1. Introduction

ACOLITE bundles the atmospheric correction algorithms and processing software developed at RBINS for aquatic applications of metre and decametre satellite data, from among others Landsat (5/7/8) and Sentinel-2 (A/B), Pléiades and PlanetScope. ACOLITE is in particular suited for processing of turbid waters and small inland water bodies, but can be used with reasonable success over clearer waters and land.

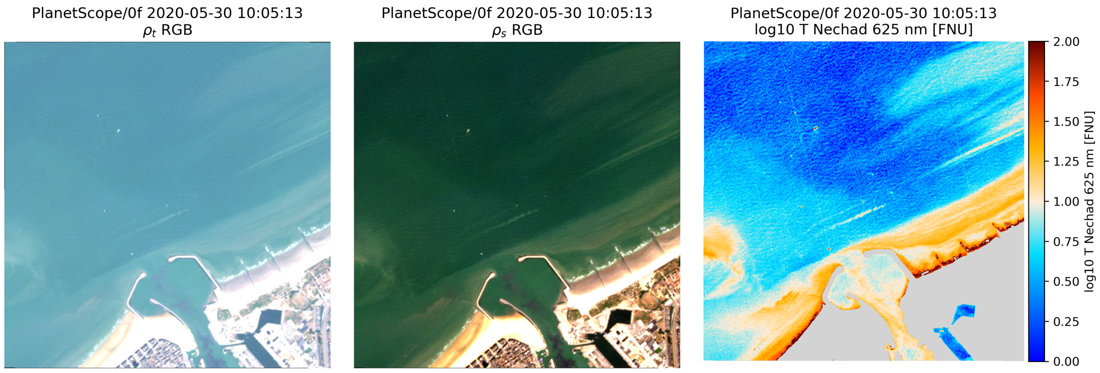

Figure 1: PlanetScope image of Oostende 2020-05-30 centred on the RT1 measurement platform. Left: ρ_t top-of-atmosphere RGB composite. Middle: ρ_s surface-level RGB composite. Right: Nechad turbidity as retrieved from the 625 nm Red band.

ACOLITE performs both the atmospheric correction and can output several parameters derived from water reflectances. RGB composites and PNG maps can also be generated. ACOLITE was originally implemented in IDL (2014–2017), and has been translated into Python (2018–2021). A new generic Python version (2021–present) was released for public beta in April 2021 that integrates the processing of metre-scale sensors such as Pléiades, WorldView, and PlanetScope, as well as the processing of Sentinel-3 (A/B) OLCI data. The original IDL and Python implementations are no longer supported or developed. Version 20210114.0 of the binary release is the last version based on the older Python codebase, versions after are based on the new generic Python codebase.

Please visit the ACOLITE forum (http://odnature.naturalsciences.be/remsem/acolite-forum/) or email Quinten (quinten.vanhellemont@naturalsciences.be) for bug reports, support and feature requests. The latest ACOLITE Python source code is available from GitHub (https://github.com/acolite/acolite). The previous Landsat and Sentinel-2 version, and the metre-scale version, are no longer supported but both versions remain available on GitHub (respectively at https://github.com/acolite/acolite_ls2 and https://github.com/acolite/acolite_mr).

2. Atmospheric correction

2.1. Overview

ACOLITE Python includes two atmospheric correction algorithms, the default "Dark Spectrum Fitting" or DSF algorithm (Vanhellemont and Ruddick, 2018; Vanhellemont, 2019a, 2020c) and the older "Exponential extrapolation" or EXP algorithm (Vanhellemont and Ruddick, 2014, 2015, 2016). The use of the DSF is recommended, but the EXP is included for completeness, and for users and applications that rely on it. The adaptation and applicability of the DSF to Landsat and Sentinel-2 is described in Vanhellemont (2019a, 2020c). DSF for metre-scale imagery can be found in Vanhellemont and Ruddick (2018); Vanhellemont (2019b, 2020c), and the adaptation to Sentinel-3 OLCI is discussed in Vanhellemont and Ruddick (2021).

The Thermal Atmospheric Correction Tool (TACT, Vanhellemont, 2020a,b) is now integrated in ACOLITE and can be used for retrieving surface temperatures from the Landsat sensors. Some additional configuration is required for running TACT as it relies on atmospheric profile inputs and libRadtran (see further).

2.2. DSF options

2.2.1. Aerosol correction

The DSF computes aerosol optical depth (τ_a at 550 nm) based on multiple dark targets in the scene or subscene, with no a priori defined dark band. For each band the darkest object is either estimated from (1) the absolute minimum, (2) a percentile, or (3) the offset from an OLS fit to the first thousand pixels in the histogram, and a "dark spectrum", ρ_{dark} is created. A Continental or Maritime aerosol model is chosen based on the lowest RMSD between the observed ρ_{dark} and the retrieved ρ_{path} for the two closest fitting bands. Two settings for the DSF are implemented:

1. A fixed τ_a option, which computes a single τ_a for the scene or subscene if limits for a ROI are provided.
2. A tiled τ_a option, which divides the full scene in approximately 6x6 km tiles, retrieves τ_a per tile, and interpolates retrieved atmospheric parameters to the full scene.

An experimental "resolved" τ_a setting is also included, which treats each pixel individually. This option is currently only included for development purposes. Note that processing time is much longer for this option.

2.2.2. Glint correction

The DSF now can include a wind speed based air-water interface correction (see Vanhellemont, 2020c; Vanhellemont and Ruddick, 2021) that is disabled by default. For viewing geometries close to the principal sun glint spot this correction is rather sensitive to the wind speed, and may provide poor results when rather coarse ancillary wind data is used. High resolution imagery often has spatially resolved sun glint, with distinct patterns varying at short length scales, which cannot be corrected using such an approach. If SWIR channels are available, or NIR channels if the water is sufficiently clear, an estimate of the glint can be made and extrapolated to the shorter wavelength channels.

The DSF uses the bands and pixels giving the lowest estimate of aerosol optical thickness over the processed region or tile. This allows for an estimation of the atmospheric path reflectance that is relatively insensitive to sun glint. The sun glint signal will however still be present in the resulting surface reflectance. A sun glint correction option is included, which by default uses the SWIR bands to estimate the glint signal, as in Harmel et al. (2018). The implementation and performance is discussed in Vanhellemont (2019a). An alternative sun glint correction is provided, which estimates the average glint reflectance in a range of SWIR bands, and extrapolates this to the VNIR bands using a modeled reflectance shape. The alternative glint correction is currently only available when processing with "fixed" aerosol optical depth estimate.

2.3. EXP options

The EXP uses Rayleigh corrected reflectance in two bands to estimate the aerosol reflectance (ρ_{am}). By default the two SWIR bands are used, and ρ_{am} is estimated in the 2.2 μm band and extrapolated to the VIS and NIR bands using an exponential function and the ratio of 1.6 μm to 2.2 μm Rayleigh corrected reflectances.

The band set can be changed (see further) to include NIR/SWIR and red/NIR options, that both use a fixed ϵ due to non-zero ρ_w in at least one of the bands. The red/NIR option needs to assume a water reflectance model, for which by default the similarity spectrum is used. For the SWIR based correction, options are included for processing with variable and fixed ϵ and ρ_{am}. A comparison between DSF and EXP methods, showing the poor performance of the latter in the blue bands is discussed in Vanhellemont (2019a).

2.4. Outputs

The atmospheric correction procedure generates two NetCDF files containing reflectance data: (1) Level 1, top-of-the atmosphere reflectances (ρ_t, rhot_*), L1R and (2) Level 2, surface level reflectances (ρ_s, rhos_*), L2R. Based on these files the RGB composites are generated, and derived parameters are computed. Derived parameters are output to an L2W NetCDF file. Certain sensors will generate an additional L1R file for their higher spatial resolution panchromatic channel. TACT processing generates an additional L2T file. Optionally the datasets from the NetCDF files can be output to GeoTIFF files if the original input data were in a projection supported by GeoTIFF.

2.4.1. Ancillary data

The use of ancillary and SRTM DEM data requires the set up of an EarthData account (https://earthdata.nasa.gov). Your EarthData username and password have to be set in the operating system environment variables "EARTHDATA_u" and "EARTHDATA_p" respectively, or they can be provided in your ACOLITE settings file or config.txt file. The differences between using ancillary data or not were found to be very small (Vanhellemont, 2020c).

3. Processor overview

ACOLITE is distributed in a compiled binary format through the OD Nature/REMSEM page, and as Python source code through GitHub. ACOLITE can be used in GUI and CLI modes. The Python source code can also be imported into custom scripts. The GUI offers only limited configuration (file input/output, ROI setting, output parameters and loading/saving of settings files) and is aimed at new users. Full configuration can be done using plain text 'settings files', that are interpreted both by the GUI and CLI. The use of 'minimal' settings files is recommended, i.e. only adding the settings you want to change. Missing settings will be set to defaults automatically. It is recommended that users do not change the defaults settings file at config/defaults.txt but rather create a new processing settings file for a specific run. GUI and CLI use is described below for the binary distribution and Python source code.

3.1. ACOLITE GUI

Clickable launchers are provided for Linux/Mac (symlink to dist/acolite/acolite) and Windows (shortcut to dist/acolite/acolite.exe). On Linux you may need to select a terminal emulator for opening the acolite symlink. This can be done by right clicking and selecting Open With Other Application... The ACOLITE GUI can also be launched by opening a terminal emulator in the ACOLITE directory and running dist/acolite/acolite on Mac/Linux or dist/acolite/acolite.exe on Windows.
Figure 2 shows the main ACOLITE GUI window which offers the following configuration:

- Inputfile as directory: This box determines whether the inputfile to be selected using "Select input" is a single file (e.g. PlanetScope zipped bundle) or a directory (e.g. extracted Landsat or Sentinel-2 bundle).
- Select input: Select the input directory or file for processing. This should be either the directory containing the L1 scene data (e.g. a Landsat bundle or a Sentinel-2 .SAFE file) or a single input file.
- Select output: Select the output directory.
- Region of interest: Input the southern, northern, western and eastern boundaries (in decimal degrees) of the scene to be extracted for processing.
- Polygon: As alternative to providing SNWE boundaries, provide a polygon file with your region of interest (e.g. geojson). If both Region of interest and Polygon are empty the whole scene will be processed.
- L2W parameters: List the required output parameters here, see Section 5 for a full list. If empty, only L1R and L2R files will be generated.
- PNG outputs: Tick boxes for the generation of RGB composites using top-of-atmosphere (ρ_t) or (ρ_s) reflectances, and the generation of PNG maps of the requested L2W parameters.
- Save/Restore: Save the current settings as a text file, or restore settings from a text file.
- Run processing: Start the processing. The processing runs in a separate thread that can be stopped using the Stop processing or Exit buttons.
- Stop processing: Stop the processing.
- Exit: Quit the GUI.
- Logging output: Feedback from the GUI and processor is displayed here. Detailed logs should be output to a .txt file by the processor.

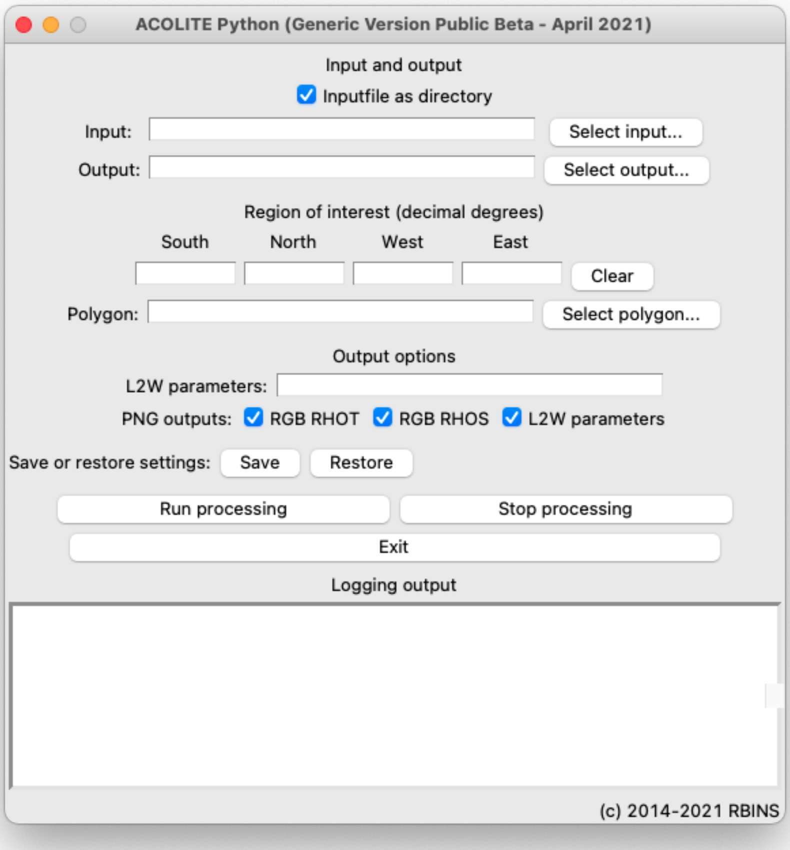

Figure 2: Screenshot from the Public Beta of the Generic ACOLITE Python (April 2021) running on Mac OS Big Sur.

3.2. ACOLITE CLI

The ACOLITE command line interface (CLI) can be launched by opening a terminal emulator and launching acolite with the –cli flag and the full path to a settings file. The following examples assume you are in the main ACOLITE directory, but ACOLITE can be launched from anywhere using its full path.

Linux and Mac:

```
dist/acolite/acolite --cli --settings=settings_file
```

Windows:

```
dist\acolite\acolite.exe --cli --settings=settings_file
```

Source code:

```
python launch_acolite.py --cli --settings=settings_file
```

Optionally a comma separated list of images (with their full paths, escape spaces with a backslash or use quotation masks around the list if needed) or a text file with paths to the images can be provided with the –inputfile flag:

Linux and Mac:

```
dist/acolite/acolite --cli --settings=settings_file --inputfile="file1,file2,file3"
```

Windows:

```
dist\acolite\acolite.exe --cli --settings=settings_file --inputfile="file1,file2,file3"
```

Source code:

```
python launch_acolite.py --cli --settings=settings_file --inputfile="file1,file2,file3"
```

The ACOLITE CLI can also be used to retrieve the LUTs required for a list of sensors before running any processing. E.g. for setting up a Docker container. In the following examples "python launch_acolite.py" can also be replaced by "dist/acolite/acolite" or "dist\acolite\acolite.exe" for the binary version.

All supported multispectral sensors:

```
python launch_acolite.py --retrieve_luts
```

Get the LUT for named sensors (here S2A/MSI and S2B/MSI):

```
python launch_acolite.py --retrieve_luts --sensor S2A_MSI,S2B_MSI
```

Get the generic LUT for processing hyperspectral sensors:

```
python launch_acolite.py --retrieve_luts --sensor hyper
```

3.3. *Inputfiles*

ACOLITE can process top-of-atmosphere imagery from a series of satellite sensors. Since the formatting by the commercial operators or space agencies is different for each sensor, the inputfile needs to be provided appropriately. Inputfiles can be either single files (e.g. CHRIS .hdf files or zipped Planet bundles) or directories (e.g. Sentinel .SAFE bundles or extracted Planet bundles). In this section the inputfile options are listed per sensor.

– **CHRIS** Provide the full path to a CHRIS .hdf file

– **Landsat 5/7/8** Provide the full path to the extracted bundle (i.e. extract the .tar archive). Make sure you have a L1 TOA bundle, not a L2 SR bundle. Level 1 data from both Collection1 and Collection2 are supported.

– **PlanetScope / RapidEye** Provide the path to a zipped file or unzipped bundle. Make sure you have a L1 TOA file (i.e. AnalyticMS.tif or AnalyticMS_clip.tif files) and not a SR file.

– **Pléiades-1 A/B** Provide the path to an unzipped bundle that contains the IMG_PHR1A_MS/IMG_PHR1B_MS directories.

– **PRISMA** Provide the path to the .he5 file (PRS_L1_STD) that also contains the official L2 file (PRS_L2C_STD). The L2 file does not have to be specified in your settings file, it will be automatically detected.

– **Sentinel-2** Provide the full path to the extracted .SAFE file. Make sure you have a L1C TOA bundle, not a L2 SR bundle.

– **Sentinel-3** Provide the full path to the extracted .SAFE file. Make sure you have a L1 TOA bundle (S3A_OL_1_ERR or S3A_OL_1_EFR for S3A), not a L2 SR bundle (e.g. S3A_OL_2_WFR or S3A_OL_2_LFR).

– **SPOT 6/7** Provide the path to the "VOL" directory that contains the "IMG" directory (e.g. IMG_SPOT6_PMS_001_A or IMG_SPOT7_PMS_001_A). Note that this may be a few levels deep in the extracted files.

– **Venμs** Provide the path to the extracted zip file that contains the main .tif and .xml files.

– **WorldView-2/3** Provide the path to an unzipped bundle that contains the main .TIF and .XML files.

3.4. *Example settings files*

Settings files can be generated with the ACOLITE GUI, or with any text editor. Use UTF-8 encoding for best results. It is recommended to use a settings file that only includes the settings you want to change. A minimum settings file contains the full paths to the inputfile to be processed and the output directory. Processing options not provided in the settings file will be set to the defaults automatically. Several examples are given below.

A settings file that will output an area around the port of Zeebrugge in the specified directory, with water-leaving radiance reflectance in all bands output in the L2W file:

```
## Example settings for Zeebrugge
inputfile=/storage/Input/LC08_L1TP_199024_20210607_20210615_02_T1
output=/storage/Output/Zeebrugge
limit=51.3,3.0,51.4,3.25
l2w_parameters=rhow_*
```

A settings file to output Sentinel-2 data resampled to a 60 m grid for the same region of interest, increasing the masking threshold, apply a glint correction, and exporting the L2W datasets as GeoTIFF files:

```
## Example settings for Zeebrugge 60 m and GeoTIFF
inputfile=/storage/Input/S2B_MSIL1C_20200922T104649_N0209_R051_T31UES_20200924T124918.SAFE
output=/storage/Output/Zeebrugge
limit=51.3,3.0,51.4,3.25
l2w_parameters=rhow_*,t_nechad
## output resolution (S2 only 10, 20, or 60 m)
s2_target_res=60
## increase default L2W masking threshold
l2w_mask_threshold=0.05
## disable wind speed based interface reflectance correction
dsf_interface_reflectance=False
## residual glint correction
dsf_residual_glint_correction=True
## output L2W to GeoTIFF
l2w_export_geotiff=True
```

A settings file for a larger region of interest, merging four Sentinel-2 tiles. Note that the scenes are listed on a single line, as a comma separated string:

```
## Example settings for Zeebrugge 60 m merging
inputfile=/storage/Input/S2B_MSIL1C_20200922T104649_N0209_R051_T31UET_20200924T124918.SAFE,
/storage/Input/S2B_MSIL1C_20200922T104649_N0209_R051_T31UES_20200924T124918.SAFE,
/storage/Input/S2B_MSIL1C_20200922T104649_N0209_R051_T31UDT_20200924T124918.SAFE,
/storage/Input/S2B_MSIL1C_20200922T104649_N0209_R051_T31UDS_20200924T124918.SAFE
output=/storage/Output/ZeebruggeMerged
limit=51.3,2.90,51.65,3.35
l2w_parameters=rhow_*,t_nechad
## output resolution (S2 only 10, 20, or 60 m)
s2_target_res=60
## files given in inputfile will be merged (if from same sensor/orbit)
merge_tiles=True
```

A settings file for a larger region of interest, merging four Sentinel-2 tiles from two different UTM zones (two from zone 33, two from zone 32). Note that the zone from the first tile (in this case 33) will be used as reference to reproject the other tiles before merging:

```
## Example settings for Po 60 m merging
inputfile=/storage/Input/S2A_MSIL1C_20210331T100021_N0300_R122_T33TUL_20210331T113321.SAFE,
/storage/Input/S2A_MSIL1C_20210331T100021_N0300_R122_T33TUK_20210331T113321.SAFE,
/storage/Input/S2A_MSIL1C_20210331T100021_N0300_R122_T32TQR_20210331T113321.SAFE,
/storage/Input/S2A_MSIL1C_20210331T100021_N0300_R122_T32TQQ_20210331T113321.SAFE
output=/storage/Output/PoMerged
limit=44.6,11.9,45.3,13.1
s2_target_res=60
merge_tiles=True
```

The L1R.nc file from a previous run can also be used as an inputfile, e.g. the L1R output from the previous run:

```
## Example settings processing merged L1R file
inputfile=/storage/Output/PoMerged/S2A_MSI_2021_03_31_10_08_25_merged_L1R.nc
output=/storage/Output/NetCDF
l2w_parameters=chl_re_gons,t_nechad
map_l2w=True
```

3.5. *Installation from source*

The Python source code is available on GitHub (https://github.com/acolite/acolite/) and includes the latest updates. The GitHub repository can be downloaded through a browser or cloned using git:

```
git clone --depth 1 https://github.com/acolite/acolite
```

To use the source code a Python 3 environment is needed with the following packages (and their dependencies):

```
numpy matplotlib scipy gdal pyproj scikit-image pyhdf pyresample netcdf4 h5py
```

Certain experimental features may require additional packages to be installed ("astropy cartopy").

A suitable Python environment can be set up using conda (either from miniconda or anaconda) and the packages on conda-forge:

```
conda create -n acolite -c conda-forge python=3
conda activate acolite
conda install -c conda-forge numpy matplotlib scipy gdal pyproj scikit-image pyhdf pyresample netcdf4 h5py
```

Activate the "acolite" environment before running ACOLITE:

```
conda activate acolite
```

ACOLITE can then be launched using Python, with optional command line options as described above:

```
python acolite/launch_acolite.py
```

ACOLITE can be imported in other Python scripts or Jupyter notebooks, e.g. when the git clone is in a git folder in your user home directory:

```
import sys, os, time
user_home = os.path.expanduser("~")
sys.path.append(user_home+"/git/acolite")
import acolite as ac
```

The processing can then be integrated in other scripts, e.g. by using a Python dict to provide settings:

```
settings = {"inputfile":"S2B_MSIL1C_20200922T104649_N0209_R051_T31UES_20200924T124918.SAFE",
"output":"~/Output/",
"s2_target_res":60,
"dsf_residual_glint_correction":True}

ac.acolite.acolite_run(settings=settings)
```

3.6. Moving the "data" directory

The data directory can be moved to a different location if the file config/config.txt is edited appropriately to include the full path to the data directories. By default these directories are inside the acolite folder, and relative paths are used:

```
## Data directory
data_dir=data

## atmospheric correction LUT data directory
lut_dir=data/LUT

## DEM SRTM HGT files
hgt_dir=data/SRTMGL3.003

## MET files
met_dir=data/MET
```

3.7. Setting up TACT processing

Running TACT requires libRadtran to be installed and compiled to simulate atmospheric upwelling and downwelling radiances as well as transmittances (Vanhellemont, 2020a). Inputs to libRadtran are atmospheric profiles from the ERA5 model that are retrieved from the Research Data Archive (RDA) at the University Corporation for Atmospheric Research (UCAR).

3.7.1. libRadtran

Please download libRadtran and follow compilation instructions at http://libradtran.org/doku.php TACT was successfully tested with libRadtran version 2.0.2 on Mac and Linux. The ACOLITE configuration file at acolite/config/-config.txt needs to be edited to include the full path to the libRadtran directory on your system in the libradtran_dir setting.

3.7.2. RDA access

And account with RDA/UCAR is required for accessing atmospheric profiles, click "Register now" at the very top of the page at (https://rda.ucar.edu/). These credentials need to be known when accessing the RDA machine, and they have to be added to your .netrc file (e.g. nano $HOME/.netrc):

```
machine rda.ucar.edu
login ADD_LOGIN_HERE
password ADD_PASSWORD_HERE
```

Your .dodsrc (e.g. nano $HOME/.dodsrc) file needs to include a full path to your .netrc file:

```
HTTP.NETRC=/path/to/.netrc
```

Note that the use of the $HOME environment variable does not work in the .dodsrc file, and that a full path needs to be given. This can be found e.g. using "find $HOME/.netrc"

3.8. Installation in Docker container

ACOLITE can be installed in a Docker container for deploying and running the processing in parallel. The example here is based on the Ubuntu 20.04 LTS image and uses conda and conda-forge for Python environment and package management.

Make and enter a docker directory, and create input and output directories:

```
mkdir acolite-docker
cd acolite-docker
mkdir input
mkdir output
```

Create an empty settings file ("nano settings") and set the following dummy settings:

```
output=output/
inputfile=input/
```

Create a Dockerfile ("nano Dockerfile"), and paste the setup code from the following page.

Add the sensors you want supported to the retrieve_luts line. Use a comma separated list from the following supported sensors: S2A_MSI, S2B_MSI, L8_OLI, L5_TM, L7_ETM, S3A_OLCI, S3B_OLCI, RapidEye, SPOT6, SPOT7, PHR1A, PHR1B, VENμS_VSSC, WorldView2, WorldView3, PlanetScope_0c, PlanetScope_0d05, PlanetScope_0d06, PlanetScope_0e, PlanetScope_0f, PlanetScope_22

After creating the Dockerfile, build the ACOLITE Docker:

```
docker build -t acolite
```

Once this is done, create another settings file (e.g. "nano user_settings.txt") with your processing settings:

```
## ACOLITE user settings for Docker
output=output/Zeebrugge
inputfile=input/Data/31UES/S2A_MSIL1C_20210329T105631_N0209_R094_T31UES_20210329T130721.SAFE
output=/output/Zeebrugge
limit=51.30,3.10,51.42,3.29
```

Run ACOLITE Docker by mounting the path to the directory containing your data (in this case "Data" is a directory in "/home/quinten/Sentinel-2/") as input and a local output directory as output. The example below uses environment variables for input/output/settings for clarity:

```
input=/home/quinten/Sentinel-2/
output=/home/quinten/Output/
settings=/home/quinten/user_settings.txt
docker run -d -v $input:/input -v $output:/output -v $settings:/settings acolite
```

ACOLITE Dockerfile

```
## Testing deployment of ACOLITE GitHub code in a docker image
## QV 2021-01-20
## updated 2021-07-15 for generic ACOLITE
## updated 2021-07-26 for retrieving LUTs

FROM ubuntu:20.04

# System packages
RUN apt-get update && apt-get install -y curl
RUN apt-get install -y git

# Install miniconda to /miniconda
RUN curl -LO http://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
RUN bash Miniconda3-latest-Linux-x86_64.sh -p /miniconda -b
RUN rm Miniconda3-latest-Linux-x86_64.sh
ENV PATH=/miniconda/bin:${PATH}
RUN conda update -y conda

# Create acolite environment
RUN conda create -n acolite -c conda-forge python=3 numpy matplotlib scipy gdal pyproj astropy cartopy scikit-image pyhdf pyresample netcdf4
RUN conda init bash
RUN echo "conda activate acolite" > ~/.bashrc

## clone acolite
RUN git clone --depth 1 https://github.com/acolite/acolite

## get LUTs
RUN conda run -n acolite python ./acolite/launch_acolite.py --retrieve_luts --sensor hyper
RUN conda run -n acolite python ./acolite/launch_acolite.py --retrieve_luts --sensor S2A_MSI,S2B_MSI,L8_OLI

# add input, output and settings mount points/files
ADD ./input /input
ADD ./output /output
ADD ./settings settings

# run processing with --cli
## for processing files in mounted input listed in settings file e.g. inputfile=/input/scene1.zip
ENTRYPOINT ["conda", "run", "-n", "acolite", "python", "./acolite/launch_acolite.py", "--cli", "--settings", "settings"]

## change output ownership
CMD ["chown", "-R", "1000:1000", "/output"]
```

4. Processing options

This section lists most of the processing options that can be specified in a user generated settings file. Overall default settings are given in config/defaults.txt and in files per sensor in config/defaults.

4.1. General options

inputfile=
: Scene or comma separated list of scenes to be processed. Full paths to the scene directory (Landsat bundle or Sentinel-2 .SAFE file) have to be specified. The comma separated list has to be on a single line. Compressed files (e.g. from EarthExplorer) have to be extracted.

output=
: Directory where outputs will be generated. Will be created if it does not exist.

limit=
: Comma separated list of the bounding box coordinates of the region of interest in decimal degrees in S,W,N,E order. If empty, and polygon=None the full scene will be processed.

polygon=None
: A polygon file (e.g. geojson) specifying the bounding box for the region of interest. If None, and limit is empty, the full scene will be processed.

region_name=
: A name for the region of interest that will be appended to the output files (if not empty).

merge_tiles=False
: Controls whether inputfiles have to be merged before processing. Merging is only supported for scenes from the same orbit, and when a region of interest is specified. Options: True/False

merge_zones=True
: Controls whether inputfiles from different UTM zones have to be merged before processing. If False, a file for each UTM zone will be generated. Options: True/False

extend_region=False
: Controls whether the output file has to cover the entire limit or polygon, or whether it can be limited to the scene extent. Options: True/False

atmospheric_correction=True
: Whether to apply an atmospheric correction of not. Options: True/False

aerosol_correction=dark_spectrum
: The aerosol correction (DSF or EXP) to use. See below for algorithm specific options. Options: dark_spectrum, exponential

l2w_parameters=
: Comma separated list of the output parameters to be computed. See section 5 for a full list. If empty, only L1R and L2R files will be generated.

resolved_geometry=False
: Use per pixel resolved view/illumination geometry grids if available.

runid=
: An identifier for the current processing run. An identifier based on date/time is generated if empty.

4.1.1. Gains

gains=False
: Controls the application of sensor specific TOA gains. Options: True/False

gains_toa=None
: Comma separated list of gains to be used for the currently processed sensor. If not None, there has to be a comma separated list of values, one per band. Defaults can also be set per sensor in the sensor specific default files.

4.1.2. Masking

l2w_mask=True
Controls whether the L2W output parameters will be masked. Options: True/False

l2w_mask_wave=1600
Wavelength (nm) for the non-water masking. The closest band will be selected.

l2w_mask_threshold=0.0215
Threshold for the non-water masking. Pixels with ρ_t in the masking band above this threshold will be masked.

l2w_mask_water_parameters=True
Controls whether water specific parameters are to be masked. Options: True/False

l2w_mask_negative_rhow=True
Controls whether pixels with negative reflectances are to be masked. Options: True/False

l2w_mask_negative_wave_range=400,900
Wavelength range to check for negative reflectances.

l2w_mask_cirrus=True
Enable masking of cirrus clouds using a band around 1375 nm. OLI and MSI only. Options: True/False

l2w_mask_cirrus_threshold=0.005
Threshold for the cirrus masking. Pixels with ρ_t in the cirrus band above this threshold will be masked.

l2w_mask_cirrus_wave=1373
Wavelength for the cirrus masking. The closest band within 5 nm of this wavelength will be used.

l2w_mask_high_toa=True
Mask pixels with high (> l2w_mask_high_toa_threshold) top-of-atmosphere reflectance in any band. Options: True/False

l2w_mask_high_toa_threshold=0.3
Threshold for TOA reflectance masking.

l2w_mask_smooth=True
Apply Gaussian smoothing to the computed L2W mask. This option can reduce speckled masks especially in Landsat 5 and 7. Options: True/False

l2w_mask_smooth_sigma=3
Kernel size for the Gaussian smoothing of the L2w mask.

flag_exponent_swir=0
Bit index that will be used to store the SWIR threshold masking step (2**0).

flag_exponent_cirrus=1
Bit index that will be used to store the cirrus masking step (2**1).

flag_exponent_toa=2
Bit index that will be used to store the top-of-atmosphere threshold masking step (2**2).

flag_exponent_negative=3
Bit index that will be used to store the negative retrievals masking step (2**3).

flag_exponent_outofscene=4
Bit index that will be used to store the out-of-scene masking step (2**4).

4.1.3. Ancillary data

ancillary_data=True
Controls whether ancillary data is to be used for retrieving ozone, water vapour and atmospheric pressure. Note that for retrieving ancillary data an EarthData account is required. Options: True/False

EARTHDATA_u=
Username for your EarthData account. Can also be set as environment variable.

EARTHDATA_p=
Password for your EarthData account. Can also be set as environment variable.

uoz_default=0.3

Fixed/fallback value for ozone concentration in cm^{-1}. This value will be used if the ancillary data is disabled or unavailable and gas_transmittance=True.

uwv_default=1.5

Fixed/fallback value for water vapour concentration in g/cm^2. This value will be used if the ancillary data is disabled or unavailable and gas_transmittance=True.

pressure=None

Fixed atmospheric pressure value. Defaults to normal pressure (1013.25 mbar) if equal to None and if no ancillary data is used.

elevation=None

Elevation to compute atmospheric pressure according to a standard atmosphere.

dem_pressure=False

Controls whether pressure is to be computed from DEM elevation. Options: True/False

dem_pressure_resolved=True

Controls whether pressure from DEM elevation is reprojected and computed per pixel or whether a scene percentile is used. Options: True/False

dem_pressure_percentile=25

Determine pressure from this percentile of DEM in the ROI/scene.

dem_pressure_write=True

Controls whether the DEM derived pressure dataset is output to the L2R NetCDF file. Options: True/-False

4.1.4. Various processing

min_tgas_aot=0.85

Minimum gas transmittance required to use a band for aerosol optical thickness retrieval.

min_tgas_rho=0.75

Minimum gas transmittance required to retrieve surface reflectance for a band.

blackfill_skip=True

Skip a region of interest when it is outside the sensor swath or in the "blackfill" around a projected satellite image, e.g. the empty borders around Landsat data. Options: True/False

blackfill_max=1.0

Fraction of "blackfill" in the region of interest to skip processing. For example, a value of 1.0 will skip scenes if the region of interest is fully outside the satellite swath, a value of 0.5 will allow up to 50% of the region of interest to be outside the swath.

blackfill_wave=1600

Band in which to check "blackfill" cover, as identified by wavelength.

*copy_datasets=lon,lat,rhot_**

Datasets that are copied from L1R to L2R NetCDF files. Editing this may break L2W generation.

luts=ACOLITE-LUT-202102-MOD1,ACOLITE-LUT-202102-MOD2

LUTs to use in processing. By default MOD1 (continental) and MOD2 (maritime) from 6SV are included. Can be reduced to a single model if preferred by the user.

4.2. Output options

rgb_rhot=True

Controls the output of RGB PNG files based on ρ_t data. Options: True/False

rgb_rhos=True

Controls the output of RGB PNG files based on ρ_s data. Options: True/False

map_l2w=False

Controls the output of PNG files from the requested L2W parameters. Options: True/False

output_geolocation=True

Controls the output of longitude and latitude to the NetCDF files. Options: True/False

output_xy=False

Controls the output of x and y easting and northing to the NetCDF files. Options: True/False

output_geometry=True
Controls the output of resolved view and illumination geometry (if available) to the NetCDF files. Options: True/False

output_rhorc=False
Controls the output of Rayleigh corrected reflectance to the NetCDF files. Will also be set if any rhorc_* output is listed in l2w_parameters. Options: True/False

output_bt=False
Controls the output of at-sensor brightness temperatures for the Landsat thermal bands to the NetCDF files. Will also be set if any bt* output is listed in l2w_parameters. Options: True/False

l1r_export_geotiff=False
Controls the output of L1R GeoTIFF files. Options: True/False

l2r_export_geotiff=False
Controls the output of L2R GeoTIFF files. Options: True/False

l2t_export_geotiff=False
Controls the output of L2T GeoTIFF files. Options: True/False

l2w_export_geotiff=False
Controls the output of L2W GeoTIFF files. Options: True/False

4.3. *Various*

slicing=False
Controls slicing of data to finite values during processing, should speed up and improve memory use for sparse (or custom) images. Options: True/False

verbosity=5
Controls the amount of detail in the terminal outputs during processing. (Verbosity levels are not consistently implemented at the moment.)

4.4. *DSF options*

dsf_aot_estimate=fixed
Whether to use the tiled or fixed DSF. An experimental resolved setting was also added. Options: fixed, tiled, resolved

dsf_wave_range=400,900
Comma separated wavelength range in nm of bands to use in the DSF, e.g. for including (400,2500) or excluding (400,900) the SWIR bands from the fitting. Default: 400,900

dsf_exclude_bands=,
Comma separated list of bands to exclude in the DSF. Names have to match names in the relative spectral response file of the sensor.

dsf_interface_reflectance=False
Controls the air-water interface reflectance correction. Disabled by default at the moment since the surface reflectance is quite sensitive to (local) wind speed in nadir viewing conditions. Options: True/False

dsf_interface_option=default
Method to use for the air-water interface reflectance correction. Currently only for all pixels with no distinction between water and non-water. Options: "default" for the OSOAA (Chami et al., 2015) based "sky" reflectance presented in (Vanhellemont, 2020c; Vanhellemont and Ruddick, 2021)

dsf_interface_lut=RSKY-202102
Version of the LUT to use for the air-water interface correction.

wind=None
Force the wind speed (m/s) for the air-water interface correction. If None, the default value will be used or the value retrieved from the ancillary data.

wind_default=None
Default wind speed (m/s) for the air-water interface correction.

dsf_write_aot_550=False

Controls output of a per pixel (interpolated or resampled) aerosol optical thickness dataset. Note that in fixed processing this is also output as a single float in the NetCDF global attributes. Options: True/False

4.4.1. Fixed DSF options

dsf_fixed_aot=None

User specified aerosol optical thickness value at 550 nm. Options: None or float value

dsf_fixed_lut=ACOLITE-LUT-202102-MOD2

User specified aerosol model for the dsf_fixed_aot option. Options: ACOLITE-LUT-202102-MOD1 or ACOLITE-LUT-202102-MOD2

4.4.2. Tiled DSF options

dsf_tile_dimensions=None

Comma separated tile size in pixels for the tiled DSF processing. Defaults per sensor are set in config/defaults/.

dsf_min_tile_cover=0.10

Minimum coverage of valid pixels in the individual tiles. Tiles with less coverage (typically at the swath edges) will be skipped and interpolated. Options: fraction between 0 and 1

dsf_min_tile_aot=0.01

Minimum aerosol optical thickness (at 550 nm) in a tile. Tiles with lower τ_a will be skipped and interpolated. Typically very low τ_a indicate cloud shadow contamination.

dsf_write_tiled_parameters=False

Controls the output of atmospheric parameters to the NetCDF file. Options: True/False

4.4.3. Glint correction options

dsf_residual_glint_correction=False

Controls the residual glint correction. Options: True/False

dsf_residual_glint_correction_method=default

Residual glint correction method to use. Options: default/alternative

dsf_residual_glint_wave_range=1500,2400

Wavelength range for the alternative residual glint correction method.

glint_force_band=None

Specify the wavelength (nm) where glint reflectance ρ_g is to be estimated. If None, then in each pixel the SWIR band giving the lowest ρ_g at 440 nm will be used.

glint_mask_rhos_band=1600

The wavelength (nm) to determine which pixels need to be glint corrected. The closest band will be selected.

glint_mask_rhos_threshold=0.05

Threshold to determine which pixels need to be glint corrected. Pixels with glint_mask_rhos_band > the threshold will not be corrected. (It is noted that ρ_g at 1.6 μm can be larger than 0.05.)

glint_write_rhog_ref=False

Controls the output of the reference glint reflectance to the NetCDF file. Options: True/False

glint_write_rhog_all=False

Controls the output of the glint reflectance in each band to the NetCDF file. Options: True/False

4.4.4. DSF development options

These are specialised development options which should not generally be changed by the user.

dsf_spectrum_option=intercept

Method to choose the ρ_{dark}. Options: darkest, percentile, intercept

dsf_percentile=1

Percentile to use for the ρ_{dark} percentile selection.

dsf_intercept_pixels=1000
Number of pixels to use for the ρ_{dark} intercept selection.

dsf_model_selection=min_drmsd
Method to choose which aerosol model to use. Options: min_drmsd

dsf_filter_toa=False
Apply percentile filter to top-of-atmosphere data for retrieving the ρ_{dark}. Options: True/False

dsf_filter_percentile=5
Percentile for the filtering of top-of-atmosphere data for retrieving the ρ_{dark}.

dsf_filter_box=5,5
Box size in pixels for the filtering of top-of-atmosphere data for retrieving the ρ_{dark}.

dsf_filter_aot=False
Filter aerosol optical thickness after retrieval using the previous two settings. Options: True/False

dsf_smooth_aot=True
Apply Gaussian smoothing to the retrieved aerosol optical thickness in resolved processing mode. Does not work in binary version due to PyInstaller astropy issue. Options: True/False

dsf_smooth_box=20,20
Box size in pixels for the smoothing of the resolved aerosol optical thickness.

4.5. EXP options

exp_wave1=1600
First wavelength (nm) for the EXP correction. The closest band will be selected.

exp_wave2=2200
Second wavelength (nm) for the EXP correction. The closest band will be selected.

exp_alpha=None
Ratio of the water reflectances in the two used bands. Only used if no SWIR band is used. If None, the similarity spectrum (Ruddick et al., 2006) is used.

exp_alpha_weighted=True
Controls the band weighting of the similarity spectrum alpha. Options: True/False

exp_epsilon=None
Fixed aerosol epsilon value in the two used bands. If None, the value is determined from the scene according to Vanhellemont and Ruddick (2015).

exp_gamma=None
Fixed transmittance ratio value in the two used bands. If None, the ratio of Rayleigh transmittances is used.

exp_fixed_epsilon=True
Controls whether a fixed or per pixel (only for SWIR/SWIR) epsilon is used. Options: True/False

exp_fixed_epsilon_percentile=50
Percentile to determine the fixed epsilon over the (sub)scene. Options: value between 0 and 100

exp_fixed_aerosol_reflectance=True
Controls whether the aerosol reflectance is fixed over the (sub)scene. Options: True/False

exp_fixed_aerosol_reflectance_percentile=5
Percentile to determine the fixed aerosol reflectance over the (sub)scene. Options: value between 0 and 100

exp_swir_threshold=0.0215
Non-water threshold for the SWIR1 band (at 1.6 μm), used to determine the scene wide fixed epsilon.

exp_output_intermediate=False
Determines output of intermediate datasets (i.e. per pixel epsilon and rhoam) from the exponential processing. Options: True/False

4.6. Sensor specific options

4.6.1. Landsat 8/OLI

l8_orange_band=True
: Controls the output of the orange contraband (Castagna et al., 2020). Options: True/False

4.6.2. Sentinel-2/MSI

s2_target_res=10
: Output resolution of Sentinel-2 products in meter. Options: 10, 20, 60

geometry_type=grids_footprint
: Method to derive per pixel geometry data for S2. Options: "grids" for simple interpolation of the 5x5 km grids, "grids_footprint" for extrapolation of the 5x5 km grids and subsetting per detector (recommended), and "gpt" use SNAP to compute the geometry. (Configure "snap_directory" in acolite/config/config.txt)

geometry_res=60
: Resolution at which to compute the per pixel geometry data. Geometry will be resampled to s2_target_res, so they can be different values. Computing the geometry at 60 m is generally enough (and faster!). Options: 10, 20, 60

geometry_per_band=False
: Compute per band rather than band average geometry. This is only important in glint viewing conditions, and then the ancillary wind speed estimate is probably not accurate enough. Can generally be left False. Only for "grids_footprint" option. Options: True/False.

geometry_fixed_footprint=False
: Determines whether band specific footprint polygons or fixed footprint polygons are used for the computation of per pixel view geometry. Only for "grids_footprint" option. Options: True/False.

4.6.3. Sentinel-3/OLCI

smile_correction=True
: Apply SMILE correction as in Bourg et al. (2008). Options: True/False

use_tpg=True
: Use longitude and latitude tie point grids for computing region of interest subsetting. Options: True/False

4.6.4. Pléiades

pleiades_skip_pan=False
: Skip Pléiades panchromatic band during processing. Much faster since the panchromatic data (16x the number of pixels compared to multispectral bands!) does not need to be read or written. Options: True/False

4.6.5. Worldview

inputfile_swir=None
: Supply WorldView SWIR bundle that corresponds to the VNIR bundle for simultaneous processing.

4.6.6. CHRIS

chris_interband_calibration=True
: Apply CHRIS top-of-atmosphere interband calibration from Lavigne et al. (2021). Options: True/False

chris_noise_reduction=True
: Apply CHRIS noise reduction based on Gómez-Chova et al. (2008). Options: True/False

4.6.7. PRISMA

prisma_output_lt=False
: Whether to output PRISMA top-of-atmosphere radiances. Options: True/False

4.7. TACT settings

tact_run=False

Controls whether the Thermal Atmospheric Correction Tool (TACT) should be run for Landsat thermal data (L5/TM B6, L7/ETM B6, L8/TIRS B10, B11). Options: True/False

tact_output_atmosphere=False

Controls whether the simulated atmospheric parameters from the Thermal Atmospheric Correction Tool (TACT) should be output to the NetCDF file. Options: True/False

tact_output_intermediate=False

Controls whether the intermediate parameters (at-sensor brightness temperature, radiance, surface radiance, surface emissivity) from the Thermal Atmospheric Correction Tool (TACT) should be output to the NetCDF file. Options: True/False

tact_map=True

Controls whether the TACT outputs should be output as PNG maps. Options: True/False

4.8. General mapping options

map_title=True
: Controls the title on PNG maps. The title consists of the parameter, satellite sensor and date. Options: True/False

map_projected=False
: Use cartopy for mapping. Does not work properly yet, does not work at all in binary version. Datasets will be reprojected and some distortion may occur. Options: True/False

map_raster=False
: Controls whether maps are generated using matplotlib or pillow. Raster maps are 1:1 and have no annotations. Options: True/False

map_pcolormesh=False
: Use pcolormesh rather than imshow for mapping using matplotlib. Options: True/False

map_dpi=300
: DPI of the output maps.

map_ext=png
: File type of the output maps.

map_limit=None
: Crop output maps to this limit (S,W,N,E).

4.8.1. RGB mapping options

rgb_red_wl=660
: The wavelength (nm) for the Red channel in the RGB composite. The band with closest wavelength will be selected.

rgb_green_wl=560
: The wavelength (nm) for the Green channel in the RGB composite. The band with closest wavelength will be selected.

rgb_blue_wl=480
: The wavelength (nm) for the Blue channel in the RGB composite. The band with closest wavelength will be selected.

rgb_min=0.0,0.0,0.0
: Comma separated minimum reflectance for linear scaling of the R,G,B channels.

rgb_max=0.15,0.15,0.15
: Maximum reflectance for linear scaling of the R,G,B channels.

4.8.2. L2W mapping options

map_colorbar=True
: Controls the plotting of colour bars for the L2W PNG maps. Options: True/False

map_colorbar_orientation=vertical
: Controls the orientation of the colour bars. Options: horizontal/vertical

map_auto_range=False
: Controls the auto ranging for the L2W maps. If False the ranges in config/parameter_labels.txt will be used. Options: True/False

map_auto_range_percentiles=1,99
: Lower and upper percentiles for the colour bar auto ranging.

map_outrange=False
: Also fill the pixels that are out of the colour bar range. If False the out of range pixels will use the colour from the respective end of the colour bar. Options: True/False

map_fillcolor=LightGrey
: Controls the fill colour in the L2W maps. Options: any Python colour name or Hex Color Code

map_default_colormap=viridis
: Default colormap to use if dataset is not configured in config/parameter_labels.txt For list of options see matplotlib documentation: https://matplotlib.org/stable/tutorials/colors/colormaps.html

5. L2W algorithms

ACOLITE includes several algorithms for the retrieval of parameters derived from reflectance. The L2R file is used as an input and an L2W file is generated containing the requested datasets. The L2W file will be used to generate maps from the parameters if requested. The map scaling for each parameter is defined in the config/acolite_l2w_labels.txt file. The included output products and algorithms are:

– **rhot_***: ρ_t, the top-of-atmosphere reflectance as derived from the original input file. A wavelength can be specified, or the asterisk will be expanded to include all available wavelengths. Sensors: all sensors

– **rhos_***: ρ_s, the surface reflectance. A wavelength can be specified, or the asterisk will be expanded to include all available wavelengths. Sensors: all sensors

– **rhow_***: ρ_w, the surface reflectance for water pixels, with a SWIR based mask applied to non-water pixels. A wavelength can be specified, or the asterisk will be expanded to include all available wavelengths. Sensors: all sensors

– **Rrs_***: the remote sensing reflectance (sr^{-1}) for water pixels, Rrs = ρ_w / π. A wavelength can be specified, or the asterisk will be expanded to include all available wavelengths. Sensors: all sensors

– **rhorc_***: ρ_{rc}, the Rayleigh corrected reflectance. This is the ρ_t with ρ_r removed and corrected for two way Rayleigh transmittance. A wavelength can be specified, or the asterisk will be expanded to include all available wavelengths. Sensors: all sensors

– **spm_nechad**: Suspended Matter Concentration (gm^{-3}) using the algorithm of Nechad et al. (2010). By default the red band will be used, but a wavelength can be specified in the parameter (e.g. spm_nechad_833 for S2B) or one of the aliases can be used for the red or NIR bands (spm_nechad_red, spm_nechad_nir). The parameters from the closest wavelength from the hyperspectral dataset will be used. Sensors: all sensors

– **spm_nechad2016**: Suspended Matter Concentration (gm^{-3}) using the algorithm of Nechad et al. (2010) recalibrated by Bouchra Nechad in September 2016, specifically for Landsat 8 and Sentinel-2. By default the red band will be used, but a wavelength can be specified in the parameter (e.g. spm_nechad2016_833 for S2B) or one of the aliases can be used for the red or NIR bands (spm_nechad2016_red, spm_nechad2016_nir). Sensors: L8/OLI, S2A/MSI, S2B/MSI

– **spm_nechadave**: Suspended Matter Concentration (gm^{-3}) using the algorithm of Nechad et al. (2010). By default the red band will be used, but a wavelength can be specified in the parameter (e.g. spm_nechadave_833 for S2B) or one of the aliases can be used for the red or NIR bands (spm_nechadave_red, spm_nechadave_nir). The hyperspectral dataset will be resampled to the band relative spectral response. Sensors: all sensors **Warning: not recommended for bands stretching outside the algorithm's optimal spectral range.**

– **tur_nechad**: Turbidity (FNU) using the algorithm of Nechad et al. (2009). By default the red band will be used, but a wavelength can be specified in the parameter (e.g. tur_nechad_833 for S2B) or one of the aliases can be used for the red or NIR bands (tur_nechad_red, tur_nechad_nir). The parameters from the closest wavelength from the hyperspectral dataset will be used. Sensors: all sensors

– **tur_nechad2016**: Turbidity (FNU) using the algorithm of Nechad et al. (2009) recalibrated by Bouchra Nechad in September 2016, specifically for Landsat 8 and Sentinel-2. By default the red band will be used, but a wavelength can be specified in the parameter (e.g. tur_nechad2016_833 for S2B) or one of the aliases can be used for the red or NIR bands (tur_nechad2016_red, tur_nechad2016_nir). Sensors: L8/OLI, S2A/MSI, S2B/MSI

– **tur_nechadave**: Turbidity (FNU) using the algorithm of Nechad et al. (2009). By default the red band will be used, but a wavelength can be specified in the parameter (e.g. tur_nechadave_833 for S2B) or one of the aliases can be used for the red or NIR bands (tur_nechadave_red, tur_nechadave_nir). The hyperspectral dataset will be resampled to the band relative spectral response. Sensors: all sensors **Warning: not recommended for bands stretching outside the algorithm's optimal spectral range.**

- **tur_dogliotti**: Turbidity (FNU) using the algorithm of Dogliotti et al. (2015). The switching algorithm uses the red band for $\rho_{red} < 0.05$, and the NIR band for $\rho_{red} > 0.07$, with a linear weighing in between. The red and NIR based products can also be output separately (tur_dogliotti_red and tur_dogliotti_nir). Sensors: all sensors **Warning: Published calibration for MODIS is used for all sensors.**
- **chl_oc2, chl_oc3**: Chlorophyll a concentration ($\mu g/l$) using the blue/green ratio algorithm. The oc2 and oc3 use two and three bands respectively. Coefficients are provided in data/Shared/algorithms/chl_oc/defaults.txt, and users can add new coefficients in that file. Calibration values are taken from (Franz et al., 2015) or from the NASA/OBPG OceanColor website. Note that these products should be used with care in coastal and inland waters, especially in the presence of sediments and CDOM. Sensors: L8/OLI, S2A/MSI, S2B/MSI, S3A/OLCI, S3B/OLCI
- **chl_re_gons, chl_re_gons740**: Chlorophyll a concentration ($\mu g/l$) using the red edge algorithm by Gons et al. (2002) with published coefficients and a mass specific chlorophyll a absorption of 0.015. By default 780 nm (band 6) is used as a reference, but the chl_re_gons740 product uses 740 nm (band 5) on MSI. Output products are by default only produced for waters where $\rho_s 664 > 0.005$ and $\rho_s 704/\rho_s 664 > 0.63$ (thresholds defined by Héloïse Lavigne). Sensors: S2A/MSI, S2B/MSI, S3A/OLCI, S3B/OLCI
- **chl_re_moses3b, chl_re_moses3b740**: Chlorophyll a concentration ($\mu g/l$) using the three band red edge algorithm by Moses et al. (2012). By default 780 nm (band 6) is used as a reference, but the chl_re_moses3b740 product uses 740 nm (band 5) on MSI. Sensors: S2A/MSI, S2B/MSI, S3A/OLCI, S3B/OLCI
- **chl_re_mishra**: Chlorophyll a concentration ($\mu g/l$) using the Normalised Difference Chlorophyll Index algorithm by Mishra and Mishra (2012). Sensors: S2A/MSI, S2B/MSI, S3A/OLCI, S3B/OLCI
- **ndci**: The Normalised Difference Chlorophyll Index algorithm by Mishra and Mishra (2012). Sensors: S2A/MSI, S2B/MSI, S3A/OLCI, S3B/OLCI
- **chl_re_bramich**: Chlorophyll a concentration ($\mu g/l$) using an updated version of the Gons algorithm (Gons et al., 2002) for Sentinel-2 by Bramich et al. (2021). Sensors: S2A/MSI, S2B/MSI, S3A/OLCI, S3B/OLCI
- **slh**: The Scattering Line Height algorithm by Kudela et al. (2015). Sensors: S2A/MSI, S2B/MSI, S3A/OLCI, S3B/OLCI
- **fai, fai_rhot**: The Floating Algal Index by Hu (2009). By default surface reflectance (ρ_s) is used. **fai_rhot** uses the top-of-atmosphere reflectance (ρ_t). Sensors: L5/TM, L7/ETM, L8/OLI, S2A/MSI, S2B/MSI
- **fait**: The Floating Algal Index adapted to Turbid waters by Dogliotti et al. (2018). Surface reflectance (ρ_s) is used instead of Rayleigh corrected reflectance. Sensors: L8/OLI, S2A/MSI, S2B/MSI
- **ndvi, ndvi_rhot**: Normalised Difference Vegetation Index. By default surface reflectance (ρ_s) is used. **ndvi_rhot** uses the top-of-atmosphere reflectance (ρ_t). Sensors: all sensors
- **qaa, qaa5, qaa6**: Outputs from the Quasi-Analytical Algorithm (QAA) of Lee et al. (2002), from both versions (qaa) or specifically from version 5 (qaa5) or 6 (qaa6). Parameters can be requested separately, see further. Sensors: L8/OLI, S2A/MSI, S2B/MSI
- **qaa_rrs_443, qaa_rrs_490, qaa_rrs_560, qaa_rrs_665**: Subsurface Remote sensing reflectance from the Quasi-Analytical Algorithm (QAA) of Lee et al. (2002). Sensors: L8/OLI, S2A/MSI, S2B/MSI
- **u_rrs_443, u_rrs_490, u_rrs_560, u_rrs_665**: u parameter from the Quasi-Analytical Algorithm (QAA) of Lee et al. (2002). Sensors: L8/OLI, S2A/MSI, S2B/MSI
- **qaa_v5_a_443, qaa_v5_a_490, qaa_v5_a_560, qaa_v5_a_665, qaa_v6_a_443, qaa_v6_a_490, qaa_v6_a_560, qaa_v6_a_665**: absorption outputs from the Quasi-Analytical Algorithm (QAA) of Lee et al. (2002). The v5 or v6 specification denote version 5 or 6 outputs. Sensors: L8/OLI, S2A/MSI, S2B/MSI

– **qaa_v5_bbp_443, qaa_v5_bbp_490, qaa_v5_bbp_560, qaa_v5_bbp_665, qaa_v6_bbp_443, qaa_v6_bbp_490, qaa_v6_bbp_560, qaa_v6_bbp_665**: particulate backscattering outputs from the Quasi-Analytical Algorithm (QAA) of Lee et al. (2002). The v5 or v6 specification denote version 5 or 6 outputs. Sensors: L8/OLI, S2A/MSI, S2B/MSI

– **qaa_v5_Kd_443, qaa_v5_Kd_490, qaa_v5_Kd_560, qaa_v5_Kd_665, qaa_v6_Kd_443, qaa_v6_Kd_490, qaa_v6_Kd_560, qaa_v6_Kd_665**: diffuse attenuation outputs from the Quasi-Analytical Algorithm (QAA) of Lee et al. (2002). The v5 or v6 specification denote version 5 or 6 outputs. Sensors: L8/OLI, S2A/MSI, S2B/MSI

– **qaa_v6_KPAR_Lee, qaa_v6_Zeu_Lee**: diffuse attenuation of photosynthetically available radiation and euphotic zone depth from the Quasi-Analytical Algorithm (QAA) of Lee et al. (2002, 2007). Sensors: L8/OLI, S2A/MSI, S2B/MSI

– **qaa_v5_KdPAR_Nechad, qaa_v6_KdPAR_Nechad**: Nechad (unpublished?) version 2 fit of Kd PAR to Kd 490 outputs of the Quasi-Analytical Algorithm (QAA) of Lee et al. (2002). The v5 or v6 specification denote version 5 or 6 outputs. Sensors: L8/OLI, S2A/MSI, S2B/MSI

– **p3qaa**: Output all the parameters from the QAA-RGB (Pitarch and Vanhellemont, submitted). Sensors: L5/TM, L7/ETM, L8/OLI, S2A/MSI, S2B/MSI, PlanetScope (0c, 0d, 0e, 0f, 2x), Pléiades-1A, Pléiades-1B, RapidEye, SPOT6, SPOT7, Venμs, WorldView2, WorldView3

– **p3qaa_a_***: Total absorption from the QAA-RGB (Pitarch and Vanhellemont, submitted). R/G/B wavelength can be specified. Sensors: L5/TM, L7/ETM, L8/OLI, S2A/MSI, S2B/MSI, PlanetScope (0c, 0d, 0e, 0f, 2x), Pléiades-1A, Pléiades-1B, RapidEye, SPOT6, SPOT7, Venμs, WorldView2, WorldView3

– **p3qaa_bb_***: Total backscattering from the QAA-RGB (Pitarch and Vanhellemont, submitted). R/G/B wavelength can be specified. Sensors: L5/TM, L7/ETM, L8/OLI, S2A/MSI, S2B/MSI, PlanetScope (0c, 0d, 0e, 0f, 2x), Pléiades-1A, Pléiades-1B, RapidEye, SPOT6, SPOT7, Venμs, WorldView2, WorldView3

– **p3qaa_Kd_***: Total Kd from the QAA-RGB (Pitarch and Vanhellemont, submitted). R/G/B wavelength can be specified. Sensors: L5/TM, L7/ETM, L8/OLI, S2A/MSI, S2B/MSI, PlanetScope (0c, 0d, 0e, 0f, 2x), Pléiades-1A, Pléiades-1B, RapidEye, SPOT6, SPOT7, Venμs, WorldView2, WorldView3

– **p3qaa_zSD**: Secchi depth with bias removed from the QAA-RGB (Pitarch and Vanhellemont, submitted). Sensors: L5/TM, L7/ETM, L8/OLI, S2A/MSI, S2B/MSI, PlanetScope (0c, 0d, 0e, 0f, 2x), Pléiades-1A, Pléiades-1B, RapidEye, SPOT6, SPOT7, Venμs, WorldView2, WorldView3

– **p3qaa_zSD_biased**: Biased Secchi depth from the QAA-RGB (Pitarch and Vanhellemont, submitted). Sensors: L5/TM, L7/ETM, L8/OLI, S2A/MSI, S2B/MSI, PlanetScope (0c, 0d, 0e, 0f, 2x), Pléiades-1A, Pléiades-1B, RapidEye, SPOT6, SPOT7, Venμs, WorldView2, WorldView3

– **p3qaa_eta**: eta parameter from the QAA-RGB (Pitarch and Vanhellemont, submitted). Sensors: L5/TM, L7/ETM, L8/OLI, S2A/MSI, S2B/MSI, PlanetScope (0c, 0d, 0e, 0f, 2x), Pléiades-1A, Pléiades-1B, RapidEye, SPOT6, SPOT7, Venμs, WorldView2, WorldView3

– **bt***: At-sensor brightness temperature from the Landsat sensors. Individual parameters listed below. Sensors: L5/TM, L7/ETM, L8/OLI

– **bt6**: At-sensor brightness temperature from the Thematic Mapper (TM) on Landsat 5. Sensors: L5/TM

– **bt6_vcid_1, bt6_vcid_2**: At-sensor brightness temperature from the Enhanced Thematic Mapper (ETM) on Landsat 7 for both low (1) and high (2) gain levels. The low gain data is less likely to saturate over hot targets. Sensors: L7/ETM

– **bt10, bt11**: At-sensor brightness temperature from the Thermal Infrared Sensor (TIRS) on Landsat 8. Sensors: L8/TIRS

- **hue_angle**: The Hue Angle (°) by van der Woerd and Wernand (2018). Sensors: L8/OLI, S2A/MSI, S2B/MSI
- **rhos_592, rhow_592**: Panchromatic band reflectance (rhos_592). These are now output by default, record kept here for reference. Sensors: L8/OLI
- **rhos_613, rhow_613**: Orange 613 nm *contra*-band reflectances by Castagna et al. (2018, 2020). These are now output by default, record kept here for reference. Sensors: L8/OLI
- **olh**: Orange *contra*-band line height by Castagna et al. (2018, 2020). Sensors: L8/OLI

6. Sensor wavelengths

ACOLITE computes the wavelengths of each sensor by averaging the RSR. Here a list of wavelengths is provided per sensor. Note that bands with low gas transmittance (<0.75) are skipped for ρ_s computation. Bands in italic here have lower gas transmittance for *italic* here for zenith sun, nadir viewing and default ozone and water vapour. CHRIS and PRISMA wavelengths are taken from the inputfile metadata.

- **L5/TM**: 486, 571, 660, 839, 1678, 2217
- **L7/ETM**: 479, 561, 661, 835, 1650, 2208, 720
- **L8/OLI**: 443, 483, 561, 655, 865, 1609, 2201, 592, *1373*
- **PHR1A**: 501, 561, 650, 835, 657
- **PHR1B**: 505, 558, 663, 844, 661
- **PlanetScope/0c**: 492, 542, 622, 813
- **PlanetScope/0d05**: 493, 542, 621, 812
- **PlanetScope/0d06**: 493, 542, 621, 812
- **PlanetScope/0e**: 518, 552, 633, 812
- **PlanetScope/0f**: 505, 546, 625, 809
- **PlanetScope/22**: 492, 566, 666, 866
- **RapidEye**: 477, 556, 658, 709, 804
- **S2A/MSI**: 443, 492, 560, 665, 704, 740, 783, 833, 865, *945*, *1373*, 1614, 2202
- **S2B/MSI**: 442, 492, 559, 665, 704, 739, 780, 833, 864, *943*, *1377*, 1610, 2186
- **S3A/OLCI**: 400, 412, 443, 490, 510, 560, 620, 665, 674, 682, 709, 754, *762*, *765*, 768, 779, 865, 884, *899*, *939*, 1016
- **S3B/OLCI**: 401, 412, 443, 490, 510, 560, 620, 665, 674, 681, 709, 754, *762*, *765*, 768, 779, 865, 884, 899, *939*, 1016
- **SPOT6**: 487, 557, 659, 813, 596
- **SPOT7**: 491, 566, 666, 816, 597
- **VENμS/VSSC**: 424, 447, 492, 555, 620, 620, 666, 702, 741, 782, 861, *909*
- **WorldView2**: 428, 479, 548, 608, 659, 724, 828, 923, 645
- **WorldView3**: 427, 482, 547, 605, 660, 723, 827, 922, 1210, 1572, 1661, 1730, 2164, 2203, 2260, 2330, 643

7. Version history

20210802

- New generic code base that includes processing of metre-scale sensors (among others Pléiades and PlanetScope) and S3A/B OLCI
- Processing of CHRIS and PRISMA hyperspectral data, Venμs data
- Option to use resolved pressure derived from SRTM DEM
- New per-pixel geometry for Sentinel-2 data, support for per-band per-pixel geometry for Sentinel-2
- Separate repository for LUT storage (acolite/acolite_luts)
- Use of "reverse" LUTs to go from rhot -> aot when estimating aot from resolved geometry
- Panchromatic and orange band (Castagna et al., 2018, 2020) reflectance are output by default for L8/OLI
- Scene merging from different UTM zones
- Polygon (e.g. geojson) ROI definition for projected imagery (e.g. Sentinel-2 and Landsat data)
- Sky and sun glint reflectance can be estimated from (ancillary) wind speed
- Added three band RGB QAA algorithm (Pitarch and Vanhellemont, submitted)
- The following setting names have been changed: dsf_path_reflectance -> dsf_aot_estimate, sky_correction -> dsf_interface_reflectance, sky_correction_option -> dsf_interface_option, sky_correction_lut -> dsf_interface_lut, glint_correction -> dsf_residual_glint_correction

20210114

- Fixed gdal issue for GeoTIFF outputs

20210106

- Update to latest GitHub codebase
- Added options presented in (Vanhellemont, 2020c): removing SWIR bands from aerosol fitting and OSOAA based sky reflectance correction
- Fixed GUI ancillary download issue
- Processing from the Windows GUI can now also be interrupted
- Added options to delete NetCDF files that are produced
- Added support for Collection 2 Landsat 8 data

20190326

- Added correct xrange and yranges to NetCDF global attributes
- Changed Landsat Y subsetting
- Improved Pan band support for merged tiles and orange band computation
- Added support for LO8 files (without TIRS)
- Added strip function to l2w_parameter names in mapping script

- Added some CF convention parameter attributes
- Fixed S2 band naming error during tile merging
- Fixed l2_flags error when masking is disabled
- Fixed plotting of L8 dark spectrum
- Fixed time parsing issue for Landsat files
- Disabled printing of numpy Runtime Warnings

20181210

- Added orange band computation for Landsat 8/OLI, including panchromatic bands output at 30 m
- Added oxygen transmittance
- Added elevation input option
- Added/Fixed SRTM DEM option for elevation computation, missing DEM tiles are printed out
- Added the option to use the resolved S2 view/illumination grids
- Changed import of some Matplotlib features, hopefully allowing map output for EXP processed scenes
- Changed but did not fix download of ancillary data when using the GUI
- Fixed FAIT output on Mac (binary distribution was missing the skimage dependency)
- Fixed output of cirrus band (at TOA)
- Fixed MSI grid naming issue when merging tiles
- Fixed transmittance ratio in glint correction
- Fixed the EXP aerosol correction when L1R NetCDF files are present

20180925

- Explicitly added file encoding to avoid 'file not recognised' errors

20180917

- Added FAIT parameter
- Added glint correction for the DSF
- Added –nogfx option for –cli runs without graphical environments
- Changed relative path identification so the acolite binary can be launched from anywhere
- Fixed blue band bug after chl_oc3 computation
- Fixed ancillary data download in Windows binary distribution
- Fixed some settings file parsing errors
- Fixed default scalebar position for projected maps

20180611

- Added simple Rayleigh corrected reflectance product (with no gas and sky correction): rhorc = (rhot - rhor)/ttr

– Added support for GeoTIFF outputs of L2R/L2W datasets. NetCDF files are still generated, datasets will be also exported as GeoTIFF (l2r_export_geotiff=True, l2w_export_geotiff=True)

– Added optional output of UTM x/y easting and northing coordinates (xy_output=True)

– Added colour table options (Calgae255 and Planck_Parchment_RGB) and possibility for user defined colour tables

– Added output of l2_flags to the L2W file, flags are currently based on SWIR threshold mask (bit 1) and negative rhos (bit 2)

– Added option to disable masking of non-water/bad pixels (l2w_mask_water_parameters=False)

– Added option to output 1:1 pixel PNG files; map_raster=True (without title, geolocation or colourbar, currently not on Windows)

– Added check for rhot < rhor in DSF processing (especially for L5)

– Added pansharpened RGB outputs for L7 and L8

– Added aliases for t_nechad_red and t_nechad_nir products

– Changed naming of RGB output products to rgb_rhot and rgb_rhos

– Sentinel-2 wavelengths are now determined from the RSR instead of the provided metadata. This gives consistent naming before and after the Dec 2017 / Jan 2018 RSR update.

– Fixed the crash when no ancillary data is found or when the oceandata server is unreachable. Fallback values will be used if ancillary data access fails.

– Fixed sub pixel geolocation error in the computation of latitude and longitude: Sentinel-2 scene extents are considered to be at pixel edges, Landsat extents at pixel centres, all ACOLITE output coordinates are reported at pixel centre

– Fixed bug when multiple files are present in the GRANULE/IMG_DATA with the same band name (check if files end in jp2)

– Fixed the plotting of Sentinel-2 dark spectrum fitting results

20180419

– Fixed full tile Sentinel-2 geolocation

– Fixed Sentinel-2 relative azimuth within 0-180°

– Datasets are now processed and stored as 32 bit floats, using less RAM and disk space

– Added rhow (rhos with non-water masking) and Rrs (ρ_w / π) datasets

– Wavelengths are now tracked as attributes in the L2W products (rhot, rhos, rhow, Rrs), allowing for the use of the Spectrum Viewer in SNAP

– Skipping unrecognised and dot files in the .SAFE "GRANULE" and "IMG_DATA" directories

– Added support for autoscaling the L2W output maps: add *,* for the range in config/acolite_l2w_labels.txt

– Added new 'map_projected' option to annotate scalebar and add lat/lon to maps, switched to 'Agg' backend in matplotlib

– Relaxed constrains on the band selection for the chl_oc algorithms (for chl_oc3 outputs of S2A/B)

- Added Hue Angle algorithm

20180327 First public beta version of ACOLITE Python.

- Fixed PNG outputs from GUI
- Fixed Landsat 8 bt10 and bt11 outputs from merged tiles
- Added chl_re_gons configuration file and masking

20180312 First internal beta version of ACOLITE Python.

- Implementation is now in Python (3.6)
- New aerosol correction (DSF) with new lookup tables for path reflectance resampled to each of the supported sensors (L5/TM, L7/ETM, L8/OLI, S2A/MSI, S2B/MSI)
- The new spectral response function of Sentinel-2A is now used
- A more robust tile merging option (including Landsat tiles) has been added

20170718 Last release of ACOLITE IDL.

8. Known issues

- **High RAM use when processing full scenes** Processing full scenes is currently very RAM intensive, especially for operations using several bands, such as computing QAA parameters or generating RGB composites. The system may become unresponsive when low on RAM. ACOLITE may segfault if it runs out of RAM when generating the very large RGB composite.
- **Crash when generating full tile RGB composites** When generating full tile RGB composites (especially for Sentinel-2) the program may segfault, due to running out of RAM, or overriding PNG size limits.
- **Large output files** When processing full scenes, very large output files will be generated. For a full tile of Sentinel-2 data at 10 metres file sizes are about 10 GB for the L1R, and about 12 GB for the L2R files. L2W file size depends on the number of requested parameters (about 0.5 GB/parameter + latitude and longitude).
- **High RAM use when processing hyperspectral data** Processing of hyperspectral data requires the loading of the generic LUTs, this uses plenty of RAM, and processing may fail after the "Loading LUTs" step if not enough RAM is available. At least 16GB of RAM is recommended for processing of hyperspectral data.
- **PRISMA processing requires both L1 and L2C files** Processing of PRISMA data requires both official L1 (PRS_L1_STD_OFFL) and L2C (PRS_L2C_STD) files since the view geometry is only provided in the L2C. The L1 needs to be provided as inputfile, and the L2C needs to be present in the same directory as the L1.

References

Bourg, L., D'Alba, L., Colagrande, P., 2008. MERIS SMILE effect characterisation and correction. ESA technical note 4921, 25.

Bramich, J., Bolch, C. J., Fischer, A., 2021. Improved red-edge chlorophyll-a detection for Sentinel 2. Ecological Indicators 120, 106876.

Castagna, A., Simis, S., Dierssen, H., Vanhellemont, Q., Sabbe, K., Vyverman, W., 2018. Extending the Operational Land Imager/Landsat 8 for freshwater research: retrieval of an orange band from PAN and MS bands. Ocean Optics XXIV Conference, 7-12 October 2018, Dubrovnik, Coratia.

Castagna, A., Simis, S., Dierssen, H., Vanhellemont, Q., Sabbe, K., Vyverman, W., 2020. Extending Landsat 8: Retrieval of an orange contra-band for inland water quality applications. Remote Sensing 12 (4), 637.

Castagna, A., Simis, S., Dierssen, H., Vanhellemont, Q., Sabbe, K., Vyverman, W., submitted. Extending Landsat 8: Retrieval of an orange *contra*-band for inland water quality applications, manuscript submitted for publication.

Chami, M., Lafrance, B., Fougnie, B., Chowdhary, J., Harmel, T., Waquet, F., 2015. OSOAA: a vector radiative transfer model of coupled atmosphere-ocean system for a rough sea surface application to the estimates of the directional variations of the water leaving reflectance to better process multi-angular satellite sensors data over the ocean. Optics Express 23 (21), 27829–27852.

Dogliotti, A., Gossn, J., Vanhellemont, Q., Ruddick, K., 2018. Detecting and quantifying a massive invasion of floating aquatic plants in the río de la plata turbid waters using high spatial resolution ocean color imagery. Remote Sensing 10 (7), 1140.

Dogliotti, A. I., Ruddick, K., Nechad, B., Doxaran, D., Knaeps, E., 2015. A single algorithm to retrieve turbidity from remotely-sensed data in all coastal and estuarine waters. Remote Sensing of Environment 156, 157–168.

Franz, B. A., Bailey, S. W., Kuring, N., Werdell, P. J., 2015. Ocean color measurements with the Operational Land Imager on Landsat-8: implementation and evaluation in SeaDAS. Journal of Applied Remote Sensing 9 (1), 096070.

Gómez-Chova, L., Alonso, L., Guanter, L., Camps-Valls, G., Calpe, J., Moreno, J., 2008. Correction of systematic spatial noise in push-broom hyperspectral sensors: application to chris/proba images. Applied Optics 47 (28), F46–F60.

Gons, H. J., Rijkeboer, M., Ruddick, K. G., 2002. A chlorophyll-retrieval algorithm for satellite imagery (medium resolution imaging spectrometer) of inland and coastal waters. Journal of Plankton Research 24 (9), 947–951.

Harmel, T., Chami, M., Tormos, T., Reynaud, N., Danis, P.-A., 2018. Sunglint correction of the Multi-Spectral Instrument (MSI)-SENTINEL-2 imagery over inland and sea waters from SWIR bands. Remote Sensing of Environment 204, 308–321.

Hu, C., 2009. A novel ocean color index to detect floating algae in the global oceans. Remote Sensing of Environment 113 (10), 2118–2129.

Kudela, R. M., Palacios, S. L., Austerberry, D. C., Accorsi, E. K., Guild, L. S., Torres-Perez, J., 2015. Application of hyperspectral remote sensing to cyanobacterial blooms in inland waters. Remote Sensing of Environment 167, 196–205.

Lavigne, H., Vanhellemont, Q., Ruddick, K., Dogliotti, A.-I., 2021. New processor and reference dataset for hyperspectral CHRIS-PROBA images over coastal and inland waters. IGARSS conference 2021, Brussels, Belgium TH4.O-15.

Lee, Z., Carder, K. L., Arnone, R. A., 2002. Deriving inherent optical properties from water color: a multiband quasi-analytical algorithm for optically deep waters. Applied Optics 41 (27), 5755–5772.

Lee, Z., Weidemann, A., Kindle, J., Arnone, R., Carder, K. L., Davis, C., 2007. Euphotic zone depth: Its derivation and implication to ocean-color remote sensing. Journal of Geophysical Research: Oceans 112 (C3).

Mishra, S., Mishra, D. R., 2012. Normalized difference chlorophyll index: A novel model for remote estimation of chlorophyll-a concentration in turbid productive waters. Remote Sensing of Environment 117, 394–406.

Moses, W. J., Gitelson, A. A., Berdnikov, S., Saprygin, V., Povazhnyi, V., 2012. Operational MERIS-based NIR-red algorithms for estimating chlorophyll-a concentrations in coastal waters—The Azov Sea case study. Remote Sensing of Environment 121, 118–124.

Nechad, B., Ruddick, K., Neukermans, G., 2009. Calibration and validation of a generic multisensor algorithm for mapping of turbidity in coastal waters. In: SPIE Europe Remote Sensing. International Society for Optics and Photonics, pp. 74730H–74730H.

Nechad, B., Ruddick, K., Park, Y., 2010. Calibration and validation of a generic multisensor algorithm for mapping of total suspended matter in turbid waters. Remote Sensing of Environment 114 (4), 854–866.

Pitarch, J., Vanhellemont, Q., submitted. The QAA-RGB: a universal three-band absorption and backscattering retrieval algorithm for high resolution satellite sensors. Development and implementation in ACOLITE. Remote Sensing of Environment.

Ruddick, K. G., De Cauwer, V., Park, Y.-J., Moore, G., 2006. Seaborne measurements of near infrared water-leaving reflectance: The similarity spectrum for turbid waters. Limnol. Oceanogr 51 (2), 1167–1179.

van der Woerd, H. J., Wernand, M. R., 2018. Hue-angle product for low to medium spatial resolution optical satellite sensors. Remote Sensing 10 (2), 180.

Vanhellemont, Q., 2019a. Adaptation of the dark spectrum fitting atmospheric correction for aquatic applications of the Landsat and Sentinel-2 archives. Remote Sensing of Environment 225, 175–192.

Vanhellemont, Q., 2019b. Daily metre-scale mapping of water turbidity using CubeSat imagery. Optics Express 27 (20), A1372–A1399.

Vanhellemont, Q., 2020a. Automated Water Surface Temperature retrieval from Landsat 8/TIRS. Remote Sensing of Environment 237, 111518.

Vanhellemont, Q., 2020b. Combined land surface emissivity and temperature estimation from landsat 8 oli and tirs. ISPRS Journal of Photogrammetry and Remote Sensing 166, 390–402.

Vanhellemont, Q., 2020c. Sensitivity analysis of the Dark Spectrum Fitting atmospheric correction for metre- and decametre-scale satellite imagery using autonomous hyperspectral radiometry. Optics Express 27 (20), A1372–A1399.

Vanhellemont, Q., Ruddick, K., 2014. Turbid wakes associated with offshore wind turbines observed with Landsat 8. Remote Sensing of Environment 145, 105–115.

Vanhellemont, Q., Ruddick, K., 2015. Advantages of high quality SWIR bands for ocean colour processing: Examples from Landsat-8. Remote Sensing of Environment 161, 89–106.

Vanhellemont, Q., Ruddick, K., 2016. ACOLITE For Sentinel-2: Aquatic Applications of MSI imagery. In: ESA Special Publication SP-740. Presented at the ESA Living Planet Symposium held in Prague, Czech Republic.

Vanhellemont, Q., Ruddick, K., 2018. Atmospheric correction of metre-scale optical satellite data for inland and coastal water applications. Remote Sensing of Environment 216, 586–597.

Vanhellemont, Q., Ruddick, K., 2021. Atmospheric correction of Sentinel-3/OLCI data for mapping of suspended particulate matter and chlorophyll-a concentration in Belgian turbid coastal waters. Remote Sensing of Environment 256, 112284.