Ocean Color – Simultaneous Marine and Aerosol Retrieval Tool (OC-SMART)

User Guide (Python)

Light and Life Laboratory (LLLab)

Physics Department, Stevens Institute of Technology

1. Setup Python environment

We recommend setup the Python environment using Miniconda.

1.1 Install Miniconda. You may download Miniconda from https://docs.conda.io/en/latest/miniconda.html

1.2 If you would like to create a new conda environment, use command:

conda create -n <envname>

and then activate this environment use command:

conda activate <envname>

1.3 Setup Python version and install dependencies use following command:

conda install python=3.6.6

conda install numpy scipy h5py netcdf4 bzip2 urllib3 basemap glob2 pyproj gdal lxml

conda install -c sunpy glymur

conda install -c conda-forge matplotlib pyhdf

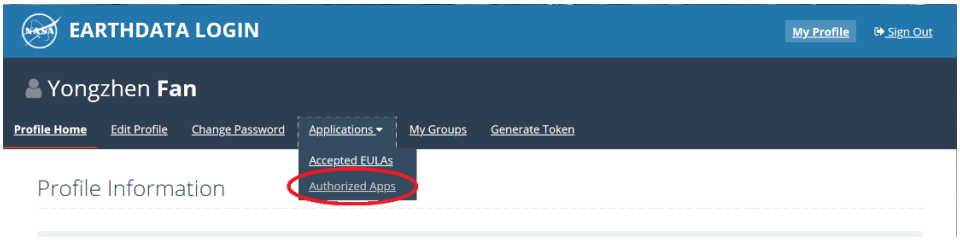
conda install -c DHI-GRAS py-l8angles (**ONLY available for Windows 64**)

2. Ancillary data downloading

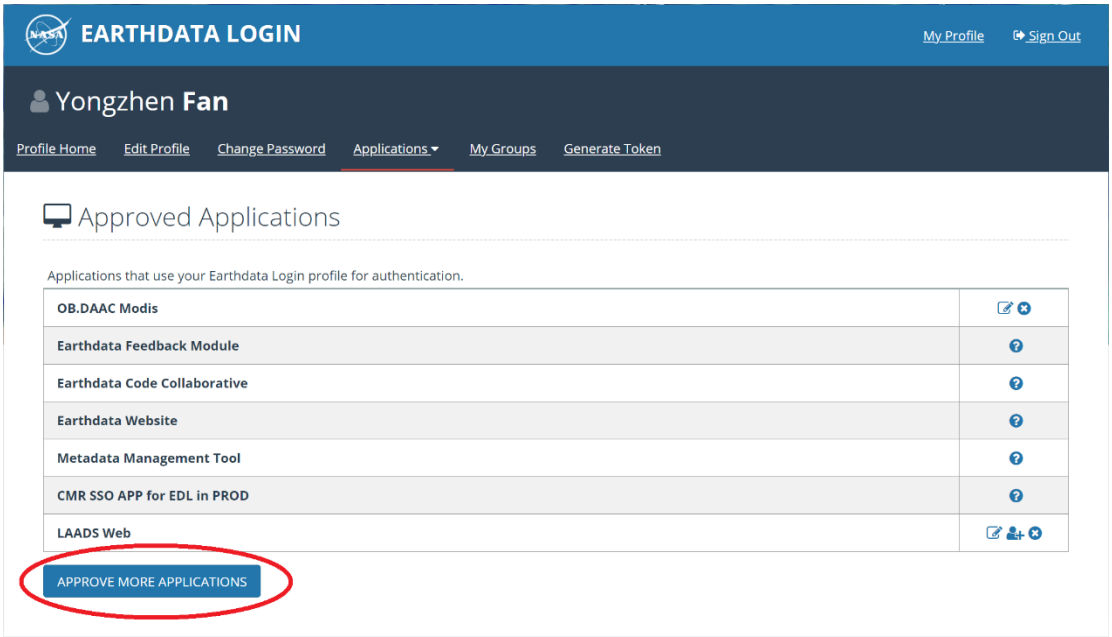
OC-SMART needs to download ancillary data from NASA OB.DAAC, the user must have an account on earthdata.nasa.gov and approve the application to download ancillary data.

2.1 If you don't have an account on Earthdata please go to https://urs.earthdata.nasa.gov/, click "Register" and follow the instruction to create your account.

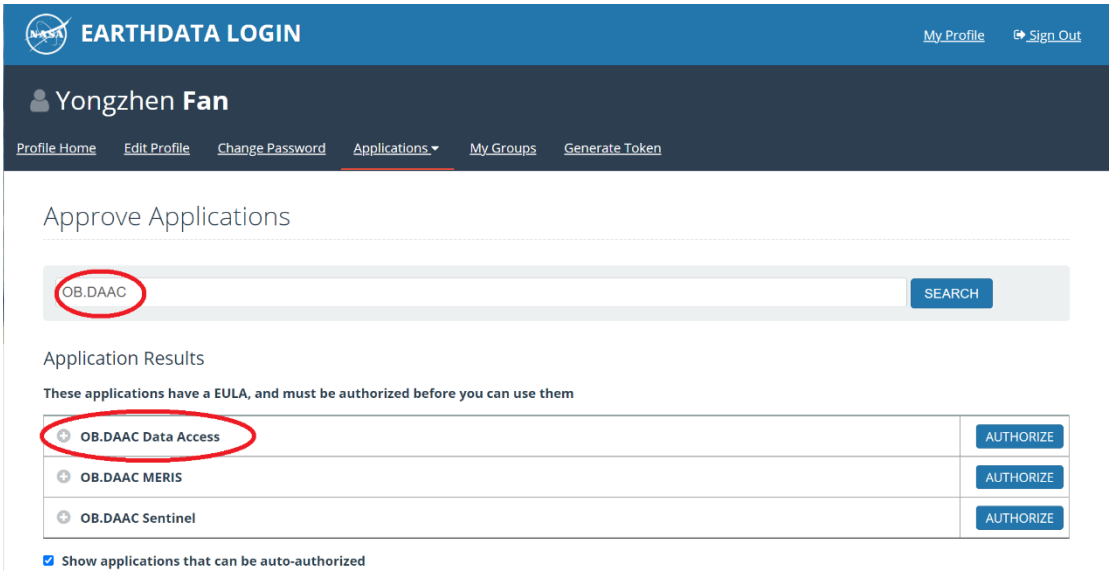
2.2 Login to your Earthdata account, click "Applications->Authorized Apps",



on the next page, click "Approve more Applications" at the bottom,



then search for "*OB.DAAC*" and check "Show applications that can be auto-authorized", then click "Authorize" on "*OB.DAAC Data Access*".



2.3 set up automatic authentication for data downloading

2.3.1 *For Linux user:*

1) In OC-SMART package, locate the files "*.netrc*" and "*.urs_cookies*" in the '*DataDownloadAuthentication/Linux*' directory.
2) Replace 'USERNAME' and 'PASSWORD' in the "*.netrc*" and "*.urs_cookies*" file with your own Earthdata login credentials.
3) Copy the "*.netrc*" and "*.urs_cookies*" file to your home directory, i.e. /home/<*user account*>

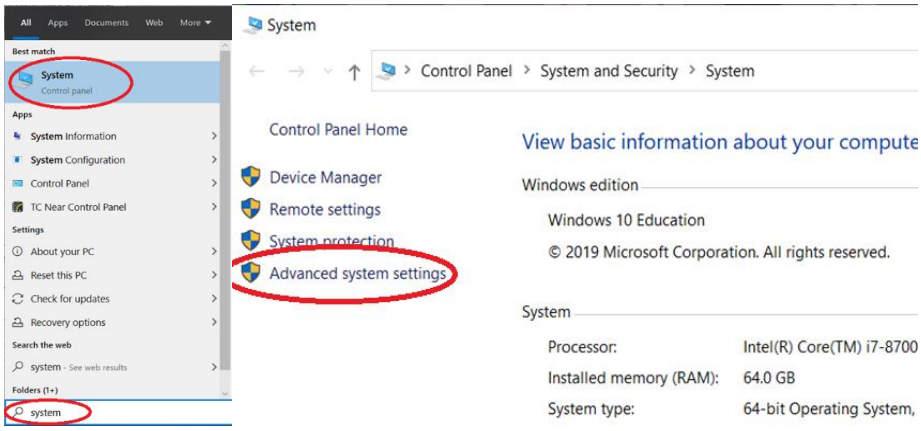
4) Open terminal, go to your home directory and run command:
chmod 0600 ~/.netrc

2.3.2 *For Windows user:*

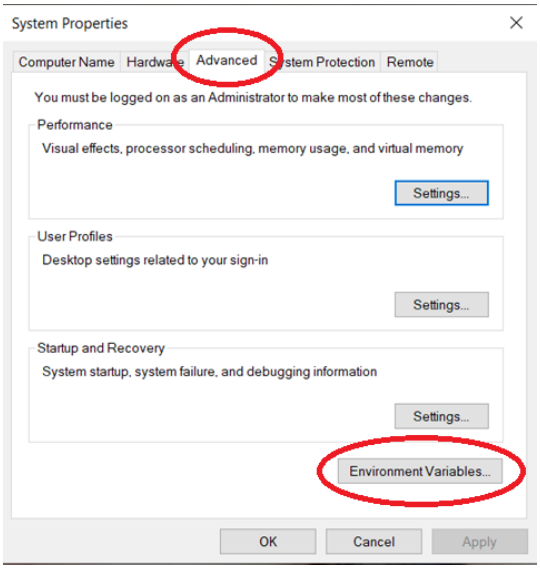
1) Setup "HOME" environment variable

WARNING: change the environment variable may jeopardize your system. If you feel you are not qualified to do so, please consult with your system administrator.

a) Search "system" and open System(control panel), then click "Advanced system settings"



b) Click on "Advanced" tab and then open "Environment Variables…"

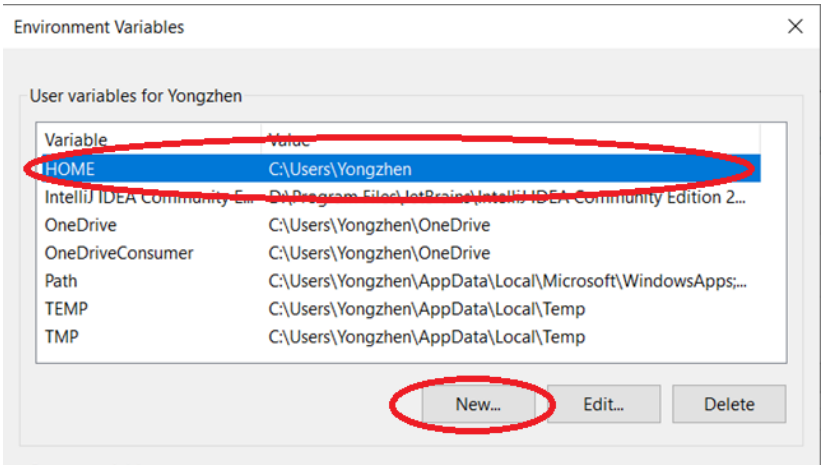


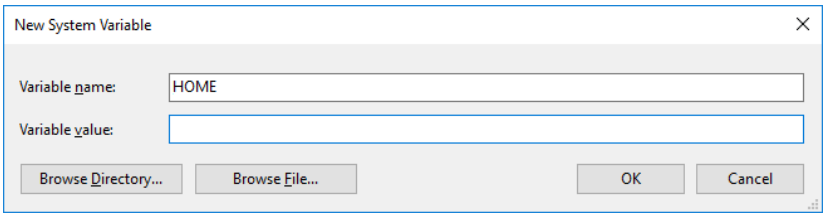
c) Create "HOME" variable

WARNING: if "HOME" variable already exists in the variable list,

please DO NOT CHANGE it, as it may jeopardize other apps in your system, simple write down the path where the "HOME" variable specified and use it in step 4).

If the "HOME" variable does not exist in the variable list, you may create the variable by click "New…" and input "HOME" in "Variable name" and input the path you would like to be your "home director" in "Variable value", for example "*C:\Users\<user name>*".





2) In OC-SMART package, locate the file "_netrc" in the "*DataDownloadAuthentication/Windows*" directory.
3) Replace 'USERNAME' and 'PASSWORD' in the "_netrc" file with your own Earthdata login credentials.
4) Copy the "_netrc" file to the location where the "HOME" variable specified.

3. How to run OC-SMART

3.1 System memory requirement

We recommend a minimum of 16GB of physical system memory to run OC-SMART, 64GB is recommended for best performance on larger images, such as GOCI, Landsat-8, Sentinel-2 and Sentinel-3. If physical memory is limited on your system, please increase the size of the virtual memory, i.e., the pagefile in Windows system or the swap file in Linux system.

3.2 Run OC-SMART

After downloading and unzip OC-SMART, you may run OC-SMART in command line or using a Python IDE, such as Spyder.

To run OC-SMART in command line, open terminal in Linux system or Anaconda Prompt in the Windows system, navigate to the OC-SMART directory then run command:

python OCSMART.py

To run OC-SMART in a Python IDE, such as Spyder, open OCSMART.py in the IDE then click run.

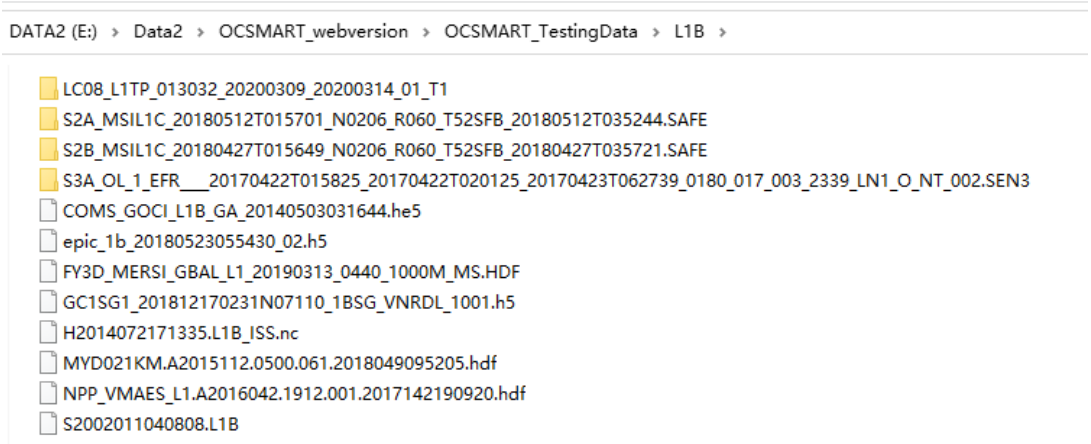
3.3 Supported sensor

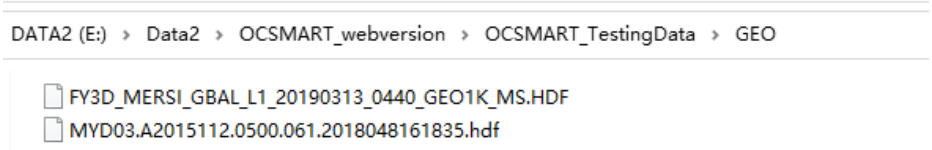
SeaWiFS, MODIS (Aqua and Terra), VIIRS (SNPP), GOCI (COMS), SGLI (GCOM-C), OLCI (Sentinel-3A), MSI (Sentinel-2A and Sentinel-2B), EPCI (DSCOVR), MERSI-II (FY-3D), HICO (ISS), OLI (Landsat-8, **NOT supported in Linux version**).

3.4 Input data files

OC-SMART requires the level-1B(L1B) satellite reflectance data and the associated geolocation file as input. L1B data file for SeaWiFS can be acquired by processing the L1A data using NASA SeaDAS software.

The L1B satellite reflectance data file from all supported sensors can be located in a common directory. Some sensors, like MODIS and FY-3D MERSI-II have separated geolocation files, which MUST be located in a different directory. An example of the L1B and GEO directory may look like:





3.5 Input parameter setup

The input parameters of OC-SMART can be set in the file *OCSMART_Input.txt*.

3.5.1 The following example will process the entire satellite image:

l1b_path = ./L1B/	Directory where L1b data is located
geo_path = ./GEO/	Directory where geolocation data is located
l2_path = ./L2/	Directory where L2 output data will be located
solz_limit = 70.0	Maximum solar zenith angle (must be <=70)
senz_limit = 70.0	Maximum sensor viewing angle (must be <=70)

3.5.2 Sub-image processing

There are 3 options to define a sub-image in OC-SMART. Your need to add following parameters for each option.

a) Define sub-image using a range of latitude and longitude. OC-MART will extract the sub-image defined by the north-west and south-east corner. To use this option, add following parameters in *OCSMART_Input.txt*

north = <max latitude>
south = <min latitude>
east = <max longitude>
west = < min longitude >

Note: use negative values for latitude in southern hemisphere and longitude in western hemisphere.

b) Define sub-image using center latitude/longitude and the size of a box. OC-SMART will find the pixel that is closest to the input center latitude/longitude and extract a sub-image in the same size as user defined. To use this option, add following parameters in *OCSMART_Input.txt*

latitude_center= <latitude>
longitude _center = < longitude >
box_width = <number of pixels>
box_height = < number of scanlines >

NOTE: i) use negative values for latitude in southern hemisphere and longitude in western hemisphere. ii) *number of pixels* and *number of scanlines* should be an odd number, if even numbers are given, the size of the output is (*number of pixels*+1) by (*number of scanlines*+1). iii) if the input center latitude/longitude is very close to the edge of the satellite image, OC-SMART only extract the part of the sub-image that is overlapped with the satellite image.

c) Define sub-image with range of the scanline and pixel number. To use this option, add following parameters in *OCSMART_Input.txt*

start_line = <start scanline number>
end_line = < end scanline number >
start_pixel = < start pixel number >
end_pixel = < end pixel number >

Note: i) input scanline and pixel number should be ZERO based. ii) This option can be used to process whole image in blocks if system memory is limited.

3.6 Output file

OC-SMART output level-2 (L2) product in HDF5 format. The output is compatible with NASA SeaDAS and ESA SNAP for visualization, and can be read by other software that support HDF5 format, such as Matlab, HDFView, Python, etc.

3.6.1 The L2 product include:

- Spectral aerosol optical depth (AOD)
- Spectral normalized remote sensing reflectance (Rrs)
- Spectral total absorption by particulates in water (ap)
- Spectral total scattering by particulates in water (bp)
- Spectral absorption by Phytoplankton (aph)
- Spectral absorption by detritus and gelbstoff (adg)
- Spectral backscattering by particulates (bbp)
- Chlorophyll_a concentration by NASA OCi algorithm (chlor_a(oci))
- Chlorophyll_a concentration by YOC algorithm (chlor_a(YOC))
- Total suspended matter by YOC algorithm (tsm(YOC))
- Solar zenith angle, sensor zenith angle and relative azimuth angle
- Latitude and Longitude
- L2_flags

3.6.2 L2_flags

- 0: Valid pixel
- 1: Satellite L1 reflectance unavailable (i.e., saturation or missing values)
- 4: Solar or sensor viewing angle out of range
- 16: Land
- 64: Cloud
- 256: Rayleigh corrected reflectance (Lrc) out of scope
- 1024: negative Rayleigh corrected reflectance (Lrc)

3.7 clean temporary files

OC-SMART will save ancillary files to your local drive, if storage space is limited, manually delete the ancillary files located in **./anc/** and **./ landmask_gsw/**.

4. Referencing

When acknowledging the use of OC-SMART for scientific papers, reports etc. please cite the following reference:

Fan, Y., Li, W., Chen, N., Ahn, J., Park, Y., Kratzer, S., Schroeder, T., Ishizaka, J., Chang, R, and Stamnes, K., (2020) "OC-SMART: A machine learning based data analysis platform for satellite ocean color sensors", Remote Sens. of Environ., Vol. 253, p11236, DOI: 10.1016/j.rse.2020.112236.

5. Term of Use

Subject to the terms and conditions of this Terms of Use, LLLab grants to the User (means the user of OC-SMART who has accepted the Terms of Use) a personal, non-exclusive, non-transferable license to use the OC-SMART software (include object code, source code, accompanying documentation, and any upgrades, enhancements and corrections made by LLLab or by the User) for the exclusive purpose of scientific research excluding any commercial purpose. The User may use the OC-SMART Software in its source code format for its own use and may translate or modify the OC-SMART Software or incorporate them into other software. The User may not, however, transfer or sublicense the OC-SMART Software to any third party, in whole or in part, in any form, whether modified or unmodified.

6. Warranty disclaimer

The User acknowledges and accepts the OC-SMART software "AS IS". LLLab provides no warranties as to the function or use of the software, whether express, implied, or statutory, including, without limitation, any implied warranties of merchantability or fitness for particular purpose. The entire risk as to the quality and performance of the software is with the user. LLLab does not warrant that the functions contained in the software will meet user's requirements or that the operation of the software will be uninterrupted or error free.