# Monitoring and assessment of water quality by ocean color remote sensing

JAXA/EORC

Taiga Nakayama

# Today's Content

- How to use G-portal
  - Registration
  - Search products
  - Download products

- SGLI-L2-SST display and mapping by python
  - Retrieve physical quantity
  - map projection

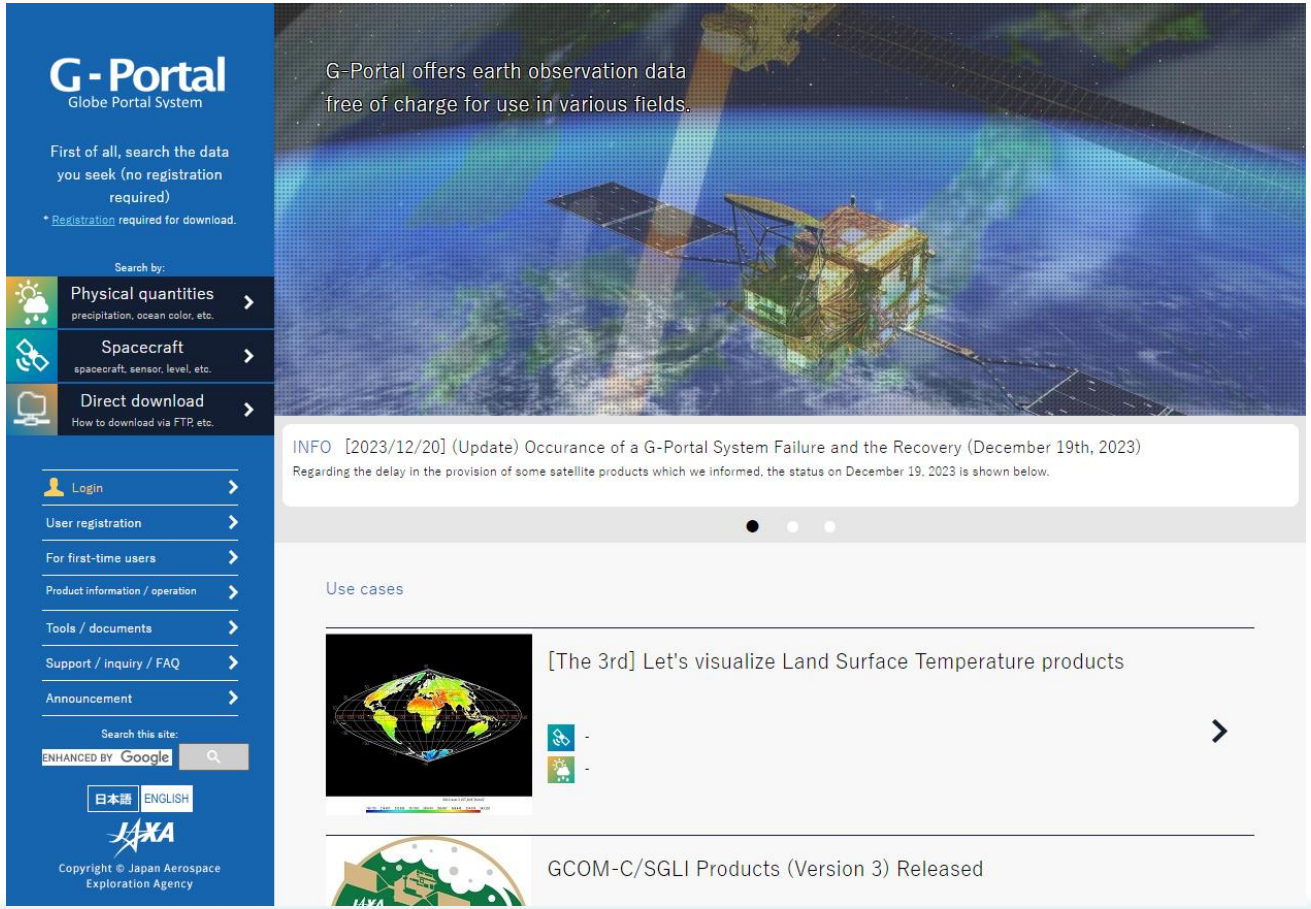
# What's G-portal?

Globe Portal System (G-Portal) is online dissemination service of valuable products acquired from sensors on Earth Observation Satellites of Japan Aerospace Exploration Agency (hereafter, JAXA).

**https://gportal.jaxa.jp/gpr/**

Product Search

- Search by physical quantities
- Search by spacecrafts / sensors
- Search using saved conditions(only available to registered users)



# User registration

# G-portal user type

User types for G-Portal is outlined in Table. You are recommended to complete user registration first in order to acquire products without restrictions

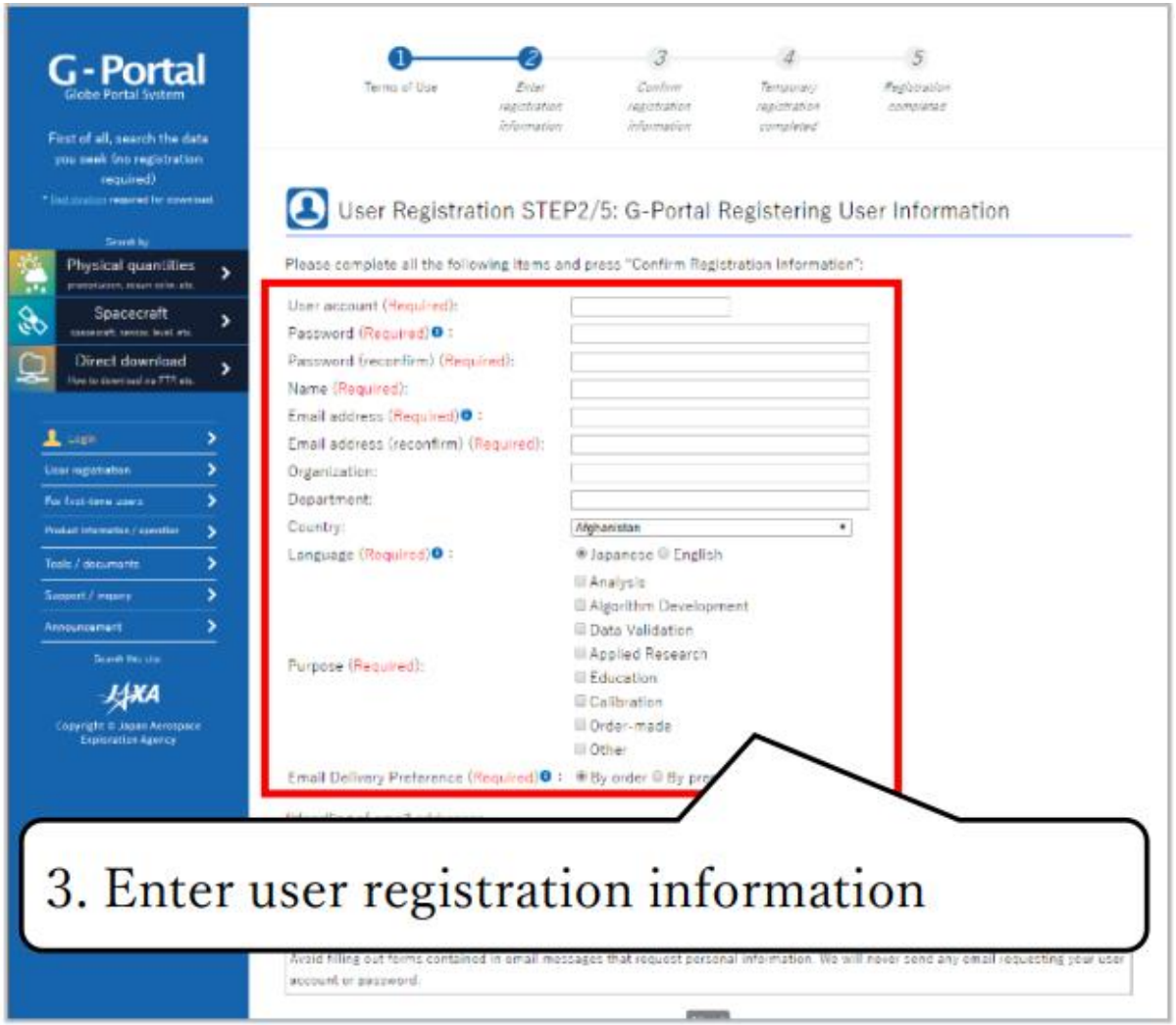
| User Type | Definition | Services Available |
|---|---|---|
| Guest User | Users who have not completed user registration | Search and browse data online.<br>Note that guest users are unable to order or acquire products. |
| Registered User | Users who have completed user registration | Search, browse, produce, process and acquire standard products online.<br>Acquire standard products and near real-time products directly from the FTP server. |
| Specified User | Register users, collaborator specially permitted by JAXA | Search, browse, produce, process and acquire standard products online.<br>In addition to the products that registered users can download, acquire special products being available to those who are JAXA-approved directly from the SFTP server. |

# User registration ①

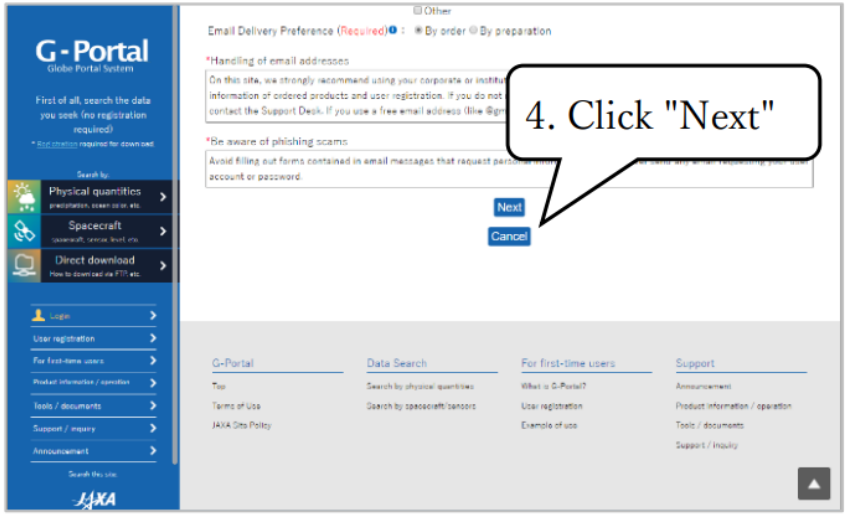

1) Click “User registration” from the menu


2) To register a user, you must agree to the terms of use. Read through the terms and click the “I Agree – Continue” button.
The page will move to the “User Registration window”.

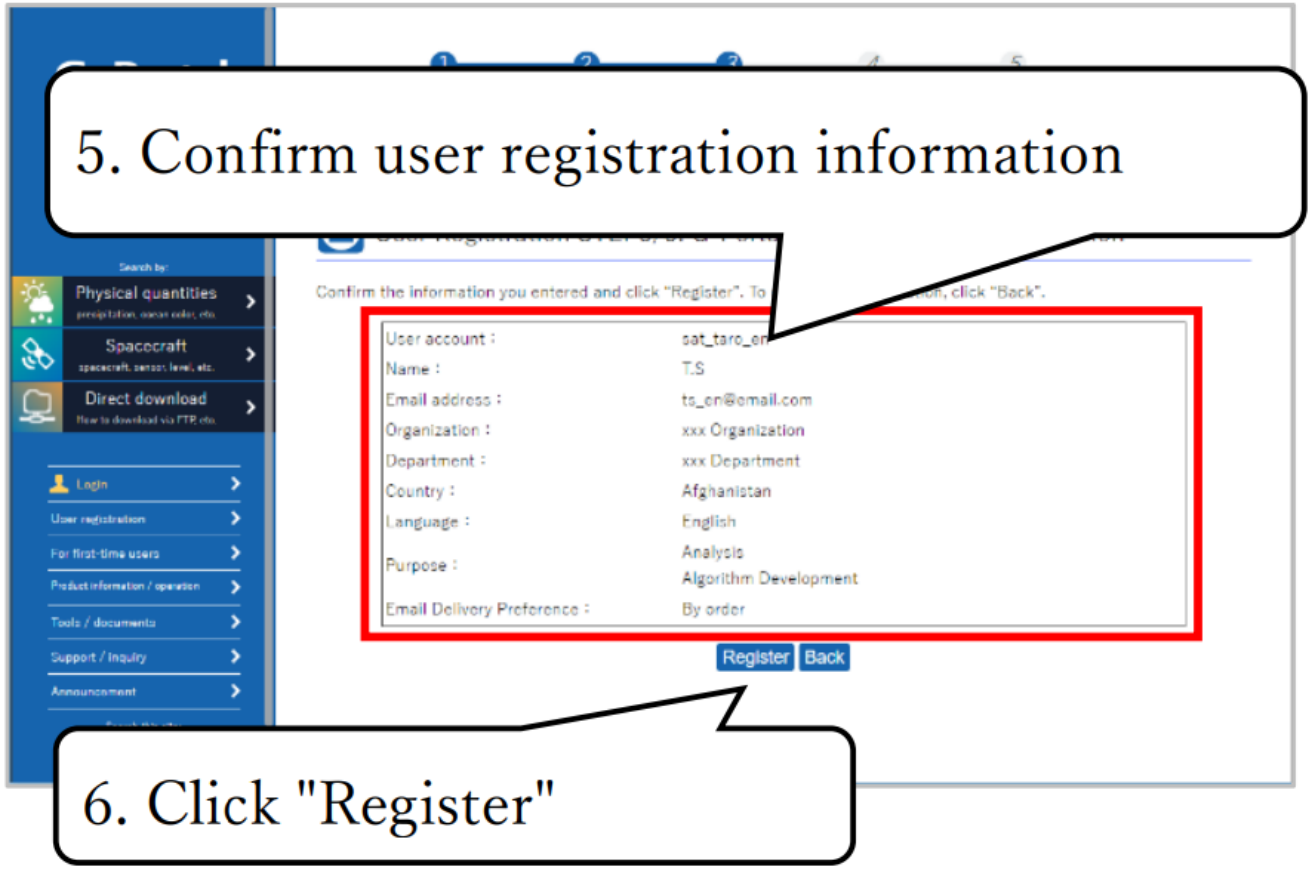
# User registration ②




3) Enter all user information to be registered (user account, password, name, email address, organization, department, country, language, purpose of use, email delivery preference).



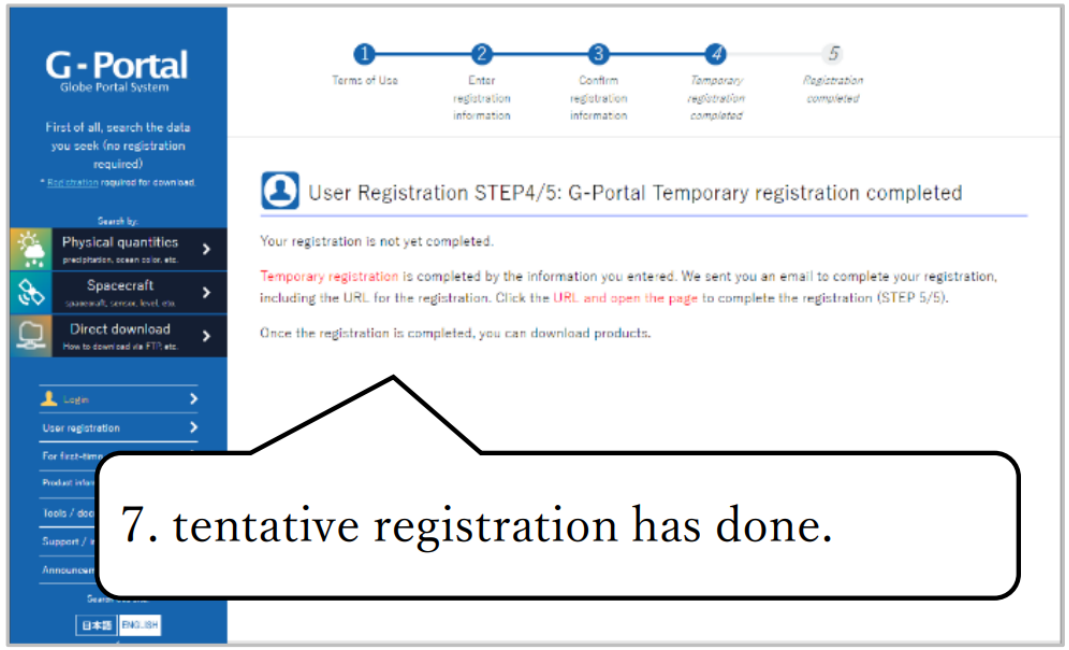
# User registration ③

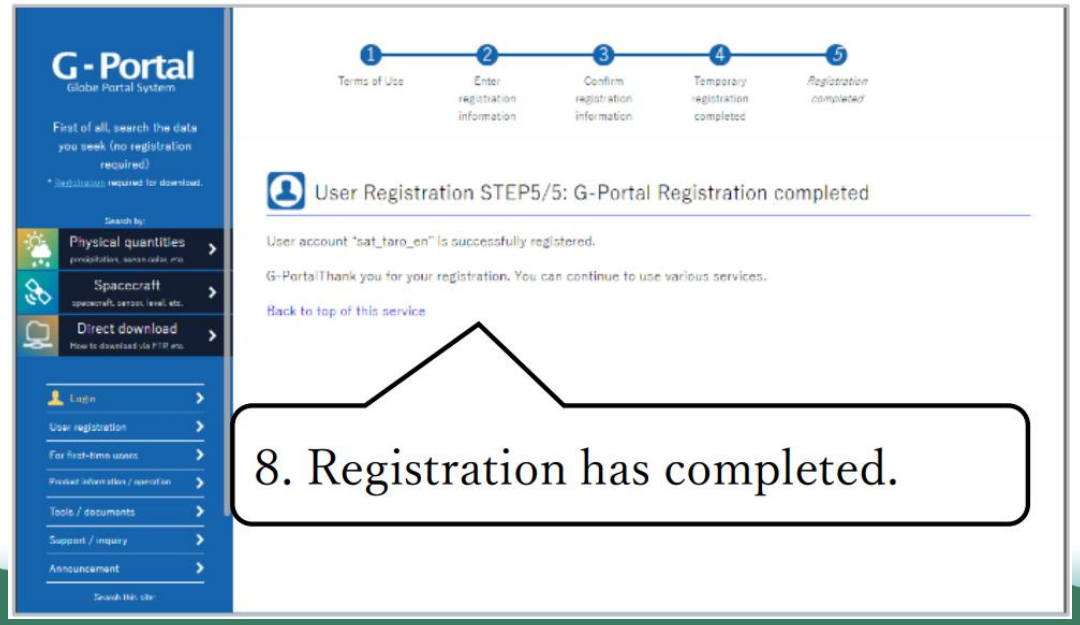



5) The entered user information will be displayed, so check if there is any mistake in the contents. To correct, please click the "return" button. You will return to the screen for entering user information.

6) Click the "Register" button to perform the provisional registration procedure.

# User registration ④




7) The temporary registration procedure is completed. "Temporary registration notification mail" will be sent to the email address you entered. Registration is completed by accessing the URL described in the mail. At the time of provisional registration, you can not log in because user registration has not been completed yet.

8) Access the URL included in the “Temporary registration notification mail”. The final Registration window will be displayed. Your user account is sent to the email address you registered

# User registration ⑤

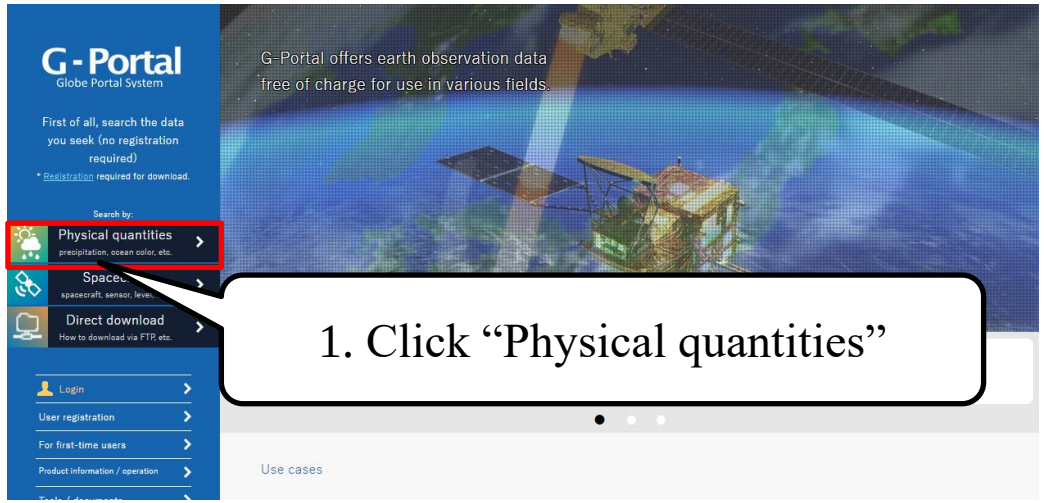

1) Click ”login” at menu on a left pane of the top window. Appear the window for the login.

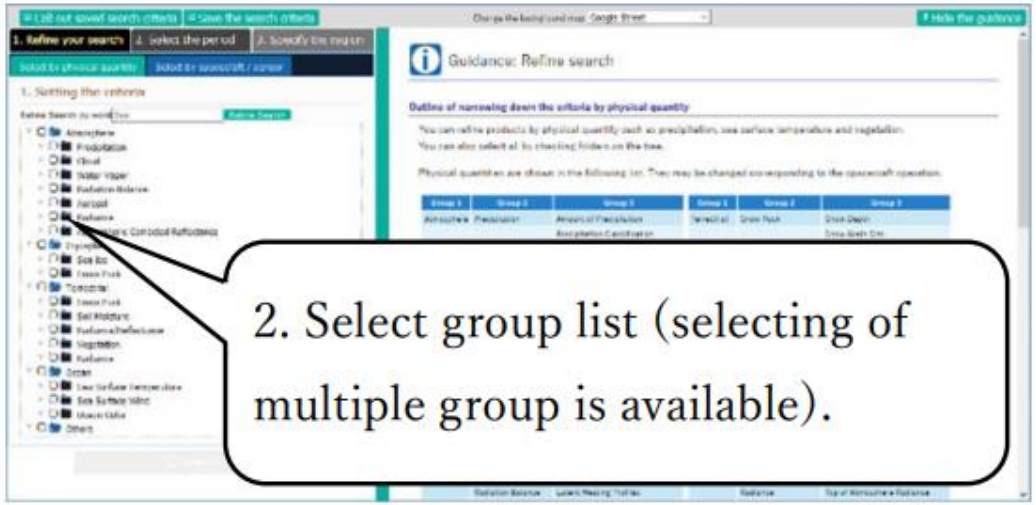

2) Input your user accounts and password as the login window is displayed. Click “login” button.

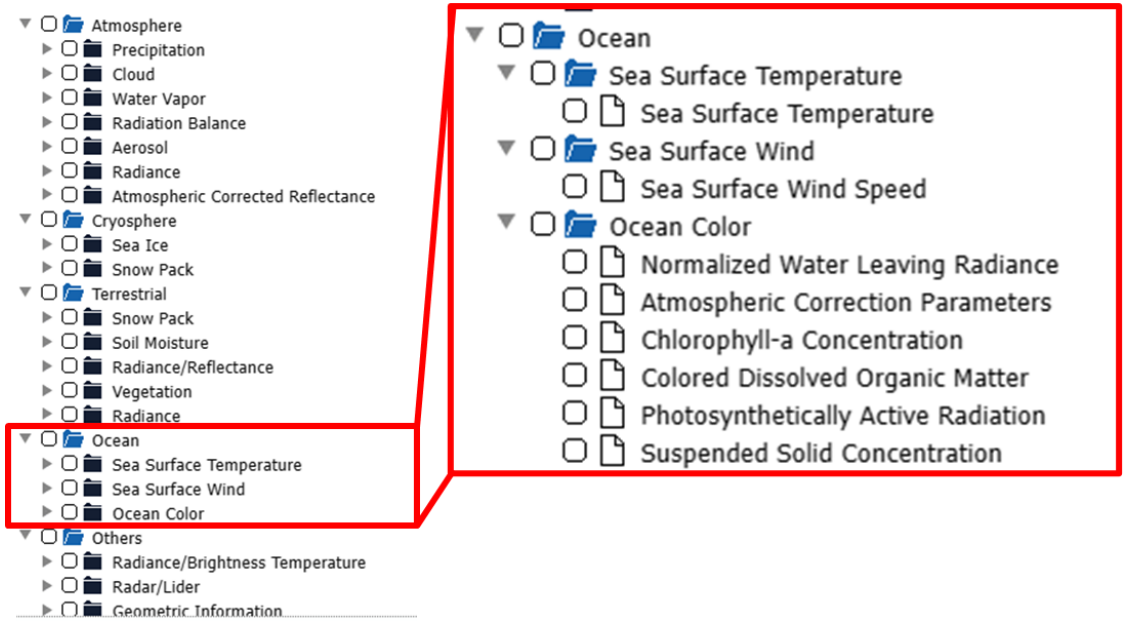
**Then, you can download satellite data from G-portal!**

# Search products

# Search by physical quantities



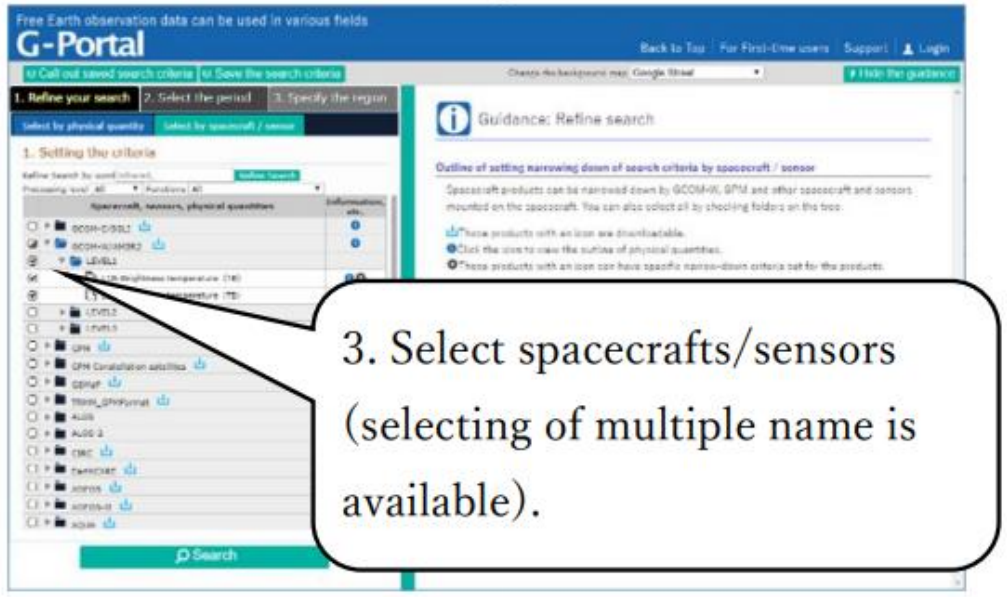



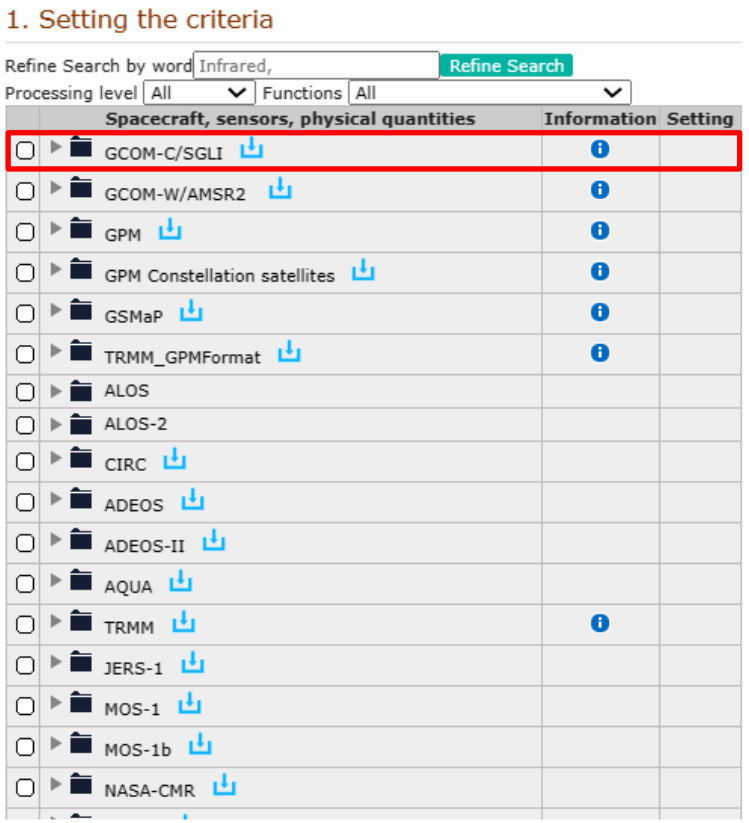

1) Click “search from physical quantities” left pane on the top menu and move to the search window shown the physical quantities tree.

2) Each category shows the group list to physical quantities to a tree format on the search window.

# Search by Spacecraft

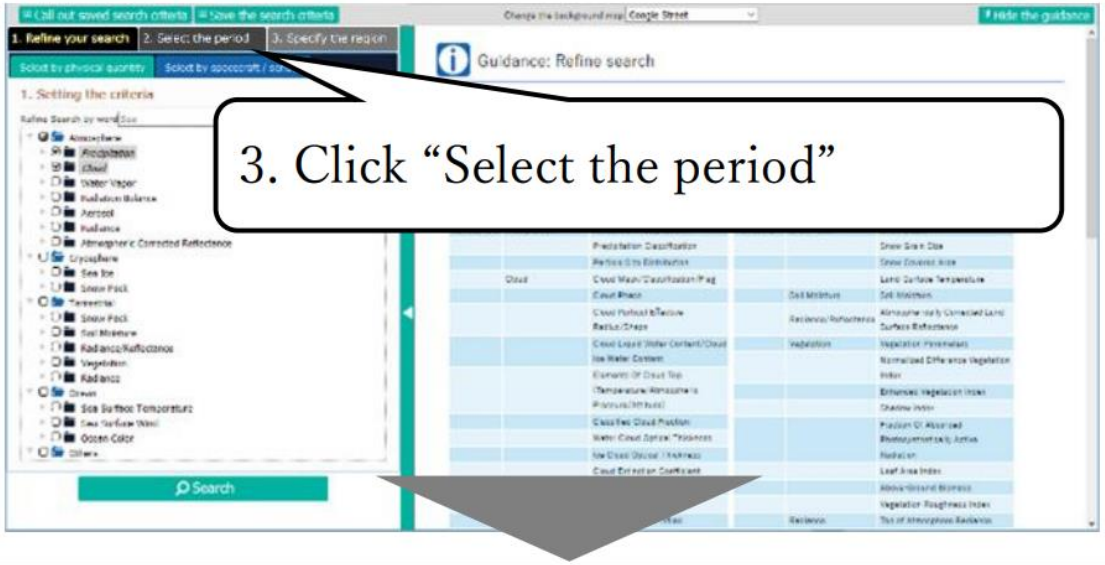




1) Click “Search by spacecraft sensors”on the top window menu and move to search window shown spacecraft seinsors tree.




3) Check a searching satellite sensor.
The detail search status show about the setting available product on the setting icon. Click the setting icon, The detail search condition input dialog is shown.

# Select Target Period




3) Click “2. Specify date ” on the top of the window with checking physical quantities.


4) The left pane displaying “data range” appears. The selections are “ Period“ and “Season“. There are three ways on “Data Range” as follow.

- Input by text
- Input from the calendar UI
- Input by bar-chart to observation period

# Select Target Area①



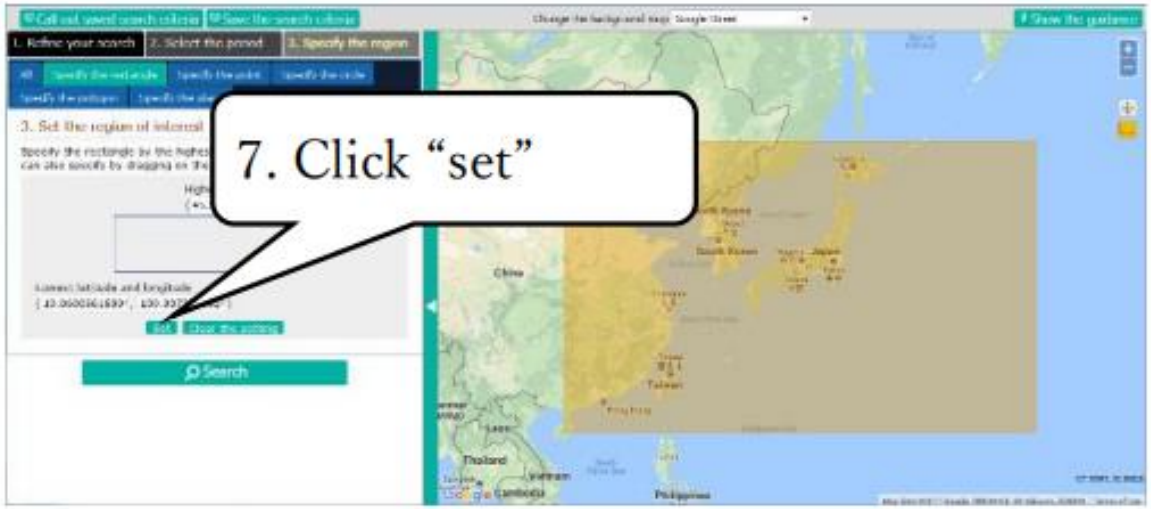
5) Click “3. Specify Area”on the top of a window.

6) Appear a window specifying search field. The selections of observation area are six applications.

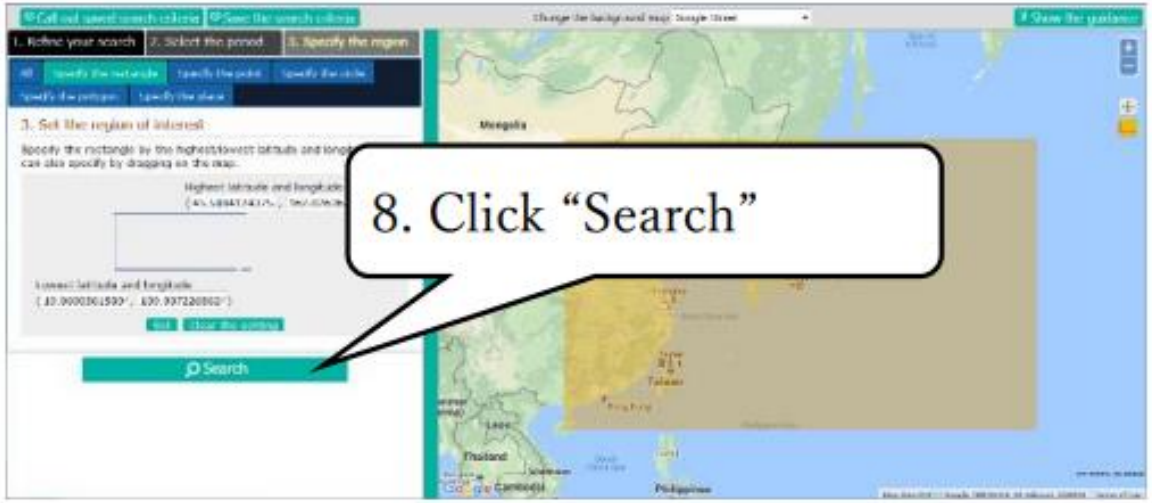
- specify a globe
- specify a rectangle
- specify a point
- specify a circle
- specify a polygon
- specify a place (name)

Choose a function from “text input” and “ drag and drop on the map”.

# Select Target Area②

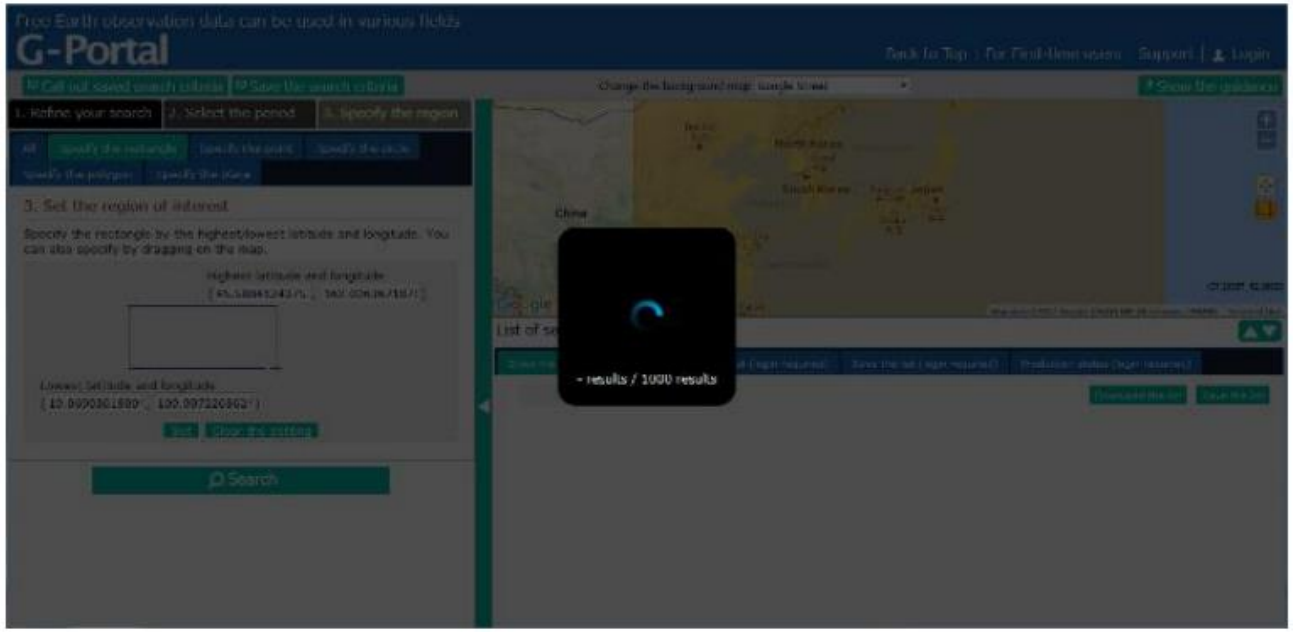



7) Click “Submit” when input the searching area by text.
(unnecessary on drug on the map)




8) Start search by clicking “search” button on the lower part window. “Search” button can click under setting a spacecraft sensor, physical quantities and observation data.

# Search loading…




9) Loading icon and searching status are displayed during searching. “Hit number”/”Visible number“

When the system is busy, the designated area is too small, or the number of applicable data is too large
→Searching cannot be performed properly.
→This problem may be solved by expanding the target area, shortening the period of time to be covered, etc.

# Download products

# Practice

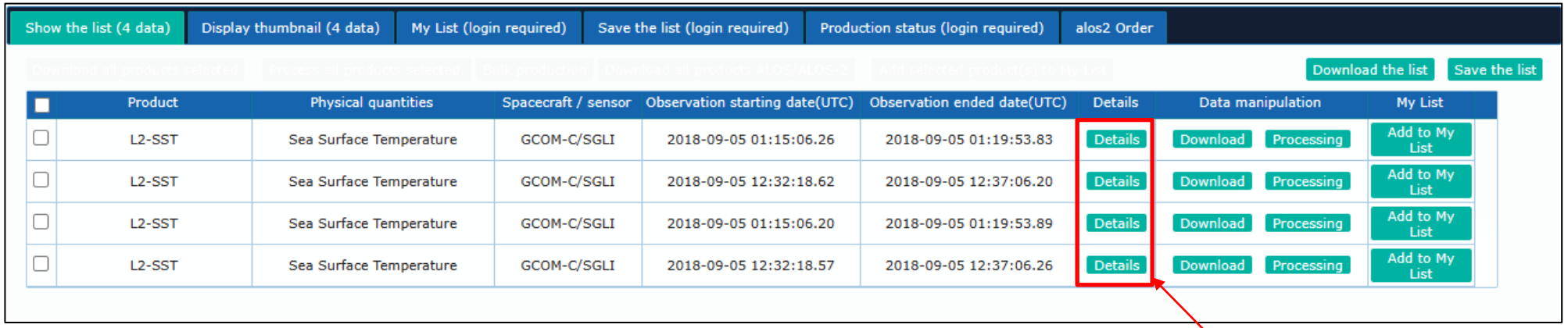
**Download SST images of GCOM-C/SGLI around Japan on September 5, 2018 !**

Sensor ID : GCOM-C/SGLI
Observation date : September 5, 2018
Target area : around Japan
Resolution : 250m
Orbit: Descending  (Daytime image)

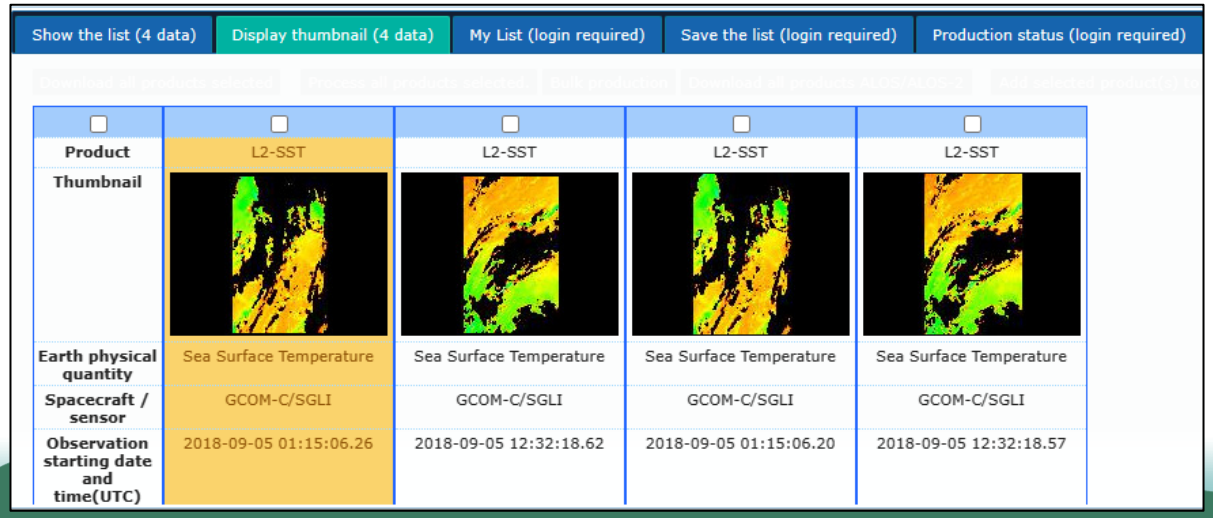
Filename = GC1SG1_20180905011 5H04610_L2SG_SSTDQ_3001.h5

# Practice

1) Click the “Show the list” tab on the search results window. The search results will be displayed as a list.

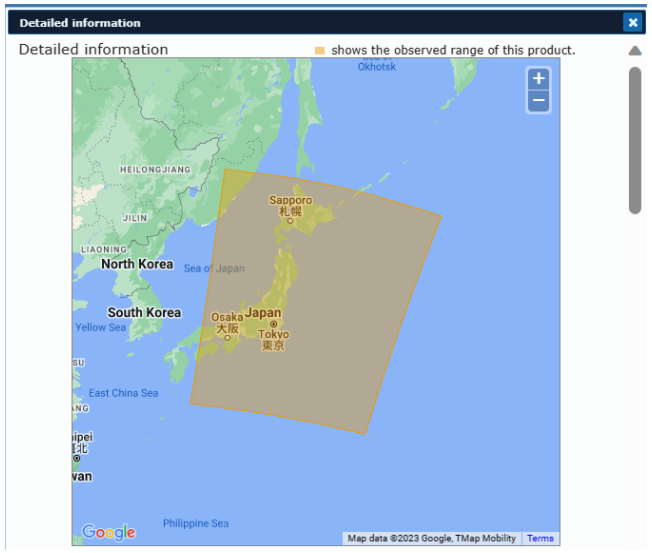



2) Click the “Display thumbnail” tab on the search results window. The search results will be displayed as thumbnails.




From this button, you can see detailed information.

# Check detailed information



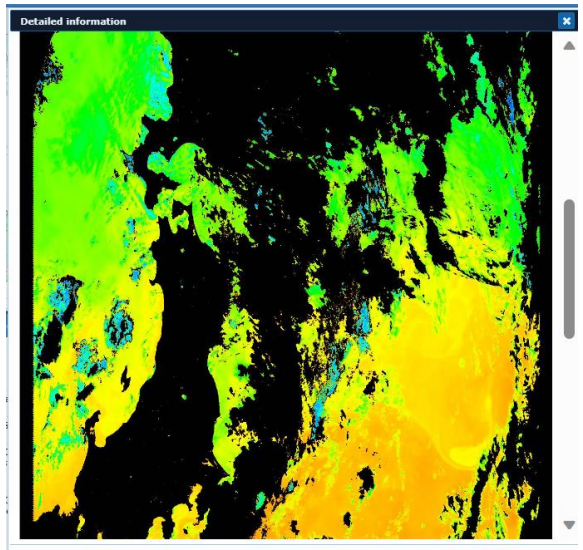




Image: Browse ! The display screen is switched by selecting.

| | |
|---|---|
| Granule ID | GC1SG1_201809050115H04610_L2SG_SSTDQ_3001 |
| Processing Date (UTC) | 2022-08-31 00:45:04.00 |
| Processing Level | L2 |
| Observation Starting Date (UTC) | 2018-09-05 01:15:06.20 |
| Observation Ended Date (UTC) | 2018-09-05 01:19:53.89 |
| Platform Short Name | GCOM-C |
| Sensor | SGLI |
| Sensor Operational Mode | NOMINAL |
| Orbit Number | 046 |
| Product File | https://gportal.jaxa.jp/download/standard/GCOM-C/GCOM-C.SGLI/L2.OCEAN.SST_/3/2018/09/05/GC1SG1_201809050115H04610_L2SG_SSTDQ_3001.h5 |
| Product size(MB) | 50 |
| Product version | 3 |
| Total Quality Code | Good |
| Cloud Coverage (%) | - |
| Compression | Compressed |
| Physical Quantity | Sea Surface Temperature |
| Product resolution | 250m |
| Scene number | 10 |
| Orbit Direction | Descending |

Show the list (4 data) | Display thumbnail (4 data) | My List (login required) | Save the list (login required) | Production status (login required) | alos2 Order

Download all products selected | Process all products selected | Bulk production | Download all products ALOS/ALOS-2 | Add selected product(s) to My List

Download the list | Save the list

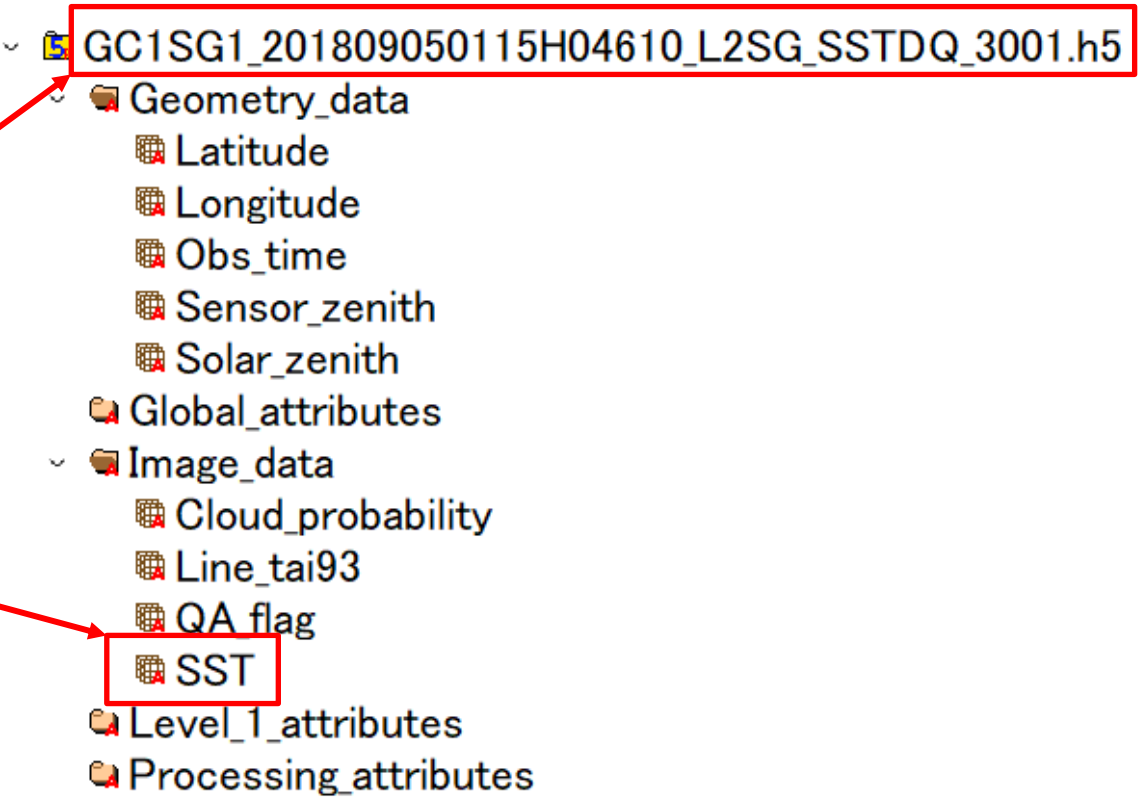
| | Product | Physical quantities | Spacecraft / sensor | Observation starting date(UTC) | Observation ended date(UTC) | Details | Data manipulation | | My List |
|---|---|---|---|---|---|---|---|---|---|
| ☐ | L2-SST | Sea Surface Temperature | GCOM-C/SGLI | 2018-09-05 01:15:06.26 | 2018-09-05 01:19:53.83 | Details | Download | Processing | Add to My List |
| ☐ | L2-SST | Sea Surface Temperature | GCOM-C/SGLI | 2018-09-05 12:32:18.62 | 2018-09-05 12:37:06.20 | Details | Download | Processing | Add to My List |
| ☐ | L2-SST | Sea Surface Temperature | GCOM-C/SGLI | 2018-09-05 01:15:06.20 | 2018-09-05 01:19:53.89 | Details | Download | Processing | Add to My List |
| ☐ | L2-SST | Sea Surface Temperature | GCOM-C/SGLI | 2018-09-05 12:32:18.57 | 2018-09-05 12:37:06.26 | Details | Download | Processing | Add to My List |

# SGLI-L2-SST display and mapping

# Data Structure and Reading

```
[ ]: #SGLI_L2_SST_mapping py to ipynb
import h5py
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import griddata
%matplotlib inline
FNAME = 'GC1SG1_201809050115H04610_L2SG_SSTDQ_3001.h5'
DNAME = '/Image_data/SST'
```

```
[ ]: file =  h5py.File(FNAME, 'r')
# Read SST data
Data0 = file[DNAME][:]

# Read attributes
Err_DN = file[DNAME].attrs['Error_DN']
Min_DN = file[DNAME].attrs['Minimum_valid_DN']
Max_DN = file[DNAME].attrs['Maximum_valid_DN']
Slope  = file[DNAME].attrs['Slope']
Offset = file[DNAME].attrs['Offset']
```




https://www.hdfgroup.org/downloads/hdfview/

# Data Structure and Reading

```
[ ]: #SGLI_L2_SST_mapping py to ipynb
import h5py
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import griddata
%matplotlib inline
FNAME = 'GC1SG1_201809050115H04610_L2SG_SSTDQ_3001.h5'
DNAME = '/Image_data/SST'
```

```
[ ]: file =  h5py.File(FNAME, 'r')
# Read SST data
Data0 = file[DNAME][:]

# Read attributes
Err_DN = file[DNAME].attrs['Error_DN']
Min_DN = file[DNAME].attrs['Minimum_valid_DN']
Max_DN = file[DNAME].attrs['Maximum_valid_DN']
Slope  = file[DNAME].attrs['Slope']
Offset = file[DNAME].attrs['Offset']
```

**Attribute of '/Image_data/SST'**

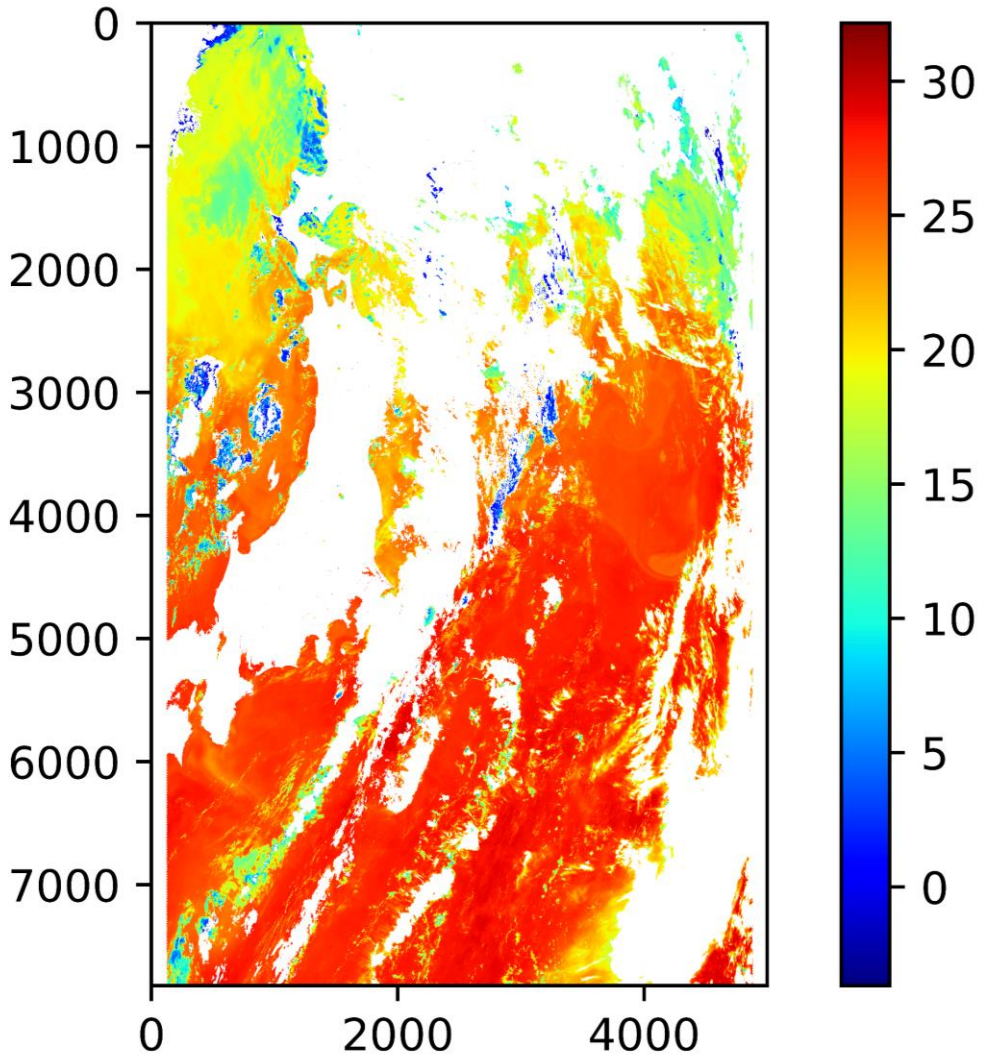
'Data_description'
'dim0'
'dim1'
'Error_DN'
'Land_DN'
'Cloud_error_DN'
'Retrieval_error_DN'
'Maximum_valid_DN'
'Minimum_valid_DN'
'Mask_for_statistics'
'Slope'
'Offset'
'Spatial_resolution'
'Spatial_resolution_unit'
'Unit'

Obtain parameters to convert from DN values to physical quantity (SST)

# Physical quantity conversion

```
[ ]: # Data processing
Data1 = Data0.astype(float)
Data1[Data0 == Err_DN] = np.nan
Data1[(Data0 <= Min_DN) | (Data0 >= Max_DN)] = np.nan
Data1 = Slope * Data1 + Offset

# Plotting
plt.figure()
plt.imshow(Data1, cmap='jet')
plt.colorbar()
plt.savefig("figure/default.png", format="png", dpi=2000)
```

Non-conforming data is replaced by missing values (NaN) and converted to floating point



# Apply QA flags (if necessary)

```
[ ]: # Read QA_flag
     QA_flag = file['/Image_data/QA_flag'][:]
     possibly_cloudy = np.bitwise_and(QA_flag, 2**12, dtype=np.uint16)
     acceptable = np.bitwise_and(QA_flag, 2**13, dtype=np.uint16)
     good = np.bitwise_and(QA_flag, 2**14, dtype=np.uint16)

     reliable = np.logical_or.reduce([good, acceptable, possibly_cloudy])

     # Apply reliability mask
     Data1[~reliable] = np.nan

     # Plotting with reliability mask
     plt.figure()
     plt.imshow(Data1, cmap='jet')
     plt.colorbar()
     plt.savefig("figure/applying_QAflag.png", format="png", dpi=2000)
     plt.show()
```

Extract and display only reliable data

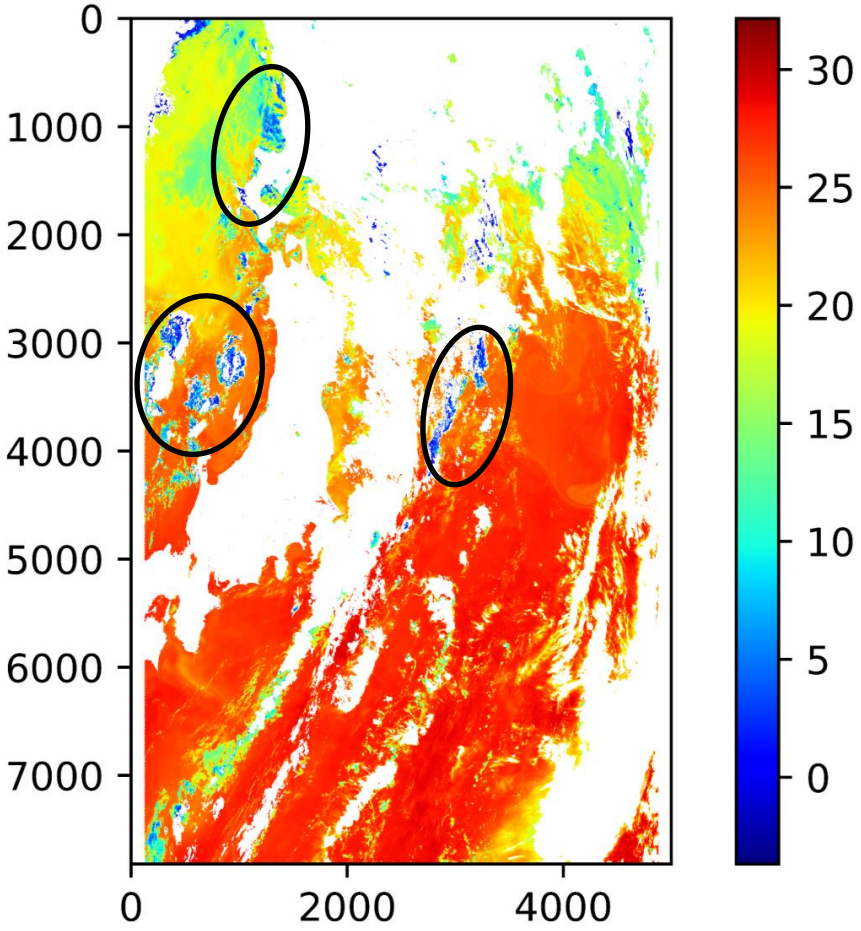
**QA_flag of SST**

| Bit | Data name |
|---|---|
| 0 | invalid data |
| 1 | Land |
| 2 | Rejected by QC |
| 3 | Retrieval error |
| 4 | Invalid data (TIR1) |
| 5 | Invalid data (TIR2) |
| 6 | reserved |
| 7 | reserved |
| 8 | 1: daytime, 0: nighttime or no visible data |
| 9 | Near Land |
| 10 | Cloudy |
| 11 | Unknown clear/cloudy |
| 12 | Possibly cloudy |
| 13 | Acceptable |
| 14 | Good |
| 15 | reserved |

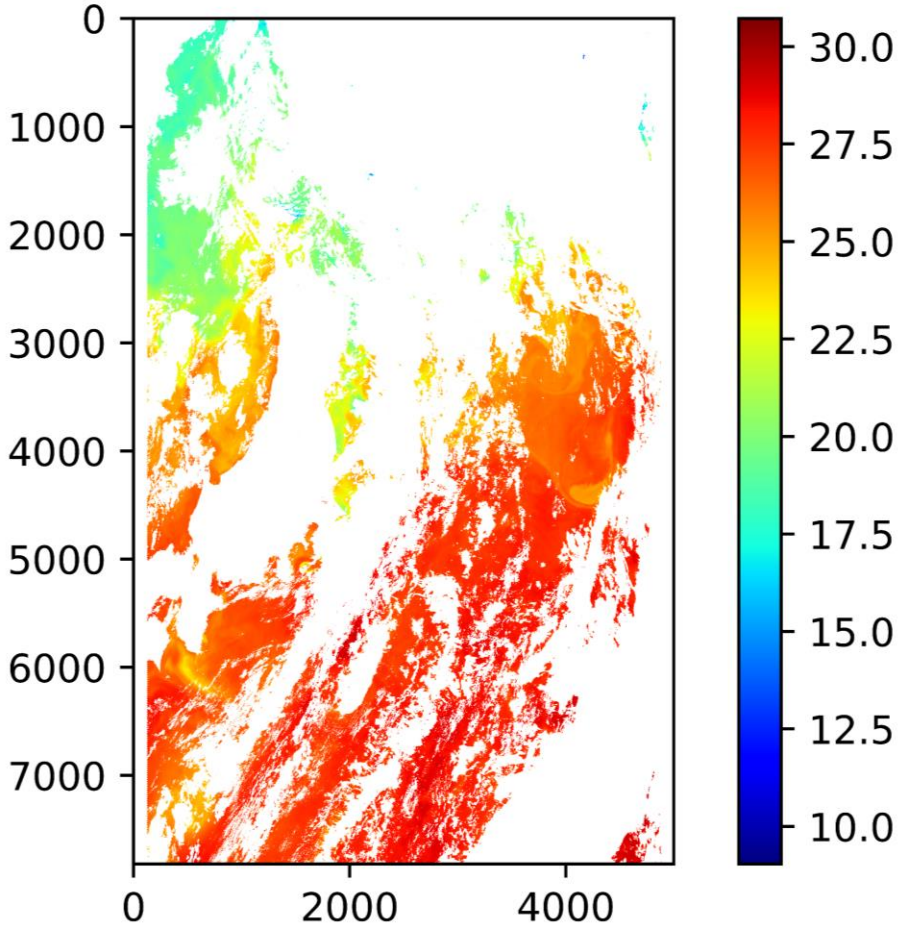
https://suzaku.eorc.jaxa.jp/GCOM_C/data/update/Algorithm_SST_ja.html

# Apply QA flags (if necessary)

Before applying QA_flag



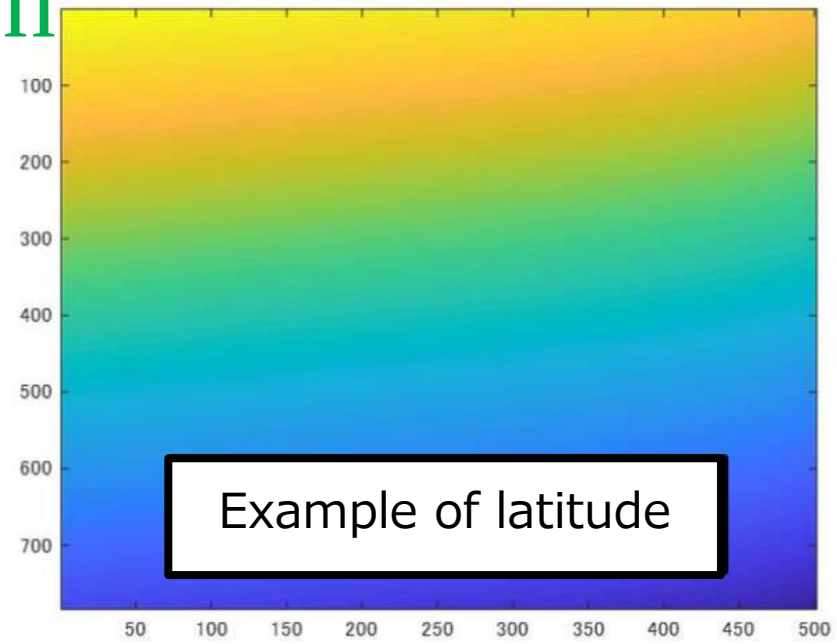

After applying QA_flag

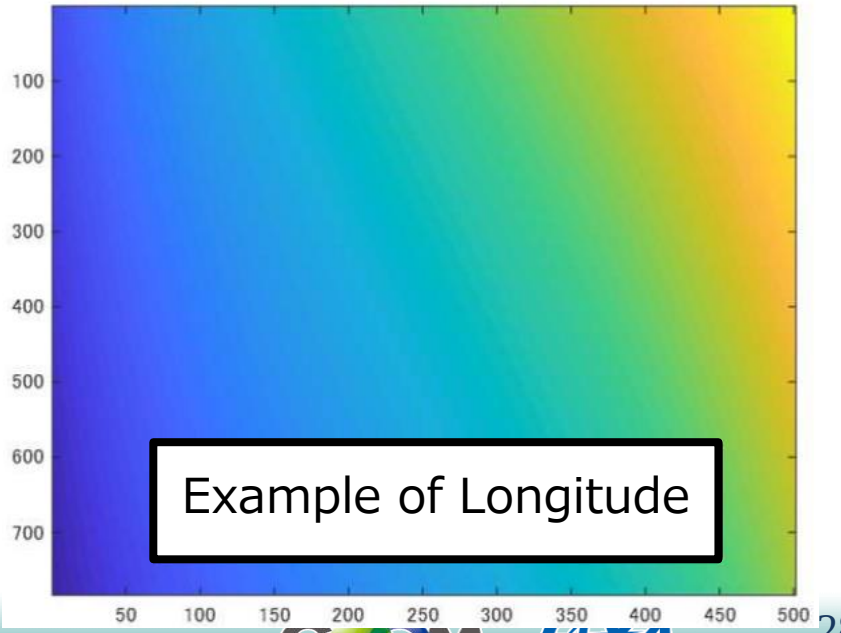



Unreliable data near clouds are masked.

# Latitude and longitude data acquisition

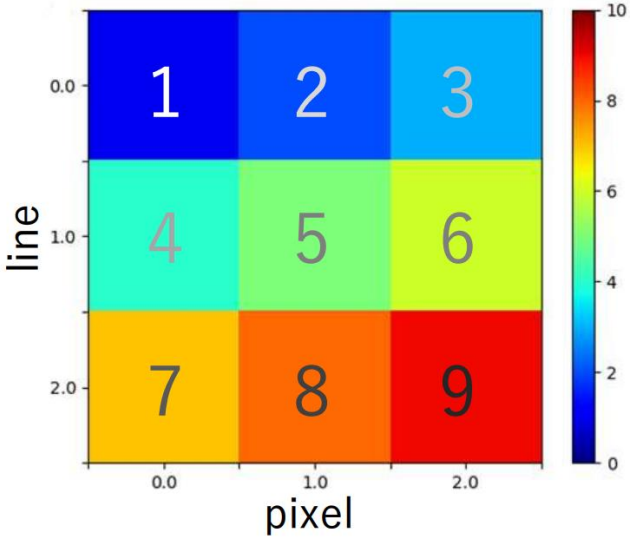
```
[ ]: # Read Latitude and Longitude
Lat = file['/Geometry_data/Latitude'][:]
Lat_r = float(file['/Geometry_data/Latitude'].attrs['Resampling_interval'])
Lon = file['/Geometry_data/Longitude'][:]
Lon_r = float(file['/Geometry_data/Longitude'].attrs['Resampling_interval'])

# Create meshgrid
X, Y = np.meshgrid(np.arange(1, Lat_r * Lat.shape[1] + 1, Lat_r),
                   np.arange(1, Lat_r * Lat.shape[0] + 1, Lat_r))

Xq, Yq = np.meshgrid(np.arange(1, Data0.shape[1] + 1),
                     np.arange(1, Data0.shape[0] + 1))

# Interpolate Latitude and Longitude
f_lat = griddata((X.flatten(), Y.flatten()), Lat.flatten(), (Xq, Yq), method='linear')
f_lon = griddata((X.flatten(), Y.flatten()), Lon.flatten(), (Xq, Yq), method='linear')

LLroi = {'Lat': f_lat,
         'Lon': f_lon}
```



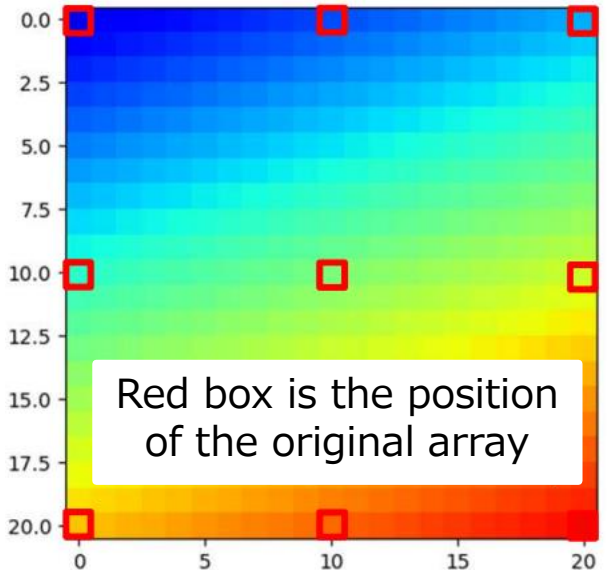

GCOM-C/SGLI ocean product files contain resampled latitude and longitude values

# Latitude and longitude data acquisition (note on interpolation method)

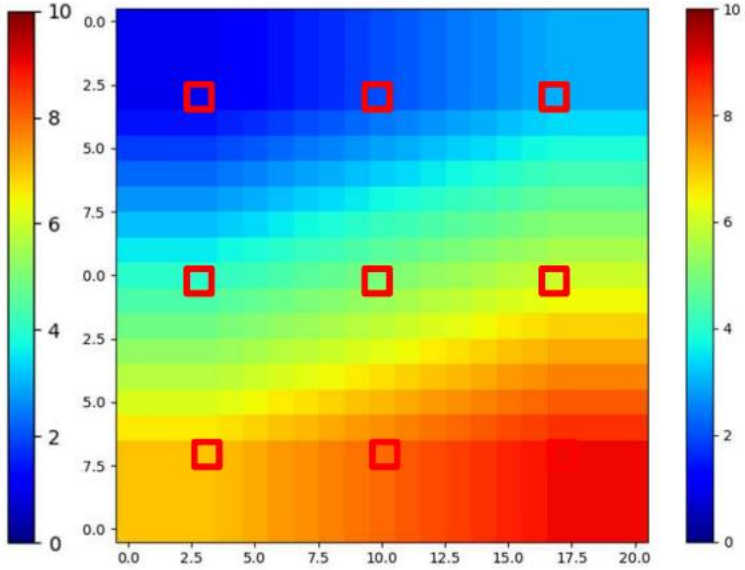


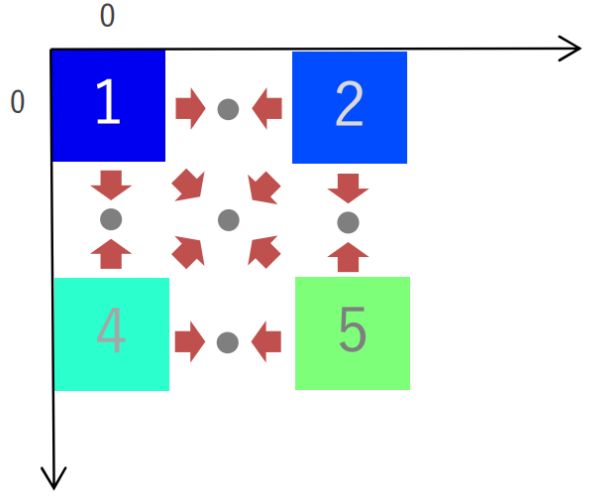

bilinear
Interpolation
(interval = 10)

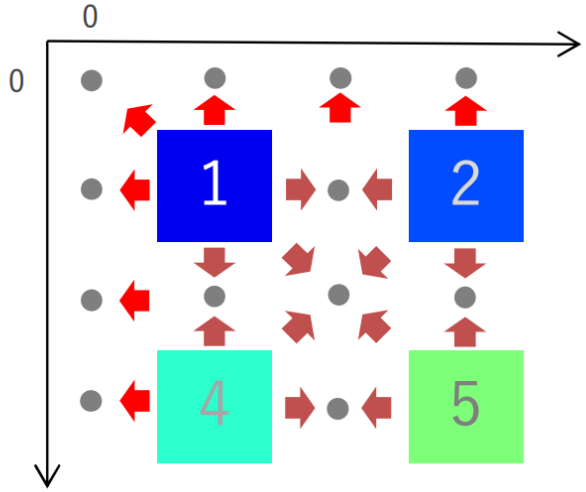
True




False



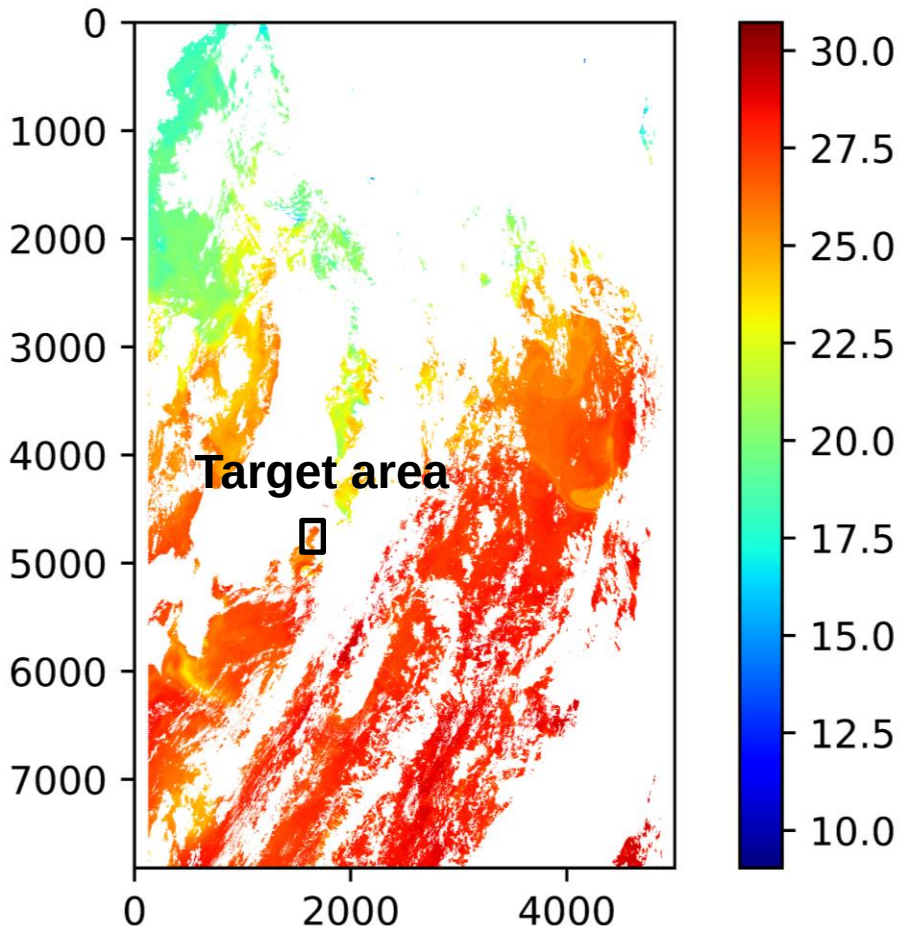




Linear interpolation of adjacent pixels with respect to the upper left corner
(note that some functions may result in a "false" completion depending on the language and function)

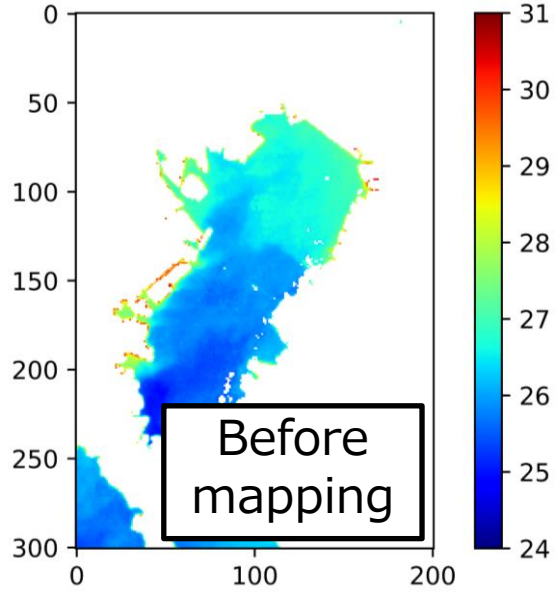
# Extract target data (Tokyo Bay)

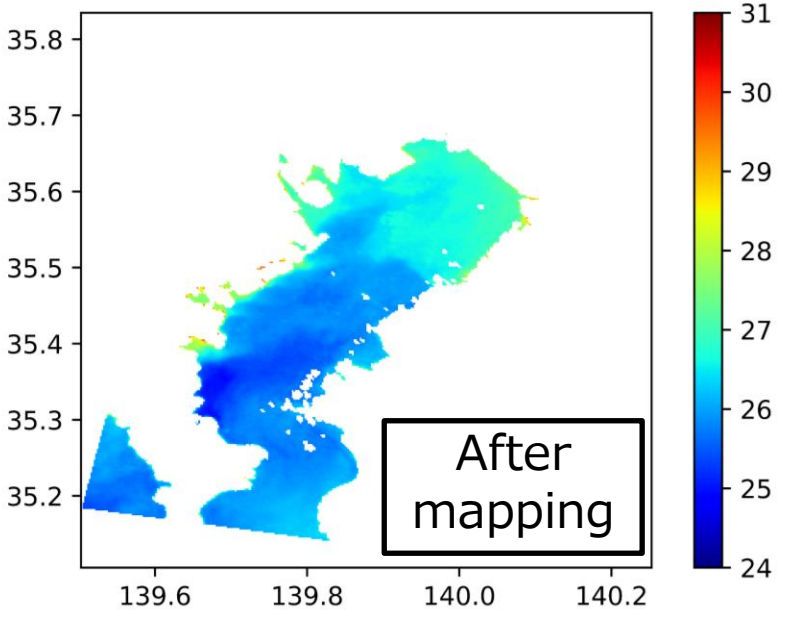
```
[ ]: # Extract some data
     IDX_X = slice(4600, 4900)
     IDX_Y = slice(1550, 1750)

     LLroi['Lat'] = LLroi['Lat'][IDX_X, IDX_Y]
     LLroi['Lon'] = LLroi['Lon'][IDX_X, IDX_Y]
     Data1 = Data1[IDX_X, IDX_Y]

     # ROI calculation
     DDeg = 10/4800
     ROI = {'LatLim': [np.min(LLroi['Lat']), np.max(LLroi['Lat'])],
            'LonLim': [np.min(LLroi['Lon']), np.max(LLroi['Lon'])]}

     Latg = np.arange(ROI['LatLim'][1], ROI['LatLim'][0] - DDeg, -DDeg)
     Long = np.arange(ROI['LonLim'][0], ROI['LonLim'][1] + DDeg, DDeg)

     LLg = np.meshgrid(Latg, Long, indexing='ij')
```



# Mapping

```
[ ]: # Scattered interpolation
points = np.column_stack((LLroi['Lat'].flatten(), LLroi['Lon'].flatten()))
values = Data1.flatten()

grid_lat, grid_lon = np.meshgrid(Latg, Long, indexing='ij')
grid_points = np.column_stack((grid_lat.flatten(), grid_lon.flatten()))

Data2 = griddata(points, values, grid_points, method='linear').reshape(grid_lat.shape)

# Plotting
plt.figure()
plt.imshow(Data1, vmin=24, vmax=31, cmap='jet')
plt.gca().set_aspect('equal', adjustable='box')
plt.colorbar()
plt.savefig("figure/Tokyo_bay.png", format="png", dpi=2000)

plt.figure()
plt.pcolormesh(Long, Latg, Data2, vmin=24, vmax=31, cmap='jet')
plt.gca().set_aspect('equal', adjustable='box')
plt.colorbar()
plt.savefig("figure/Tokyo_bay_mapping.png", format="png", dpi=2000)
plt.show()
```



We hope you will take advantage of the G-portal and GCOM-C/SGLI data!