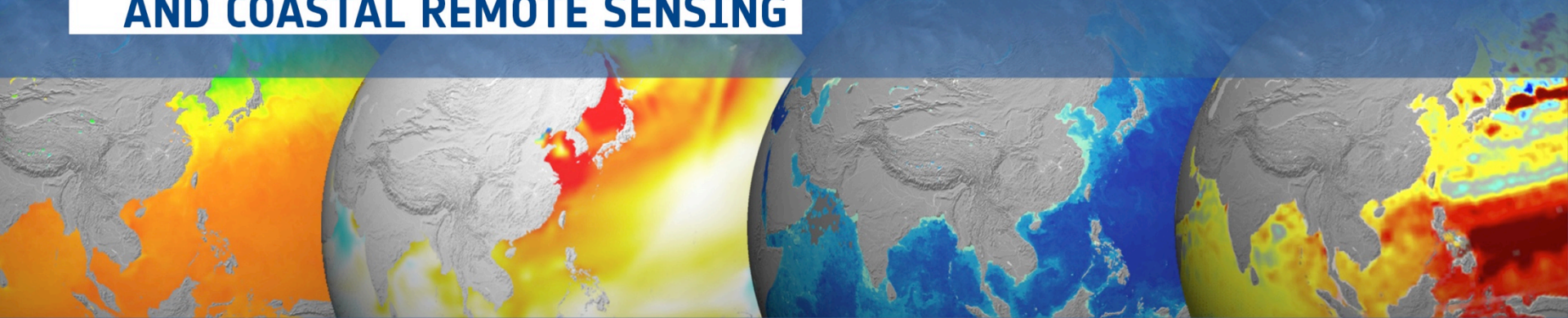




ESA–MOST China Dragon 4 Cooperation

→ **ADVANCED TRAINING COURSE IN OCEAN  
AND COASTAL REMOTE SENSING**



12 to 17 November 2018 | Shenzhen University | P.R. China

**Ocean Colour Retrievals SNAP & S3 OLCI  
Data**

T.Jackson (Plymouth Marine Laboratory)



In this practical session you will use the SNAP GUI to:

- 1) Visualise data and masks.
- 2) Filter a product based on user defined criteria.
- 3) Create a 'true colour' image from level 2 and level 1 files.
- 5) Create a chlorophyll-a estimate from reflectance bands.
- 6) Extract a data from a transect.



## IMPORTANT NOTE:

This lesson requires **Sentinels Application Platform (SNAP) software** and **Sentinel Toolboxes** which can be downloaded at :

<http://step.esa.int/main/download/>

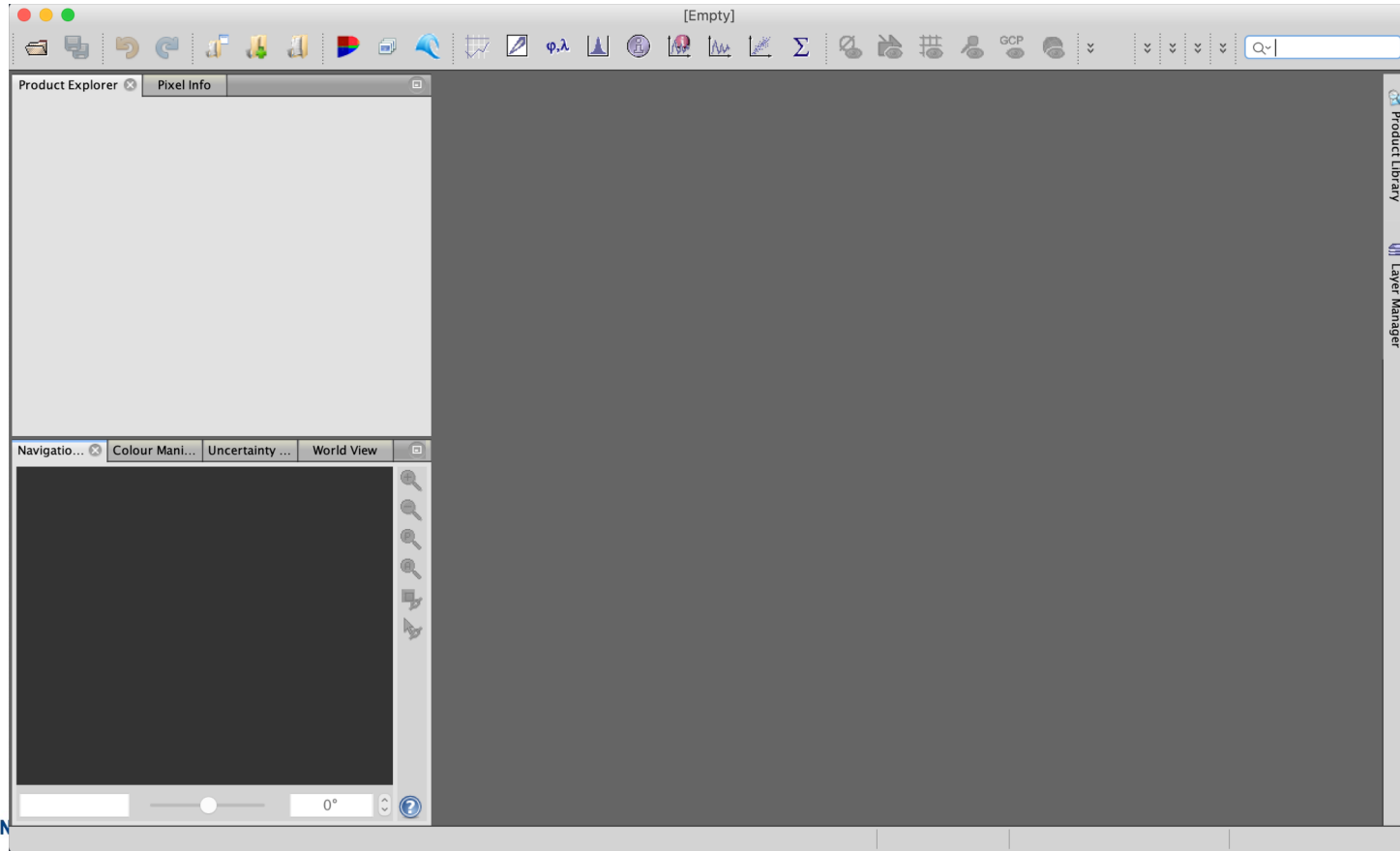
This should already be available on your machines.

# Exercise 1: Exploring and Visualising data



Begin by starting the SNAP program (you can simply type snap and hit enter on the command line).

This should provide an interface that looks something like this →



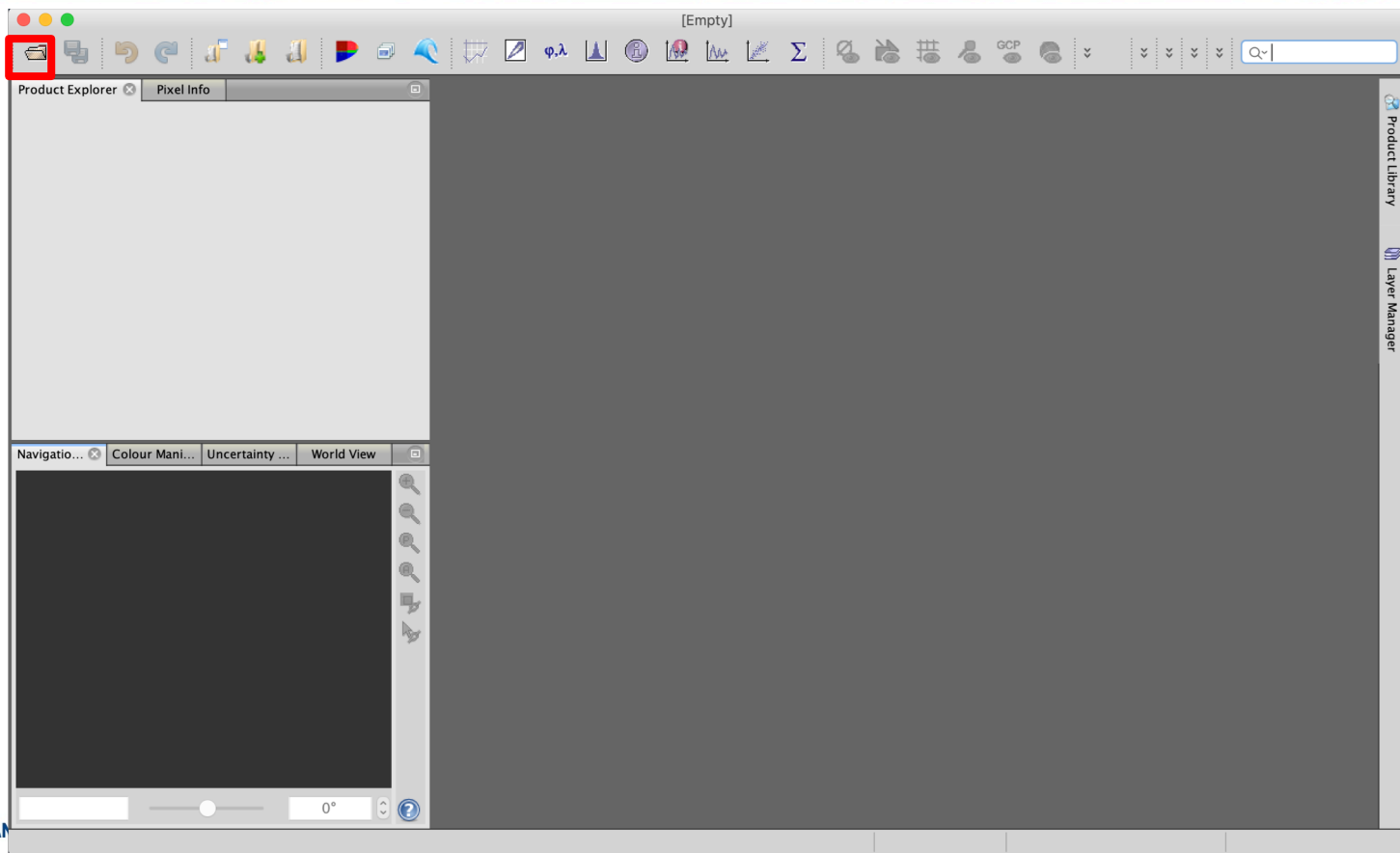
# Exercise 1: Exploring and Visualising data



Click the open file icon.

Navigate to the OLCI level 2 folder

S3A\_OL\_2\_WFR  
\_\_\_\_20181002T  
020947\_201810  
02T021247\_201  
81003T112306\_  
0179\_036\_217\_  
2340\_MAR\_O\_N  
T\_002.SEN3

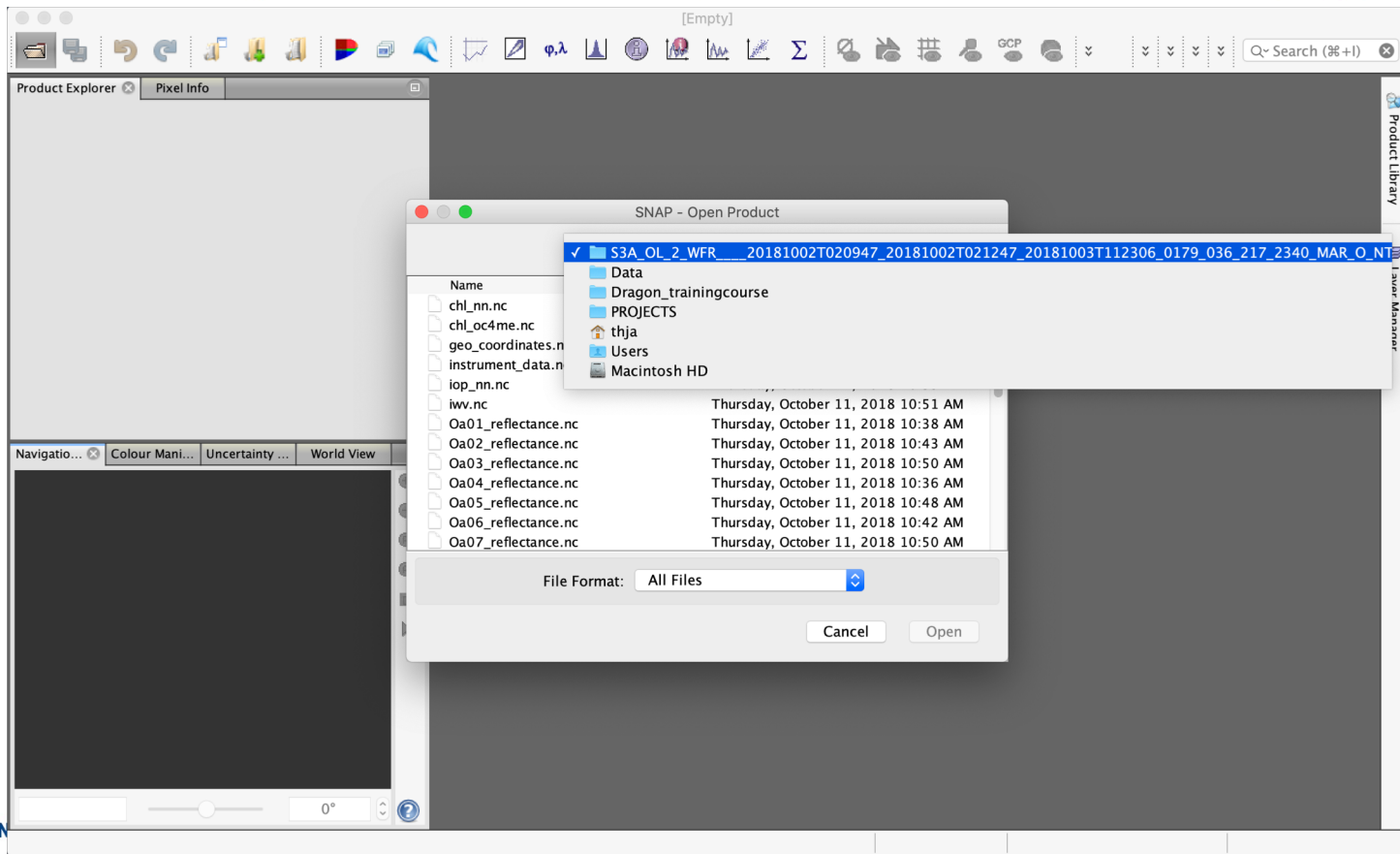


# Exercise 1: Exploring and Visualising data



You will find a number of netcdf files and an xml file.

As netcdfs are self describing you load single files/variables but here we will select the xml file and load all the variables at once.

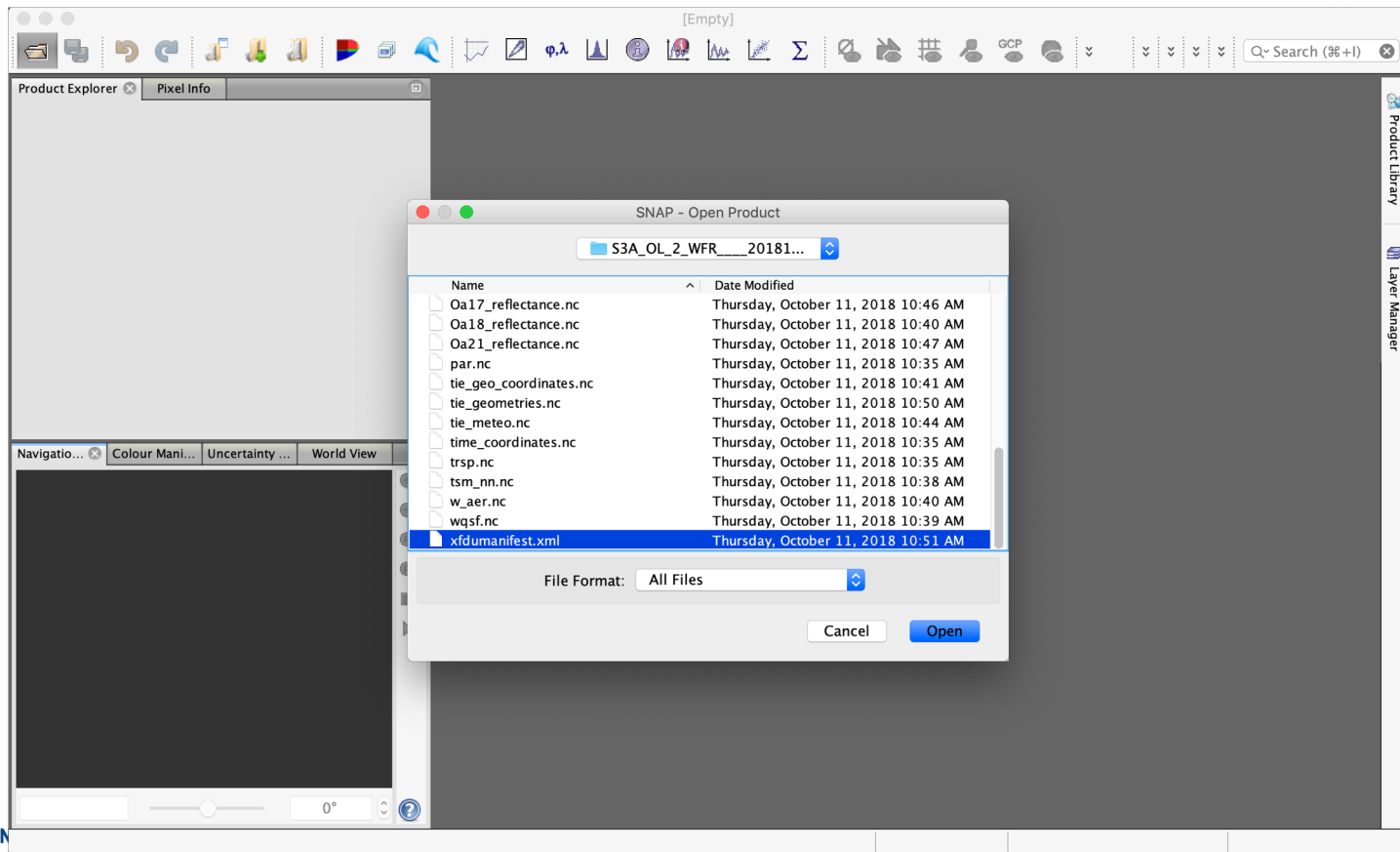


# Exercise 1: Exploring and Visualising data



You will find a number of netcdf files and an xml file.

As netcdfs are self describing you load single files/variables but here we will select the xml file and load all the variables at once.

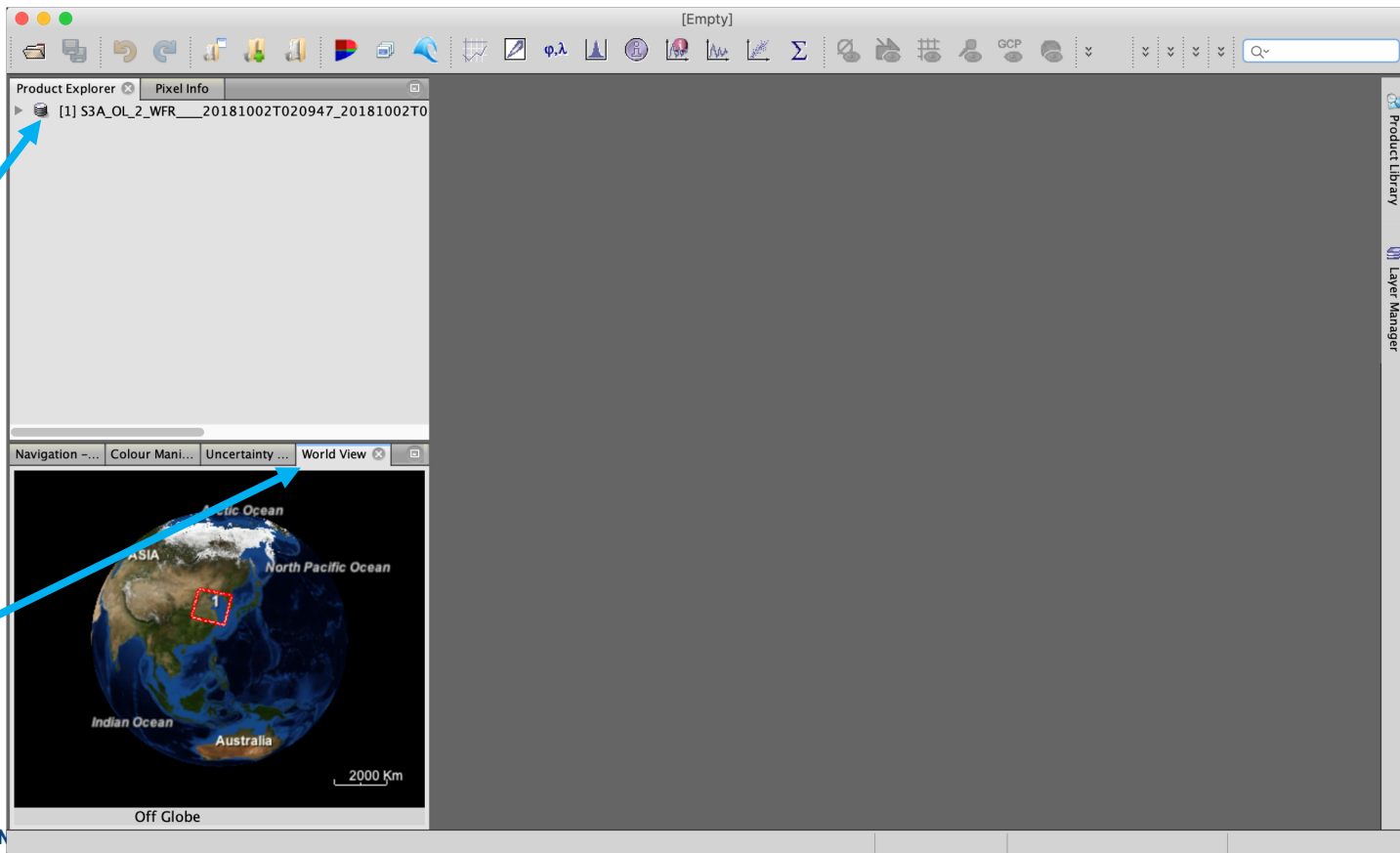


# Exercise 1: Exploring and Visualising data



The data are now available for browsing in the product explorer window.

Also, if you switch the lower panel to 'world view' you should see where the granule is located.



# Exercise 1: Exploring and Visualising data



Take a few minutes to investigate the structure of the data.

The reflectance bands all have wavelengths assigned.

The screenshot shows a software interface with a 'Product Explorer' panel on the left. The panel is expanded to show a list of reflectance bands under the 'Oa\*\_reflectance' folder. The bands listed are:

- Oa01\_reflectance (400 nm)
- Oa02\_reflectance (412.5 nm)
- Oa03\_reflectance (442.5 nm) - highlighted in blue
- Oa04\_reflectance (490 nm)
- Oa05\_reflectance (510 nm)
- Oa06\_reflectance (560 nm)
- Oa07\_reflectance (620 nm)
- Oa08\_reflectance (665 nm)
- Oa09\_reflectance (673.75 nm)

Below the product explorer is a 'World View' window showing a globe. A red box on the globe indicates the location of the data, which is in the North Pacific Ocean. Labels on the globe include 'Arctic Ocean', 'ASIA', 'North Pacific Ocean', 'Indian Ocean', and 'Australia'. A scale bar at the bottom right of the globe indicates 2000 Km. The text 'Off Globe' is visible at the bottom of the globe view.



# Exercise 1: Exploring and Visualising data



Check the metadata for a reflectance band and you will find:

- orbit numbers
- data creation time
- contact details
- Much more

Using metadata, confirm you have the same file as shown here.

The screenshot shows a GIS application window with the following components:

- Product Explorer:** A tree view showing a folder structure for 'S3A\_OL\_2\_WFR\_20181002T020947\_20181002T021247\_20181003T112306\_0179\_036\_217\_2340\_MAR\_OL\_NT\_002.SEN3'. The 'Metadata' folder is expanded, showing a list of reflectance bands from Oa01 to Oa16.
- Metadata Table:** A table titled '[1] Oa03\_reflectance' with columns for Name, Value, Type, Unit, and Description. Two blue arrows point from the text 'orbit numbers' and 'data creation time' in the list to the 'absolute\_orbit\_number' and 'creation\_time' rows in the table, respectively.
- Globe View:** A satellite image of Earth showing Asia, Australia, and the Indian Ocean. A red square highlights a specific geographic area in the Indian Ocean. A scale bar indicates 2000 Km.

Name	Value	Type	Unit	Description
absolute_orbit_number	13665	int32		
subsampling_factor	64	int16		
al_subsampling_factor	1	int16		
comment		ascii		
contact	ops@eumetsat.int	ascii		
creation_time	2018-10-03T11:23:06Z	ascii		
history	2018-10-03T11:23:06Z: PUGCore	ascii		
institution	MAR	ascii		
netCDF_version	4.2 of Jul 5 2012 17:07:43 \$	ascii		
product_name	S3A_OL_2_WFR_20181002T0209	ascii		
references	S3IPF PDS 004.3 - i2r2 - Product Da	ascii		
resolution	[ 270 294 ]	ascii		
source	IPF-OL-2 06.12	ascii		
start_time	2018-10-02T02:09:46.756184Z	ascii		
stop_time	2018-10-02T02:12:46.721179Z	ascii		
title	OLCI Level 2 WATER Product, Reflecta	ascii		

# Exercise 1: Exploring and Visualising data



Now within the Chl product set you will find CHL\_OC4ME.

You can also switch the lower left panel to colour options.

Your window should then look like this.

The screenshot shows a GIS application window with a dark background. The main view displays a satellite image of the North Atlantic region, showing cloud cover and landmasses. On the left, there is a 'Product Explorer' panel with a tree view showing the following structure:

- [1] S3A\_OL\_2\_WFR\_\_20181002T020947\_20181002
  - Metadata
  - Flag Codings
  - Vector Data
  - Tie-Point Grids
  - Bands
    - Oa\*\_reflectance
    - Oa\*\_reflectance\_err
    - A865
    - ADG
    - CHL
      - CHL\_NN
      - CHL\_NN\_err
      - CHL\_OC4ME
      - CHL\_OC4ME\_err
    - IWV
    - ...

Below the Product Explorer is a 'Navigation' panel with tabs for 'Colour M...', 'Uncertainty ...', and 'World View'. The 'Colour M...' tab is active, showing a histogram for the 'CHL\_OC4ME' product. The histogram has a white line representing a linear scale. The x-axis has markers at 0.31, 2.12, and 14.64. The y-axis has markers at 95% and 100%. The histogram shows a distribution of values with a peak around 2.12. The text 'Rough statistics!' is displayed in red. The histogram also shows a 'Log10' scale option.

At the bottom of the window, there is a status bar with the following information: X 1526 Y 3002 Lat 33°40'52" N Lon 116°38'... Zoom 1:6.1 Level 2

# Exercise 1: Exploring and Visualising data



In the 'colour manipulation' window you can modify the colour palette used for the image.

Use CHL\_SeaWiFS.cpd and click 'no' on auto scaling.

Once the image has changed click the 'log10' icon as chl-a is log-normally distributed.

The screenshot shows a GIS application window with a satellite image of the ocean. The 'Product Explorer' on the left shows a tree view of data layers, including 'CHL\_OC4ME'. The 'Colour M...' window is open, showing a histogram of the data. The histogram has a 'Log10' icon selected. The 'Import Colour Palette' dialog box is open, showing a list of color palette files. The file 'CHL\_SeaWiFS.cpd' is selected. The 'File Format' is set to 'Colour palette files (\*.cpd)'. The 'Open' button is highlighted.

Name	Date Modified
5_colors.cpd	Monday, December 5, 2016 1:22 PM
7_colors.cpd	Monday, December 5, 2016 1:22 PM
cc_chl.cpd	Monday, December 5, 2016 1:22 PM
cc_general.cpd	Monday, December 5, 2016 1:22 PM
cc_iop_quality.cpd	Monday, December 5, 2016 1:22 PM
cc_tsm.cpd	Monday, December 5, 2016 1:22 PM
cc_yellowsubstance.cpd	Monday, December 5, 2016 1:22 PM
cc_z90.cpd	Monday, December 5, 2016 1:22 PM
CHL_SeaWiFS.cpd	Monday, December 5, 2016 1:22 PM
classes_5_colors.cpd	Monday, December 5, 2016 1:24 PM
classes_7_colors.cpd	Monday, December 5, 2016 1:24 PM
cubehelix_cycle.cpd	Monday, December 5, 2016 1:24 PM
Deimos_color_palette.cpd	Wednesday, October 3, 2018 2:02 PM

# Exercise 2: Masking/filtering data

Note the regions of masked pixels from land, cloud etc.

Select the 'Mask manager' tool to allow visualisation of product flags.

The screenshot shows a GIS application window with the following components:

- Product Explorer:** A tree view on the left showing metadata, flag codings, vector data, and bands. The 'Bands' folder is expanded, showing 'CHL\_OC4ME' and 'CHL\_OC4ME\_err'.
- Main Visualization:** A large map area displaying a satellite-derived chlorophyll-a concentration map. The map shows a coastal region with varying colors from green to red, indicating different concentrations. A red box highlights a specific tool in the right-hand toolbar.
- Toolbar:** A vertical toolbar on the right side of the map area. A red box highlights the 'Mask manager' tool, which is used for visualizing product flags.
- Histogram:** A histogram at the bottom left of the interface. It shows the distribution of values for the 'CHL\_OC4ME' band. The y-axis is labeled 'Name: CHL\_OC4ME' and 'Unit: mg.m-3'. The x-axis shows values from 0.0 to 30.0. The histogram shows a peak around 1.0 mg.m-3.

# Exercise 2: Masking/filtering data



Turn on/off some flags and find where they are set to 'true'. As a minimum, try:

LAND  
CLOUD  
HIGHGLINT

Then zoom in on the area shown here (the navigation sub-panel can help)

The screenshot shows a software interface with a main satellite image, a left-hand panel with metadata, and a right-hand 'Mask Manager' panel. The 'Mask Manager' panel contains a table of flags with checkboxes, names, types, colors, and descriptions. Three blue arrows originate from the text 'LAND', 'CLOUD', and 'HIGHGLINT' on the left and point to the corresponding entries in the 'Mask Manager' table.

Name	Type	Colour	Tran...	Description
<input type="checkbox"/> WQSF_Isb.INVALID	Maths	Blue	0.5	WQSF_Isb.INVALID
<input type="checkbox"/> WQSF_Isb.WATER	Maths	Blue	0.5	WQSF_Isb.WATER
<input type="checkbox"/> WQSF_Isb.LAND	Maths	Green	0.5	WQSF_Isb.LAND
<input type="checkbox"/> WQSF_Isb.CLOUD	Maths	Grey	0.5	WQSF_Isb.CLOUD
<input type="checkbox"/> WQSF_Isb.CLOUD_AMI	Maths	Grey	0.5	WQSF_Isb.CLOUD_AMI
<input type="checkbox"/> WQSF_Isb.SNOW_ICE	Maths	White	0.5	WQSF_Isb.SNOW_ICE
<input type="checkbox"/> WQSF_Isb.INLAND_WA	Maths	Green	0.5	WQSF_Isb.INLAND_WA
<input type="checkbox"/> WQSF_Isb.TIDAL	Maths	Light Green	0.5	WQSF_Isb.TIDAL
<input type="checkbox"/> WQSF_Isb.COSMETIC	Maths	Cyan	0.5	WQSF_Isb.COSMETIC
<input type="checkbox"/> WQSF_Isb.SUSPECT	Maths	Magenta	0.5	WQSF_Isb.SUSPECT
<input type="checkbox"/> WQSF_Isb.HISOLZEN	Maths	Yellow	0.5	WQSF_Isb.HISOLZEN
<input type="checkbox"/> WQSF_Isb.SATURATEE	Maths	Dark Blue	0.5	WQSF_Isb.SATURATEE
<input type="checkbox"/> WQSF_Isb.MEGLINT	Maths	Dark Green	0.5	WQSF_Isb.MEGLINT
<input type="checkbox"/> WQSF_Isb.HIGHGLINT	Maths	Teal	0	WQSF_Isb.HIGHGLINT
<input type="checkbox"/> WQSF_Isb.WHITECAPS	Maths	Light Green	0.5	WQSF_Isb.WHITECAPS
<input type="checkbox"/> WQSF_Isb.ADJAC	Maths	Dark Red	0.5	WQSF_Isb.ADJAC
<input type="checkbox"/> WQSF_Isb.WV_FAIL	Maths	Purple	0.5	WQSF_Isb.WV_FAIL
<input type="checkbox"/> WQSF_Isb.PAR_FAIL	Maths	Purple	0.5	WQSF_Isb.PAR_FAIL
<input type="checkbox"/> WQSF_Isb.AC_FAIL	Maths	Olive	0.5	WQSF_Isb.AC_FAIL
<input type="checkbox"/> WQSF_Isb.OC4ME_FAI	Maths	Grey	0.5	WQSF_Isb.OC4ME_FAI
<input type="checkbox"/> WQSF_Isb.OCNN_FAIL	Maths	Light Blue	0.5	WQSF_Isb.OCNN_FAIL
<input type="checkbox"/> WQSF_Isb.KDM_FAIL	Maths	Light Green	0.5	WQSF_Isb.KDM_FAIL
<input type="checkbox"/> WQSF_Isb.BPAC_ON	Maths	Light Green	0.5	WQSF_Isb.BPAC_ON
<input type="checkbox"/> WQSF_Isb.WHITE_SCA	Maths	Cyan	0.5	WQSF_Isb.WHITE_SCA
<input type="checkbox"/> WQSF_Isb.LOWER	Maths	Magenta	0.5	WQSF_Isb.LOWER
<input type="checkbox"/> WQSF_Isb.HIGHRW	Maths	Orange	0.5	WQSF_Isb.HIGHRW
<input type="checkbox"/> WQSF_msb.ANNOT_A	Maths	Red	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_msb.ANNOT_A	Maths	Magenta	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_msb.ANNOT_A	Maths	Yellow	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_msb.ANNOT_A	Maths	Dark Blue	0.5	WQSF_msb.ANNOT_A

# Exercise 2: Masking/filtering data



Also note that changing to pixel info and moving the mouse over the map will provide information (such as flags) on a per-pixel basis which updates actively.

The screenshot shows a software interface with a central satellite image. On the left, a 'Pixel Info' panel displays coordinates and flags. On the right, a 'Mask Manager' panel lists various mask types and their descriptions. The interface includes a toolbar at the top and a navigation panel on the right.

**Pixel Info Panel:**

- Position: Image-X: 4115 pixel, Image-Y: 2453 pixel, Longitude: 124°23'2... degree, Latitude: 33°45'02" N degree
- Time: [empty]
- Bands: CHL\_OC4ME: 0.83418 mg.m-3
- Tie-Point Grids: [empty]
- Flags: WQSF\_Isb.INVALID: false, WQSF\_Isb.WATER: true, WQSF\_Isb.LAND: false, WQSF\_Isb.CLOUD: false, WQSF\_Isb.CLOUD\_...: false, WQSF\_Isb.CLOUD\_...: false

**Mask Manager Panel:**

Name	Type	Colour	Tran...	Description
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.INVALID
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.WATER
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.LAND
<input type="checkbox"/> WQSF_I...	Maths	Grey	0.5	WQSF_Isb.CLOUD
<input type="checkbox"/> WQSF_I...	Maths	Grey	0.5	WQSF_Isb.CLOUD_AMI
<input type="checkbox"/> WQSF_I...	Maths	White	0.5	WQSF_Isb.SNOW_ICE
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.INLAND_WA
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.TIDAL
<input type="checkbox"/> WQSF_I...	Maths	Cyan	0.5	WQSF_Isb.COSMETIC
<input type="checkbox"/> WQSF_I...	Maths	Magenta	0.5	WQSF_Isb.SUSPECT
<input type="checkbox"/> WQSF_I...	Maths	Yellow	0.5	WQSF_Isb.HISOLZEN
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.SATURATEE
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.MEGLINT
<input type="checkbox"/> WQSF_I...	Maths	Teal	0	WQSF_Isb.HIGHGLINT
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.WHITECAPS
<input type="checkbox"/> WQSF_I...	Maths	Red	0.5	WQSF_Isb.ADJAC
<input type="checkbox"/> WQSF_I...	Maths	Purple	0.5	WQSF_Isb.WV_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Purple	0.5	WQSF_Isb.PAR_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Olive	0.5	WQSF_Isb.AC_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Grey	0.5	WQSF_Isb.OC4ME_FAI
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.OCNN_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.KDM_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Light Green	0.5	WQSF_Isb.BPAC_ON
<input type="checkbox"/> WQSF_I...	Maths	Cyan	0.5	WQSF_Isb.WHITE_SCA
<input type="checkbox"/> WQSF_I...	Maths	Pink	0.5	WQSF_Isb.LOWRW
<input type="checkbox"/> WQSF_I...	Maths	Orange	0.5	WQSF_Isb.HIGHRW
<input type="checkbox"/> WQSF_...	Maths	Red	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_...	Maths	Pink	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_...	Maths	Yellow	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_...	Maths	Dark Blue	0.5	WQSF_msb.ANNOT_A

# Exercise 2: Masking/filtering data



In the mid-right of the image you will find a region of missing Chl-data due to cloud. There is also some suspicious data near to the cloud edges and between blocks of cloud.

The screenshot shows a software interface with a central satellite image of a coastal region. The image is overlaid with a mask, where different colors represent different data quality or status. A 'Mask Manager' panel on the right lists various mask types and their descriptions. A 'Product Explorer' panel on the left shows the current product and its metadata.

**Product Explorer**

**Position**

- Image-X: 4290 pixel
- Image-Y: 2136 pixel
- Longitude: 125°08'3... degree
- Latitude: 34°27'31" N degree

**Time**

**Bands**

- CHL\_OC4ME: 1.28895 mg.m-3

**Flags**

Flag	Value
WQSF_Isb.INVALID	false
WQSF_Isb.WATER	true
WQSF_Isb.LAND	false
WQSF_Isb.CLOUD	false
WQSF_Isb.CLOUD_...	false
WQSF_Isb.CLOUD_...	false
WQSF_Isb.SNOW_ICE	false

**Mask Manager**

Name	Type	Colour	Tran...	Description
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.INVALID
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.WATER
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.LAND
<input checked="" type="checkbox"/> WQSF_I...	Maths	Grey	0.5	WQSF_Isb.CLOUD
<input type="checkbox"/> WQSF_I...	Maths	Grey	0.5	WQSF_Isb.CLOUD_MA
<input checked="" type="checkbox"/> WQSF_I...	Maths	Red	0.5	WQSF_Isb.CLOUD_MA
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.SNOW_ICE
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.INLAND_WF
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.TIDAL
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.COSMETIC
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.SUSPECT
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.HISOLZEN
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.SATURATEE
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.MEGLINT
<input type="checkbox"/> WQSF_I...	Maths	Green	0	WQSF_Isb.HIGHGLINT
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.WHITCAPS
<input type="checkbox"/> WQSF_I...	Maths	Red	0.5	WQSF_Isb.ADJAC
<input type="checkbox"/> WQSF_I...	Maths	Purple	0.5	WQSF_Isb.WV_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Purple	0.5	WQSF_Isb.PAR_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Olive	0.5	WQSF_Isb.AC_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Grey	0.5	WQSF_Isb.OC4ME_FAI
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.OCNN_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.KDM_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.BPAC_ON
<input type="checkbox"/> WQSF_I...	Maths	Cyan	0.5	WQSF_Isb.WHITE_SCA
<input type="checkbox"/> WQSF_I...	Maths	Pink	0.5	WQSF_Isb.LOWRW
<input type="checkbox"/> WQSF_I...	Maths	Orange	0.5	WQSF_Isb.HIGHRW
<input type="checkbox"/> WQSF_I...	Maths	Red	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_I...	Maths	Pink	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_I...	Maths	Yellow	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_I...	Maths	Dark Blue	0.5	WQSF_msb.ANNOT_A

# Exercise 2: Masking/filtering data



Change the **CLOUD\_MARGIN** flag to red and turn it on.

This shows these pixels are likely influenced by cloud edge effects.

So we shall create a chl-a product with extra masking applied.

Product Explorer

Pixel Info

[1] CHL\_OC4ME

Mask Manager

Name	Type	Colour	Tran...	Description
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.INVALID
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.WATER
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.LAND
<input checked="" type="checkbox"/> WQSF_I...	Maths	Red	0.5	WQSF_Isb.CLOUD
<input type="checkbox"/> WQSF_I...	Maths	Grey	0.5	WQSF_Isb.CLOUD_AM
<input checked="" type="checkbox"/> WQSF_I...	Maths	Red	0.5	WQSF_Isb.CLOUD_MA
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.SNOW_ICE
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.INLAND_WF
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.TIDAL
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.COSMETIC
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.SUSPECT
<input type="checkbox"/> WQSF_I...	Maths	None		/QSF_Isb.HISOLZEN
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.SATURATEE
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.MEGLINT
<input type="checkbox"/> WQSF_I...	Maths	Green	0	WQSF_Isb.HIGHGLINT
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.WHITCAPS
<input type="checkbox"/> WQSF_I...	Maths	Red	0.5	WQSF_Isb.ADJAC
<input type="checkbox"/> WQSF_I...	Maths	Purple	0.5	WQSF_Isb.WV_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Purple	0.5	WQSF_Isb.PAR_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Olive	0.5	WQSF_Isb.AC_FAIL
<input type="checkbox"/> WQSF_I...	Maths	Grey	0.5	WQSF_Isb.OC4ME_FAI
<input type="checkbox"/> WQSF_I...	Maths	Blue	0.5	WQSF_Isb.OCNN_FAI
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.KDM_FAI
<input type="checkbox"/> WQSF_I...	Maths	Green	0.5	WQSF_Isb.BPAC_ON
<input type="checkbox"/> WQSF_I...	Maths	Cyan	0.5	WQSF_Isb.WHITE_SCA
<input type="checkbox"/> WQSF_I...	Maths	Pink	0.5	WQSF_Isb.LOWRW
<input type="checkbox"/> WQSF_I...	Maths	Orange	0.5	WQSF_Isb.HIGHRW
<input type="checkbox"/> WQSF_...	Maths	Red	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_...	Maths	Pink	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_...	Maths	Yellow	0.5	WQSF_msb.ANNOT_A
<input type="checkbox"/> WQSF_...	Maths	Dark Blue	0.5	WQSF_msb.ANNOT_A

1: 1 0°

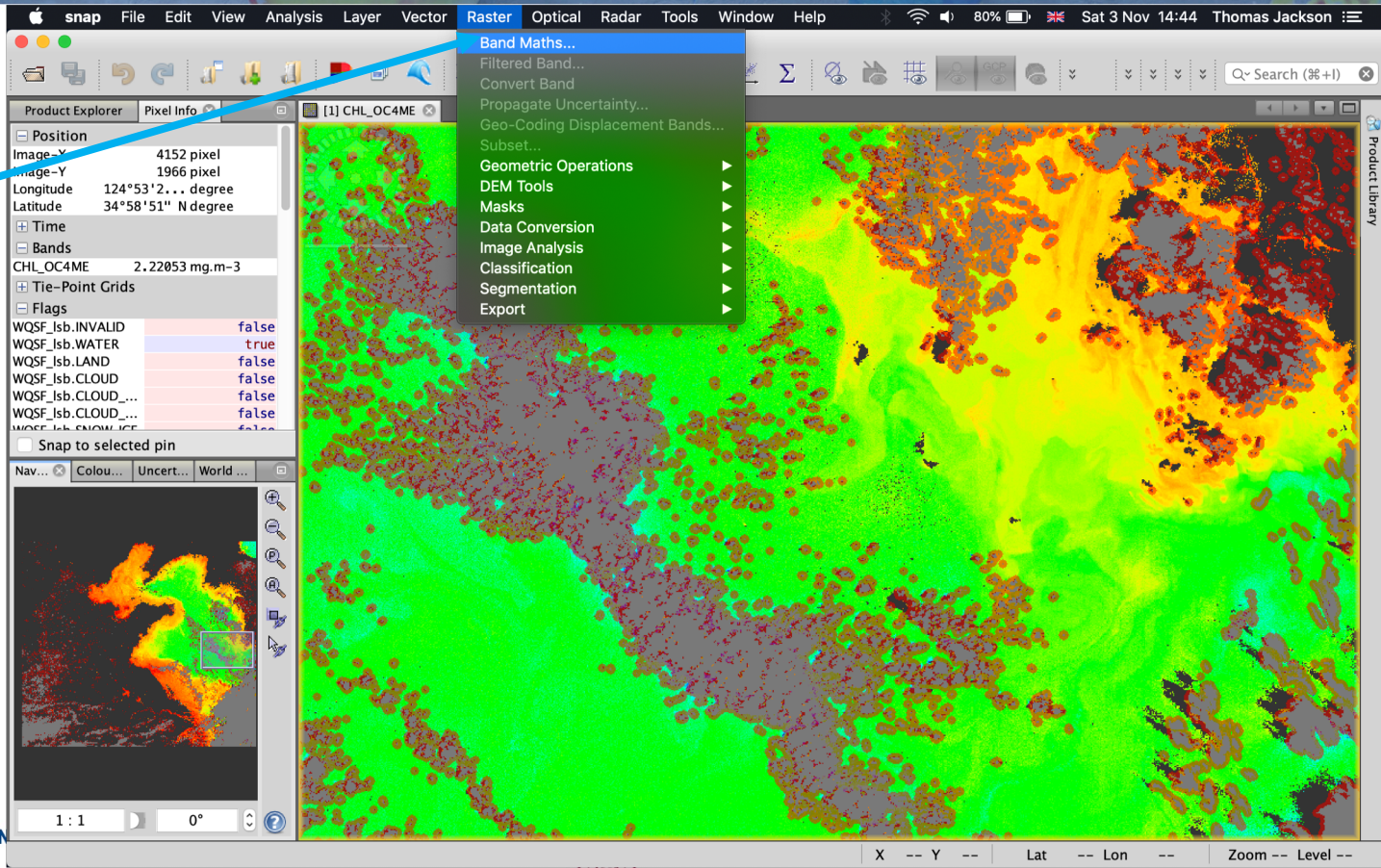
X -- Y -- Lat -- Lon -- Zoom -- Level --



# Exercise 2: Masking/filtering data

Go to the Raster section at the top of the window and select band maths.

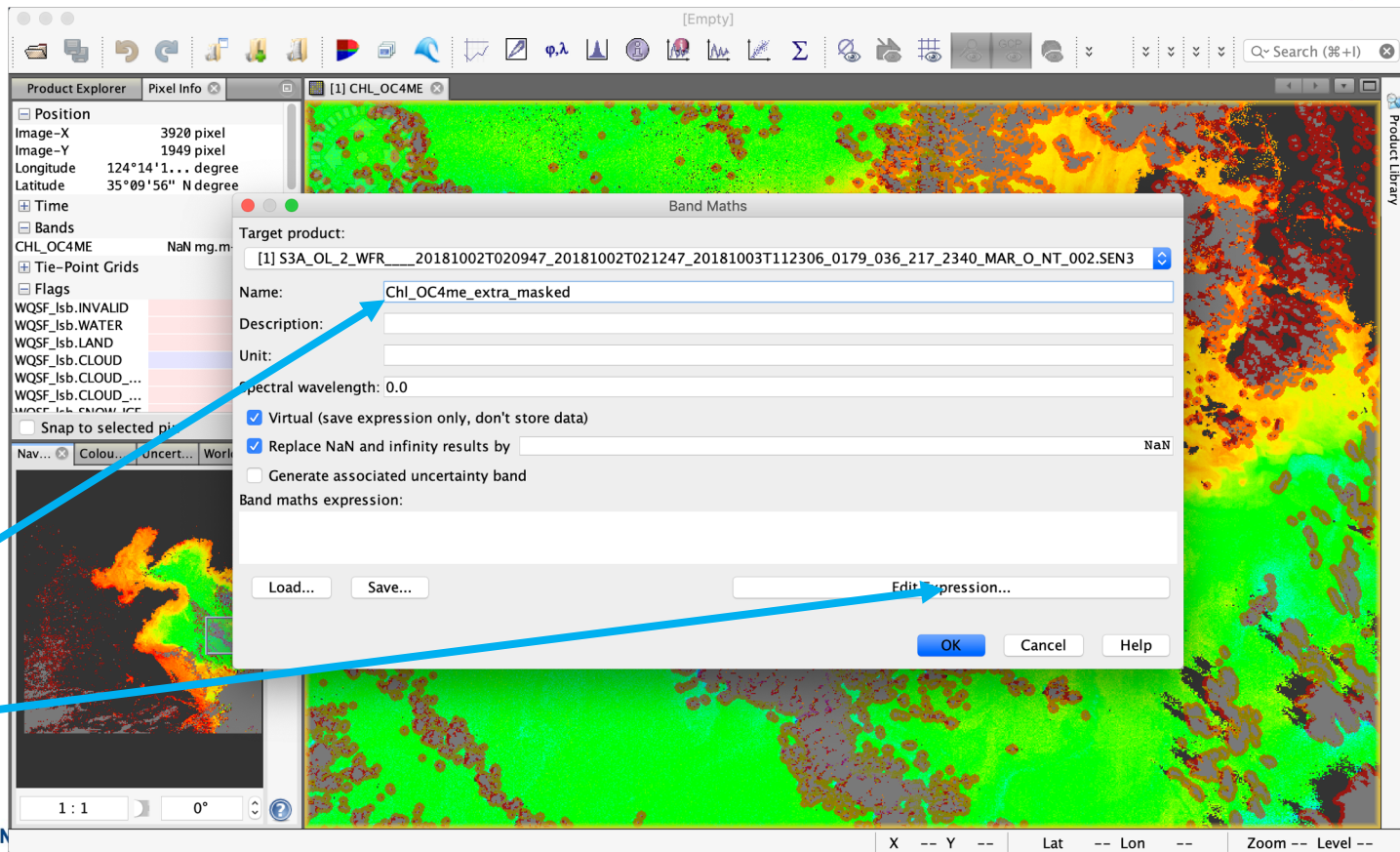
Name the new band you are going to create something appropriate, then click 'Edit Expression'.



# Exercise 2: Masking/filtering data

Go to the Raster section at the top of the window and select band maths.

Name the new band you are going to create something appropriate, then click 'Edit Expression'.



The screenshot shows a GIS application window with a 'Band Maths' dialog box open. The dialog box has the following fields and options:

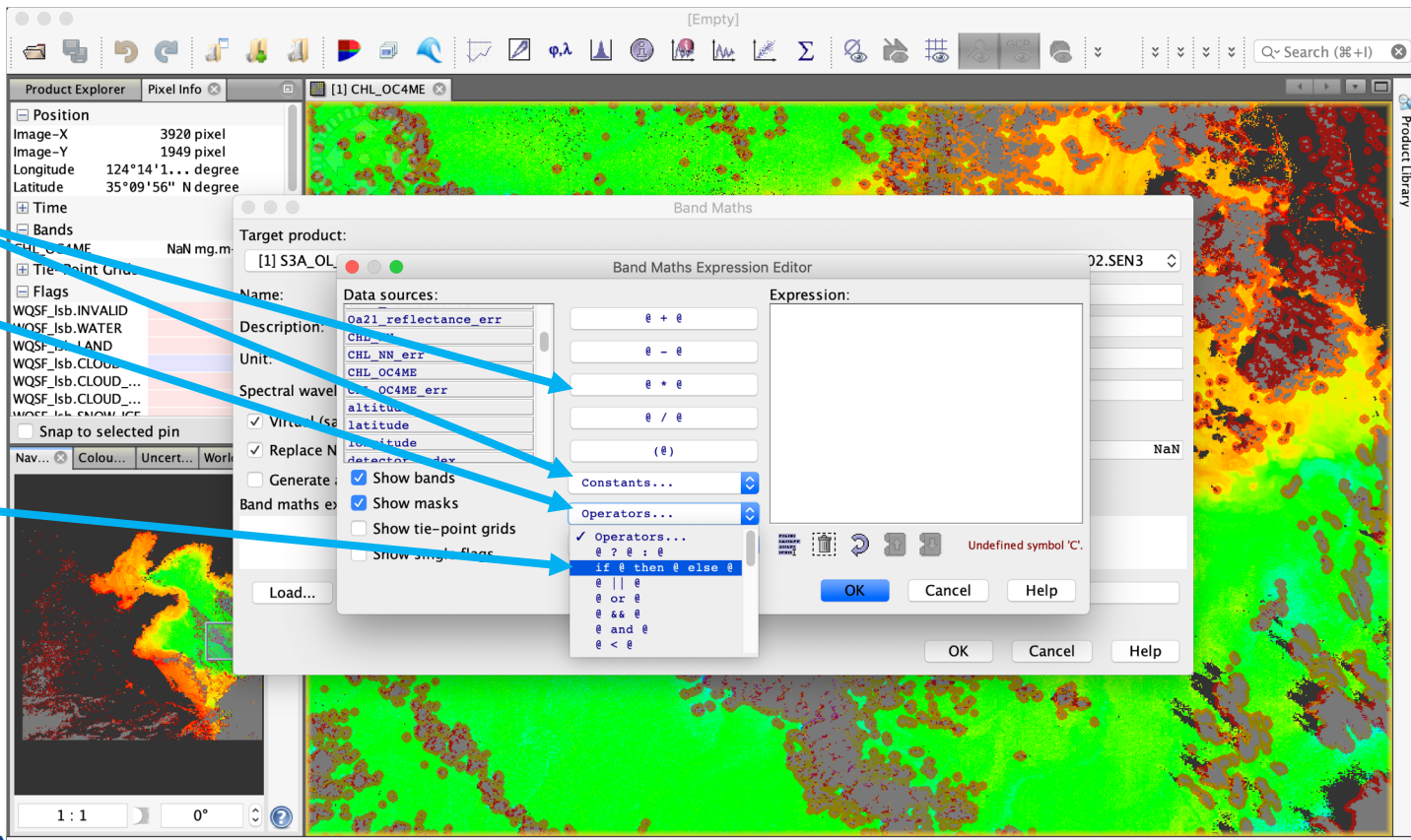
- Target product: [1] S3A\_OL\_2\_WFR\_20181002T020947\_20181002T021247\_20181003T112306\_0179\_036\_217\_2340\_MAR\_O\_NT\_002.SEN3
- Name: Chl\_OC4me\_extra\_masked
- Description: (empty)
- Unit: (empty)
- Spectral wavelength: 0.0
- Virtual (save expression only, don't store data)
- Replace NaN and infinity results by NaN
- Generate associated uncertainty band
- Band maths expression: (empty)
- Buttons: Load..., Save..., Edit Expression..., OK, Cancel, Help

# Exercise 2: Masking/filtering data

The band math expression window has a lot of functionality and you will find familiar functions, constants and operators here.

For now select the Operators section and choose 'if @ then @ else @'.

Each @ symbol will need to be replaced with a variable, constant or logical comparison.



# Exercise 2: Masking/filtering data



Masks values are 'TRUE' or 'FALSE'.

We want to remove chl if the cloud margin mask=TRUE.

The equation is:  
**if** WQSF\_Isb\_CLOUD\_MARGIN==false **then** CHL\_OC4ME **else** NaN

Note, it tells you when your equation is valid.

Product Explorer Pixel Info [1] CHL\_OC4ME

Position  
Image-X 3920 pixel  
Image-Y 1949 pixel  
Longitude 124°14'1... degree  
Latitude 35°09'56" N degree

Time  
Bands  
CHL\_OC4ME NaN mg.m

Tie-Point Grids

Flags  
WQSF\_Isb.INVALID  
WQSF\_Isb.WATER  
WQSF\_Isb.LAND  
WQSF\_Isb.CLOUD  
WQSF\_Isb.CLOUD\_...  
WQSF\_Isb.CLOUD\_...  
WQSF\_Isb.CLOUD\_...

Snap to selected pin

Band Maths

Target product:  
[1] S3A\_OL

Band Maths Expression Editor

Name:  
Description:  
Unit:  
Spectral wavel:

Data sources:  
CHL\_NN  
CHL\_NN\_err  
CHL\_OC4ME  
CHL\_OC4ME\_err  
altitude  
longitude  
latitude  
detector\_index  
FWHM\_band\_1

Expression:  
if WQSF\_Isb\_CLOUD\_MARGIN==false then  
CHL\_OC4ME else NaN

Operators...  
Functions...

Constants...  
Operators...  
Functions...

Band maths e:  
 Show bands  
 Show masks  
 Show tie-point grids  
 Show single flags

Ok, no errors.

OK Cancel Help

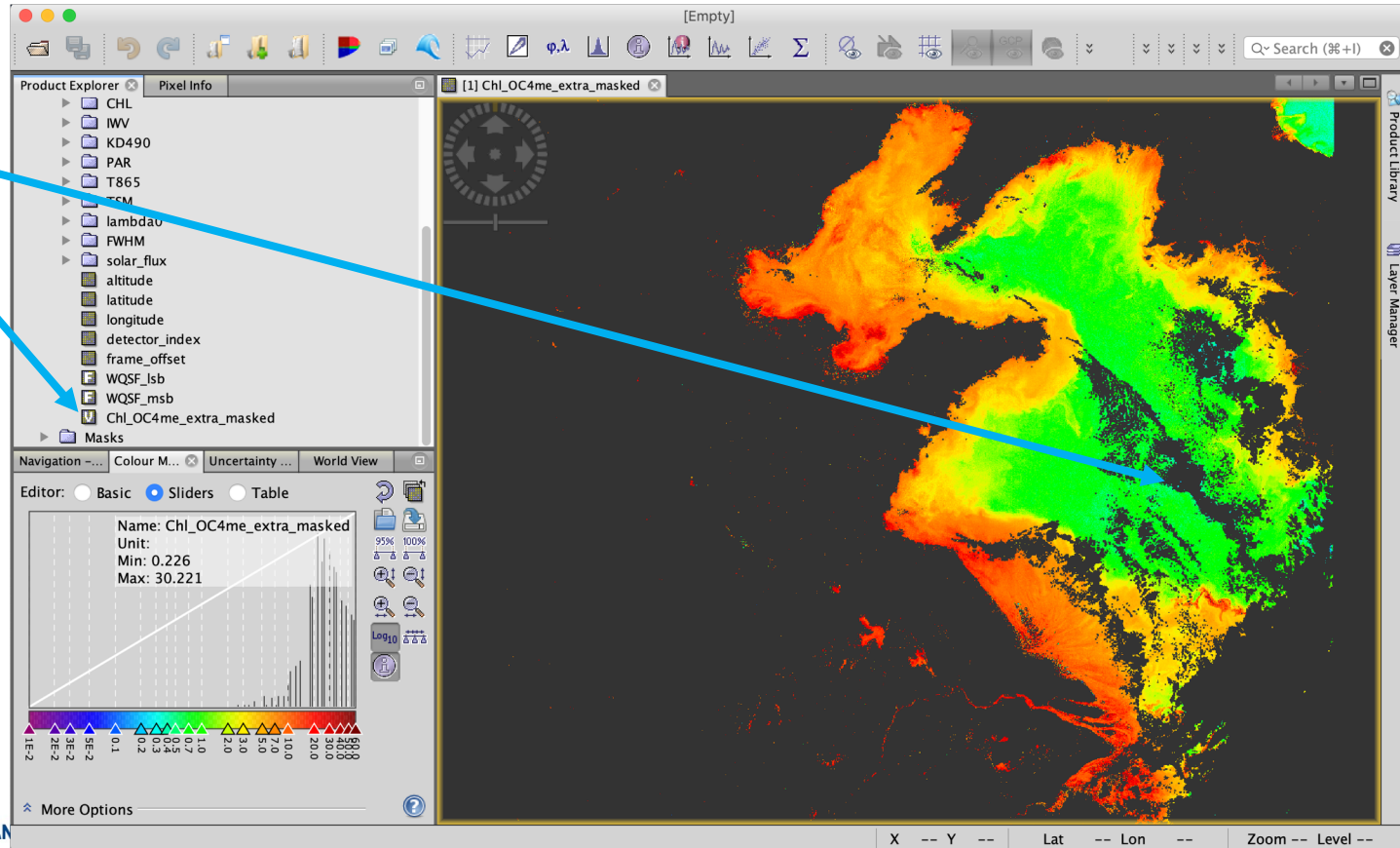
# Exercise 2: Masking/filtering data



You should now have a new band. It will be missing bad data from cloud margins. You can modify the colour scale on this to match the other chl product.

Feel free to experiment with other masks/mathematical operations. Perhaps you are only interested in low chl waters. You could apply a filter on pixels that have a chl value  $> 5.0$ .

→ ADVANCED TRAINING COURSE IN OCEAN

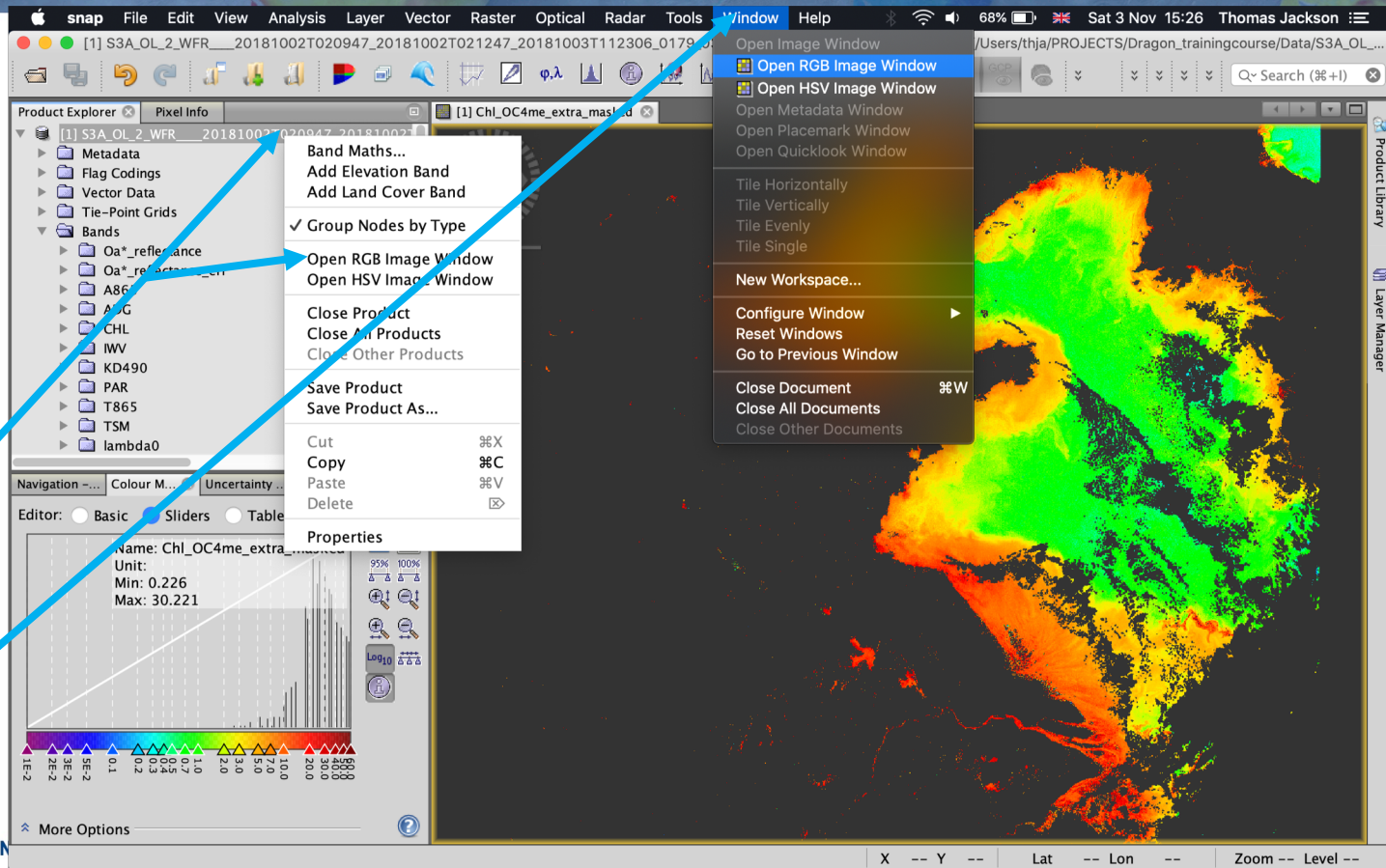


# Exercise 3: Tri-stimulus 'true colour'



Creating the 'true colour' image requires the RGB image window.

This can be accessed by right clicking on the top level folder  
**Or**  
through the 'Window' options



# Exercise 3: Tri-stimulus 'true colour'



If you want to create your own RGB image then you can tailor the equations here but for now keep the default for the OLCI L2 W Tristimulus.

Later why not come back and try something different.

The screenshot shows a software interface with a satellite image of a coastal area. A dialog box titled "Select RGB-Image Channels" is open, showing the following configuration:

- Profile: OLCI L2 W - Tristimulus
- Red:  $a_{09\_reflectance} + 0.04 * a_{a10\_reflectance}$
- Green:  $a_{09\_reflectance} + 0.02 * a_{a10\_reflectance}$
- Blue:  $a_{04\_reflectance} + 0.16 * a_{a05\_reflectance}$
- Store RGB channels as virtual bands in current product

The background interface includes a Product Explorer on the left, a main image view, and a bottom status bar with coordinates and zoom controls.

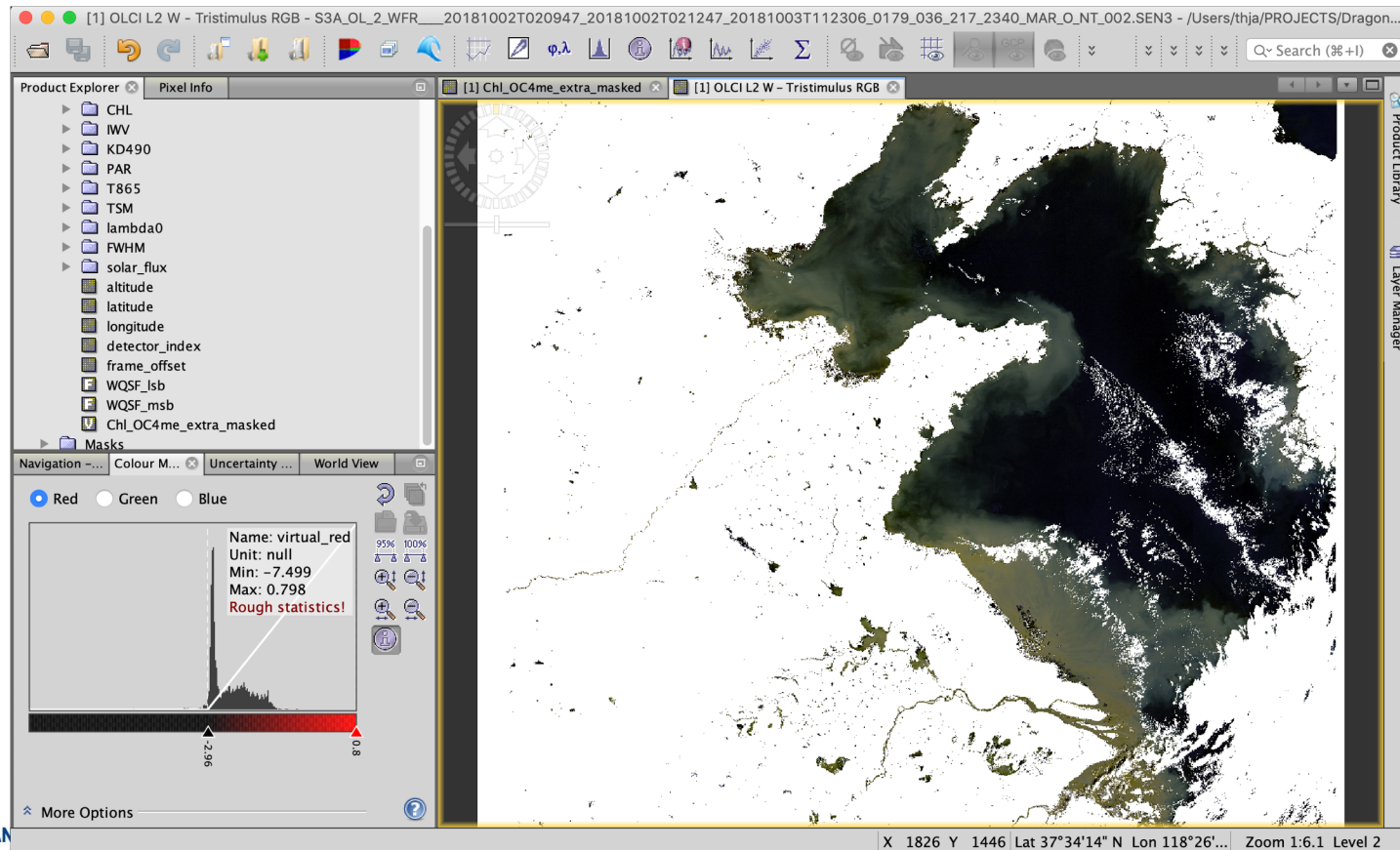


# Exercise 3: Tri-stimulus 'true colour'



This should give you an image that looks like this.

Here you can clearly identify sediment plumes and the influence of the Yangtze river.



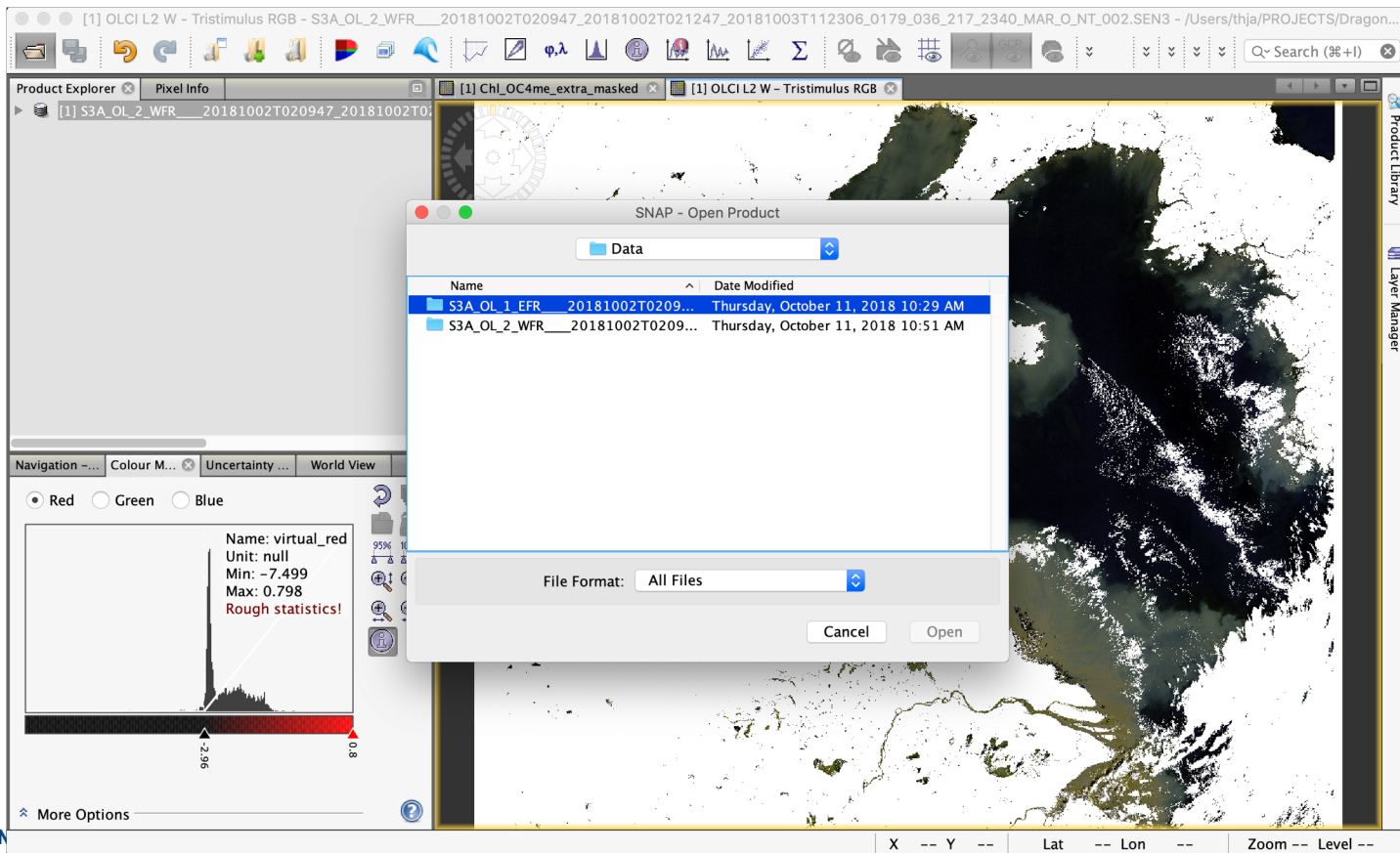


# Exercise 3: Tri-stimulus 'true colour'



Now load the Level 1 file that corresponds to the level 2 file you have just processed.

Again, load the xfdumanifest.xml to give access to all the variables etc.





# Exercise 3: L2 to Tri-stimulus 'true colour'



This should be the image you are provided with after L1 tristimulus.

Compare L1 and L2 'true colour' image and consider the location of the 'viewer' (one is 'top of atmosphere', the other at the surface)

Product Explorer

- [1] S3A\_OL\_2\_WFR\_20181002T020947\_20181002T02
- [2] S3A\_OL\_1\_EFR\_20181002T020947\_20181002T02
- Metadata
- Flag Codings
- Vector Data
- Tie-Point Grids
- Bands
  - Oa\*\_radiance
  - lambda0
  - FWHM
  - solar\_flux
  - altitude
  - latitude
  - longitude
  - detector\_index
  - frame\_offset
  - quality\_flags

Navigation ... Colour M... Uncertainty ... World View

Red Green Blue

Name: virtual\_red  
Unit: null  
Min: 3.81  
Max: 7.316  
Rough statistics!

3.81 7.32

More Options

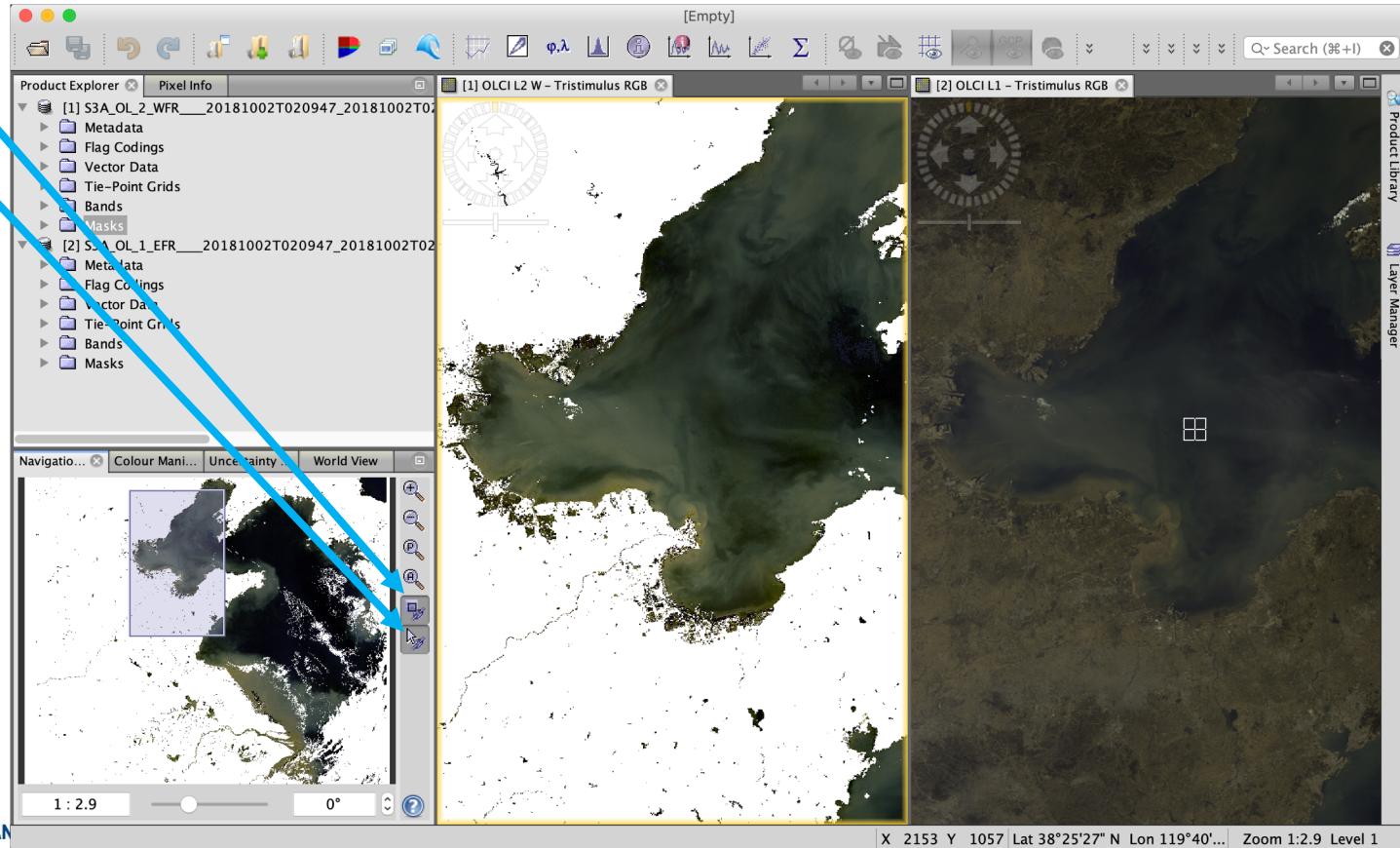
X -- Y -- Lat -- Lon -- Zoom -- Level --

# Exercise 3: L2 to Tri-stimulus 'true colour'



Note, by using the 'synchronise view' and 'synchronise cursor' options on the navigation tab you can easily compare products side by side.

The 'Window→Tile evenly' option gives a simple split screen view.



# Exercise 4: Custom products

In this exercise we will create a tailored chlorophyll-a product following the formulation of Shen et al (2010) which was designed to provide MERIS Chl-a products in the sediment rich waters of the (Changjiang) Yangtze.

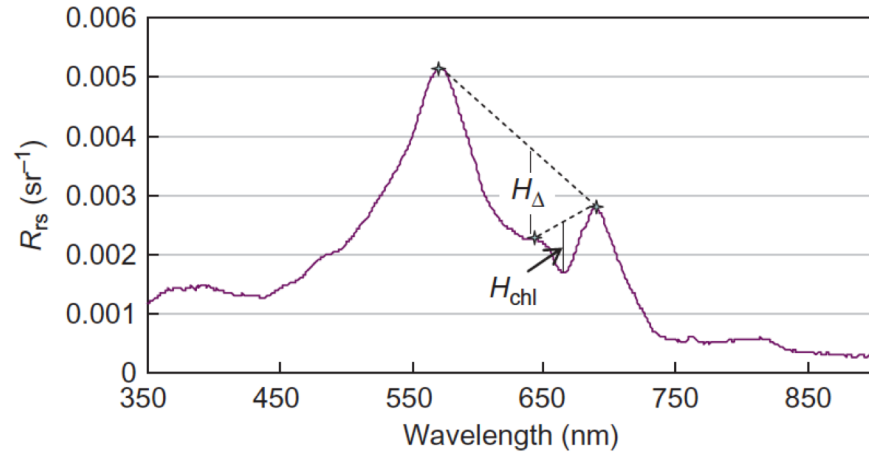


Figure 3. Sketch map of the synthetic chlorophyll index (SCI) designed by modelling of remote-sensing reflectance ( $R_{rs}$ ) and chlorophyll-a concentration (Chl-a).  $H_{chl}$  is a measure of the distinct absorption dip of the reflectance spectrum at the MERIS band 665 nm below (positive) and above (negative) the baseline connecting the  $R_{rs}$  at MERIS bands 620 and 681 nm (620 and 681 nm are reference wavebands).  $H_{\Delta}$  is the relative height of  $R_{rs}$  at 620 nm below (negative) and above (positive) the baseline (two reference wavebands: 560 and 681 nm).

Shen, *et al.* (2010). Medium resolution imaging spectrometer (MERIS) estimation of chlorophyll-a concentration in the turbid sediment-laden waters of the Changjiang (Yangtze) Estuary. *International Journal of Remote Sensing* - INT J REMOTE SENS. 31. 4635-4650.

# Exercise 4: Custom products

The Algorithm consists of 4 steps.

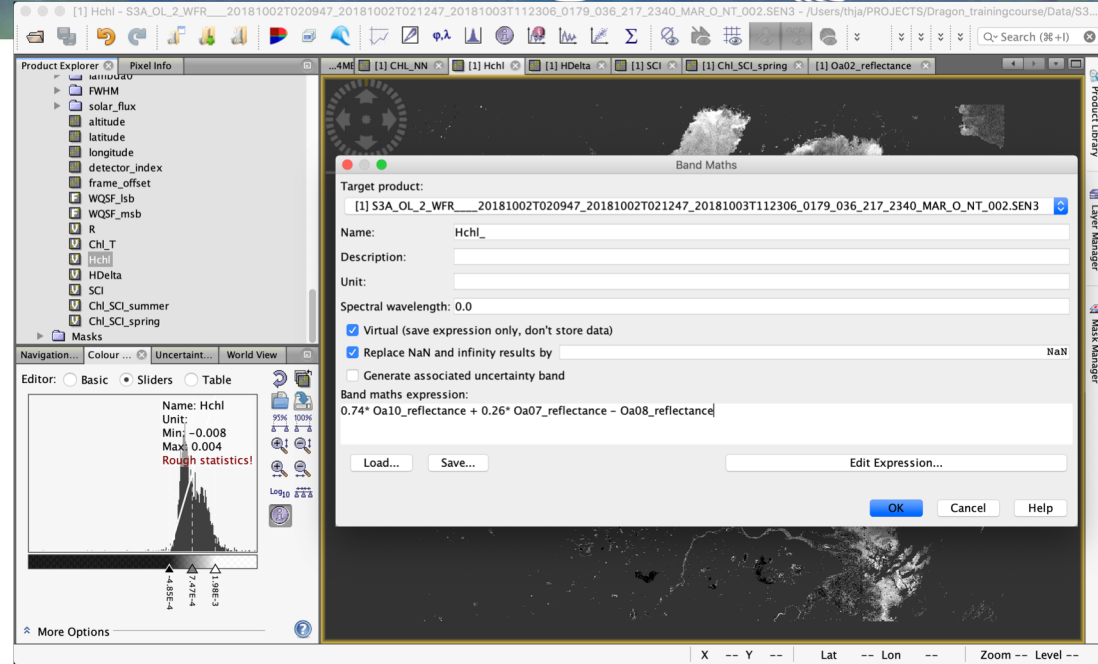
1) Calculate Hchl.

2) Calculate HΔ

3) Calculate the SCI

4) Convert the SCI to Chl-a

Remember that for OLCI:  
560nm=band6  
620nm=band7  
665nm=band8  
681nm=band 10



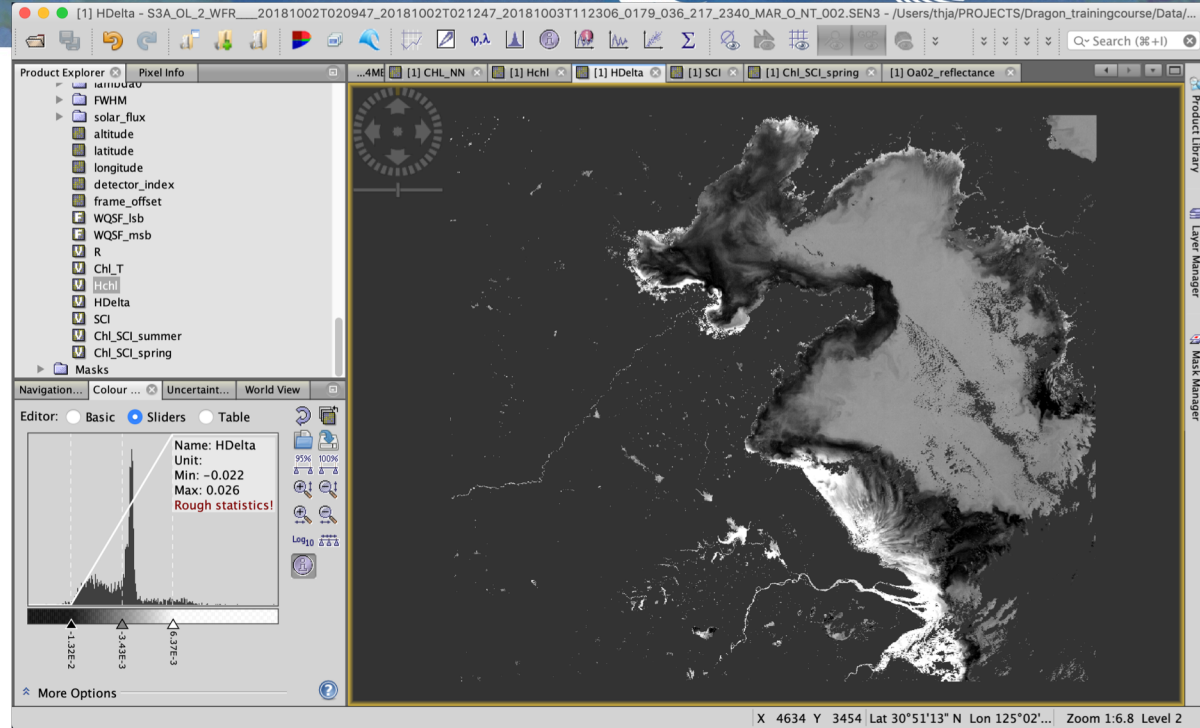
Following Shen et al (2010) equation 2  
 $Hchl = 0.74 * Rrs.681 + 0.26 * Rrs.620 - Rrs.665$   
(Or as you would type in SNAP)

$0.74 * Oa10\_reflectance + 0.26 * Oa07\_reflectance - Oa08\_reflectance$

# Exercise 4: Custom products

The Algorithm consists of 4 steps.

- 1) Calculate Hchl.
- 2) Calculate HΔ
- 3) Calculate the SCI
- 4) Convert the SCI to Chl-a



$$H\Delta = Rrs.620 - 0.5 * (Rrs.560 + Rrs.681)$$

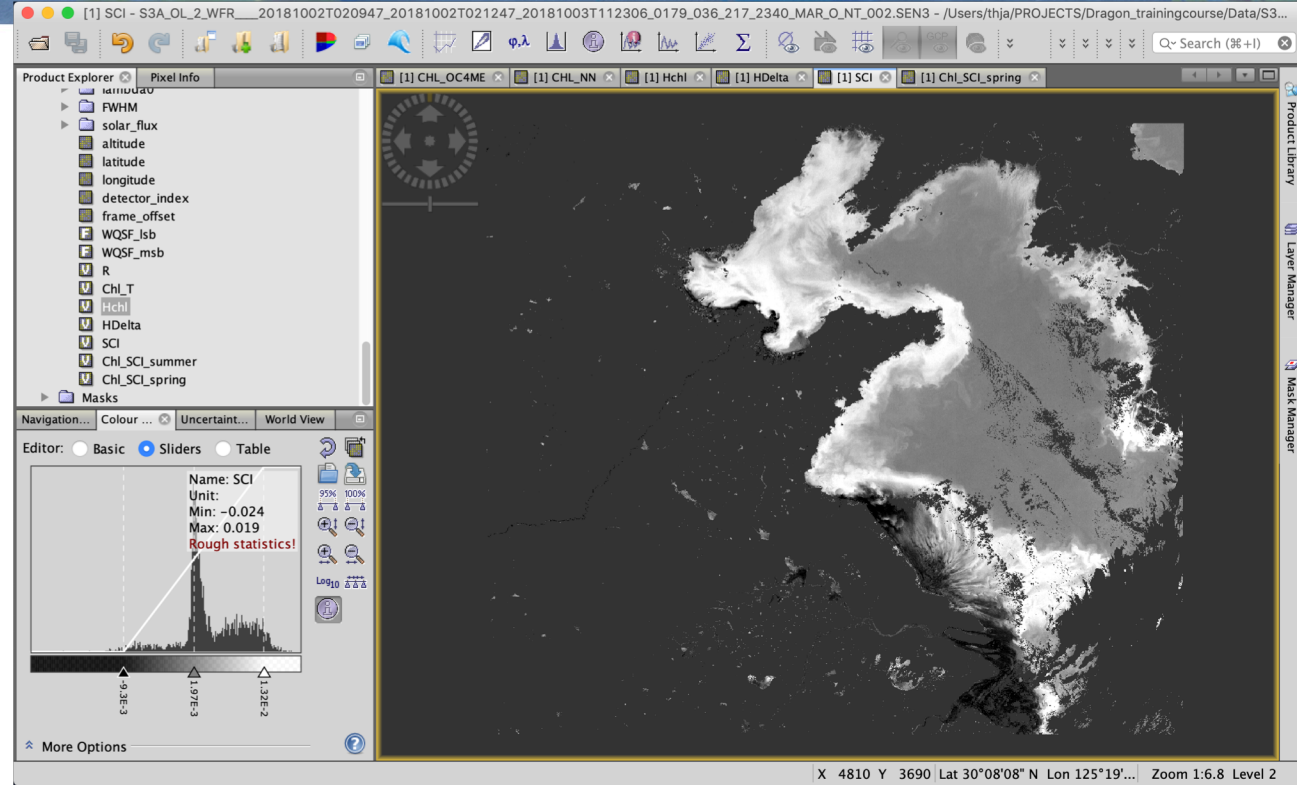
(Or as you would type in SNAP)

$$Oa07\_reflectance - 0.5 * (Oa06\_reflectance + Oa10\_reflectance)$$

# Exercise 4: Custom products

The Algorithm consists of 4 steps.

- 1) Calculate Hchl.
- 2) Calculate  $H\Delta$
- 3) Calculate the SCI
- 4) Convert the SCI to Chl-a



$$SCI = Hchl - H\Delta$$



# Exercise 4: Custom products

The Algorithm consists of 4 steps.

- 1) Calculate Hchl.
- 2) Calculate  $H\Delta$
- 3) Calculate the SCI

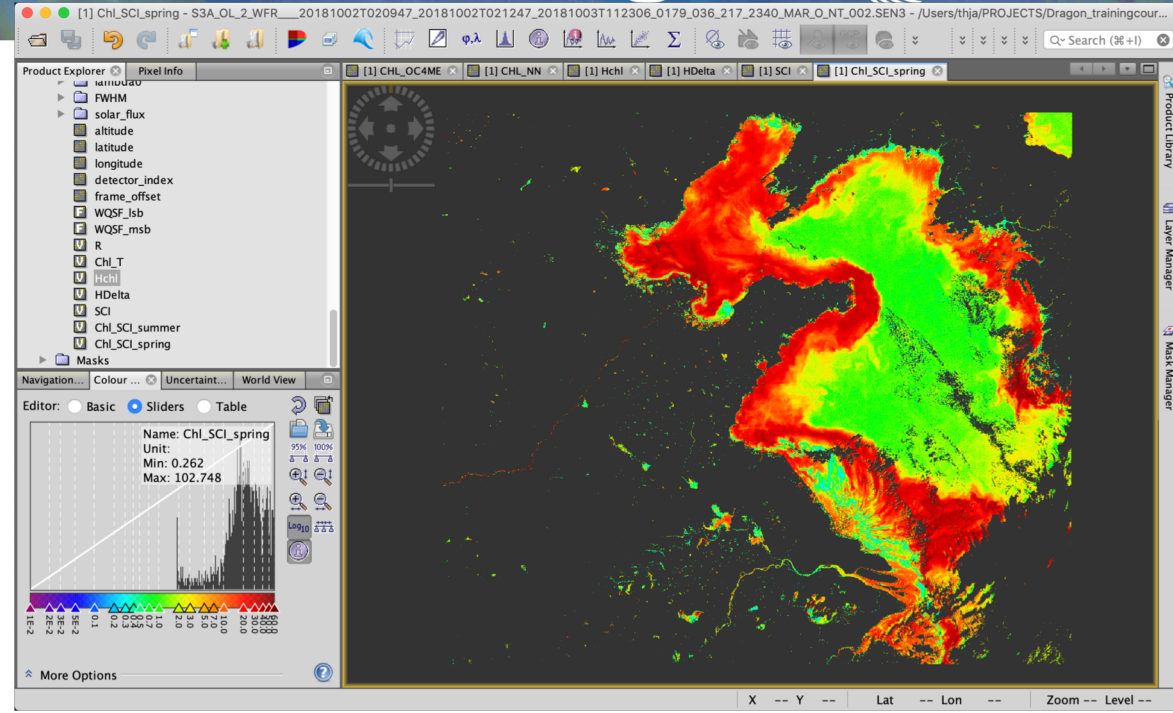
As this image is from Autumn (and the paper only gives conversion factors for spring and summer) we will use the spring conversion factor here.

4) Convert the SCI to Chl-a

$$\text{Chl} = 179378 \cdot \text{SCI}^2 + 92.934 \cdot \text{SCI} + 0.2736$$

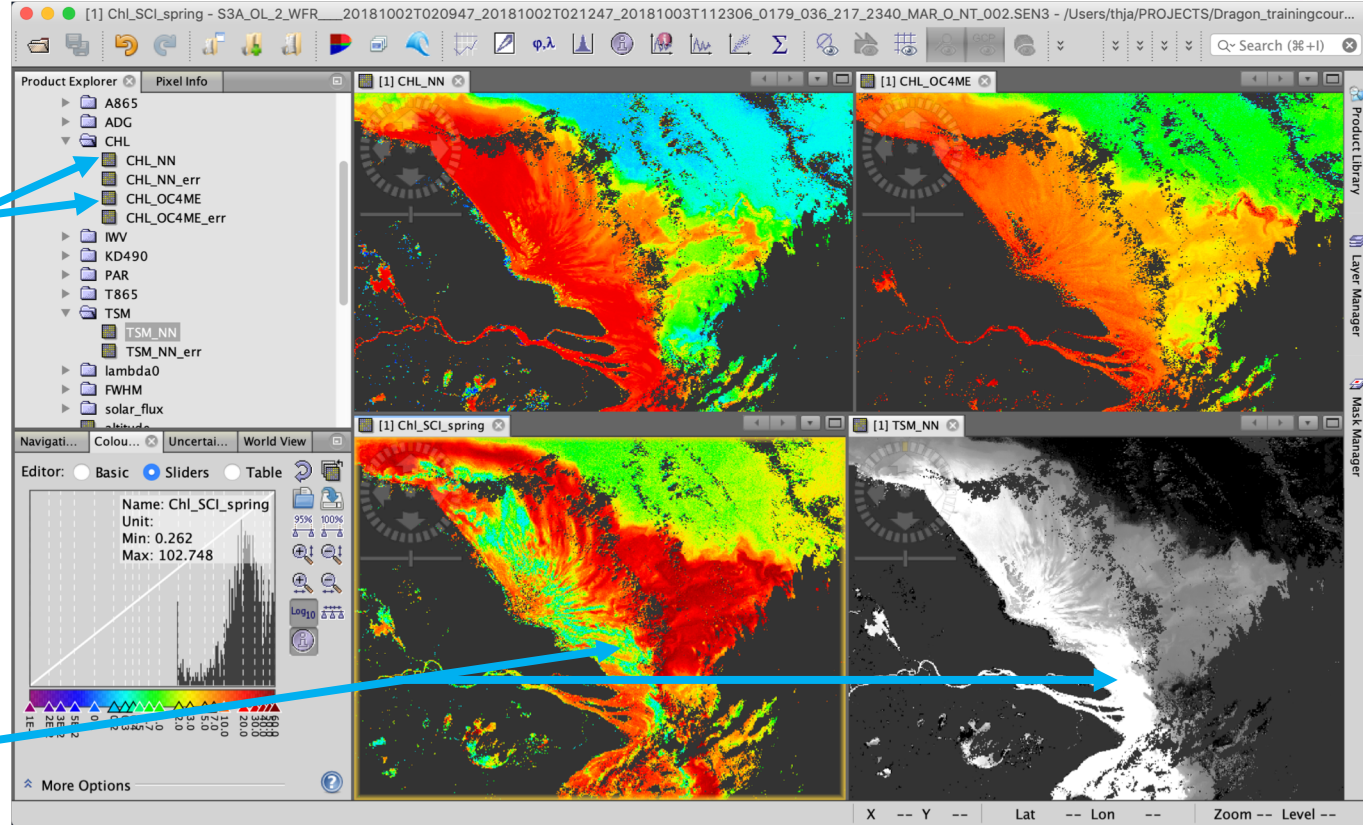
(Or as you would type in SNAP)

$$179378 \cdot \text{pow}(\text{SCI}, 2) + 92.934 \cdot \text{SCI} + 0.2736$$



# Exercise 4: Custom products

This means that you should now have 3 chlorophyll-a estimates. The OLCI product contains 2 (OC4me and NN) and you have created one here. If you also add the TSM (total suspended matter) product to the view you can see how the Shen 2010 algorithm reduces chl estimates in high sediment waters.



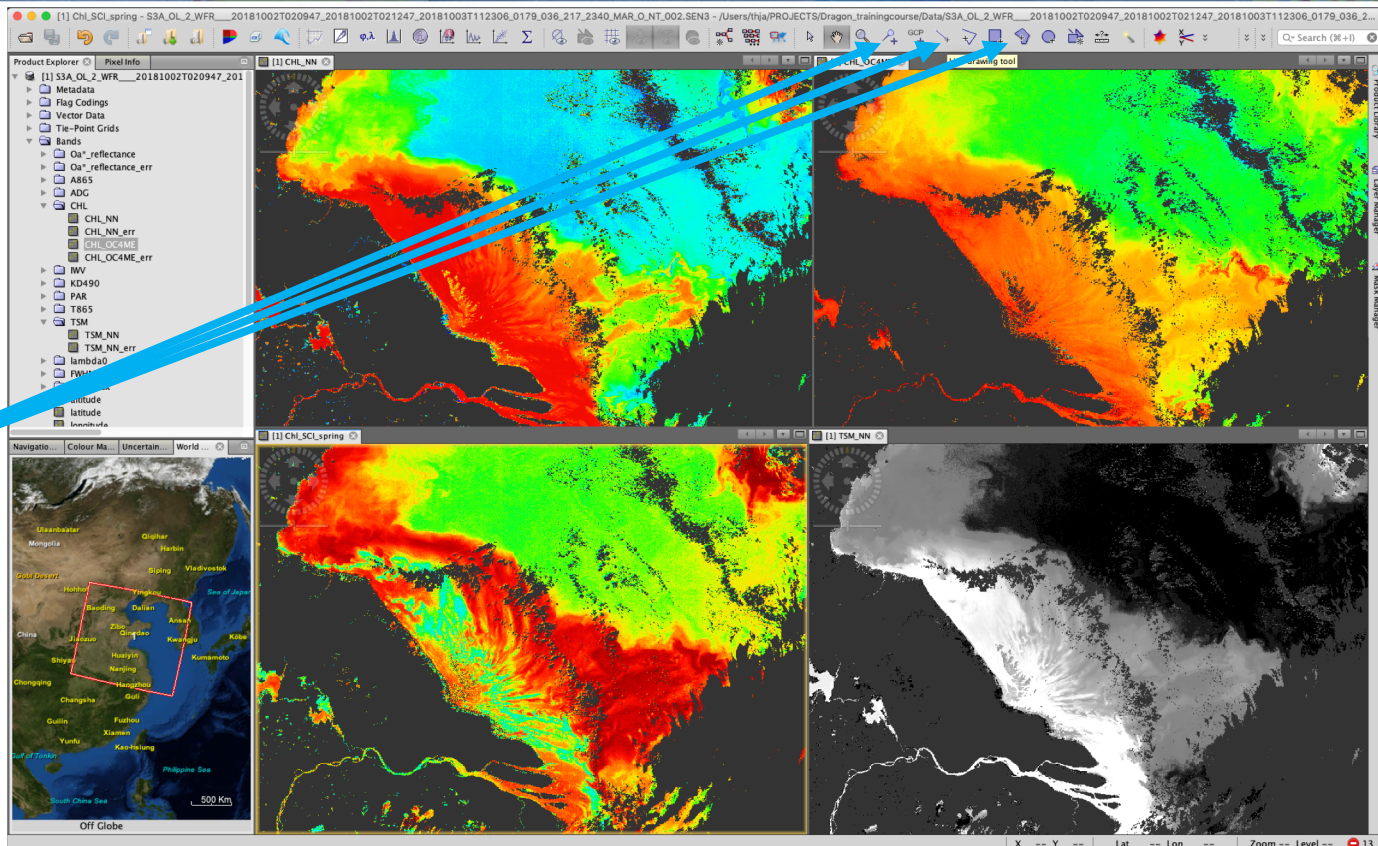
Unfortunately, without in-situ data we cannot prove which of the chlorophyll-a algorithms is giving the most accurate results for the image but for now we can at least look at where the estimates agree and disagree and that in itself will tell us something about the waters that are being observed.

We will now move onto extracting data from given pixels in an image. This could be used for match-ups with in-situ data or it can be used to extract transect/profiles.

# Exercise 5: Extracting pixel data

There are a number of tools for extracting pixel data based on pins, lines and polygons. These can be found in the toolbar.

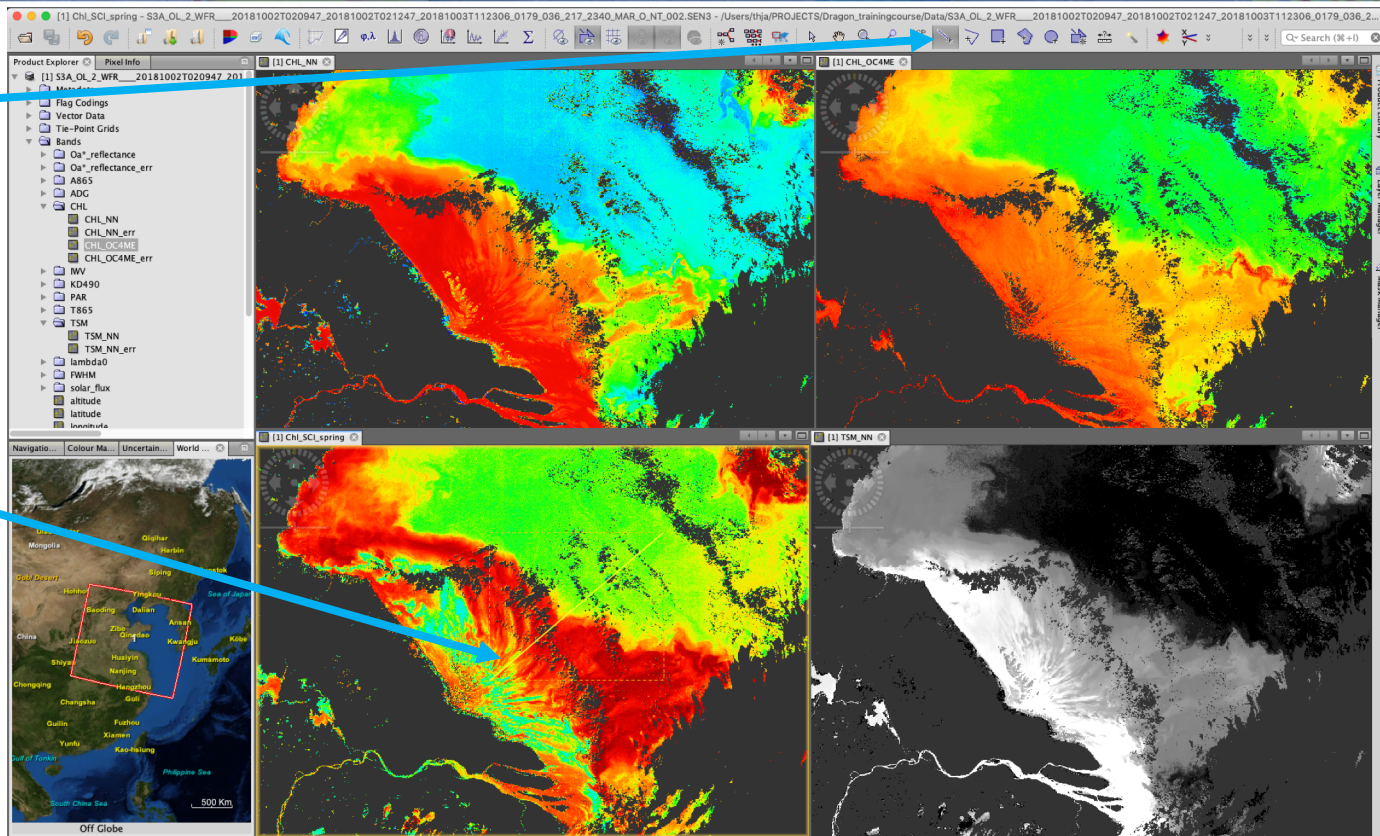
You can also read in tracks, polygons, etc from files.



# Exercise 5: Extracting pixel data

Begin by selecting 'line' and drawing (click and hold) a transect running offshore from the sediment rich waters to the clearer ocean waters.

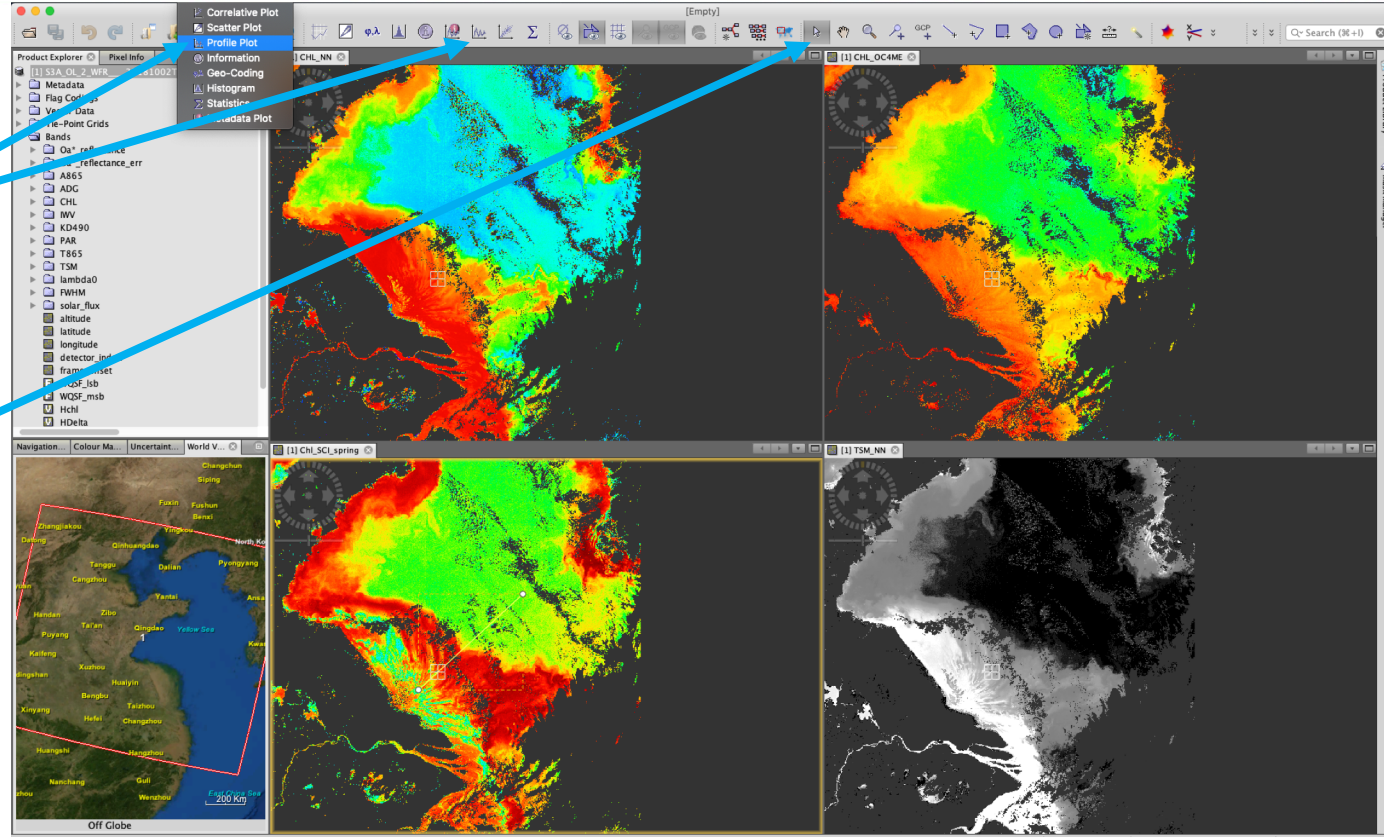
Here I have drawn on the CHL\_SCI product



# Exercise 5: Extracting pixel data

Now select profile plot to generate a plot of the transect.

If you want to edit the line then switch to 'select' mode and you can edit the line.

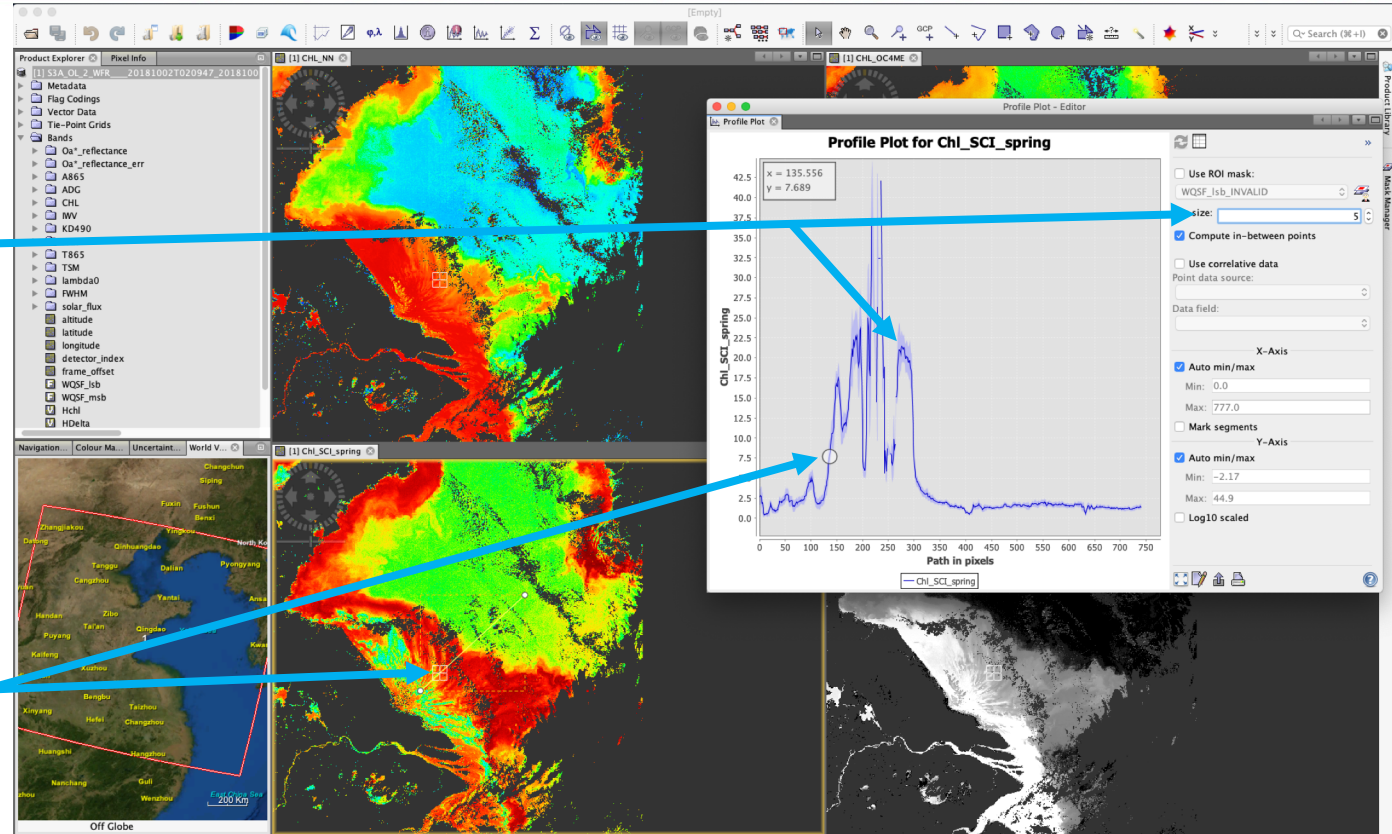


# Exercise 5: Extracting pixel data

This should produce a profile plot similar to that shown here.

Editing the 'bin width' modifies smoothing along the line and provides information on the variability of point along the transect (shaded area is  $\pm \sigma$ ).

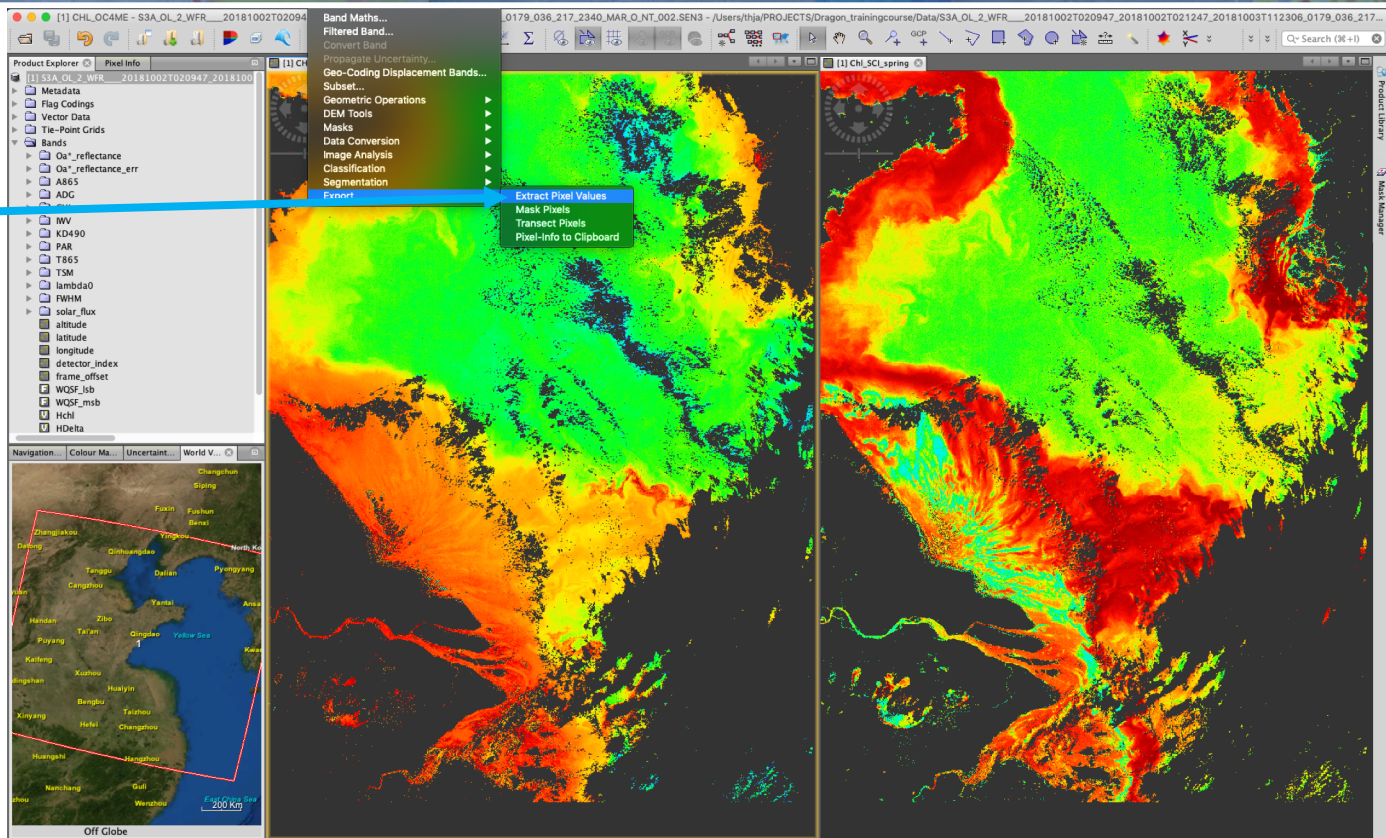
As you mouse over the plot the map location is shown.



# Exercise 5: Extracting pixel data

Begin by clicking  
Raster → Export  
→ Export Pixel  
Values

Please also take  
a minute to  
browse the  
options available  
under the  
'Analysis',  
'Raster' and  
'Optical' program  
tabs.





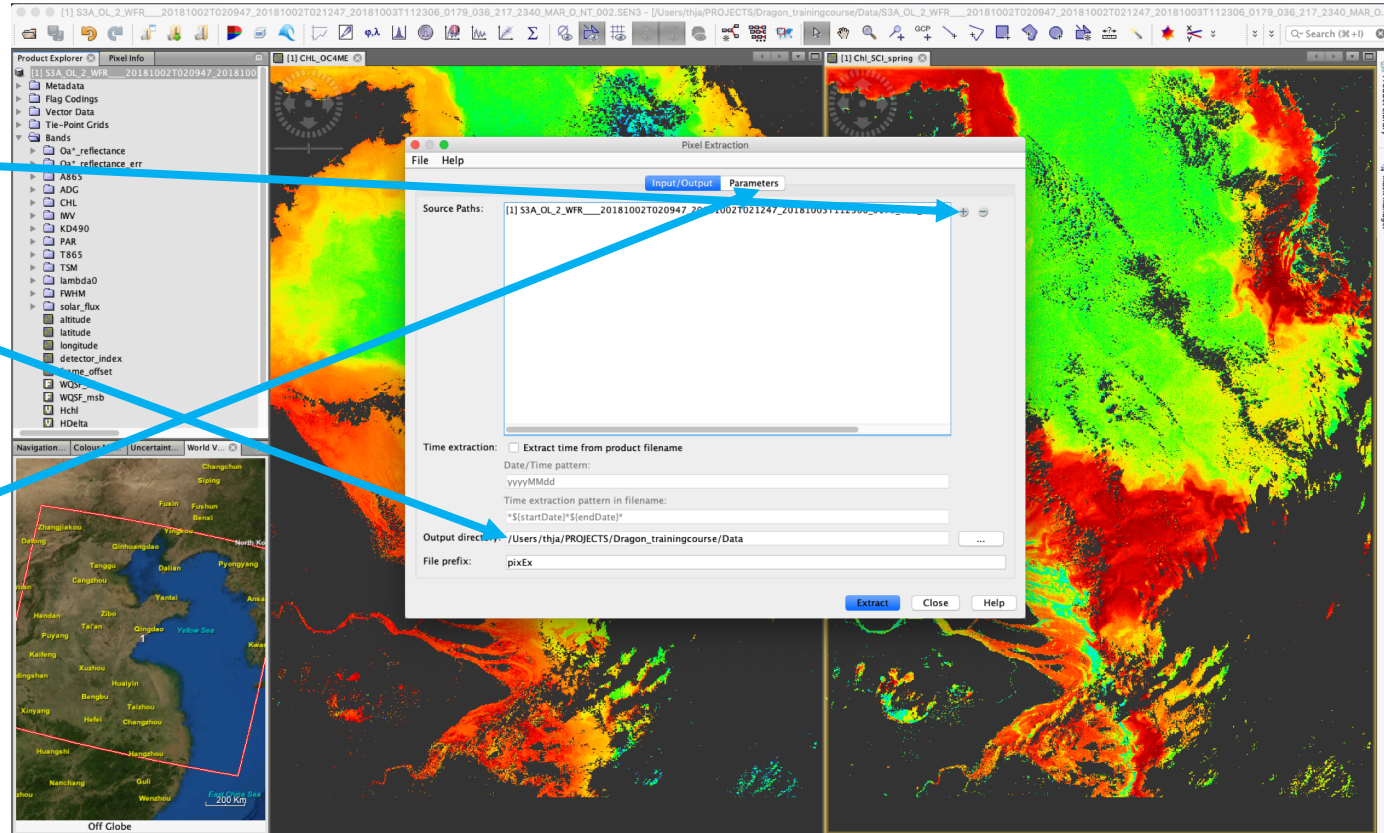
# Exercise 5: Extracting pixel data



Add the current file/products to the source path.

Set the output directory to a desired location.

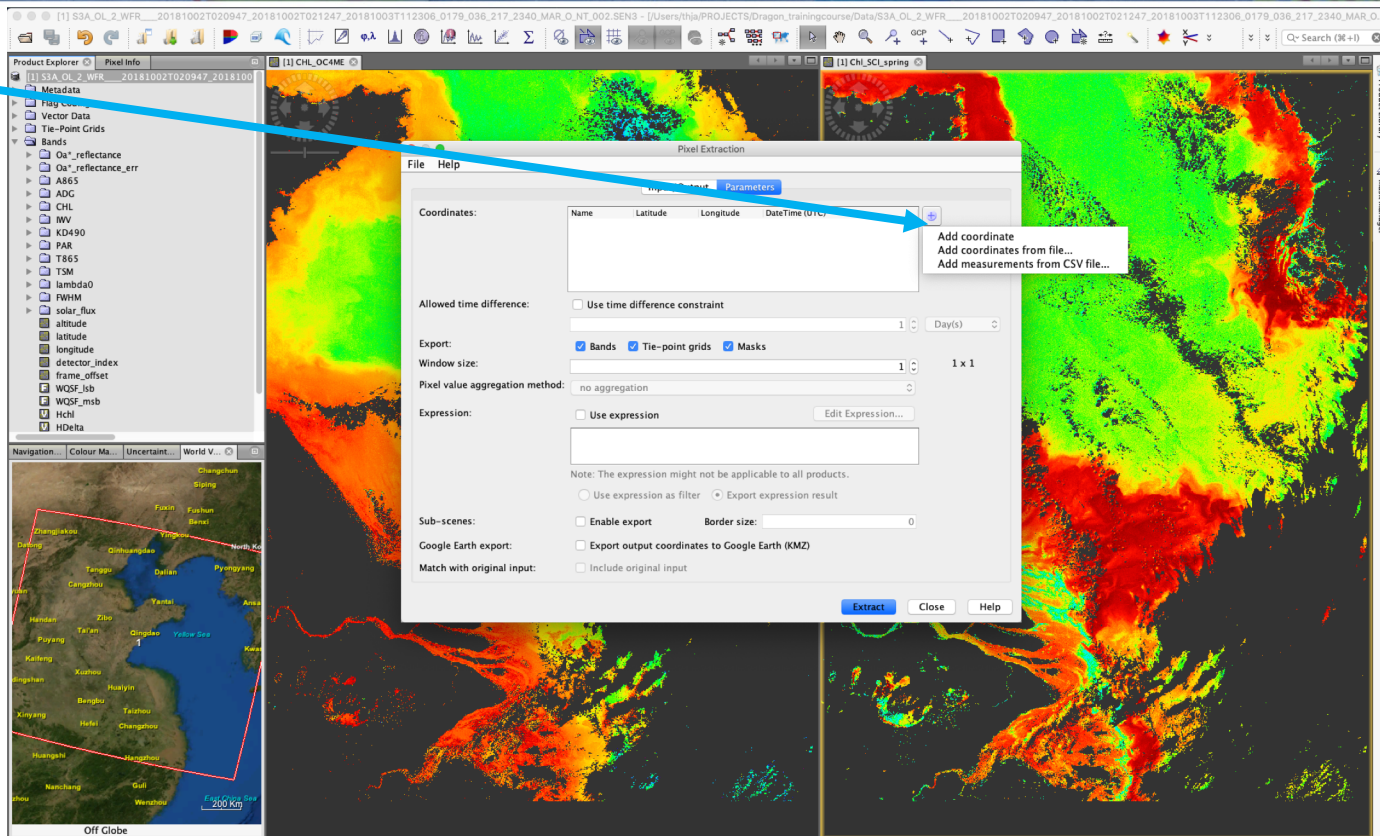
Next, select the 'Parameters' section and we can read in a set of locations for matchup.



# Exercise 5: Extracting pixel data

Now we can add locations from a csv.

You can also add individual points by typing in the 'coordinates' window but this is time consuming.



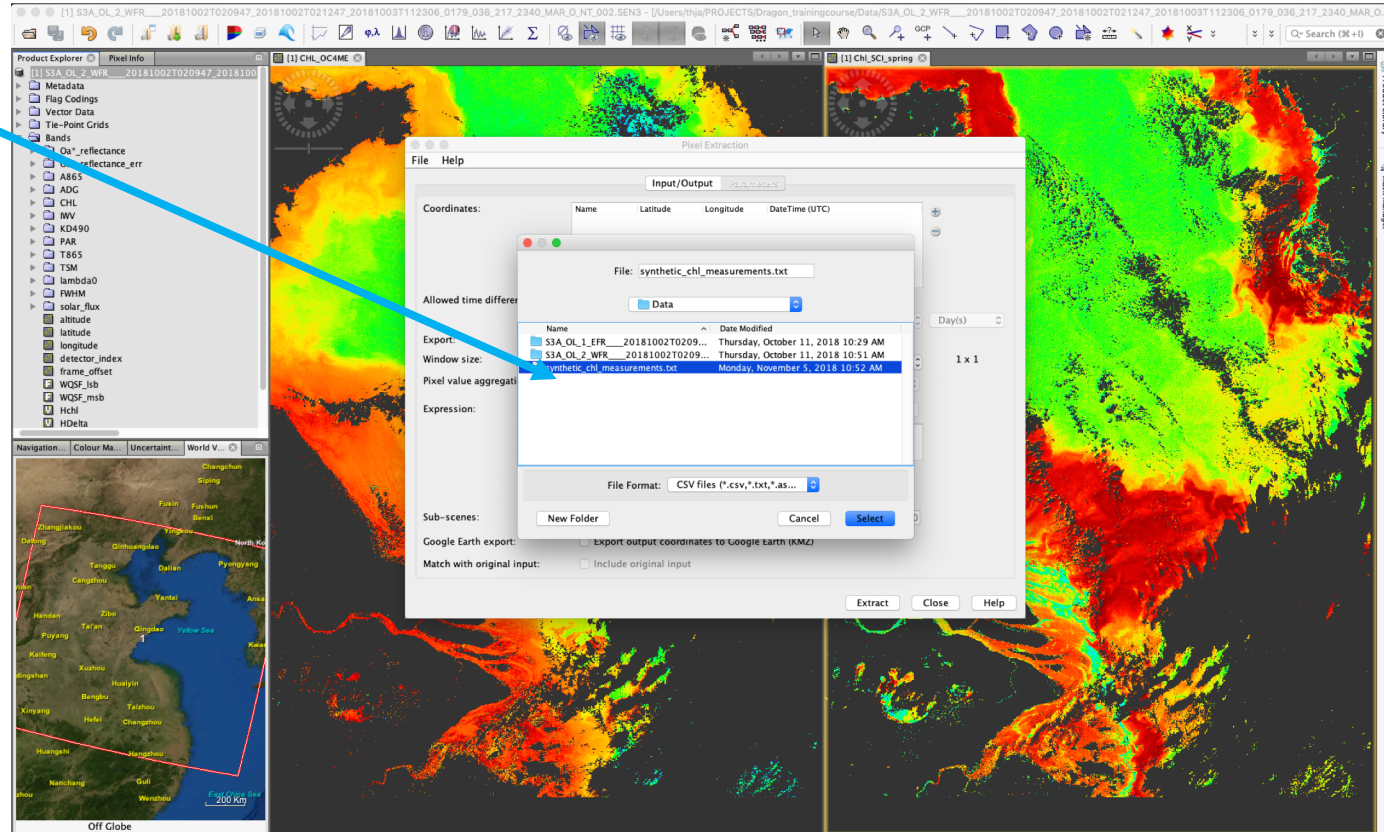
# Exercise 5: Extracting pixel data



Select the file 'synthetic\_chl\_measurements.txt' from the Data folder.

This should populate the Coordinates window.

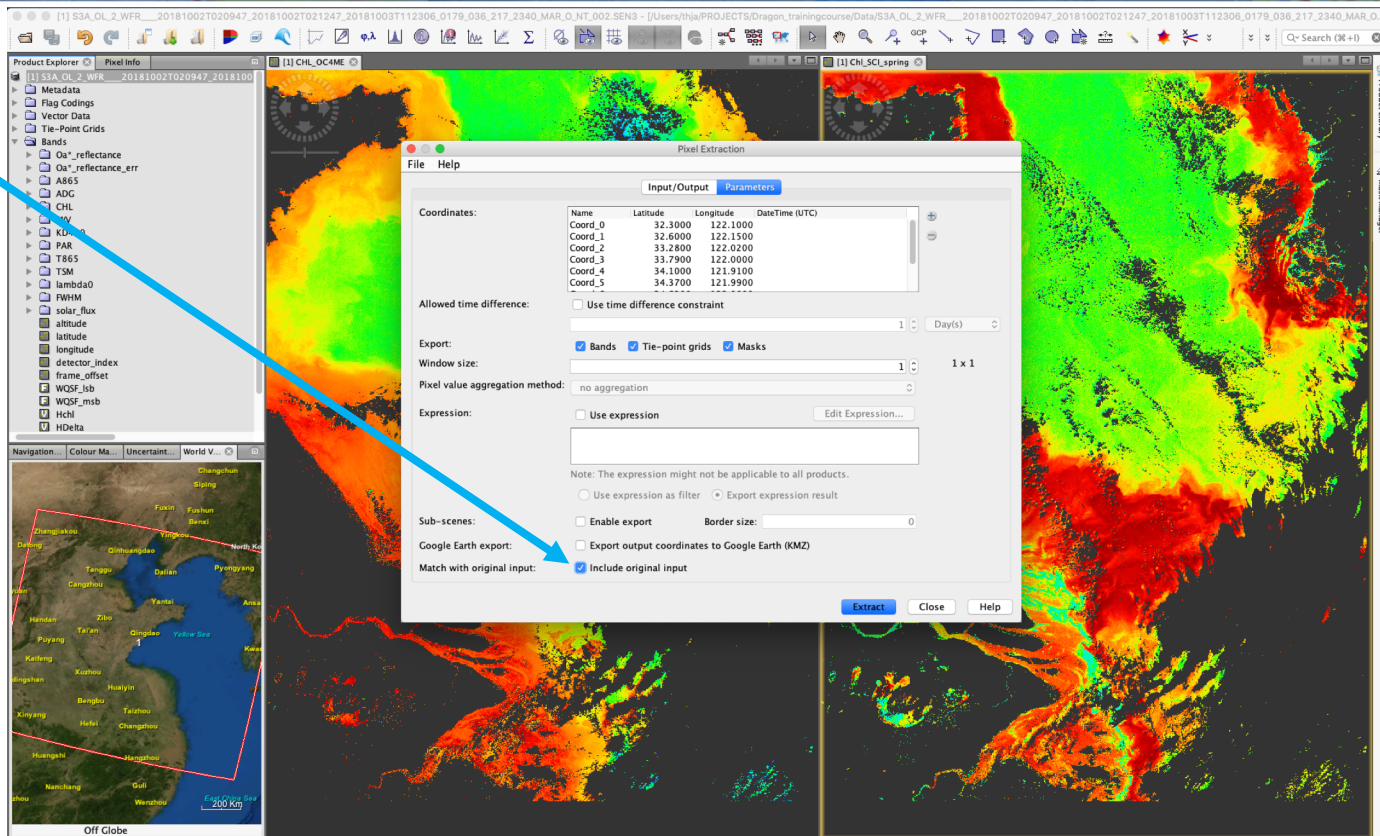
NOTE: This file reader requires a specific tab-delimited file format (the term CSV is not technically correct here)!



# Exercise 5: Extracting pixel data

Ensure you have ticked the 'include original input' option in order to preserve the original input information in the output data file.

If you don't do this you will have to manually combine the 'in-situ' and 'satellite match' data yourself.



The screenshot shows a software interface with a 'Pixel Extraction' dialog box open. The dialog has two tabs: 'Input/Output' and 'Parameters'. The 'Parameters' tab is active, showing the following options:

- Coordinates: A table with columns 'Name', 'Latitude', 'Longitude', and 'DateTime (UTC)'. The table contains 6 rows of coordinate data.
- Allowed time difference:  Use time difference constraint. A dropdown menu shows '1' and 'Day(s)'.
- Export:  Bands,  Tie-point grids,  Masks.
- Window size: A dropdown menu shows '1' and '1 x 1'.
- Pixel value aggregation method: no aggregation.
- Expression:  Use expression. An 'Edit Expression...' button is next to it.
- Note: The expression might not be applicable to all products.  Use expression as filter,  Export expression result.
- Sub-scenes:  Enable export. Border size: 0.
- Google Earth export:  Export output coordinates to Google Earth (KMZ).
- Match with original input:  Include original input.

Buttons at the bottom of the dialog are 'Extract', 'Close', and 'Help'. The background shows a satellite image of a coastal area with a 'Product Explorer' on the left and a 'Navigation' map at the bottom left.

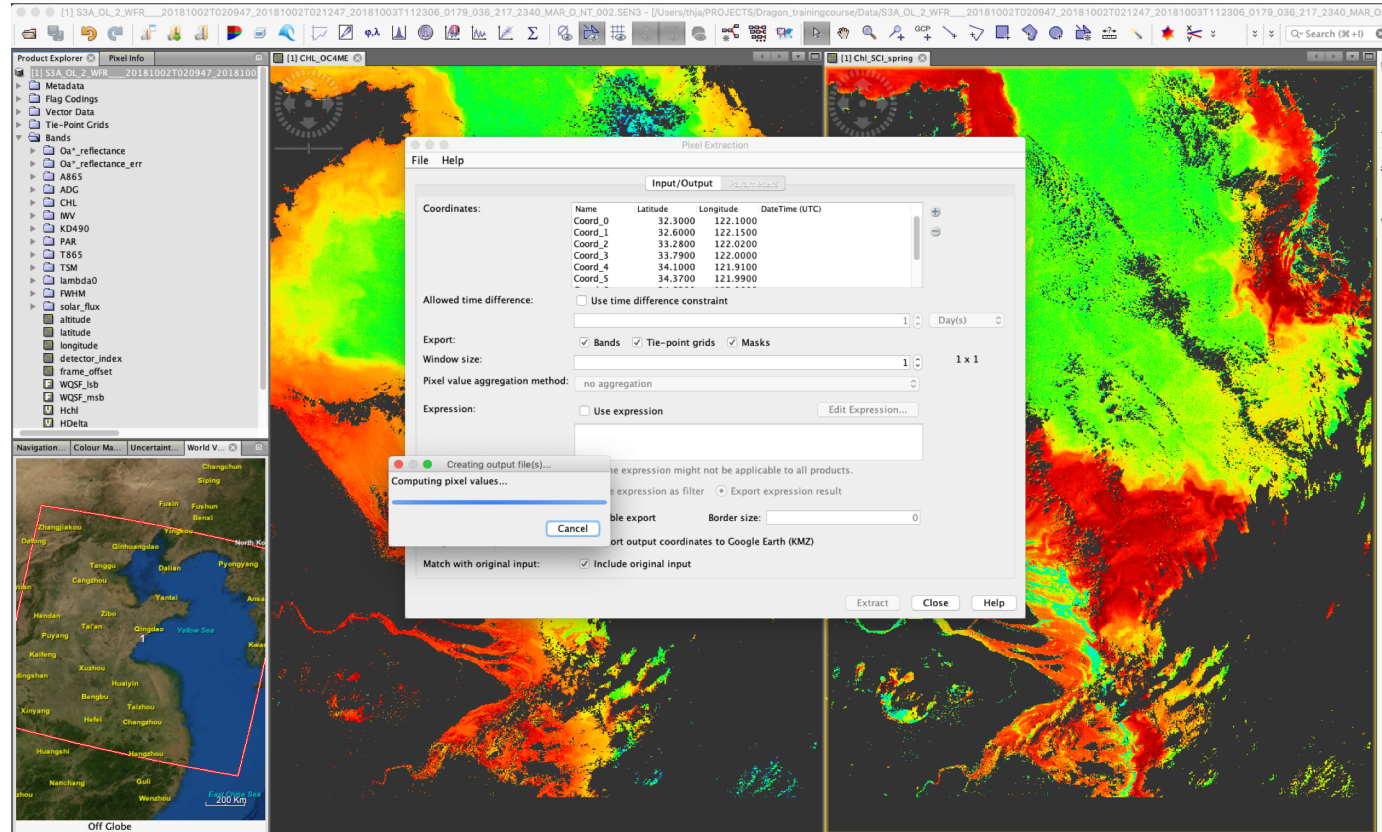
# Exercise 5: Extracting pixel data



Click extract and you should soon see the 'computing' and 'run successfully' messages.

Now if you go to the output location you should find 2 new files.

The matchup data will be in a file called pixEx\_OL\_2\_WFR\_m easurements.txt



# Exercise 5: Extracting pixel data



The output matchup file contains information from all the variables and you could now load it into your personally preferred data processing program to analyse the matchups.

The first 6 lines of the output file contain information on the extraction.

```
# SNAP pixel extraction export table
#
# Window size: 1
# Created on: 2018-11-05 13:14:08

# Wavelength:
42.5 490.0 490.0 510.0 510.0 560.0 560.0 620.0 620.0 665.0 665.0 673.75 673.75 681.25 681.25 708.75 708.75 753.75 753.75 778.75 778.75 86
5.0 865.0 885.0 885.0 1020.0 1020.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```



# Exercise 5: Extracting pixel data



Then there is a header row with all the column names.

NOTE this includes the variables we have created during the exercise.

Name	Latitude	Longitude	Date(yy-yy-MM-dd)	chl	geometry	ProdID	CoordID	Name	Latitude	Longitude	PixelX	PixelY	Date(yy-yy-MM-dd)																																																																																																																																																																				
rr	0a04_reflectance	0a04_reflectance_err	0a01_reflectance	0a01_reflectance_err	0a02_reflectance	0a02_reflectance_err	0a03_reflectance	0a03_reflectance_err	0a06_reflectance	0a06_reflectance_err	0a07_reflectance	0a07_reflectance_err	0a08_reflectance	0a08_reflectance_err	0a09_reflectance	0a09_reflectance_err	0a10_reflectance	0a10_reflectance_err	0a11_reflectance	0a11_reflectance_err	0a12_reflectance	0a12_reflectance_err	0a16_reflectance	0a16_reflectance_err	0a17_reflectance	0a17_reflectance_err	0a18_reflectance	0a18_reflectance_err	0a21_reflectance	0a21_reflectance_err	chl_nn	chl_nn_err	chl_0c4me	chl_0c																																																																																																																																															
4ME_err	altitude	latitude	longitude	detector_index	FWHM_band_1	FWHM_band_2	FWHM_band_3	FWHM_band_4	FWHM_band_5	FWHM_band_6	FWHM_band_7	FWHM_band_8	FWHM_band_9	FWHM_band_10	FWHM_band_11	FWHM_band_12	FWHM_band_13	FWHM_band_14	FWHM_band_15	FWHM_band_16	FWHM_band_17	FWHM_band_18	FWHM_band_19	FWHM_band_20	FWHM_band_21	frame_offset	lambda0_band_1	lambda0_band_2	lambda0_band_3	lambda0_band_4	lambda0_band_5	lambda0_band_6	lambda0_band_7	lambda0_band_8	lambda0_band_9	lambda0_band_10	lambda0_band_11	lambda0_band_12	lambda0_band_13	lambda0_band_14	lambda0_band_15	lambda0_band_16	lambda0_band_17	lambda0_band_18	lambda0_band_19	lambda0_band_20	lambda0_band_21	solar_flux_band_1	solar_flux_band_2	solar_flux_band_3	solar_flux_band_4	solar_flux_band_5	solar_flux_band_6	solar_flux_band_7	solar_flux_band_8	solar_flux_band_9	solar_flux_band_10	solar_flux_band_11	solar_flux_band_12	solar_flux_band_13	solar_flux_band_14	solar_flux_band_15	solar_flux_band_16	solar_flux_band_17	solar_flux_band_18	solar_flux_band_19	solar_flux_band_20	solar_flux_band_21	ADG443_NN	ADG443_NN_err	IWA	IWA_err	PAR	PAR_err	KD490_M07	KD490_M07_err	TSM_NN	TSM_NN_err	A865	A865_err	T865	T865_err	WQSF_lsb	WQSF_msb	Hchl	HDelta	SCI	chl_SCI_spring	TP_latitude	TP_longitude	OAA	OZA	SAA	SZA	atmospheric_temperature_profile_pressure_level_1	atmospheric_temperature_profile_pressure_level_2	atmospheric_temperature_profile_pressure_level_3	atmospheric_temperature_profile_pressure_level_4	atmospheric_temperature_profile_pressure_level_5	atmospheric_temperature_profile_pressure_level_6	atmospheric_temperature_profile_pressure_level_7	atmospheric_temperature_profile_pressure_level_8	atmospheric_temperature_profile_pressure_level_9	atmospheric_temperature_profile_pressure_level_10	atmospheric_temperature_profile_pressure_level_11	atmospheric_temperature_profile_pressure_level_12	atmospheric_temperature_profile_pressure_level_13	atmospheric_temperature_profile_pressure_level_14	atmospheric_temperature_profile_pressure_level_15	atmospheric_temperature_profile_pressure_level_16	atmospheric_temperature_profile_pressure_level_17	atmospheric_temperature_profile_pressure_level_18	atmospheric_temperature_profile_pressure_level_19	atmospheric_temperature_profile_pressure_level_20	atmospheric_temperature_profile_pressure_level_21	atmospheric_temperature_profile_pressure_level_22	atmospheric_temperature_profile_pressure_level_23	atmospheric_temperature_profile_pressure_level_24	atmospheric_temperature_profile_pressure_level_25	horizontal_wind_vector_1	horizontal_wind_vector_2	humidity	sea_level_pressure	total_columnar_water_vapour	total_ozone	WQSF_lsb_INVALID	WQSF_lsb_LAND	WQSF_lsb_WATER	WQSF_lsb_CLOUD	WQSF_lsb_CLOUD_MARGIN	WQSF_lsb_SNOW_ICE	WQSF_lsb_INLAND_WATER	WQSF_lsb_TIDAL	WQSF_lsb_COSMETIC	WQSF_lsb_SUSPECT	WQSF_lsb_HISOLZEN	WQSF_lsb_SATURATED	WQSF_lsb_MEGLINT	WQSF_lsb_HIGHGLINT	WQSF_lsb_WHITECAPS	WQSF_lsb_ADJAC	WQSF_lsb_WV_FAIL	WQSF_lsb_PAR_FAIL	WQSF_lsb_AC_FAIL	WQSF_lsb_OC4ME_FAIL	WQSF_lsb_OCNN_FAIL	WQSF_lsb_KDM_FAIL	WQSF_lsb_BPAC_ON	WQSF_lsb_WHITE_SCATT	WQSF_lsb_LOWRW	WQSF_lsb_HIG	HRW	WQSF_msb_ANNOT_ANGSTROM	WQSF_msb_ANNOT_AERO_B	WQSF_msb_ANNOT_ABSO_D	WQSF_msb_ANNOT_ACLIM	WQSF_msb_ANNOT_ABSOA	WQSF_msb_ANNOT_MIXR1	WQSF_msb_ANNOT_DROUT	WQSF_msb_ANNOT_TAU06	WQSF_msb_RWNEG_01	WQSF_msb_RWNEG_02	WQSF_msb_RWNEG_03	WQSF_msb_RWNEG_04	WQSF_msb_RWNEG_05	WQSF_msb_RWNEG_06	WQSF_msb_RWNEG_07	WQSF_msb_RWNEG_08	WQSF_msb_RWNEG_09	WQSF_msb_RWNEG_010	WQSF_msb_RWNEG_011	WQSF_msb_RWNEG_012	WQSF_msb_RWNEG_016	WQSF_msb_RWNEG_017	WQSF_msb_RWNEG_018	WQSF_msb_RWNEG_021	geometry

GUIs and portals give a great way to quickly view and interrogate data but if you want to do any significant bulk processing of data you will probably end up using some sort of dedicated coding approach. A number of languages can be used for processing of remote sensing data including:

- C (and its derivatives)
- python
- IDL
- R
- java

Each language has its own benefits and drawbacks such as memory usage, ease of use, speed of processing, compilation requirements etc, etc.

In order to choose the best language for the task it is worth framing and understanding your process before you begin coding.



# Coding and batch processing

When installing SNAP there is the option to install a snap specific python module (snappy) that allows much of the functionality shown earlier in this practical (and more) to be scripted.

Here is a script example for creating the chl product we produced in exercise 4.

You will find this script in the Practical Directory.

Please look through the script and check that you understand how its operation mirrors exercise 4.

```
Dragon_OLCI_OC_chl_shen.py
...
1 # coding: utf-8
2 # The aim of this code is to demonstrate how one might apply an ocean colour algorithm to level 2 reflectances in
  order to estimate chlorophyll-a concentration.
3
4 # Import a number of required modules and functions to aid in the processing
5 import shutil, os, sys
6 import argparse
7 import numpy as np
8 import matplotlib.pyplot as plt
9 # snappy modules
10 import snappy
11 from snappy import Product, ProductData, ProductIO, ProductUtils, ProgressMonitor
12
13
14 # No we define the input data from level 2 (the example used here shows eddy features in the chl field but you could
  change this to any other level 2 data file set for a region of your own interest).
15 # We copy the file to the local /tmp space to avoid overwriting the original data.
16 #example area of the Yellow Sea
17 indir='./DATA/S3A_OL_2_WFR_____20181002T020947_20181002T021247_20181003T112306_0179_036_217_2340_MAR_0_NT_002.SEN3'
18 COPY_DIR = indir.replace('.SEN3', '_copy.SEN3')
19 COPY_DIR= '/tmp/'+os.path.basename(COPY_DIR)
20 if os.path.isdir(COPY_DIR):
21     print('COPY_DIR+' already exists')
22 else:
23     shutil.copytree(indir, COPY_DIR)
24
25 # Here we use the xml file to load all the information about the data so that we can process it with snap but we do
  not need to load the variables into python memory (hence the commented out section to show that it is not actually
  required to get the bands if you use the expression correctly).
26 SOURCE_PRODUCT = ProductIO.readProduct(COPY_DIR+'/xfdmanifest.xml')
27
28 #Shen 2010 equations 2,4 and 5
29 Hchl_expressions='(0.74+0a10_reflectance + 0.26+0a07_reflectance - 0a08_reflectance)'
30 newband1 = SOURCE_PRODUCT.addBand('Hchl', Hchl_expression, snappy.ProductData.TYPE_FLOAT32)
31 newband1.setDescription("Hchl from Shen 2010")
32
33 HDelta_expressions='(0a07_reflectance-0.5*(0a06_reflectance+0a10_reflectance)'
34 newband2 = SOURCE_PRODUCT.addBand('HDelta', HDelta_expression, snappy.ProductData.TYPE_FLOAT32)
35 newband2.setDescription("HDelta from Shen 2010")
36
37 SCI_expression='Hchl-HDelta'
38 newband3 = SOURCE_PRODUCT.addBand('SCI', SCI_expression, snappy.ProductData.TYPE_FLOAT32)
39 newband3.setDescription("SCI index from shen 2010 MERIS chl algorithm")
40
41 # Now we convert SCI to Chl
42 Chl_expression='179378.0*pow(SCI,2) +92.934*SCI + 0.2736'
43 newband2 = SOURCE_PRODUCT.addBand('Chl_SCI_spring', Chl_expression)
44 newband2.setDescription("Trial implimentation of the shen 2010 MERIS chl algorithm")
45
46 # We will write the file out with the new band
47 product_writer = snappy.ProductIO.getProductWriter("NETCDF4-CF")
48 SOURCE_PRODUCT.setProductWriter(product_writer)
49 target_file='/tmp/test_netcdf.nc'
50 print('about to write')
51 product_writer.writeProductNodes(SOURCE_PRODUCT, target_file)
52 print('finished writing')
53 SOURCE_PRODUCT.closeIO()
```



# Coding and batch processing

You should be able to identify:

- The modules imported
- The loading of the data from netcdf
- The creation of new products
- The saving of the new file.

(look back to lecture notes if not clear)

If your machine has python and snappy available then you should be able to run the script from the command line with:  
*python Dragon\_OLCI\_OC\_chl\_shen.py*

A good starting point for snappy is:

<https://senbox.atlassian.net/wiki/spaces/SNAP/pages/19300362/How+to+use+the+SNAP+API+from+Python>

```
Dragon_OLCI_OC_chl_shen.py
...
1 # coding: utf-8
2 # The aim of this code is to demonstrate how one might apply an ocean colour algorithm to level 2 reflectances in
  order to estimate chlorophyll-a concentration.
3
4 # Import a number of required modules and functions to aid in the processing
5 import shutil, os, sys
6 import argparse
7 import numpy as np
8 import matplotlib.pyplot as plt
9 # snappy modules
10 import snappy
11 from snappy import Product, ProductData, ProductIO, ProductUtils, ProgressMonitor
12
13
14 # No we define the input data from level 2 (the example used here shows eddy features in the chl field but you could
  change this to any other level 2 data file set for a region of your own interest).
15 # We copy the file to the local /tmp space to avoid overwriting the original data.
16 #example area of the Yellow Sea
17 indir='./DATA/S3A_OL_2_WFR_____20181002T020947_20181002T021247_20181003T112306_0179_036_217_2340_MAR_0_NT_002.SEN3'
18 COPY_DIR = indir.replace('.SEN3', '_copy.SEN3')
19 COPY_DIR= '/tmp/'+os.path.basename(COPY_DIR)
20 if os.path.isdir(COPY_DIR):
21     print(COPY_DIR+' already exists')
22 else:
23     shutil.copytree(indir, COPY_DIR)
24
25 # Here we use the xml file to load all the information about the data so that we can process it with snap but we do
  not need to load the variables into python memory (hence the commented out section to show that it is not actually
  required to get the bands if you use the expression correctly).
26 SOURCE_PRODUCT = ProductIO.readProduct(COPY_DIR+'/xfdmanifest.xml')
27
28 #Shen 2010 equations 2,4 and 5
29 Hchl_expressions='(0.74*0a10_reflectance + 0.26*0a07_reflectance - 0a08_reflectance)'
30 newband1 = SOURCE_PRODUCT.addBand('Hchl', Hchl_expression, snappy.ProductData.TYPE_FLOAT32)
31 newband1.setDescription("Hchl from Shen 2010")
32
33 Hdelta_expressions='(0a07_reflectance-0.5*(0a06_reflectance+0a10_reflectance)'
34 newband2 = SOURCE_PRODUCT.addBand('Hdelta', Hdelta_expression, snappy.ProductData.TYPE_FLOAT32)
35 newband2.setDescription("Hdelta from Shen 2010")
36
37 SCI_expression='Hchl-Hdelta'
38 newband3 = SOURCE_PRODUCT.addBand('SCI', SCI_expression, snappy.ProductData.TYPE_FLOAT32)
39 newband3.setDescription("SCI index from shen 2010 MERIS chl algorithm")
40
41 # Now we convert SCI to Chl
42 Chl_expression='179378.0*pow(SCI,2) +92.934*SCI + 0.2736'
43 newband2 = SOURCE_PRODUCT.addBand('Chl_SCI_spring', Chl_expression)
44 newband2.setDescription("Trial implimentation of the shen 2010 MERIS chl algorithm")
45
46 # We will write the file out with the new band
47 product_writer = snappy.ProductIO.getProductWriter("NETCDF4-CF")
48 SOURCE_PRODUCT.setProductWriter(product_writer)
49 target_file='/tmp/test_netcdf.nc'
50 print('about to write')
51 product_writer.writeProductNodes(SOURCE_PRODUCT, target_file)
52 print('finished writing')
53 SOURCE_PRODUCT.closeIO()
```



# Coding and batch processing

No snappy? Don't worry! If you have just python then the script `Dragon_OLCI_OC_chl_shen_no_snappy.py` should perform the same functionality.

Again, inspect the code in the file to see key sections.

Run with the following on the command line:

```
python Dragon_OLCI_OC_chl_shen_no_snappy.py
```

You can then compare load `/tmp/test_output.nc` into SNAP and compare it to the product created in exercise 4.

```
Dragon_OLCI_OC_chl_shen_no_snappy.py
1 # coding: utf-8
2 # The aim of this code is to demonstrate how one might apply an ocean colour algorithm to level 2 reflectances in
  order to estimate chlorophyll-a concentration.
3
4 # Import a number of required modules and functions to aid in the processing
5 import shutil, os, sys
6 import argparse
7 import numpy as np
8 import matplotlib.pyplot as plt
9 import netCDF4 as nc
10
11
12
13 # No we define the input data from level 2 (the example used here shows eddy features in the chl field but you could
  change this to any other level 2 data file set for a region of your own interest).
14 # We copy the file to the local /tmp space to avoid overwriting the original data.
15 #example area of the Yellow Sea
16 indir='./DATA/S3A_OL_2_WFR_____20181002T020947_20181002T021247_20181003T112306_0179_036_217_2340_MAR_0T_NT_002.SEN3'
17 COPY_DIR = indir.replace('.SEN3', '_copy.SEN3')
18 COPY_DIR= '/tmp'+os.path.basename(COPY_DIR)
19 if os.path.isdir(COPY_DIR):
20     print(COPY_DIR+' already exists')
21 else:
22     shutil.copytree(indir, COPY_DIR)
23
24 # Here we load the required netcdfs
25 Oa06_reflectance=nc.Dataset(COPY_DIR+'/Oa06_reflectance.nc', 'r')
26 Oa06_reflectance_values=nc.Dataset(COPY_DIR+'/Oa06_reflectance.nc', 'r').variables['Oa06_reflectance']
27 Oa07_reflectance_values=nc.Dataset(COPY_DIR+'/Oa07_reflectance.nc', 'r').variables['Oa07_reflectance']
28 Oa08_reflectance_values=nc.Dataset(COPY_DIR+'/Oa08_reflectance.nc', 'r').variables['Oa08_reflectance']
29 Oa10_reflectance_values=nc.Dataset(COPY_DIR+'/Oa10_reflectance.nc', 'r').variables['Oa10_reflectance']
30
31 # #Shen 2010 equations 2,4 and 5
32 Hchl=(0.74*Oa10_reflectance_values[:] + 0.26*Oa07_reflectance_values[:] - Oa08_reflectance_values[:])
33
34 HDelta=Oa07_reflectance_values[:] - 0.5*(Oa06_reflectance_values[:] + Oa10_reflectance_values[:])
35
36 SCI=Hchl-HDelta
37
38 # # Now we convert SCI to Chl
39 CHL_SCI_Spring=179378.0*pow(SCI,2) +92.934*SCI + 0.2736
40 # # We will write the file out the new band to a file
41 # print('about to write')
42
43 dims=[x for x in Oa06_reflectance.dimensions]
44 ds = nc.Dataset('/tmp/test_output.nc','w', format="NETCDF4")
45 for dimension in Oa06_reflectance.dimensions:
46     ds.createDimension(dimension)
47 ds.createVariable('Chl_SCI_spring', 'f8',dimensions=dims, fill_value=CHL_SCI_Spring.get_fill_value())
48 ds.variables['Chl_SCI_spring'][:]=CHL_SCI_Spring[:]
49 ds.close()
50 # print('finished writing')
51
```



## ***Essential OLCI (Copernicus Marine Data Service) links:***

- CODA for download of data from last 365 days: <https://coda.eumetsat.int>
  - CODAREP (Reprocessed historical data): <https://codarep.eumetsat.int>
  - CODA user manual: <https://coda.eumetsat.int/manual/CODA-user-manual.pdf>
  - Data centre (for older data):  
<https://www.eumetsat.int/website/home/Data/DataDelivery/EUMETSATDataCentre/index.html>
  - Batch scripting for CODA download: <https://coda.eumetsat.int/manual/CODA-user-manual.pdf> (page 34)
  - Video tutorial for CODA downloads:  
[https://www.youtube.com/watch?v=l4oeRYj6\\_5U&list=PLOQg9n6Apif2Qw\\_gLhwzhJb3XUoAiUkoq&index=2](https://www.youtube.com/watch?v=l4oeRYj6_5U&list=PLOQg9n6Apif2Qw_gLhwzhJb3XUoAiUkoq&index=2)
- Video for OLCI data download and visualisation in SNAP:  
[https://www.youtube.com/watch?v=V3NAuafvIFM&index=3&list=PLOQg9n6Apif2Qw\\_gLhwzhJb3XUoAiUkoq](https://www.youtube.com/watch?v=V3NAuafvIFM&index=3&list=PLOQg9n6Apif2Qw_gLhwzhJb3XUoAiUkoq)

## ***Useful links for other types of ocean satellite data you may want to use:***

-CMEMS (Level 3 and 4, merged, model products): <http://marine.copernicus.eu/>

-NASA ocean colour (for MODIS and VIIRS, and other historical sensors, plus some in situ data): <https://oceancolor.gsfc.nasa.gov/>

-Ocean colour CCI (Global merged sensor product for climate studies):

<http://www.oceancolour.org/>

## ***Useful general Python links:***

-Beginners (general) python tutorials:

<https://wiki.python.org/moin/BeginnersGuide/Programmers>

-Working with marine data: <https://oceanpython.org/>

For those who work with/wish to work with more GIS based applications, consider

-GDAL: <https://pcjericks.github.io/py-gdalogr-cookbook/>

-Installing Jupyter notebooks (comes with Anaconda)

<http://jupyter.org/install.html>

-Installing netCDF4: type 'conda install -c anaconda netcdf4' in to the command line (if you have used anaconda install)