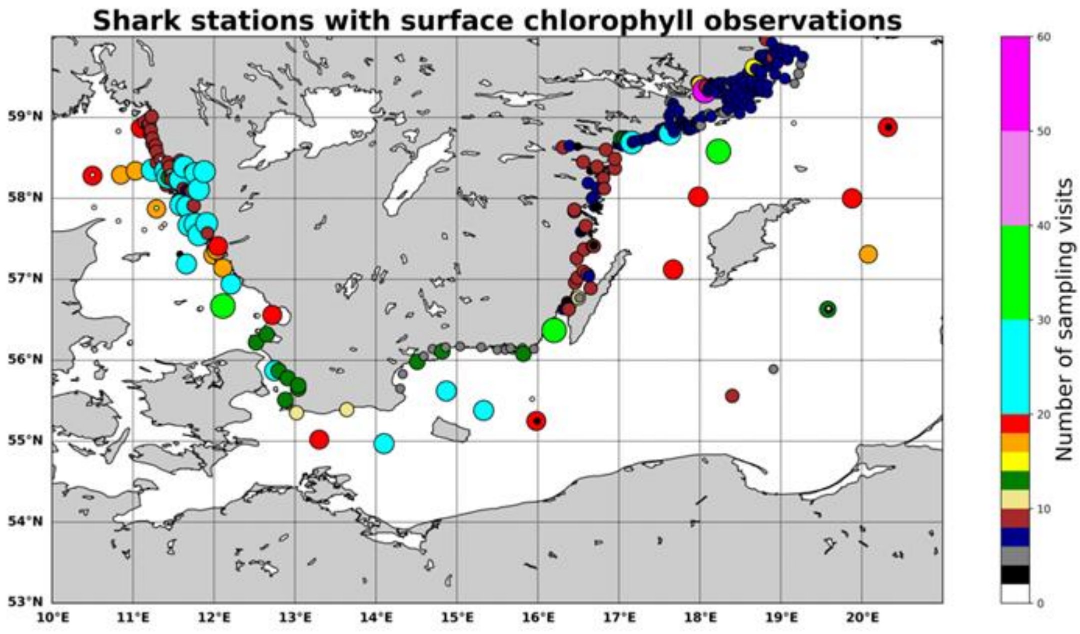

Figure 2. Example map with number of sampling visits at different positions. Similar maps are produced for matching dates of SHARK and EO data. (The unique SHARK station positions are defined by a round off of lat and lon decimals by 0.01 degree. Hence several stations may coincide in a larger circle on the map which indicate the number of the most frequently visited position.)

7. Save data (*year, month, day, lat, lon, chl* for SHARK and EO data) from all stations in a JSON file for later analysis and plotting.
8. Use the NetCDF files (from point 3) to plot overview maps with mean values for EO data and the number of data used for the calculation in each pixel (Fig.3).

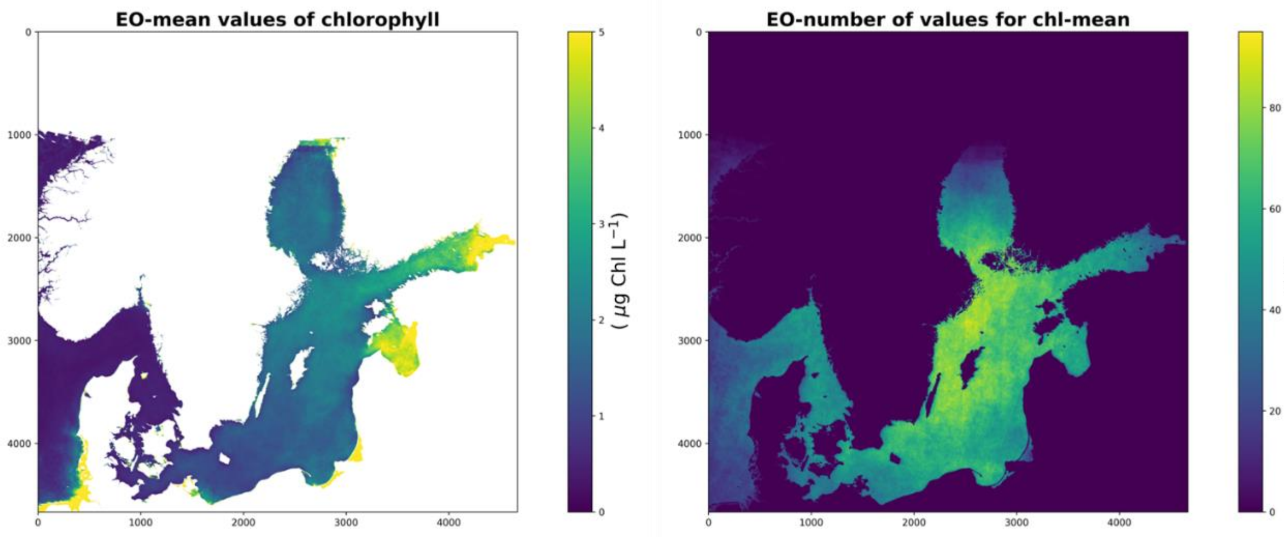

Figure 3. Example EO data June-August 2016-2018. Geometric mean (left) and the number of values used for the calculation (right).

Step 2: Validate subsets of data from JSON file with EO data corresponding to the positions of SHARK stations. Inspect results from specific stations using an interactive interface.

The aim of this package is to allow for work with selected station data from smaller areas without the heavy handling of the large satellite dataset.

1. Select a defined area smaller or equal to the area used in Step 1 (Fig.4).

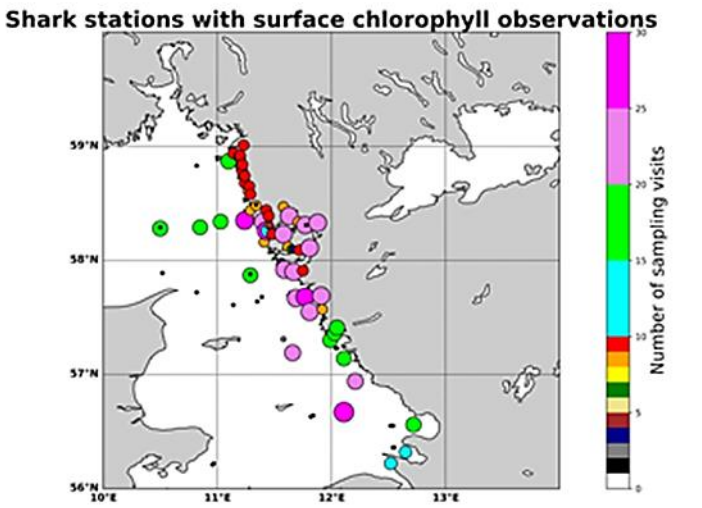

Figure 4. Example of a subset with SHARK stations extracted from the JSON file.

2. Plot histogram with statistics for all SHARK stations within the subset (Fig.5).

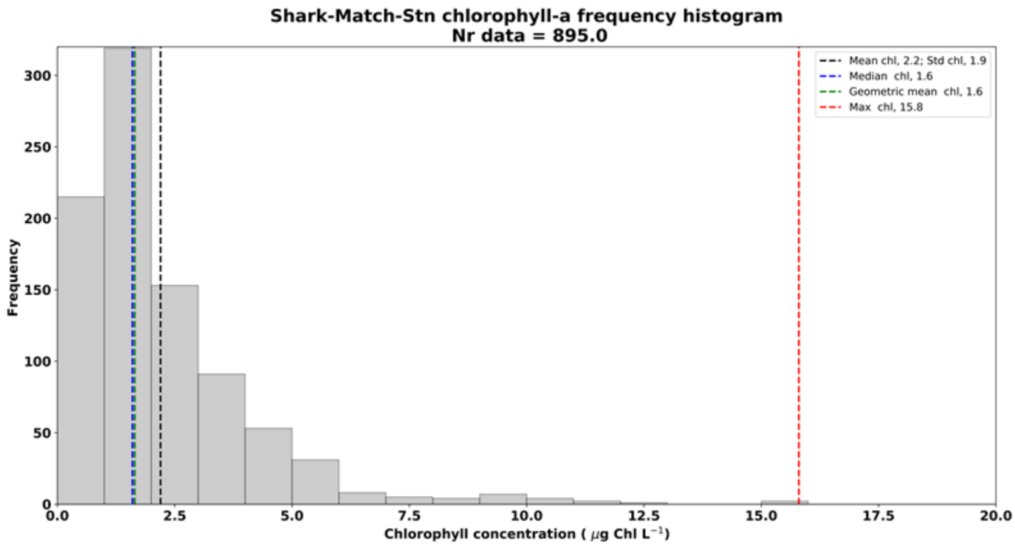

Figure 5. Example histogram of SHARK data within the subset (similar to Fig.1).

3. Plot histogram for data with matching date at SHARK stations within the subset (Fig.6).

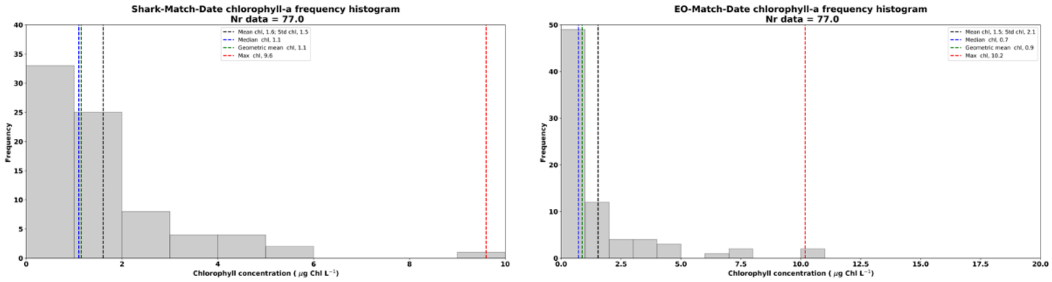

Figure 6. Example histogram for SHARK (left) and EO (right) data within the subset (similar to Fig.5).

4. Plot linear correlation between EO and SHARK data with matching dates.

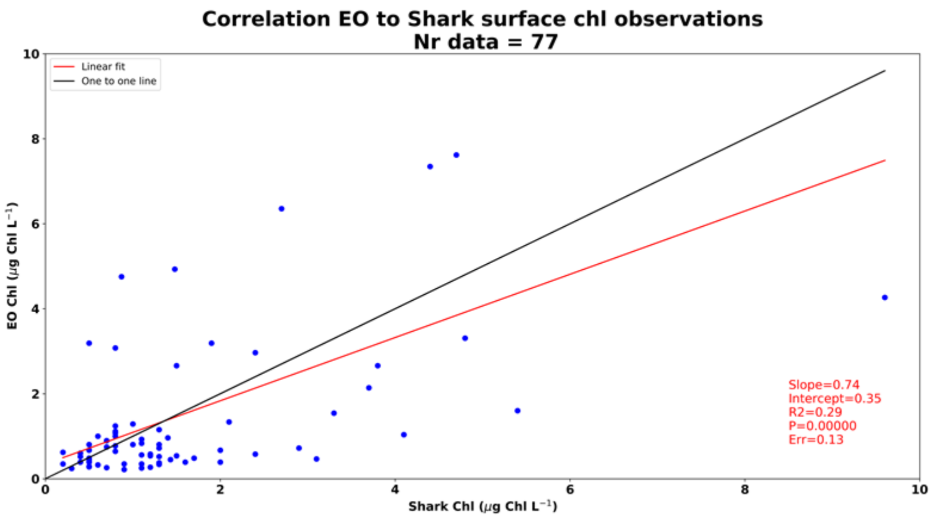

Figure 7: A linear least-squares regression line (red) for EO and SHARK data (blue dots). slope = slope of the regression line, intercept = intercept of the regression line, R2 = squared r-value correlation coefficient, P = p-value two-sided p-value for a hypothesis test whose null hypothesis is that the slope is zero using Wald Test with t-distribution of the test statistic, Err = stderr Standard error of the estimated gradient. The black line shows a hypothetical one to one fit.

5. Plot maps with mean values from SHARK data and from EO data (Fig.8).

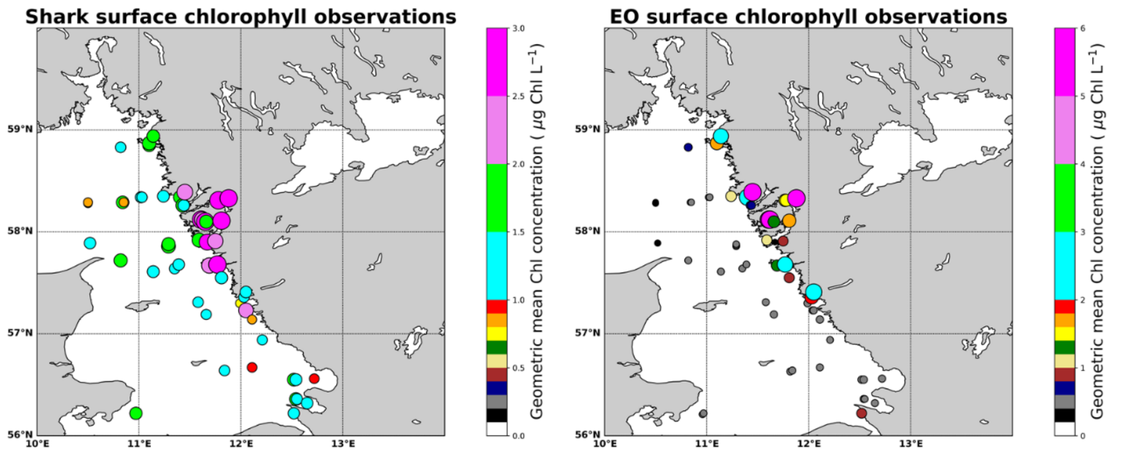

Figure 8. *Example geometric mean from SHARK (left) and EO (right) station data.*

6. Plot maps with bias between EO and SHARK mean values (Fig.9).

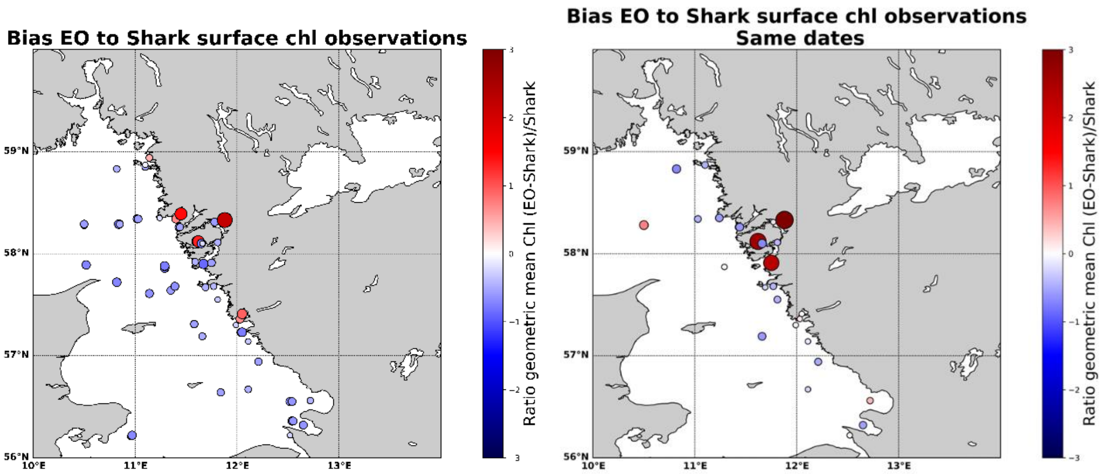

Figure 9. *Example of bias in geometric mean (GM) values between EO and SHARK data. The calculation is based on all available data at stations (left) and on data from matching dates (right). The bias is defined by the difference between the means from EO and SHARK data divided by the mean from SHARK data, i.e. Bias=(GM_{EO}-GM_{SHARK})/GM_{SHARK}.*

7. Plot interactive maps similar to Fig.9. Click on a station and obtain detailed information from stations within a certain radius of the pointer.

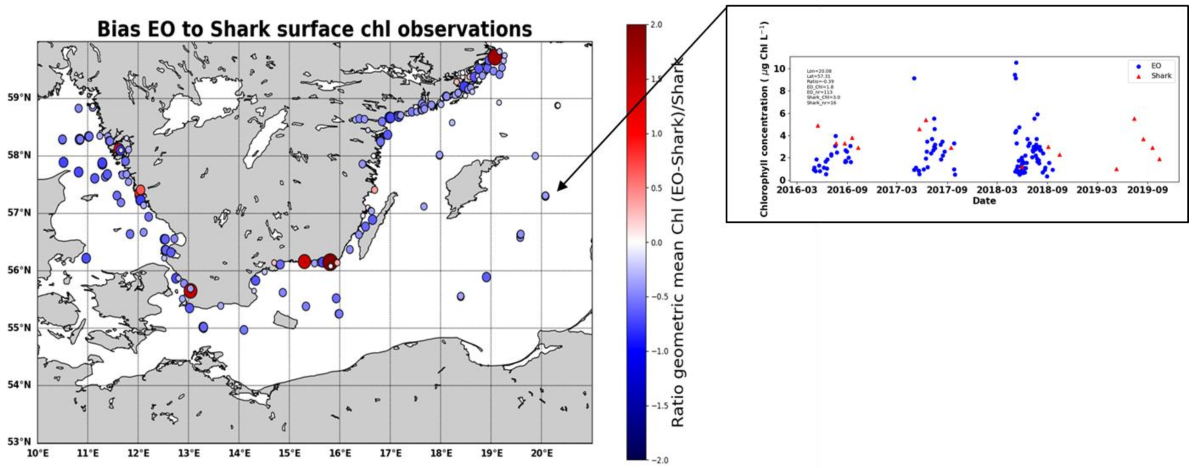

Figure 10. Example of interactive map similar to Fig.9 with a possibility to click on stations. The click activates separate plots (from stations within a certain radius of the pointer) including time series of SHARK data (blue dots) and EO data (red triangles). Information in the upper left box in the time series figure includes; the position of station (Lat, Lon), the bias (called ratio) of geometric means from SHARK and EO data at that position, the geometric mean value of EO (EO_Chl) and SHARK (SHARK_Chl) data and the number of observations (EO_nr, SHARK_nr) used in the calculation of the mean value.

Configuration setup

A config file (eo_validation_config_tab.txt) is used to do some preliminary settings.

Step 1 config

The path to the config file is hard coded in the file EO_SHARK_Validation_v200107.py within the script run_eo_validation() which is used to define the setup from the config file eo_validation_config_tab.txt in case of the larger area during Step 1.

The eo_validation_config_tab is used to define

- Limit the latitude and longitude of the investigated area
- Set path to SHARK data
- Set path to EO Sentinel 3 OLCI data
- Set Bin step for Chl histogram
- Set maximum value for x-axis in Chl Histogram
- Set EO chl parameter (use Chl_nn)
- Set BAWS grid size (use 300 m)
- Set EO reader (use olci_l2)
- Set EO sensor (use olci)

Step 2 config

The path to the config file is hard coded in the file Station_map_validation_v200108.py part of definitions which is used to define the setup from the config file eo_plot_validation_config.txt in case of Step 2.

The eo_plot_validation_config is used to define additional information for the interactive plotting and possibility to save plots.

- Limit the latitude and longitude of the investigated area
- Set path to SHARK data
- Set path to EO Sentinel 3 OLCI data
- Set Bin step for Chl histogram
- Set maximum value for x-axis in Chl Histogram
- Set EO chl parameter (use Chl_nn)
- Set BAWS grid size (use 300 m)
- Set EO reader (use olci_l2)
- Set EO sensor (use olci)
- Pick (Use value 1 to plot interactive maps. Else set value =0)
- Plot all (Use value 1 to plot stations maps and statistics similar to Figs 5-9. Else set value =0)
- Plot maps (Use value 1 to plot maps similar to Fig.3. Else set value =0)

How To

1. Download Satpy from PYTROLL to folder satpy
2. Save your EO data in folder eo_files
3. Save your SHARK data in folder shark_files
4. Create empty folders:
    a. stations_data (for saving the json data)
    b. map_data (for saving the EC NetCDF data)
    c. figures (for saving figures)
5. Set up the configuration eo_validation_config_tab.txt (for Step 1) and eo_plot_validation_config.txt (for Step 2) in folder config_files
6. Use script EO_SHARK_Validation_v200107.py to run the program run_eo_validation() which performs tasks described in Step 1.
7. Use script Station_map_validation_v200108.py to run the station validation program described in Step 2.