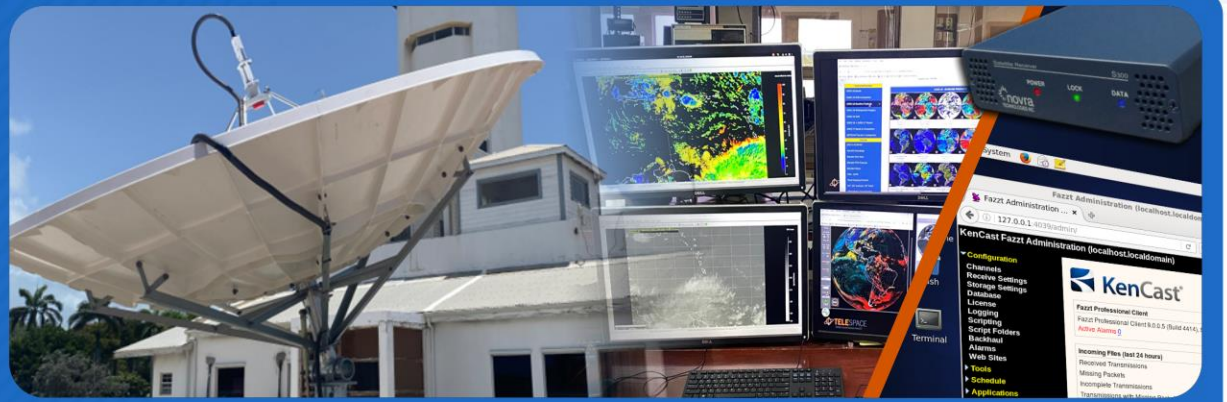
GEONETCast-Americas Training for the Eastern Caribbean States

Day 1 - May 3rd

Session 4:

Hands-on
Introduction to Python



Diego Souza
diego.souza@inpe.br

DISSM - Meteorological Satellites and Sensors' Division
CGCT - General Coordination of Earth Sciences
INPE - National Institute for Space Research

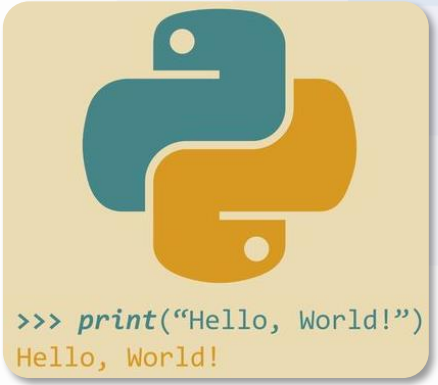
SDCR PROJECT
Strengthening Disaster and Climate Resilience in the Eastern and Southern Caribbean

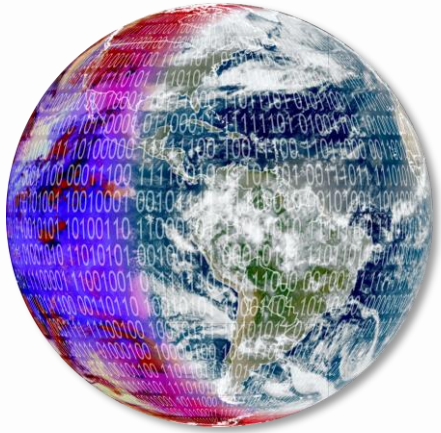
USAID FROM THE AMERICAN PEOPLE

Caribbean Institute for Meteorology and Hydrology

Presentation Outline

- **Introduction**
  - The Python programming language
  - Useful libraries
- **Application Examples**
  - DSAT, SHOWCast, NWP and other
- **Basic concepts of Google COLAB**
  - Creating a copy of the notebook, text and code cells, etc.
- **Demonstrations**





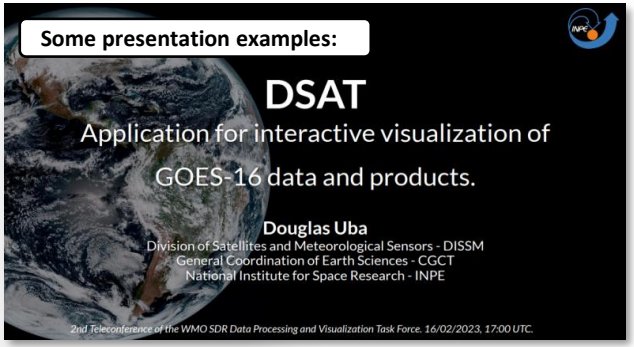
WMO Data Processing and Visualization Task Force

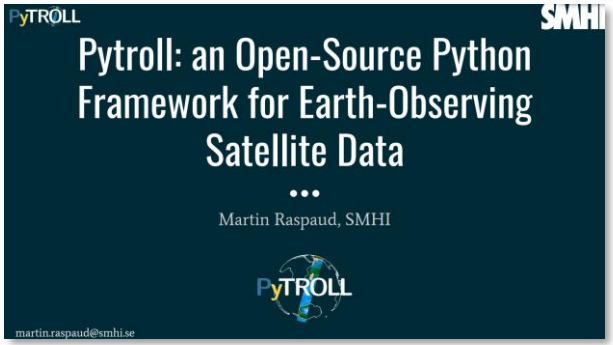
https://sdr.ucr.ac.cr/data-processing-and-visualization

Goals

1. Facilitate the generation and utilization of satellite products joining efforts from developers from different member countries.

2. Encourage the usage, creation and distribution of open-source solutions (scripts and visualization tools).





To participate in the Group, send an email to diego.souza@inpe.br y marcial.garbanzo@ucr.ac.cr

Receiving Mechanisms and Python Libraries

amazon web services

Environmental Data Commons

Google Cloud Platform

unidata IDD

CLASS

EUMETSAT Data Centre

PyProj

pyHDF

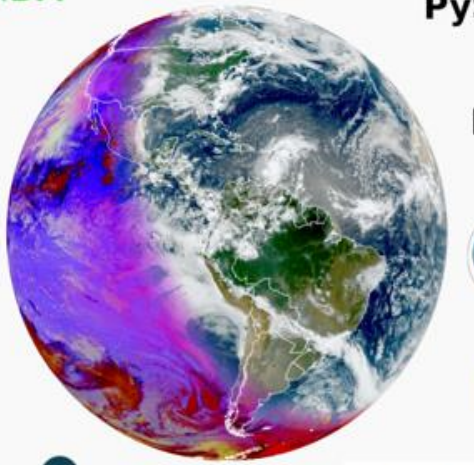
GMT THE GENERIC MAPPING TOOLS

ANACONDA

python

PySpectral

SciPy



PyOrbital

cartopy

GDAL

NumPy

Folium

matplotlib

Siphon

PyTROLL SatPy

NetCDF4

MetPy

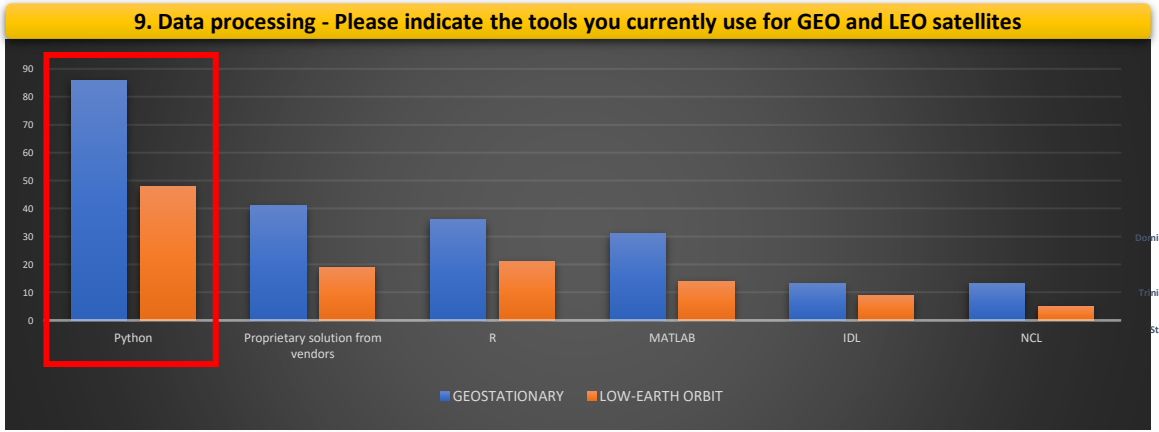
pyGRIB

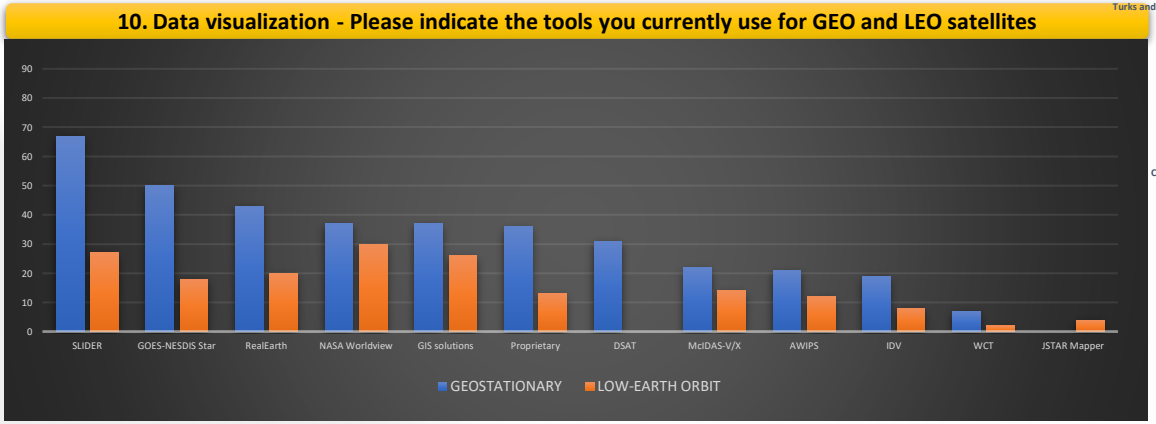
Basemap

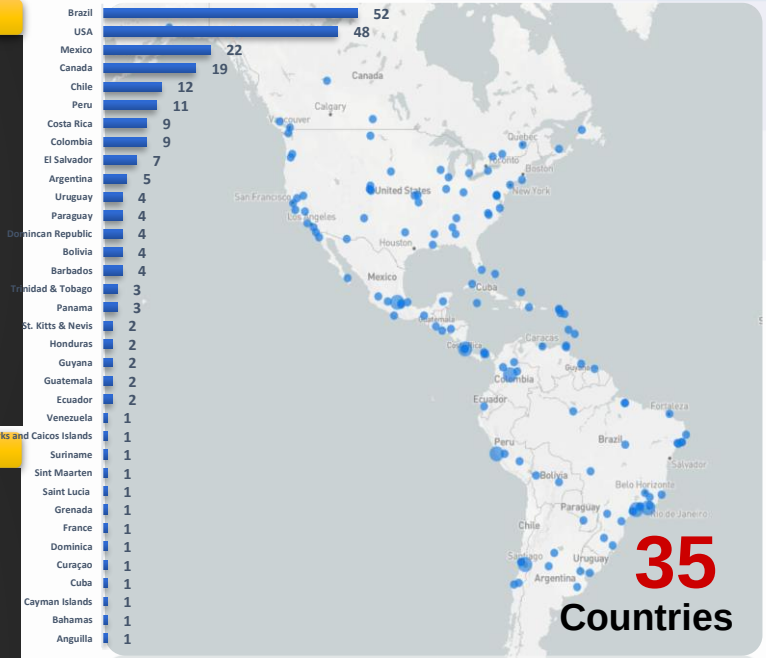
The Python Programming Language

- Freely Available
- Easy to Get Started
- Gradual Learning Curve
- (Very) Active Community
- Many (Many!) Libraries / Modules
- Multiple Input and Output Formats
- Works on Different Operating Systems
- Integration with Other Software / Applications

Use of Python by the Community







The results of the Regional Survey can be seen at the following link: Link

Why I Started Using Python

There was a need to process data from various satellites, from different providers, in many formats

Different Satellites

- GOES-Este / Oeste
- METEOSAT
- NOAA-18/19/20/21
- S-NPP
- GCOM-W1
- METOP

Different Data Providers

- NOAA
- CIMMS
- CIRA
- EUMETSAT
- NWS
- INPE
- MARN

Different Formats

- NetCDF4
- GeoTIFF
- GRIB2
- HDF
- BUFR
- HRIT
- SHAPEFILE
- TEXT

python™

Around 2017 (with the launch of GOES-16), I realized that all the tools I was using (separately) to process satellite data had some kind of interface with Python.

GDAL

Has a Python Interface ✓

GMT THE GENERIC MAPPING TOOLS

Has a Python Interface ✓

McIDAS V

Has a "Jython" Interface ✓

QGIS

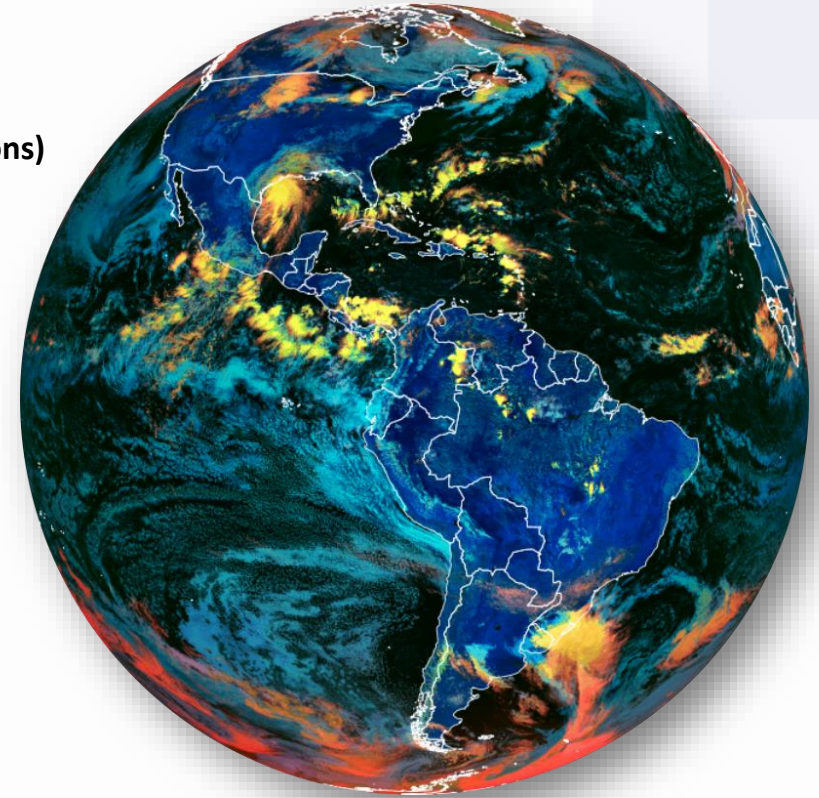
Has a Python Interface ✓

ILWIS

Has a Python Interface ✓

Examples of Python Libraries

- **GDAL:** Support various geospatial data formats
- **Matplotlib:** Graphs and data visualization in general
- **Numpy:** Supports arrays and multidimensional matrices (various functions)
- **NetCDF-4:** Python interface to the NetCDF4 format
- **PyGrib:** Python interface to the GRIB format
- **PyHDF:** Python interface to the HDF format
- **Cartopy:** Creating plots on maps / projections
- **PySpectral:** Spectral response functions and basic operations (AVHRR, VIIRS, MODIS, ABI, AHI y SEVIRI)
- **Satpy/Pytroll:** Data manipulation for various satellites (GOES-R, Himawari-8, MSG, Sentinel-1 ~3, NPP, Aqua / Terra, and others)
- **Metpy:** Creation of Skew-T, Meteograms and others)
- **PyOrbital:** Orbital and astronomical parameters



Examples: GOES-R Data Processing

- **NetCDF-4:**

- Read the NetCDF Radiances / Reflectances / Brightness Temperatures

- **GDAL:**

- Reproject from the GOES projection to others (e.g.: Equidistant Cylindrical / Mercator)

- Export image in GeoTIFF format (and others)

PyOrbital:

- Calculates the solar zenith angle, elevation and azimuth in relation to the satellite

- **PySpectral:**

- Calculation of the solar component of the 3.7 um band

- Rayleigh correction

- **Numpy:**

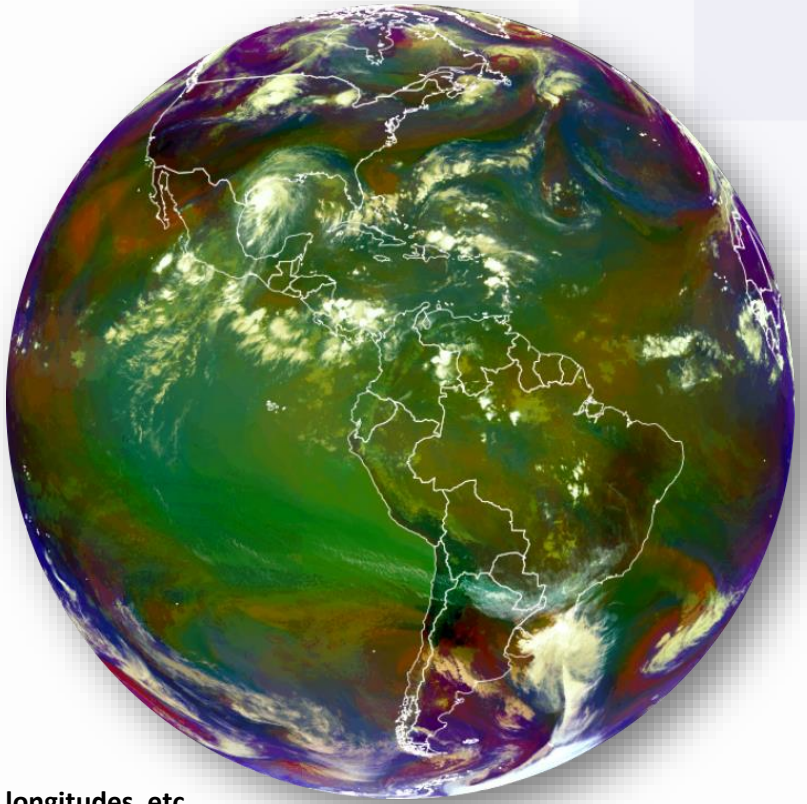
- Create a "stack" of bands to create RGBs

- Apply the necessary corrections

- Many other calculations

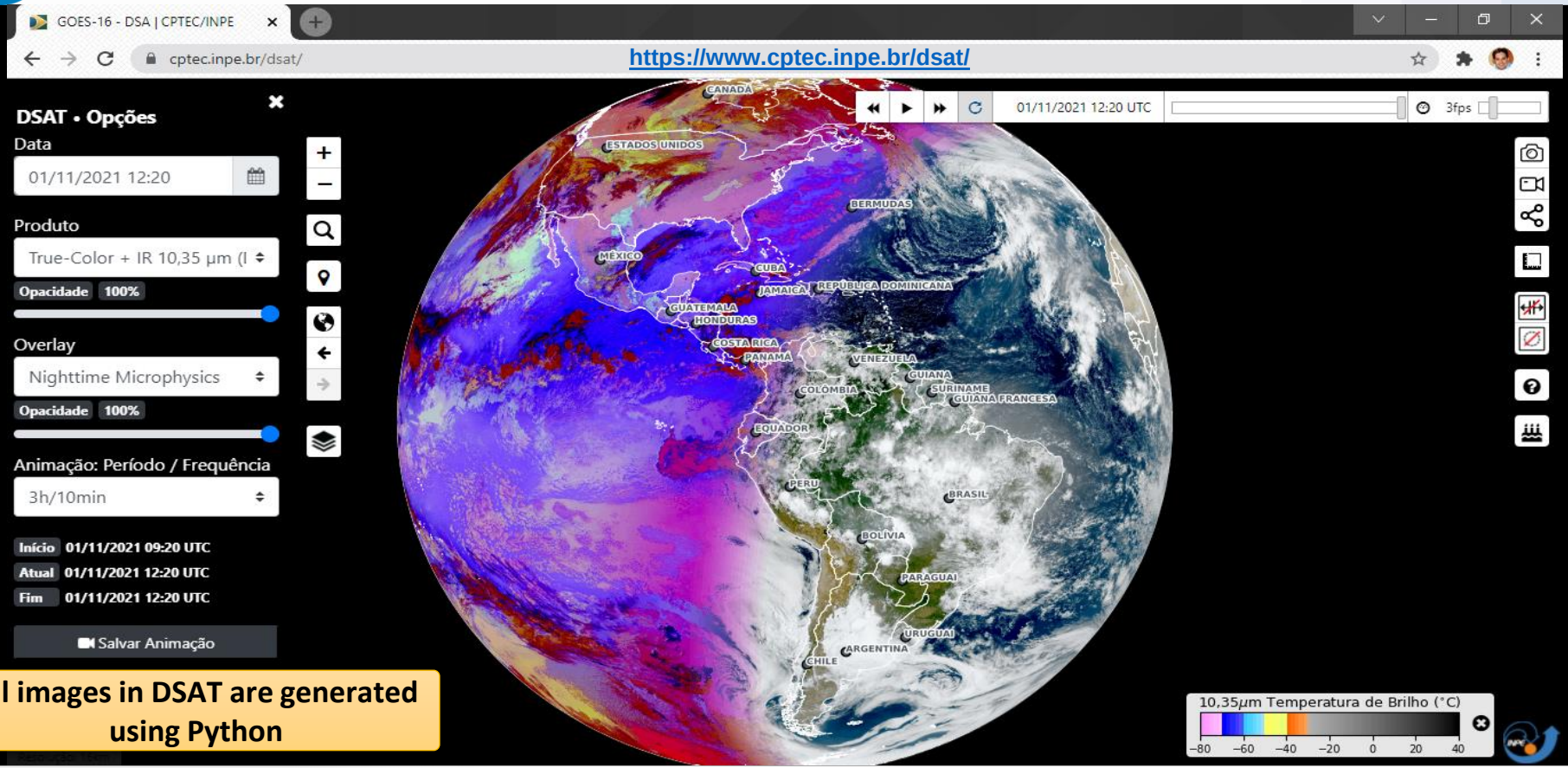
- **Cartopy:**

- Add maps, shapefiles (countries, states, continents, etc.), parallels, meridians, latitudes, longitudes, etc.



LET’S SEE SOME EXAMPLES

Application Examples: DSAT

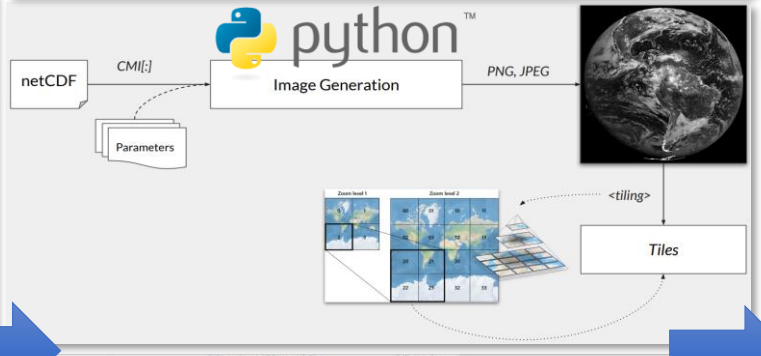


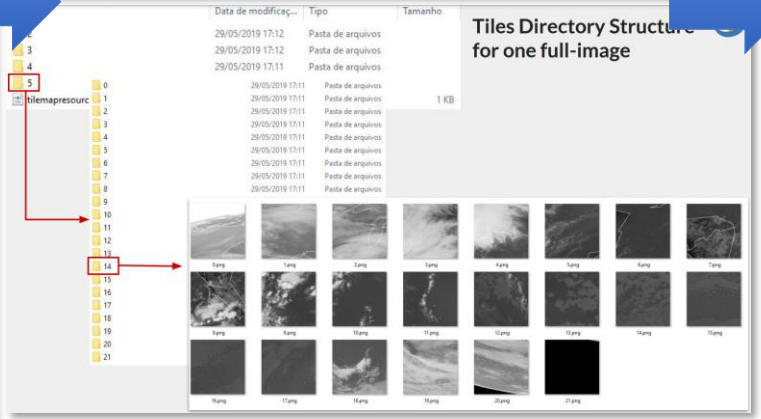
All images in DSAT are generated using Python

Application Examples: DSAT

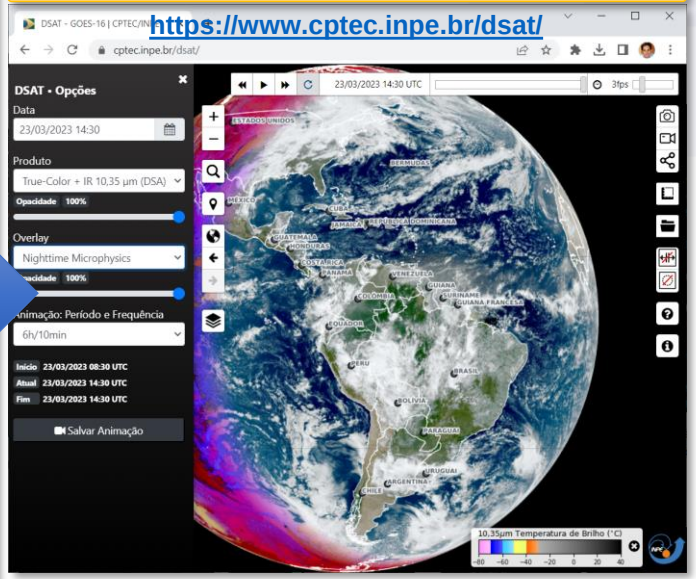
Data Reception

Data Processing



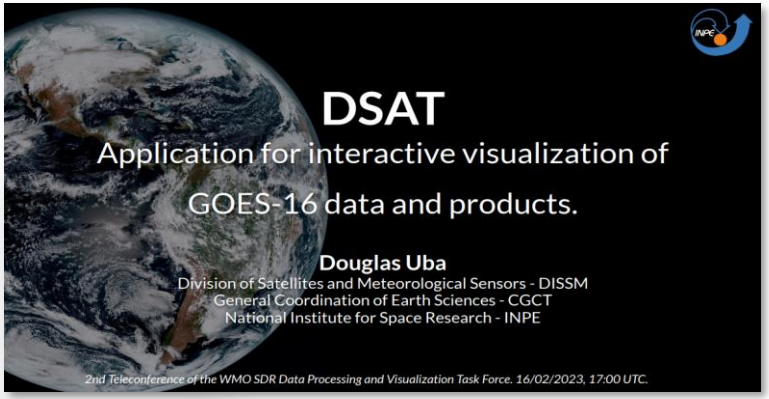


Visualization



More details about the development of DSAT in this presentation: Link

Application Examples: DSAT



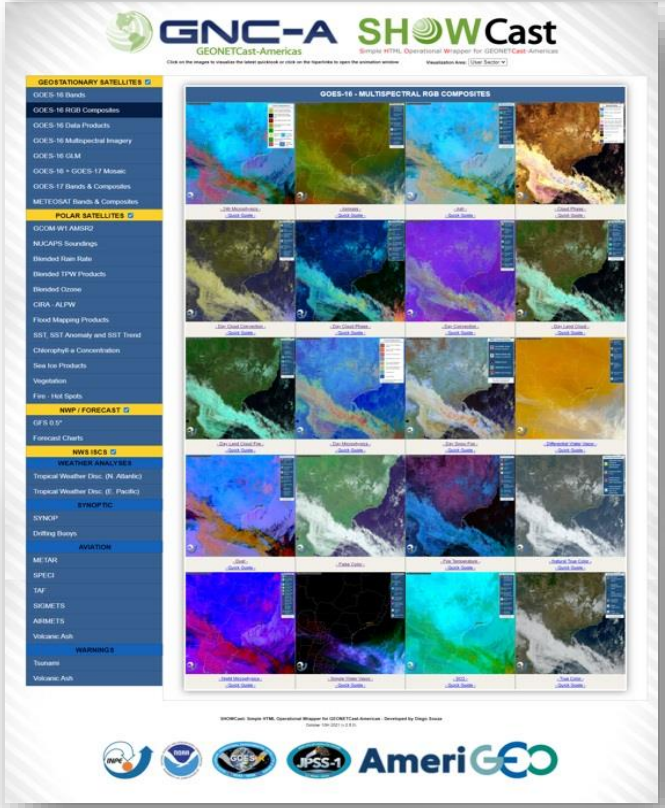
DSAT - Technologies (Summary)

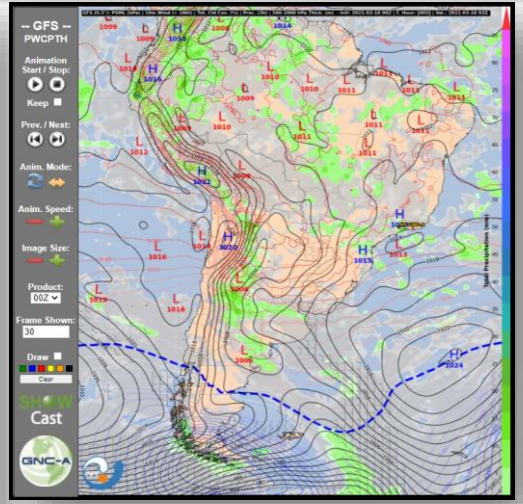
- **Front-end (Web App)**
  - HTML, Javascript and CSS
  - Leaflet
  - leaflet-rastercoords
  - Leaflet.TimeDimension
  - Other Leaflet's plugins
  - Bootstrap (for UI Style)
  - Tiles files are retrieved using simple HTTP protocol and grouped at viewing time by Leaflet, according to user interaction
  - JSON are retrieved in order to know some metadata about images; (for example, last date processed, available times, etc)
  - **It's simple, but efficient!**
- **Back-end (processing)**
  - Python + packages
  - GDAL, numpy, netCDF4 and matplotlib
  - gdal2tiles.py *("the rockstar!")*
  - Manages operational routines based on INPE's specific computational structure
  - No database is used
  - JSON files inform some metadata
  - Storage uses INPE's defined structure + OSGeo Tile Map Service
  - **It's simple, but efficient!**
  - ⚠️ Tiles are good, but:
    - Due the high temporal frequency of images, a lot (A LOT!) of tiles are generated daily.

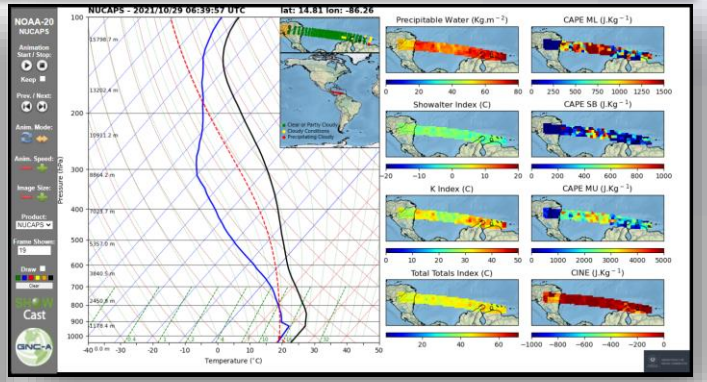
More details on the development of DSAT: https://sdr.ucr.ac.cr/data-processing-and-visualization

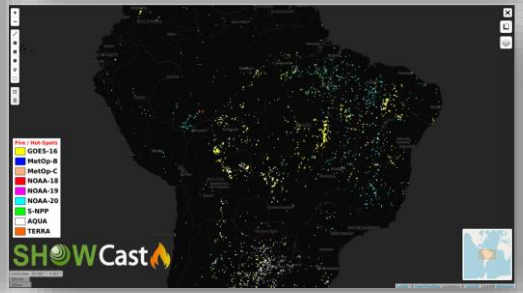
Application Examples: SHOWCast

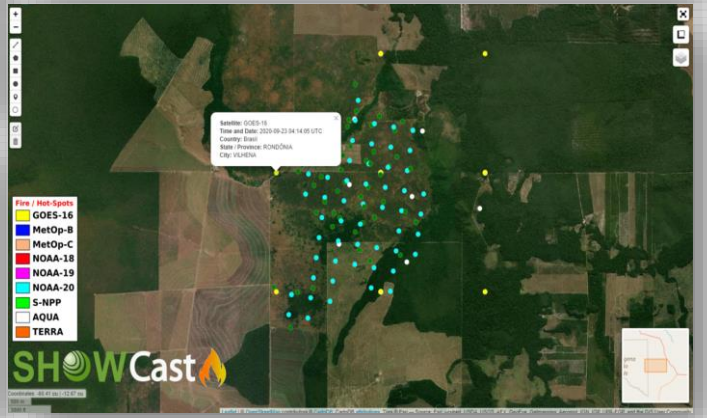
All SHOWCast products are processed with Python (Channels, RGBs, Models, Skew-T, Alerts, etc.)







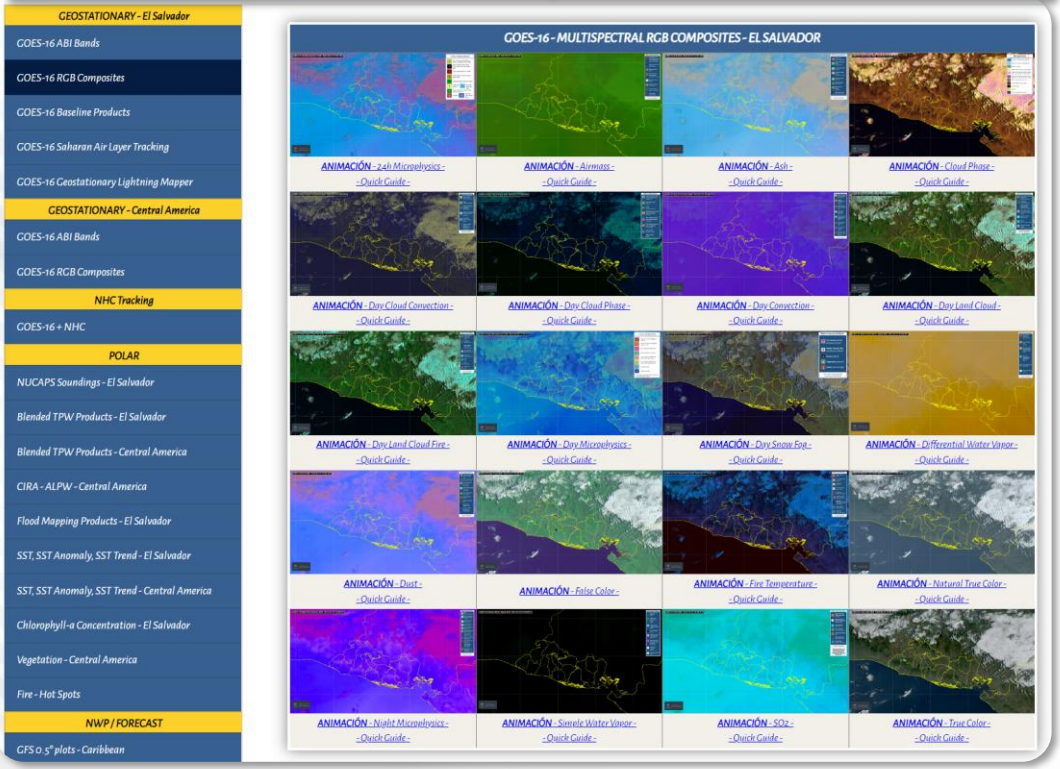




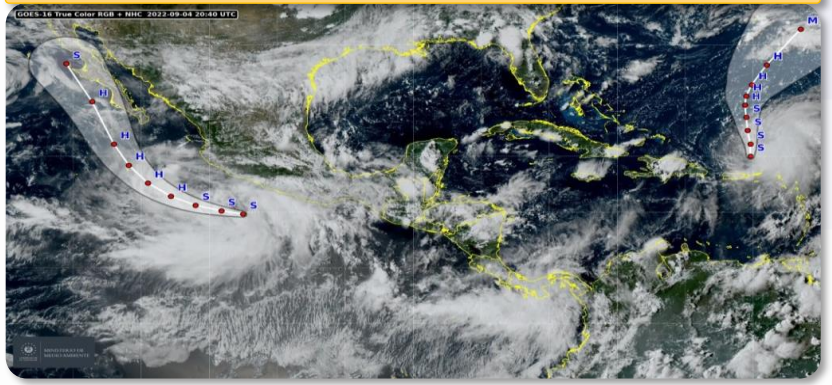
Application Examples: MARN

http://srt.marn.gob.sv/SHOWCast/HTML/SHOWCast_SEC.html

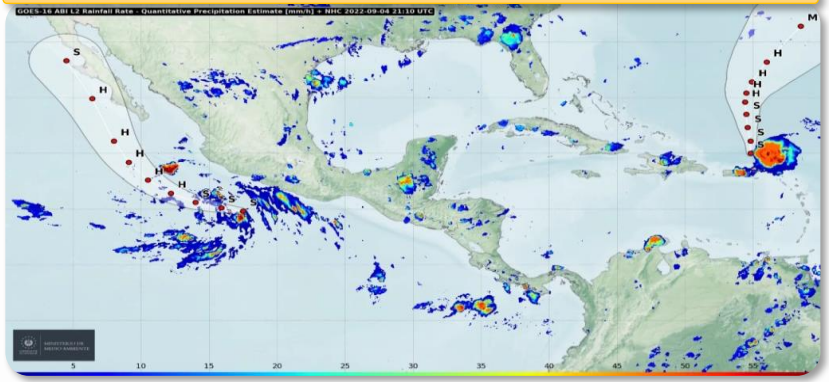
RGB Composites



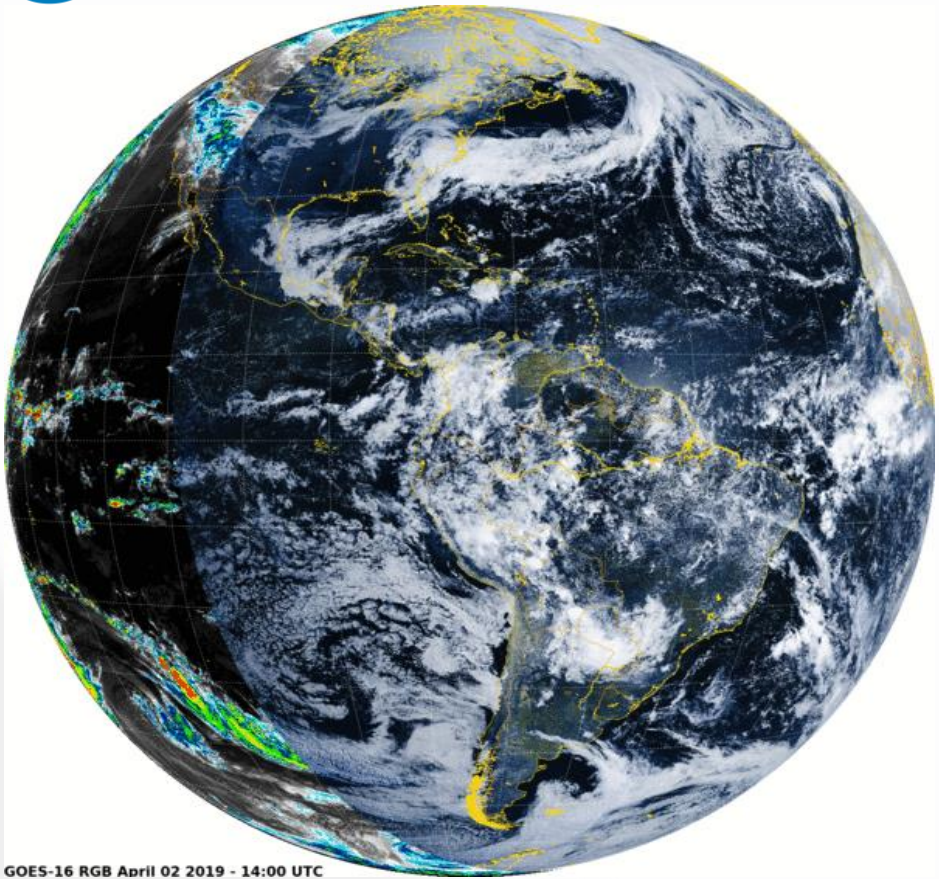
True Color RGB + Tracking NHC



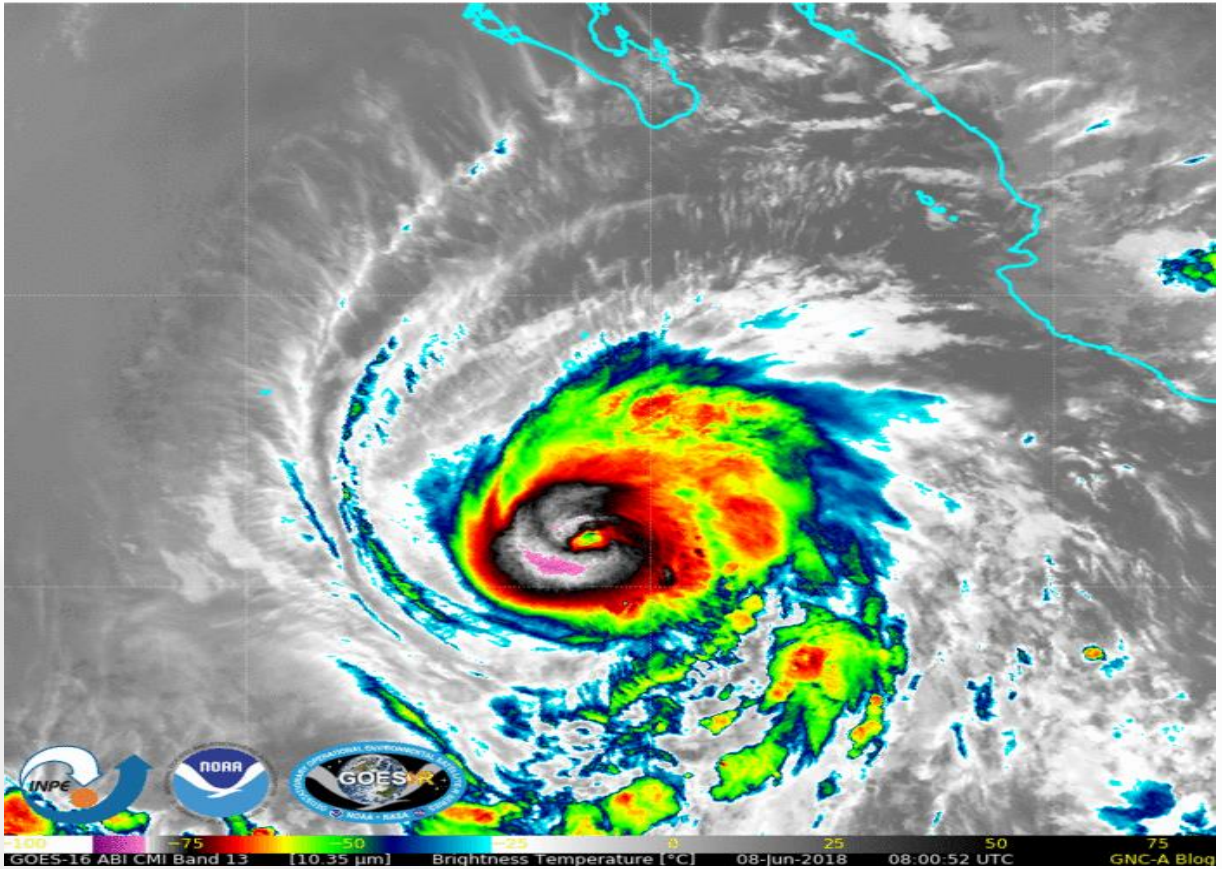
RRQPEF + Tracking NHC

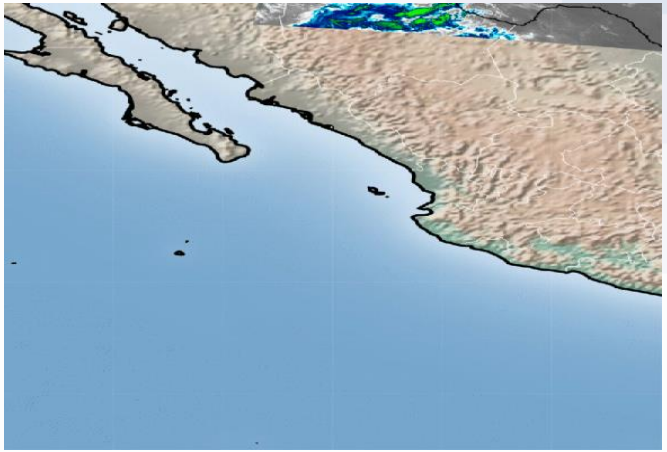


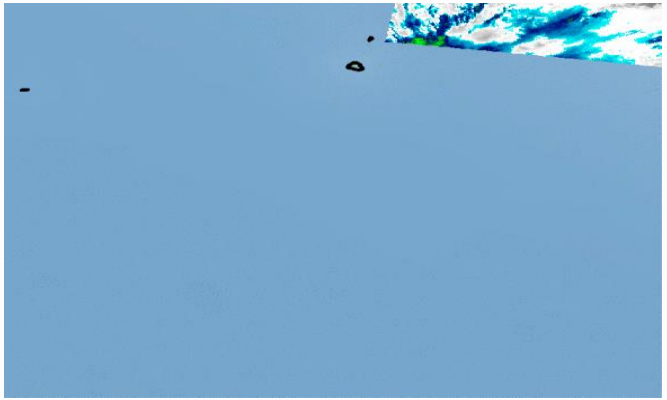
Application Examples: GEO and LEO



Application Examples: GEO and LEO

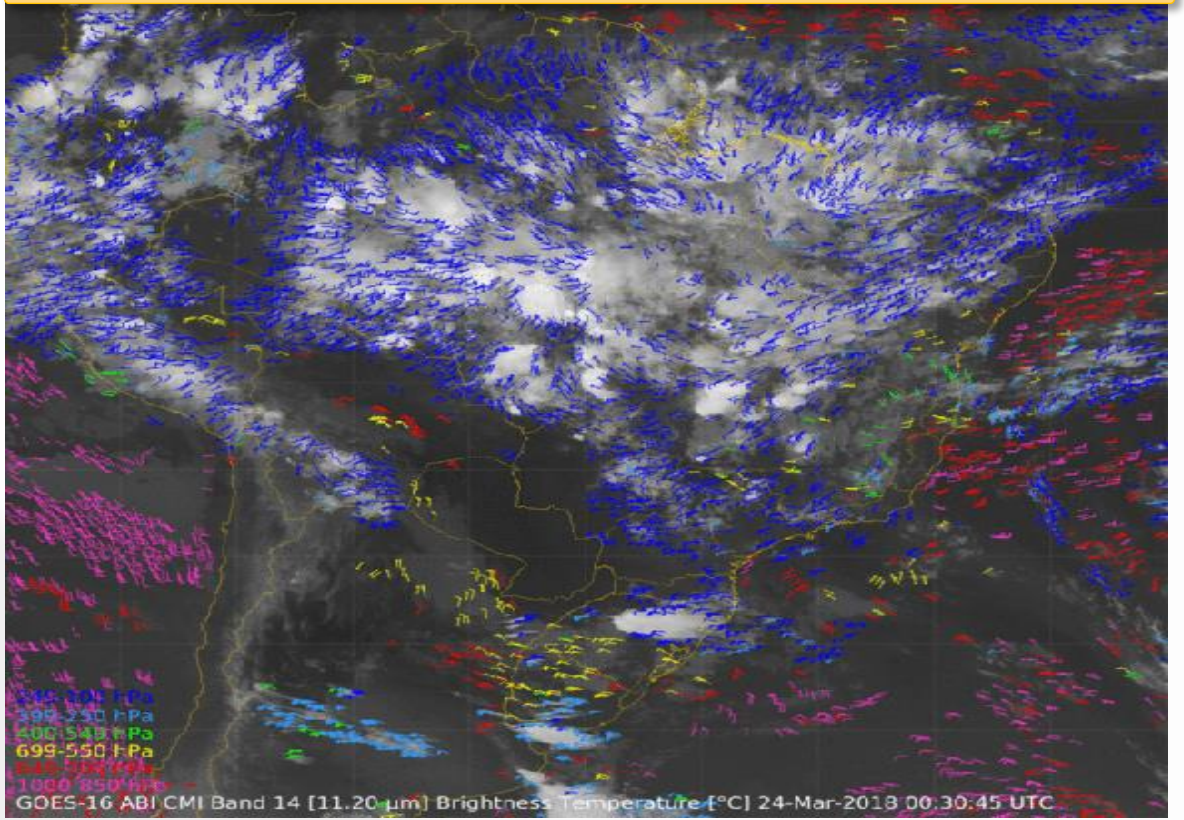






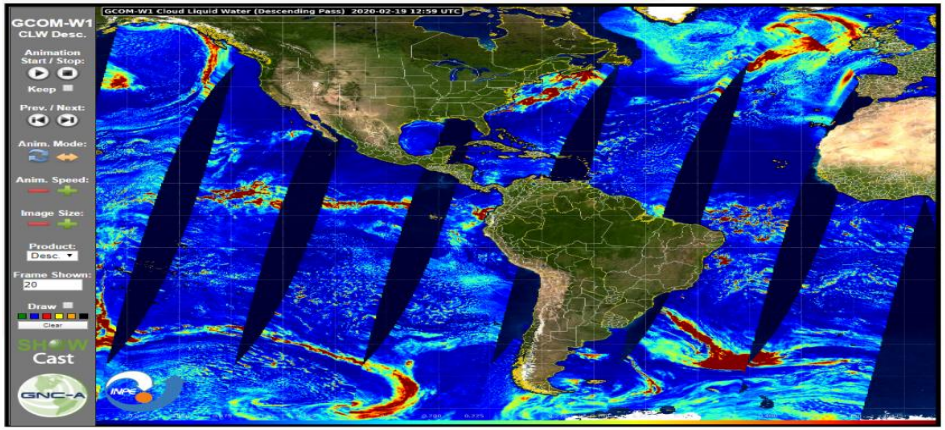
Application Examples: GEO and LEO

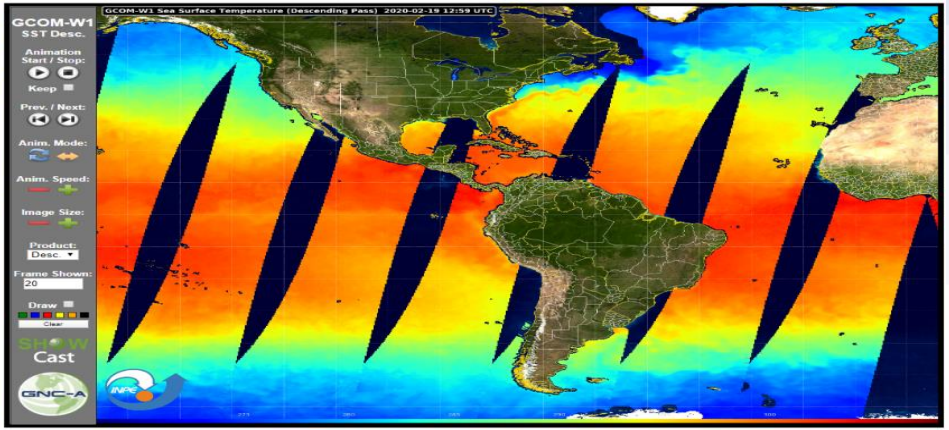
Vientos Derivados por Satélite (ABI - GOES-16)

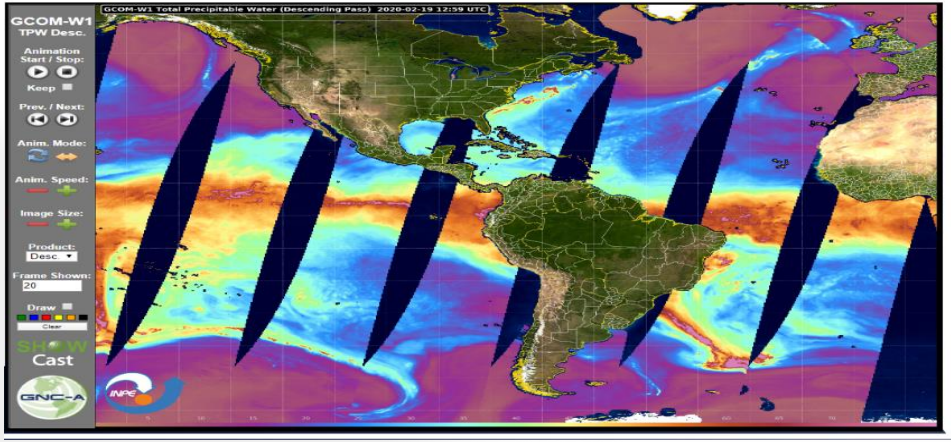


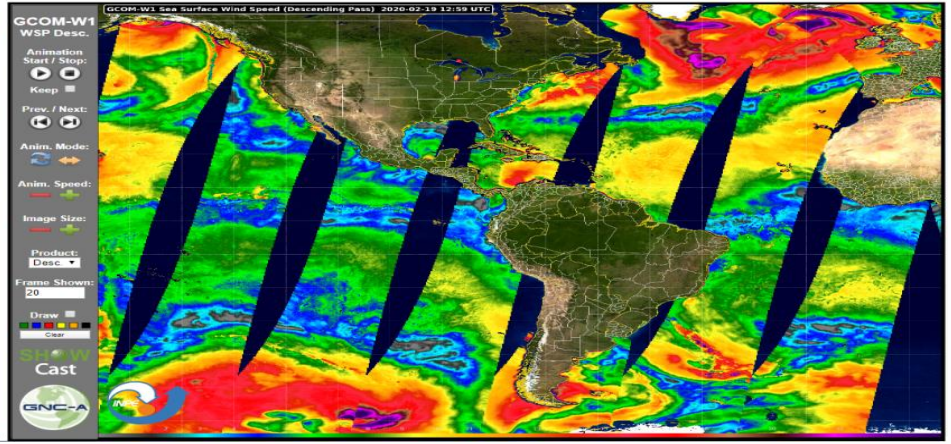
Banda Día Noche (VIIRS - NPP)

Application Examples: LEO (GCOM)

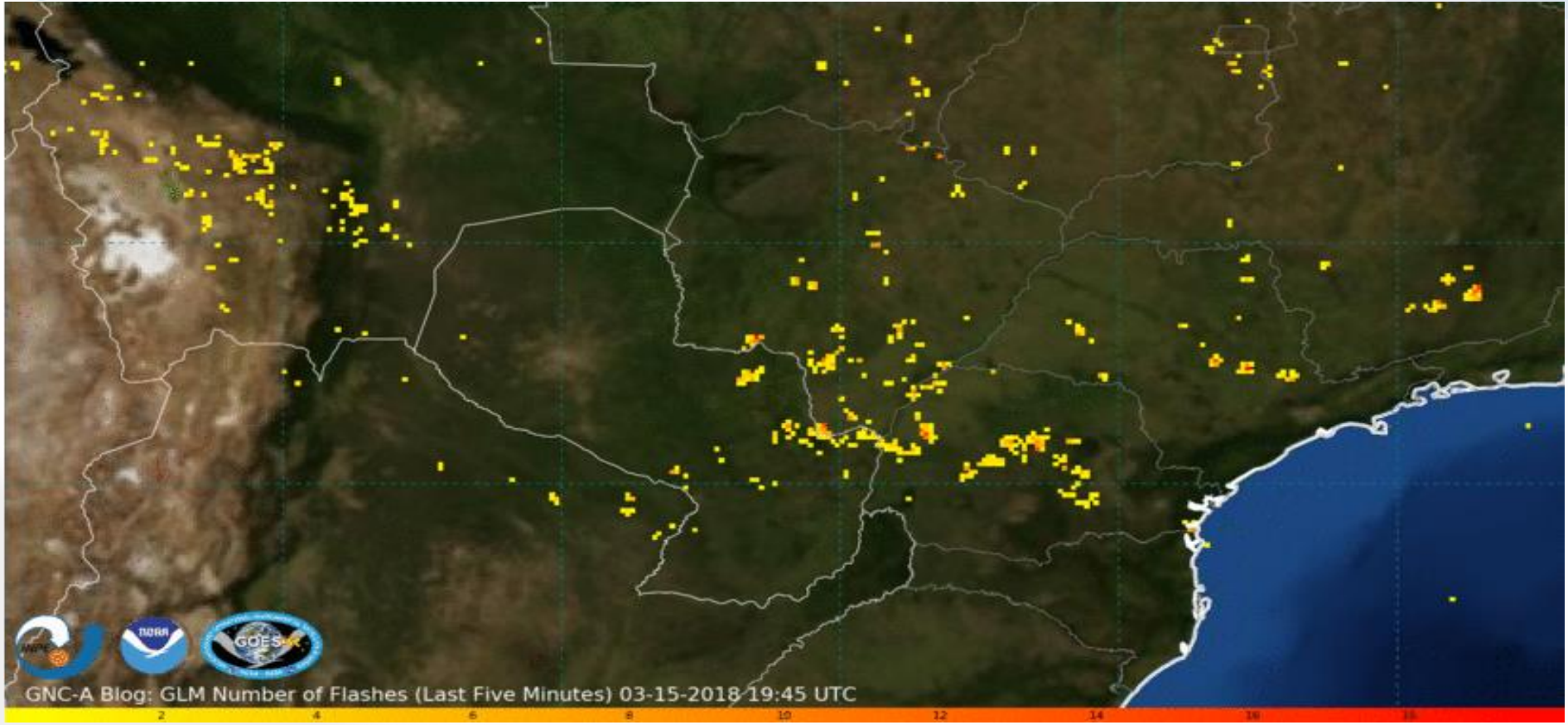






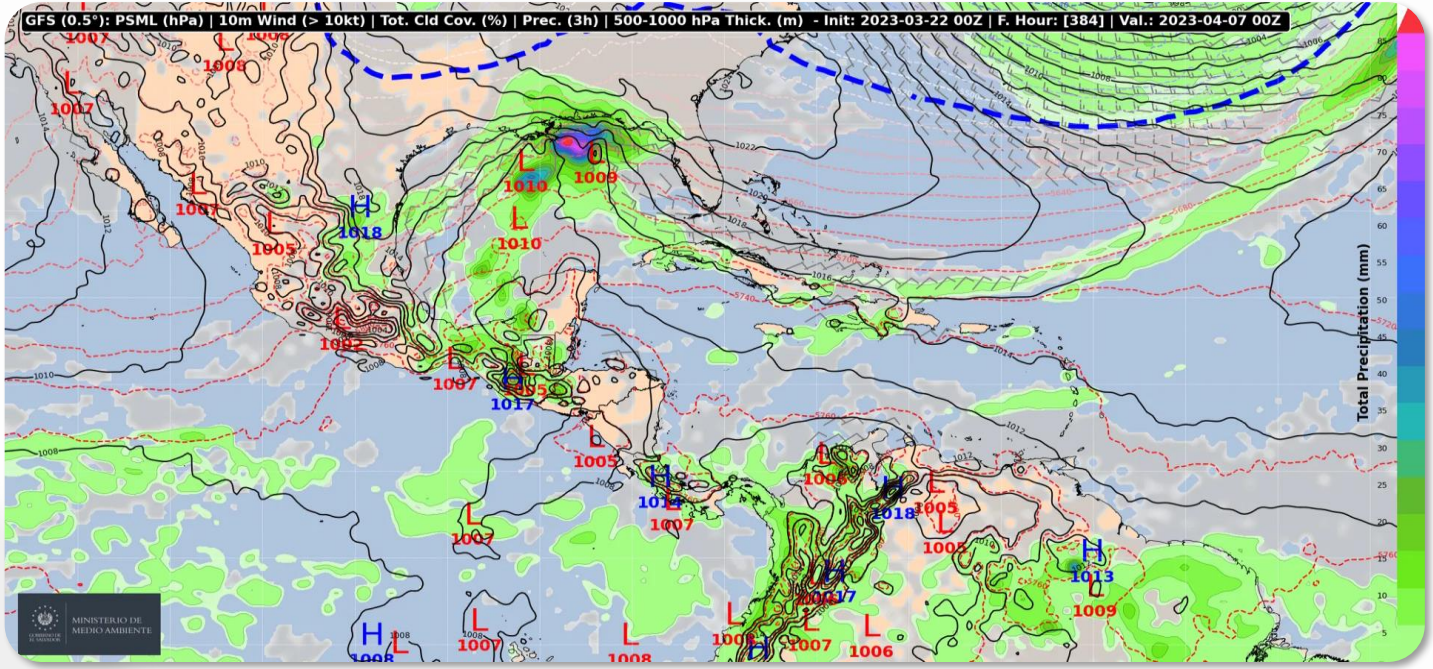


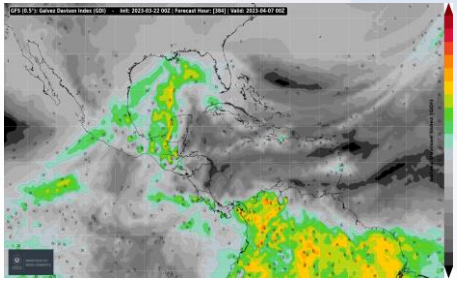
Application Examples: GLM

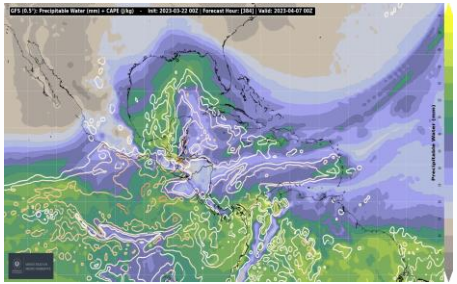


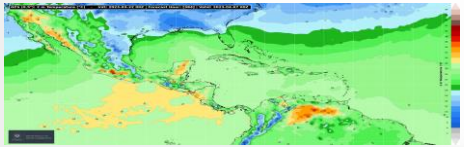
Application Examples: NWP

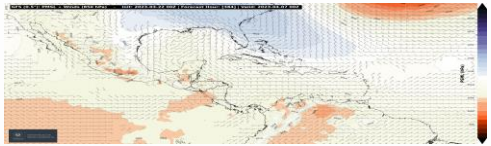
http://srt.marn.gob.sv/SHOWCast/HTML/SHOWCast_SEC.html

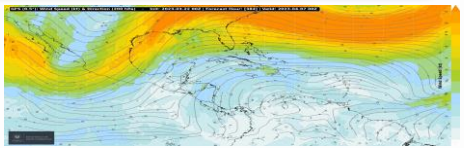






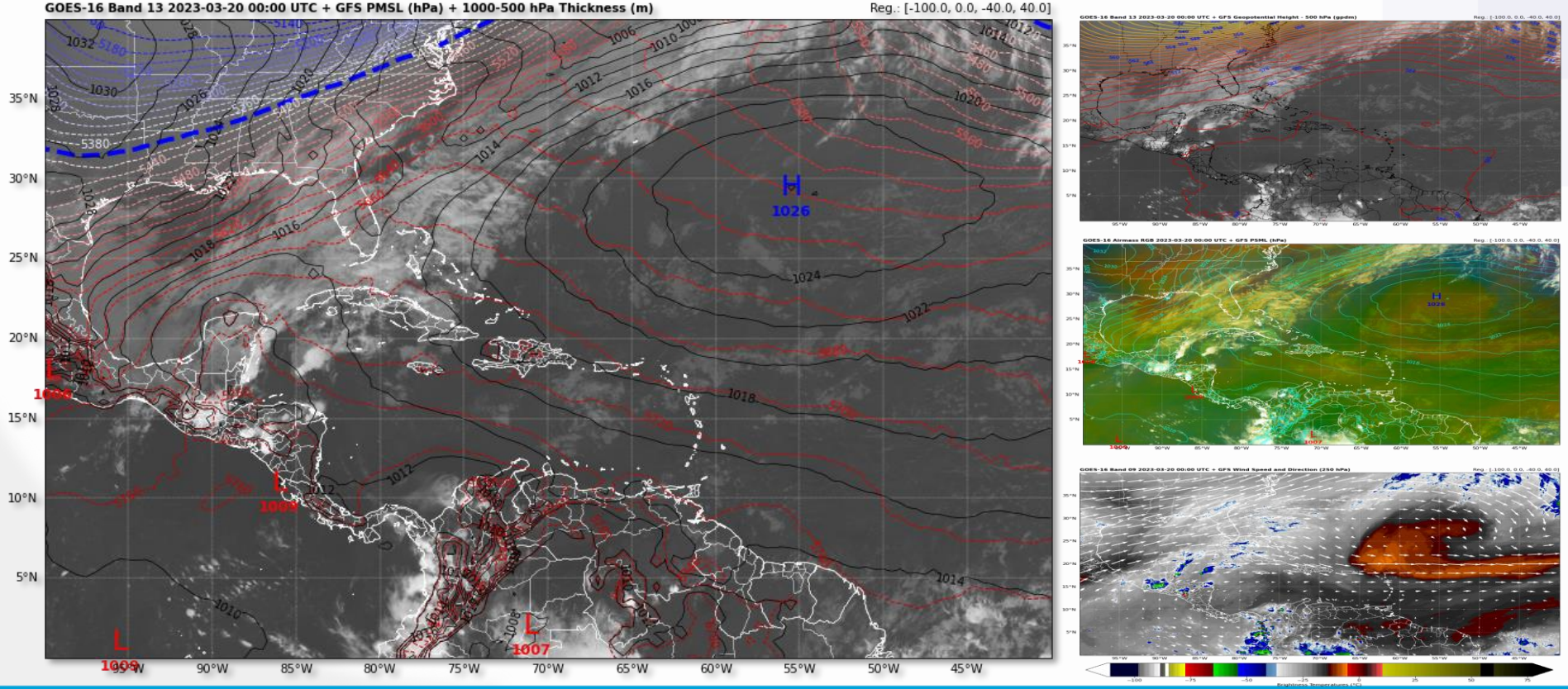






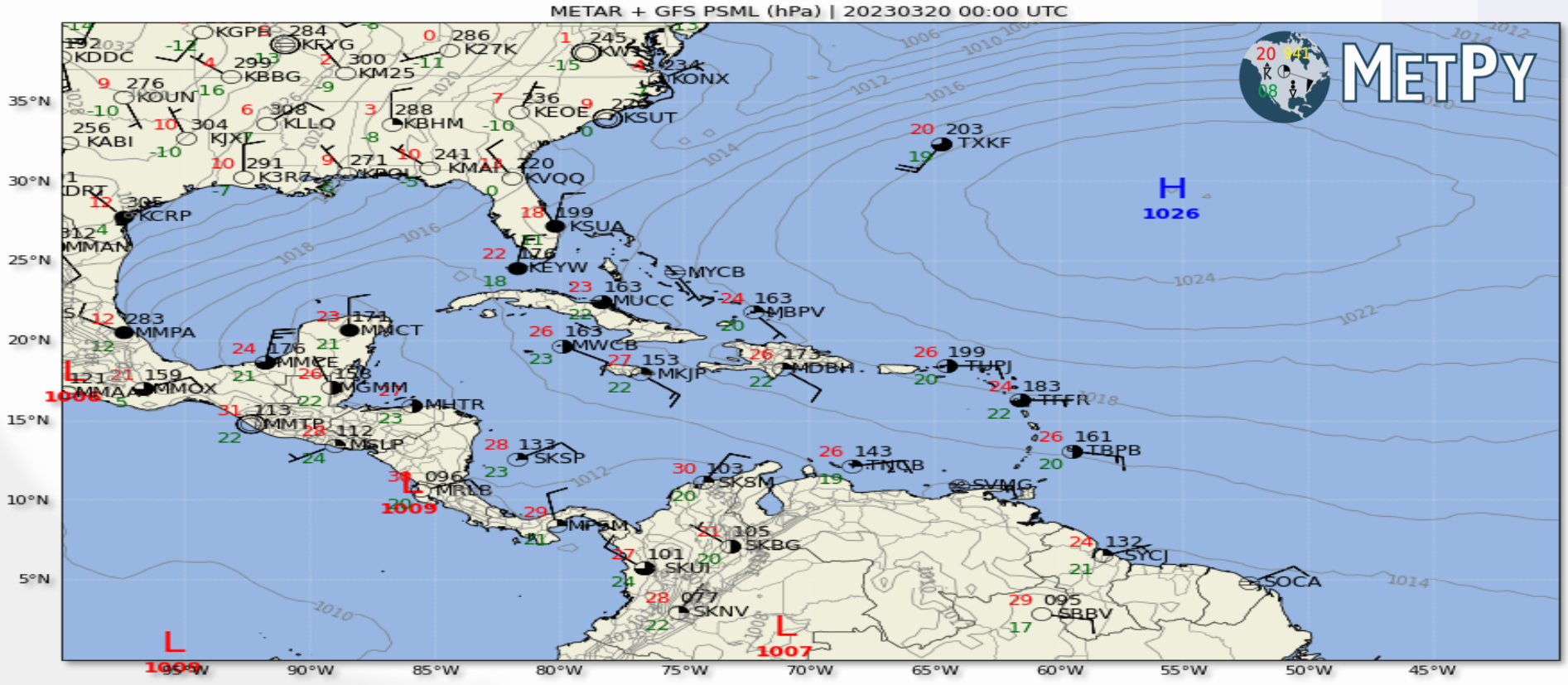
Application Examples: Satellite + NWP

https://geonetcast.wordpress.com/2021/07/27/vlab-processamento-de-dados-de-modelos-de-previsao-numerica-do-tempo-conteudo-completo/



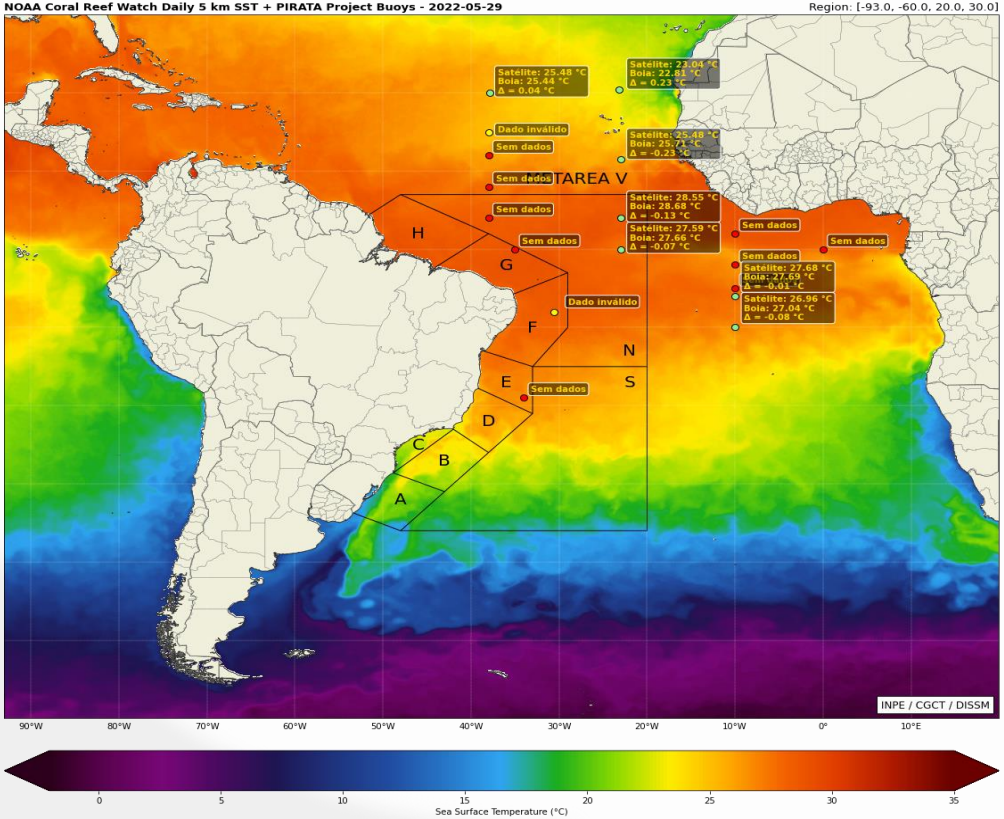
Application Examples: METAR

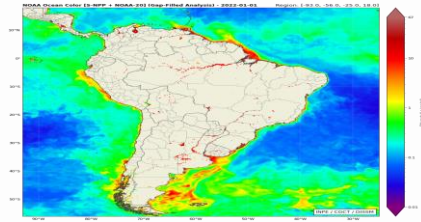
https://geonetcast.wordpress.com/2021/07/27/vlab-processamento-de-dados-de-modelos-de-previsao-numerica-do-tempo-conteudo-completo/

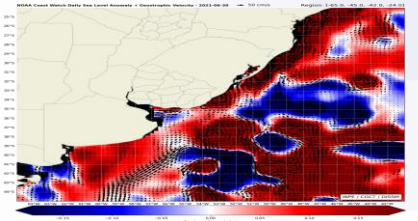


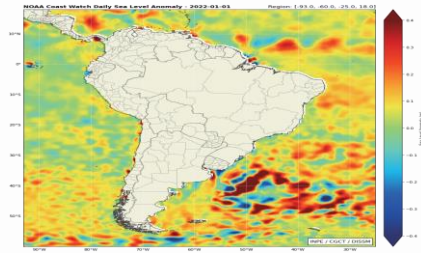
Application Examples: Ocean

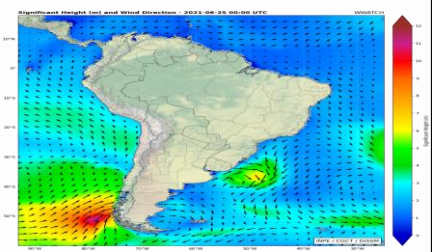
https://geonetcast.wordpress.com/2022/06/08/vlab-produtos-de-oceanografia-por-satelite-conteudo-completo/

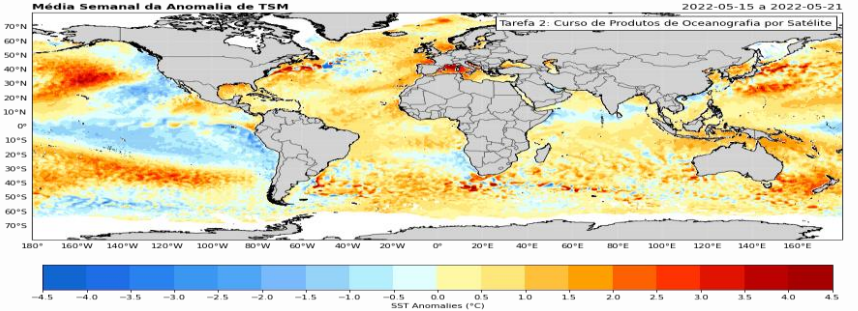






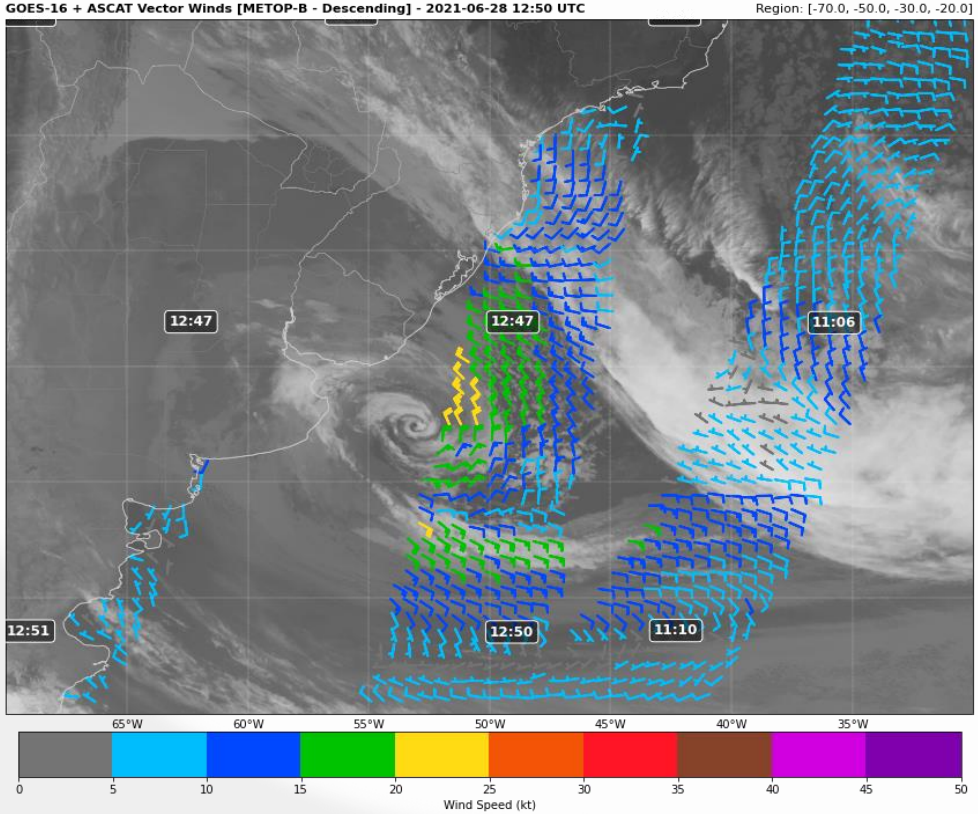


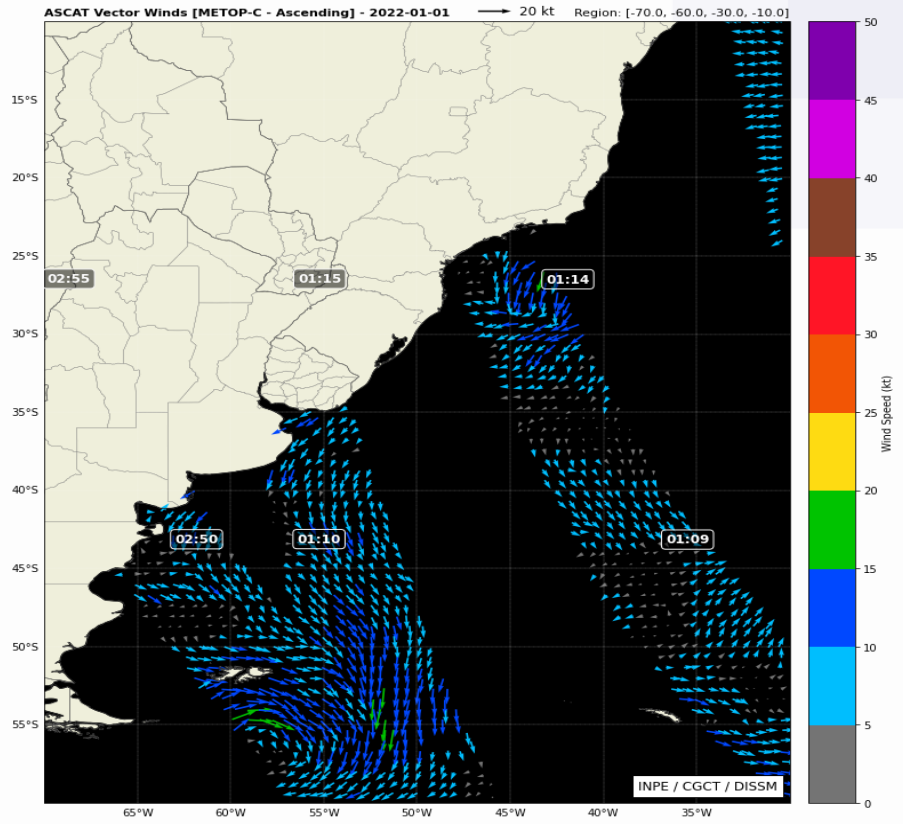




Application Examples: ASCAT Winds

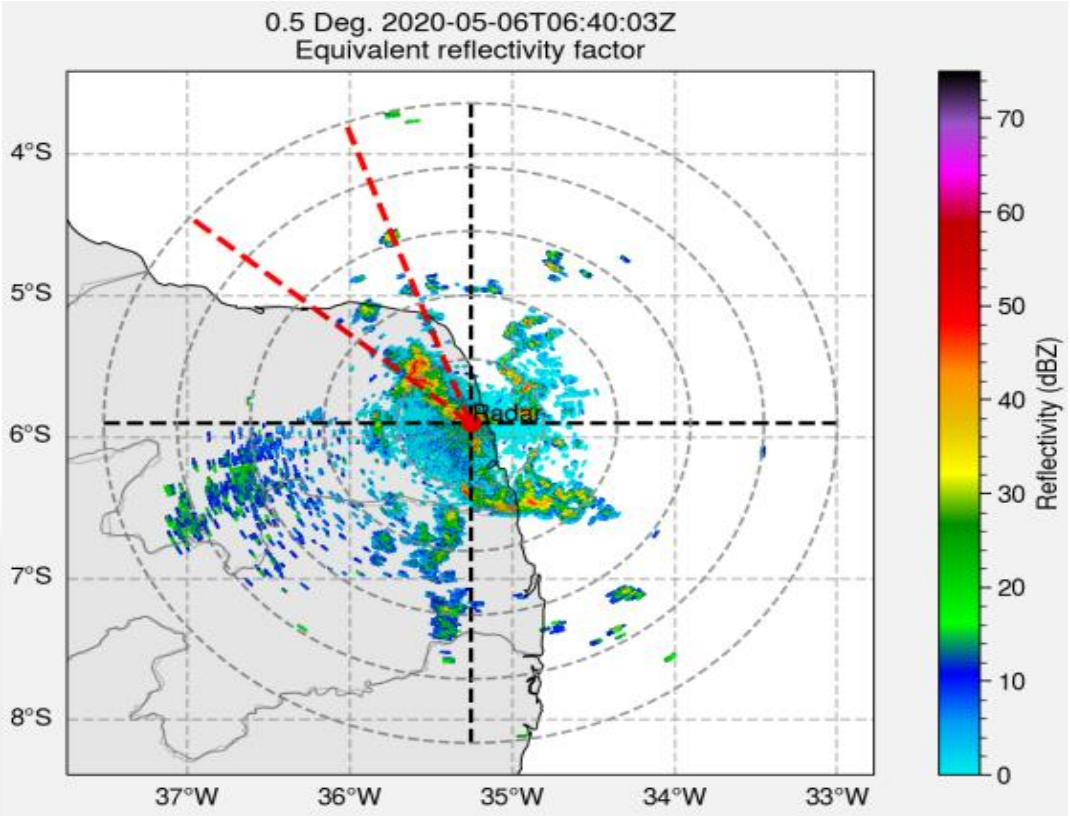
https://geonetcast.wordpress.com/2022/06/08/vlab-produtos-de-oceanografia-por-satelite-conteudo-completo/

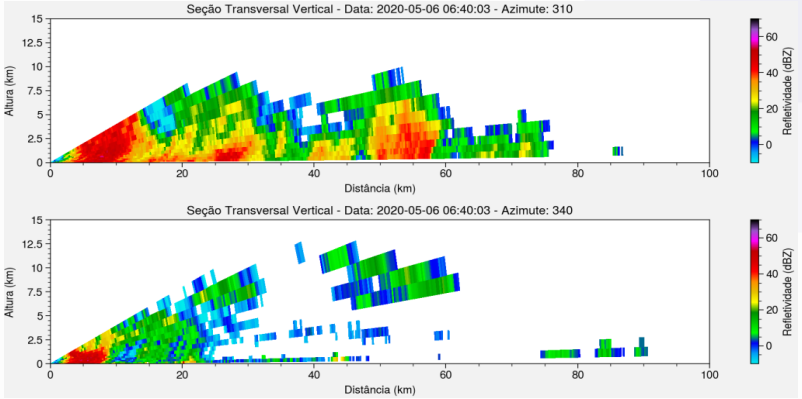


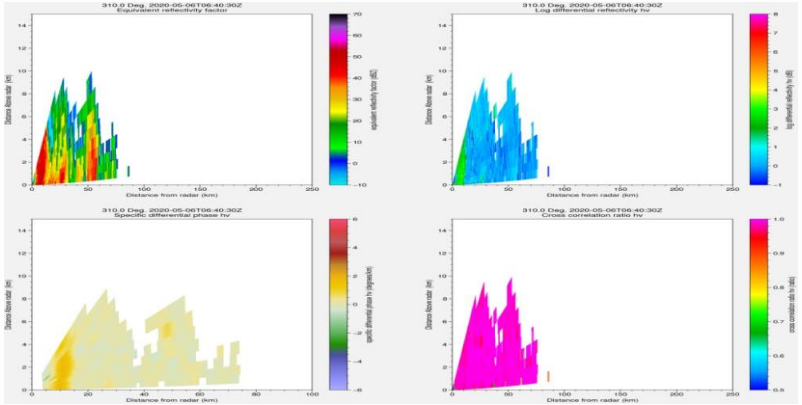


Application Examples: Weather Radar

https://geonetcast.wordpress.com/2023/01/26/new-vlab-training-weather-radar-fundamentals-and-data-processing-2023/





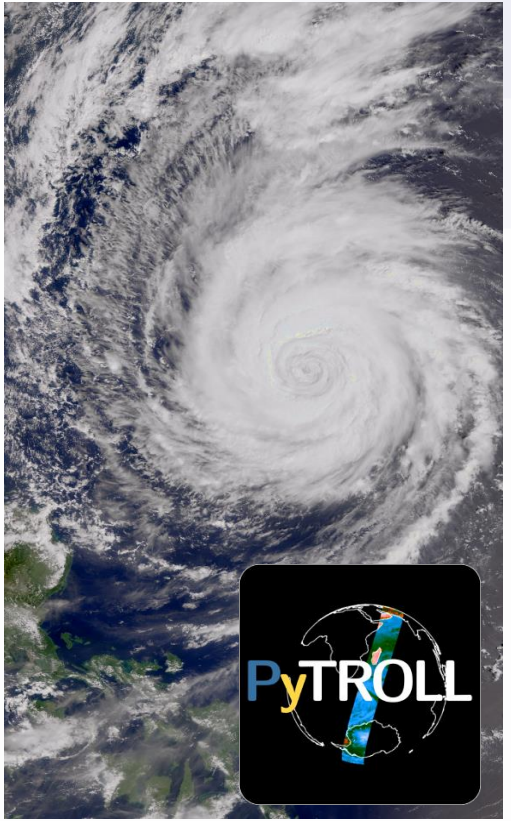


Application Examples: SENTINEL

https://geonetcast.wordpress.com/2018/09/26/processing-sentinel-2-data-with-python/

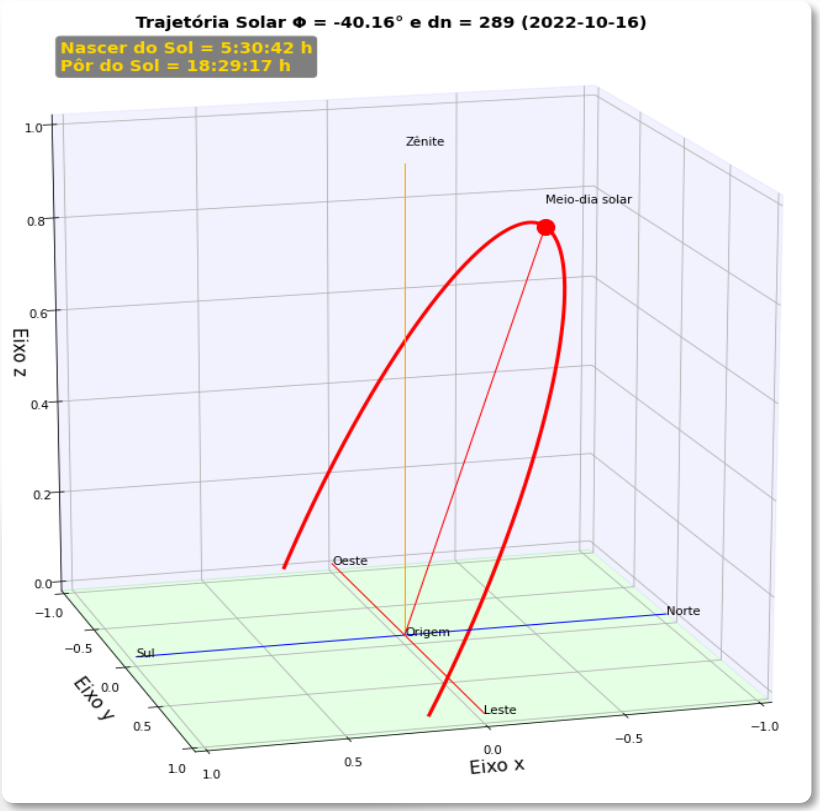
https://geonetcast.wordpress.com/2018/09/26/processing-sentinel-3-data-with-python/

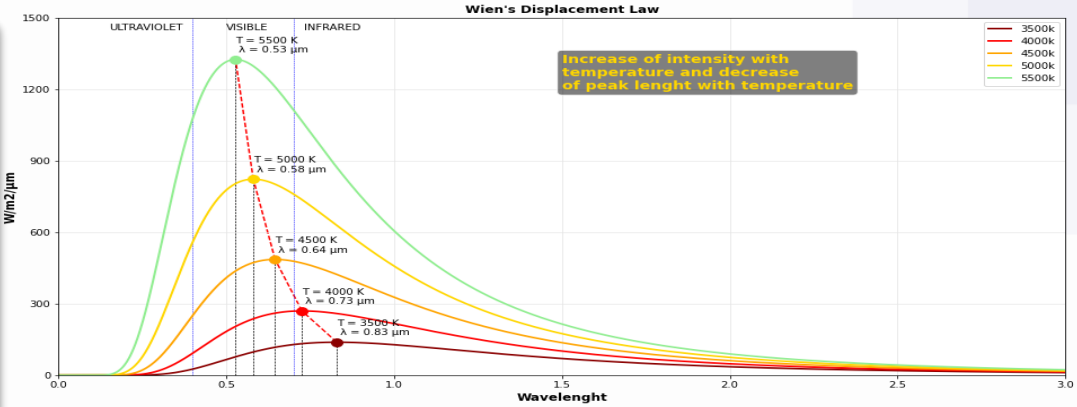

```python
from satpy.scene import Scene
from satpy import find_files_and_readers
from datetime import datetime

files = find_files_and_readers(base_dir="C:\\MSI",
                               reader='safe_msi')

scn = Scene(filenames=files)
scn.load(['true_color'])
scn.save_dataset('true_color', filename='true_color_S2_gnc_tutorial'+'.png')
```

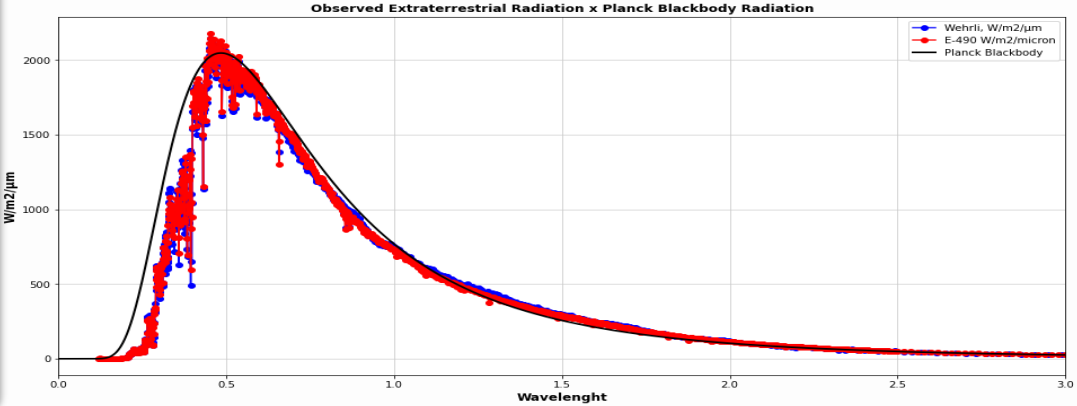


Application Examples: Graphs in General

https://geonetcast.wordpress.com/2022/01/27/new-vlab-training-this-year-coe-brazil-radiation-products-concepts-data-access-and-processing/







It is Also Possible to Access Data

In this example, a Python script is used to download GOES-R L1b data from UNIDATA THREDDS and the L1b data radiances can also be processed with Python (examples in the image).

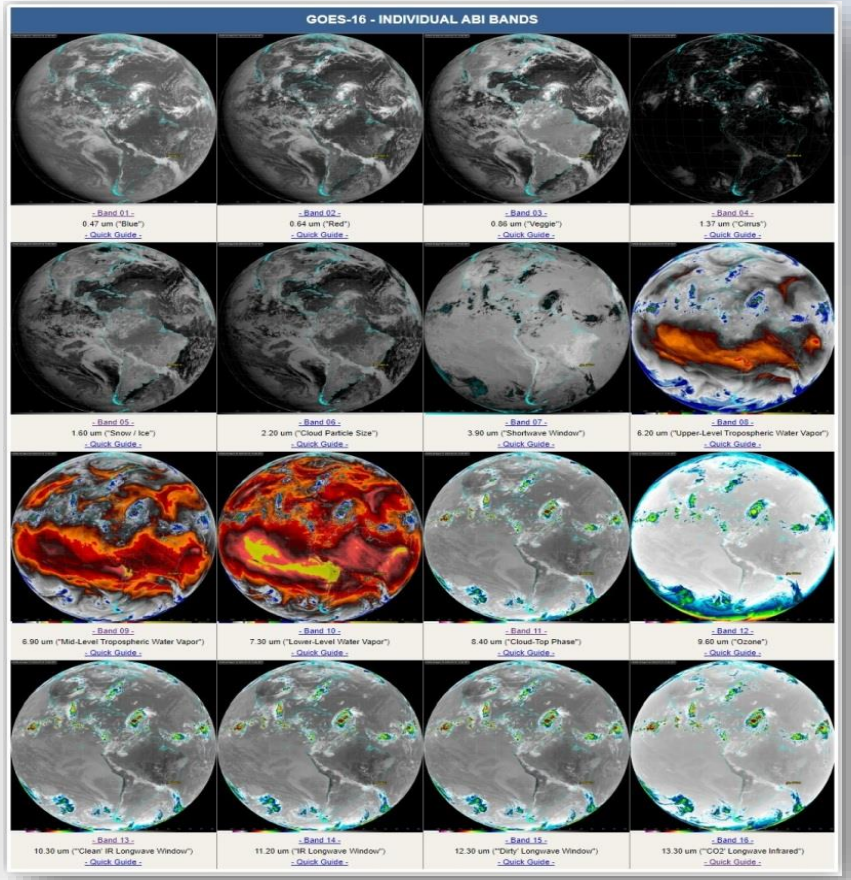
```
from siphon.catalog import TDSCatalog    # Code to support reading and
import urllib.request                    # Defines functions and classe
```

```
# Unidata THREDDS Data Server Catalog URL
base_cat_url = 'https://thredds-test.unidata.ucar.edu/thredds/catalog/satellite/{
```

```
# Select the desired bucket, in this case, the satellite(s)
SATELLITES = ['goes16'] # Choose from ['goes16', 'goes17'] or both
PLATFORM = ['GRB16'] # Choose from ['GRB16', 'GRB17'] or both
```

```
# Catalog URL
cat_url = base_cat_url.format(satellite = satellite, platform = platform, c
print(cat_url)
# Access the catalog
cat = TDSCatalog(cat_url)
# Get the latest dataset available
ds = cat.datasets[-1]
```

```
# Get the URL
url = ds.access_urls['HTTPServer']
# Download the most recent file for this particular hour
urllib.request.urlretrieve(url, OUTDIR + str(ds))
```



It is Also Possible to Access Data

In this example, a python script is used to download GOES-R L2 data from Amazon Web Services

```python
from netCDF4 import Dataset           # Read / Write NetCDF4 files
import matplotlib.pyplot as plt       # Plotting library
from datetime import datetime         # Basic Dates and time types
import cartopy, cartopy.crs as ccrs   # Plot maps
import os                             # Miscellaneous operating system interfaces
import boto3                          # Amazon Web Services (AWS) SDK for Python
from botocore import UNSIGNED         # boto3 config
from botocore.config import Config    # boto3 config
```

LIBRARIES

```python
# Input and output directories
input = "Samples"; os.makedirs(input, exist_ok=True)
output = "Output"; os.makedirs(output, exist_ok=True)
```

INPUT AND OUTPUT DIRECTORIES

```python
# Amazon repository information
# https://noaa-goes16.s3.amazonaws.com/index.html
bucket_name = 'noaa-goes16'
product_name = 'ABI-L2-CMIPF'
year = 2021
day_of_year = 37
hour = 17
min = 00
band = 1

# Initializes the S3 client
s3_client = boto3.client('s3', config=Config(signature_version=UNSIGNED))
#-------------------------------------------------------------------------------
# File structure
prefix = f'{product_name}/{year}/{day_of_year:03.0f}/{hour:02.0f}/OR_{product_name}-M6C{band:02.0f}_G16_s{year}{day_of_year:03.0f}{hour:02.0f}{min:02.0f}'

# Seach for the file on the server
s3_result = s3_client.list_objects_v2(Bucket=bucket_name, Prefix=prefix, Delimiter = "/")
#-------------------------------------------------------------------------------
# Check if there are files available
if 'Contents' not in s3_result:
    # There are no files
    print("No files found for the date: ",year,day_of_year)
    quit()
else:
    # There are files
    for obj in s3_result['Contents']:
        key = obj['Key']
        # Print the file name
        print(key)
```

amazon web services

DESIRED DATA AND SEARCH IN THE AWS "BUCKET"

https://noaa-goes16.s3.amazonaws.com/index.html

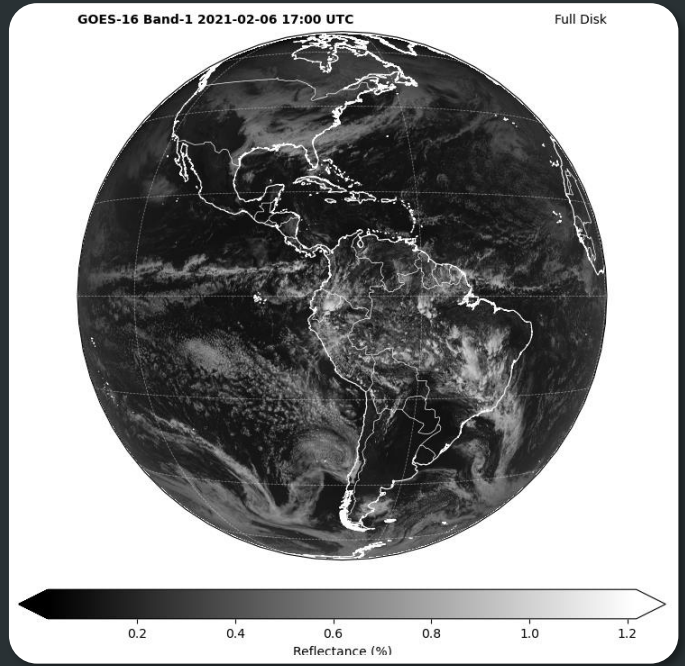
CHECK IF THE DESIRED DATA IS AVAILABLE

It is Also Possible to Access Data

```
if 'Contents' not in s3_result:
    # There are no files
    print("No files found for the date: ",year,day_of_year)
    quit()
else:
    # There are files
    for obj in s3_result['Contents']:
        key = obj['Key']
        # Print the file name
        print(key)
        file_name = key.split('/')[-1].split('.')[0]

        # Download the file
        if not os.path.exists(f'{input}/{file_name}.nc'):
            s3_client.download_file(bucket_name, key, f'{input}/{file_name}.nc')

        # If the file exists
        if (os.path.exists(f'{input}/{file_name}.nc')):

            # Open the GOES-R image
            file = Dataset(f'{input}/{file_name}.nc')

            # Get the pixel values
            data = file.variables['CMI'][:]
            #---------------------------------------------------------------------------------------------------------
            # Choose the plot size (width x height, in inches)
            plt.figure(figsize=(10,10))

            # Use the Geostationary projection in cartopy
            ax = plt.axes(projection=ccrs.Geostationary(central_longitude=-75.0, satellite_height=35786023.0))
            img_extent = (-5434894.67527,5434894.67527,-5434894.67527,5434894.67527)

            # Add coastlines, borders and gridlines
            ax.coastlines(resolution='10m', color='white', linewidth=0.8)
            ax.add_feature(cartopy.feature.BORDERS, edgecolor='white', linewidth=0.5)
            ax.gridlines(color='white', alpha=0.5, linestyle='--', linewidth=0.5)

            # Define the color scale based on the channel
            if band <= 6:
                colormap = "gray"   # Black to white for visible channels
                prodname = "Reflectance (%)"
            else:
                colormap = "gray_r" # White to black for IR channels
                prodname = "Brightness Temperatures (°C)"

            # Plot the image
            img = ax.imshow(data, origin='upper', extent=img_extent, cmap=colormap)

            # Add a colorbar
            plt.colorbar(img, label=prodname, extend='both', orientation='horizontal', pad=0.05, fraction=0.05)

            # Extract the date
            date = (datetime.strptime(file.time_coverage_start, '%Y-%m-%dT%H:%M:%S.%fZ'))

            # Add a title
            plt.title('GOES-16 Band-' + str(band) + ' ' + date.strftime('%Y-%m-%d %H:%M') + ' UTC', fontweight='bold', fontsize=10, loc='left')
            plt.title('Full Disk', fontsize=10, loc='right')
            #---------------------------------------------------------------------------------------------------------
            # Save the image
            plt.savefig(f'{output}/{file_name}.png')

            # Show the image
            plt.show()
```

DATA NOT AVAILABLE: END THE SCRIPT

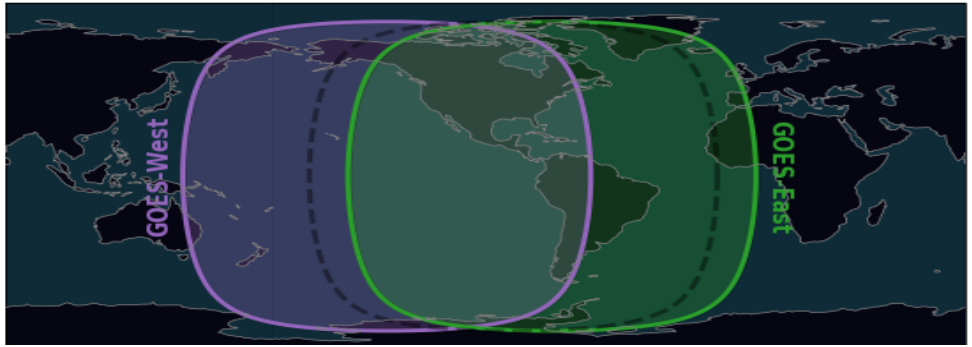
DATA AVAILABLE: PROCESS THE DATA



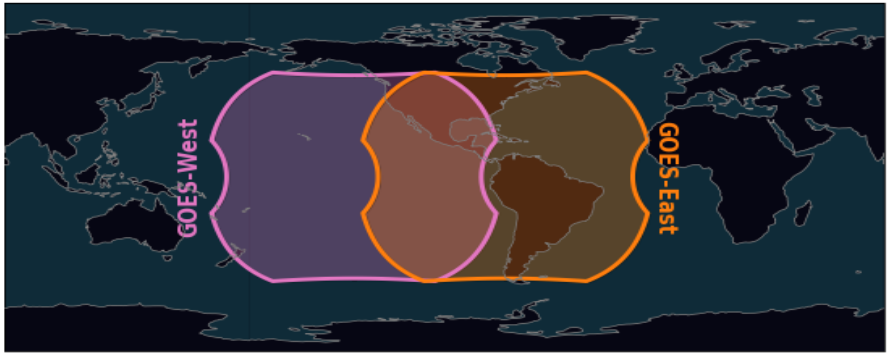
Everything is Possible

Even plot the coverage of the satellite sensors using Python =-)

GOES ABI Field of View



GOES GLM Field of View



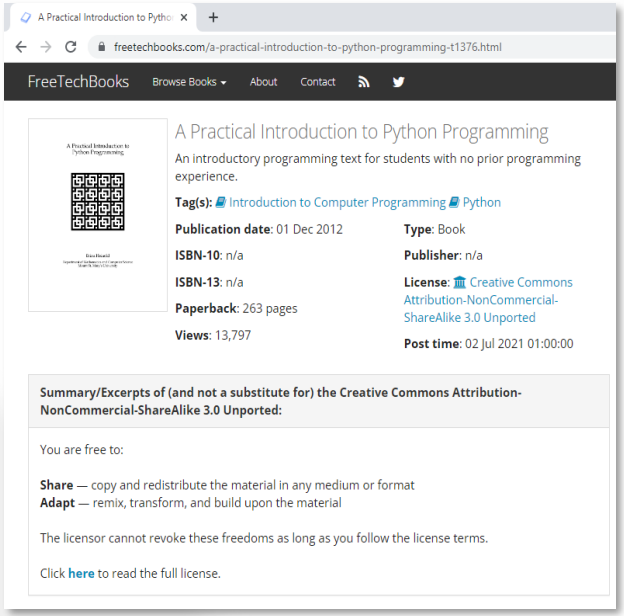
GOES-West ABI Field of View and Mesoscale Domains

https://github.com/blaylockbk/goes2go/tree/main/docs/user_guide/notebooks

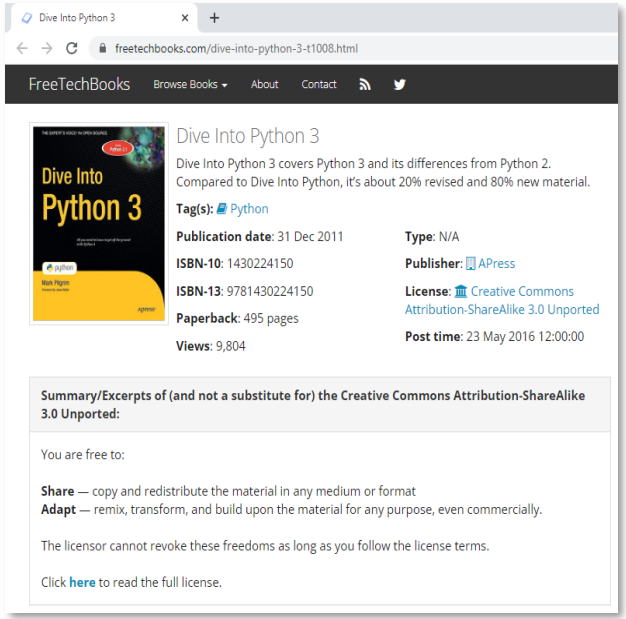
Some Useful Resources (Basic)

Practical introduction to Python from scratch: There are many materials freely available

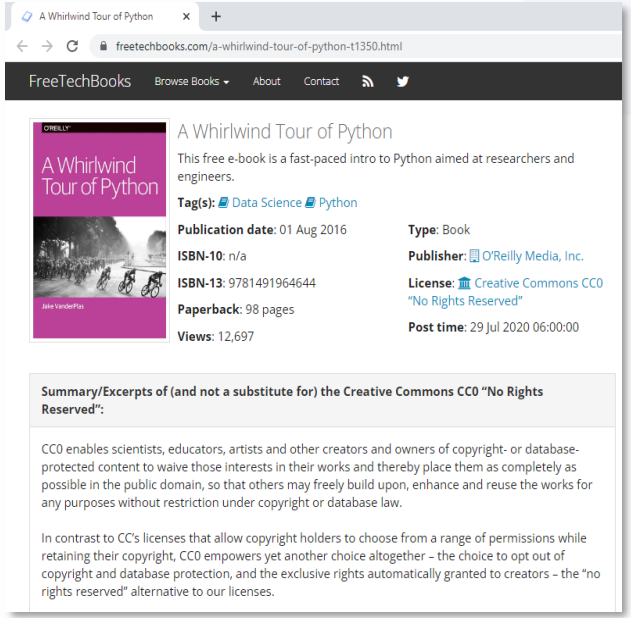
E-books distributed free of charge by the authors (CC licences): https://www.freetechbooks.com/python-f6.html



https://www.freetechbooks.com/a-practical-introduction-to-python-programming-t1376.html



http://getpython3.com/diveintopython3/

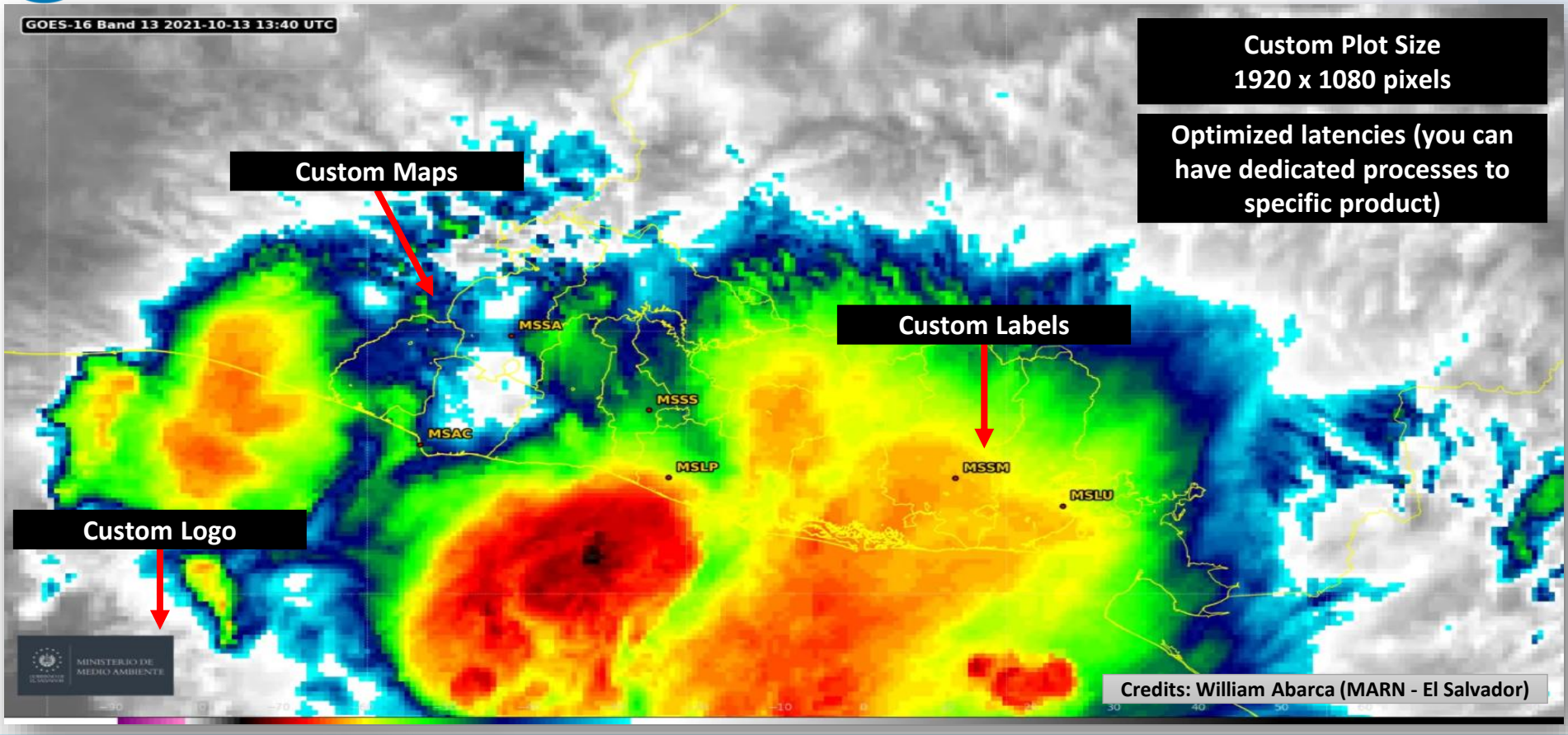


https://jakevdp.github.io/WhirlwindTourOfPython/

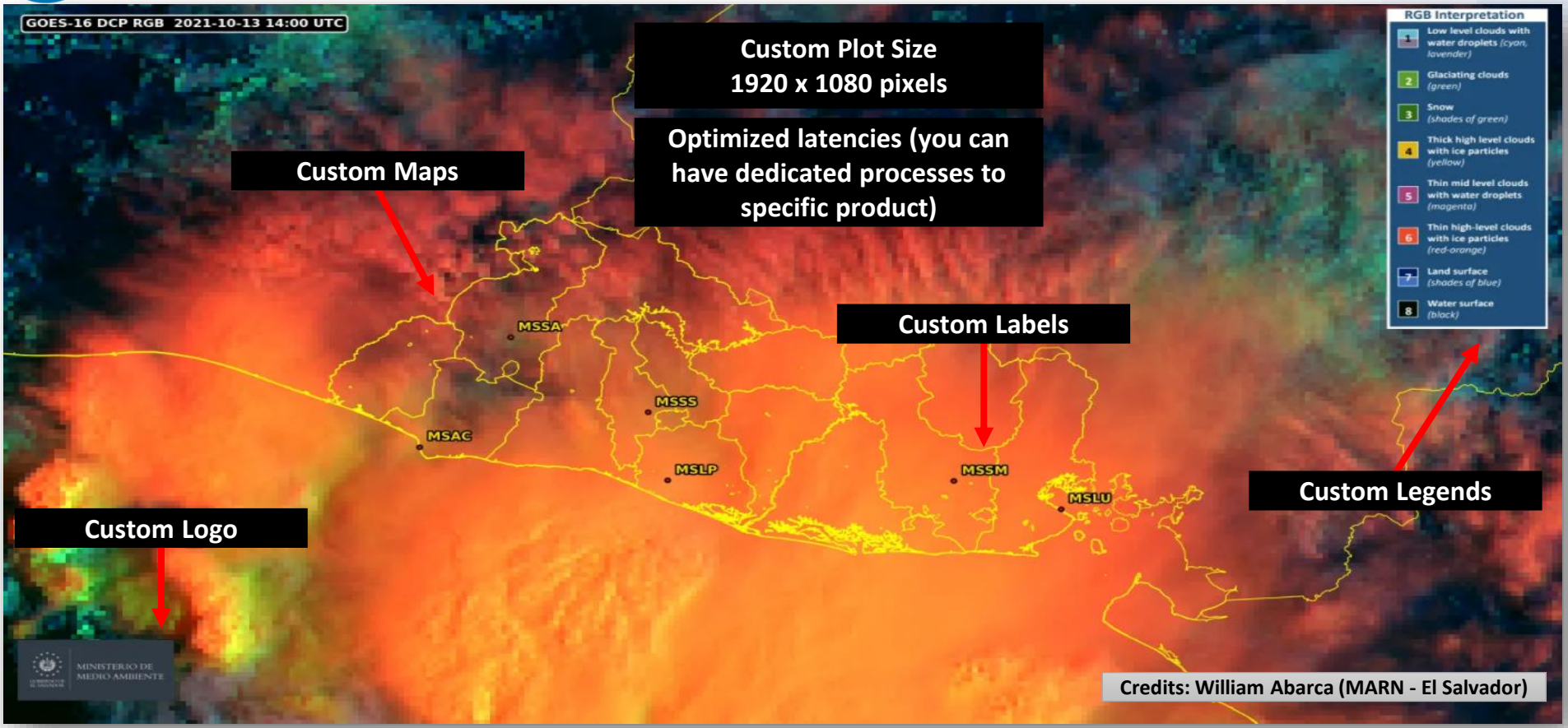
...and much more

BENEFITS OF ACESSING DIGITAL FILES?

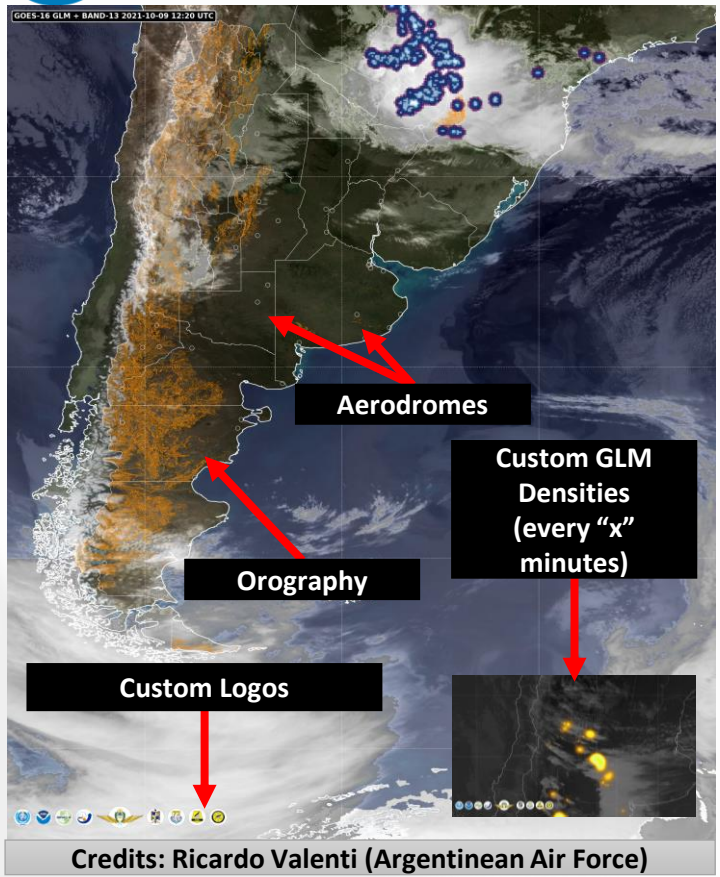
Advantages of Accessing Digital Data



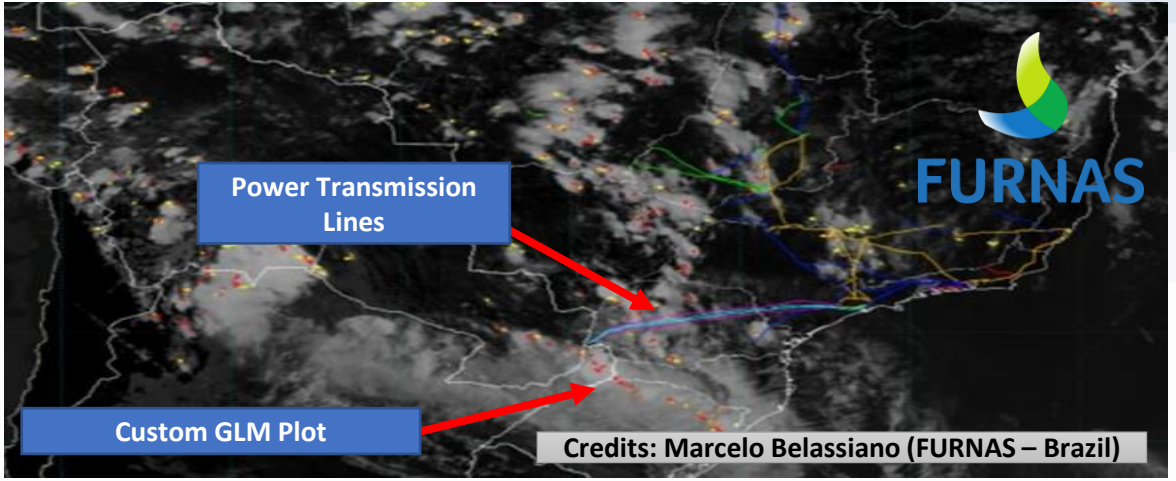
Advantages of Accessing Digital Data



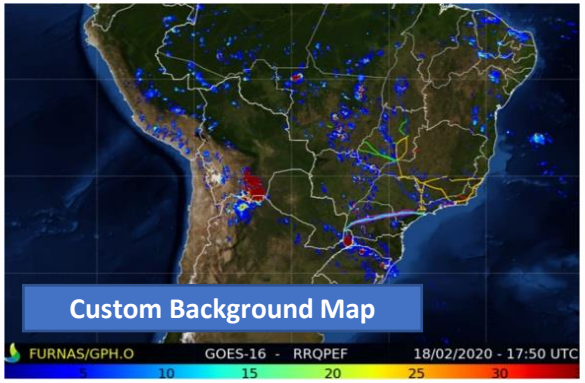
Advantages of Accessing Digital Data

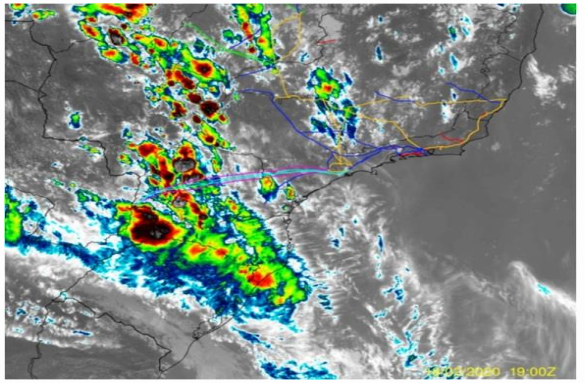


Credits: Ricardo Valenti (Argentinean Air Force)

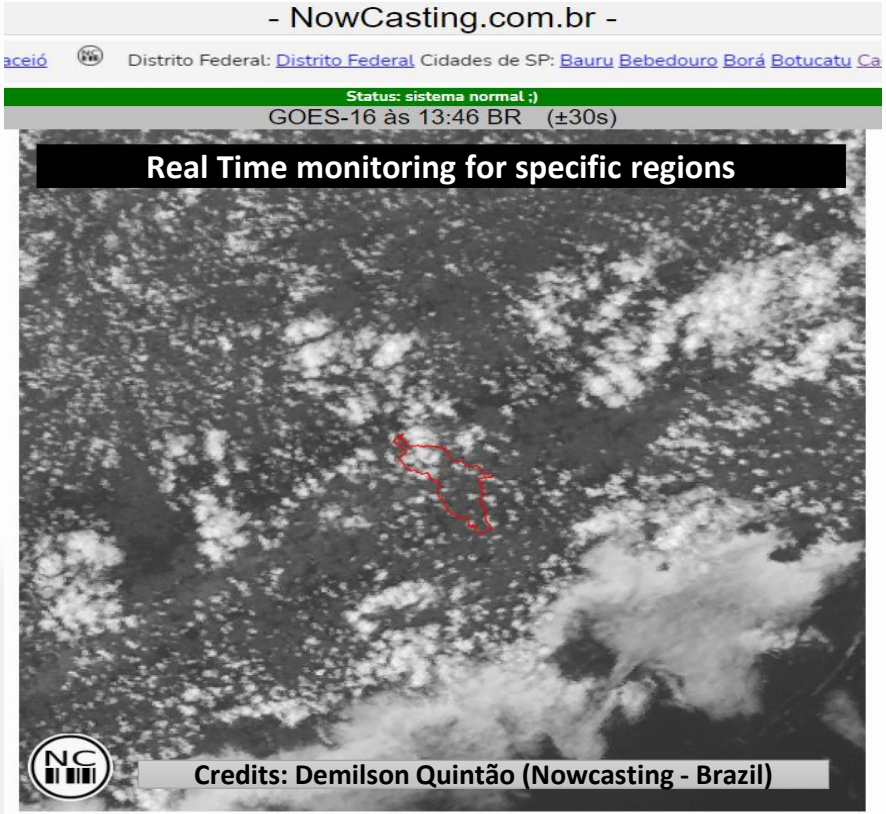


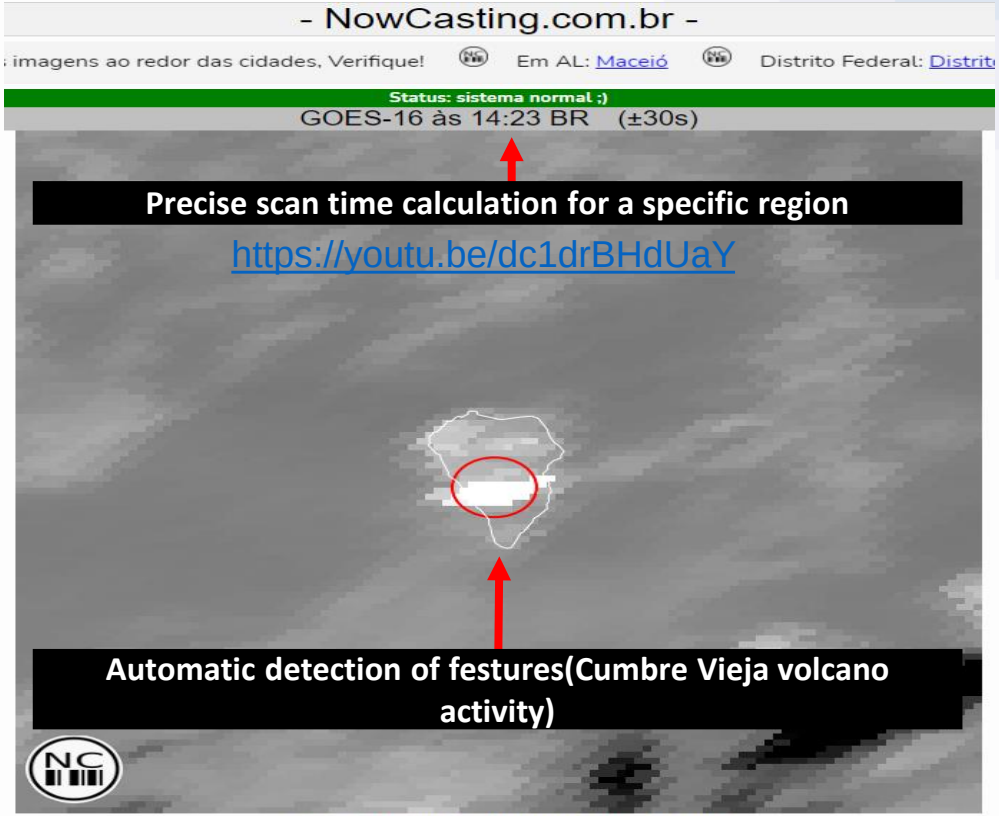
Credits: Marcelo Belassiano (FURNAS – Brazil)





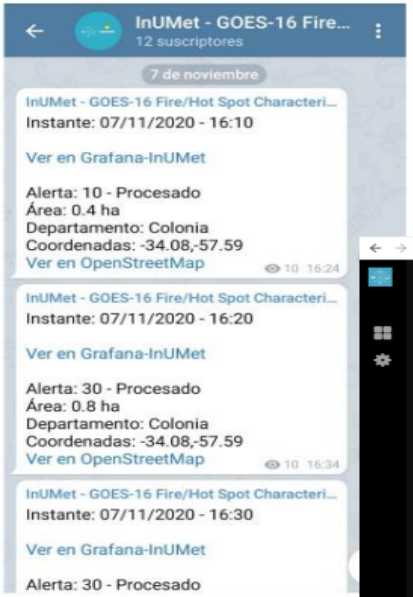
Advantages of Accessing Digital Data





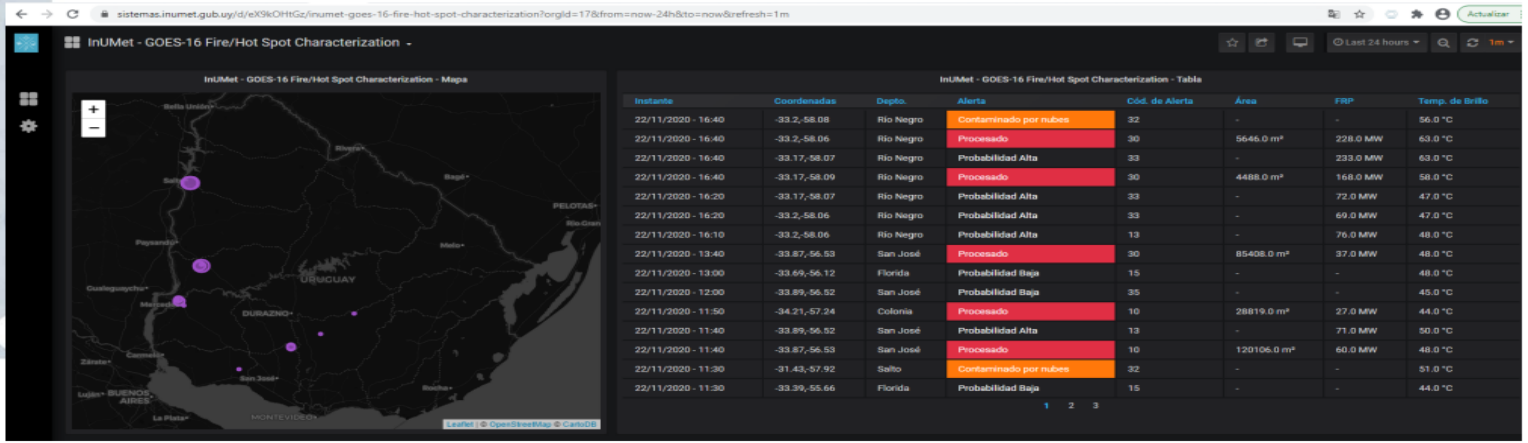
Advantages of Accessing Digital Data

Vías de comunicación de avisos de incendios



Tres vías de acceso a la información:

- Web Service REST.
- **Dashboard de monitoreo** sobre plataforma de BI.
- **Canal privado de noticias** a través de **Telegram**.

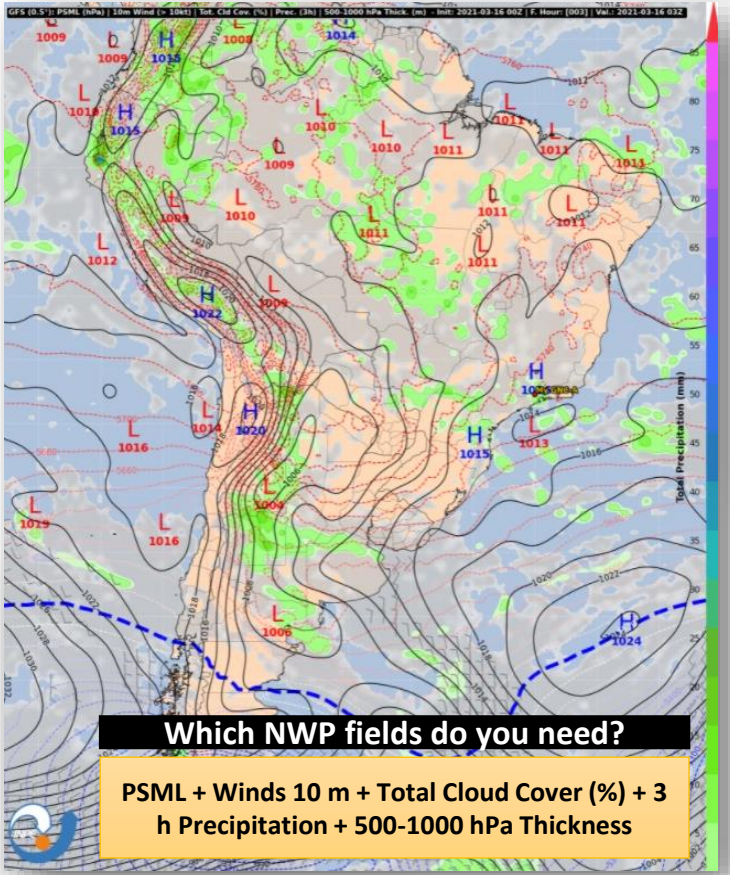


GEONETCast - Uruguay

Credits: Roxana Sagarra (INUMET - Uruguay)

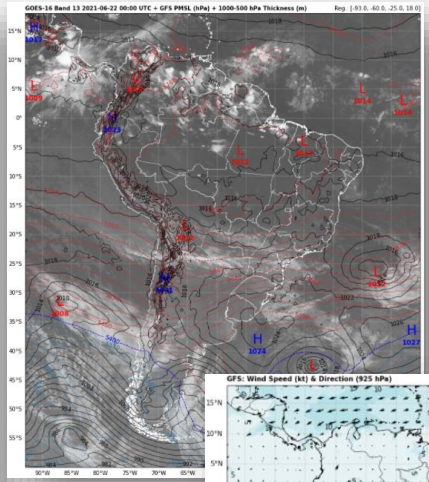
For more user case studies, please access: https://geonetcast.wordpress.com/gnc-a-user-study-cases/

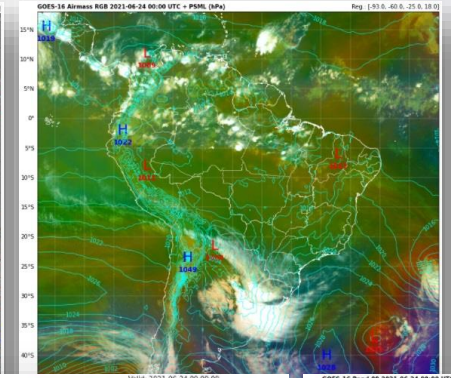
Advantages of Accessing Digital Data

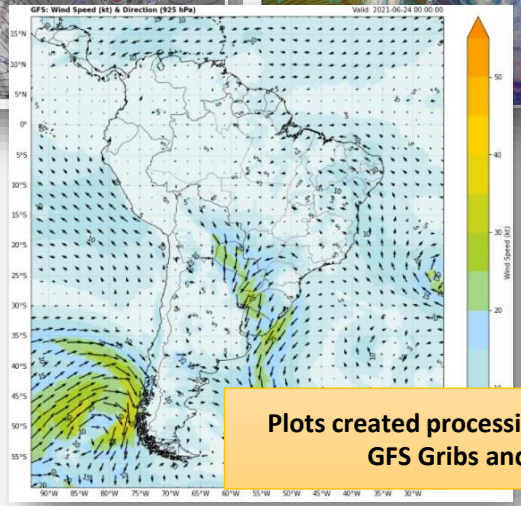


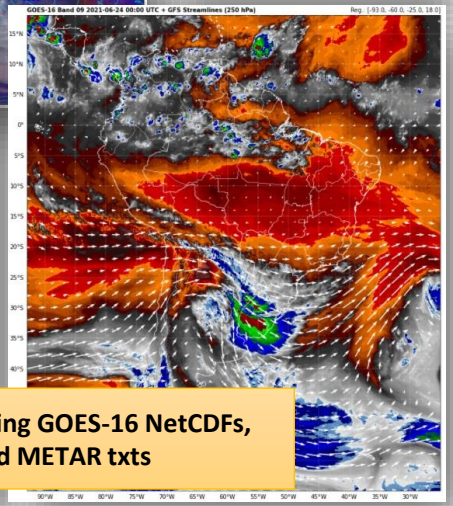
Which NWP fields do you need?

PSML + Winds 10 m + Total Cloud Cover (%) + 3 h Precipitation + 500-1000 hPa Thickness





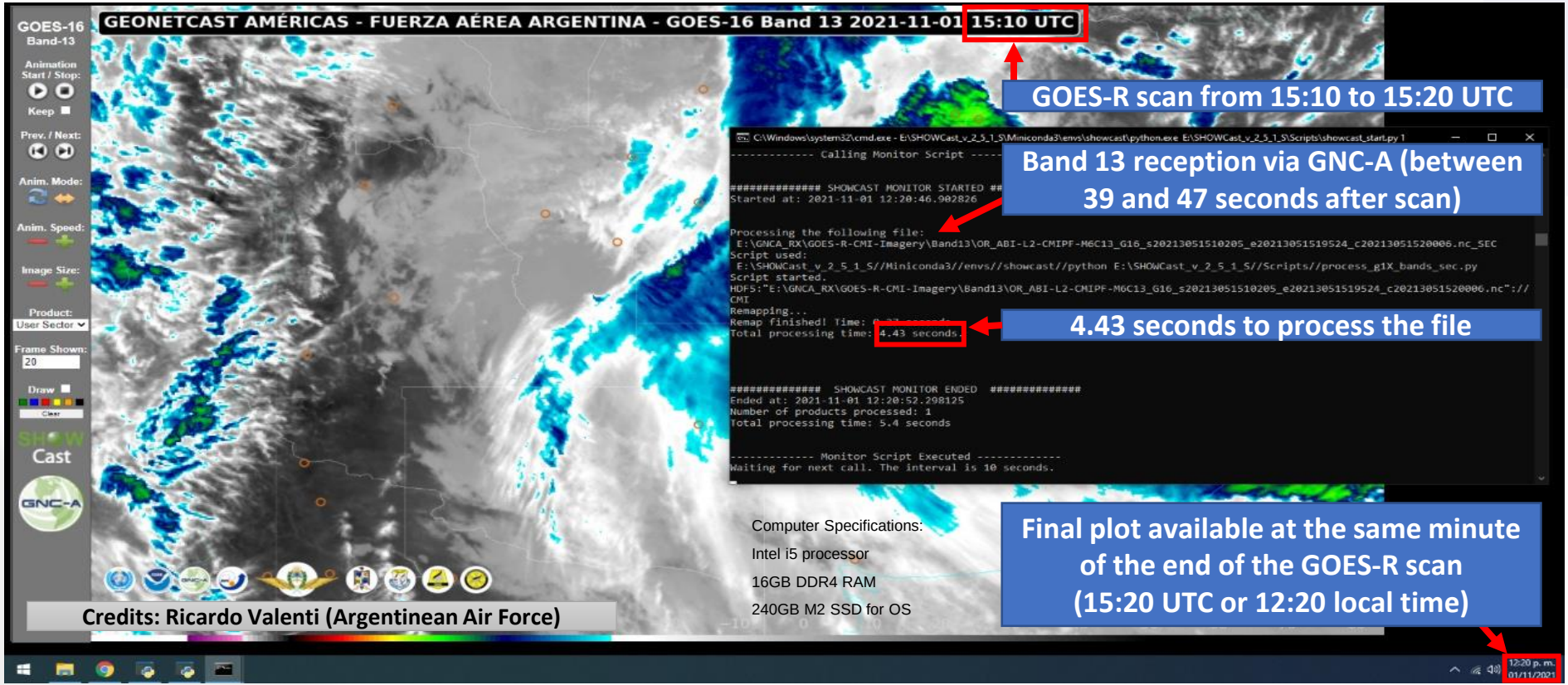




Plots created processing GOES-16 NetCDFs, GFS Gribs and METAR txts

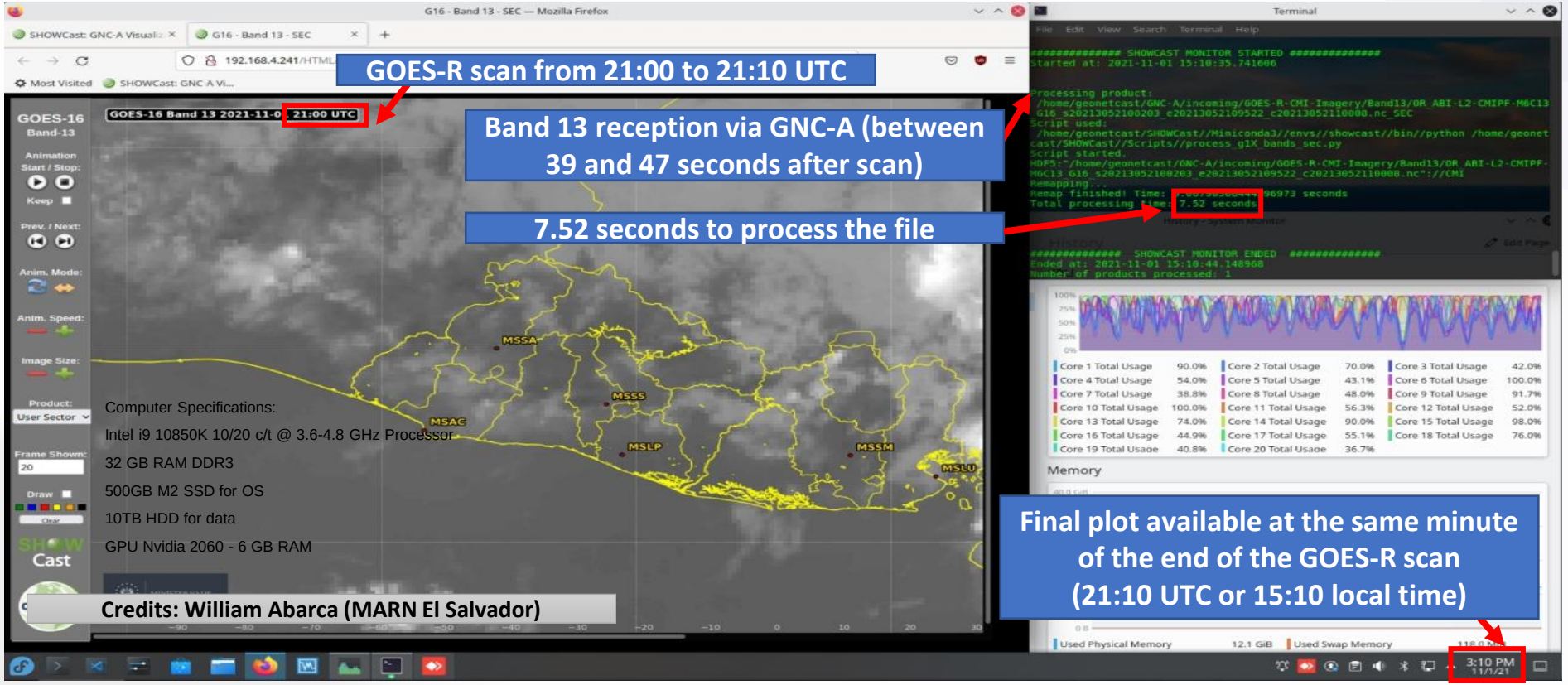
Advantages of Accessing Digital Data

Optimized Latencies



Advantages of Accessing Digital Data

Optimized Latencies

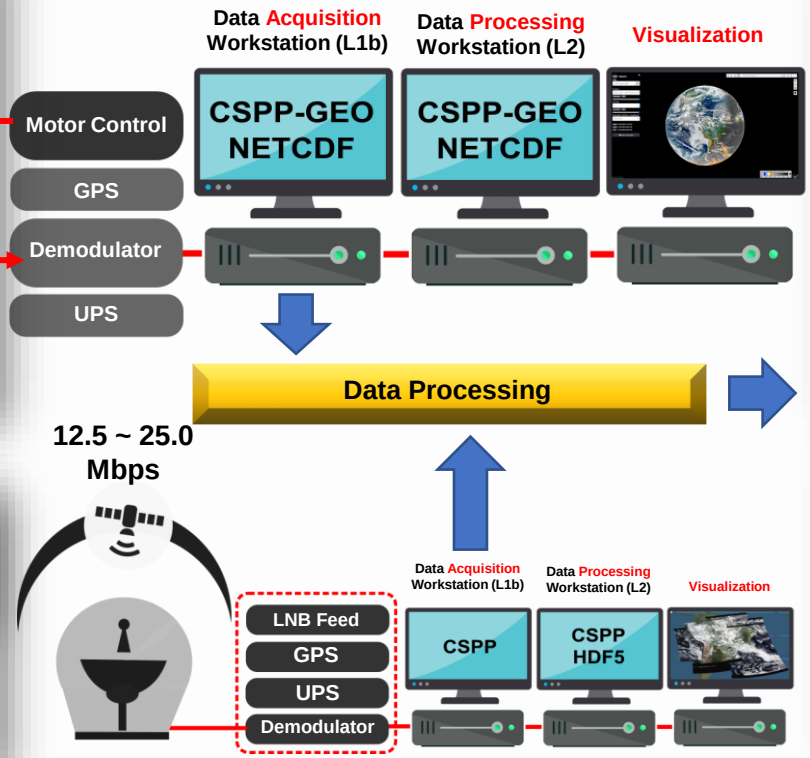


Example: INPE's Meteorological Satellites Division

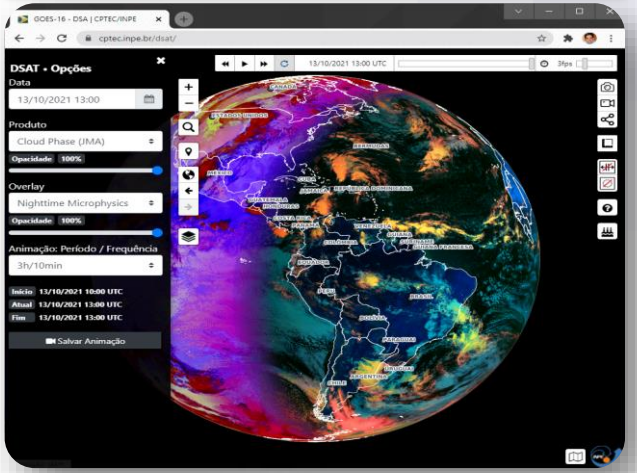
GOES-R

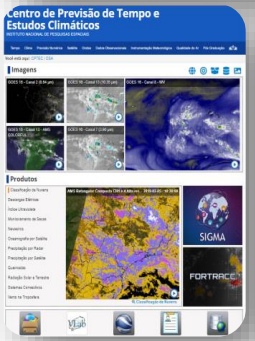
JPSS / LEO

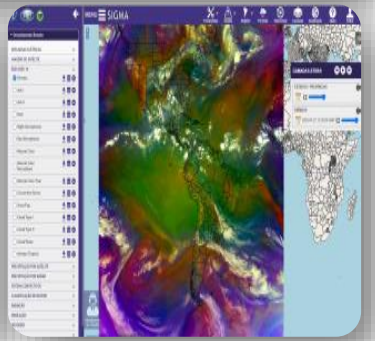
Note: GRB / GNC-A / AWS / PDA (we process what is available first)



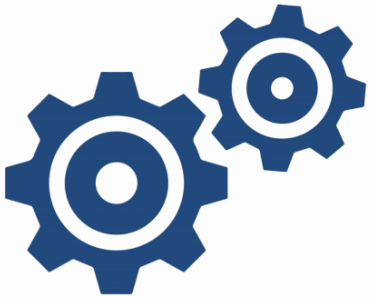
Web Pages, Research and Development



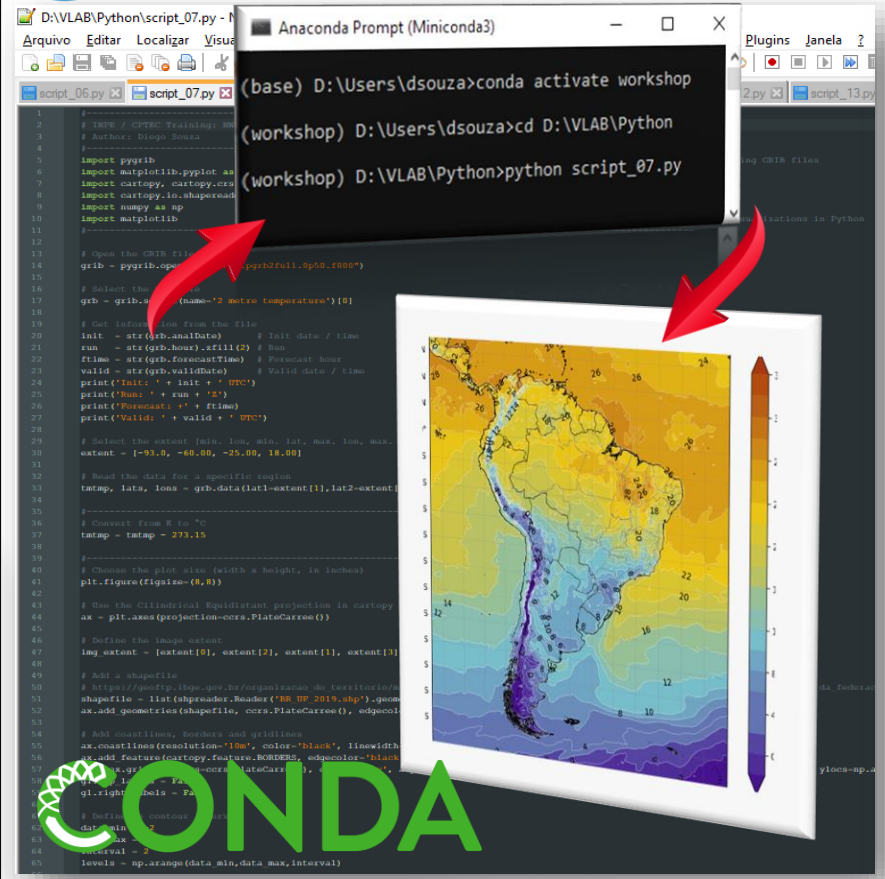


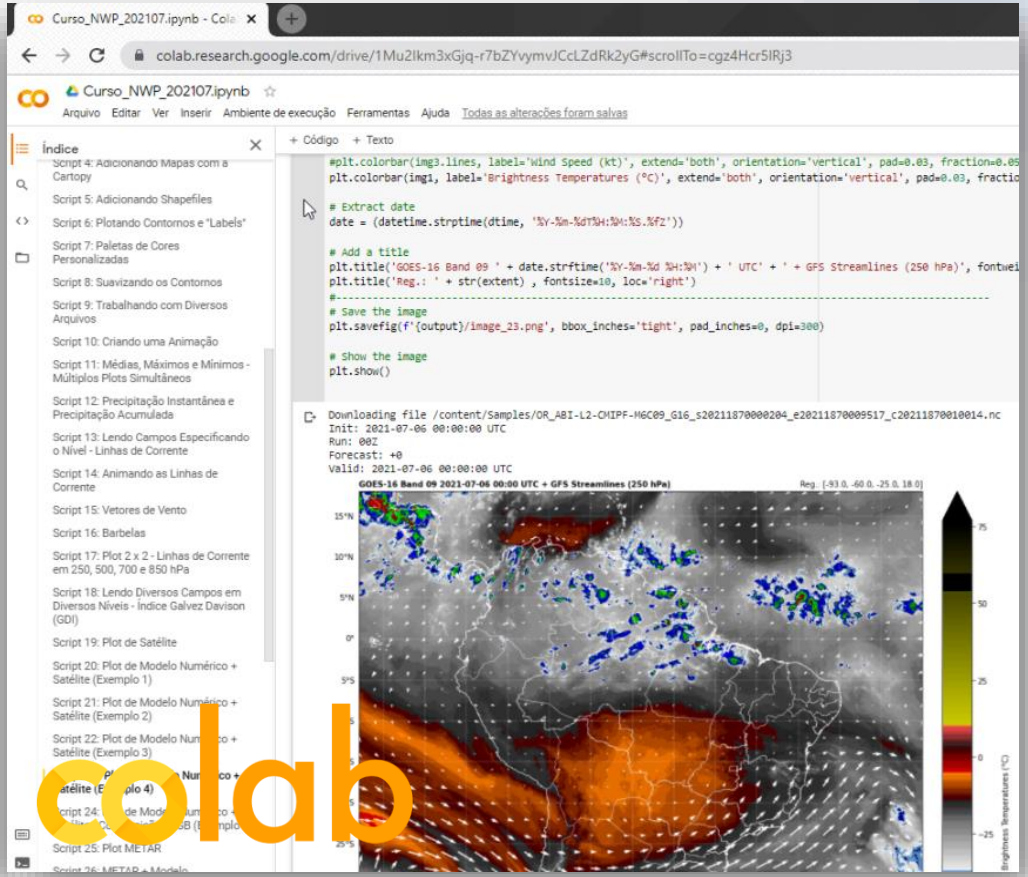


NECESSARY TOOLS



Running Scripts Locally or in the Cloud

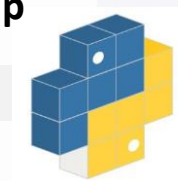




Running Scripts Locally

- Python
- Package Manager
- Virtual Environment Manager

pip



- Text Editor / IDE (Integrated Development Environment)

VS Code

PyCharm

and many others!

- Sample Data

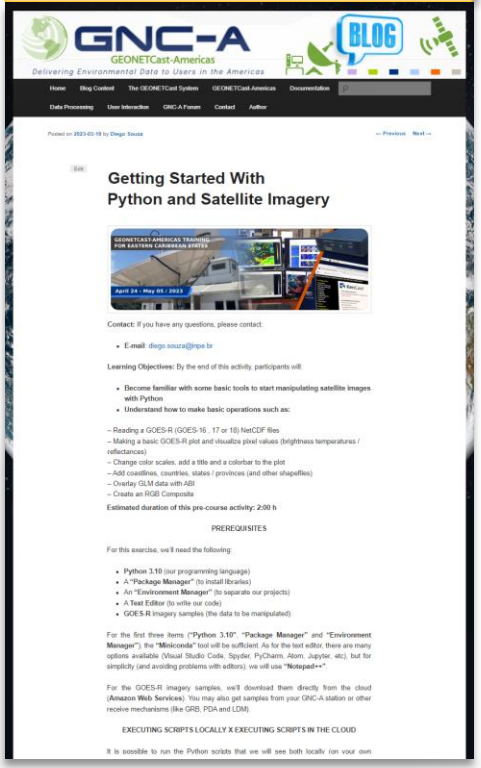
A variety of mechanisms available

Running Scripts Locally

password: workshop

https://geonetcast.wordpress.com/2023/03/19/getting-started-with-python-and-satellite-imagery/

Objectives of the Activity



GNC-A
GEONETCast-Americas
Delivering Environmental Data to Users in the Americas
BLOG

Getting Started With Python and Satellite Imagery

Contact: If you have any questions, please contact:

- E-mail: diego.souza@inpe.br

Learning Objectives: By the end of this activity, participants will:

- Become familiar with some basic tools to start manipulating satellite images with Python
- Understand how to make basic operations such as:

– Reading a GOES-R (GOES-16 , 17 or 18) NetCDF files
– Making a basic GOES-R plot and visualize pixel values (brightness temperatures / reflectances)
– Change color scales, add a title and a colorbar to the plot
– Add coastlines, countries, states / provinces (and other shapefiles)
– Overlay GLM data with ABI
– Create an RGB Composite

Estimated duration of this pre-course activity: 2:00 h

PREREQUISITES

For this exercise, we'll need the following:

- Python 3.10 (our programming language)
- A "Package Manager" (to install libraries)
- An "Environment Manager" (to separate our projects)
- A Text Editor (to write our code)
- GOES-R imagery samples (the data to be manipulated)

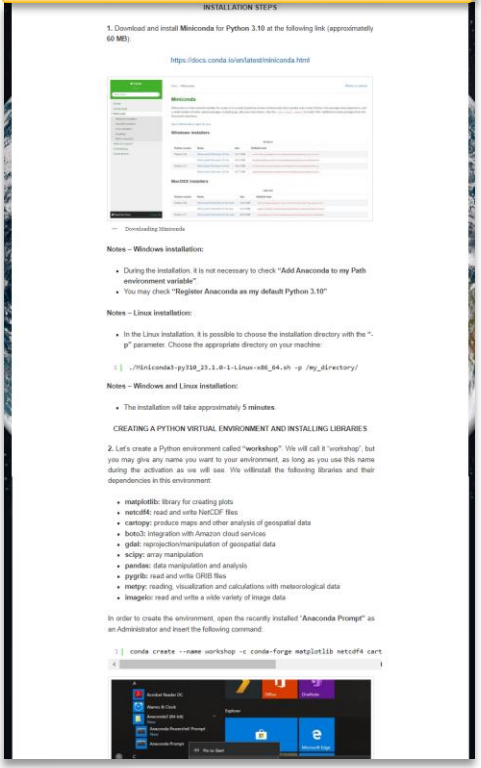
For the first three items ("Python 3.10", "Package Manager" and "Environment Manager"), the "Miniconda" tool will be sufficient. As for the text editor, there are many options available (Visual Studio Code, Spyder, PyCharm, Atom, Jupyter, etc), but for simplicity (and avoiding problems with editors), we will use "Notepad++".

For the GOES-R imagery samples, we'll download them directly from the cloud (Amazon Web Services). You may also get samples from your GNC-A station or other receive mechanisms (like GRB, PDA and LDM).

EXECUTING SCRIPTS LOCALLY X EXECUTING SCRIPTS IN THE CLOUD

It is possible to run the Python scripts that we will see both locally (on your own

Installation of Tools



INSTALLATION STEPS

1. Download and install **Miniconda** for Python 3.10 at the following link (approximately 60 MB):

https://docs.conda.io/en/latest/miniconda.html

Notes – Windows installation:

- During the installation, it is not necessary to check "Add Anaconda to my Path environment variable".
- You may check "Register Anaconda as my default Python 3.10"

Notes – Linux installation:

- In the Linux installation, it is possible to choose the installation directory with the "-p" parameter. Choose the appropriate directory on your machine:

```
./Miniconda3-py310_23.1.0-1-Linux-x86_64.sh -p /my_directory/
```

Notes – Windows and Linux installation:

- The installation will take approximately 5 minutes.

CREATING A PYTHON VIRTUAL ENVIRONMENT AND INSTALLING LIBRARIES

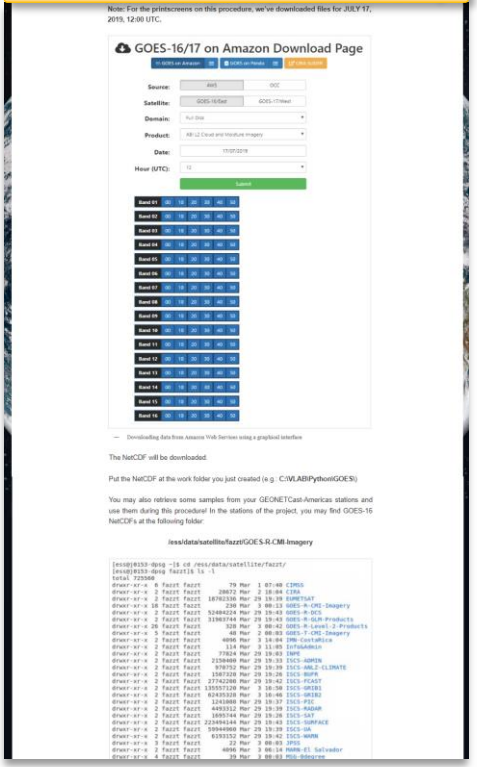
2. Let's create a Python environment called "workshop". We will call it 'workshop', but you may give any name you want to your environment, as long as you use this name during the activation as we will see. We willinstall the following libraries and their dependencies in this environment:

- matplotlib: library for creating plots
- netcdf4: read and write NetCDF files
- cartopy: produce maps and other analysis of geospatial data
- boto3: integration with Amazon cloud services
- gdal: reprojection/manipulation of geospatial data
- scipy: array manipulation
- pandas: data manipulation and analysis
- pygrib: read and write GRIB files
- metpy: reading, visualization and calculations with meteorological data
- imageio: read and write a wide variety of image data

In order to create the environment, open the recently installed "Anaconda Prompt" as an Administrator and insert the following command:

```
conda create --name workshop -c conda-forge matplotlib netcdf4 cart
```

Access to Samples (GNC or Cloud)



Note: For the printscreens on this procedure, we've downloaded files for JULY 17, 2019, 12:00 UTC.

GOES-16/17 on Amazon Download Page

Source:
Satellite:
Domain:
Product:
Date:
Hour (UTC):

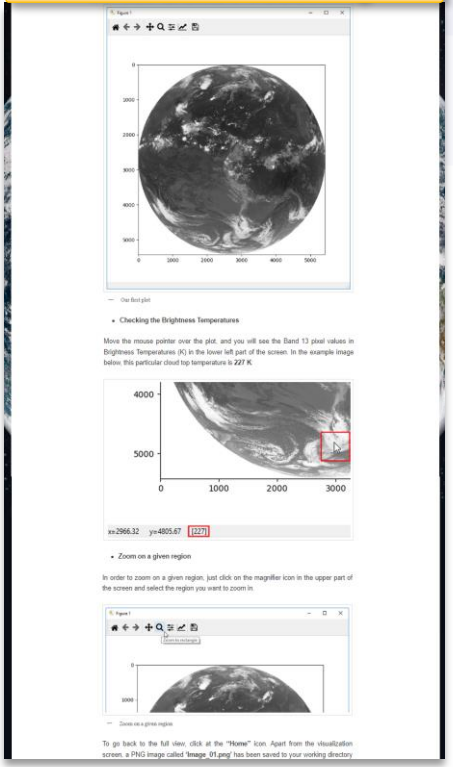
The NetCDF will be downloaded.

Put the NetCDF at the work folder you just created (e.g.: C:\VLAB\Python\GOES\)

You may also retrieve some samples from your GEONETCast-Americas stations and use them during this procedure! In the stations of the project, you may find GOES-16 NetCDFs at the following folder:

/ess/data/satellite/fazzt/GOES-R.CMI-Imagery

Local Data Processing



- Checking the Brightness Temperatures

Move the mouse pointer over the plot, and you will see the Band 13 pixel values in Brightness Temperatures (K) in the lower left part of the screen. In the example image below, this particular cloud top temperature is 227 K

- Zoom on a given region

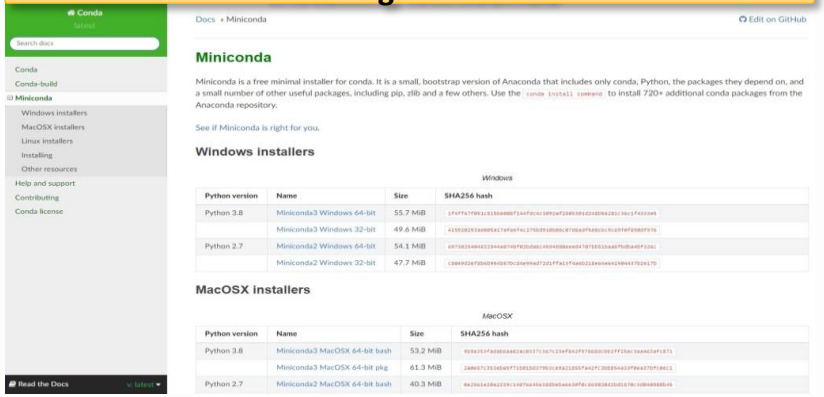
In order to zoom on a given region, just click on the magnifier icon in the upper part of the screen and select the region you want to zoom in

To go back to the full view, click at the "Home" icon. Apart from the visualization screen, a PNG image called 'Image_01.png' has been saved to your working directory

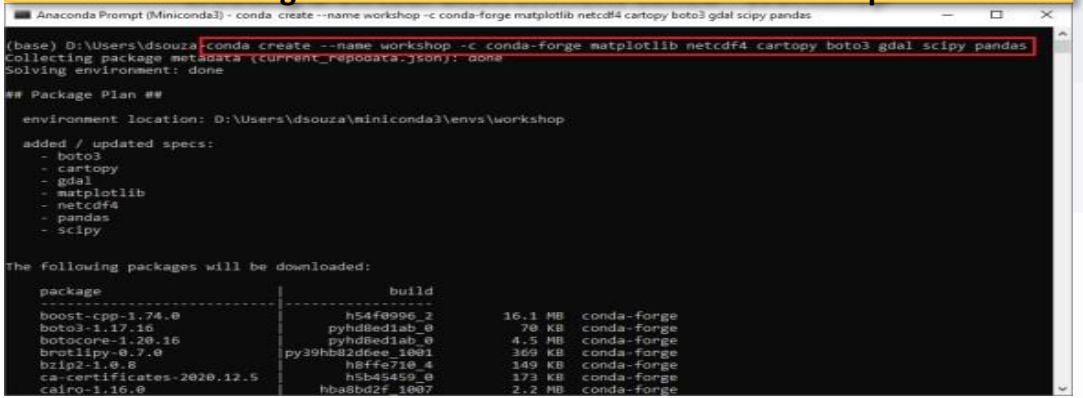
Running Scripts Locally: Summary

https://geonetcast.wordpress.com/2023/03/19/getting-started-with-python-and-satellite-imagery/

Installing Miniconda

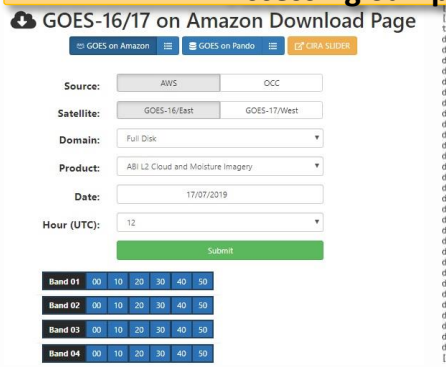


Creating a Virtual Environment called “Workshop”

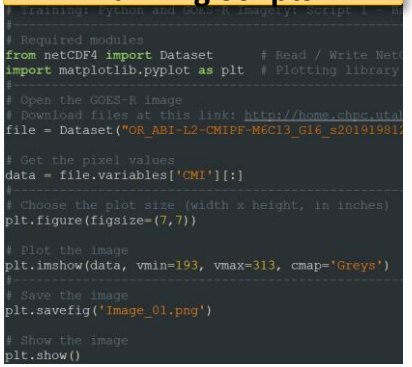


Installing a Text Editor

Accessing Sample Imagery

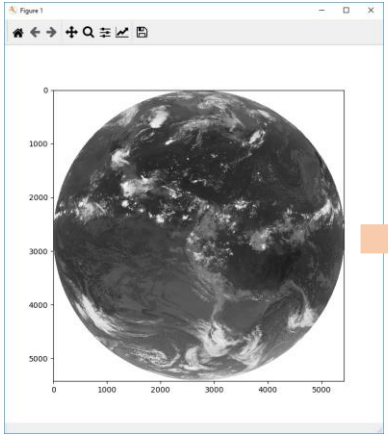


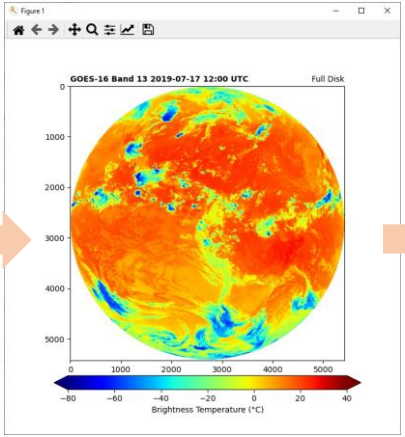
Running Scripts

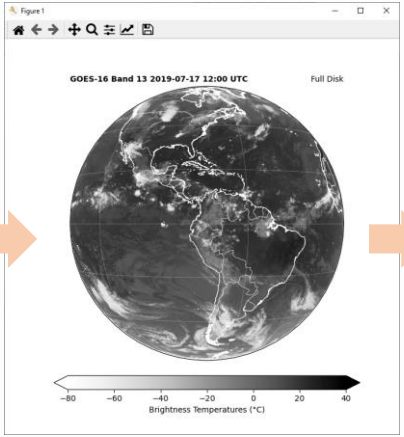


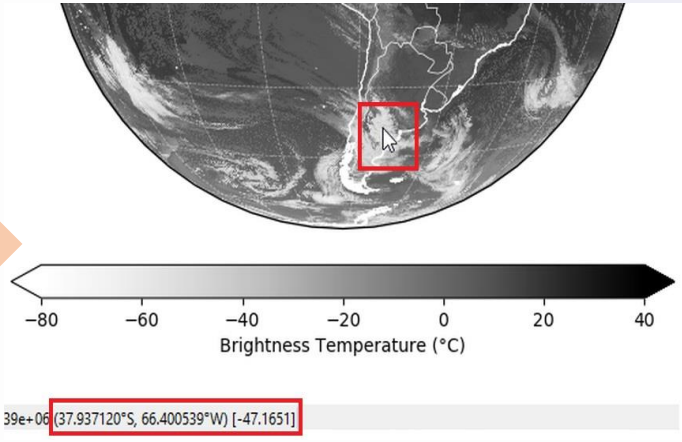
Running Scripts Locally: Summary

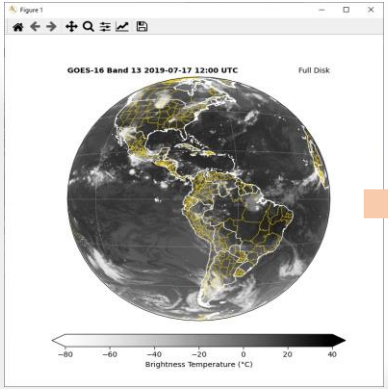
https://geonetcast.wordpress.com/2023/03/19/getting-started-with-python-and-satellite-imagery/

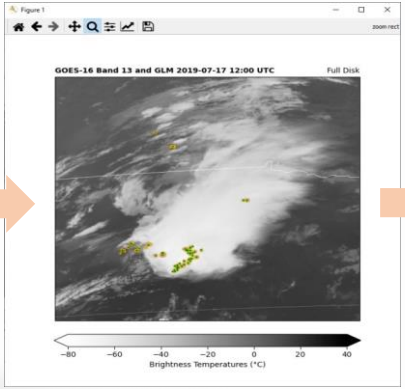


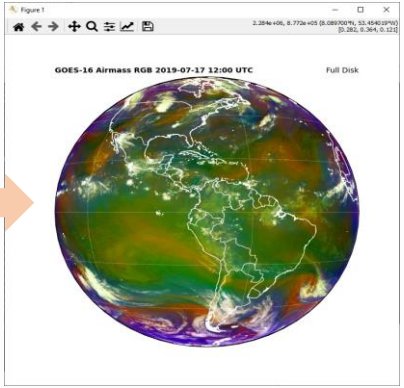












If you have any questions or difficulties, please send me an email: diego.souza@inpe.br

And if possible, please try to follow this tutorial until next Wednesday (May 10th), when we will learn how to create advanced animations.

Running Scripts Locally: Summary

https://geonetcast.wordpress.com/2023/03/19/getting-started-with-python-and-satellite-imagery/

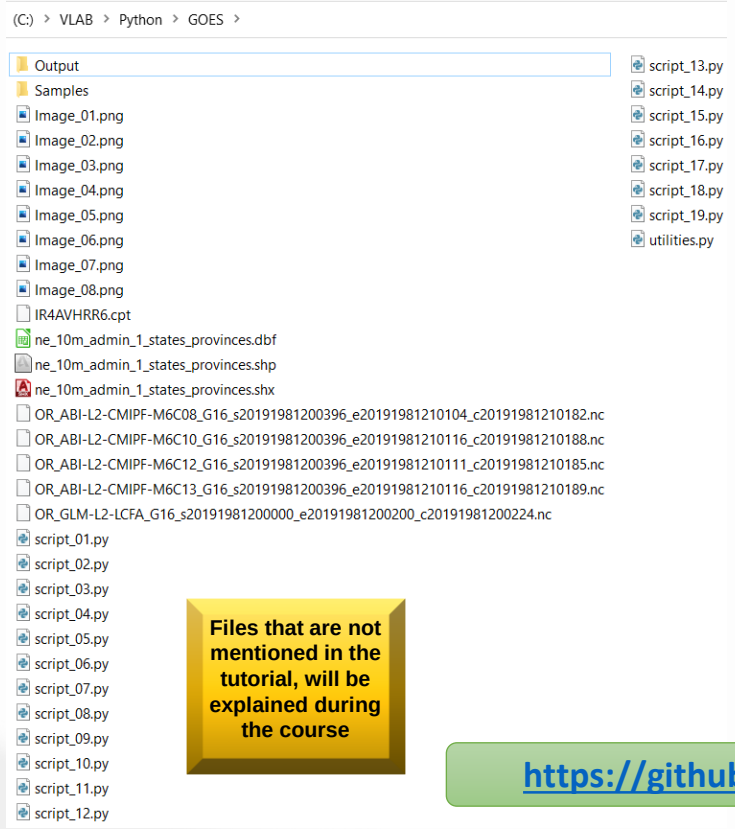
SAMPLES AND OUTPUT DIR. (CREATED VIA CODE)

IMAGES CREATED IN THE PRE-COURSE ACTIVITY

SHAPEFILE

NETCDFs ACCESSED IN THE PRE-COURSE ACTIVITY

PYTHON SCRIPTS



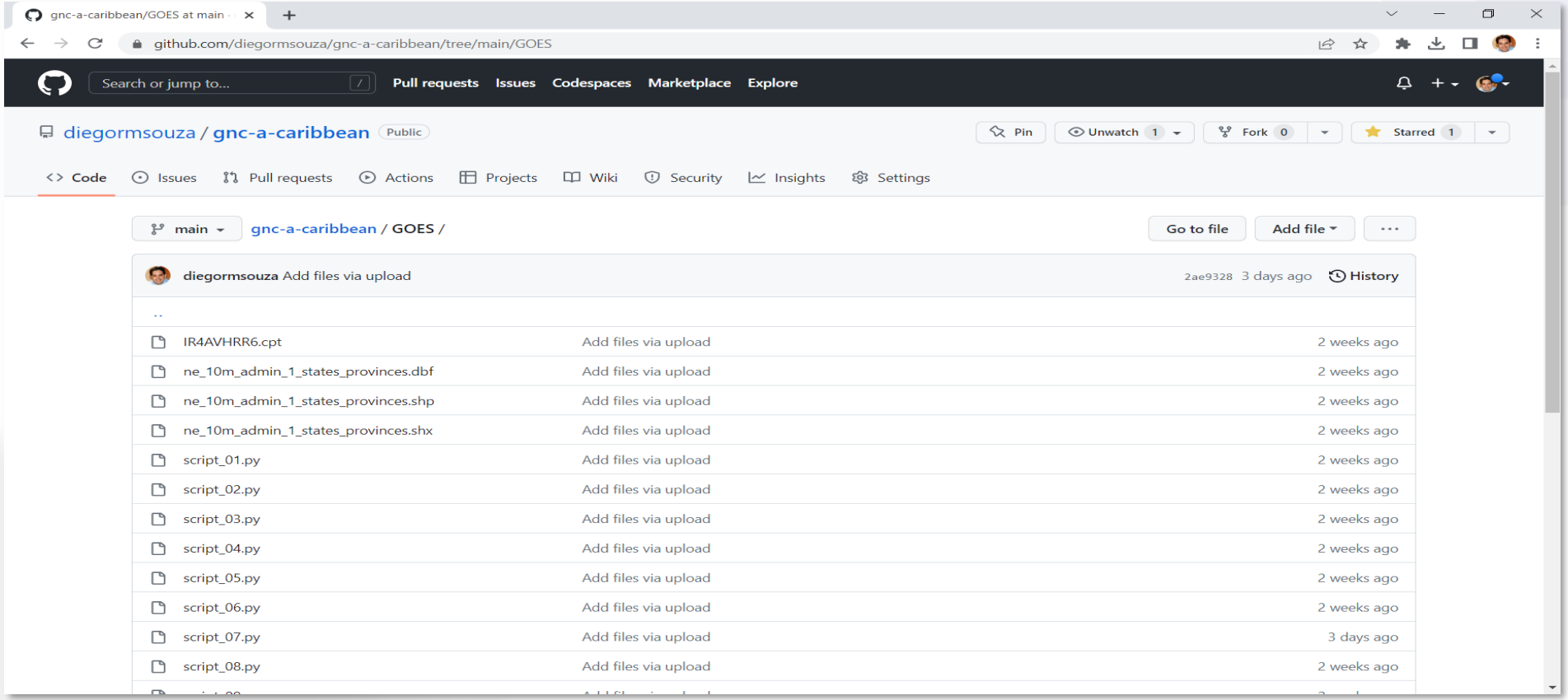
Files that are not mentioned in the tutorial, will be explained during the course

Script 1: Basic Plot / Reading Pixel Values
Script 2: Basic Operation / Colorbar / Title / Date
Script 3: Overlaying Maps with Cartopy
Script 4: Overlaying Maps with Cartopy (Reading the Metadata)
Script 5: Reading a Shapefile
Script 6: ABI + GLM (Basic Plot)
Script 7: Creating RGB Composites
Script 8: Custom Colormaps - Enhancing IR Channels
Script 9: Downloading Data from the Cloud - Amazon Web Services (AWS)
Script 10: Downloading Data from the Cloud - AWS (Using a Function)
Script 11: Downloading Data from the Cloud - AWS (Importing a Function from "utilities.py")
Script 12: Cropping a Full Disk Image
Script 13: Cropping a Full Disk Image and Creating an RGB Composite
Script 14: Reprojection with GDAL
Script 15: Level 2 Products (SST) and Data Quality Flags (DQF)
Script 16: Level 2 Products (SST) and Daily Average
Script 17: Level 2 Products (RRQPE - Rainfall Rate) and Data Accumulation
Script 18: GLM Density
Script 19: GLM Heatmap

https://github.com/diegormsouza/gnc-a-caribbean/tree/main/GOES

Running Scripts Locally: GitHub Page - GOES

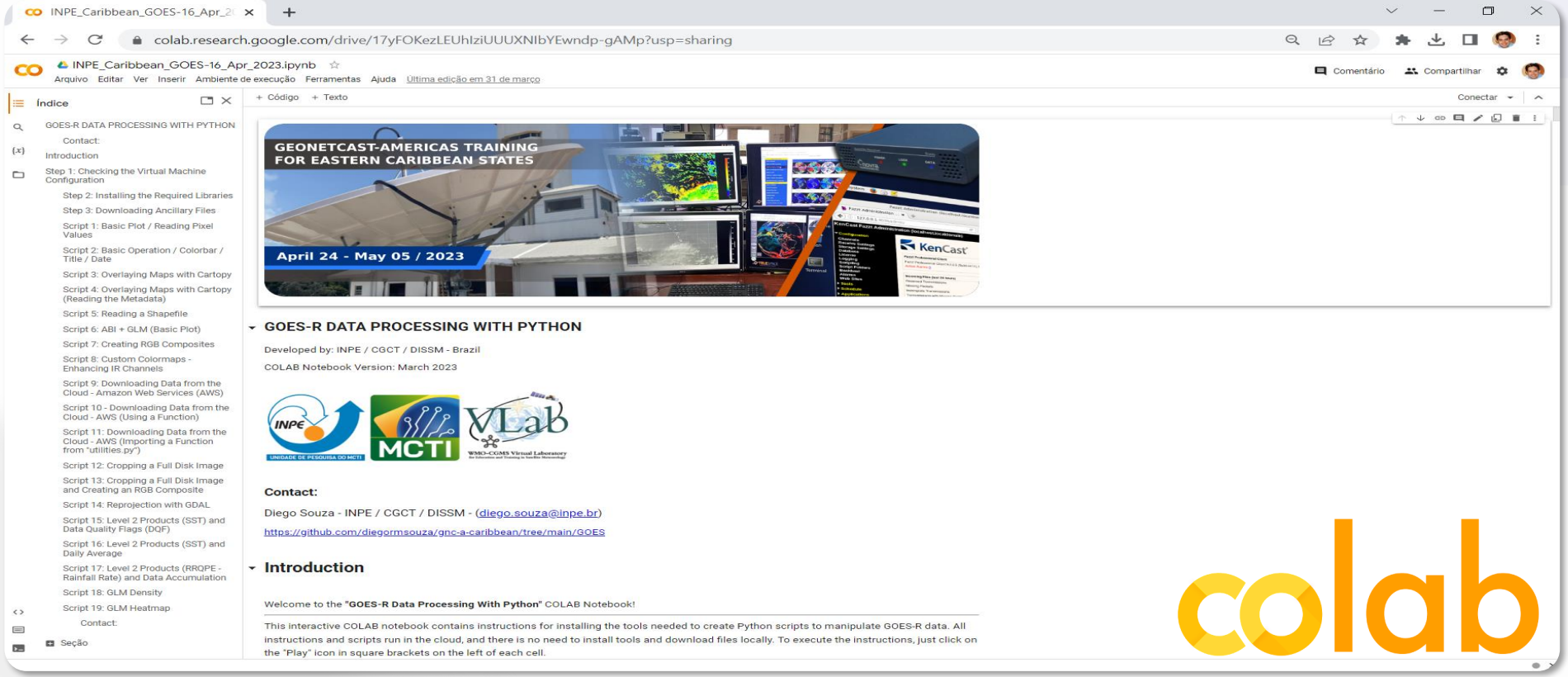
https://github.com/diegormsouza/gnc-a-caribbean/tree/main/GOES



DEMONSTRATION: RUNNING SCRIPTS LOCALLY

Running Scripts in the Cloud

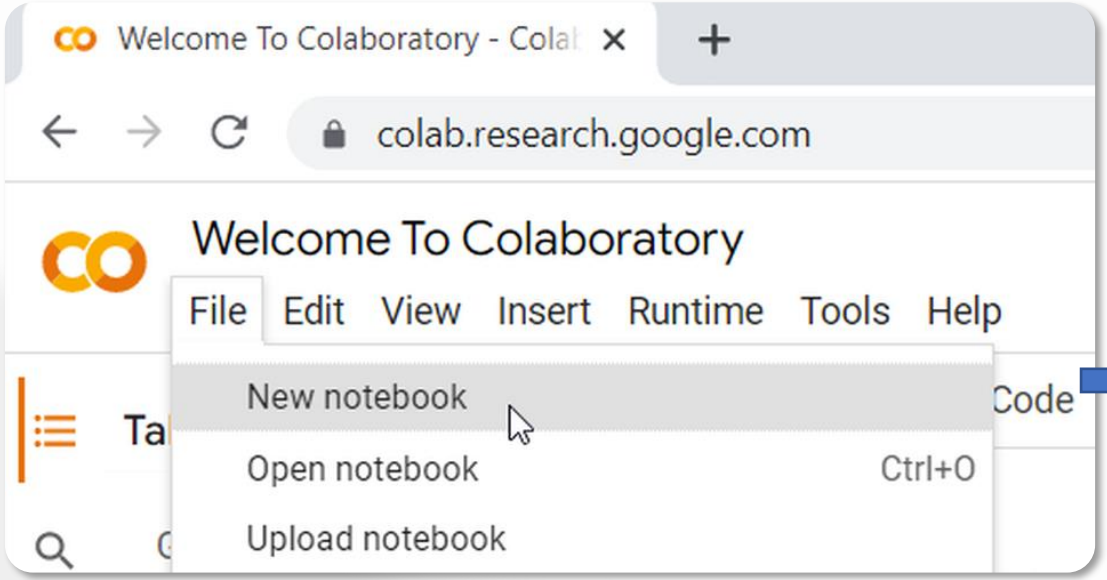
https://colab.research.google.com/drive/17yFOKezLEUhIziUUUXNIbYEwndp-gAMp?usp=sharing

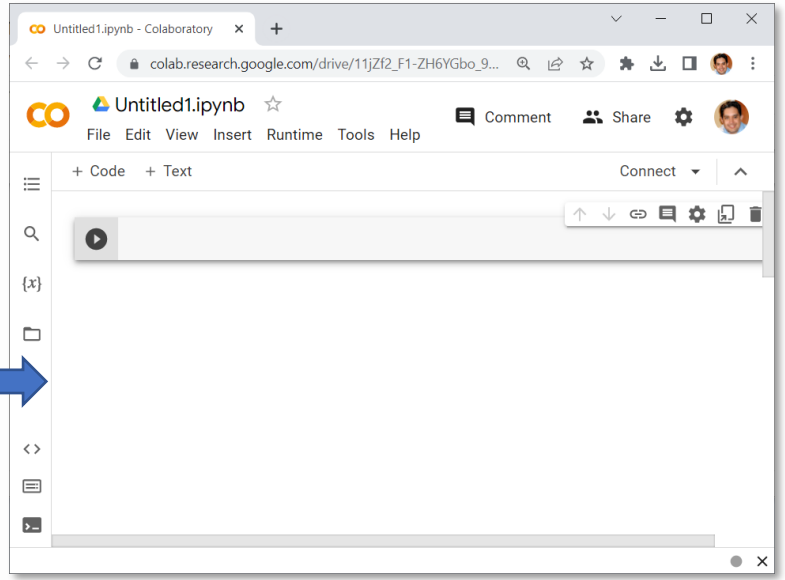


GOOGLE COLAB: BASIC CONCEPTS

Getting started with Google COLAB:

https://colab.research.google.com/

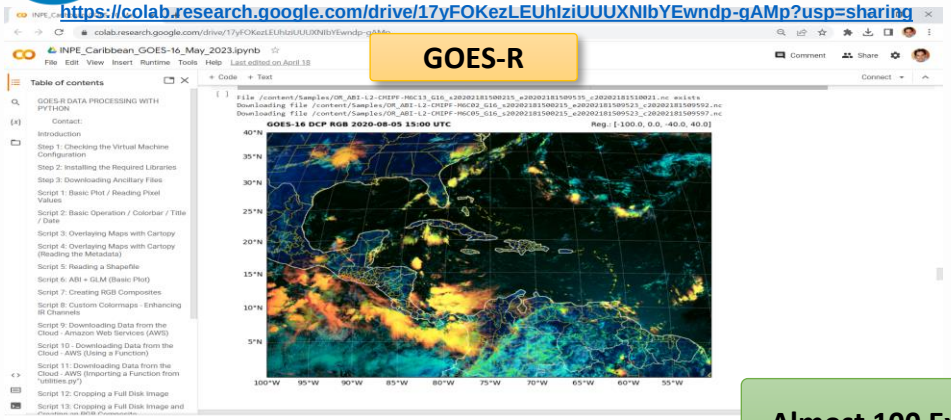




Running Scripts in the Cloud

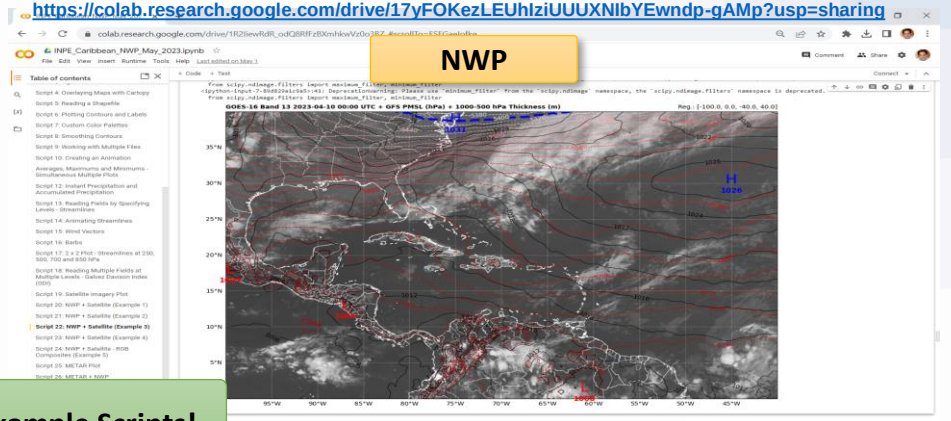
https://colab.research.google.com/drive/17yFOKezLEUhIzIUUXNIbYEwndp-gAMp?usp=sharing

GOES-R



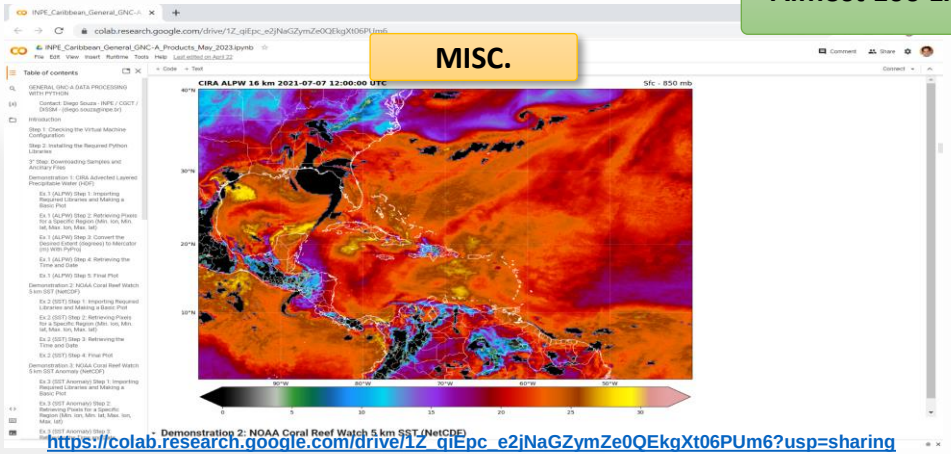
https://colab.research.google.com/drive/17yFOKezLEUhIzIUUXNIbYEwndp-gAMp?usp=sharing

NWP



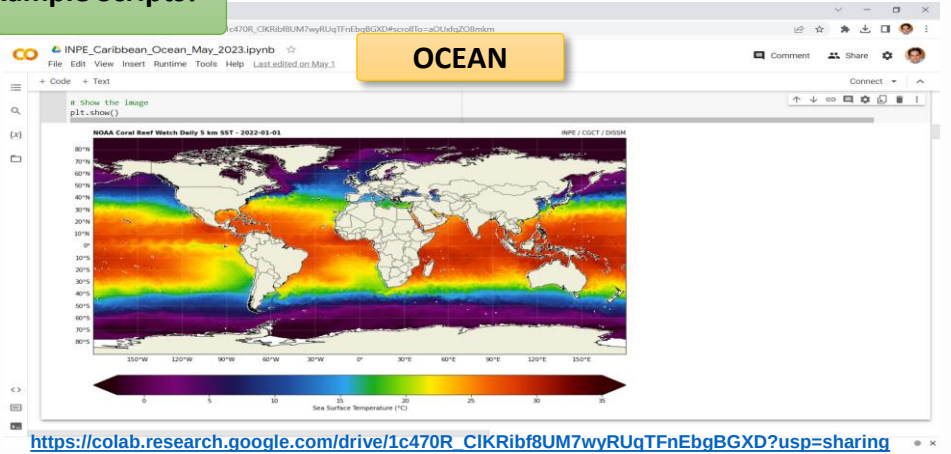
Almost 100 Example Scripts!

MISC.



https://colab.research.google.com/drive/1Z_qiEpc_e2jNaGZymZe0QEkgXt06PUm6?usp=sharing

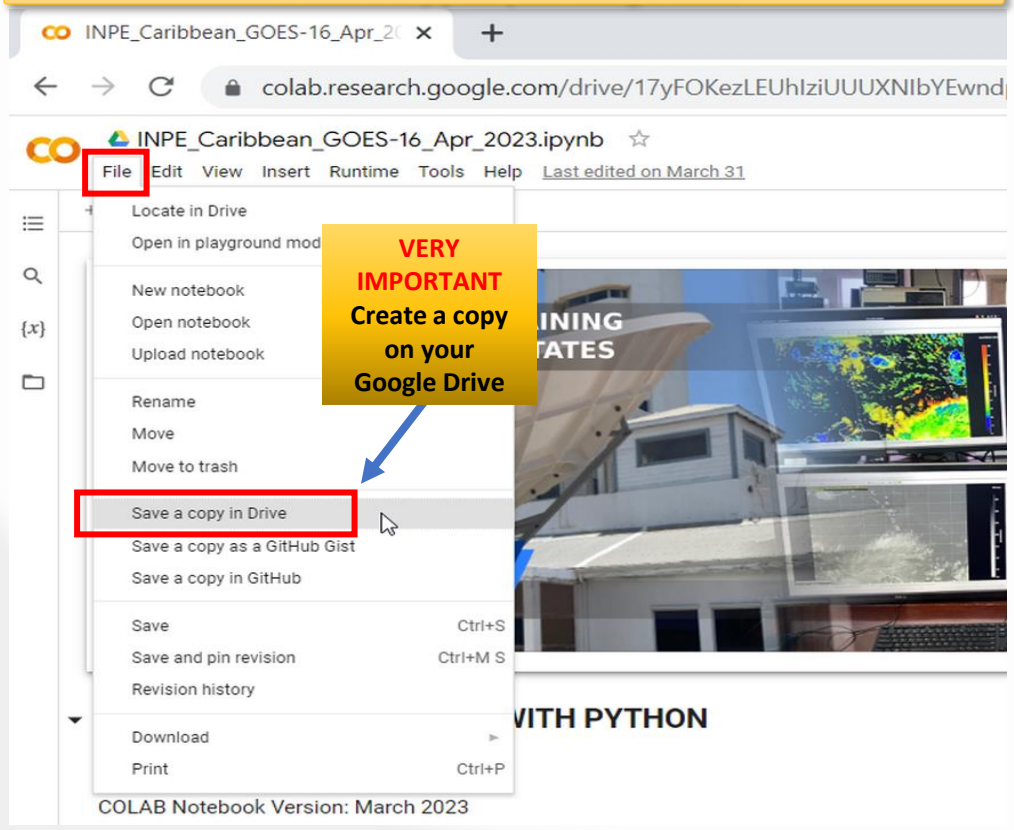
OCEAN



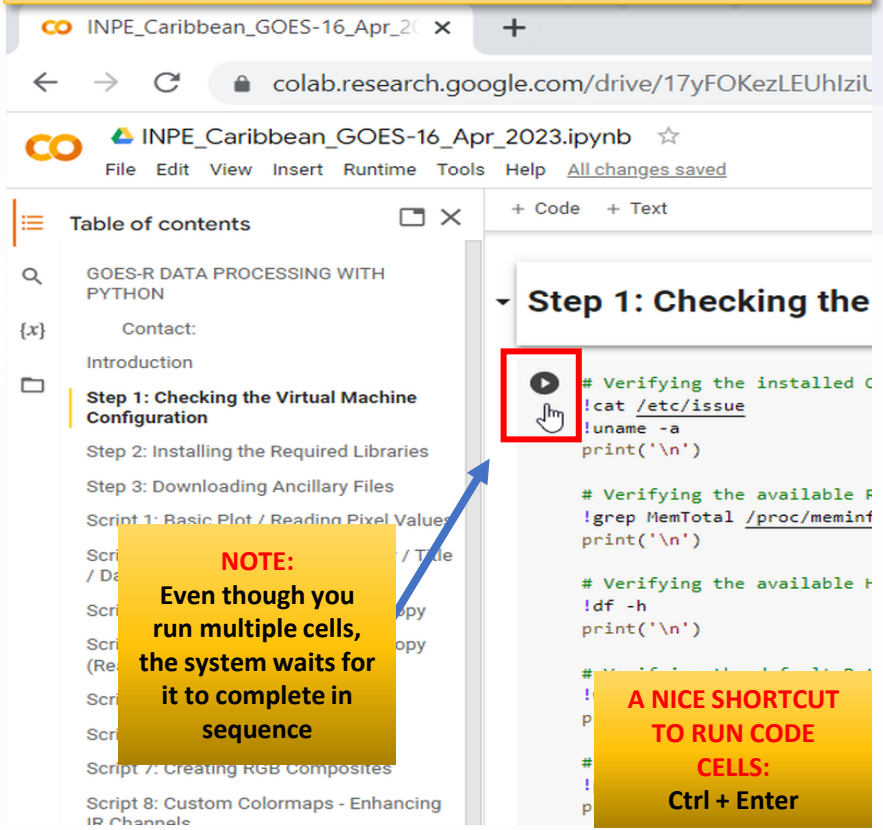
https://colab.research.google.com/drive/1c470R_ClKRibf8UM7wyRUqTFnEbgBGXD?usp=sharing

Running Scripts in the Cloud

Create a copy on your Google Drive



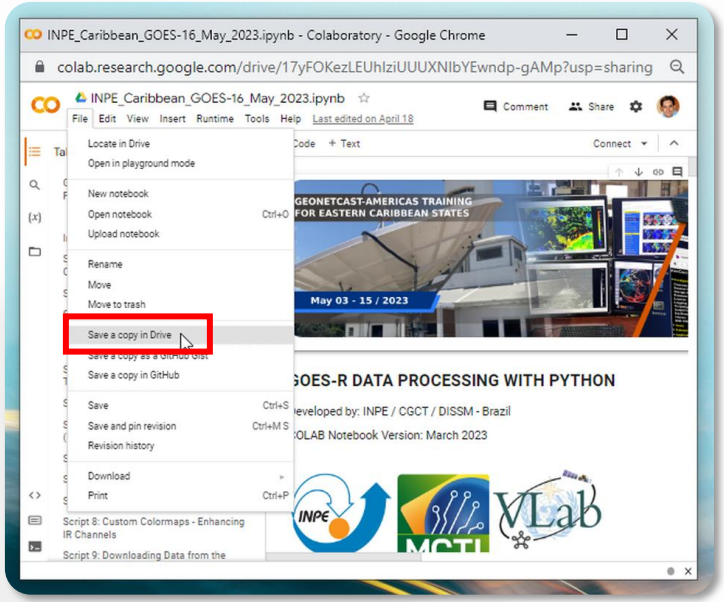
Run the instructions in each cell, sequentially



Pause: Please Make a Copy and Run Steps 2 and 3

https://colab.research.google.com/drive/17yFOKezLEUhIziUUUXNIbYEwndp-gAMp?usp=sharing

It takes some minutes to install the required libraries (step 2) and download auxiliary data (step 3). If you run these cells now, when we start the hands-on, your notebook will be ready!



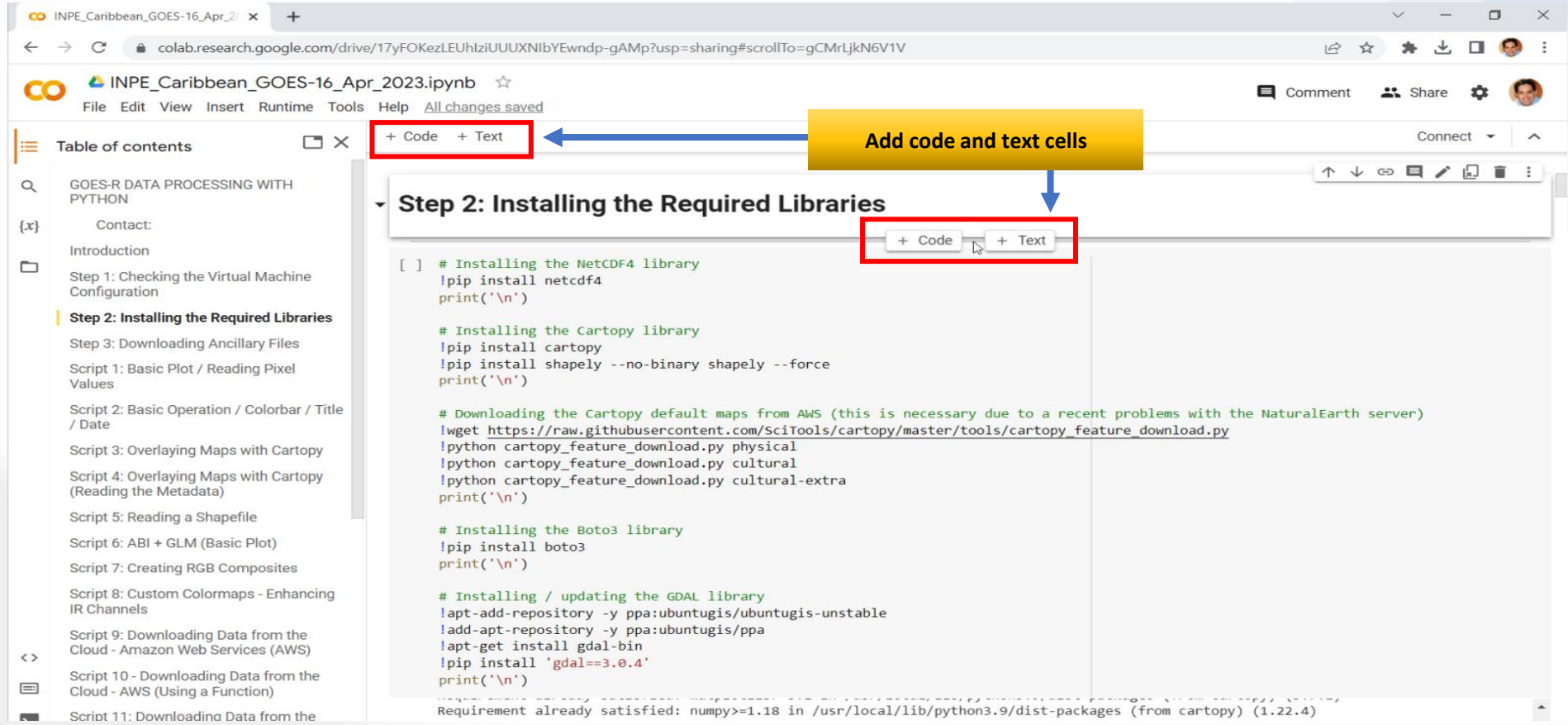
Step 2: Installing the Required Libraries

```
# Installing the NetCDF4 library
!pip install netcdf4
print('\n')

# Installing the Cartopy library
!pip install cartopy
!pip install shapely --no-binary shapely --force
print('\n')
```

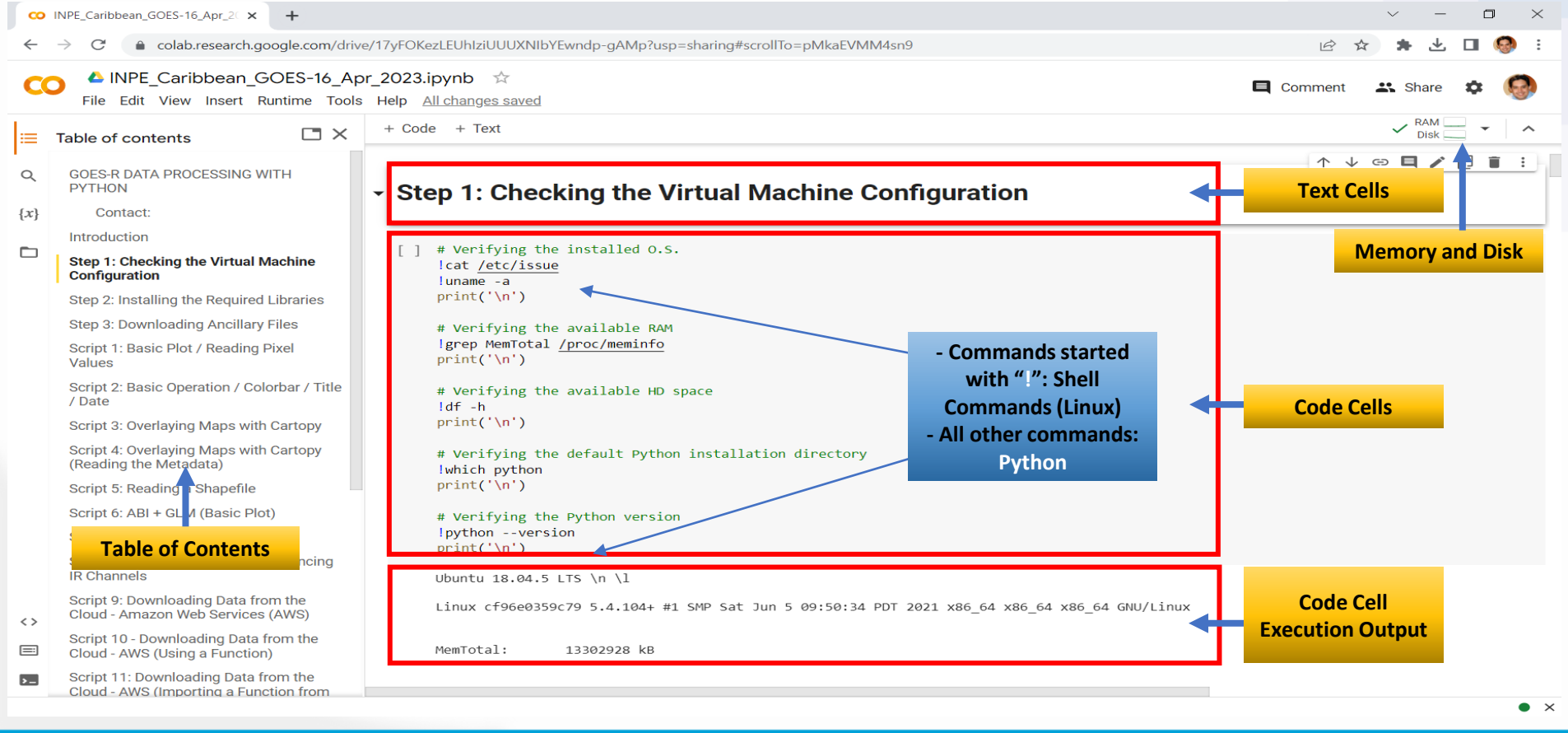
Step 3: Downloading Ancillary Files

```
# Download the utilities.py script, created by INPE (con
!wget -c https://www.dropbox.com/s/i8j1g9xg1g7o56l/utili
print('\n')

# Download a shapefile (world states and provinces)
!wget -c https://github.com/nvkelso/natural-earth-vector
!wget -c https://github.com/nvkelso/natural-earth-vector
```

Running Scripts in the Cloud

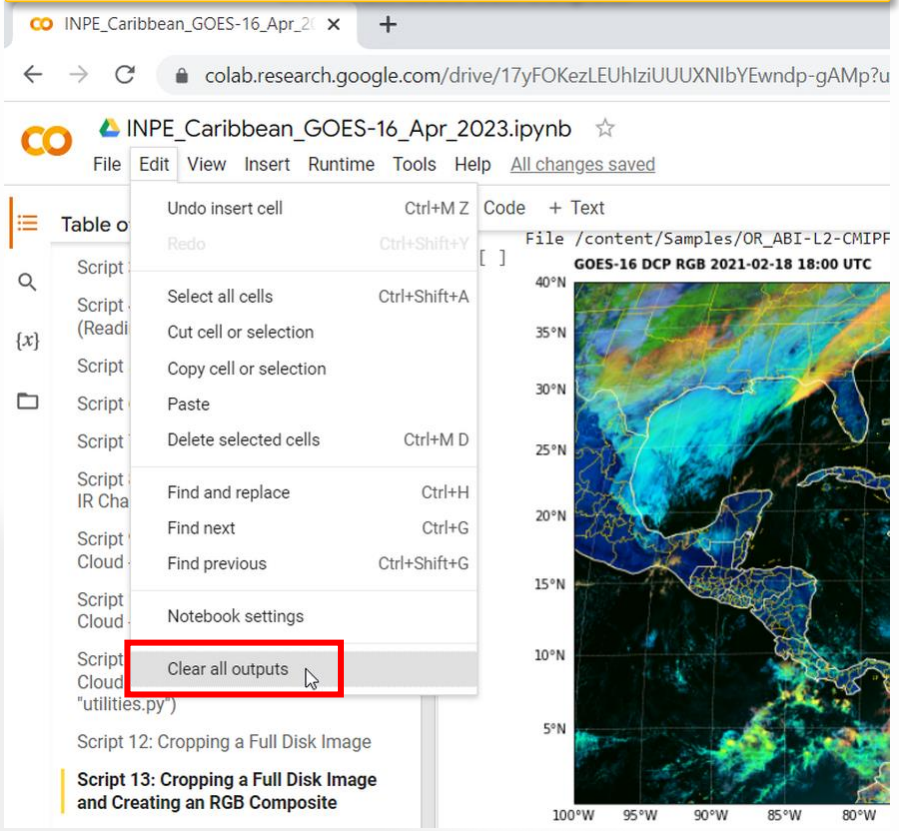


Running Scripts in the Cloud

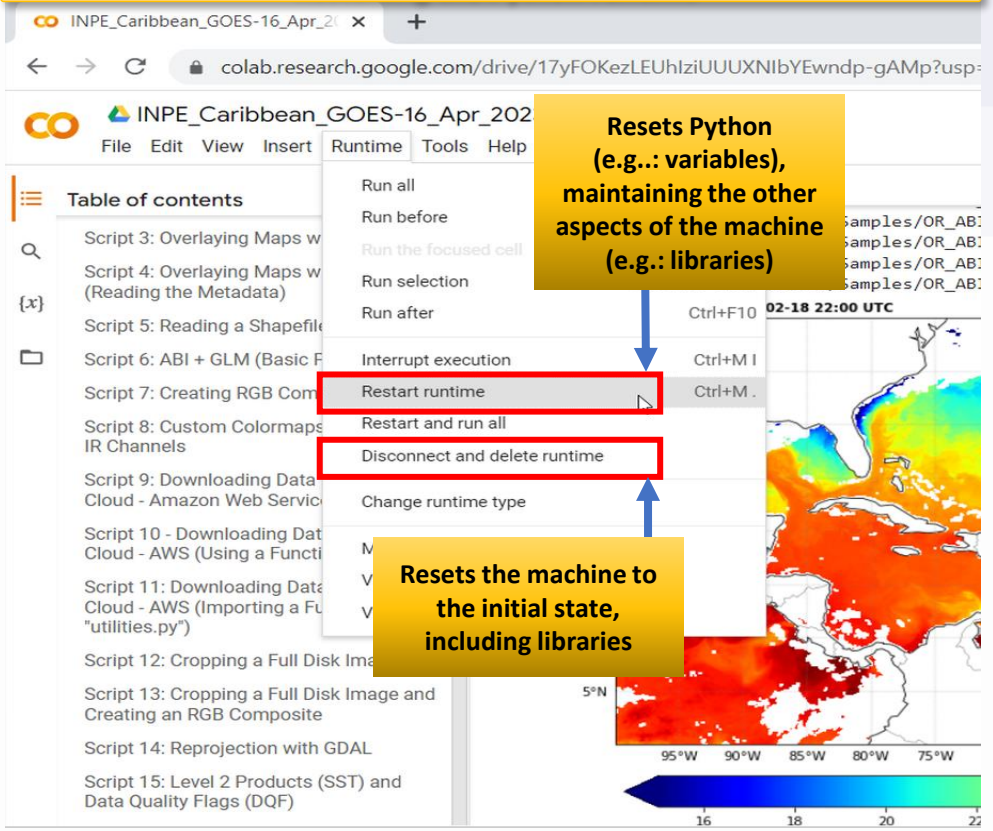


Running Scripts in the Cloud

Clears all code cell outputs



Restarting the Python runtime



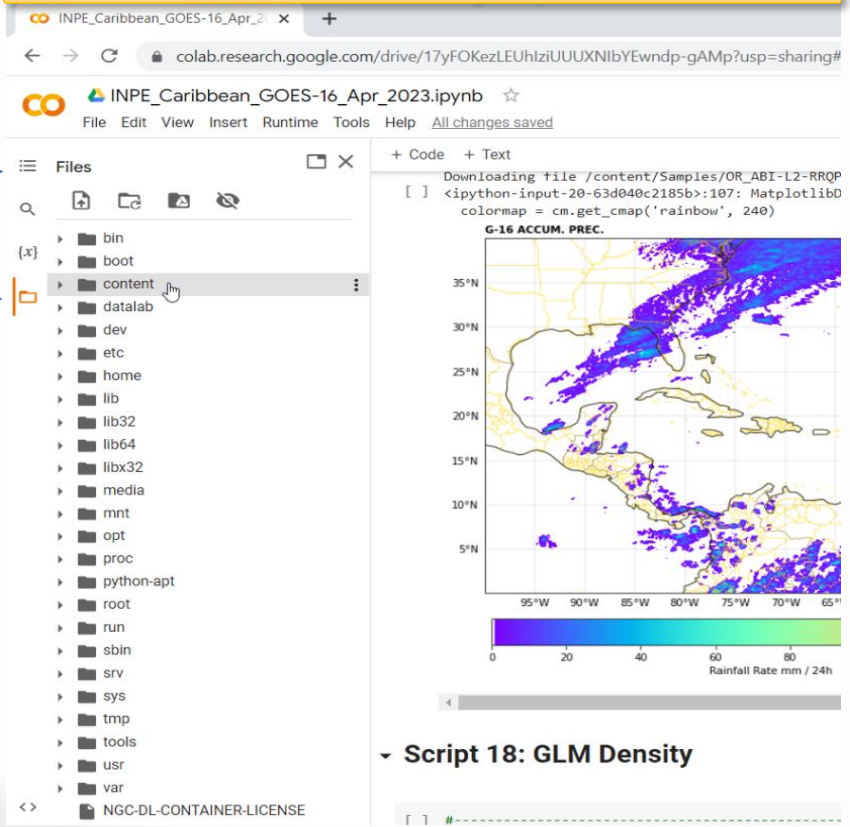
Running Scripts in the Cloud

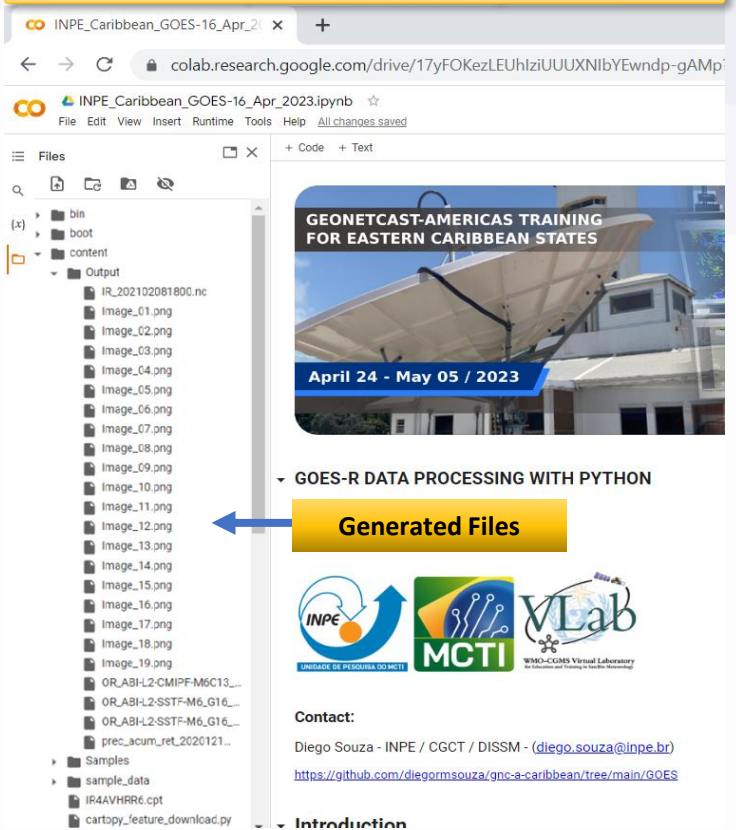
COLAB Work Directory: "/content"

"/content" Directory

View Table of Contents

View Directories





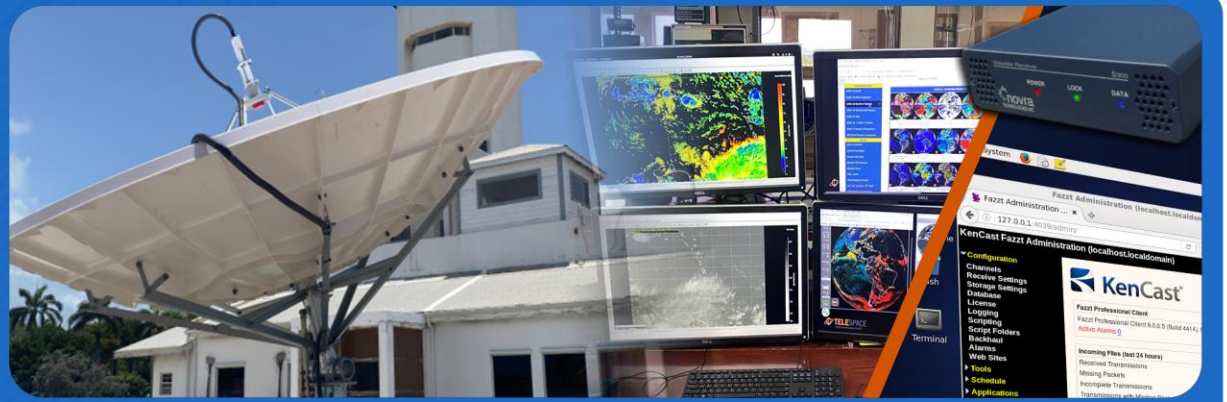
GEONETCast-Americas Training for the Eastern Caribbean States

Day 1 - May 3rd

Session 4:

Hands-on
Introduction to Python

THANK YOU! QUESTIONS?



Diego Souza
diego.souza@inpe.br

DISSM - Meteorological Satellites and Sensors' Division
CGCT - General Coordination of Earth Sciences
INPE - National Institute for Space Research

This is made possible by the generous support of the American people through the United States Agency for International Development (USAID)