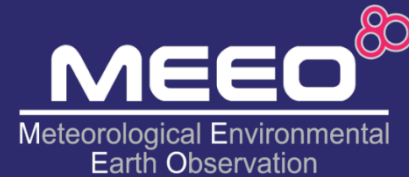


# FRIEND User Interface User's Guide

Flood Risk & Impact assessment through automatic change  
Detection of S-1+S-2 images



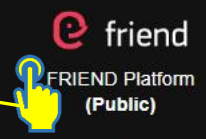


This pilot aims at assessing the flood risk in selected areas, its impact on urban areas and the associated risk for population. It uses Sentinel-1 and Sentinel-2 change detection data and relevant processing chains to generate time-series of imagery and automatically detect changes on flood events dates, in the following areas: 1) Char-Piya region, an island used also as validation area (Bangladesh), 2) the eastern part of Australia between Queensland and New South Wales, an area relevant for Climate security issues and for its correlation with flood events caused by both hydro-meteorological factors and sea level anomalies (as for case 1), 3) the west Darfur, as additional example of inland area.


The main output will be exposed through the FRIEND platform public GUI based on MEEO data cube technology, to provide citizens and non-expert users with a Flood Risk & Impact Assessment dashboard based on maps and time-series charts.


The FRIEND platform also addresses expert users through the Jupyter Notebook web-based interactive development environment, thus allowing to configure and run customized pipelines and workflow.

Access the public version (free login)

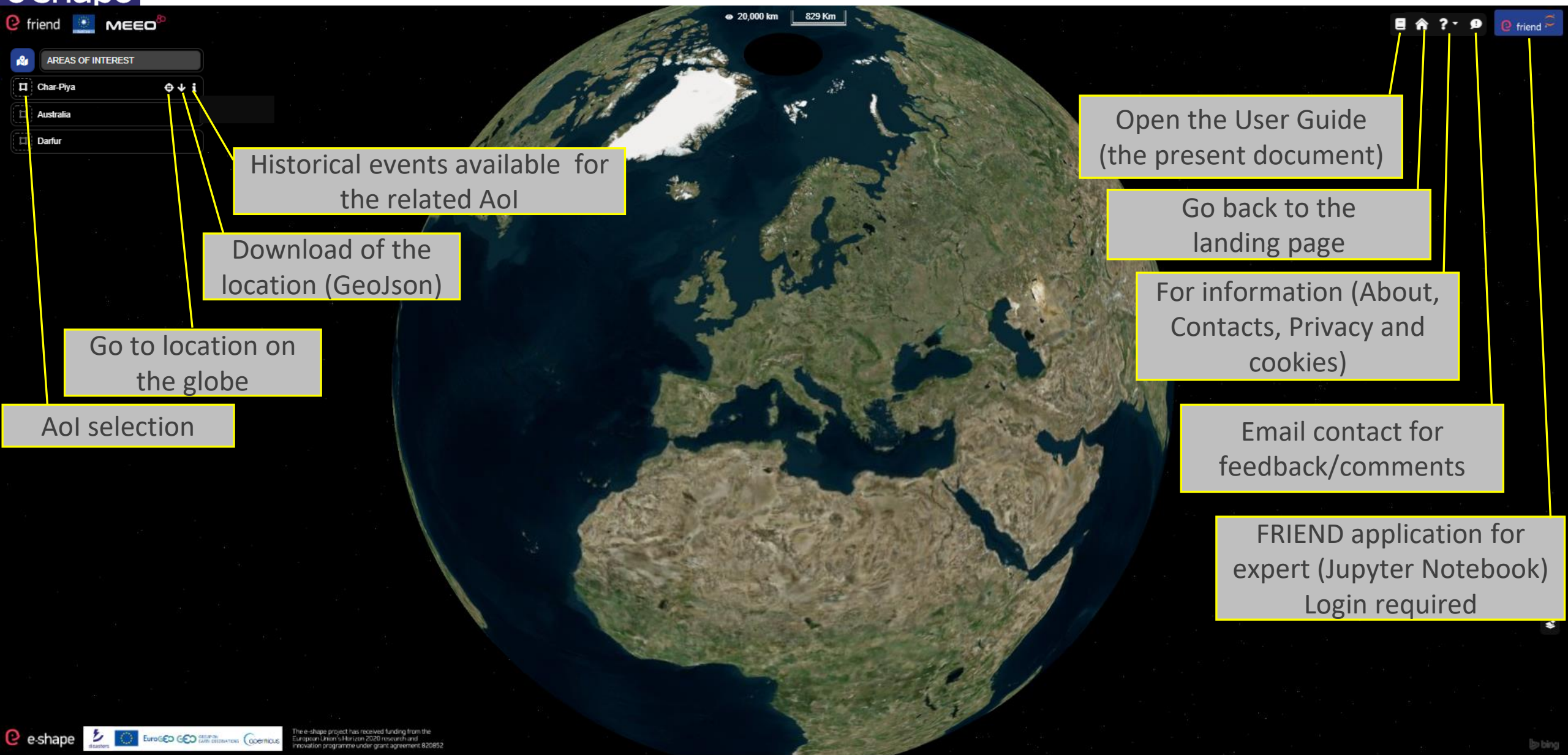


Most of the data are ingested and provided in near-real-time mode, thus covering the last recent days.

 The Flood Impact Assessment products, based on Sentinel-1 and Sentinel-2, are produced in the frame of SatCen RTDI activities, in particular they are processed by the GEO-DAMP pipelines.

 The inland GloFAS products are provided by the Copernicus Emergency Management Service – Global Flood Awareness System (GloFAS). GloFAS Terms & conditions

# Main options (buttons on the GUI)



Historical events available for the related Aol

Download of the location (GeoJson)

Go to location on the globe

Aol selection

Open the User Guide (the present document)

Go back to the landing page

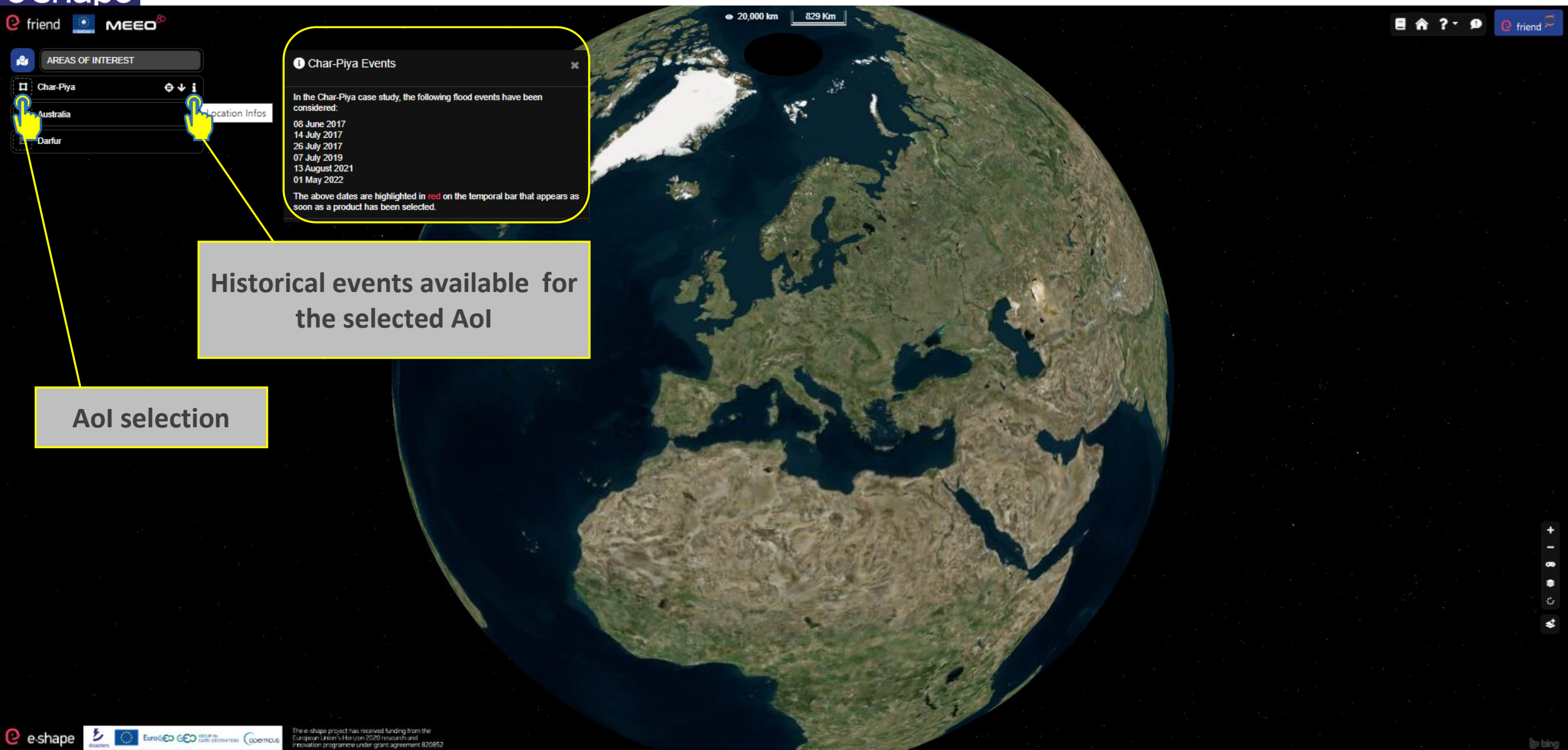
For information (About, Contacts, Privacy and cookies)

Email contact for feedback/comments

FRIEND application for expert (Jupyter Notebook)  
Login required

# Aoi (Area of Interest) selection

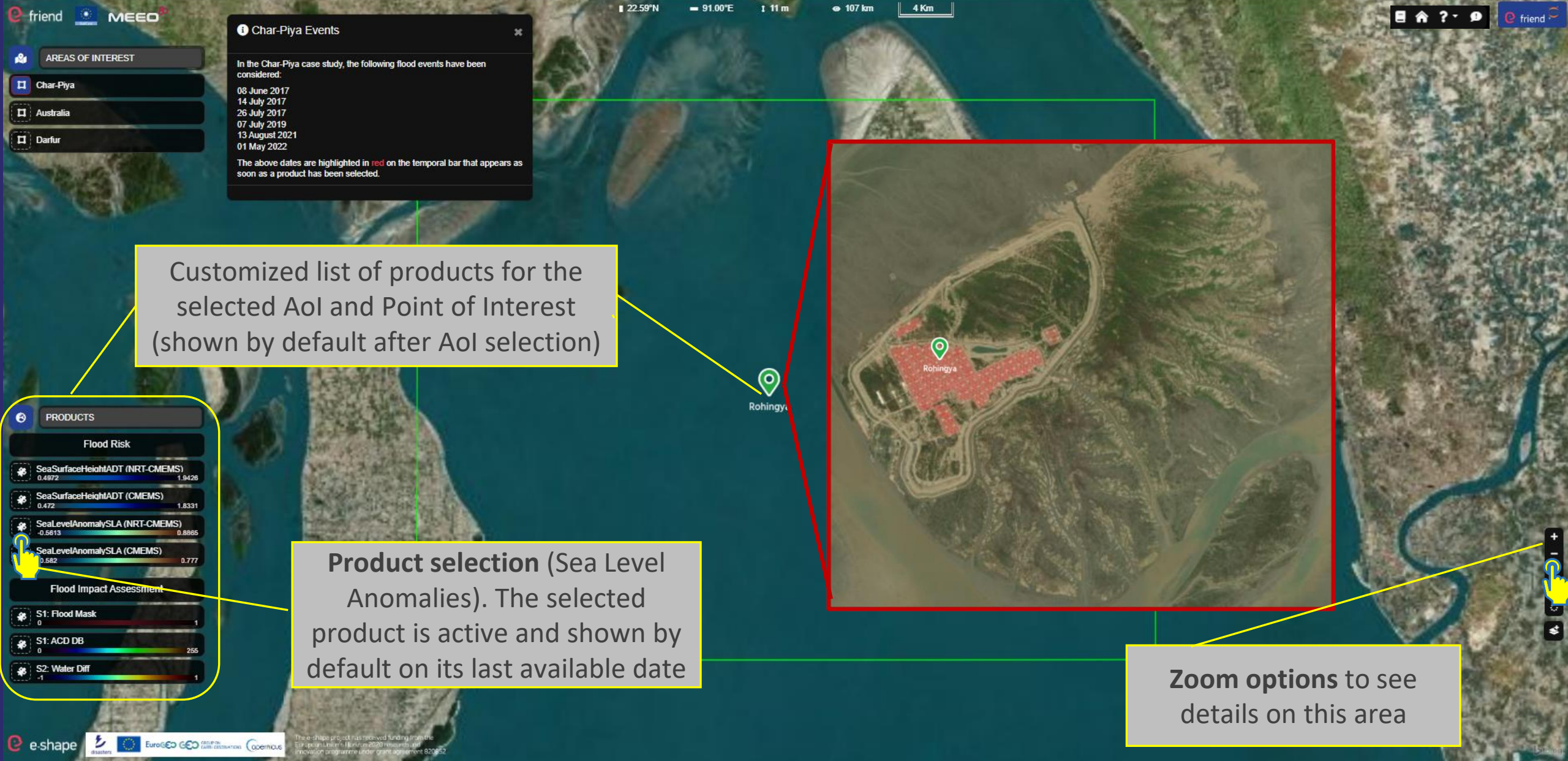
## Three available areas and information on events for each Aoi



Aoi selection

Historical events available for the selected Aoi

# Product selection (Sea Level Anomalies, by CMEMS)



**Char-Piya Events**

In the Char-Piya case study, the following flood events have been considered:

- 08 June 2017
- 14 July 2017
- 26 July 2017
- 07 July 2019
- 13 August 2021
- 01 May 2022

The above dates are highlighted in red on the temporal bar that appears as soon as a product has been selected.

Customized list of products for the selected Aoi and Point of Interest (shown by default after Aoi selection)

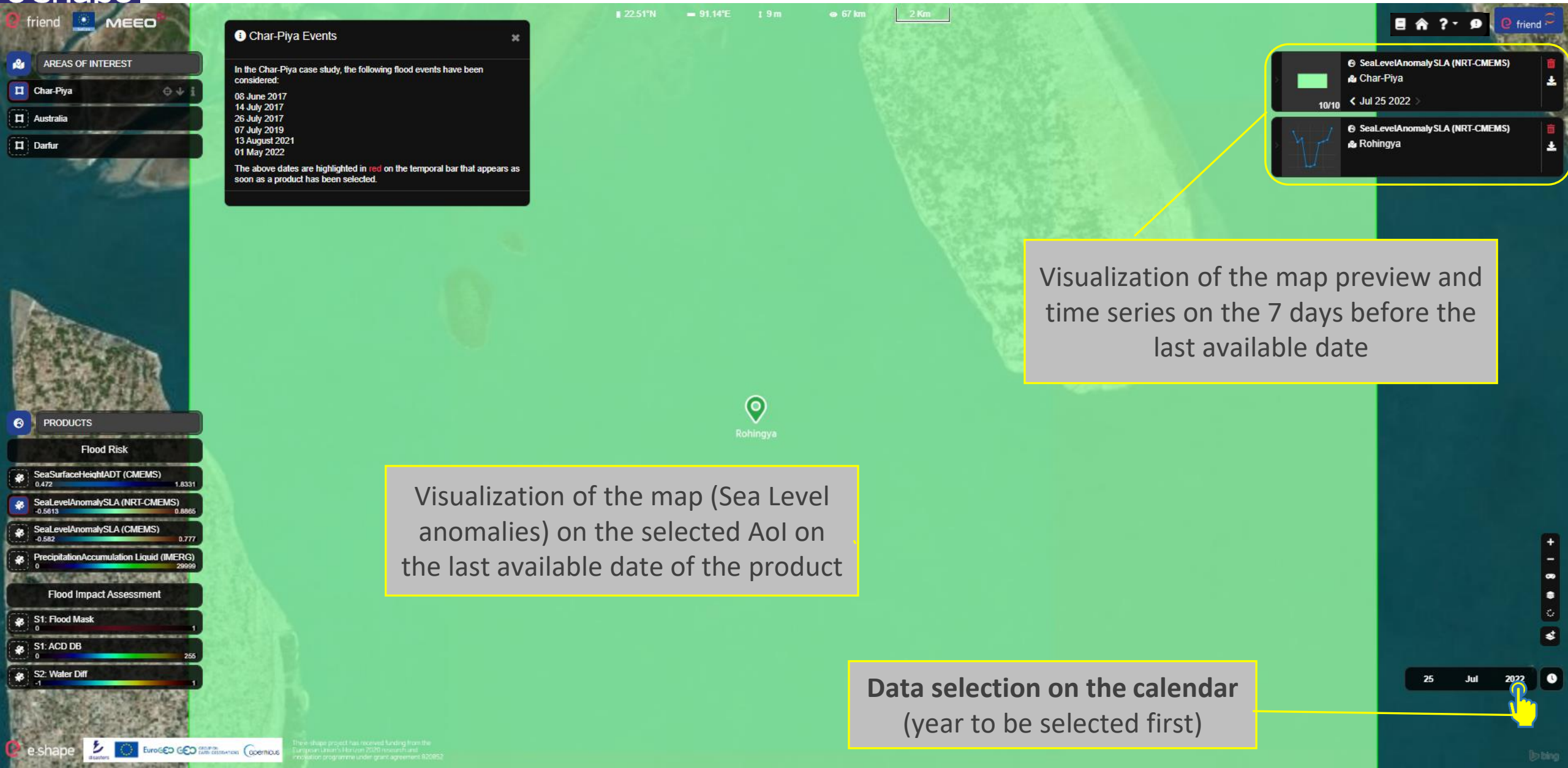
- PRODUCTS**
- Flood Risk
- SeaSurfaceHeightADT (NRT-CMEMS) 0.4972 1.9428
  - SeaSurfaceHeightADT (CMEMS) 0.472 1.8331
  - SeaLevelAnomalySLA (NRT-CMEMS) -0.5813 0.8885
  - SeaLevelAnomalySLA (CMEMS) 0.582 0.777
- Flood Impact Assessment
- S1: Flood Mask 0 1
  - S1: ACD DB 0 255
  - S2: Water Diff -1 1

Product selection (Sea Level Anomalies). The selected product is active and shown by default on its last available date



Zoom options to see details on this area

# Map and time series (by default after product selection)



Visualization of the map (Sea Level anomalies) on the selected Aoi on the last available date of the product

Visualization of the map preview and time series on the 7 days before the last available date

Data selection on the calendar (year to be selected first)

# Date selection on the Calendar (to select an historical event date)

**Char-Piya Events**

In the Char-Piya case study, the following flood events have been considered:

- 08 June 2017
- 14 July 2017
- 26 July 2017
- 07 July 2019
- 13 August 2021
- 01 May 2022

The above dates are highlighted in red on the temporal bar that appears as soon as a product has been selected.

**PRODUCTS**

- Flood Risk
  - SeaSurfaceHeightIADT (CMEMS) 0.472 1.8331
  - SeaLevelAnomalySLA (NRT-CMEMS) -0.5813 0.8865
  - SeaLevelAnomalySLA (CMEMS) -0.582 0.777
  - PrecipitationAccumulation Liquid (IMERG) 0 29999
- Flood Impact Assessment
  - S1: Flood Mask 0 1
  - S1: ACD DB 0 255
  - S2: Water Diff -1 1

**Selection of the year, month and day.**

Note: the temporal bar shows the dates on which the selected and active products are available.

In Red color the dates related to historical events for the selected Aol;  
In White color the other dates (usually around the events) on which the product data are available.

25 Jul 2022

2019 2020 2021 2022







# Impact assessment products (S1 Flood mask)

The screenshot displays the e-shape web application interface. At the top, there are navigation elements including 'friend' and 'MEOO'. The main map area shows a satellite view of a coastal region with a location pin for 'Rohingya'. A 'Char-Piya Events' pop-up window lists several dates: 08 June 2017, 14 July 2017, 26 July 2017, 07 July 2019, 13 August 2021, and 01 May 2022. A text box explains that these dates are highlighted in red on the temporal bar. On the left, there are panels for 'AREAS OF INTEREST' (Char-Piya, Australia, Darfur) and 'PRODUCTS' (Flood Risk, SeaSurfaceHeightADT, SeaLevelAnomalySLA, and PrecipitationAccumulation Liquid). The 'Flood Impact Assessment' section is highlighted with a yellow box, showing 'S1: Flood Mask' selected. On the right, there are time series plots for 'SeaLevelAnomalySLA (NRT-CMEMS) Rohingya', 'PrecipitationAccumulation Liquid (IMERG) Char-Piya', and 'PrecipitationAccumulation Liquid (IMERG) Rohingya'. The bottom of the interface features a calendar bar with the date '13 Aug 2021' selected.

Visualization of the map related to the last selected product (precipitation) and related time series

Select a new product (Flood mask, from the Impact assessment section)

# Impact assessment products (map visualization, time series not available)

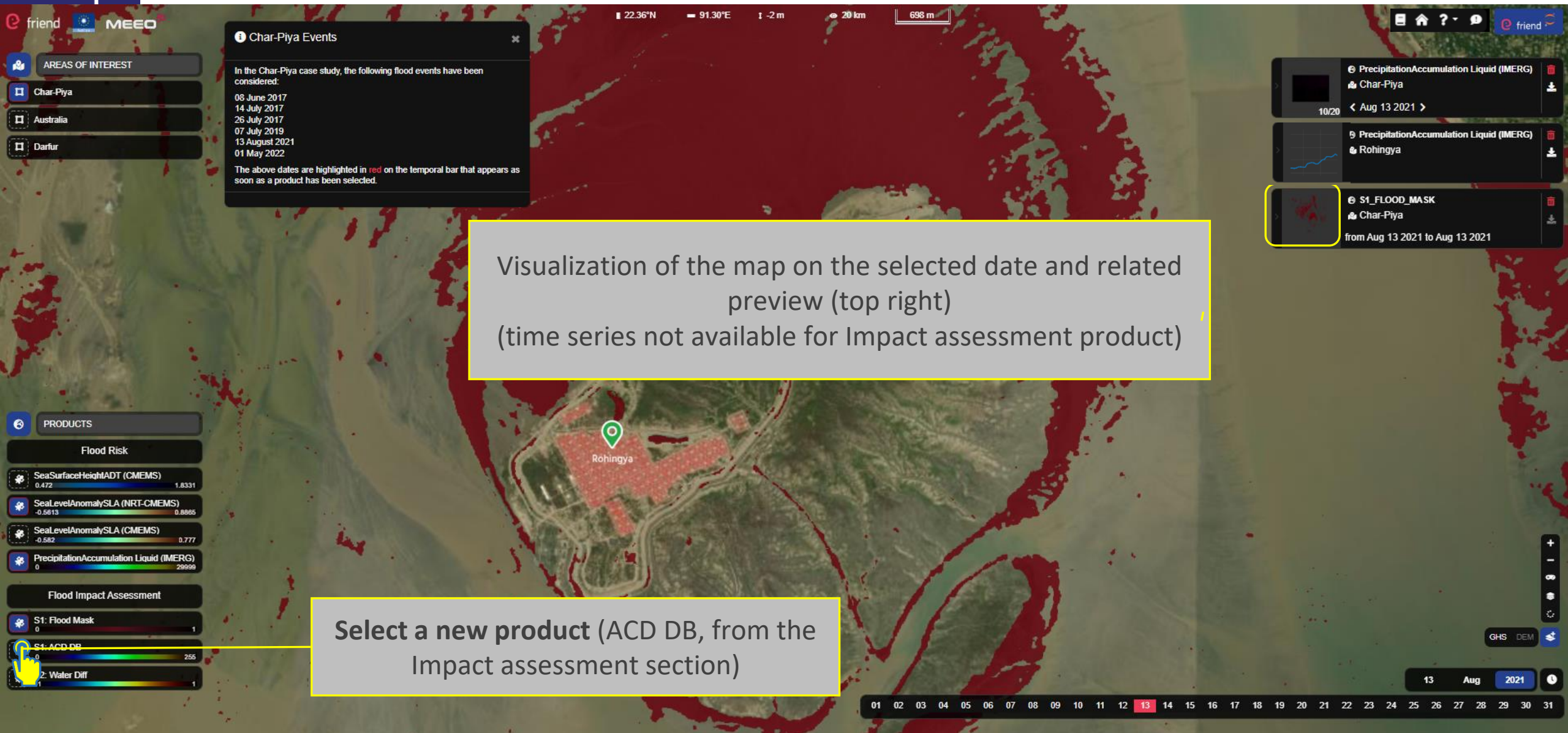
The screenshot displays the e-shape web application interface. At the top left, there are logos for 'friend' and 'MEE0'. Below them is a sidebar with 'AREAS OF INTEREST' including 'Char-Piya', 'Australia', and 'Darfur'. A central panel titled 'Char-Piya Events' lists dates: 08 June 2017, 14 July 2017, 26 July 2017, 07 July 2019, 13 August 2021, and 01 May 2022. The main map area shows a satellite view with red flood risk areas. A 'Rohingya' location is marked with a green pin. On the right, a layer list includes 'PrecipitationAccumulation Liquid (IMERG) Char-Piya', 'PrecipitationAccumulation Liquid (IMERG) Rohingya', and 'S1\_FLOOD\_MASK Char-Piya from May 01 2022 to May 01 2022'. At the bottom, a 'PRODUCTS' sidebar lists 'Flood Risk' and 'Flood Impact Assessment' with various data series. A temporal bar at the bottom shows dates from 01 to 31, with '01 May 2022' selected. A yellow box highlights the 'S1\_FLOOD\_MASK' layer in the layer list.

Visualization of the map of the selected product S1 Flood mask on its last available date and related preview (top right)

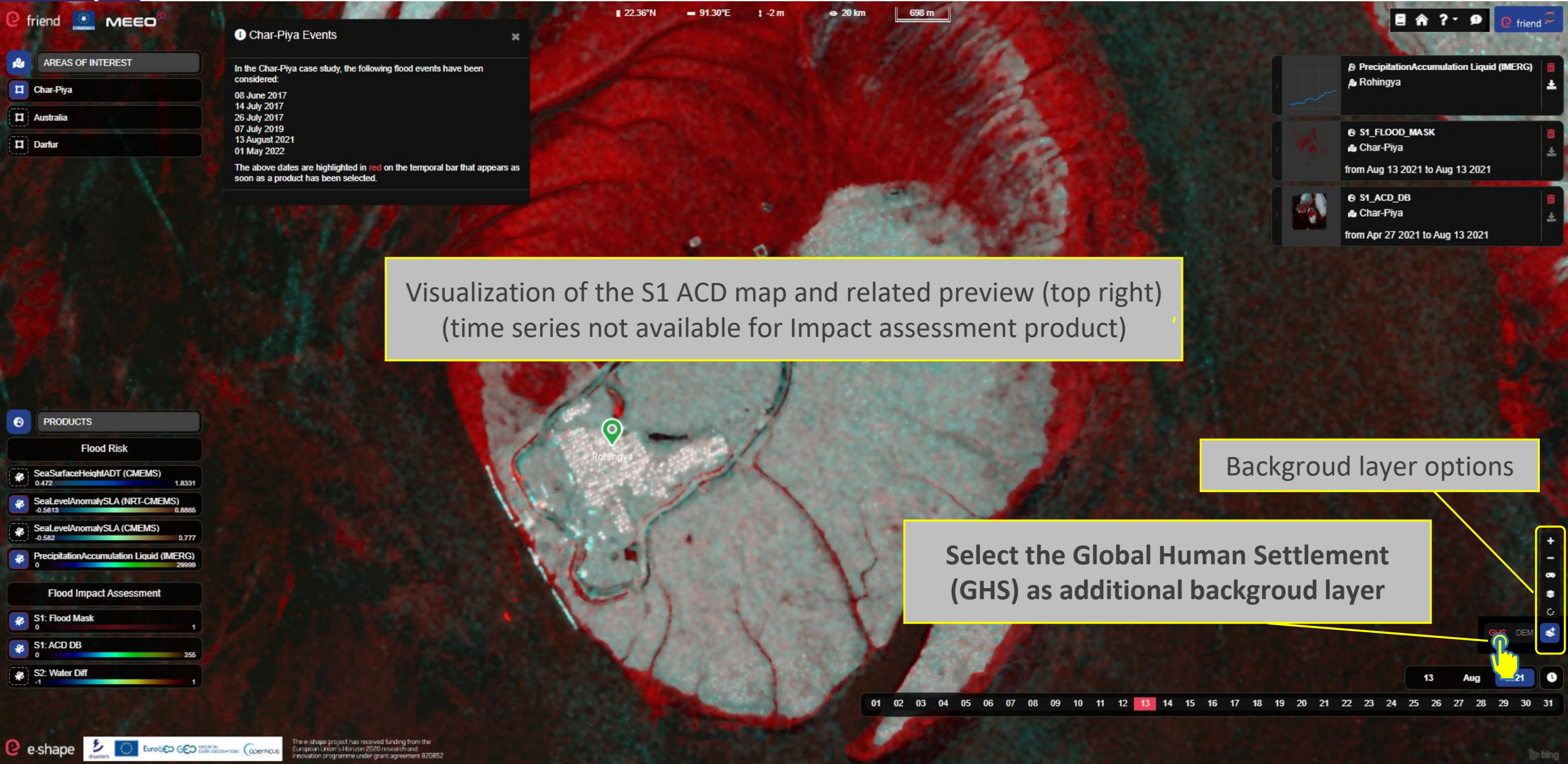
**Warning:** the temporal bar shows the dates on which ALL the selected and active products are available

Change the date

# Impact assessment products (S1 Change detection - ACD DB)



# Additional Background layer (GHS)

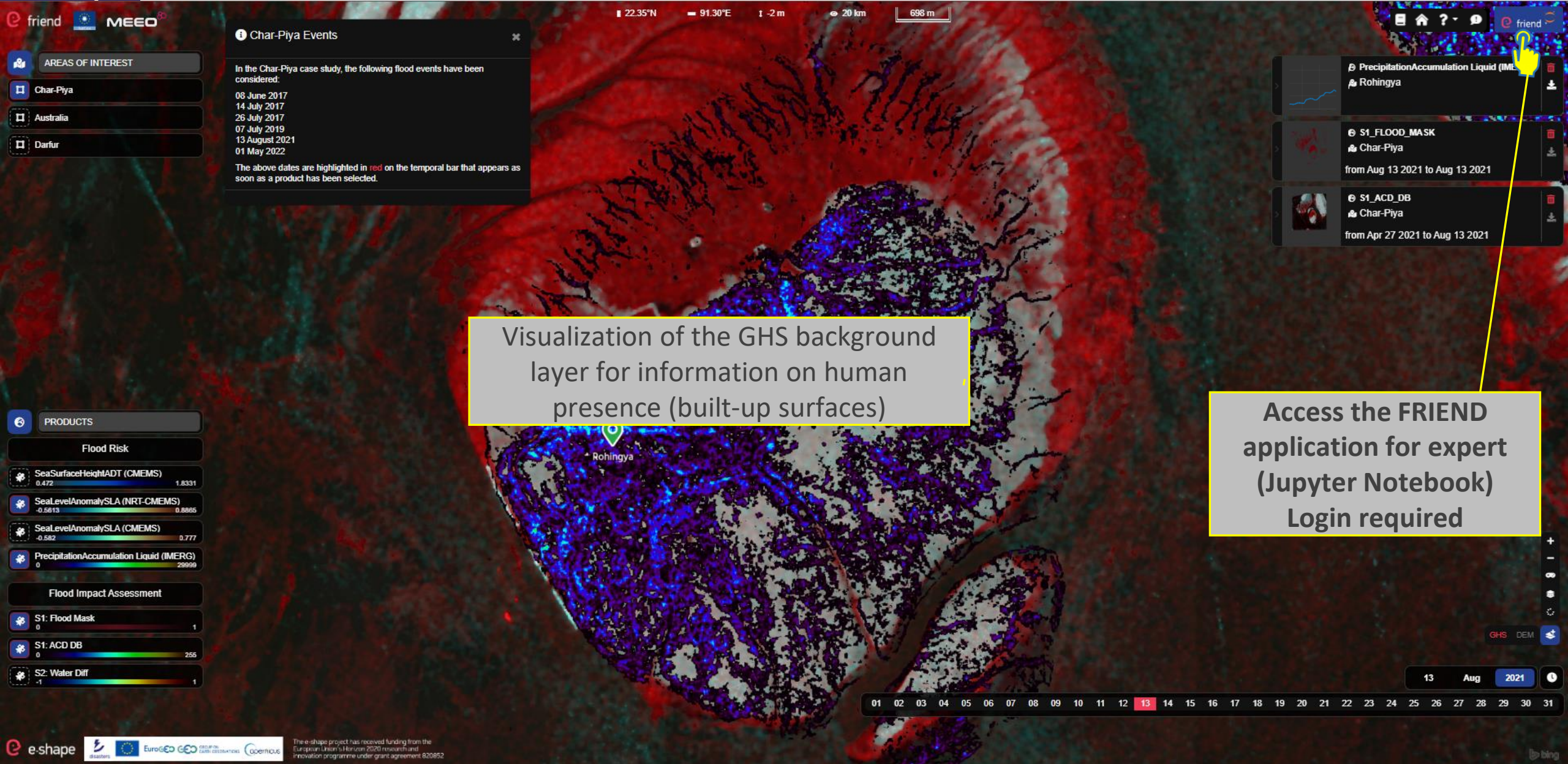


Visualization of the S1 ACD map and related preview (top right) (time series not available for Impact assessment product)

Background layer options

Select the Global Human Settlement (GHS) as additional background layer

# GHS layer and access the expert version (access to the Jupyter Notebook)



Visualization of the GHS background layer for information on human presence (built-up surfaces)

Access the FRIEND application for expert (Jupyter Notebook) Login required

(or in alternative from Landing page)

<https://friend.adamplatform.eu/>



This pilot aims at assessing the flood risk in selected areas, its impact on urban areas and the associated risk for population. It uses Sentinel-1 and Sentinel-2 change detection data and relevant processing chains to generate time-series of imagery and automatically detect changes on flood events dates, in the following areas: 1) Char-Piya region, an island used also as validation area (Bangladesh), 2) the eastern part of Australia between Queensland and New South Wales, an area relevant for Climate security issues and for its correlation with flood events caused by both hydro-meteorological factors and sea level anomalies (as for case 1), 3) the west Darfur, as additional example of inland area.

The main output will be exposed through the FRIEND platform public GUI based on MEEO data cube technology, to provide citizens and non-expert users with a Flood Risk & Impact Assessment dashboard based on maps and time-series charts.

The FRIEND platform also addresses expert users through the Jupyter Notebook web-based interactive development environment, thus allowing to configure and run customized pipelines and workflow.

  
FRIEND Platform  
(Public)

  
FRIEND Jupyter Notebook  
(Expert)

Access the version for expert (login required)

Most of the data are ingested and provided in near-real-time mode, thus covering the last recent days.



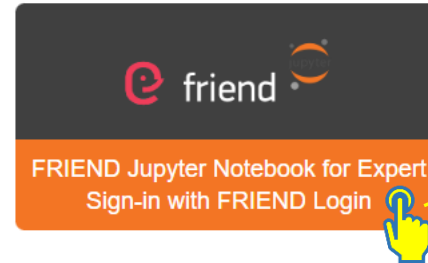
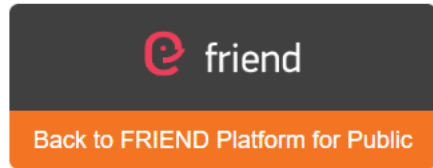
The Flood Impact Assessment products, based on Sentinel-1 and Sentinel-2, are produced in the frame of SatCen RTDI activities, in particular they are processed by the GEO-DAMP pipelines.



The inland GloFAS products are provided by the Copernicus Emergency Management Service – Global Flood Awareness System (GloFAS). GloFAS Terms & conditions

# FRIEND application for expert users

(please read before logging in)



Access the version for expert (login required)

Dear user, welcome to the FRIEND interface for experts, which is based on Jupyterlab Version 3.4.2.  
You will be able to work freely on the Base Image that you will find as default.

Keep in mind that your work environment will be cleaned up after 7 days of inactivity. We recommend to download your last version of the .ipynb file to avoid losing your changes.

# FRIEND application for expert users

(insert your credentials or self register for a new user)

### ADAM Platform Login

[Login](#)

[Forgot password?](#)

[Sign up New Account](#)

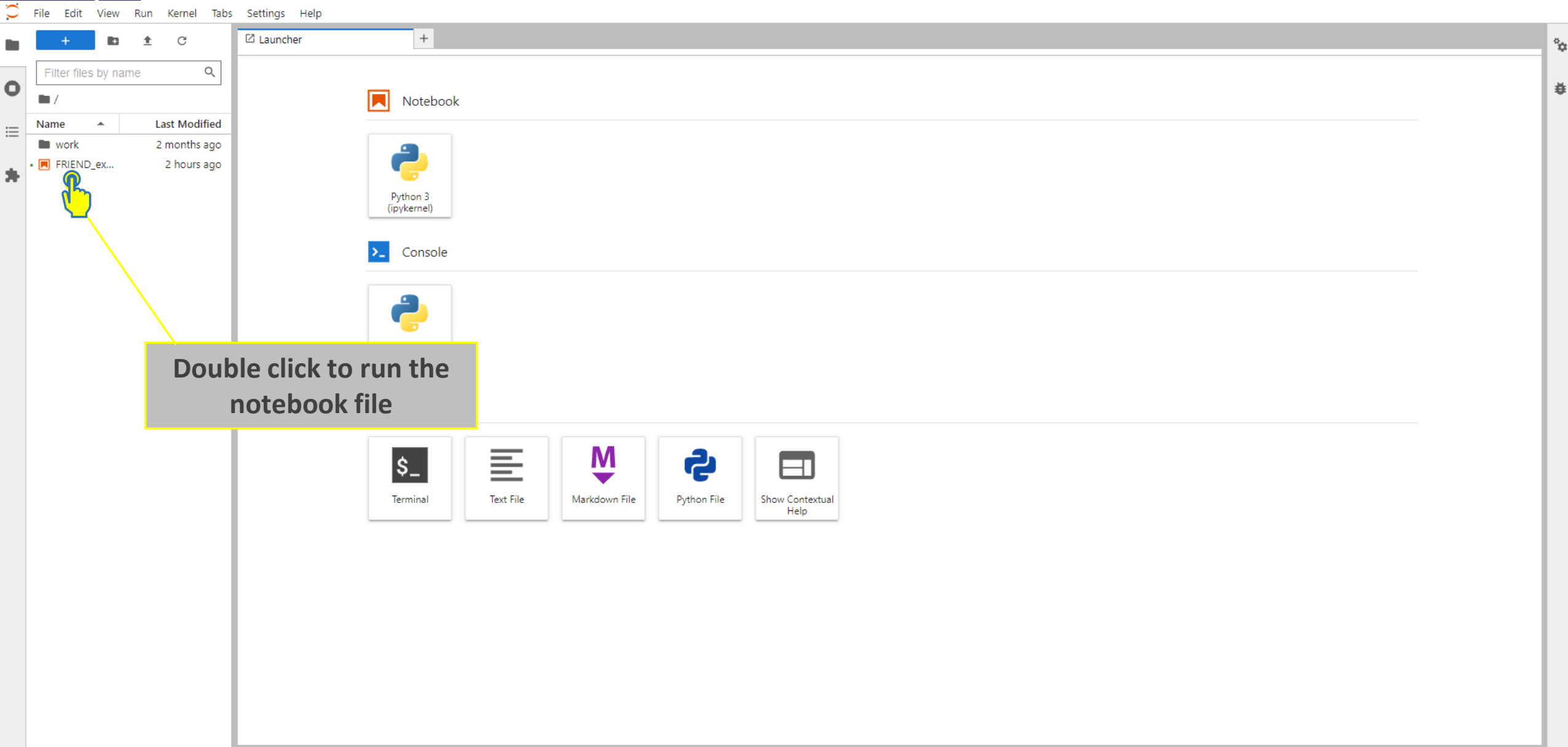
Insert your credentials

Or create a new one



# FRIEND application for expert users

(select the notebook file to run)



The screenshot shows the FRIEND application interface. On the left is a file explorer with a search bar and a table of files. The table has columns for 'Name' and 'Last Modified'. A file named 'FRIEND\_ex...' is highlighted, and a yellow callout box points to it with the text 'Double click to run the notebook file'. On the right is a 'Launcher' panel with a 'Notebook' section containing a 'Python 3 (ipykernel)' icon and a 'Console' section. At the bottom of the launcher are five icons: Terminal, Text File, Markdown File, Python File, and Show Contextual Help.

Name	Last Modified
work	2 months ago
FRIEND_ex...	2 hours ago

Launcher

Notebook

Python 3 (ipykernel)

Console

Terminal

Text File

Markdown File

Python File

Show Contextual Help



# FRIEND application for expert users

(and see the related results)

File Edit View Run Kernel Tabs Settings Help

Launcher x FRIEND\_expert\_notebook.ipynb Python 3 (ipykernel)

Filter files by name

Name	Last Modified
/	
work	2 months ago
FRIEND_ex...	a minute ago


## DAS queries to get the data (WMS and WCS examples)

```
[5]: #Area selection
#Please define desired bbox in 'bbox' variable
#some examples with already processed data are provided: charpiya, aus1, aus2 ...
bbox_charpiya='22.359375,91.3359375,22.4296875,91.40625' #TODO put proper coordinates
bbox_aus1='-33.73,150.66,-33.50,150.95'
bbox_aus2=None #TODO
bbox_aus3=None #TODO
bbox_aus4=None #TODO
bbox=bbox_aus1 #TO BE FILLED BY USER

#Visualize bbox
bbox_split=bbox.split(',')
minLat=float(bbox_split[0])
maxLat=float(bbox_split[2])
minLon=float(bbox_split[1])
maxLon=float(bbox_split[3])

m = folium.Map([(minLat+maxLat)/2, (minLon+maxLon)/2], zoom_start=8)
folium.Rectangle([(maxLat,minLon), (minLat,maxLon)],
                 color="green",
                 weight=2,
                 fill=True,
                 fill_color="pink",
                 fill_opacity=0.5).add_to(m)

m
```



[5]:

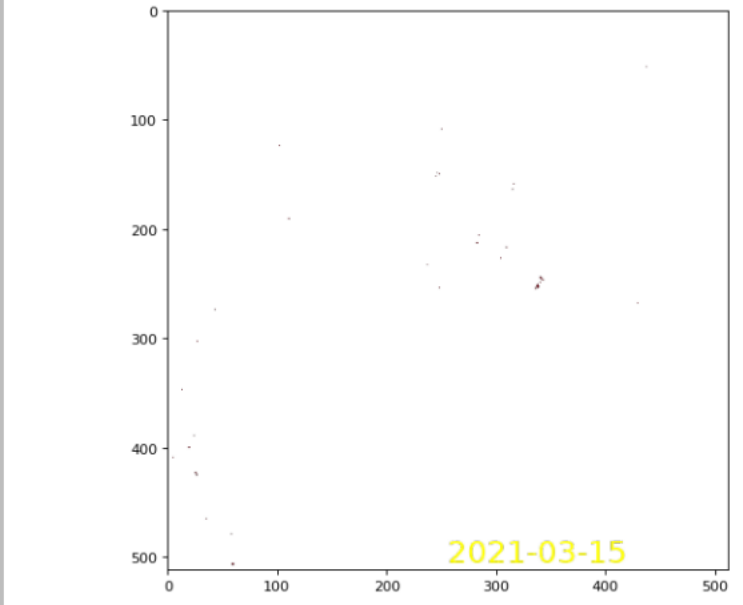
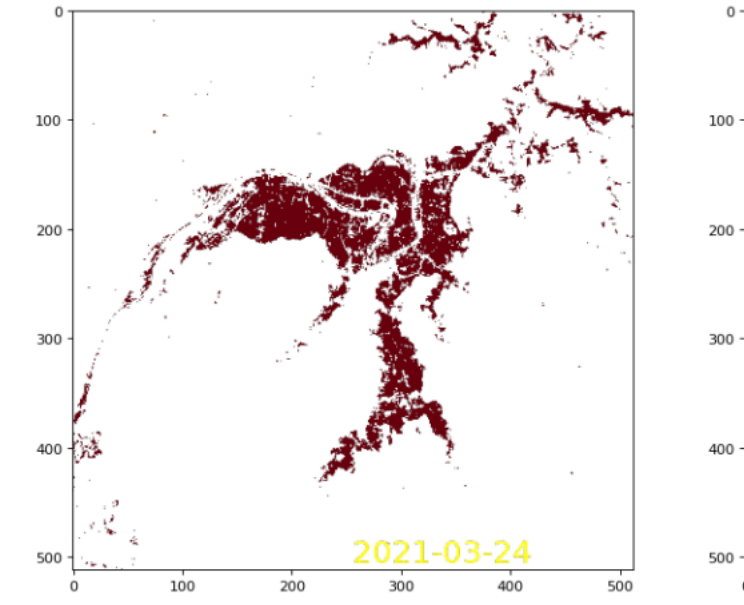
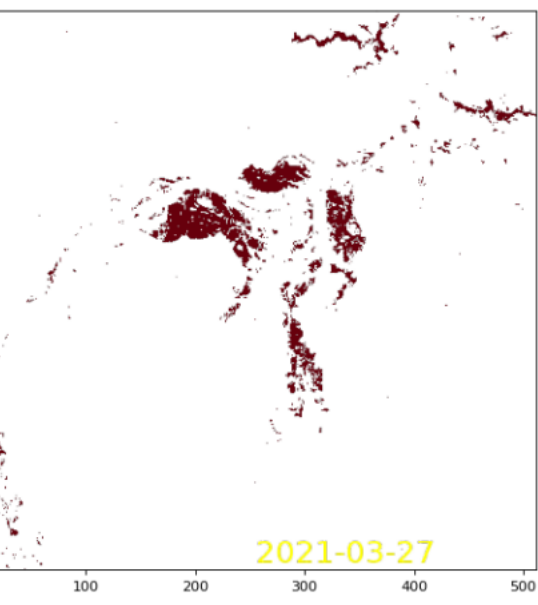
Simple 0 s 1 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 FRIEND\_expert\_notebook.ipynb

# FRIEND application for expert users (and see the related results)

File Edit View Run Kernel Tabs Settings Help

Launcher x FRIEND\_expert\_notebook.ipynb +

Python 3 (ipykernel)

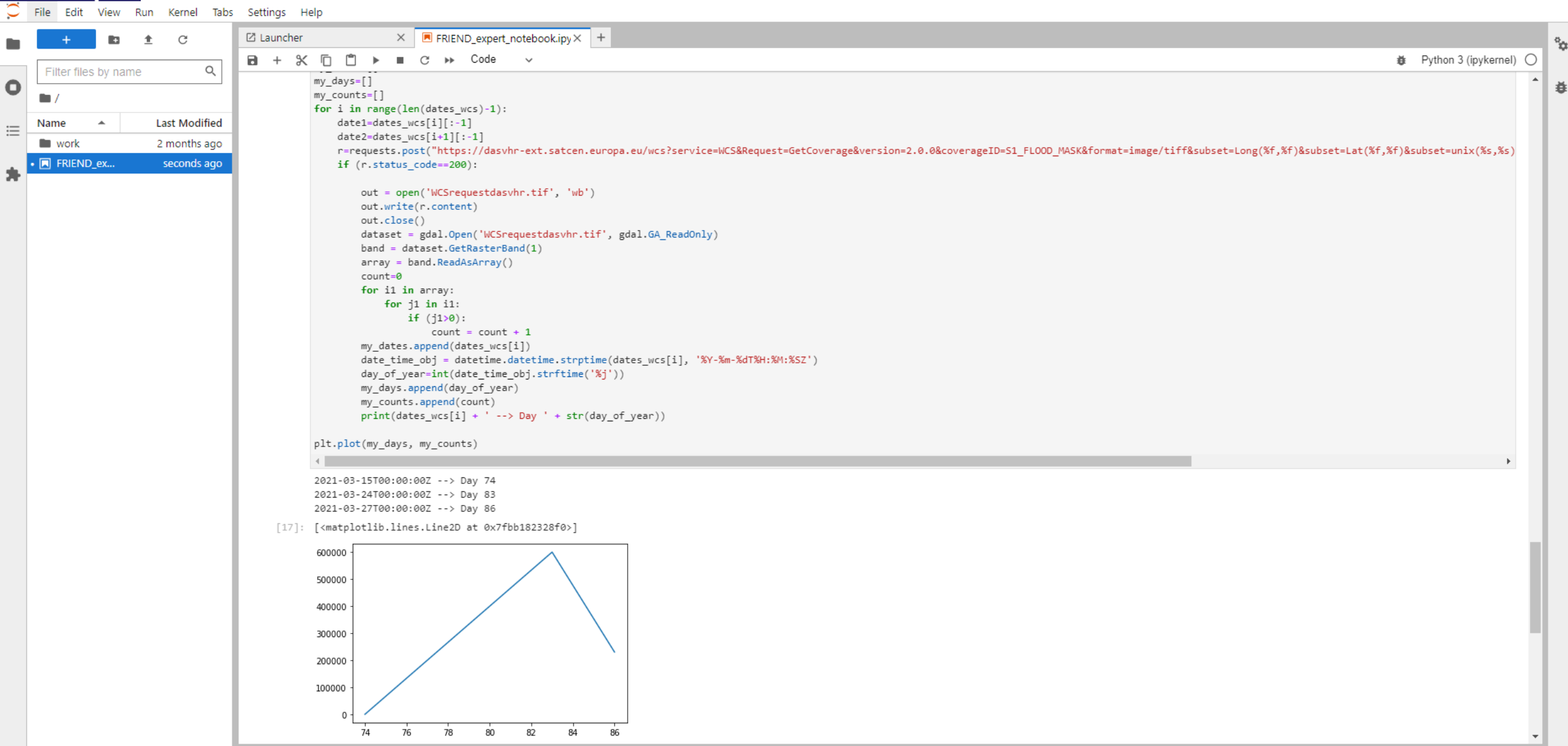
```
[17]: #Represent number of flooded pixels according to S1
from osgeo import gdal

#select time range
initial_date_wcs = date.fromisoformat('2021-03-01') #TO BE FILLED BY USER
final_date_wcs = date.fromisoformat('2021-03-31') #TO BE FILLED BY USER
interval_wcs = timedelta(days=1)
dates_wcs=[]
current_date_wcs=initial_date_wcs
while current_date_wcs <=final_date_wcs:
    dates_wcs.append(current_date_wcs.strftime("%Y-%m-%dT%H:%M:%SZ"))
    current_date_wcs = current_date_wcs + interval_wcs

my_dates=[]
my_days=[]
my_counts=[]
for i in range(len(dates_wcs)-1):
    date1=dates_wcs[i][:-1]
    date2=dates_wcs[i+1][:-1]
    r=requests.post("https://dasvhr-ext.satcen.europa.eu/wcs?service=WCS&Request=GetCoverage&version=2.0.0&coverageID=S1_FLOOD_MASK&format=image/tiff&subset=Long(%f,%f)&subset=Lat(%f,%f)&subset=unix(%s,%s)
```

Simple 0 1 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 FRIEND\_expert\_notebook.ipynb

# FRIEND application for expert users (and see the related results)



The screenshot shows a Jupyter Notebook interface with a file browser on the left and a code editor on the right. The code in the notebook is as follows:

```
my_days=[]
my_counts=[]
for i in range(len(dates_wcs)-1):
    date1=dates_wcs[i][:-1]
    date2=dates_wcs[i+1][:-1]
    r=requests.post("https://dasvhr-ext.satcen.europa.eu/wcs?service=WCS&Request=GetCoverage&version=2.0.0&coverageID=S1_FLOOD_MASK&format=image/tiff&subset=Long(%f,%f)&subset=Lat(%f,%f)&subset=unix(%s,%s)"
    if (r.status_code==200):

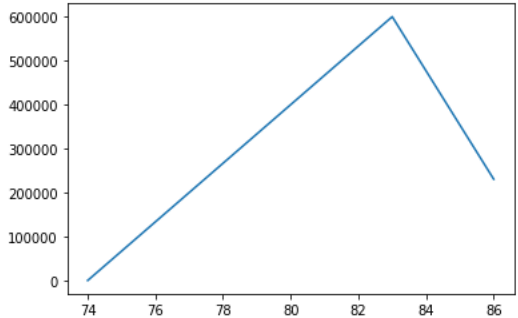
        out = open('WCSrequestdasvhr.tif', 'wb')
        out.write(r.content)
        out.close()
        dataset = gdal.Open('WCSrequestdasvhr.tif', gdal.GA_ReadOnly)
        band = dataset.GetRasterBand(1)
        array = band.ReadAsArray()
        count=0
        for i1 in array:
            for j1 in i1:
                if (j1>0):
                    count = count + 1
        my_dates.append(dates_wcs[i])
        date_time_obj = datetime.datetime.strptime(dates_wcs[i], '%Y-%m-%dT%H:%M:%SZ')
        day_of_year=int(date_time_obj.strftime('%j'))
        my_days.append(day_of_year)
        my_counts.append(count)
        print(dates_wcs[i] + ' --> Day ' + str(day_of_year))

plt.plot(my_days, my_counts)
```

The output of the code is:

```
2021-03-15T00:00:00Z --> Day 74
2021-03-24T00:00:00Z --> Day 83
2021-03-27T00:00:00Z --> Day 86
```

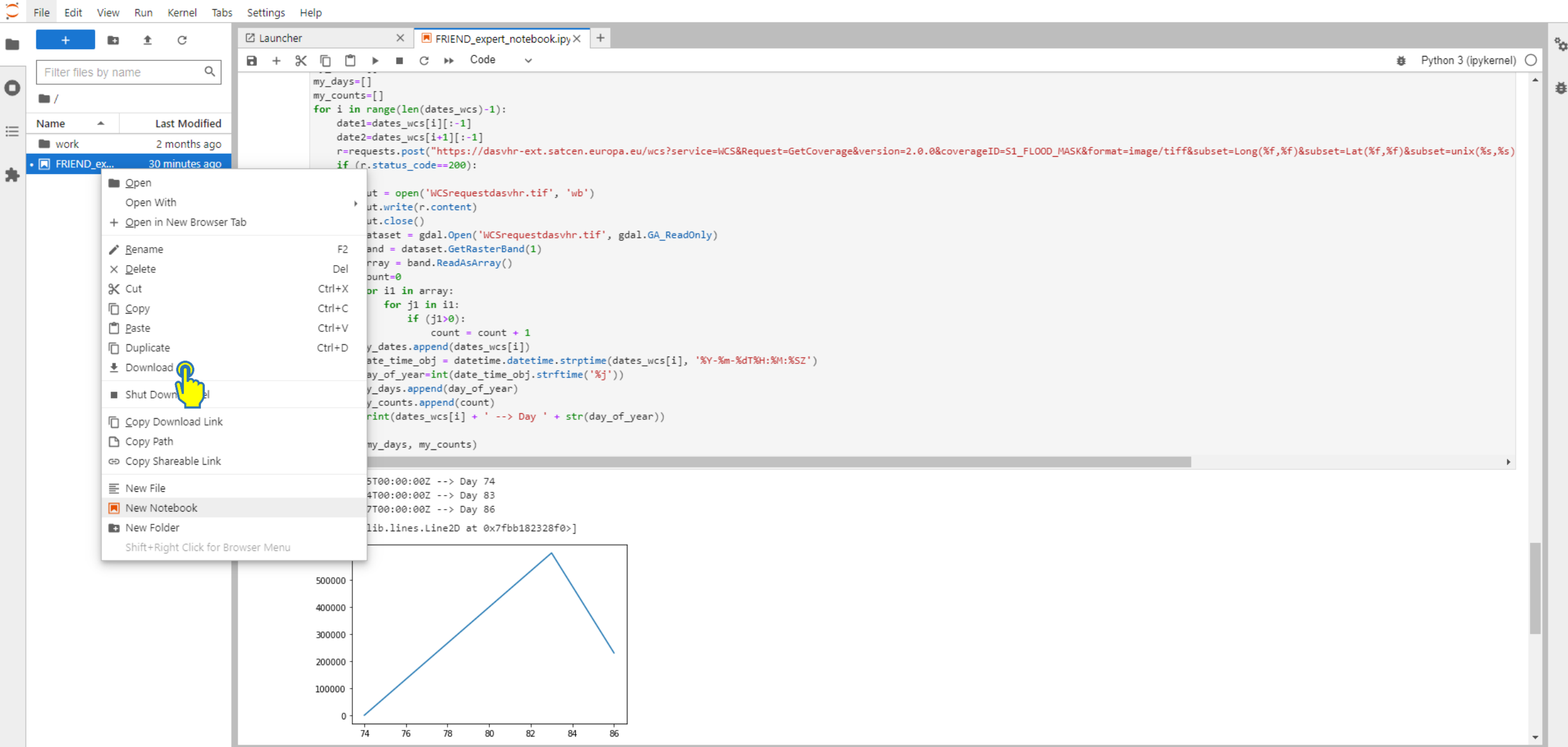
[17]: [matplotlib.lines.Line2D at 0x7fbb182328f0]



Day	Count
74	0
83	600000
86	250000

# FRIEND application for expert users

(save the file locally if you have modified it)

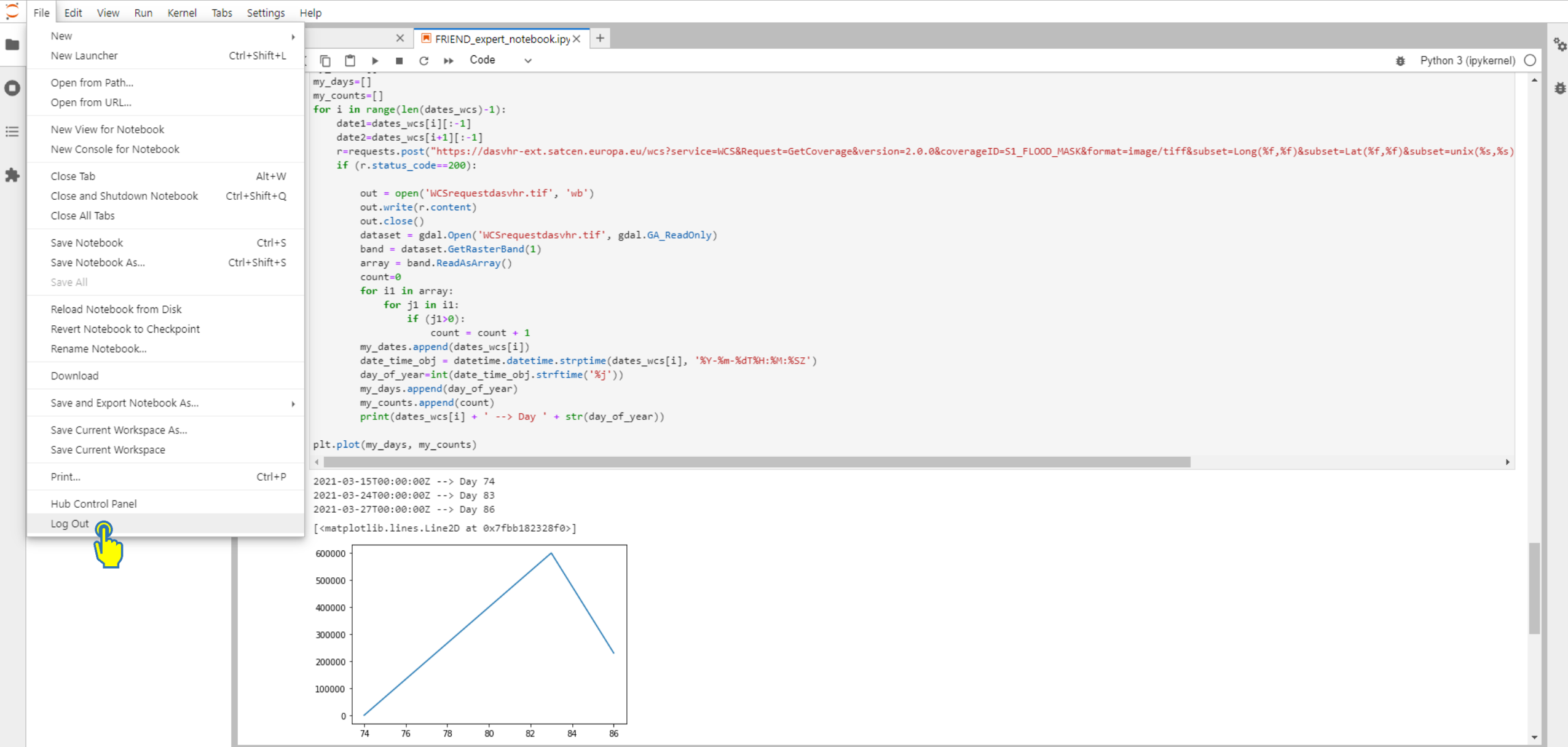


The screenshot displays the FRIEND application interface. On the left, a file explorer shows a directory structure with a file named 'FRIEND\_ex...' selected. A context menu is open over this file, with the 'Download' option highlighted by a yellow mouse cursor. The main area is a code editor showing Python code that interacts with a web service to retrieve flood coverage data. Below the code, a plot shows the number of days (y-axis, 0 to 500,000) versus the day of the year (x-axis, 74 to 86). The plot shows a sharp increase in the number of days starting around day 74, peaking at approximately 500,000 around day 83, and then decreasing.

```
my_days=[]
my_counts=[]
for i in range(len(dates_wcs)-1):
    date1=dates_wcs[i][:-1]
    date2=dates_wcs[i+1][:-1]
    r=requests.post("https://dasvhr-ext.satcen.europa.eu/wcs?service=WCS&Request=GetCoverage&version=2.0.0&coverageID=S1_FLOOD_MASK&format=image/tiff&subset=Long(%f,%f)&subset=Lat(%f,%f)&subset=unix(%s,%s)
    if (r.status_code==200):
        out = open('WCSrequestdasvhr.tif', 'wb')
        out.write(r.content)
        out.close()
        dataset = gdal.Open('WCSrequestdasvhr.tif', gdal.GA_ReadOnly)
        band = dataset.GetRasterBand(1)
        array = band.ReadAsArray()
        count=0
        for i1 in array:
            for j1 in i1:
                if (j1>0):
                    count = count + 1
        my_dates.append(dates_wcs[i])
        date_time_obj = datetime.datetime.strptime(dates_wcs[i], '%Y-%m-%dT%H:%M:%SZ')
        day_of_year=int(date_time_obj.strftime('%j'))
        my_days.append(day_of_year)
        my_counts.append(count)
    print(dates_wcs[i] + ' --> Day ' + str(day_of_year))

my_days, my_counts)
5T00:00:00Z --> Day 74
4T00:00:00Z --> Day 83
7T00:00:00Z --> Day 86
lib.lines.Line2D at 0x7fbb182328f0>]
```

# FRIEND application for expert users (log out)



The screenshot shows a JupyterLab notebook with a Python script in the code editor and its output in the console and a plot.

```

my_days=[]
my_counts=[]
for i in range(len(dates_wcs)-1):
    date1=dates_wcs[i][::-1]
    date2=dates_wcs[i+1][::-1]
    r=requests.post("https://dasvhr-ext.satcen.europa.eu/wcs?service=WCS&Request=GetCoverage&version=2.0.0&coverageID=S1_FLOOD_MASK&format=image/tiff&subset=Long(%f,%f)&subset=Lat(%f,%f)&subset=unix(%s,%s)")
    if (r.status_code==200):

        out = open('WCSrequestdasvhr.tif', 'wb')
        out.write(r.content)
        out.close()
        dataset = gdal.Open('WCSrequestdasvhr.tif', gdal.GA_ReadOnly)
        band = dataset.GetRasterBand(1)
        array = band.ReadAsArray()
        count=0
        for i1 in array:
            for j1 in i1:
                if (j1>0):
                    count = count + 1
        my_dates.append(dates_wcs[i])
        date_time_obj = datetime.datetime.strptime(dates_wcs[i], '%Y-%m-%dT%H:%M:%SZ')
        day_of_year=int(date_time_obj.strftime('%j'))
        my_days.append(day_of_year)
        my_counts.append(count)
        print(dates_wcs[i] + ' --> Day ' + str(day_of_year))

plt.plot(my_days, my_counts)

```

Output in the console:

```

2021-03-15T00:00:00Z --> Day 74
2021-03-24T00:00:00Z --> Day 83
2021-03-27T00:00:00Z --> Day 86

```

The plot shows a line graph with the following data points:

Date	Day	Count
2021-03-15T00:00:00Z	74	0
2021-03-24T00:00:00Z	83	600000
2021-03-27T00:00:00Z	86	250000

The 'Log Out' option in the File menu is highlighted with a yellow hand cursor.



e-shape

# Thank you!

Follow us:



eshapeh2020



@eshape\_eu



e-shape project

[www.e-shape.eu](http://www.e-shape.eu)