

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



Caribbean Institute
for Meteorology
and Hydrology



Presentation Outline

- **Introduction**

- The Python programming language

- Useful libraries

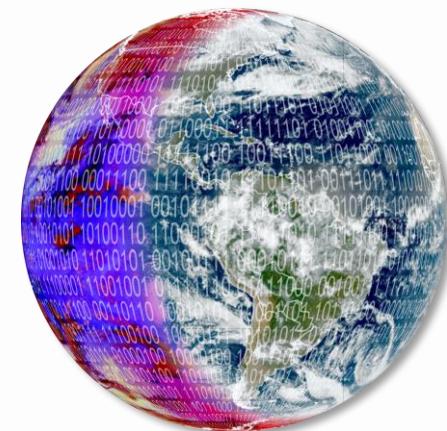
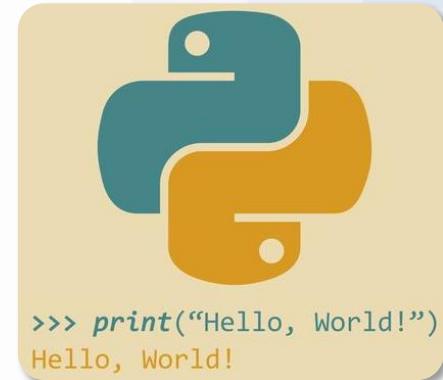
- **Application Examples**

- 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).



Some presentation examples:

DSAT

Application for interactive visualization of GOES-16 data and products.

Douglas Uba

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

2nd Teleconference of the WMO SDR Data Processing and Visualization Task Force, 16/02/2023, 17:00 UTC.



PyTROLL

Pytroll: an Open-Source Python Framework for Earth-Observing Satellite Data

...

Martin Raspaud, SMHI



martin.raspaud@smhi.se

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

Receiving Mechanisms and Python Libraries



PyProj



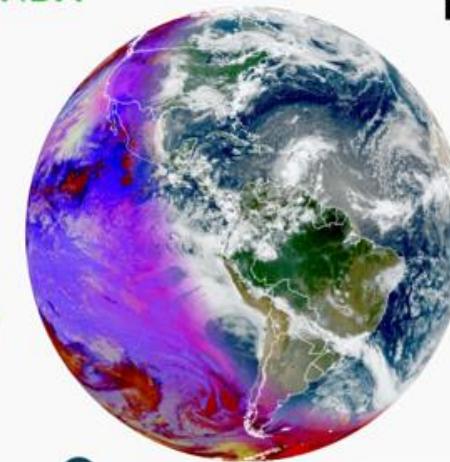
pyHDF



PySpectral



PyOrbital



Folium



matplotlib



pyGRIB



NetCDF4

Basemap

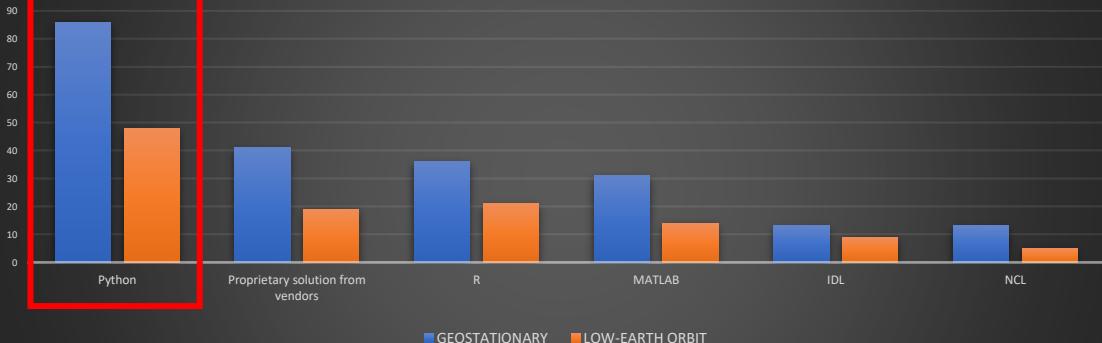
The Python Programming Language



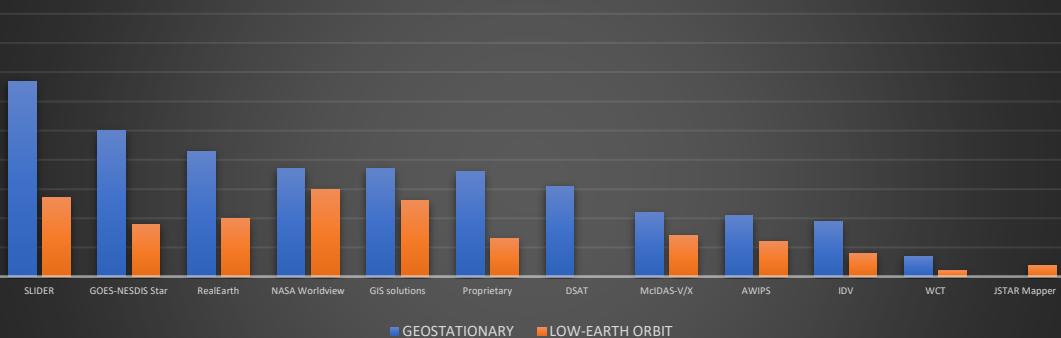
- Freely Available
- Easy to Get Started
- Gradual Learning Curve
- (Very) Active Community
- Many (Many!) Libraries / Modules
- Multiple Input and Output Formats
- Works on Different Operating Systems
- Integration with Other Software / Applications

Use of Python by the Community

9. Data processing - Please indicate the tools you currently use for GEO and LEO satellites



10. Data visualization - Please indicate the tools you currently use for GEO and LEO satellites



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

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

Different Satellites



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

Different Data Providers



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

Different Formats



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



THE GENERIC MAPPING TOOLS

Has a Python Interface



Has a "Jython" Interface



Has a Python Interface

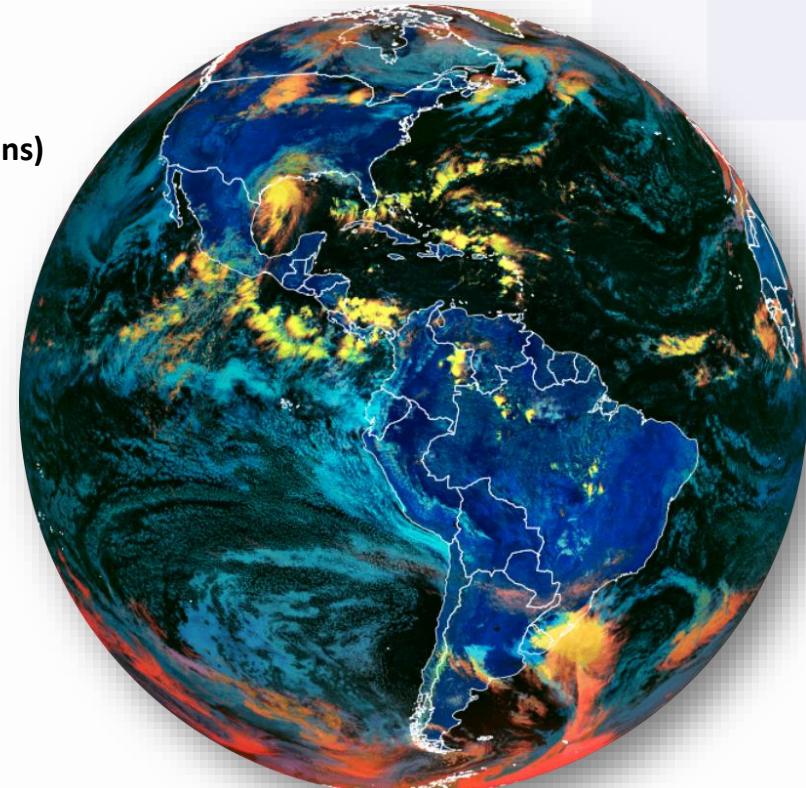


ILWIS

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:**

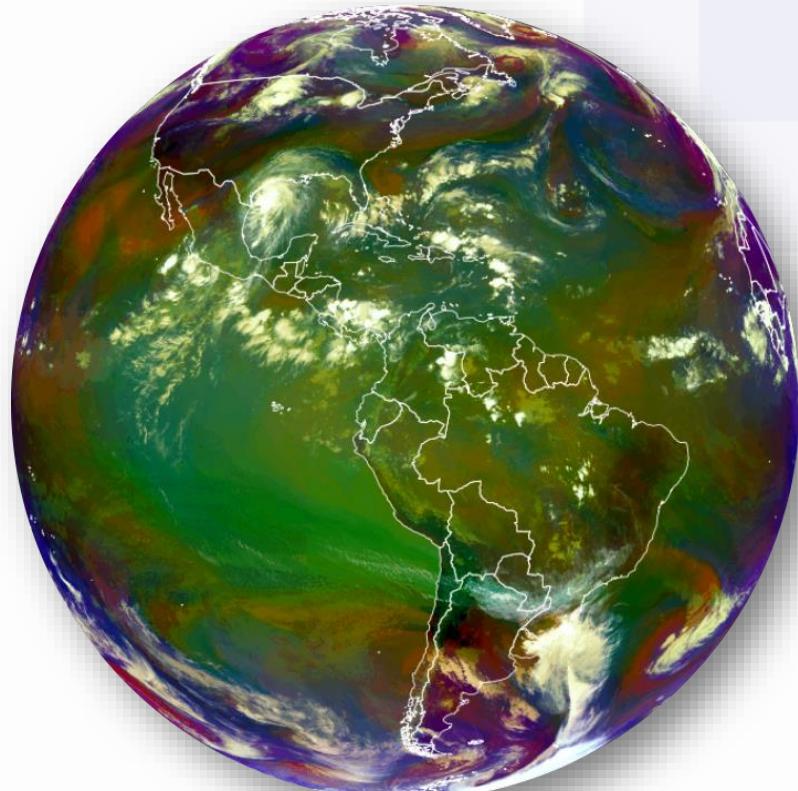
- Create a "stack" of bands to create RGBs

- Apply the necessary corrections

- Many other calculations

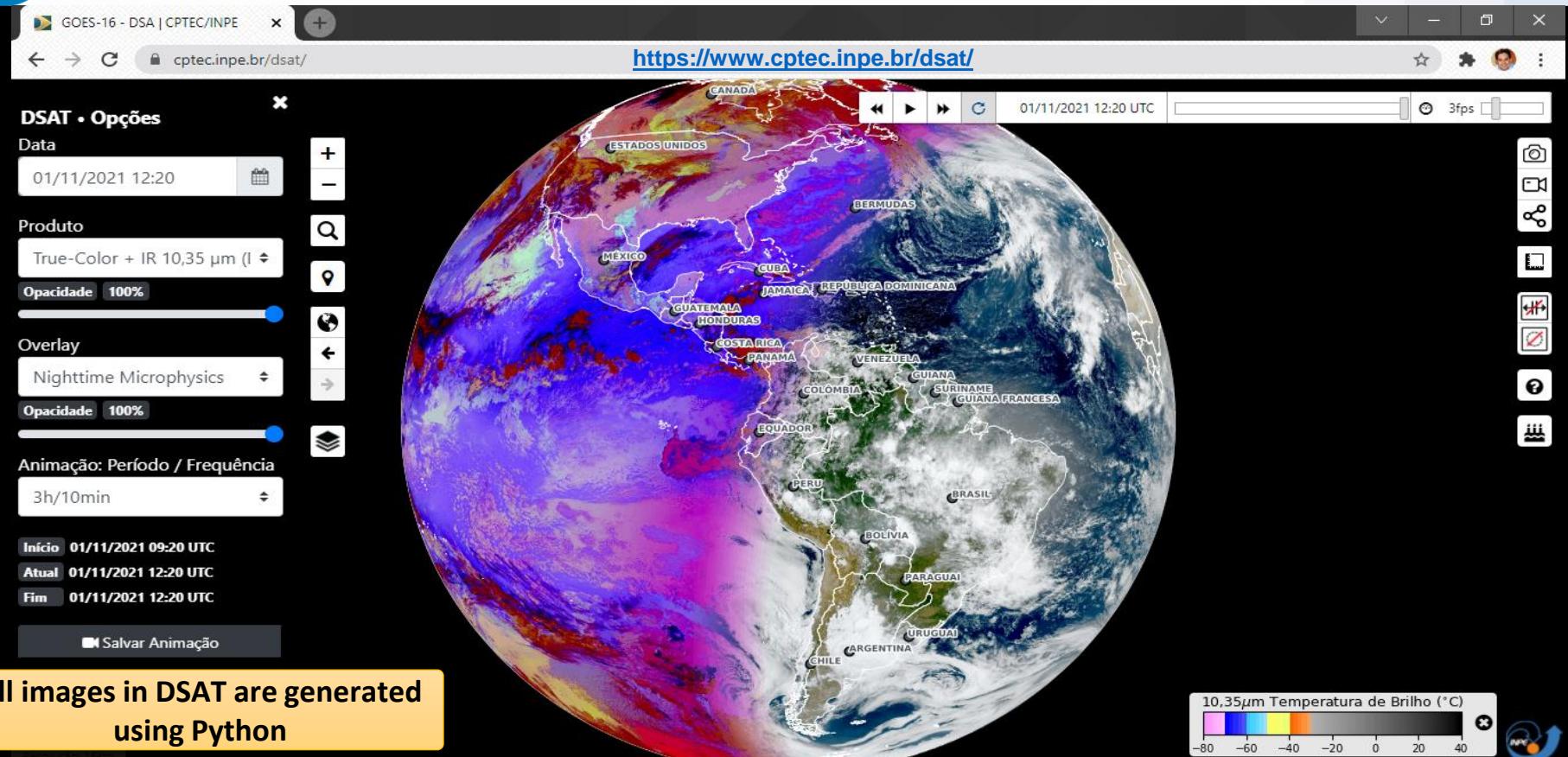
- **Cartopy:**

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



LET'S SEE SOME EXAMPLES

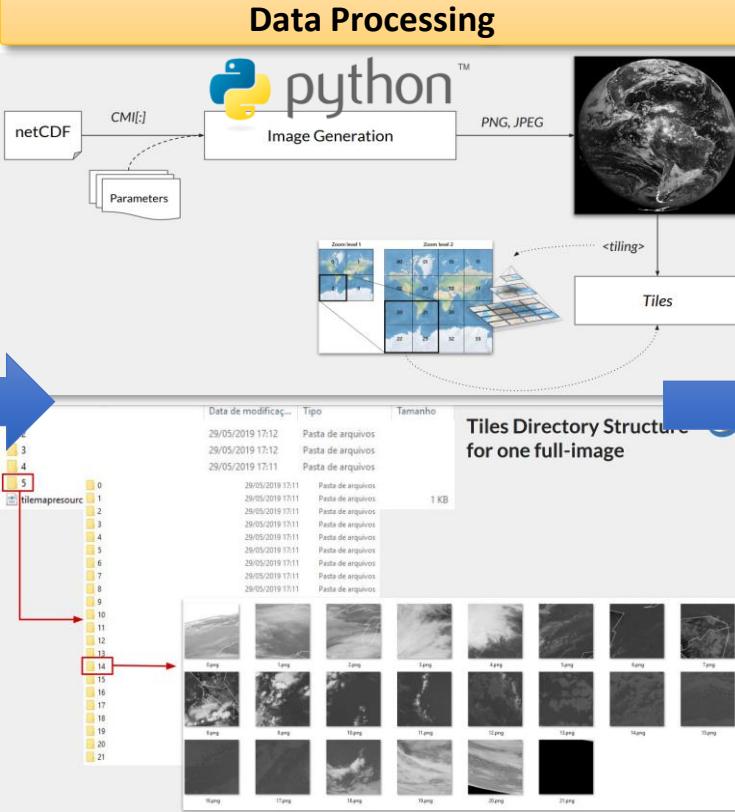
Application Examples: DSAT



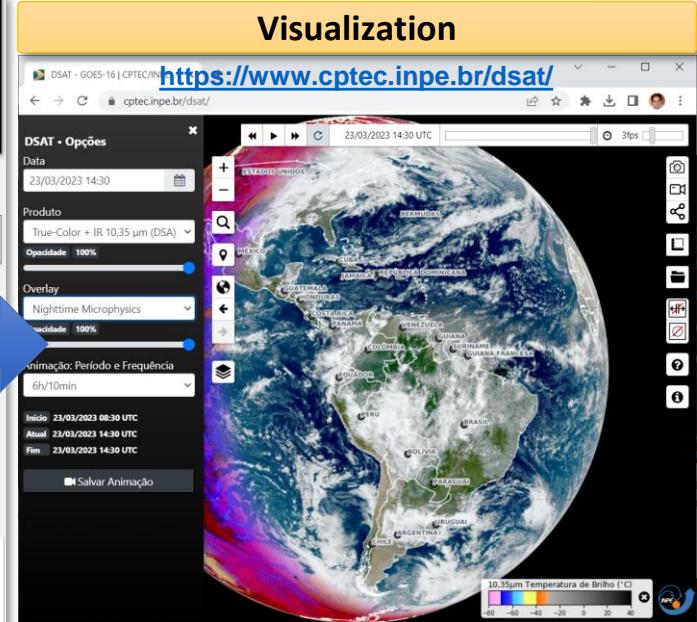


Application Examples: DSAT

Data Reception

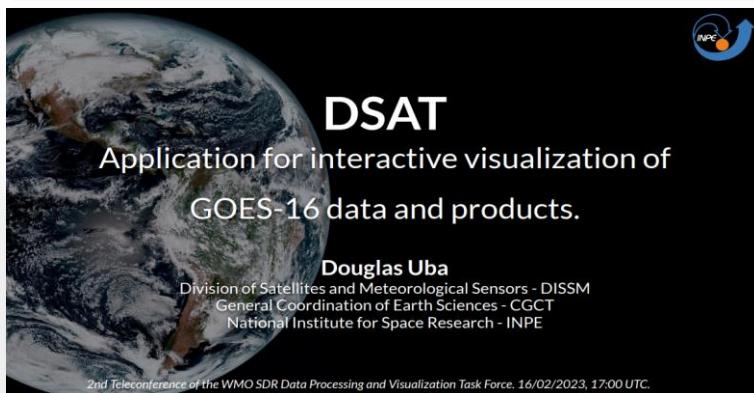


Visualization



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!

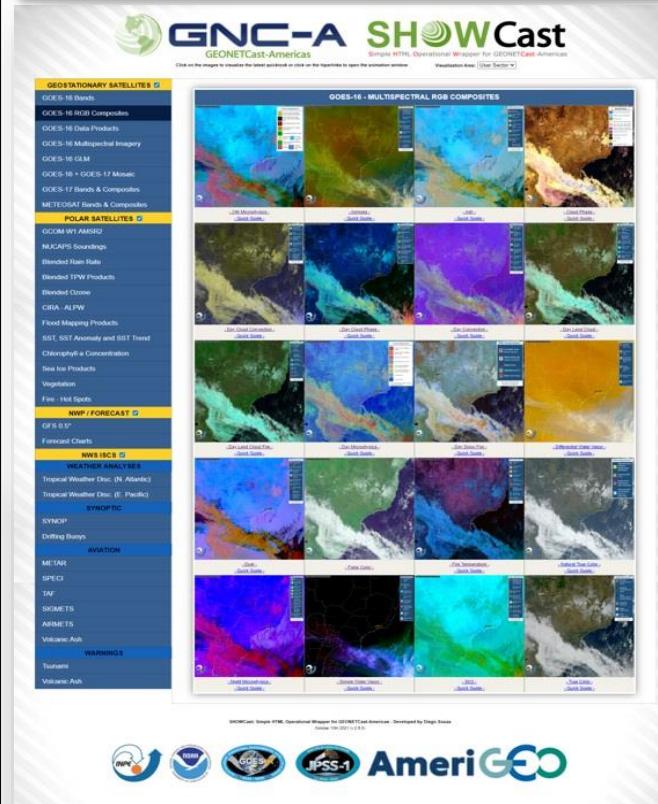
• Back-end (processing)

- 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 to the high temporal frequency of images, a lot (A LOT!) of tiles are generated daily.

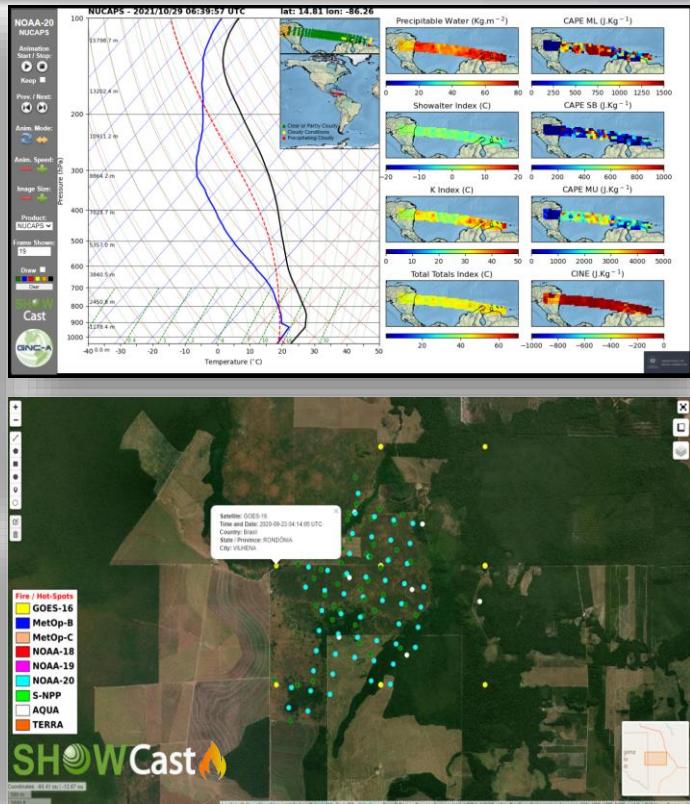
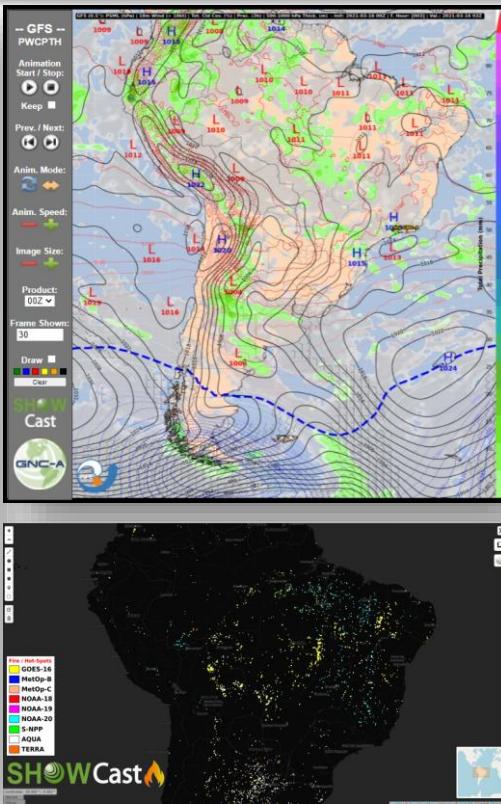
More details on the development of DSAT: <https://sdr.ucr.ac.cr/data-processing-and-visualization>

Application Examples: SHOWCast

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



The screenshot displays the GNC-A SHOWCast interface. On the left, there is a sidebar with links to various satellite datasets: GEOSTATIONARY SATELLITES (GOES-16 Bands, GOES-16 RGB Composites, GOES-16 Data Products, GOES-16 Multispectral Imagery, GOES-16 GLM, GOES-16 GOES-17 Mosaic, GOES-17 Bands & Composites, METEOSAT Bands & Composites), POLAR SATELLITES (GCCM, NUCAPS, NUCAPS Soundings, Blended Rain Rate, Blended TPW Products, Blended Ozone, CIRA-ALPW, Flood Mapping Products, SST, SST Anomaly and SST Trend, Chlorophyll Concentration, Sea Ice Products, Vegetation, Fire / Hot Spots), NWP / FORECAST (GFS 0.9°, Forecast Charts), NWS (ISOB), WEATHER ANALYSES (Tropical Weather Disc. (N. Atlantic), Tropical Weather Disc. (E. Pacific)), SYNOP, Drifting Buoys, METAR, SPECI, TAF, SIGMETs, SIGMETs, Volcanic Ash, Volcanic Ash, Volcanic Ash, and a section for WARNINGs (Volcanic Ash, Tsunami). The main area shows a grid of satellite imagery for GOES-16, including Multispectral RGB Composites and various data products. Below the grid, there is a legend for fire/hot spots and logos for AmeriGEO and other agencies.

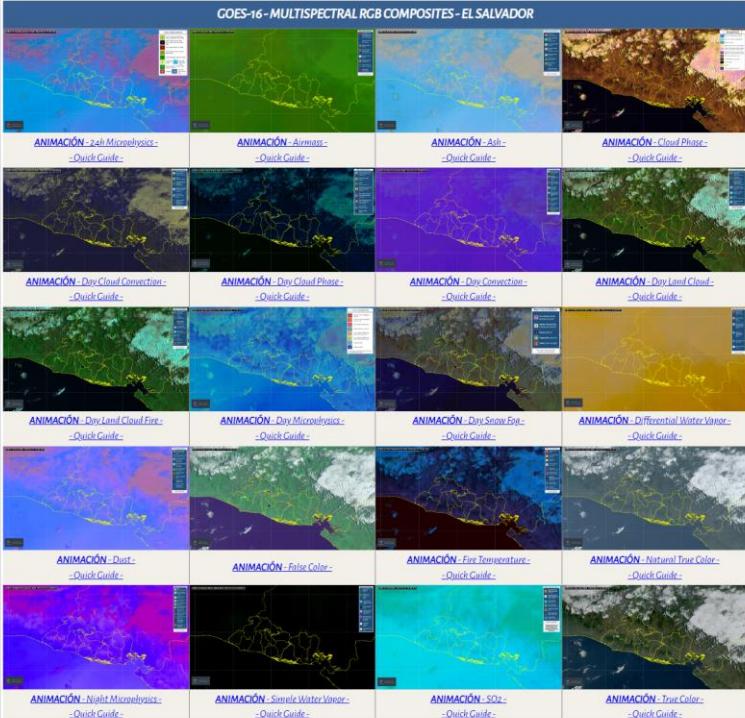


Application Examples: MARN

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

RGB Composites

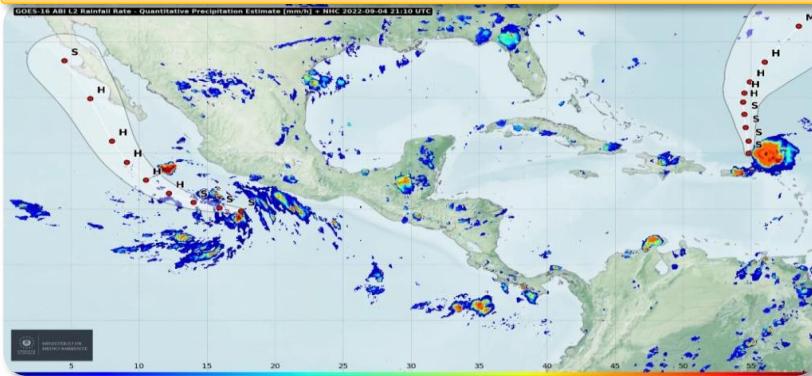
- GEOSTATIONARY - El Salvador
 - GOES-16 ABI Bands
 - GOES-16 RGB Composites
 - GOES-16 Baseline Products
 - GOES-16 Saharan Air Layer Tracking
 - GOES-16 Geostationary Lightning Mapper
- GEOSTATIONARY - Central America
 - GOES-16 ABI Bands
 - GOES-16 RGB Composites
- NHC Tracking
 - GOES-16 + NHC
- POLAR
 - NUCAPS Soundings - El Salvador
 - Blended TPW Products - El Salvador
 - Blended TPW Products - Central America
 - CIRA - ALPW - Central America
 - Flood Mapping Products - El Salvador
 - SST, SST Anomaly, SST Trend - El Salvador
 - SST, SST Anomaly, SST Trend - Central America
 - Chlorophyll-a Concentration - El Salvador
 - Vegetation - Central America
 - Fire - Hot Spots
- NWP / FORECAST
 - GFS 0.5° plots - Caribbean



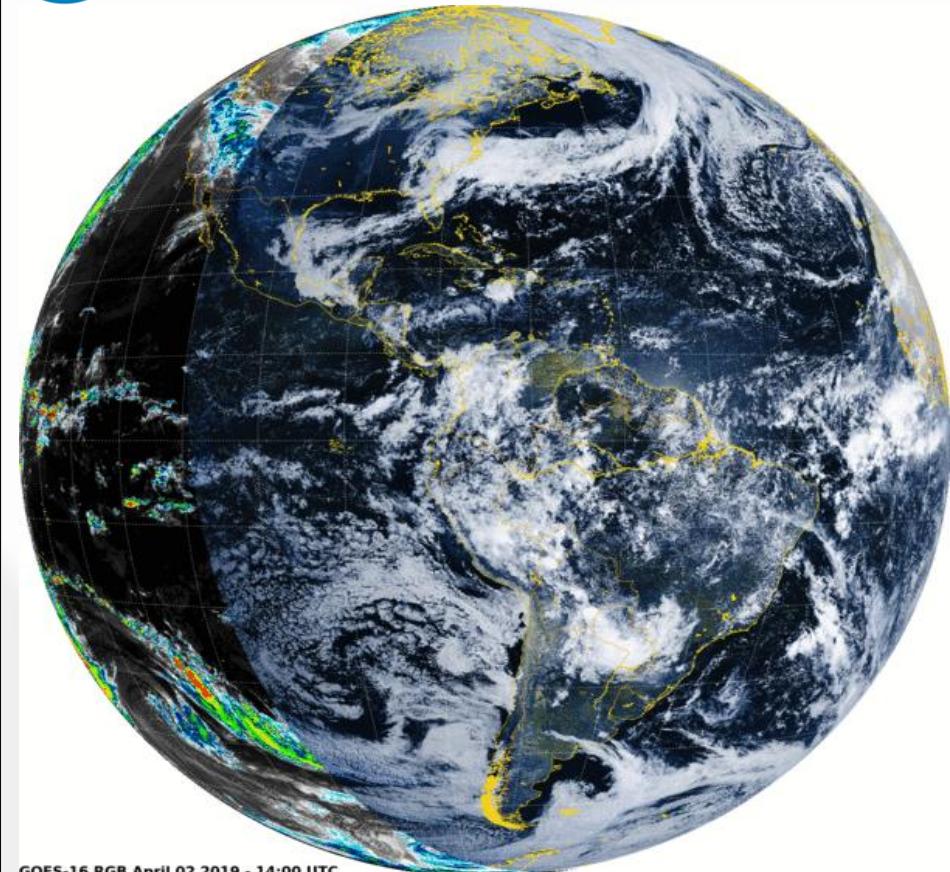
True Color RGB + Tracking NHC



RRQPEF + Tracking NHC



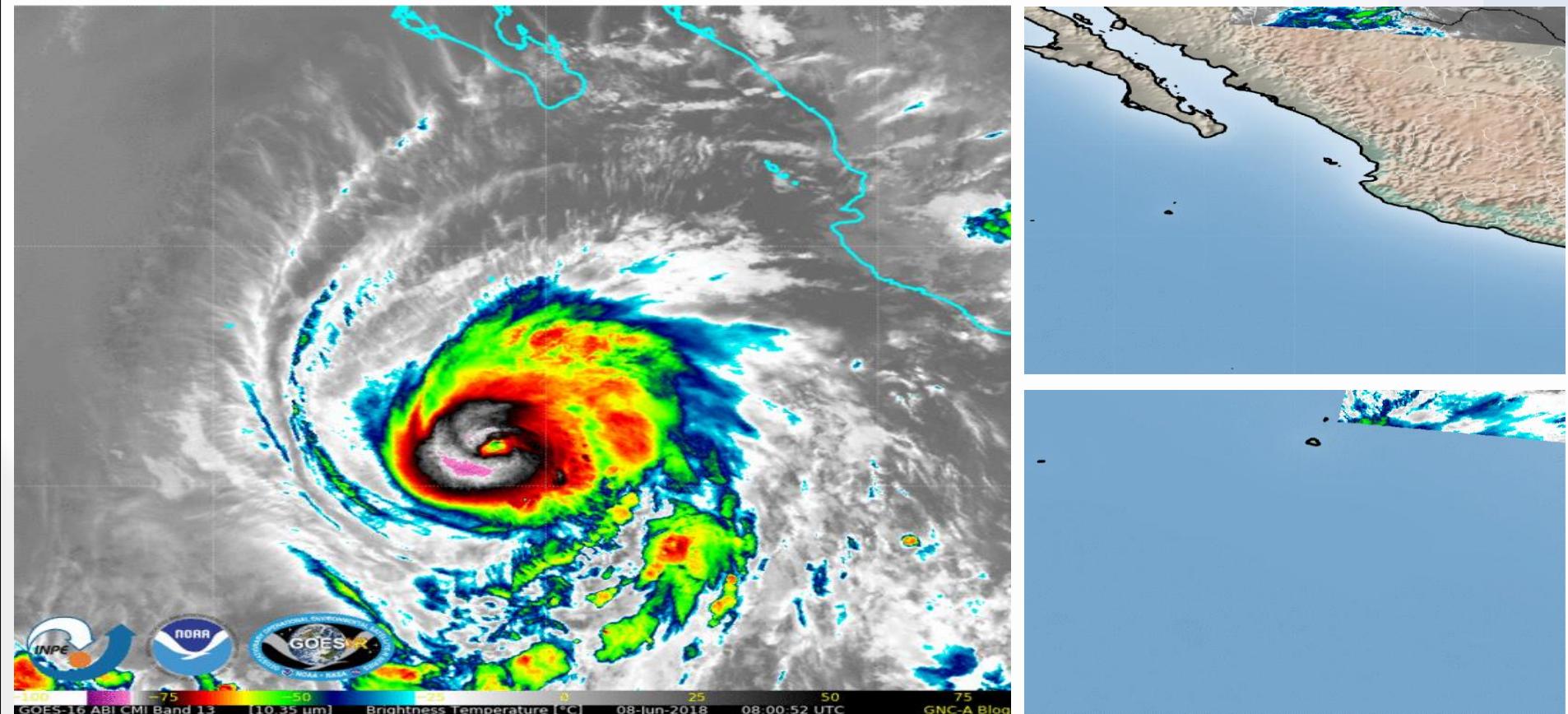
Application Examples: GEO and LEO



GOES-16 RGB April 02 2019 - 14:00 UTC

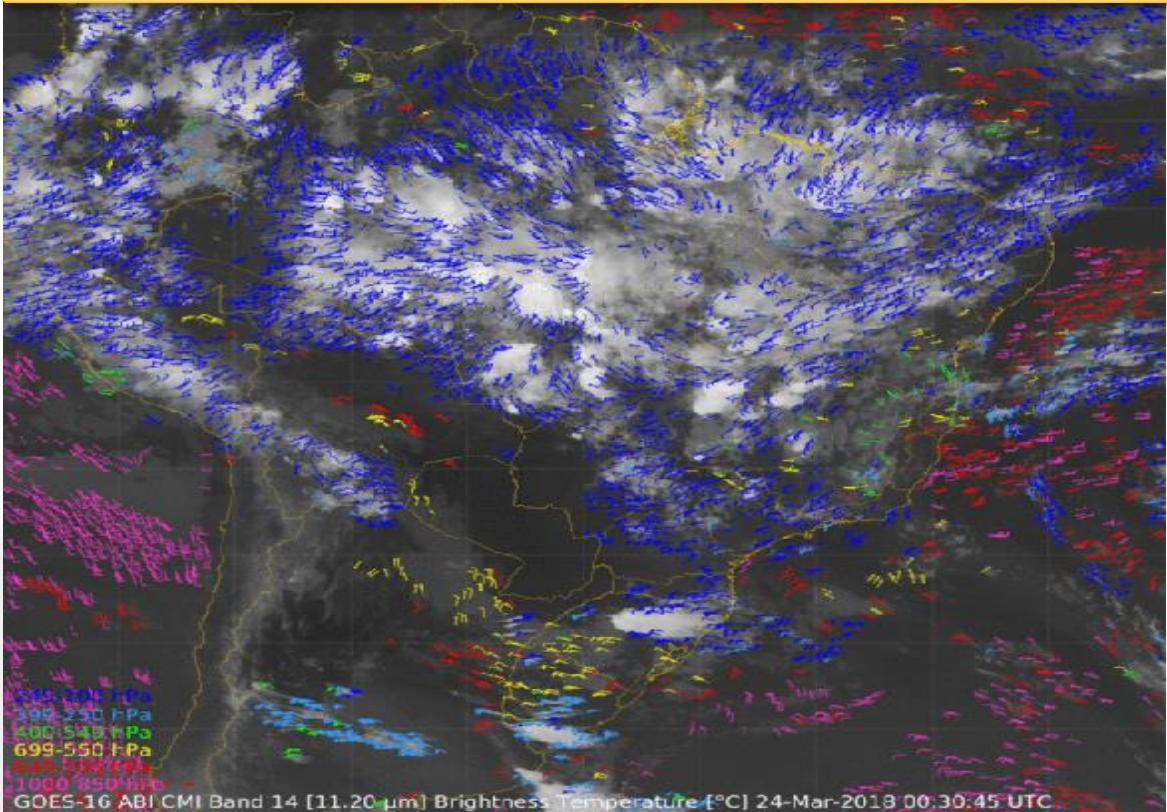


Application Examples: GEO and LEO



Application Examples: GEO and LEO

Vientos Derivados por Satélite (ABI - GOES-16)

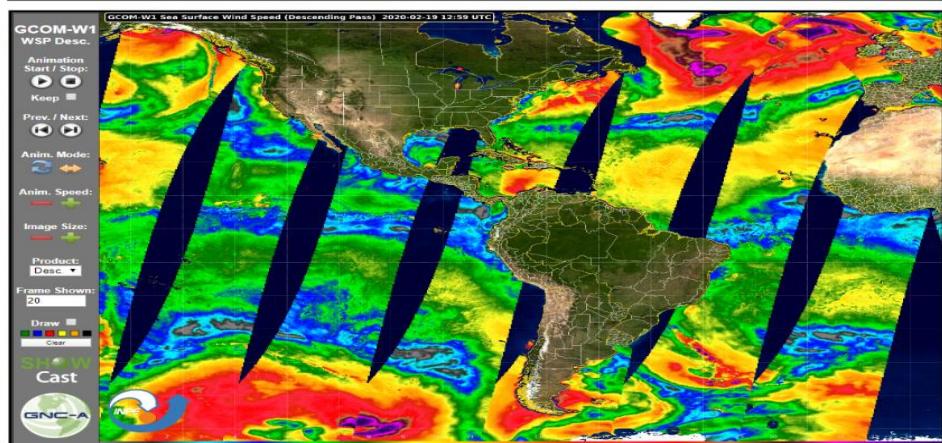
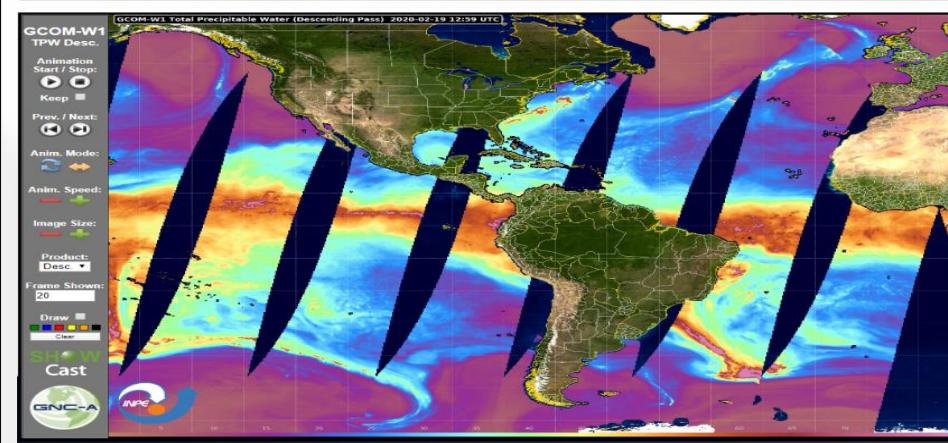
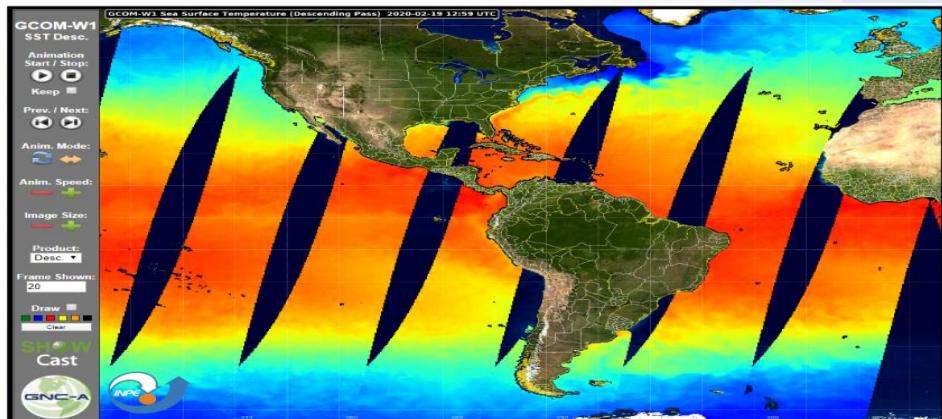
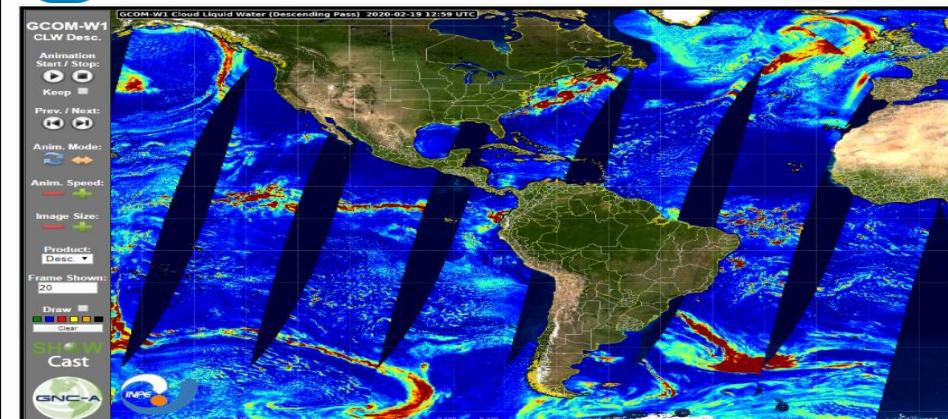


Banda Día Noche (VIIRS - NPP)

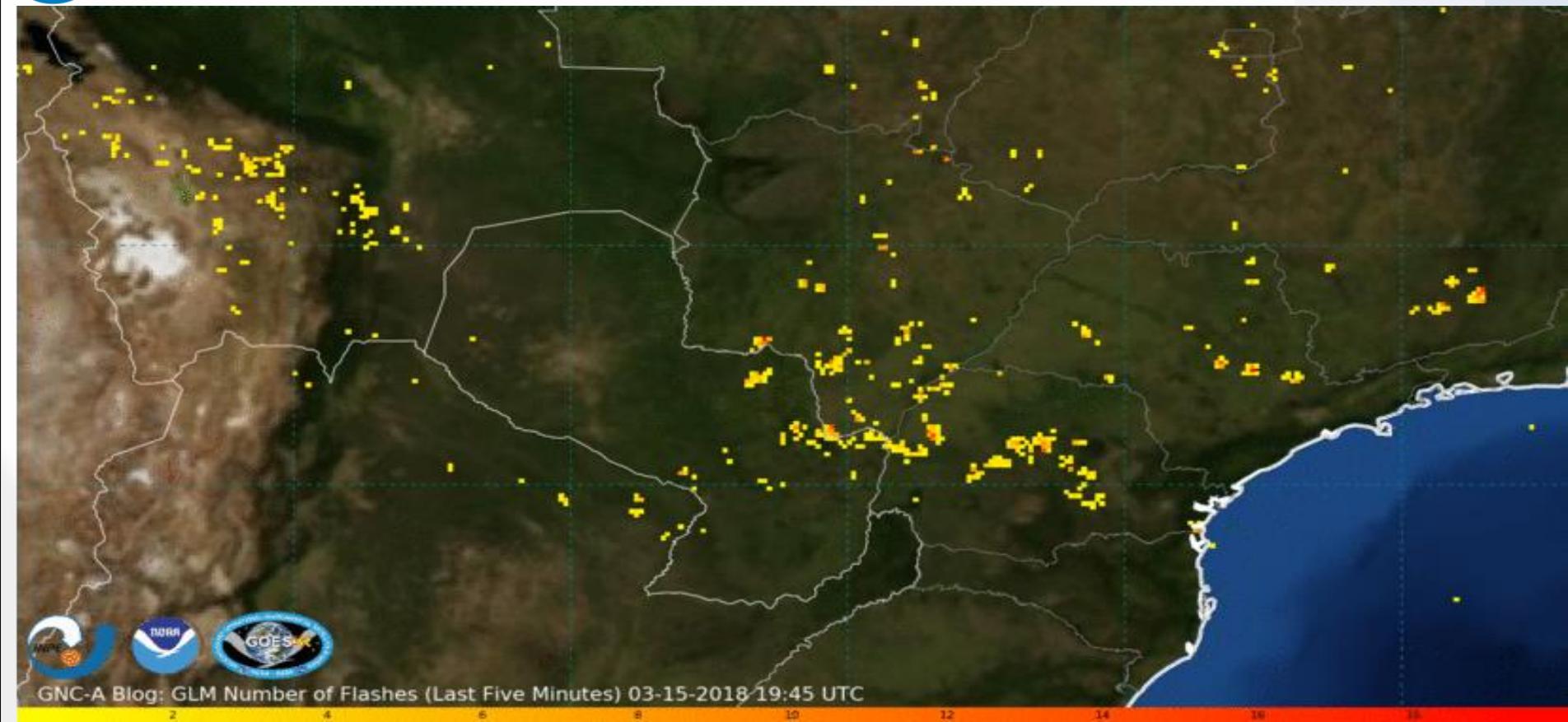




Application Examples: LEO (GCOM)

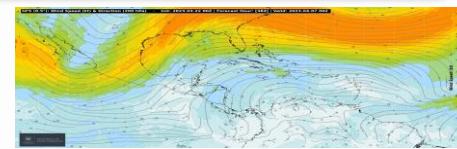
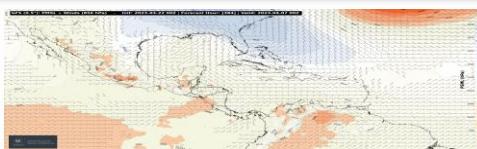
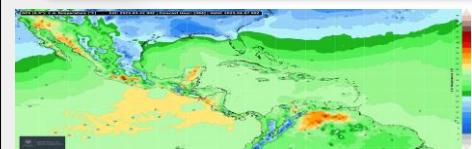
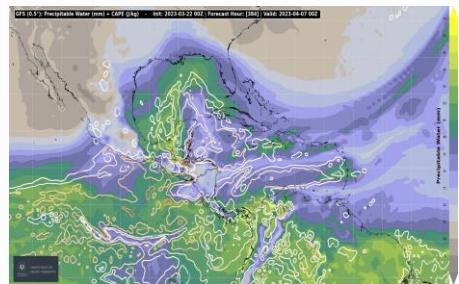
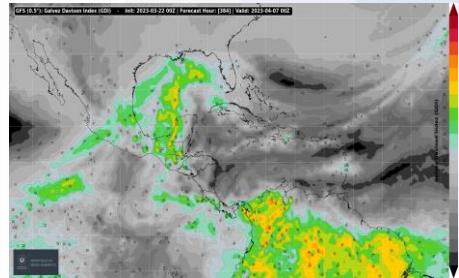
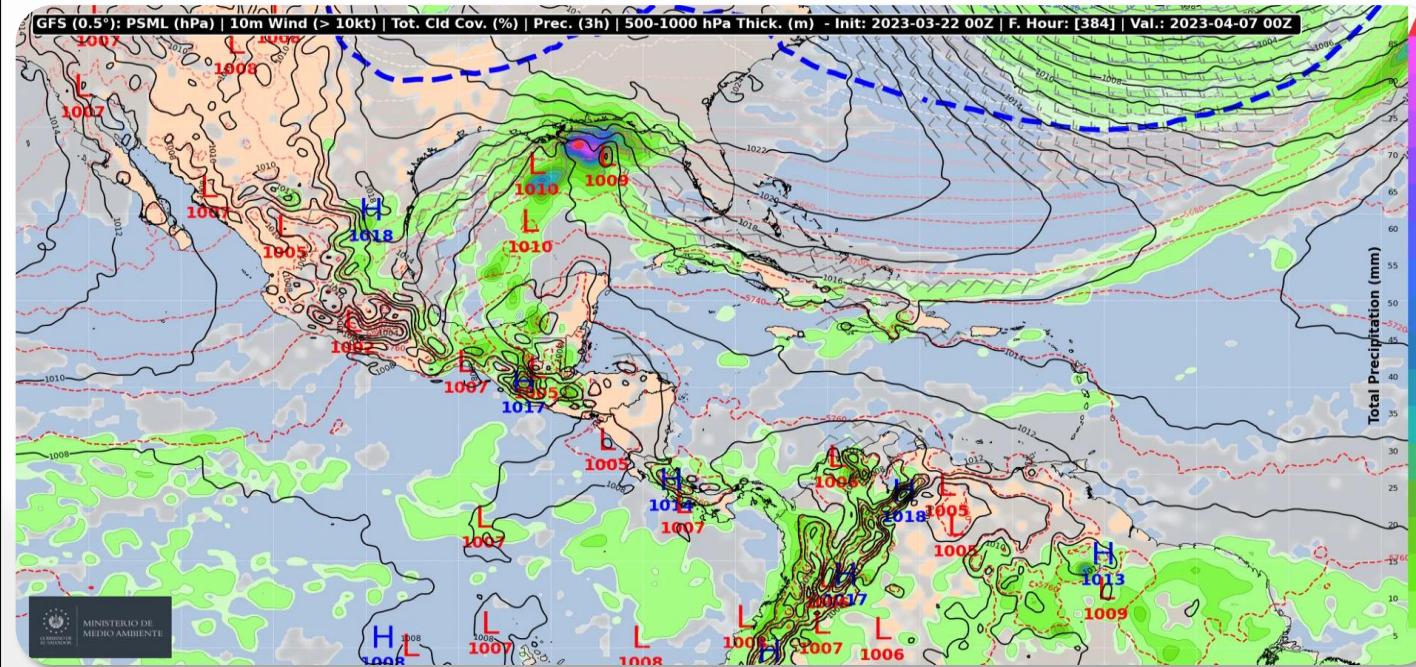


Application Examples: GLM



Application Examples: NWP

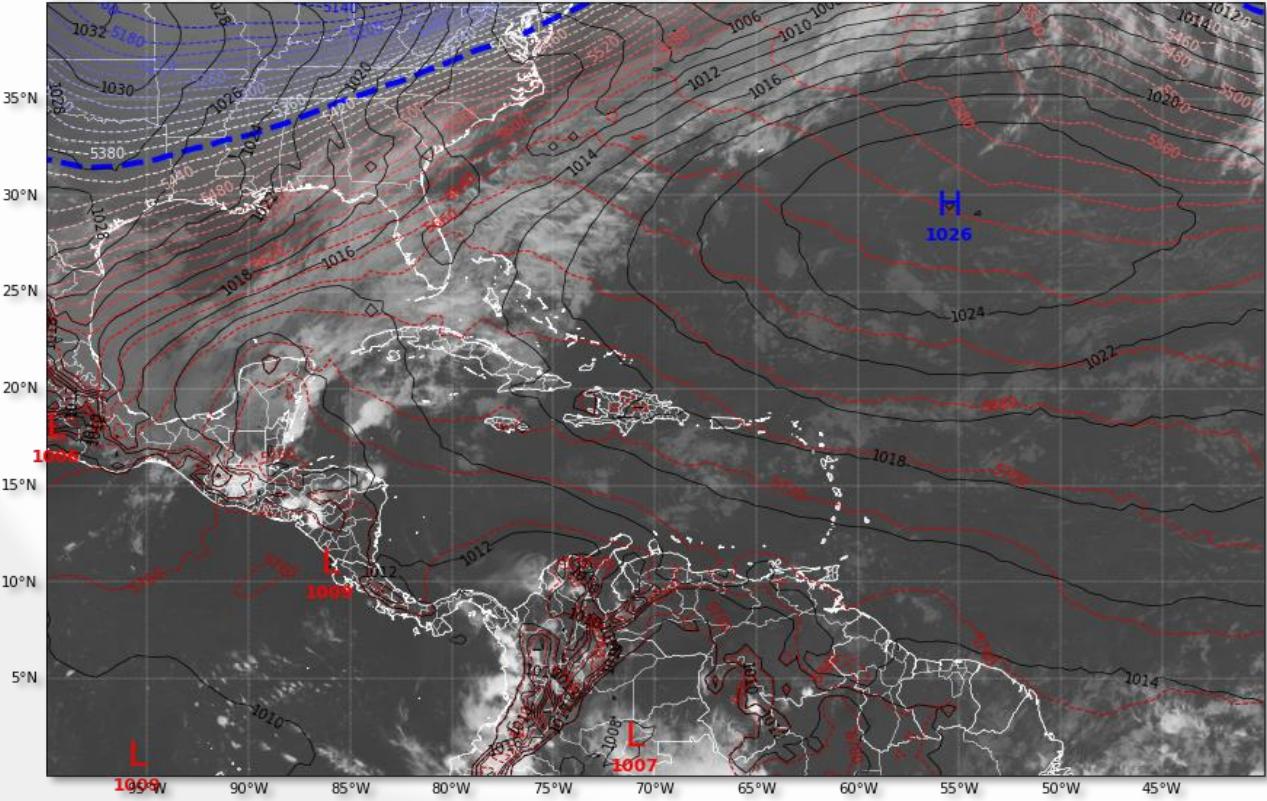
http://srt.marn.gob.sv/SHOWCast/HTML/SHOWCast_SEC.html



Application Examples: Satellite + NWP

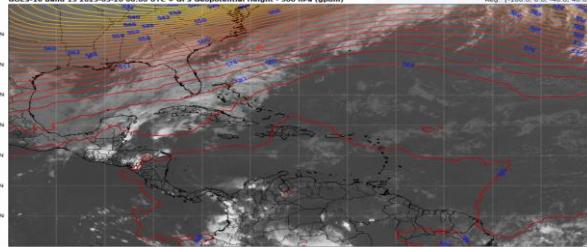
<https://geonetcast.wordpress.com/2021/07/27/vlab-processamento-de-dados-de-modelos-de-previsao-numerica-do-tempo-conteudo-completo/>

GOES-16 Band 13 2023-03-20 00:00 UTC + GFS PMSL (hPa) + 1000-500 hPa Thickness (m)



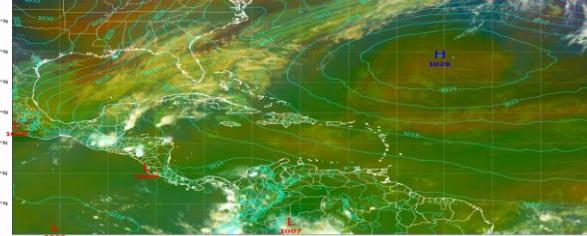
Reg.: [-100.0, 0.0, -40.0, 40.0]

GOES-16 Band 13 2023-03-20 00:00 UTC + GFS Geopotential Height - 500 hPa (gpm)



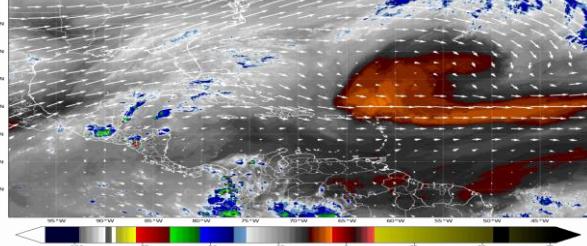
Reg.: [-100.0, 0.0, -40.0, 40.0]

GOES-16 Airmass RGB 2023-03-20 00:00 UTC + GFS PMSL (hPa)



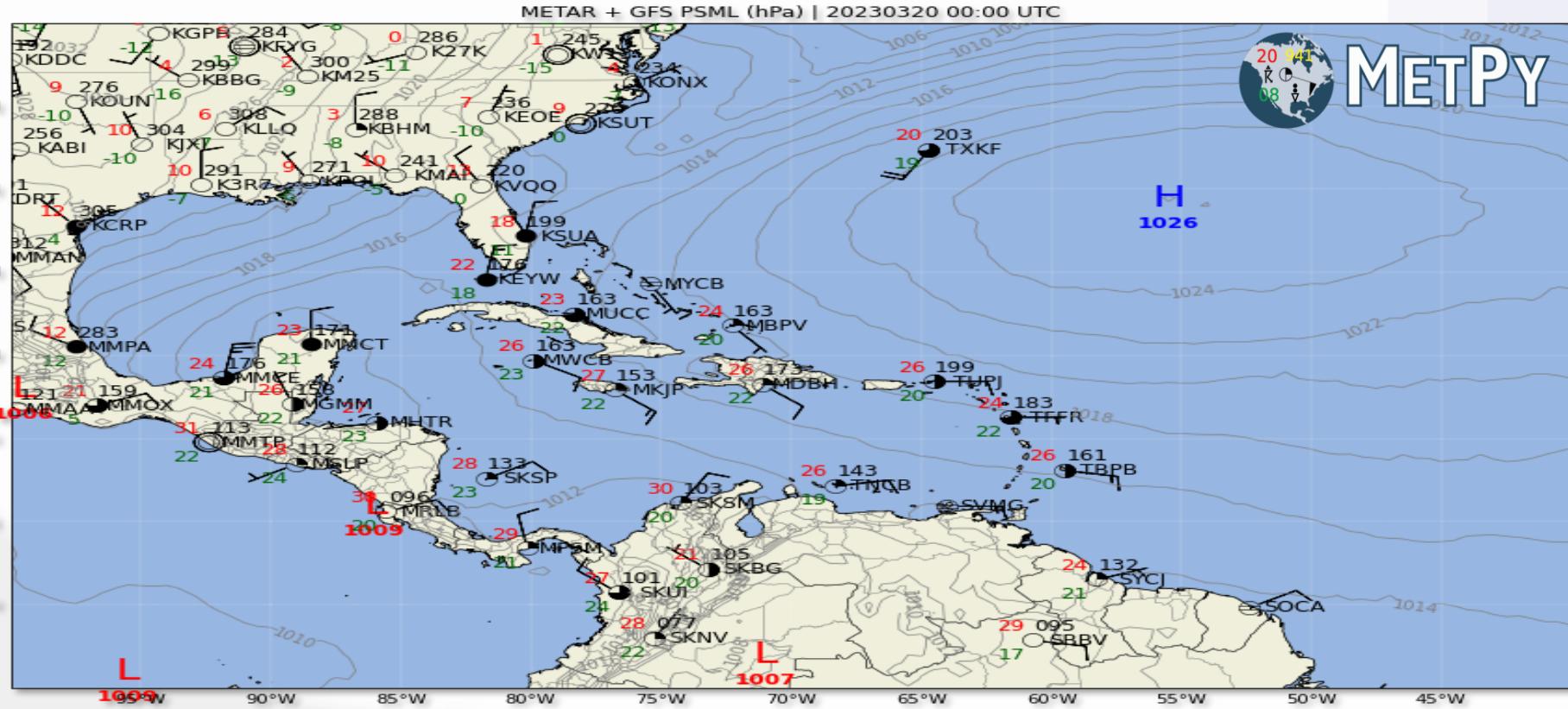
Reg.: [-100.0, 0.0, -40.0, 40.0]

GOES-16 Band 09 2023-03-20 00:00 UTC + GFS Wind Speed and Direction (250 hPa)



Application Examples: METAR

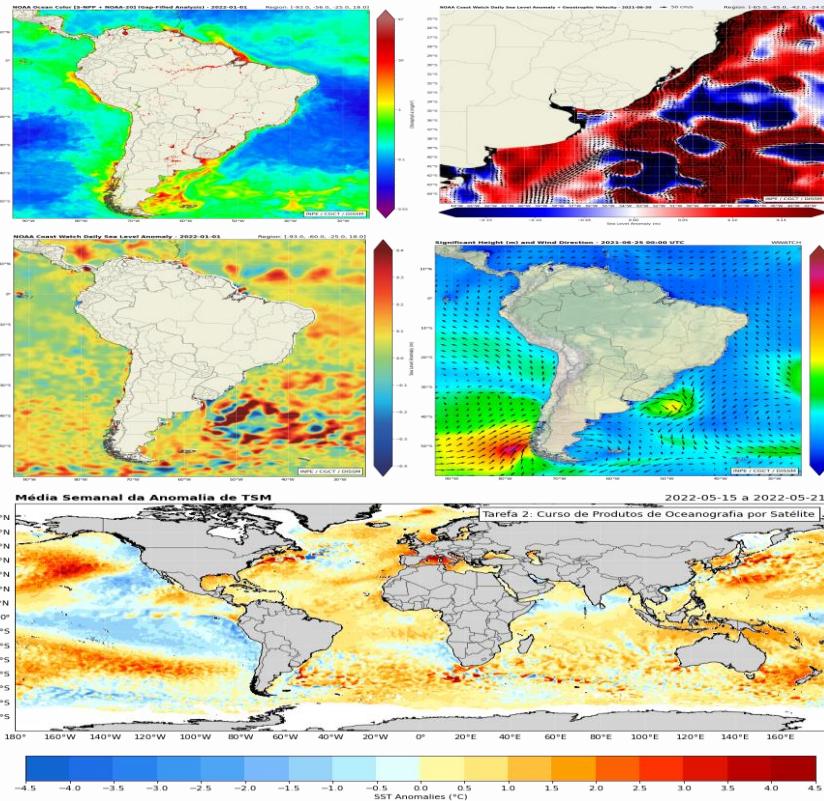
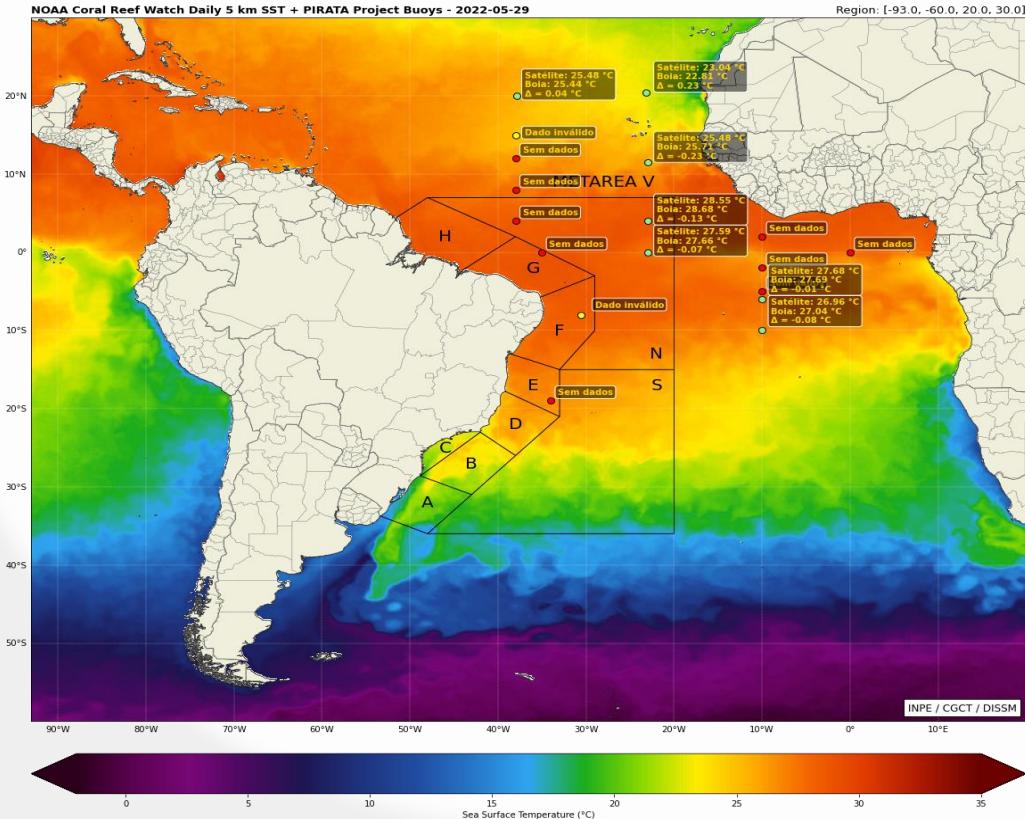
<https://geonetcast.wordpress.com/2021/07/27/vlab-processamento-de-dados-de-modelos-de-previsao-numerica-do-tempo-conteudo-completo/>



Application Examples: Ocean

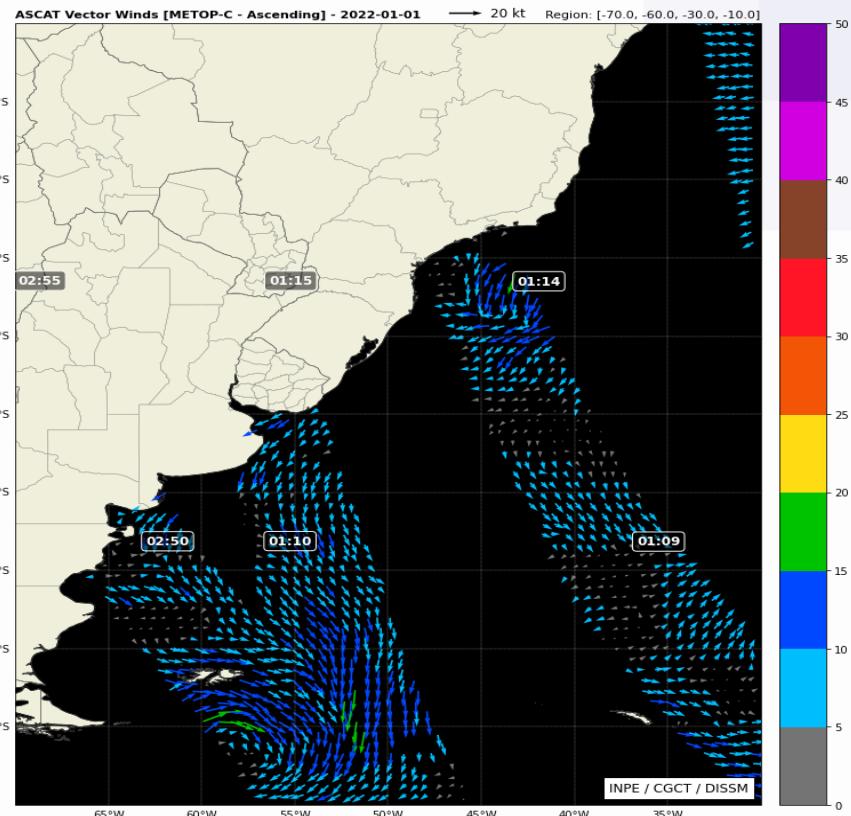
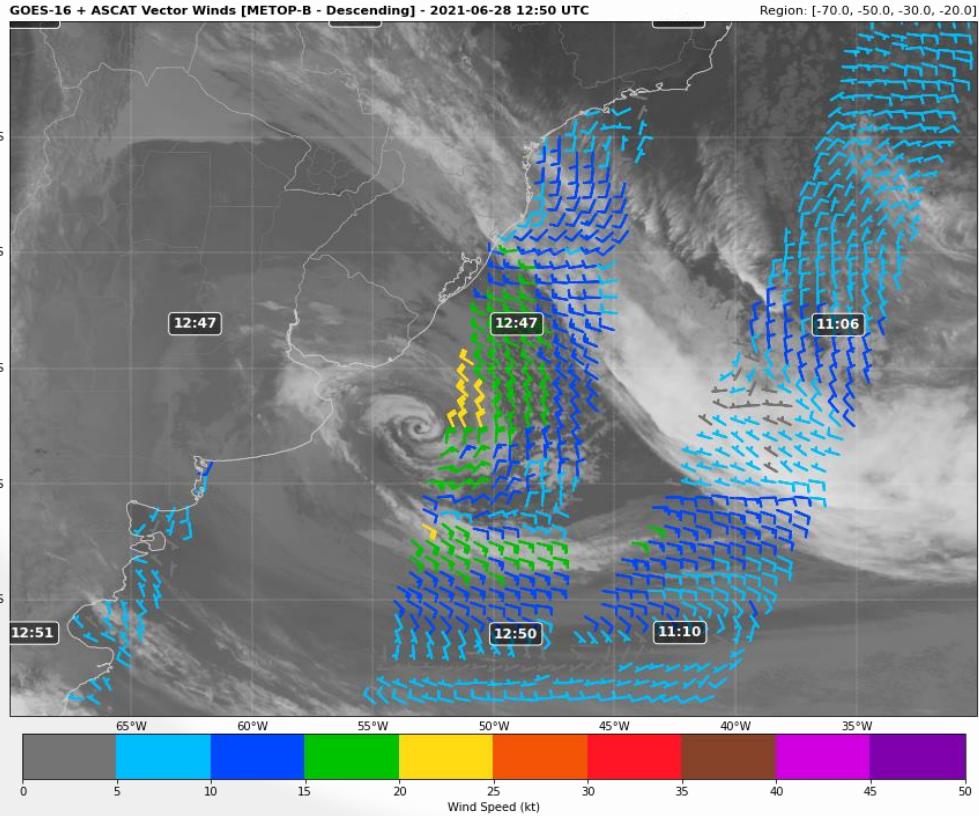
<https://geonetcast.wordpress.com/2022/06/08/vlab-produtos-de-oceanografia-por-satelite-conteudo-completo/>

NOAA Coral Reef Watch Daily 5 km SST + PIRATA Project Buoys - 2022-05-29



Application Examples: ASCAT Winds

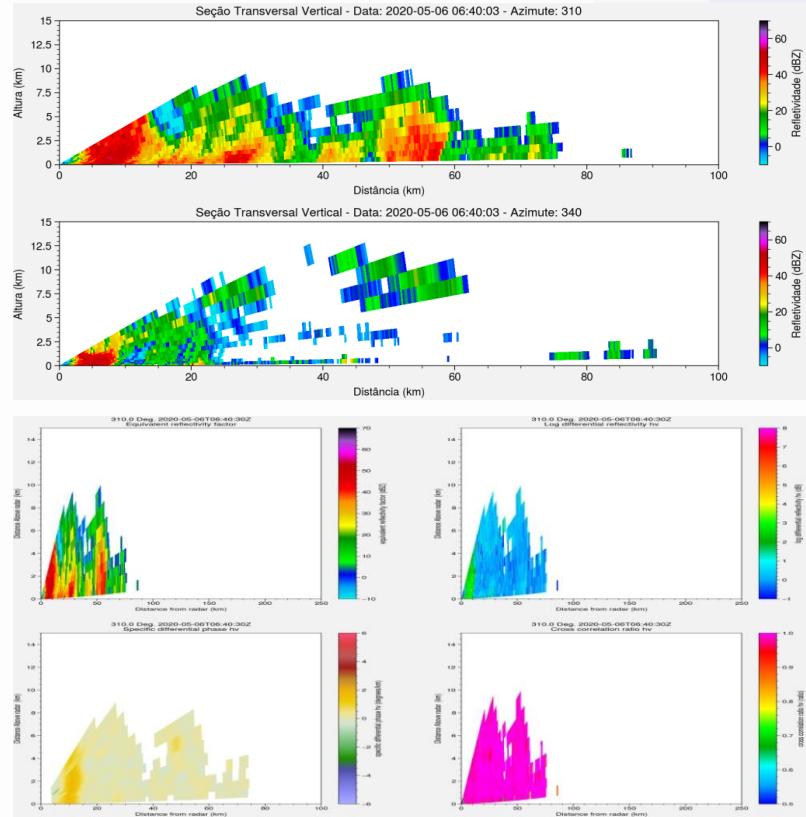
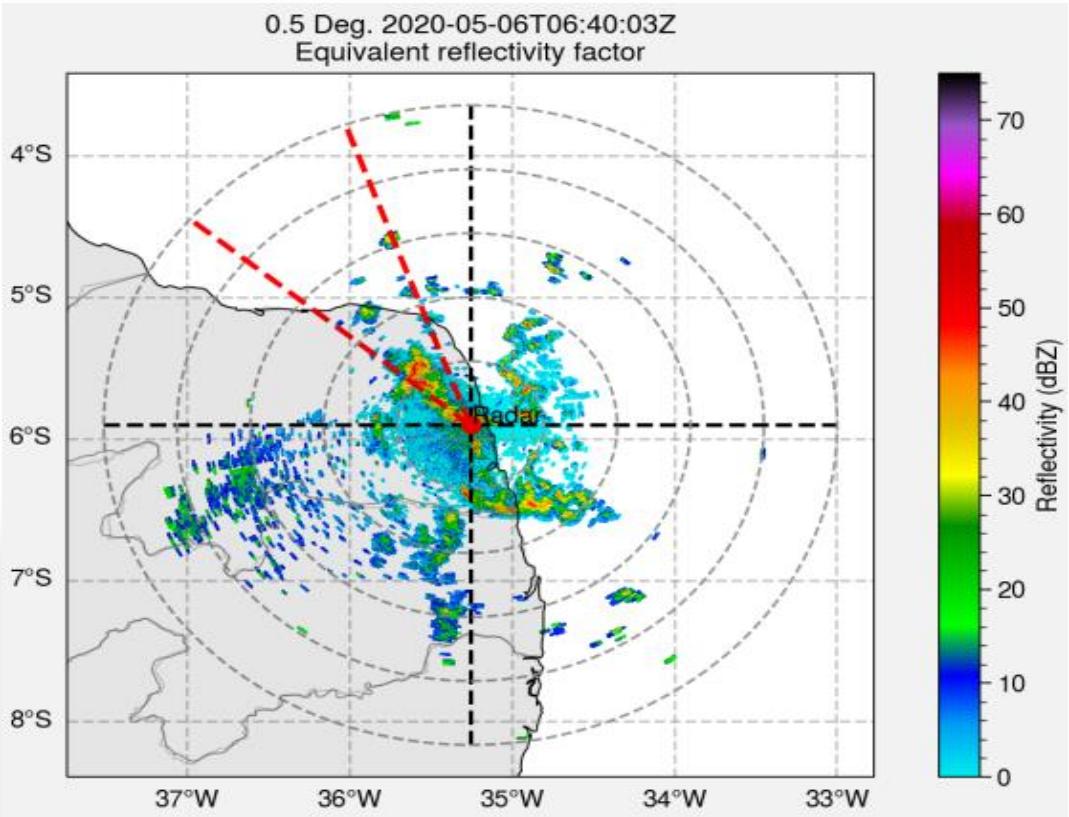
<https://geonetcast.wordpress.com/2022/06/08/vlab-produtos-de-oceanografia-por-satelite-conteudo-completo/>



Application Examples: Weather Radar

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

0.5 Deg. 2020-05-06T06:40:03Z
Equivalent reflectivity factor



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/>

SENTINEL-2 and 3 images processed with the Pytroll / Satpy library

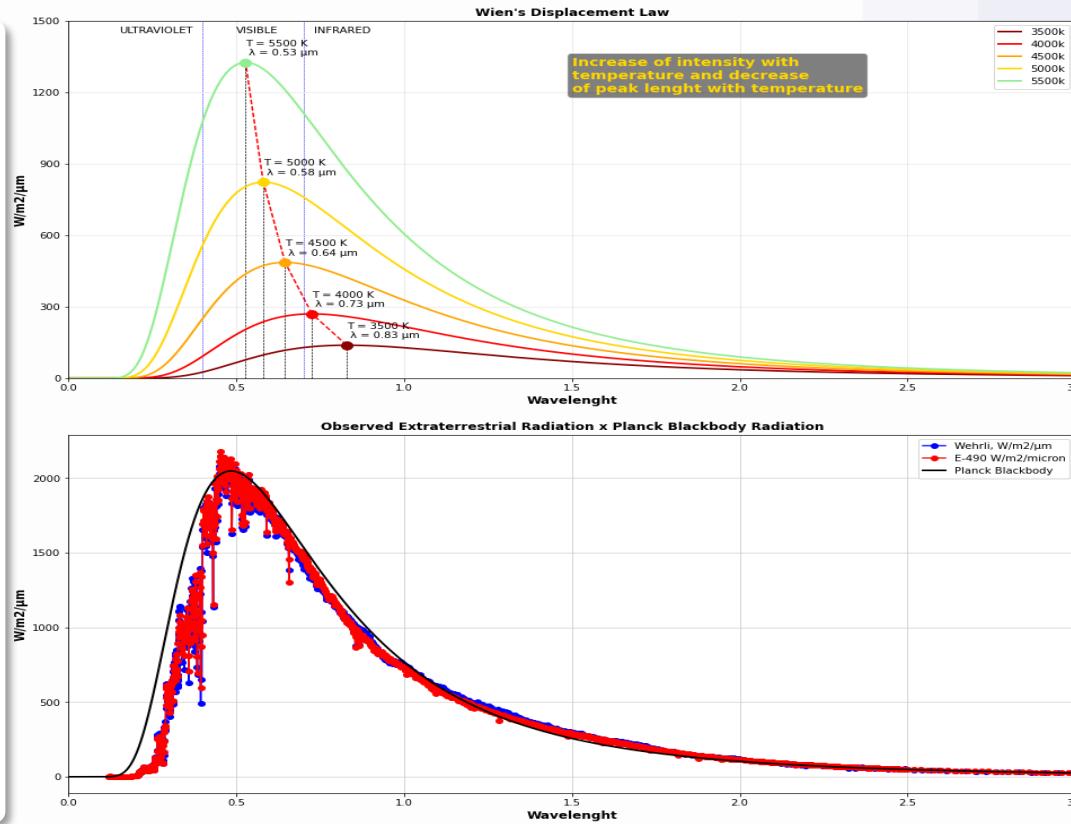
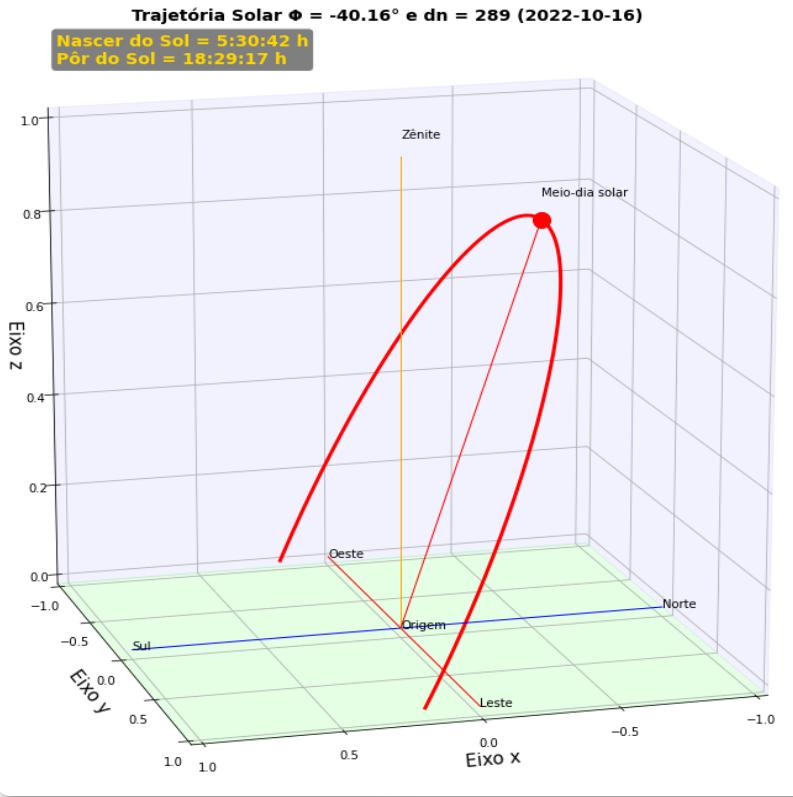


```
1  from satpy.scene import Scene
2  from satpy import find_files_and_readers
3  from datetime import datetime
4
5  files = find_files_and_readers(base_dir="C:\\\\MSI",
6                                reader='safe_msi')
7
8  scn = Scene(filenames=files)
9  scn.load(['true_color'])
10 scn.save_dataset('true_color', 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).

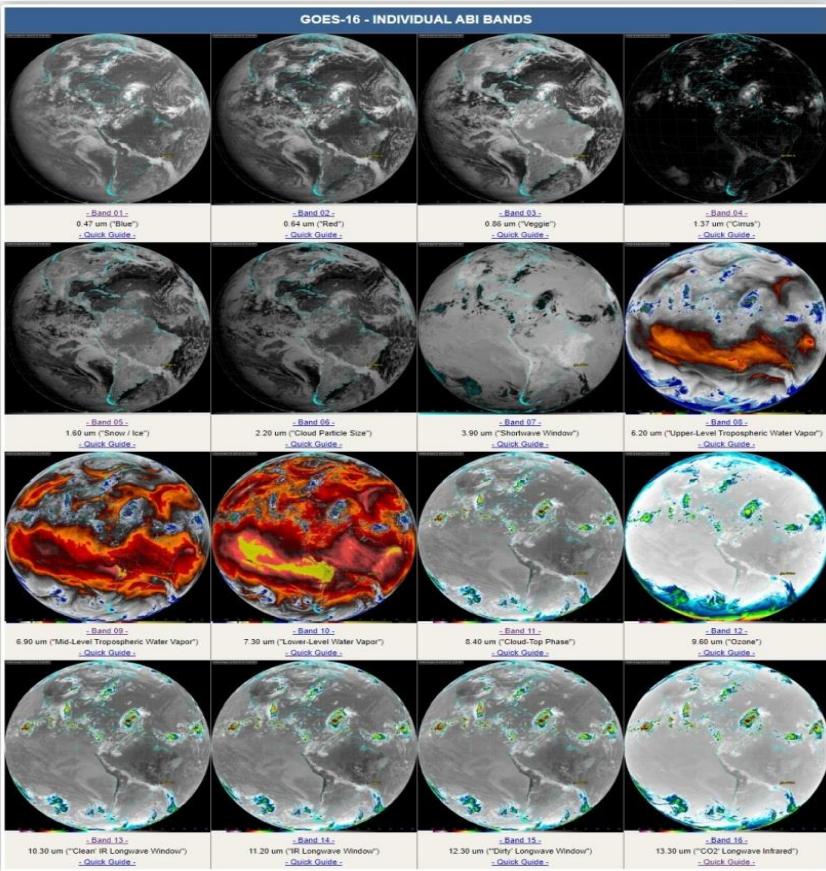
```
from siphon.catalog import TDSCatalog      # Code to support reading and
import urllib.request                      # Defines functions and classes

# 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

```

4   from netCDF4 import Dataset      # Read / Write NetCDF4 files
5   import matplotlib.pyplot as plt  # Plotting library
6   from datetime import datetime    # Basic Dates and time types
7   import cartopy, cartopy.crs as ccrs # Plot maps
8   import os                       # Miscellaneous operating system interfaces
9   import boto3                     # Amazon Web Services (AWS) SDR for Python
10  from botocore import UNSIGNED   # boto3 config
11  from botocore.config import Config # boto3 config
12
13
14  # Input and output directories
15  input = "Samples"; os.makedirs(input, exist_ok=True)
16  output = "Output"; os.makedirs(output, exist_ok=True)
17
18  # AMAZON Repository information
19  # https://noaa-goes16.s3.amazonaws.com/index.html
20  bucket_name = 'noaa-goes16'
21  product_name = 'ABI-L2-CMIPF'
22  year = 2021
23  day_of_year = 37
24  hour = 17
25  min = 00
26  band = 1
27
28  # Initializes the S3 client
29  s3_client = boto3.client('s3', config=Config(signature_version=UNSIGNED))
30
31  # File structure
32  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}'
33
34  # Search for the file on the server
35  s3_result = s3_client.list_objects_v2(Bucket=bucket_name, Prefix=prefix, Delimiter = "/")
36
37  # Check if there are files available
38  if 'Contents' not in s3_result:
39      # There are no files
40      print("No files found for the date: ",year,day_of_year)
41      quit()
42  else:
43      # There are files
44      for obj in s3_result['Contents']:
45          key = obj['Key']
46          # Print the file name
47          print(key)

```



INPUT AND OUTPUT DIRECTORIES

LIBRARIES

DESIRED DATA AND SEARCH IN THE AWS “BUCKET”

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

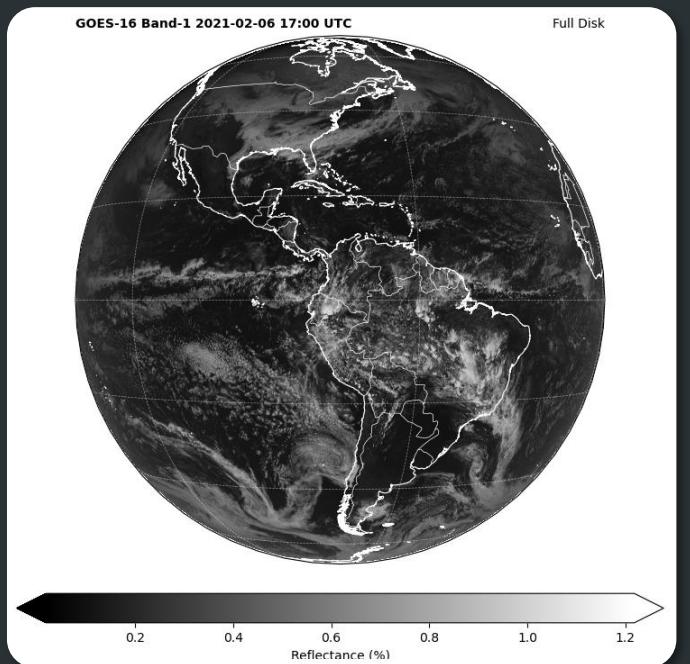
CHECK IF THE DESIRED DATA IS AVAILABLE

It is Also Possible to Access Data

DATA NOT AVAILABLE: END THE SCRIPT

DATA AVAILABLE: PROCESS THE DATA

```
36
37 if 'Contents' not in s3_result:
38     # There are no files
39     print('No files found for the date: ',year,day_of_year)
40     quit()
41 else:
42     # Create a session
43     for obj in s3_result['Contents']:
44         key = obj['Key']
45         # Print the file name
46         print(key)
47         file_name = key.split('/')[-1].split('.')[0]
48
49         # Download the file
50         if not os.path.exists(F'{input}/{file_name}.nc'):
51             s3_client.download_file(bucket_name, key, F'{input}/{file_name}.nc')
52
53         # If the file exists
54         if (os.path.exists(F'{input}/{file_name}.nc')):
55
56             # Open the GOES-R image
57             file = Dataset(F'{input}/{file_name}.nc')
58
59             # Get the pixel values
60             data = file.variables['CM1'][:]
61
62             # Choose the plot size (width x height, in inches)
63             plt.figure(figsize=(10,10))
64
65             # Use the Geostationary projection in cartopy
66             ax = plt.axes(projection=ccrs.Geostationary(central_longitude=-75.0, satellite_height=35786023.0))
67             img_extent = (-5434894.67527,5434894.67527,-5434894.67527,5434894.67527)
68
69             # Add coastlines, borders and gridlines
70             ax.coastlines(resolution='10m', color='white', linewidth=0.8)
71             ax.add_feature(cartopy.feature.BORDERS, edgecolor='white', linewidth=0.5)
72             ax.gridlines(color='white', alpha=0.5, linestyle='--', linewidth=0.5)
73
74             # Define the color scale based on the channel
75             if band < 6:
76                 colormap = "gray" # Black to white for visible channels
77                 prodname = "Reflectance (%)"
78             else:
79                 colormap = "gray_r" # White to black for IR channels
80                 prodname = "Brightness Temperatures (C)"
81
82             # Plot the image
83             img = ax.imshow(data, origin='upper', extent=img_extent, cmap=colormap)
84
85             # Add a colorbar
86             plt.colorbar(img, label=prodname, extend='both', orientation='horizontal', pad=0.05, fraction=0.05)
87
88             # Extract the date
89             date = (datetime.strptime(file.time_coverage_start, '%Y-%m-%dT%H:%M:%S.%fZ'))
90
91             # Add a title
92             plt.title('GOES-16 Band-' + str(band) + ' ' + date.strftime('%Y-%m-%d %H:%M') + ' UTC', fontweight='bold', fontsize=10, loc='left')
93             plt.title('Full Disk', fontsize=10, loc='right')
94
95             # Save the image
96             plt.savefig(F'{output}/{file_name}.png')
97
98             # Show the image
99             plt.show()
```



Everything is Possible

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

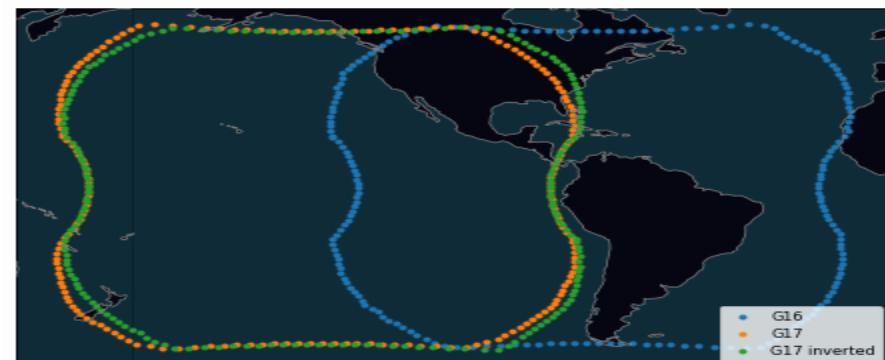
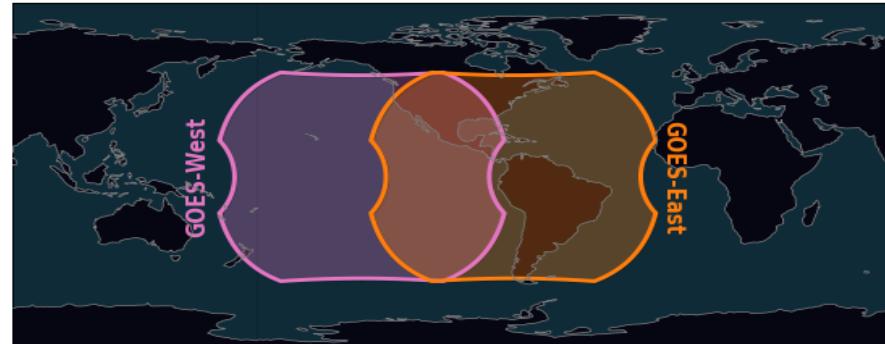
GOES ABI Field of View



GOES-West ABI Field of View and Mesoscale Domains



GOES GLM Field of View



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>

A Practical Introduction to Python Programming

An introductory programming text for students with no prior programming experience.

Tag(s): [Introduction to Computer Programming](#) [Python](#)

Publication date: 01 Dec 2012 **Type:** Book
ISBN-10: n/a **Publisher:** n/a
ISBN-13: n/a **License:** [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported](#)
Paperback: 263 pages **Paperback:** 495 pages
Views: 13,797 **Post time:** 02 Jul 2021 01:00:00

Summary/Excerpts of (and not a substitute for) the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported:

You are free to:

Share — copy and redistribute the material in any medium or format
Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

Click [here](#) to read the full license.

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

Dive Into Python 3

Dive Into Python 3 covers Python 3 and its differences from Python 2. Compared to Dive Into Python, it's about 20% revised and 80% new material.

Tag(s): [Python](#)

Publication date: 31 Dec 2011 **Type:** N/A
ISBN-10: 1430224150 **Publisher:** [APress](#)
ISBN-13: 9781430224150 **License:** [Creative Commons Attribution-ShareAlike 3.0 Unported](#)
Paperback: 495 pages **Post time:** 23 May 2016 12:00:00
Views: 9,804

Summary/Excerpts of (and not a substitute for) the Creative Commons Attribution-ShareAlike 3.0 Unported:

You are free to:

Share — copy and redistribute the material in any medium or format
Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Click [here](#) to read the full license.

<http://getpython3.com/diveintopython3/>

A Whirlwind Tour of Python

This free e-book is a fast-paced intro to Python aimed at researchers and engineers.

Tag(s): [Data Science](#) [Python](#)

Publication date: 01 Aug 2016 **Type:** Book
ISBN-10: n/a **Publisher:** [O'Reilly Media, Inc.](#)
ISBN-13: 9781491964644 **License:** [Creative Commons CC0 "No Rights Reserved"](#)
Paperback: 98 pages **Post time:** 29 Jul 2020 06:00:00
Views: 12,697

Summary/Excerpts of (and not a substitute for) the Creative Commons CC0 "No Rights Reserved":

CC0 enables scientists, educators, artists and other creators and owners of copyright- or database-protected content to waive those interests in their works and thereby place them as completely as possible in the public domain, so that others may freely build upon, enhance and reuse the works for any purposes without restriction under copyright or database law.

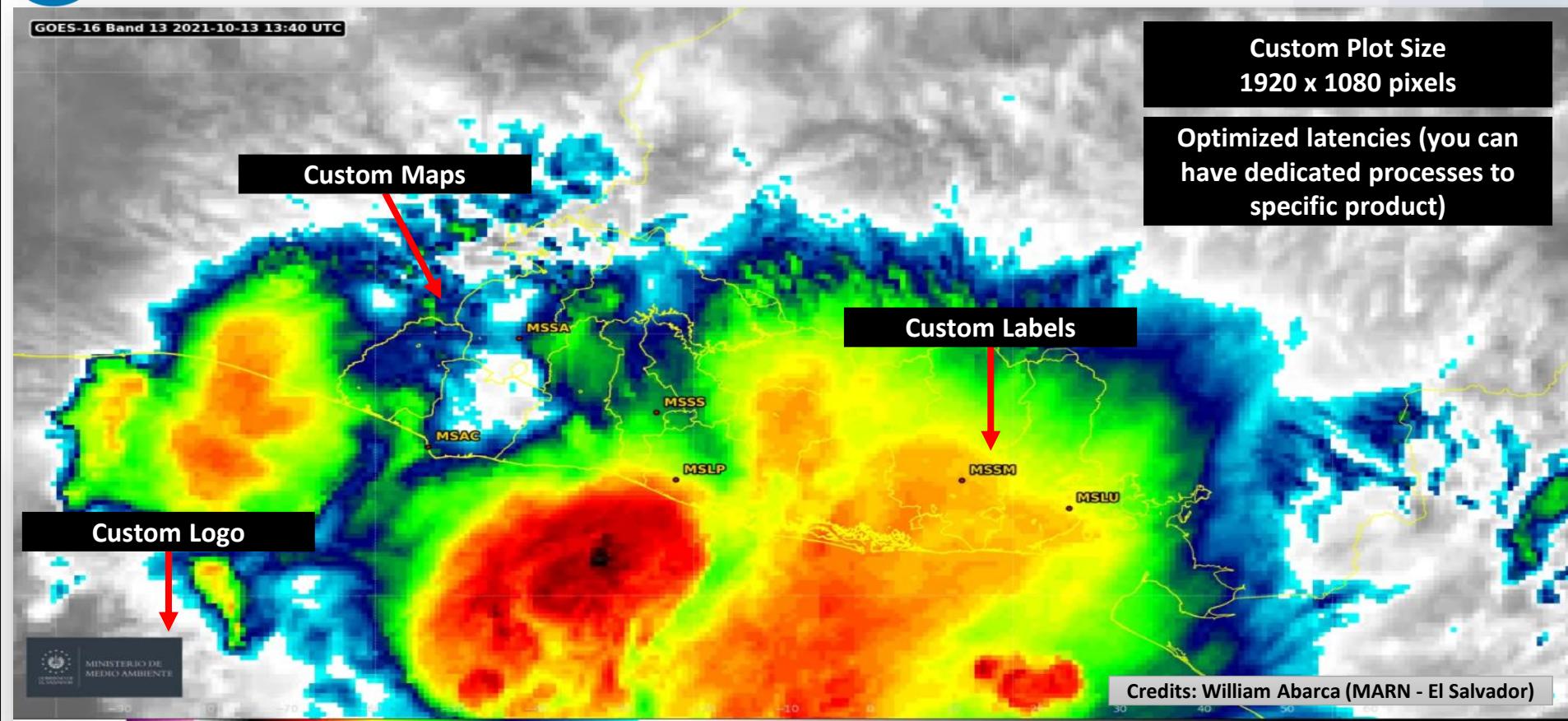
In contrast to CC's licenses that allow copyright holders to choose from a range of permissions while retaining their copyright, CC0 empowers yet another choice altogether - the choice to opt out of copyright and database protection, and the exclusive rights automatically granted to creators - the "no rights reserved" alternative to our licenses.

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

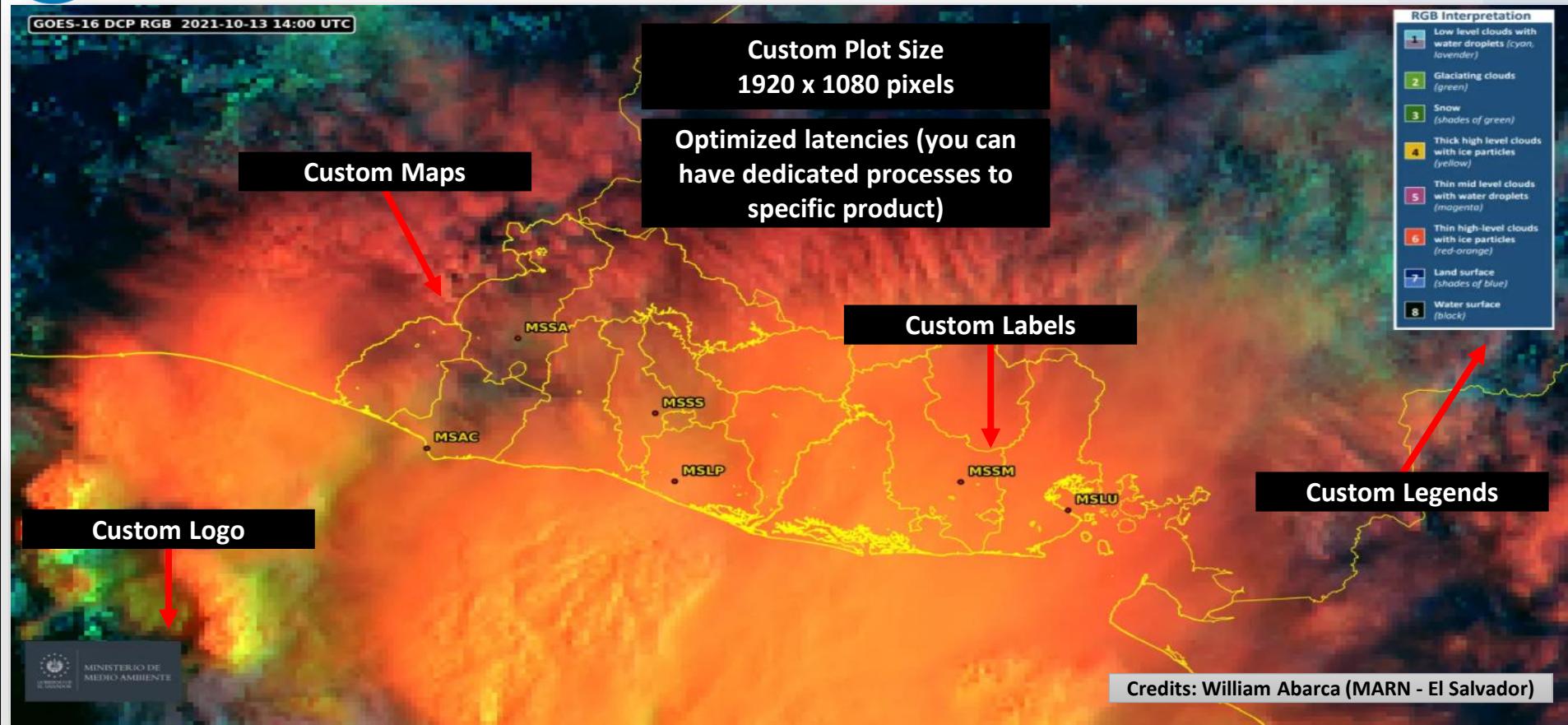
...and much more

BENEFITS OF ACCESSING DIGITAL FILES?

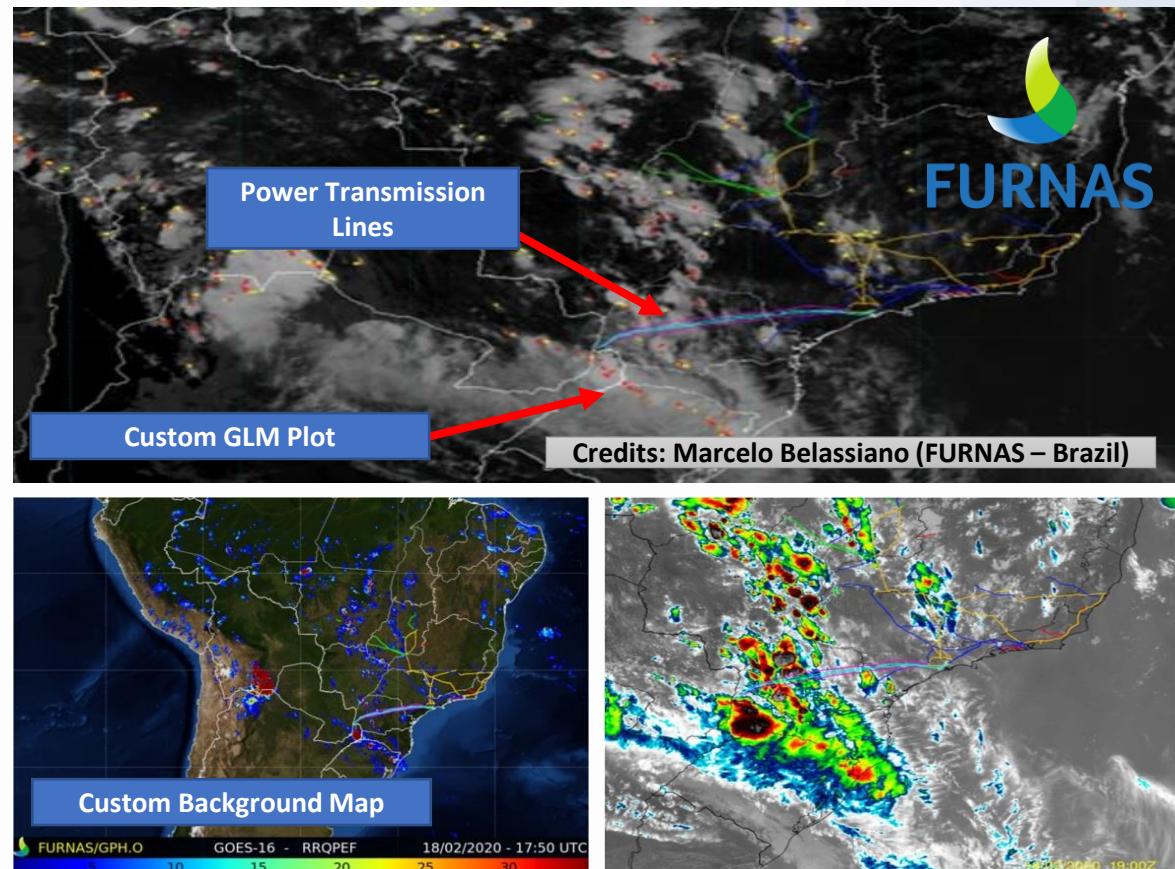
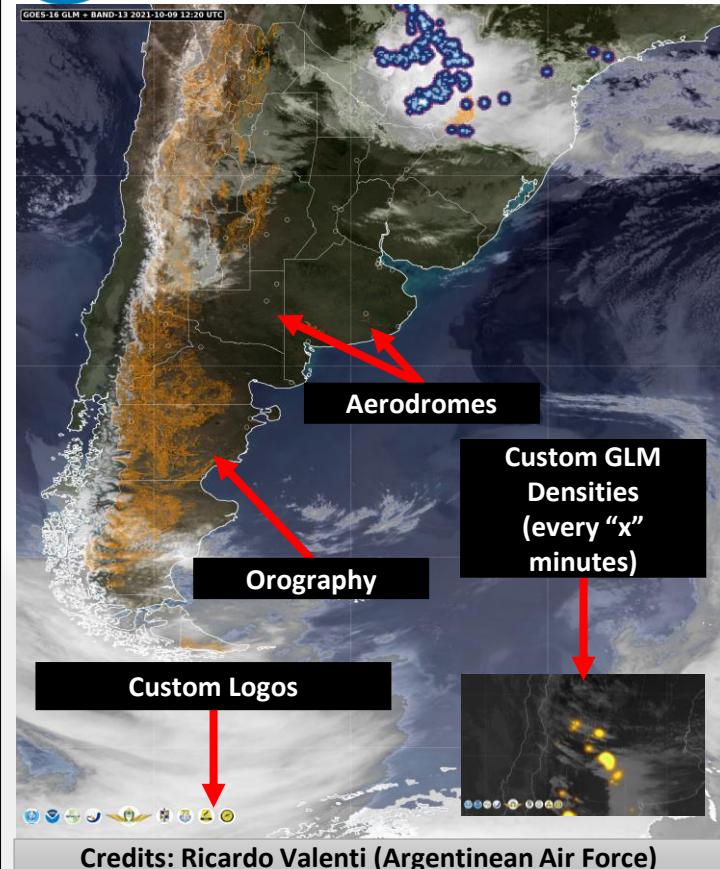
Advantages of Accessing Digital Data



Advantages of Accessing Digital Data



Advantages of Accessing Digital Data



Advantages of Accessing Digital Data

- NowCasting.com.br -

aceiô

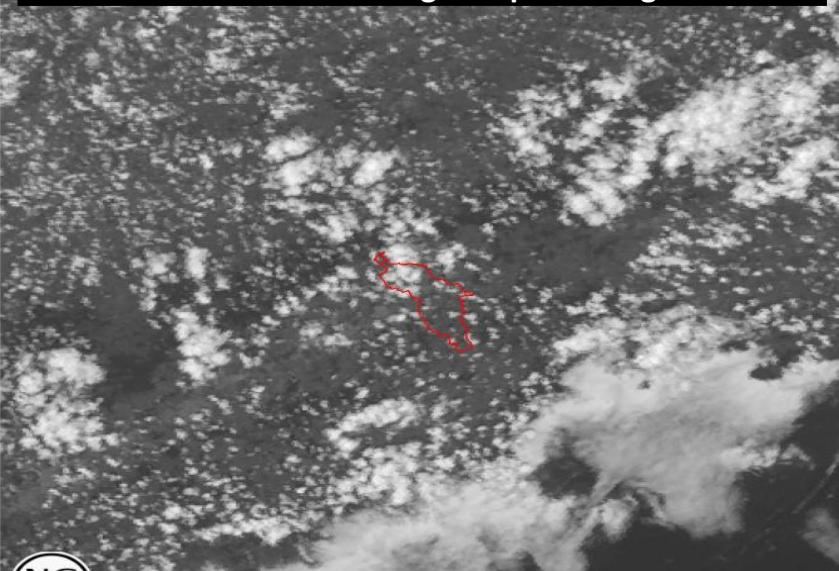
HC Distrito Federal: [Distrito Federal](#)

Cidades de SP: [Bauru](#) [Bebedouro](#) [Borá](#) [Botucatu](#) Ca

Status: sistema normal ;)

GOES-16 às 13:46 BR (±30s)

Real Time monitoring for specific regions



Credits: Demilson Quintão (Nowcasting - Brazil)



Fonte dos dados: GNC-A / NOAA / GOES-16 / ABI / GLM

["Visível"](#) ["Queimadas"](#) ["Temperatura"](#) ["Chuva Estimada"](#)

- NowCasting.com.br -

HC imagens ao redor das cidades, Verifique!

HC Em AL: [Maceió](#)

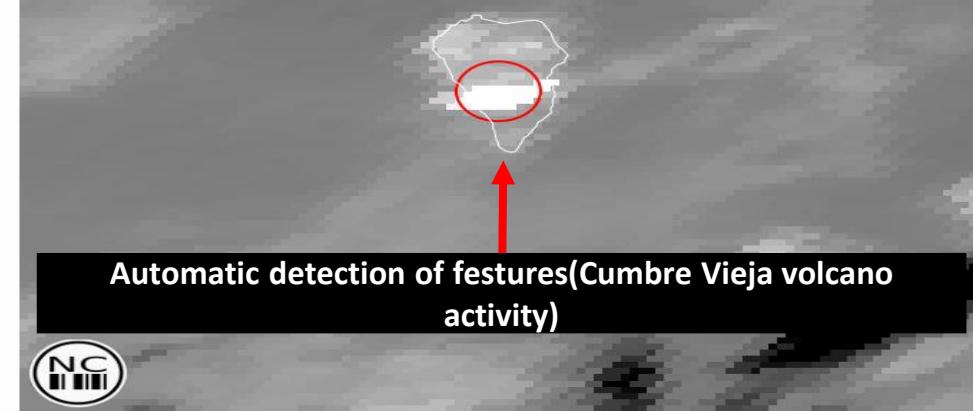
HC Distrito Federal: [Distrito](#)

Status: sistema normal ;)

GOES-16 às 14:23 BR (±30s)

Precise scan time calculation for a specific region

<https://youtu.be/dc1drBHdUaY>



Automatic detection of festures(Cumbre Vieja volcano activity)



Fonte dos dados: GNC-A / NOAA / GOES-16 / ABI / GLM

["Visível"](#) ["Queimadas"](#) ["Temperatura"](#) ["Chuva Estimada"](#)

Advantages of Accessing Digital Data

Vías de comunicación de avisos de incendios

InUMet - GOES-16 Fire/Hot Spot Characterization
12 suscriptores
7 de noviembre
InUMet - GOES-16 Fire/Hot Spot Characterization
Instante: 07/11/2020 - 16:10
Ver en Grafana-InUMet
Alerta: 10 - Procesado
Área: 0.4 ha
Departamento: Colonia
Coordenadas: -34.08,-57.59
Ver en OpenStreetMap
10 16:24

InUMet - GOES-16 Fire/Hot Spot Characterization
Instante: 07/11/2020 - 16:20
Ver en Grafana-InUMet
Alerta: 30 - Procesado
Área: 0.8 ha
Departamento: Colonia
Coordenadas: -34.08,-57.59
Ver en OpenStreetMap
10 16:34

InUMet - GOES-16 Fire/Hot Spot Characterization
Instante: 07/11/2020 - 16:30
Ver en Grafana-InUMet
Alerta: 30 - Procesado

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.

Instante	Coordenadas	Dept.	Alerta	Cód. de Alerta	Área	FRP	Temp. de Brillo
22/11/2020 - 16:40	-33.2,-58.08	Río Negro	Contaminado por nubes	32	-	-	56.0 °C
22/11/2020 - 16:40	-33.2,-58.08	Río Negro	Procesado	30	5646.0 m²	228.0 MW	63.0 °C
22/11/2020 - 16:40	-33.17,-58.07	Río Negro	Probabilidad Alta	33	-	-	63.0 °C
22/11/2020 - 16:40	-33.17,-58.09	Río Negro	Procesado	30	4488.0 m²	168.0 MW	58.0 °C
22/11/2020 - 16:20	-33.17,-58.07	Río Negro	Probabilidad Alta	53	-	72.0 MW	47.0 °C
22/11/2020 - 16:20	-33.2,-58.06	Río Negro	Probabilidad Alta	33	-	69.0 MW	47.0 °C
22/11/2020 - 16:10	-33.2,-58.06	Río Negro	Probabilidad Alta	13	-	76.0 MW	48.0 °C
22/11/2020 - 13:40	-33.87,-56.53	San José	Procesado	30	85408.0 m²	37.0 MW	48.0 °C
22/11/2020 - 13:00	-33.69,-56.12	Florida	Probabilidad Baja	15	-	-	48.0 °C
22/11/2020 - 12:00	-33.89,-56.52	San José	Probabilidad Baja	35	-	-	45.0 °C
22/11/2020 - 11:50	-34.21,-57.24	Colonia	Procesado	10	28819.0 m²	27.0 MW	44.0 °C
22/11/2020 - 11:40	-33.89,-56.52	San José	Probabilidad Alta	13	-	71.0 MW	50.0 °C
22/11/2020 - 11:40	-33.87,-56.53	San José	Procesado	10	120106.0 m²	60.0 MW	48.0 °C
22/11/2020 - 11:30	-31.43,-57.92	Salto	Contaminado por nubes	32	-	-	51.0 °C
22/11/2020 - 11:30	-33.39,-55.66	Florida	Probabilidad Baja	15	-	-	44.0 °C

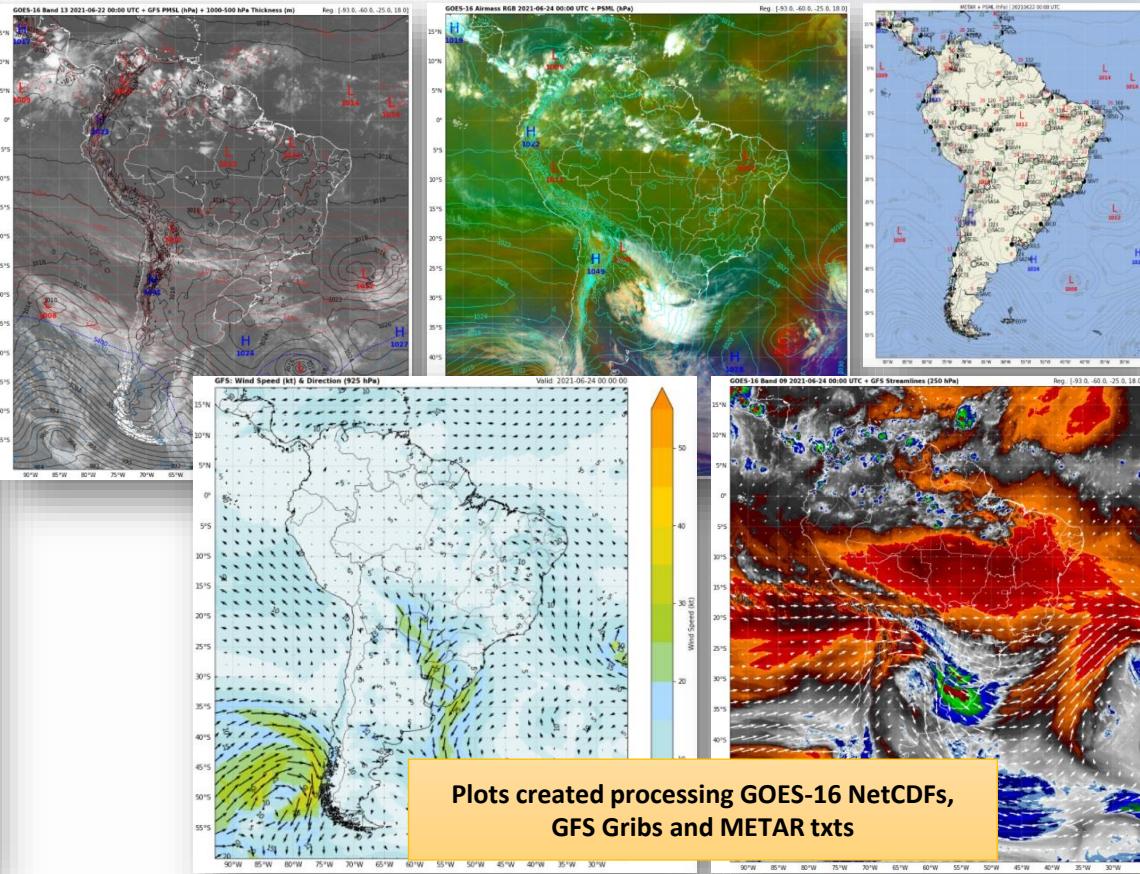
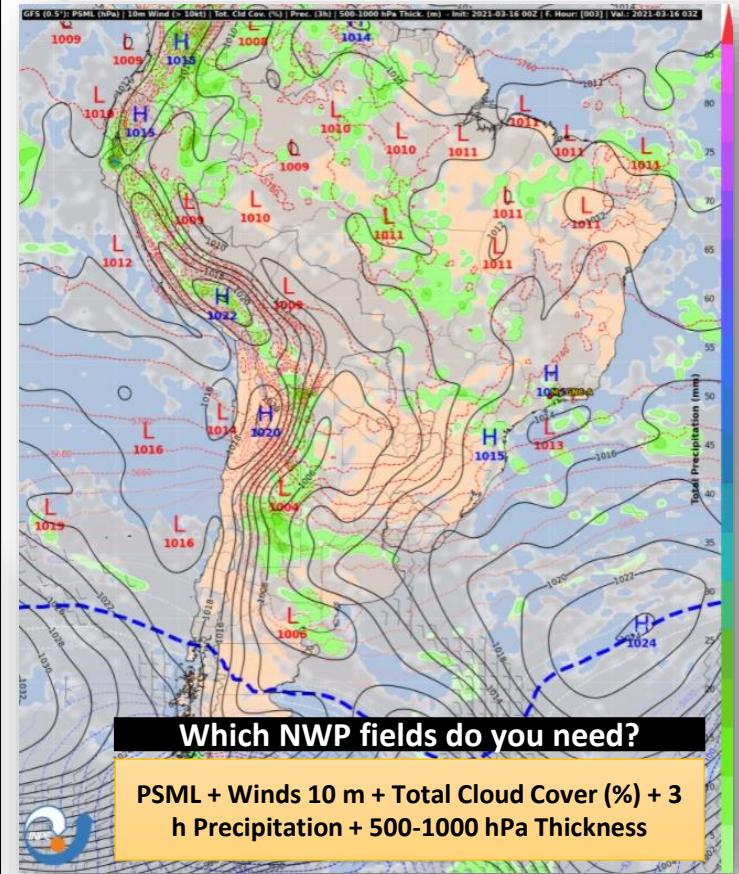
GEONETCast - Uruguay

Credits: Roxana Sagarra (INUMET - Uruguay)

GEONETCast inumet

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

Advantages of Accessing Digital Data



Advantages of Accessing Digital Data

Optimized Latencies

The screenshot displays the GEONETCAST AMÉRICAS software interface. On the left, a satellite map of South America shows cloud cover and precipitation patterns. A red box highlights the timestamp "15:10 UTC" at the top of the map. To the right, a command-line window titled "Calling Monitor Script" shows the process of receiving and processing a GOES-R scan. The log indicates the start of the monitor script at 12:28:46, the start of the scan at 12:28:46.902826, and the end of the monitor script at 12:29:52.298125. It also notes a total processing time of 5.4 seconds, with 4.43 seconds specifically mentioned for file processing. A red arrow points from the "15:10 UTC" box to the "15:10 UTC" in the log. Another red arrow points from the "4.43 seconds" to the "4.43 seconds" in the log. At the bottom of the software interface, a message credits Ricardo Valenti (Argentinean Air Force). A blue box on the right states: "GOES-R scan from 15:10 to 15:20 UTC", "Band 13 reception via GNC-A (between 39 and 47 seconds after scan)", and "4.43 seconds to process the file". A final blue box at the bottom right states: "Final plot available at the same minute of the end of the GOES-R scan (15:20 UTC or 12:20 local time)". The Windows taskbar at the bottom right shows the date and time as "12:20 p.m. 01/11/2021".

GEONETCAST AMÉRICAS - FUERZA AÉREA ARGENTINA - GOES-16 Band 13 2021-11-01 15:10 UTC

GOES-16 Band 13 2021-11-01 15:10 UTC

GOES-R scan from 15:10 to 15:20 UTC

Band 13 reception via GNC-A (between 39 and 47 seconds after scan)

4.43 seconds to process the file

Final plot available at the same minute of the end of the GOES-R scan (15:20 UTC or 12:20 local time)

Credits: Ricardo Valenti (Argentinean Air Force)

Computer Specifications:
Intel i5 processor
16GB DDR4 RAM
240GB M2 SSD for OS

12:20 p.m.
01/11/2021

Advantages of Accessing Digital Data

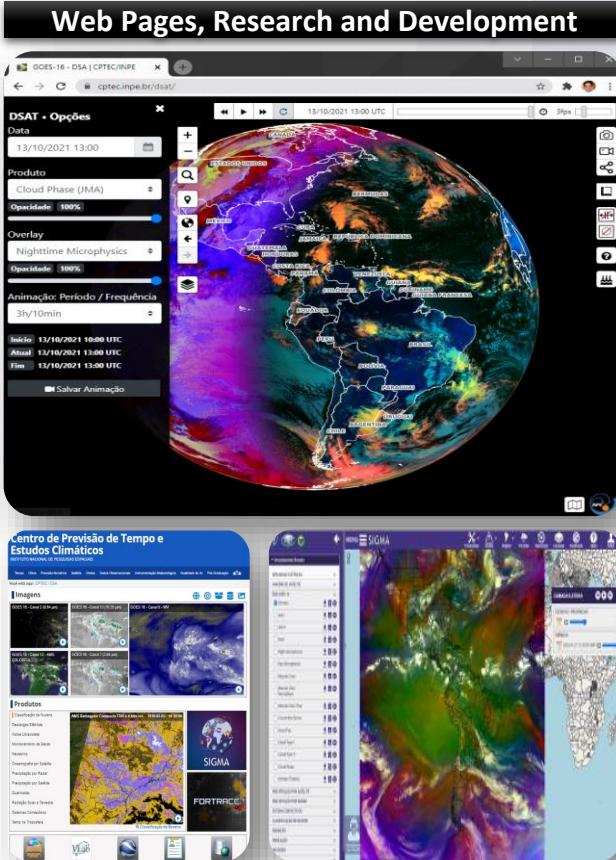
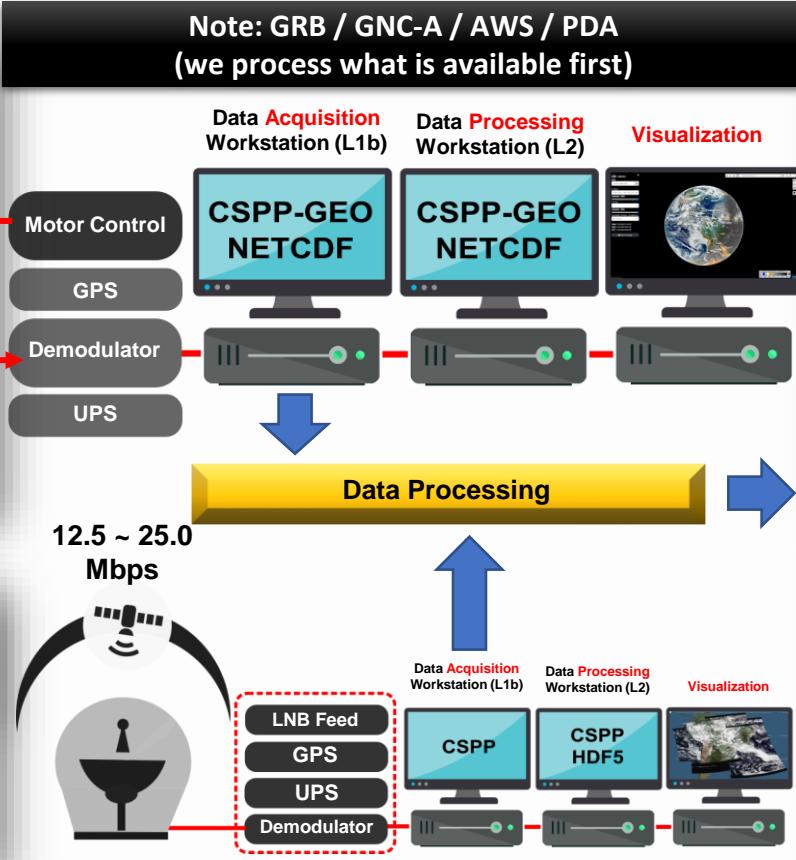
Optimized Latencies

The screenshot displays a dual-monitor setup illustrating the rapid processing of GOES-R satellite imagery.

Left Monitor (Mozilla Firefox): Shows a "SHOWCast: GNC-A Visualizer" interface. The URL is 192.168.4.241/HTML. A red box highlights the timestamp "21:00 UTC". The main window displays a "GOES-16 Band 13 2021-11-0" image. A blue box contains the text: "GOES-R scan from 21:00 to 21:10 UTC". Below the image, a yellow box contains: "Band 13 reception via GNC-A (between 39 and 47 seconds after scan)". Another yellow box contains: "7.52 seconds to process the file". The image shows a map of Central America with several weather stations labeled: MSSA, MSSS, MSLP, MSSM, and MSU. A legend indicates "Computer Specifications: Intel i9 10850K 10/20 c/t @ 3.6-4.8 GHz Processor, 32 GB RAM DDR3, 500GB M2 SSD for OS, 10TB HDD for data, GPU Nvidia 2060 - 6 GB RAM". Credits at the bottom read: "Credits: William Abarca (MARN El Salvador)".

Right Monitor (Terminal): Shows a terminal window titled "Terminal". It displays log output for a "SHOWCAST MONITOR STARTED" at 2021-11-01 15:10:35.741666. The log includes details about the processing product, script used, and total processing time: "Total processing time: 7.52 seconds". A red box highlights the "7.52 seconds" value. The terminal also shows a "SHOWCAST MONITOR ENDED" message at 2021-11-01 15:10:44.148908, indicating one product was processed. A "Memory" usage chart is shown below the logs, and a red box highlights the "7.52 seconds" value again. The system tray at the bottom right shows the date and time as 3:10 PM 11/1/21.

Example: INPE's Meteorological Satellites Division



NECESSARY TOOLS



Running Scripts Locally or in the Cloud

D:\VLAB\Python\script_07.py - N

Arquivo Editar Localizar Visualizar

script_06.py script_07.py

```
(base) D:\Users\dsouza>conda activate workshop
(base) D:\Users\dsouza>cd D:\VLAB\Python
(base) D:\VLAB\Python>python script_07.py
```

```

1 # INPE / CPTC Training
2 # Author: Diego Souza
3
4 import pygrib
5 import matplotlib.pyplot as plt
6 import cartopy, cartopy.crs
7 import cartopy.io.shapereader
8 import numpy as np
9 import matplotlib
10
11
12 # Open the GRIB File
13 grib = pygrib.open("D:/VLAB/GRIB/GRIBFull_0p50.r000")
14
15 # Select the GRIB
16 grb = grib.select(name="2 metre temperature")[0]
17
18 # Get information from the file
19 init = str(grb.analDate) # Init date / time
20 run = str(grb.hour).rjust(2) # Run
21 ftime = str(grb.forecastTime) # Forecast hour
22 valid = str(grb.validDate) # Valid date / time
23 print("Init: " + init + " UTC")
24 print("Run: " + run + " UTC")
25 print("Forecast: " + ftime)
26 print("Valid: " + valid + " UTC")
27
28 # Select the extent (min, lon, min, lat, max, lon, max)
29 extent = [-93.0, -60.0, -25.0, 18.0]
30
31 # Read the data for a specific region
32 tmap, lats, lons = grb.data(latiExtent[1],lat2Extent[1])
33
34
35 # Convert from K to °C
36 tmap = tmap - 273.15
37
38 # Set the resolution
39 # Choose the plot size (width x height, in inches)
40 plt.figure(figsize=(8,8))
41
42 # Use the Cylindrical Equidistant projection in cartopy
43 ax = plt.axes(projection=ccrs.PlateCarree())
44
45 # Define the image extent
46 img_extent = [extent[0], extent[2], extent[1], extent[3]]
47
48 # Add a shapefile
49 shapereader = 'http://geonftp.ibge.gov.br/organizacao do territorio/municipios/municipios.shp'
50 shapereader = list(shapereader.Reader('RR_IB_2019.hpp')).geometries()
51 ax.add_geometries(shapereader, ccrs.PlateCarree(), edgecolor='black')
52
53 # Add coastlines, borders and gridlines
54 ax.coastlines(resolution='10m', color='black', linewidth=1)
55 ax.add_feature(cartopy.feature.BORDERS, edgecolor='black')
56 ax.add_feature(cartopy.feature.LAKES, alpha=0.5)
57 ax.add_feature(cartopy.feature.RIVERS)
58
59 # Define contour lines and gridlines
60 ax.contours(lines, colors='black', linewidth=1)
61
62 # Data min, max and interval
63 data_min = -10
64 data_max = 28
65 interval = 2
66 levels = np.arange(data_min,data_max,interval)
67
68 # Define the contour lines
69 data_min = -10
70 data_max = 28
71 interval = 2
72
73 # Define the contour lines
74 data_min = -10
75 data_max = 28
76 interval = 2
77
78 # Plot the map
79 plt.pcolormesh(lons, lats, tmap, transform=ccrs.PlateCarree(), shading='gouraud', cmap='Spectral')
80
81 # Add a colorbar
82 plt.colorbar(img3.lines, label='Wind Speed (kt)', extend='both', orientation='vertical', pad=0.03, fraction=0.05)
83 plt.colorbar(img1, label='Brightness Temperatures (°C)', extend='both', orientation='vertical', pad=0.03, fraction=0.05)
84
85 # Extract date
86 date = (datetime.strptime(datetime, '%Y-%m-%dT%H:%M:%S.%f'))
87
88 # Add a title
89 plt.title('GOES-16 Band 09 ' + date.strftime('%Y-%m-%d %H:%M') + ' UTC' + ' + GFS Streamlines (250 hPa)', fontweight='bold')
90
91 # Save the image
92 plt.savefig(f'{output}/image_23.png', bbox_inches='tight', pad_inches=0, dpi=300)
93
94 # Show the image
95 plt.show()

```

CONDA

Plugins Janela

2.py script_13.py

curso_NWP_202107.ipynb - Colab

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas

índice

- Script 4: Adicionando Mapas com a Cartopy
- Script 5: Adicionando Shapefiles
- Script 6: Plotando Contornos e "Labels"
- Script 7: Paletas de Cores Personalizadas
- Script 8: Suavizando os Contornos
- Script 9: Trabalhando com Diversos Arquivos
- Script 10: Criando uma Animação
- Script 11: Médias, Máximos e Mínimos - Múltiplos Plots Simultâneos
- Script 12: Precipitação Instantânea e Precipitação Acumulada
- Script 13: Lendo Campos Especificando o Nível - Linhas de Corrente
- Script 14: Animando as Linhas de Corrente
- Script 15: Vetores de Vento
- Script 16: Barbelas
- Script 17: Plot 2 x 2 - Linhas de Corrente em 250, 500, 700 e 850 hPa
- Script 18: Lendo Diversos Campos em Diversos Níveis - Índice Galvez Davison (GDI)
- Script 19: Plot de Satélite
- Script 20: Plot de Modelo Numérico + Satélite (Exemplo 1)
- Script 21: Plot de Modelo Numérico + Satélite (Exemplo 2)
- Script 22: Plot de Modelo Numérico + Satélite (Exemplo 3)
- Script 23: Plot de Modelo Numérico + Satélite (Exemplo 4)
- Script 24: Plot de Modelo Numérico + Satélite (Exemplo 5)
- Script 25: Plot MEI/AR
- Script 26: METAR + Modelo

Extract date
date = (datetime.strptime(datetime, '%Y-%m-%dT%H:%M:%S.%f'))

Add a title
plt.title('GOES-16 Band 09 ' + date.strftime('%Y-%m-%d %H:%M') + ' UTC' + ' + GFS Streamlines (250 hPa)', fontweight='bold')

Save the image
plt.savefig(f'{output}/image_23.png', bbox_inches='tight', pad_inches=0, dpi=300)

Show the image
plt.show()

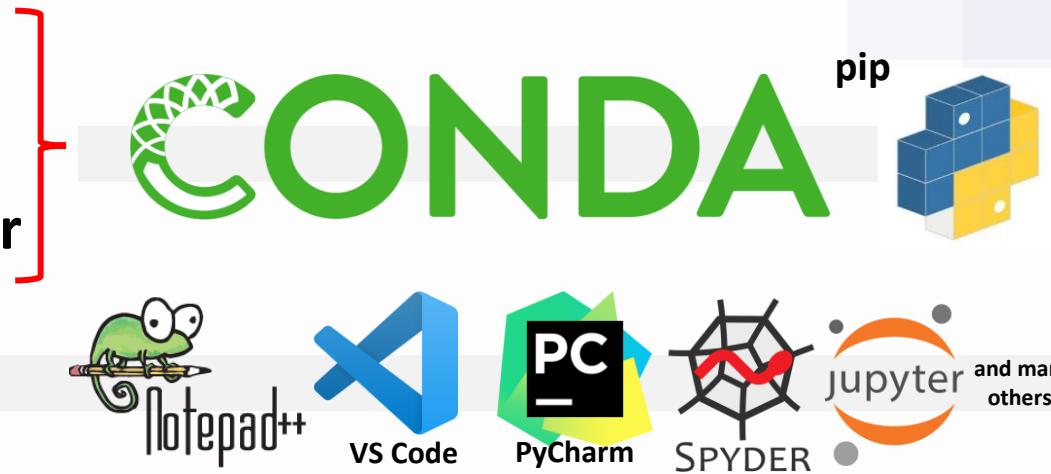
Downloading file /content/Samples/OR_ABI-L2-CMIPF-M6C09_G16_s20211870000204_e20211870009517_c20211870010014.nc

Init: 2021-07-06 00:00:00 UTC
Run: 08Z
Forecast: +0
Valid: 2021-07-06 00:00:00 UTC

Reg: [-93.0, -60.0, -25.0, 18.0]

Running Scripts Locally

- Python
- Package Manager
- Virtual Environment Manager



- Text Editor / IDE

(Integrated Development Environment)

- 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

[Home](#) [Blog Content](#) [The GEONETCast System](#) [GEONETCast-Americas](#) [Documentation](#) [Search](#)

[Data Processing](#) [User Interaction](#) [GNC-A Forum](#) [Contact](#) [Author](#)

Printed on 2023-03-19 by Diego Souza

-- Previous | Next --

(84) Getting Started With Python and Satellite Imagery



Contact: If you have any questions, please contact:

• E-mail: diego.souza@nasa.gov

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 visualise pixel values (brightness temperatures / reflectance)
 - Change color scales, add a title and a colour 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 "Packer Manager" (to install libraries)
- A "Environment Manager" (to separate our projects)
- A code editor (we recommend VS Code)
- GOES-R imagery samples (the data will be manipulated)

For the first three items ("Python 3.10", "Package Manager" and "Environment Manager"), the "Miniconda" tool will be sufficient. As for the next items there are many options available (Visual Studio Code, Spyder, PyCharm, Atom, Jupyter, etc); but for simplicity (and avoiding problems with platform 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 receiving 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 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>



— Downloading Miniconda

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

```
|:| /usr/local/conda/pkgs/23_3.10-1-linux-x86_64.sh -p /my_directory/
```

Notes – Windows and Linux installation:

- The installation will take approximately 5 minutes

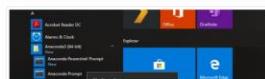
CREATING A PYTHON VIRTUAL ENVIRONMENT AND INSTALLING LIBRARIES

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 will need the following libraries and their dependencies in this environment:

- matplotlib: library for creating plots
- netcdf4: read and write NetCDF files
- cantools: produce maps and other analysis of geospatial data
- geopandas: geospatial data manipulation and analysis
- gdal: representation/manipulation of geospatial data
- scipy: array manipulation
- pandas: data manipulation and analysis
- ipython: python shell
- matplotlib: 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 input the following command:

```
|:| conda create --name workshop <-- conda-forge matplotlib netcdf4 cart
```



Access to Samples (GNC or Cloud)

Local Data Processing

The screenshot shows a software interface titled "Local Data Processing". The main window displays a circular Earth image with a coordinate grid. The vertical axis ranges from 0 to 9000, and the horizontal axis ranges from 0 to 5000. Below the plot, a status bar shows coordinates: x=2956.32, y=4805.67, and a magnifying glass icon. A legend at the bottom left indicates "One foot plot".

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.

A second window titled "Figure 1" is shown, which is a zoomed-in view of the Earth's surface. It focuses on a small rectangular region in the lower right quadrant of the original plot. The zoomed-in plot has axes ranging from 0 to 3000 horizontally and 4000 to 5000 vertically. The same status bar and legend are present. The zoomed-in region is highlighted with a red box.

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.

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

The screenshot shows the official Miniconda website. It features a sidebar with links for Conda, Conda-build, Miniconda, Windows installers, Mac OS X installers, Linux installers, and other resources. The main content area has sections for "Miniconda" and "Windows installers". The "Windows" section lists Python versions 3.8 and 2.7 with their respective file names, sizes, and SHA256 hashes. The "Mac OS X" section also lists Python versions 3.8 and 2.7 with their file names, sizes, and SHA256 hashes.

Creating a Virtual Environment called “Workshop”

```
(base) D:\Users\dsouza conda create --name workshop -c conda-forge matplotlib netcdf4 cartopy boto3 gdal scipy pandas
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: D:\Users\dsouza\miniconda3\envs\workshop

added / updated specs:
- boto3
- cartopy
- gdal
- matplotlib
- netcdf4
- pandas
- scipy

The following packages will be downloaded:

```

package	build	size	source
boost-cpp-1.74.0	h54f0996_2	16.1 MB	conda-forge
boto3-1.17.16	pyhd8ed1ab_0	70 KB	conda-forge
botocore-1.20.16	pyhd8ed1ab_0	4.5 MB	conda-forge
cartopy-0.7.0	py39hb7eefc_1001	369 KB	conda-forge
bz2file-1.0.8	hf7f7e30_4	149 KB	conda-forge
certifi-certificates-2020.12.5	hs545a59_0	1.75 KB	conda-forge
cairo-1.16.0	hba1bbd2_1007	2.2 MB	conda-forge

Installing a Text Editor

The screenshot shows the Notepad++ website. It features a sidebar with links for Home, Download, News, Online Help, RSS, Donate, and Author. The main content area has a "Downloads" section with links for "Notepad++ 7.9: Stand with Hong Kong", "Notepad++ 7.8.9: Stand with Hong Kong", "Notepad++ 7.8.8 release", "Notepad++ 7.8.7 release", and "Notepad++ 7.8.6 release".

Accessing Sample Imagery

The screenshot shows the GOES-16/17 Amazon Download Page. It features a search interface with fields for Source (AWS, OCC), Satellite (GOES-16/East, GOES-17/West), Domain (Full Disk), Product (ABI L2 Cloud and Moisture Imagery), Date (17/07/2019), and Hour (UTC) (12). Below the search form is a table of search results:

Hour (UTC)	Product	Date	Source	Satellite	Domain
12	ABI L2 Cloud and Moisture Imagery	17/07/2019	AWS	GOES-16/East	Full Disk
12	ABI L2 Cloud and Moisture Imagery	17/07/2019	OCC	GOES-16/East	Full Disk

Running Scripts

```
Training: Python and GOES-R Imagery; Script 1 - http://geonetcast.wordpress.com/2023/03/19/getting-started-with-python-and-satellite-imagery/

Required modules
from netCDF4 import Dataset # Read / Write NetCDF files
import matplotlib.pyplot as plt # Plotting library

# Open the GOES-R image
# Download files at this link: http://home.chpc.utah.edu/~abirch/ABI-L2-CMIPF-M6C13\_G16\_s201919812

# Get the pixel values
data = file.variables['CMF'][:]

# Choose the plot size (width x height, in inches)
plt.figure(figsize=(7,7))

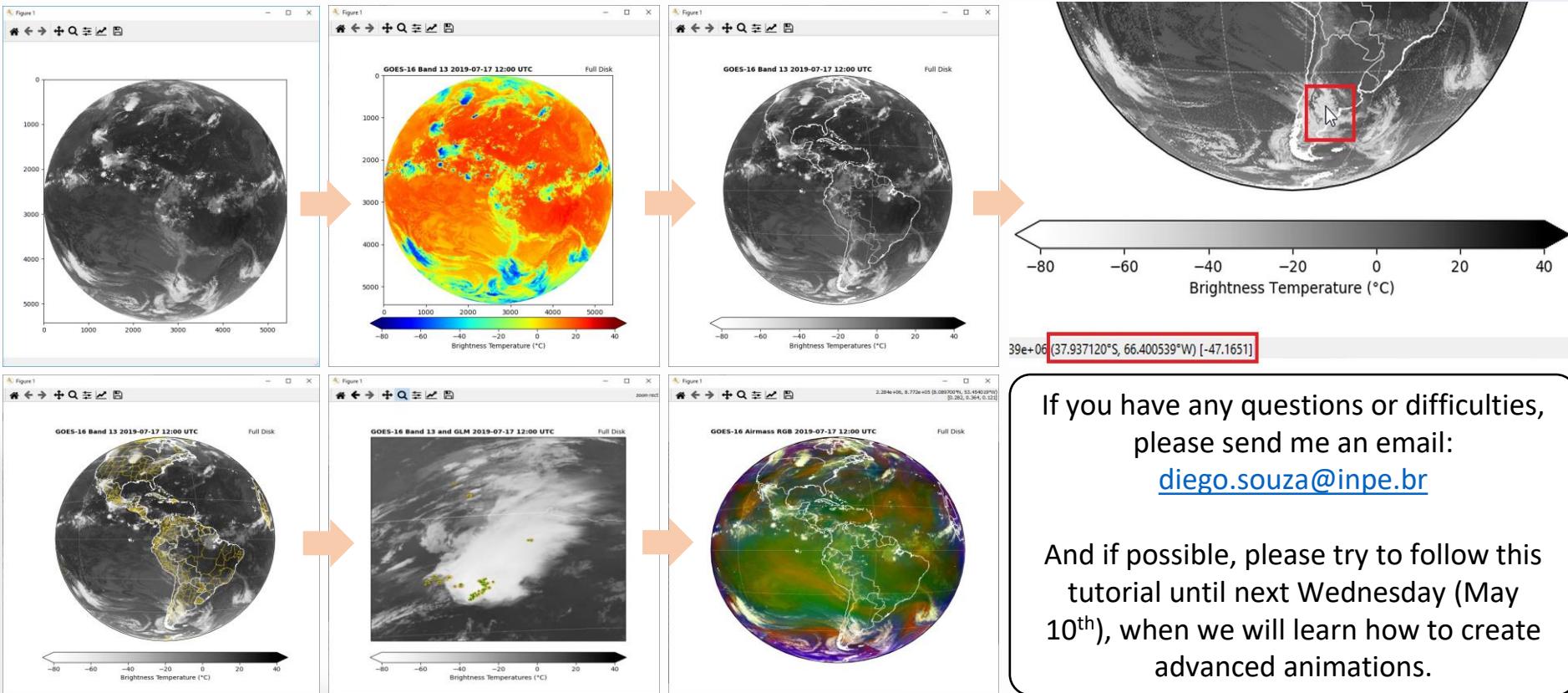
# Plot the image
plt.imshow(data, vmin=193, vmax=313, cmap='Greys')

# Save the image
plt.savefig('Image_01.png')

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

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)

(C:) > VLAB > Python > GOES >

Output
Samples
Image_01.png
Image_02.png
Image_03.png
Image_04.png
Image_05.png
Image_06.png
Image_07.png
Image_08.png
IR4AVHRR6.cpt

ne_10m_admin_1_states_provinces.dbf
ne_10m_admin_1_states_provinces.shp
ne_10m_admin_1_states_provinces.shx

OR_ABI-L2-CMIPF-M6C08_G16_s20191981200396_e20191981210104_c20191981210182.nc
OR_ABI-L2-CMIPF-M6C10_G16_s20191981200396_e20191981210116_c20191981210188.nc
OR_ABI-L2-CMIPF-M6C12_G16_s20191981200396_e20191981210111_c20191981210185.nc
OR_ABI-L2-CMIPF-M6C13_G16_s20191981200396_e20191981210116_c20191981210189.nc
OR_GLM-L2-LCFA_G16_s20191981200000_e20191981200200_c20191981200224.nc

script_01.py
script_02.py
script_03.py
script_04.py
script_05.py
script_06.py
script_07.py
script_08.py
script_09.py
script_10.py
script_11.py
script_12.py

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

IMAGES
CREATED IN THE
PRE-COURSE
ACTIVITY

SHAPEFILE

NETCDFs
ACCESSED IN
THE PRE-COURSE
ACTIVITY

PYTHON
SCRIPTS

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>

gnc-a-caribbean/GOES at main · +
github.com/diegormsouza/gnc-a-caribbean/tree/main/GOES

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

diegormsouza / gnc-a-caribbean Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main / gnc-a-caribbean / GOES / Go to file Add file ...

diegormsouza Add files via upload	2ae9328 3 days ago	History
..		
IR4AVHRR6.cpt	Add files via upload	2 weeks ago
ne_10m_admin_1_states_provinces.dbf	Add files via upload	2 weeks ago
ne_10m_admin_1_states_provinces.shp	Add files via upload	2 weeks ago
ne_10m_admin_1_states_provinces.shx	Add files via upload	2 weeks ago
script_01.py	Add files via upload	2 weeks ago
script_02.py	Add files via upload	2 weeks ago
script_03.py	Add files via upload	2 weeks ago
script_04.py	Add files via upload	2 weeks ago
script_05.py	Add files via upload	2 weeks ago
script_06.py	Add files via upload	2 weeks ago
script_07.py	Add files via upload	3 days ago
script_08.py	Add files via upload	2 weeks ago
..		

DEMONSTRATION: RUNNING SCRIPTS LOCALLY

Running Scripts in the Cloud

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

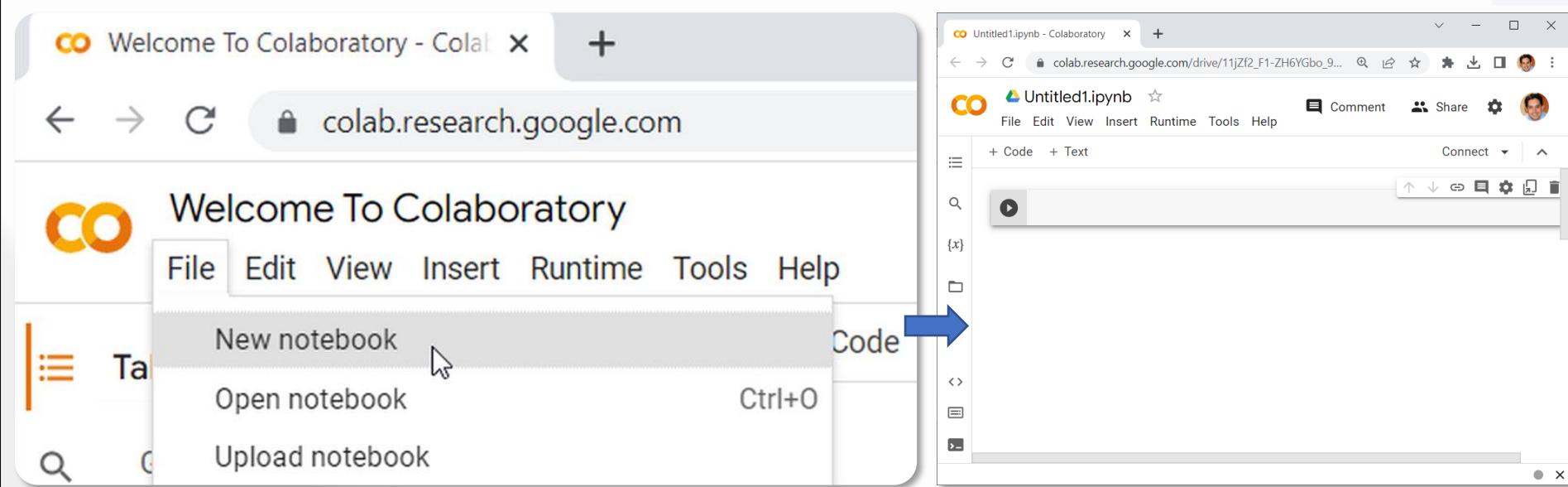


The screenshot shows a Google Colab notebook interface. The title bar reads "colab.research.google.com/drive/17yFOKezLEUhIziUUUXNlbYEwndp-gAMp?usp=sharing". The left sidebar contains a table of contents with sections like "Indice", "GOES-R DATA PROCESSING WITH PYTHON", "Introduction", and "Step 1: Checking the Virtual Machine Configuration". The main content area displays a banner for "GEONETCAST-AMERICAS TRAINING FOR EASTERN CARIBBEAN STATES" from April 24 - May 05 / 2023. Below the banner, there's a section titled "GOES-R DATA PROCESSING WITH PYTHON" developed by INPE / CGCT / DISSM - Brazil, dated March 2023. It features logos for INPE, MCTI, and VLabs. The content includes instructions for setting up a virtual machine, installing libraries, and various Python scripts for processing GOES-R data. A large "colab" watermark is visible in the bottom right corner.

GOOGLE COLAB: BASIC CONCEPTS

Getting started with Google COLAB:

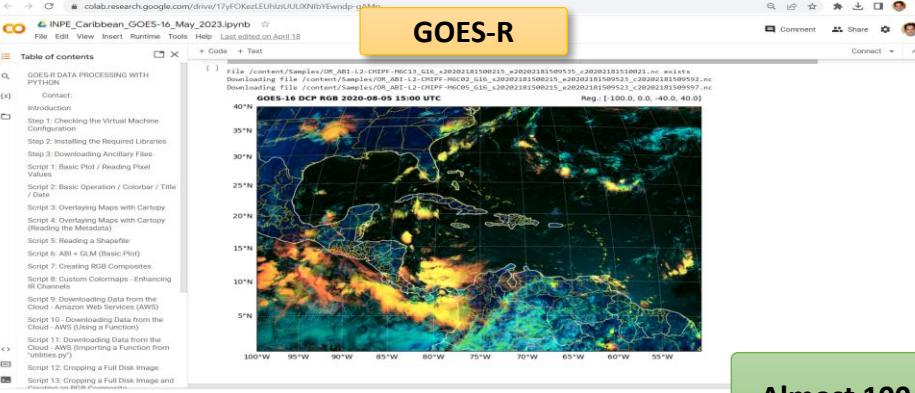
<https://colab.research.google.com/>



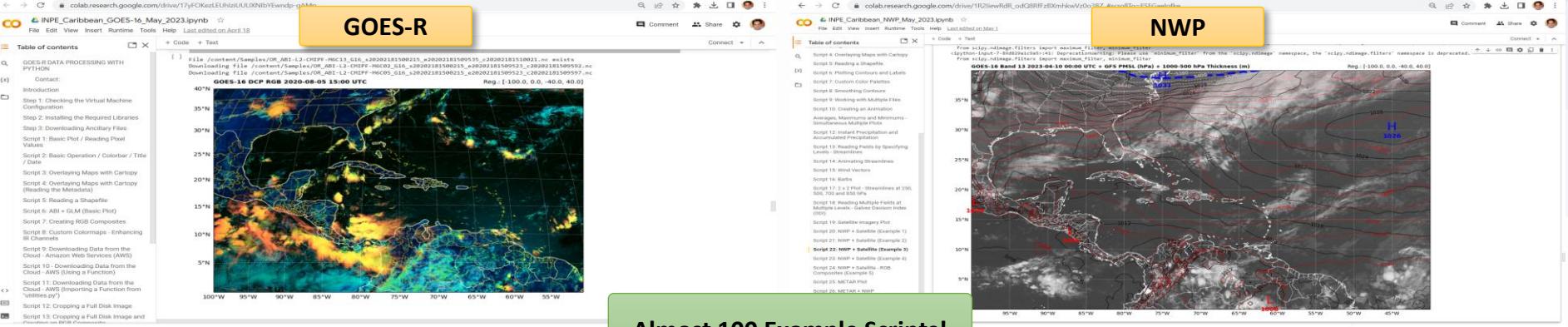


Running Scripts in the Cloud

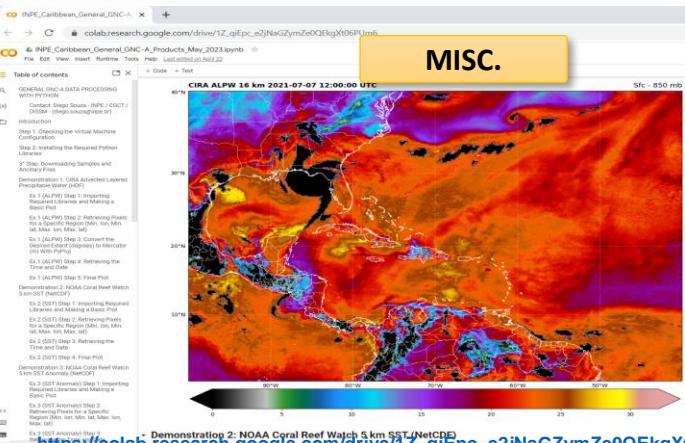
<https://colab.research.google.com/drive/17yFOKezLEUhIziUUUXNlbYEwndp-gAMp?usp=sharing>



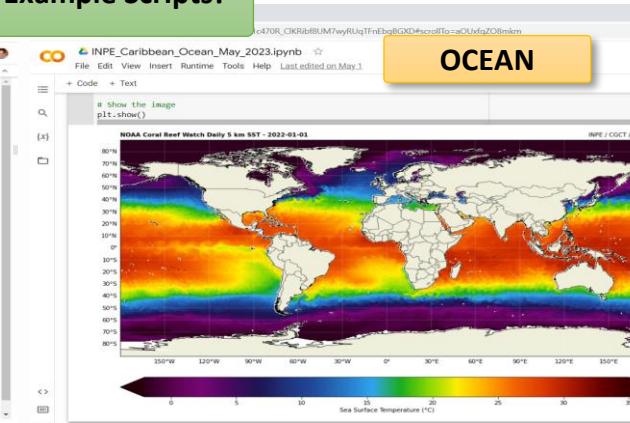
<https://colab.research.google.com/drive/17yFOKezLEUhIziUUUXNlbYEwndp-gAMp?usp=sharing>



Almost 100 Example Scripts!



https://colab.research.google.com/drive/1Z_qlEpc_e2jNaGZymZe0QEkgXt06PUm6?usp=sharing



https://colab.research.google.com/drive/1c470R_CIKRibf8UM7wyRUqTFnEbqBGXD?usp=sharing

Running Scripts in the Cloud



Create a copy on your Google Drive

INPE_Caribbean_GOES-16_Apr_2023.ipynb

File Edit View Insert Runtime Tools Help Last edited on March 31

Locate in Drive Open in playground mode New notebook Open notebook Upload notebook Rename Move Move to trash Save a copy in Drive Save a copy as a GitHub Gist Save a copy in GitHub Save Save and pin revision Revision history Download Print

Ctrl+S Ctrl+M S Ctrl+P

COLAB Notebook Version: March 2023

VERY IMPORTANT
Create a copy on your Google Drive

Run the instructions in each cell, sequentially

INPE_Caribbean_GOES-16_Apr_2023.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Table of contents

GOES-R DATA PROCESSING WITH PYTHON
Contact:
Introduction

Step 1: Checking the Virtual Machine Configuration
Step 2: Installing the Required Libraries
Step 3: Downloading Ancillary Files
Script 1: Basic Plot / Reading Pixel Values
Script 2: / Data
Script 3: / Read
Script 4: (Re)Process
Script 5: / Tiler
Script 6: / Tiler
Script 7: Creating RGB Composites
Script 8: Custom Colormaps - Enhancing IP Channels

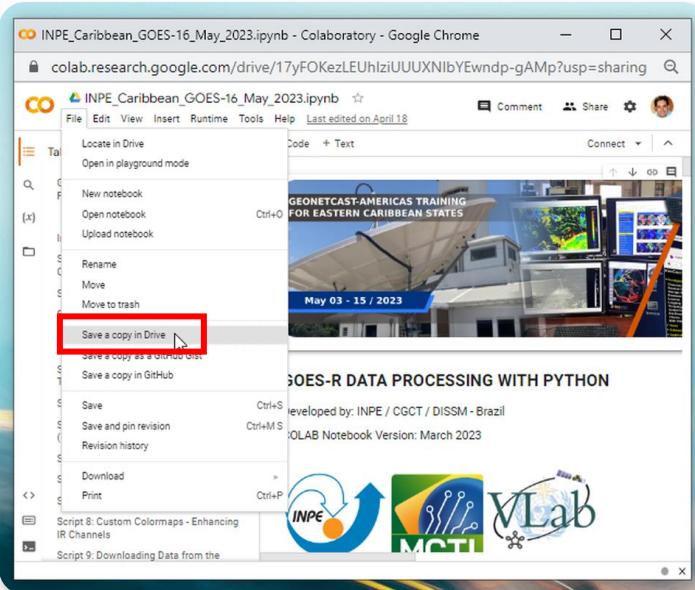
NOTE:
Even though you run multiple cells, the system waits for it to complete in sequence

A NICE SHORTCUT TO RUN CODE CELLS:
Ctrl + Enter

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

<https://colab.research.google.com/drive/17yFOKezLEUhIziUUUXNlbYEwndp-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/i8j1g9xg1g70561/utiliti  
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

INPE_Caribbean_GOES-16_Apr_2023.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Table of contents

GOES-R DATA PROCESSING WITH PYTHON

Contact:

Introduction

Step 1: Checking the Virtual Machine Configuration

Step 2: Installing the Required Libraries

Step 3: Downloading Ancillary Files

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

Add code and text cells

```
[ ] # 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')

# Downloading the Cartopy default maps from AWS (this is necessary due to a recent problems with the NaturalEarth server)
!wget https://raw.githubusercontent.com/SciTools/cartopy/master/tools/cartopy_feature_download.py
!python cartopy_feature_download.py physical
!python cartopy_feature_download.py cultural
!python cartopy_feature_download.py cultural-extra
print('\n')

# Installing the Boto3 library
!pip install boto3
print('\n')

# Installing / updating the GDAL library
!apt-add-repository -y ppa:ubuntugis/ubuntugis-unstable
!add-apt-repository -y ppa:ubuntugis/ppa
!apt-get install gdal-bin
!pip install 'gdal==3.0.4'
print('\n')

Requirement already satisfied: numpy>=1.18 in /usr/local/lib/python3.9/dist-packages (from cartopy) (1.22.4)
```

Running Scripts in the Cloud

The screenshot shows a Google Colab notebook interface. On the left is a sidebar with a 'Table of contents' section containing various script and step titles. A yellow box highlights the title 'Step 1: Checking the Virtual Machine Configuration'. The main area contains a code cell with several shell commands. To the right of the code cell is a blue box containing notes about command types. Below the code cell is its execution output showing system information. On the right side of the interface, there are four yellow boxes with arrows pointing to specific parts: 'Text Cells' points to the title, 'Memory and Disk' points to the top status bar, 'Code Cells' points to the code cell, and 'Code Cell Execution Output' points to the output box.

Step 1: Checking the Virtual Machine Configuration

```
[ ] # Verifying the installed o.s.  
!cat /etc/issue  
!uname -a  
print('\n')  
  
# Verifying the available RAM  
!grep MemTotal /proc/meminfo  
print('\n')  
  
# Verifying the available HD space  
!df -h  
print('\n')  
  
# Verifying the default Python installation directory  
!which python  
print('\n')  
  
# Verifying the Python version  
!python --version  
print('\n')
```

**- Commands started with “!”: Shell Commands (Linux)
- All other commands: Python**

Ubuntu 18.04.5 LTS \n \1
Linux cf96e0359c79 5.4.104+ #1 SMP Sat Jun 5 09:50:34 PDT 2021 x86_64 x86_64 x86_64 GNU/Linux
MemTotal: 13302928 kB

Text Cells

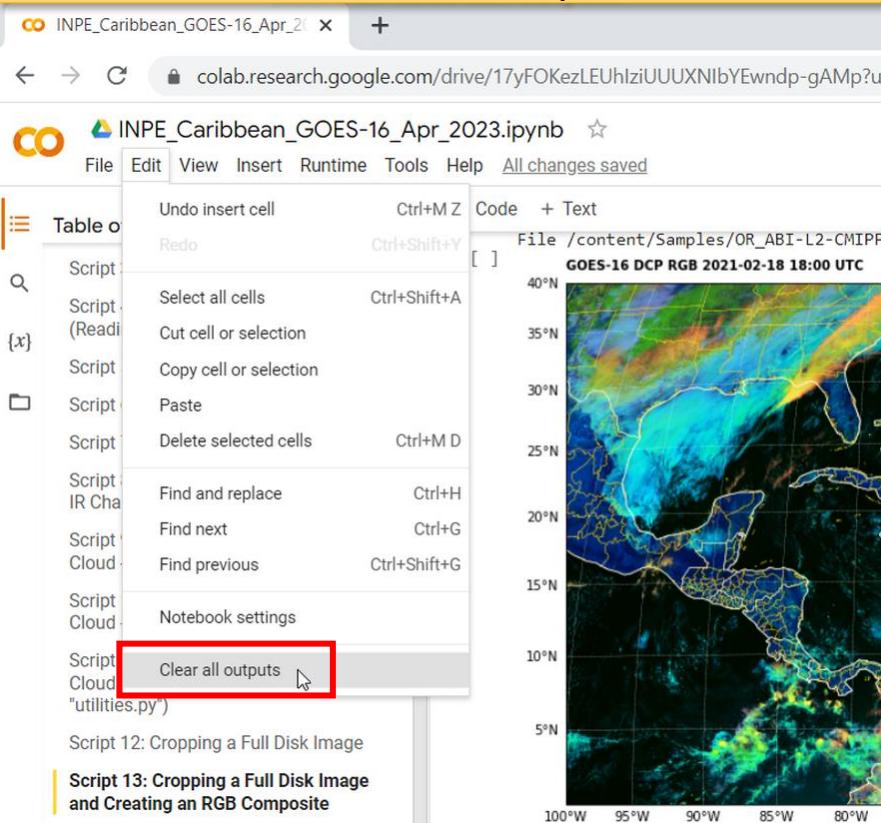
Memory and Disk

Code Cells

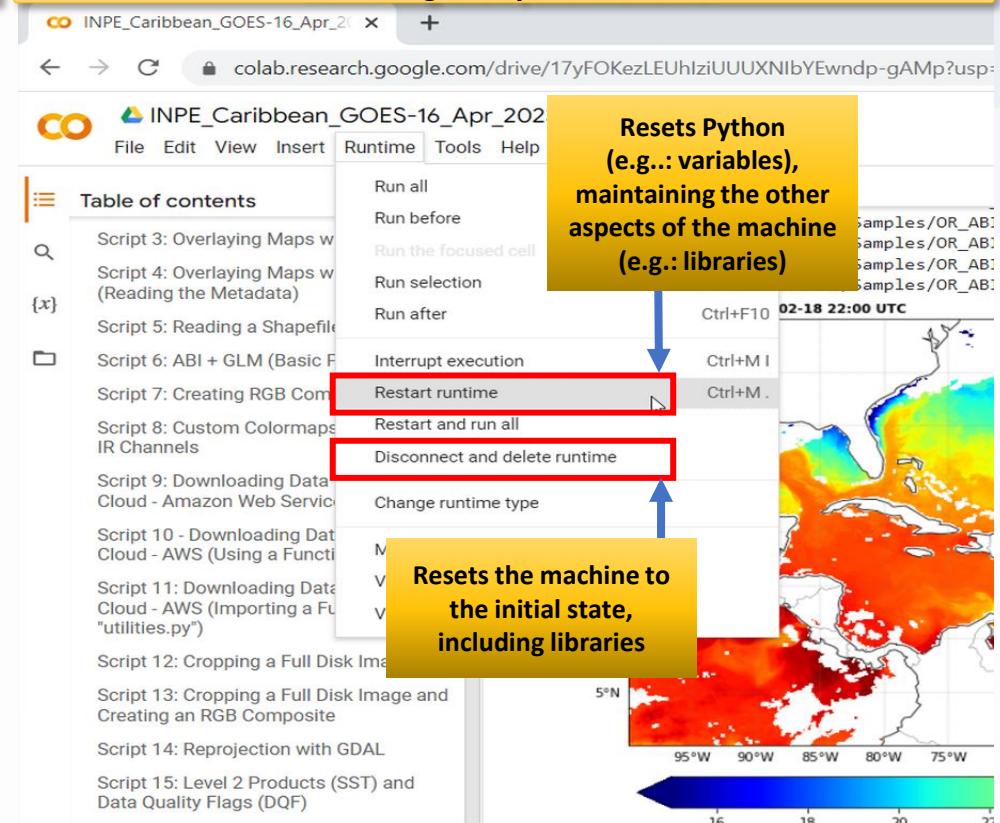
Code Cell Execution Output

Running Scripts in the Cloud

Clears all code cell outputs



Restarting the Python runtime



Running Scripts in the Cloud

COLAB Work Directory: "/content"

```
+ Code + Text
Downloading file /content/Samples/OR_ABI-L2-RQPF
[ ] <ipython-input-20-63d040c2185b>:10: MatplotlibDeprecationWarning: colormap = cm.get_cmap('rainbow', 240)
G-16 ACCUM. PREC.
35°N
30°N
25°N
20°N
15°N
10°N
5°N
95°W 90°W 85°W 80°W 75°W 70°W 65°W
Rainfall Rate mm / 24h
Script 18: GLM Density
```

View Table of Contents

View Directories



"/content" Directory

Generated Files

- GOES-R DATA PROCESSING WITH PYTHON
- GEONETCAST-AMERICAS TRAINING FOR EASTERN CARIBBEAN STATES
- April 24 - May 05 / 2023
- INPE-CLOUD COMPUTING FOR DISASTER MONITORING
- MCTI
- VLab

Contact:

Diego Souza - INPE / CGCT / DISSM - (diego.souza@inpe.br)
<https://github.com/diegomsouza/gnc-a-caribbean/tree/main/GOES>

Introduction

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)

