GNC-A

GEONETCast-Americas

*Delivering Environmental Data to Users in the Americas*

# SHOWCast

## SHOWCast Installation Manual

October 26, 2021



NOAA — National Oceanic and Atmospheric Administration — U.S. Department of Commerce

INPE

AmeriGEOSS

Implementing GEOSS in the Americas

# CONTENTS

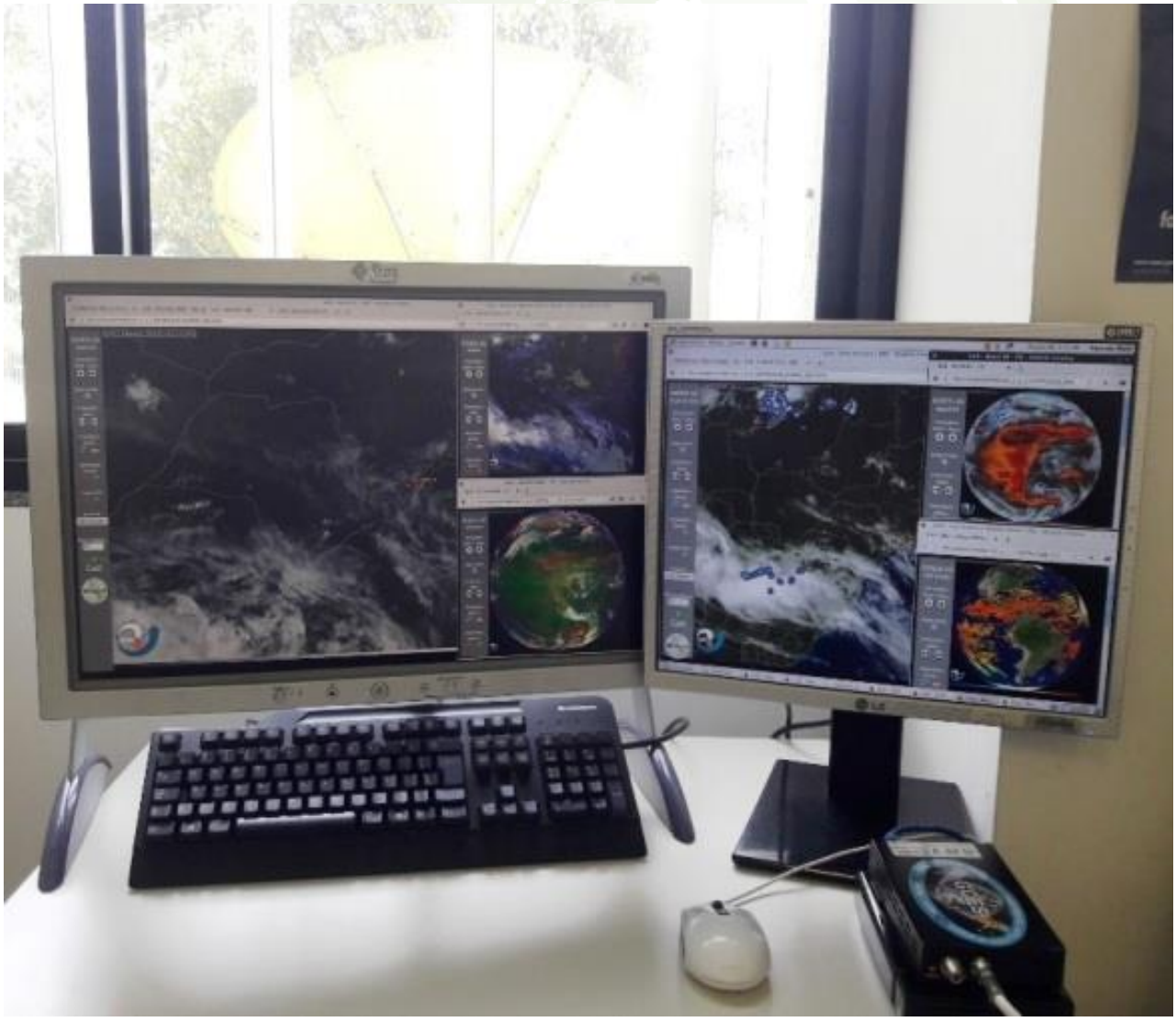


Fig. 1: SHOWCast running at INPE - National Institute for Space Reseach - Brazil (January 16, 2020).

# 1 INTRODUCTION

This document will introduce SHOWCast (**S**imple **H**TML **O**perational **W**rapper for GEONET**Cast**-Americas), a processing and visualization tool for GEONETCast-Americas users created by Diego Souza (INPE - Brazil). SHOWCast provides a basic HTML structure for product selection and animation, and Python scripts to convert satellite data into imagery automatically. SHOWCast can also be used with other satellite reception mechanisms like GRB, Amazon AWS and UNIDATA THREDDS. The package runs on both Windows and Linux operational systems.

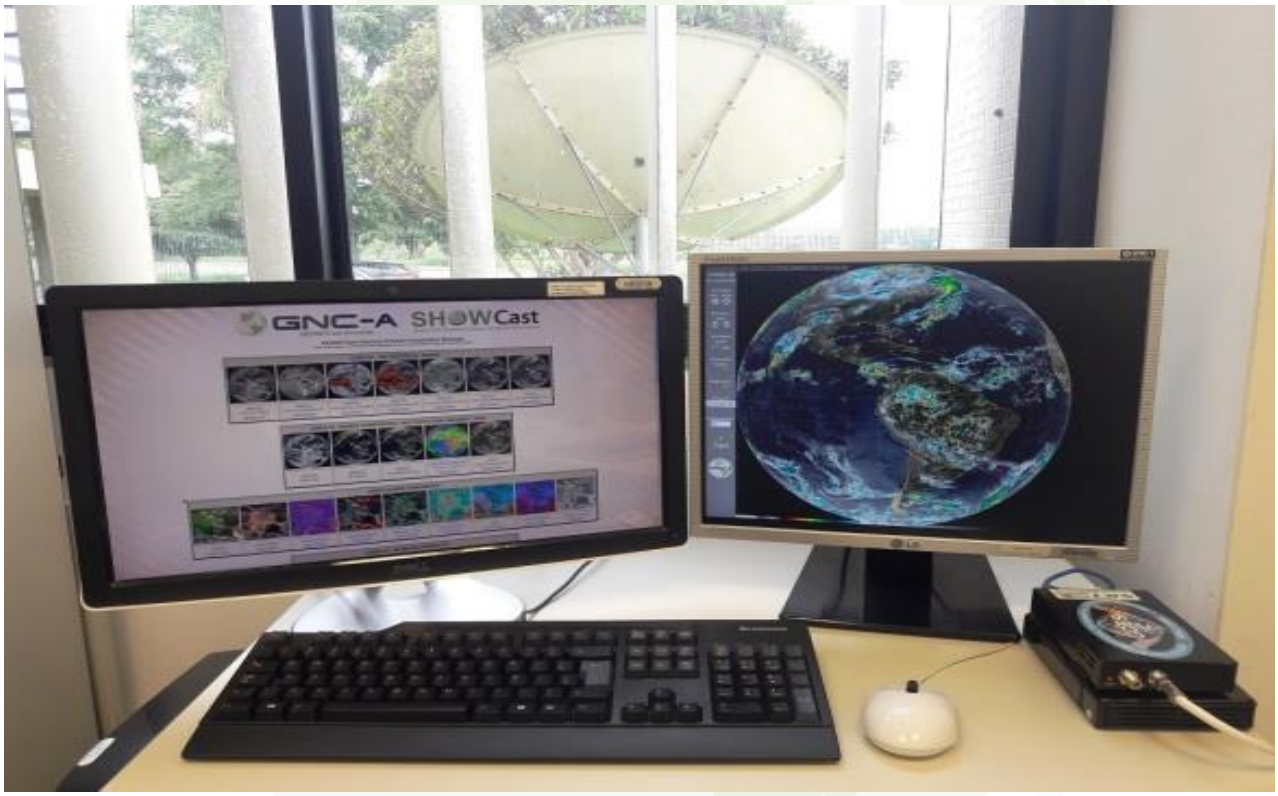



Fig. 2: First version of SHOWCast running in a GNC-A station (November 6, 2019).

## 1.1 Objective

The primary goals for the development of SHOWCast are: Provide a free tool that can be customized (both processing and visualization) and put into operations without the need of having a BIG knowledge in programming and web development (human resources issue) and provide a free tool that can be adapted to the available hardware (technology resources issue).

## 2 DOWNLOADING SHOWCAST

You may download the latest version of SHOWCast at the following link:

https://geonetcast.wordpress.com/showcast/

A compressed file called **"SHOWCast_v_X_X_X.zip"** will be downloaded (the "X"'s will be the package version number). You may extract it anywhere in your machine. All the necessary files for running SHOWCast will be stored at this directory.

## 3 SHOWCAST DIRECTORY STRUCTURE

After unzipping the package, you will see the following directory structure when accessing the SHOWCast directory for the first time:

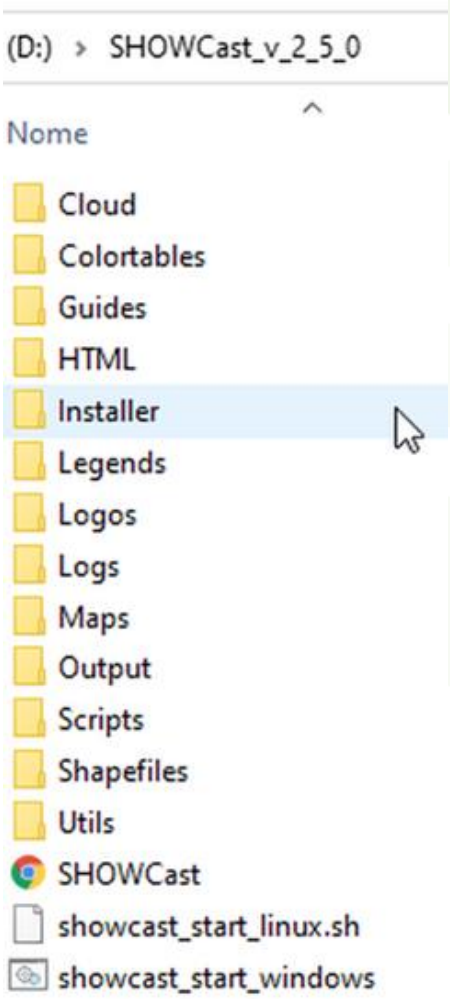



Fig. 3: The SHOWCast directory.

**Cloud:** SHOWCast "Cloud" Module

**Colortables:** Some color palettes used by part of the Python scripts.

**Guides:** "Quick Guides" (PDF format) for various satellite products visualized.

**HTML:** The HTML and visualization structure.

**Installer:** The SHOWCast installation files.

**Legends:** Some legends used by part of the plots.

**Logos:** Logos used by the scripts.

**Logs:** Log files (what have been already processed).

**Maps:** Background maps used by part of the scripts.

**Output:** Historical plots generated by SHOWCast (they are not part of the animation or HTML structure).

**Scripts:** Python scripts used to process the products.

**Shapefiles:** Shapefiles used by the scripts.

**Utils:** Software utilities (e.g.: third-party)

**SHOWCast.html:** SHOWCast visualization interface.

**showcast_start_*:** Start the SHOWCast processing on Windows (".bat") or Linux (".sh").

## 4 OPENING THE SHOWCAST INTERFACE FOR THE FIRST TIME

If you double-click at the **"SHOWCast.html"** icon, the SHOWCast visualization interface will be opened.

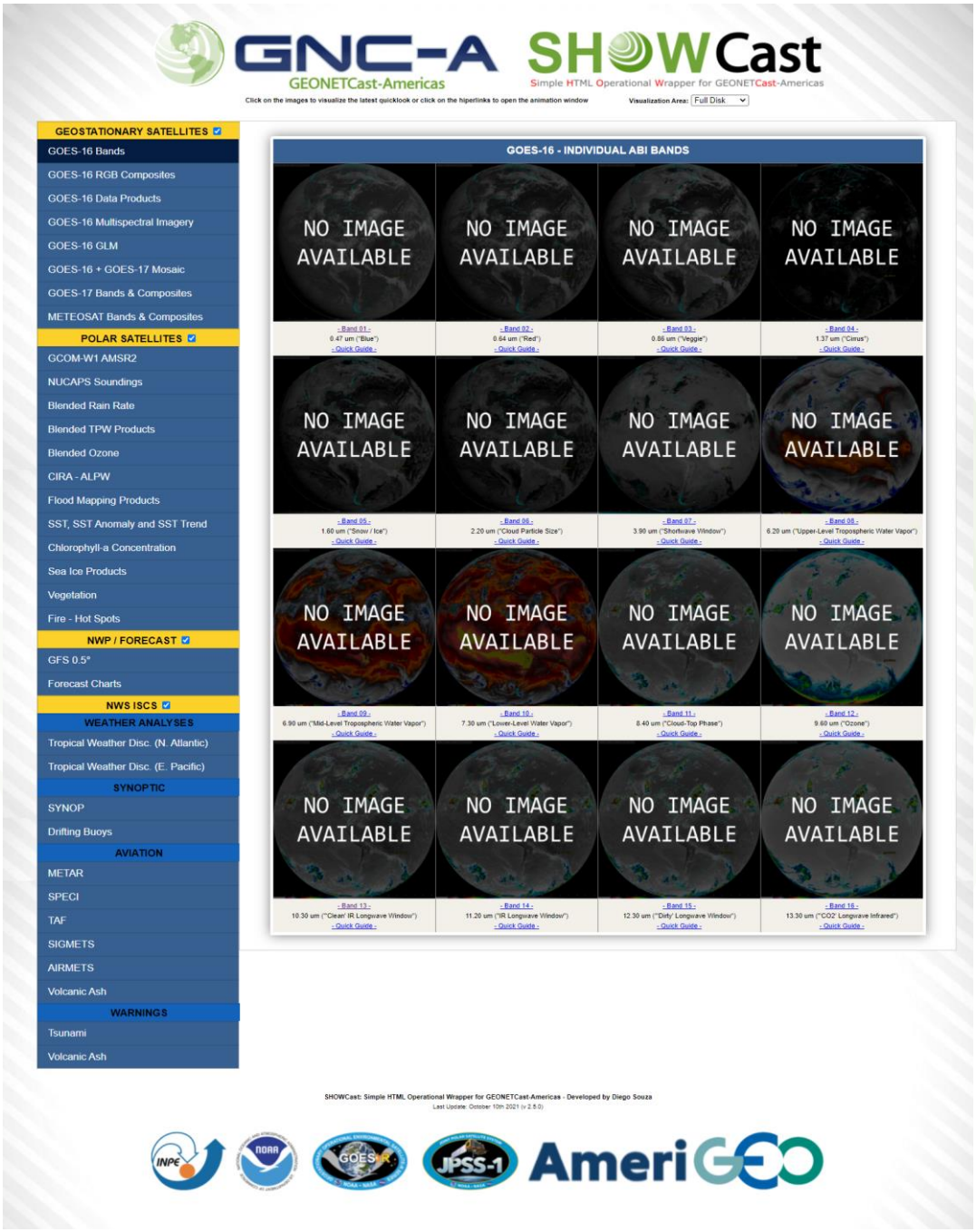



Fig. 4: The SHOWCast visualization menu opened for the first time

On the left side of the page, you may navigate through the product categories. You will see the message **"NO IMAGE AVAILABLE"** for all the product thumbnails, because we didn't activate the SHOWCast processing yet.

## 5 INSTALLING THE SHOWCAST PROCESSING MODULE

SHOWCast uses Python to process information, and provides an easy way to install Python and all the libraries needed. First, access the **"Installer"** folder in the SHOWCast main directory:

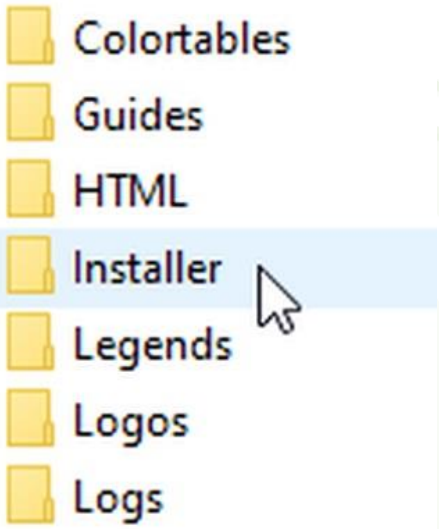



Fig. 5: Accessing the SHOWCast "Installer" folder

Inside this folder, we have two installation files, one for Windows (**showcast_install_windows.bat**) and one for Linux (**showcast_install_linux.sh**). Also, we have a folder called "Miniconda3" (where the installation files are found). There's no need to access this "Miniconda3" folder.

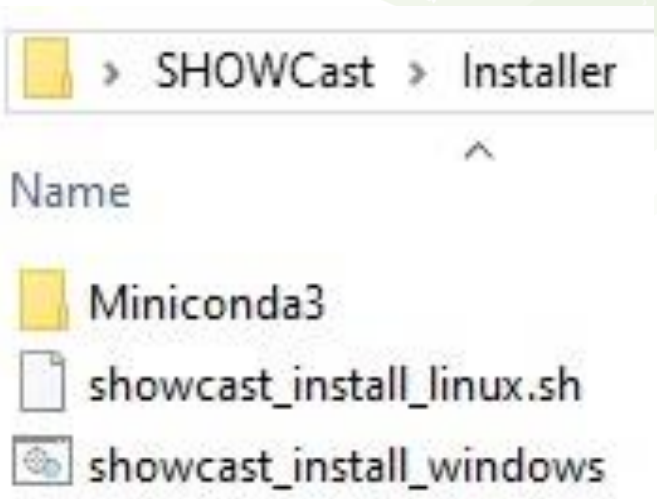



Fig. 6: Installation scripts for Linux and Windows operational systems

### 5.1 Installing on Windows

If you are installing on Windows, just double click the **"showcast_install_windows.bat"**.

### 5.2 Installing on Linux

If you are using Linux, please change the permissions of the SHOWCast subfolders:

**chmod -R 777 ***

And execute the **"showcast_install_linux.sh"** script: **./showcast_install_linux.sh**

### 5.3 The SHOWCast Installer Terminal

On **both Windows or Linux**, the SHOWCast installer will show up. It has the same structure, **independent if you are using Windows or Linux**:

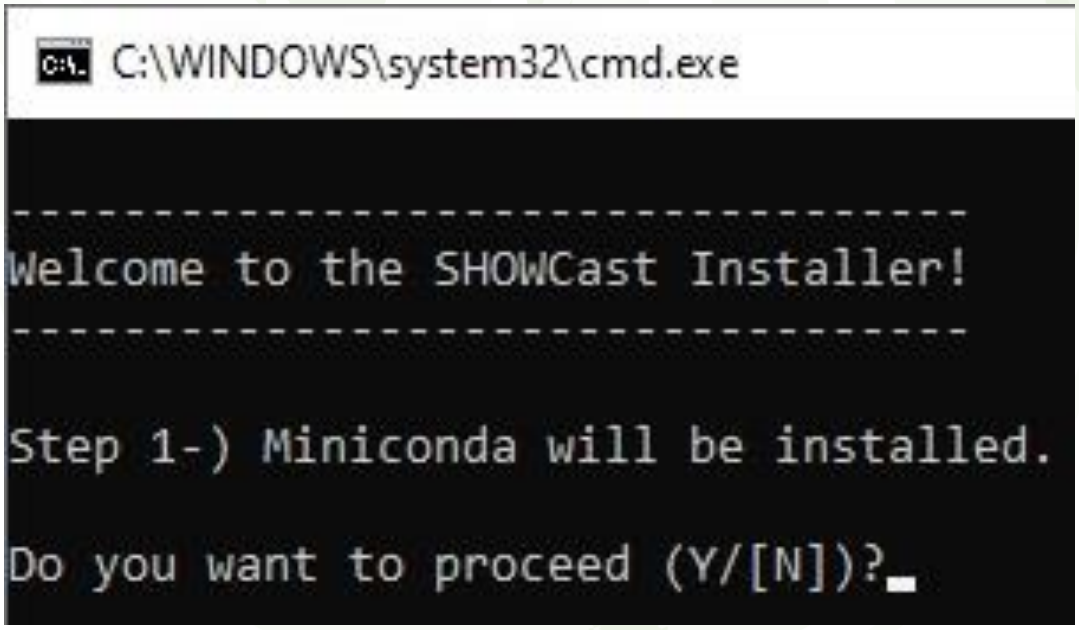



Fig. 7: The SHOWCast installer prompt

First, the prompt will ask if you want to proceed with the Miniconda installation. Miniconda provides everything we need to process data: Python (the programming language used to process our data), a "Library Manager" (used to install the required Python libraries) and a Virtual Environment Manager (where our libraries will be installed). Enter 'y' + 'Enter' (or just 'y' on Linux) and Miniconda will be installed automatically (this will take some minutes).

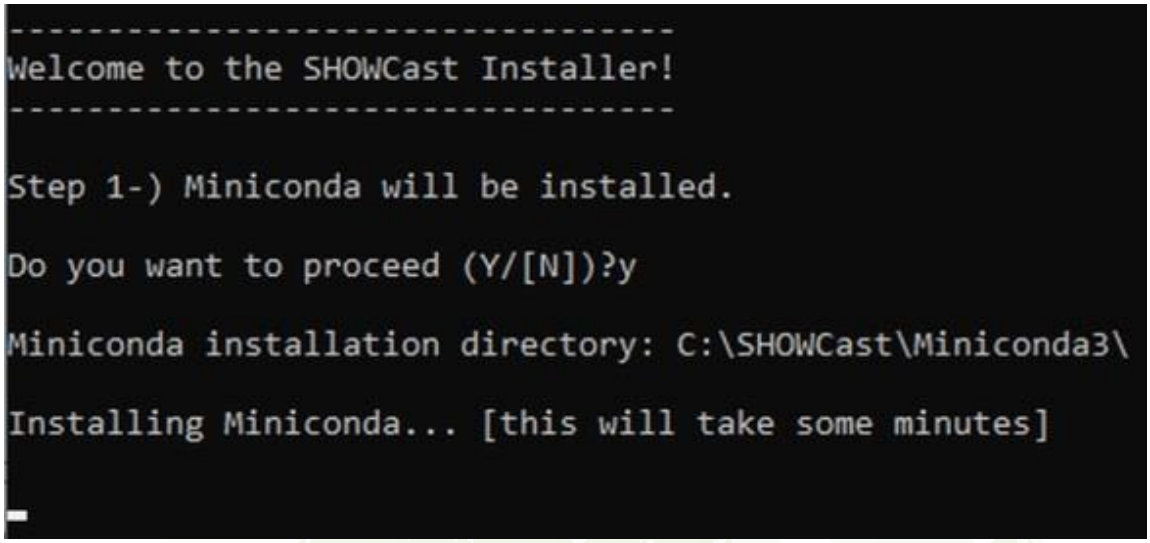



Fig. 8: Miniconda being installed automatically by the SHOWCast installer

After this step, a new “Miniconda3” folder will appear in the SHOWCast main directory. This is where all Python related files will be. There’s no need to access this folder.

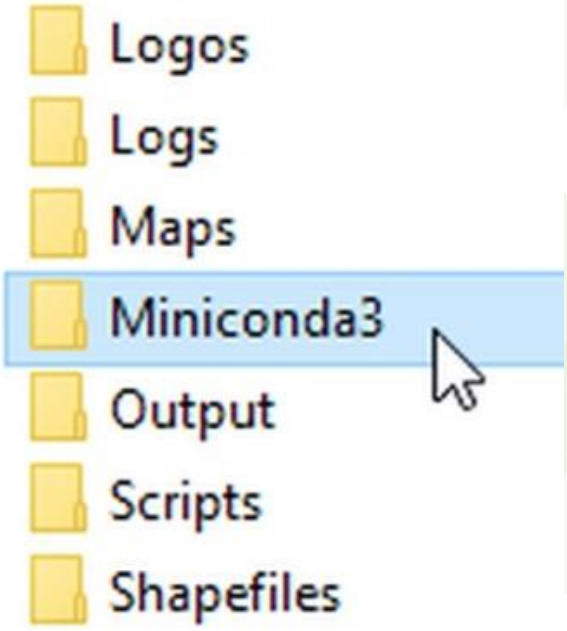



Fig. 9: New “Miniconda3” folder at the SHOWCast main directory

When the Miniconda installation is finished, the prompt will ask if you want to proceed with the SHOWCast environment installation. This step will install the Python libraries in a virtual environment called “showcast”. Enter ‘y’ + ‘Enter’ (or just ‘y’ on Linux) and the environment will be created automatically.

This step could take a considerable amount of time, depending on your internet and hardware capabilities.

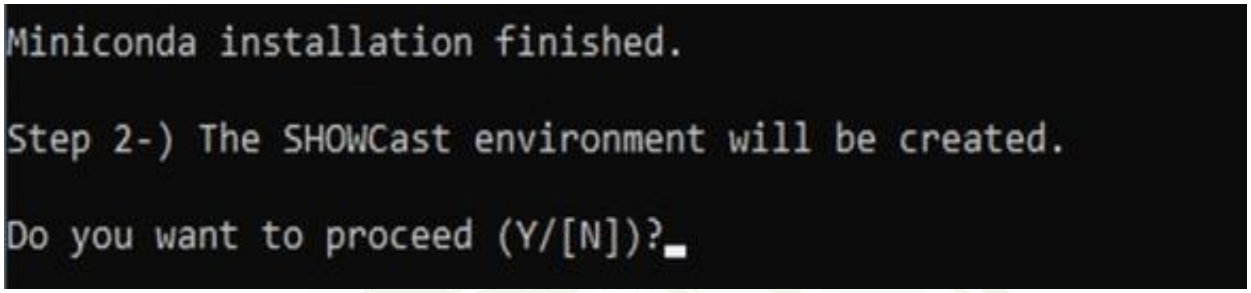



Fig. 10: Installing the SHOWCast virtual environment

This images below show the installation in process. First, the Python libraries that will be installed are listed. The following Python packages and its dependencies are set to be installed:

**affine, cartopy, folium, gdal, geopandas, glymur, matplotlib, metpy, netcdf4, pandas, pygrib, pyhdf, pyorbital, pyproj, pyresample, rasterstats, satpy, siphon, pip and pillow.**

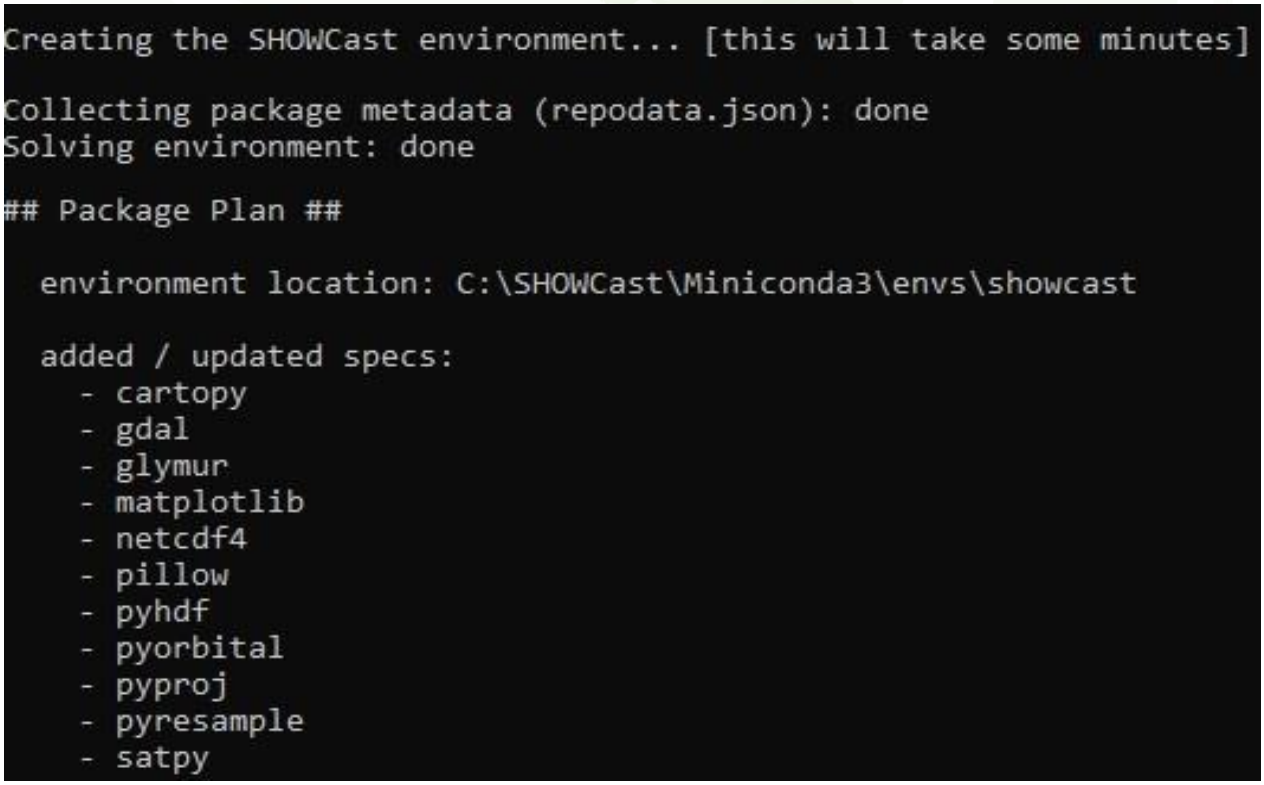



Fig. 11: SHOWCast environment creation (the libraries are listed)

Then, the libraries are installed:

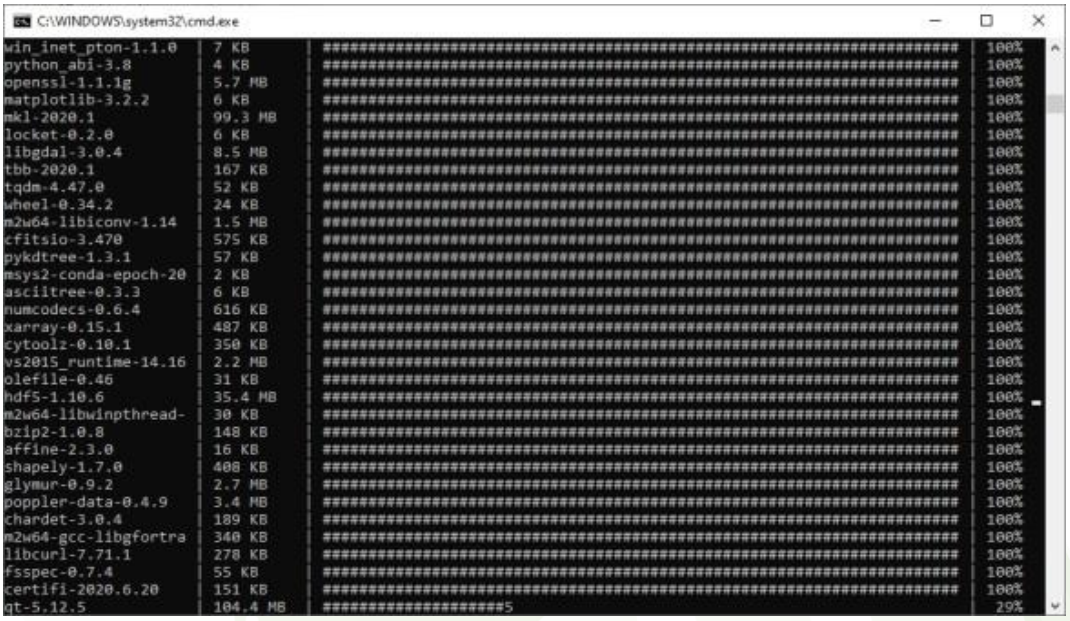



Fig. 12: SHOWCast environment creation (the libraries are installed)

After some minutes, when you see the following message, the SHOWCast processing modeule installation has been finished.

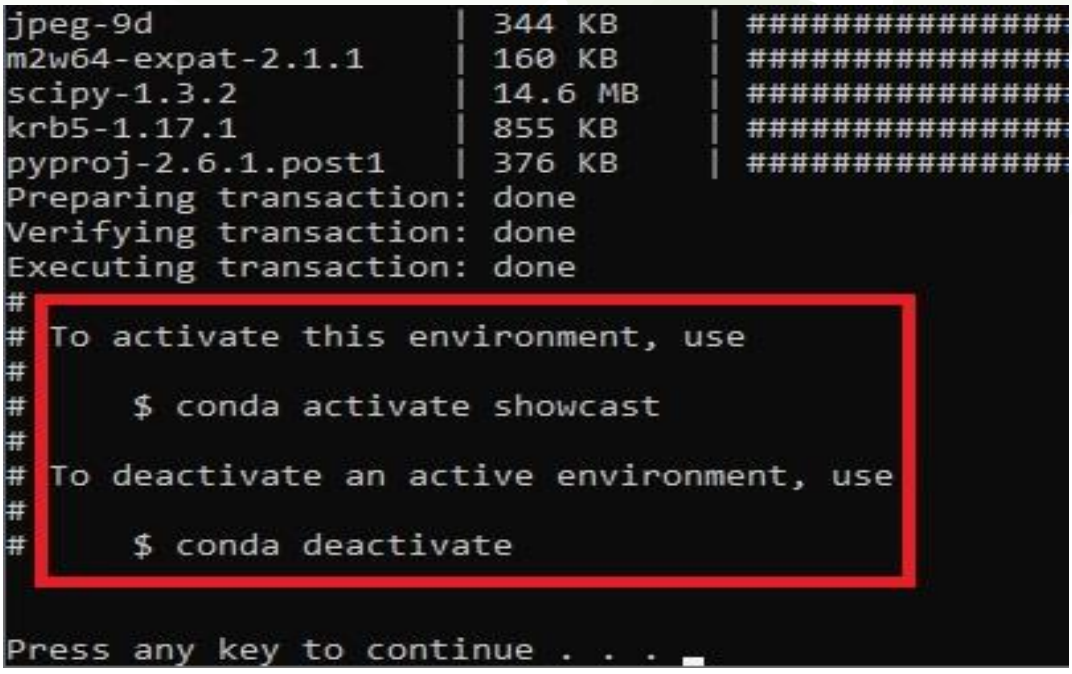



Fig. 13: SHOWCast installation finished

At this point, SHOWCast is installed. There's no need to execute any commands, just close the terminal. Now we can proceed with the SHOWCast configuration.

# 6 BASIC SHOWCAST CONFIGURATION

We need to configure SHOWCast so it processes the data according to our needs. Basically, we have four files that need to be configured:

1-) SHOWCast_v_X_X_X\Scripts**showcast_start.py**: On this script, we'll configure our **data ingestion directory** (from GNC-A, GRB, Amazon, etc.).
2-) SHOWCast_v_X_X_X\Scripts**showcast_config.py**: On this script, we'll configure **which data we want to process and how we want to process** (region, resolution, etc.).

3-) SHOWCast_v_X_X_X\Scripts**showcast_cleaner.py**: This is optional. On this script, we configure the **automatic deletion of historical files (both ingestion and or processing).**
4-) SHOWCast_v_X_X_X**showcast_start_linux.sh**

or

SHOWCast_v_X_X_X**showcast_start_windows.bat**:

On this script, we select the **number of parallel processed that will be started by SHOWCast** (more details on 6.4).

Let's see how to configure each file:

## 6.1 Configuring the showcast_start.py file

In the variable **"ingest_dir"**, insert the name of the directory where you are ingesting data. By default, it is configured as **'D://data//fazzt//'**. Change it according to your needs.
In the variable **"vis_dir"**, insert the name of the directory where you want your images to be stored. By default, it is configured as **showcast_dir + '//HTML//Output//'**. There's no need to change this, except if you want to have multiple machines running SHOWCast in your internal network.

Some important considerations:

- Do not use directories that has spaces in their names (e.g: 'D://my data//fazzt//').
- It is mandatory to use double slashes ('//'). Windows use backslashes '\' and Linux use forward slashes '/'. By using double slashes, the code will work for both O.S.

- It is mandatory to use double slashes (‘//’) at the end of the variable.
- You may use network addresses like: ‘//192.168.10.1//fazzt//’ for both variables.

```
#-----------------------------------------------------------------------------------------------------------
# GEONETCast-Americas ingestion directory (AVOID USING DIRECTORIES WITH SPACES)
ingest_dir = 'D://data//fazzt//'      # Windows Example - Change it according to your GNC-A Station

# SHOWCast visualization directory
vis_dir = showcast_dir + '//HTML//Output//'
#-----------------------------------------------------------------------------------------------------------
```

Fig. 14: Configuring the showcast_start.py variables

## 6.2 Configuring the showcast_config.py file

Let’s configure the data we want to process and how we want to process them.

For each product that SHOWCast has been tested and configured to process, you will find the following block of code inside the **showcast_config.py** script:

```
#-----------------------------------------------------------------------------------------------------------
g16_band13_sec            = True # GOES-16 L2 CMI - Band 13 - USER SECTOR

g16_band13_sec_process    = 1                                                    # Process cicle for this product
g16_band13_sec_directory  = ingest_dir + 'GOES-R-CMI-Imagery//Band13//'          # Folder where the data is found
g16_band13_sec_identifier = '*L2-CMIPF-M*C13_G16*.nc'                            # Unique string on the file name
g16_band13_sec_max_files  = 1                                                    # Max number of historical files to be processed
g16_band13_sec_extent     = [-63.0, -35.0, -35.0, -10.0]                         # [min_lon, min_lat, max_lon, max_lat]
g16_band13_sec_resolution = 2  # Max Res.: 2 km                                  # Final plot resolution
g16_band13_sec_interval   = '00,10,20,30,40,50'                                  # Processing interval
g16_band13_sec_config     = '_SEC'                                               # Configuration string
g16_band13_sec_script     = showcast_dir + '//Scripts//process_g1X_bands_sec.py' # Script to activate
g16_band13_sec_output     = showcast_dir + '//Output//'                          # Output folder

products.append('g16_band13_sec') # Add the product to the list
#-----------------------------------------------------------------------------------------------------------
```

Fig. 15: Configuring the showcast_config.py variables (sectorized GOES-16 Band 13)

For each block of code, you will see the following variables:

**sss_pppppp_rrr** = **True** or **False**

where:

**sss:** three letters to designate the satellite or category

**pppppp:** six letter to designate the product type

**rrr:** three letter to designate the region (fdk - Full Disk, or sec - Sectorized)

example: **g16_band13_sec** (sectorized GOES-16 Band-13)

If set as **True**, this product will be processed, if set as **False**, it will not be processed.

**sss_pppppp_rrr_process:** Number of the parallel processing cycle this product will be processed (more on item 6.4). The default value for all the products is 1 (a single processing cycle). By changing this value for different group of products, you may optimize your processing scheme.

**sss_pppppp_rrr_directory:** Sub directory within the ingestion directory this product is found.

**sss_pppppp_rrr_identifier:** A unique part of the file names so this file can be detected by the scripts.

Example: '*L2-CMIPF-M*C13_G16*.nc'

**sss_pppppp_rrr_max_files:** Maximum number of non processef files SHOWCast will process in a single run.

**sss_pppppp_rrr_extent:** For the sectorized products, the region you would like to plot (min. lon, min. lat, max. lon, max. lat).

Example: [-55.0, -25.0, -40.0, -10.0]

**sss_pppppp_rrr_resolution:** The desired plot resolution in km.

**sss_pppppp_rrr_interval:** For the GOES-R products, the scan minute interval we want to plot (00, 10, 20, 30, 40 and / or 50).

Example: [‘10’,’20’,’30’,’40’,’50’] - Note: Do not use spaces!

**sss_pppppp_rrr_config:** A configuration string used by the scripts. Mainly, this is used to differentiate sectorized products and products that use the same files to be produced (so it may be included in the log file, even if it is using the same file as other products).

**sss_pppppp_rrr_script:** The script used to process this product.

**sss_pppppp_rrr_output:** Where the historical plots will be stored. You may use network addresses like: ‘//192.168.10.1//Output//’ to direct the plot to another machine in your network.

### 6.3 Configuring the showcast_cleaner.py file

Let's configure if and how we want to delete historical data.

In the variable **"ingest_dir"**, insert the name of the directory where you are ingesting data. By default, it is configured as **'D://data//fazzt//'**. Change it according to your needs.

In the variable **"delete_historical_output"**, you select if you **want to delete** SHOWCast historical data (**True**) or if you **do not want to delete** SHOWCast historical data (**False**). By default, this is set as **True**.

In the variable **"delete_historical_ingest"**, you select if you **want to delete** your ingestion historical data (**True**) or if you **do not want to delete** ingestion historical data (**False**). By default, this is set as **False**.

In the variables **"max_days_output"** and **"max_hours_output"** you configure how many days and hours you want to store the **SHOWCast** data. By default, these are configured to 3 days and 0 hours.

In the variables **"max_days_ingest"** and **"max_hours_ingest"** you configure how many days and hour you want to store the **ingestion** data. By default, these are configured to 3 days and 0 hours.

```
#-----------------------------------------------------------------------------------------------------------
# USER CONFIGURATION BEGIN
#-----------------------------------------------------------------------------------------------------------
#-----------------------------------------------------------------------------------------------------------
# GEONETCast-Americas ingestion directory (AVOID USING DIRECTORIES WITH SPACES)
ingest_dir = 'D://data//fazzt//'      # Windows Example - Change it according to your GNC-A Station

# To delete historical files in the output folder, set as True
delete_historical_output = True
# To delete historical files in the ingest folder, set as True
delete_historical_ingest = False

# Number of days and hours to keep files in the Output directory
# The number of hours will be added to the number of days
# e.g: Delete files older than 5 hours (max_days = 0 / max_hours = 5)
# e.g: Delete files older than 1 day and 2 hours (max_days = 1 / max_hours = 2)
max_days_output = 3
max_hours_output = 0

# Number of days and hours to keep files in the Ingest directory
# The number of hours will be added to the number of days
# e.g: Delete files older than 5 hours (max_days = 0 / max_hours = 5)
# e.g: Delete files older than 1 day and 2 hours (max_days = 1 / max_hours = 2)
max_days_ingest = 3
max_hours_ingest = 0
#-----------------------------------------------------------------------------------------------------------
#-----------------------------------------------------------------------------------------------------------
# USER CONFIGURATION END
#-----------------------------------------------------------------------------------------------------------
#-----------------------------------------------------------------------------------------------------------
```

Fig. 16: Configuring the showcast_cleaner.py variables

### 6.4 Configuring the showcast_start_windows.bat or showcast_start_linux.sh file

We saw on item 6.2 that we may configure in which parallel processing cycles we would like to add a given product. In order to configure how many parallel processing cycles showcast will create when executed, change the **num_process** variable in the showcast_start_windows.bat or showcast_start_linux.sh script.

```
4  :: Select the number of parallel SHOWCast processes
5  SET /A num_process=1
```

Fig. 17: Changing the number of parallel processing cycles for Windows

```
2  # Select the number of SHOWCast parallel processes
3  declare -i num_process=1
```

Fig. 18: Changing the number of parallel processing cycles for Linux

## 7 ADVANCED SHOWCAST CONFIGURATION

### 7.1 Parallel processing

In older versions of SHOWCast, processing products sequentially was the only option, and this could cause huge delays in the processing scheme. If one of the products take long to process, all the other subsequent products would be delayed.

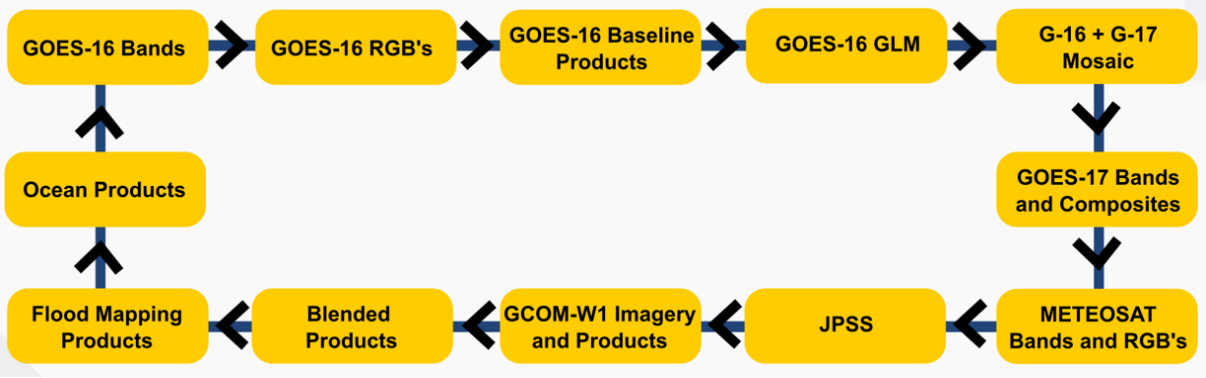



Fig. 19: Sequential processing in old versions of SHOWCast

In newer versions of SHOWCast, we have the option to create multiple parallel process, as seen in 6.4 and select which group of products will be processed in each parallel cycle, as seen in 6.2.

This allows users to test different setups and optimize the solution for their hardware.

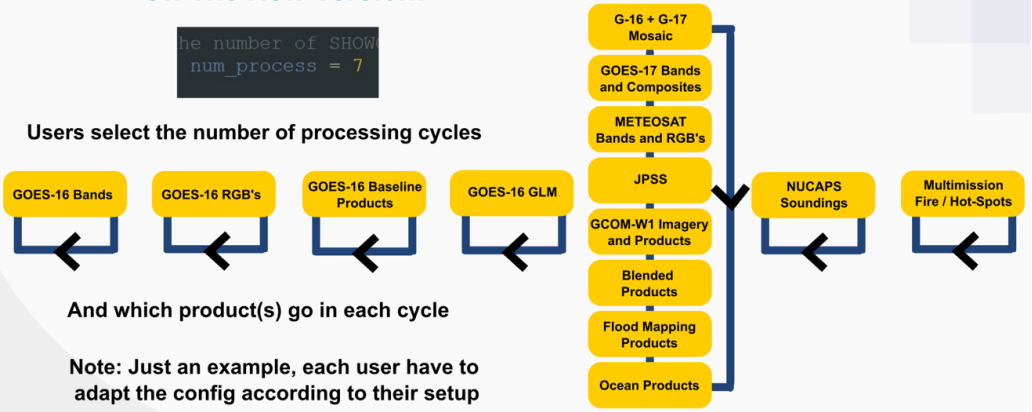



Fig. 20: Parallel processing in newer versions of SHOWCast

Any number of processes can be configured in the **showcast_start_windows.bat** or **showcast_start_linux.sh** and any number of products can be added to each process. By default, SHOWCast has a single process ("num_process" = 1) and all the products are processed in this single process ("sss_pppppp_rrr_process" = 1). Users should change this accordingly. A suggestion is to have very frenquent files (like GLM data) in a dedicated process. Also, products that demands processing power (like NUCAPS) is also recommended to have its own cycle.

## 7.2 Network configuration

In older versions of SHOWCast, it was not possible to share SHOWCast files running in different workstations in a local network. We had the possibility of running the ingestion, processing, storage and visualization in a single workstation:

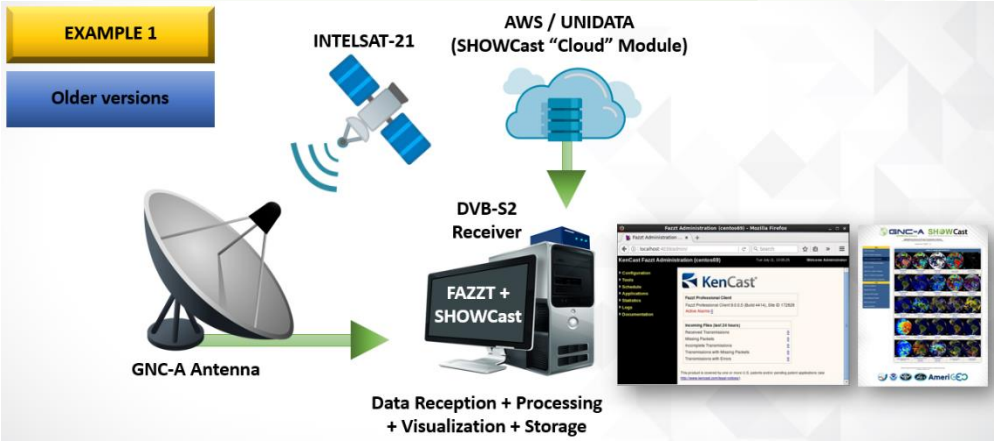



Fig. 21: Ingestion, processing, storage and visualization in a single workstation

We also have the possibility to run the ingestion in one workstation and the processing, storage and visualization in another workstation, using a network address when configuring the **"ingest_dir"** variable in the **showcast_start.py** script (item **6.1**), using for example (**ingest_dir = '//192.168.10.1//fazzt//'**) in the second workstation.

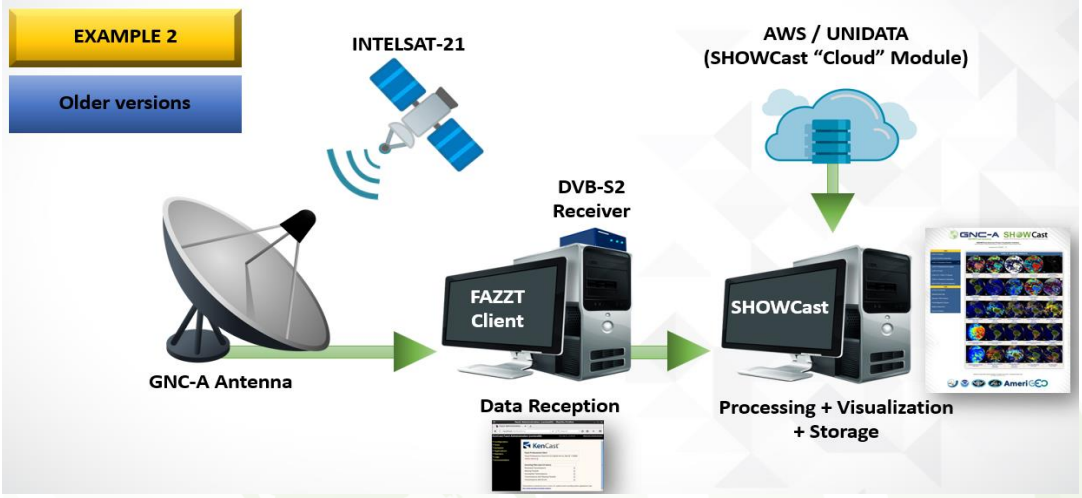



Fig. 22: Ingestion in one workstation, processing, storage and visualization in a second workstation

In newer versions of SHOWCast, it is possible to have multiple workstations sharing SHOWCast data. If we configure the **"vis_dir"** variable in the **showcast_start.py** script (item **6.1**) using a network address, we could have one workstation ingesting data, another workstation for data processing only, and another workstation for both data processing and visualization. By doing this we can split the data processing in multiple machines.

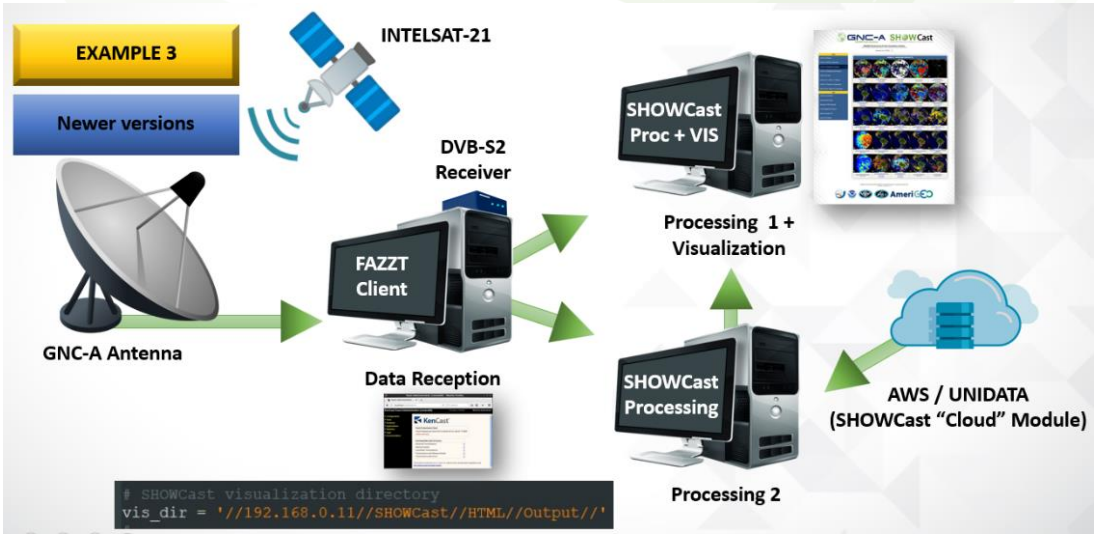



Fig. 23: Ingestion in one workstation, processing and visualization split in two workstations

Another possibility is to direct the historical plots to a dedicated workstation, using a network address when configuring the **sss_pppppp_rrr_output** variable for each product in the **showcast_config.py** (item **6.2**).

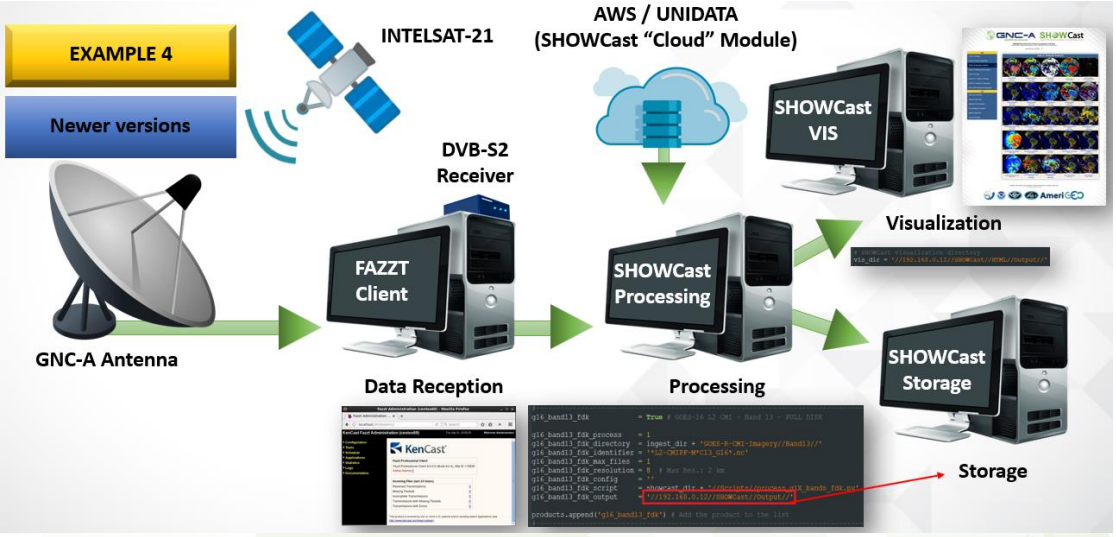



Fig. 24: Configuring a dedicated workstation for historical data storage

## 8 STARTING THE SHOWCAST PROCESSING MODULE

To start the SHOWCast processing module, in the SHOWCast main directory, if you are using Windows, just double click the **showcast_start_windows.bat**, if you are using Linux, execute the **showcast_start_linux.sh** script (**./showcast_start_linux.sh)**.

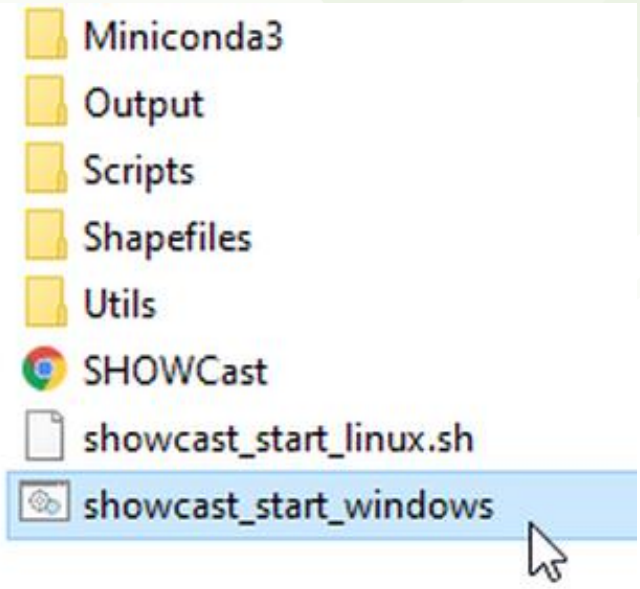



Fig. 25: Starting the SHOWCast processing module

When starting the processing module, the following terminals will be open:

- A main terminal that will call the others
- One terminal for each parallel process configured on **showcast_start.py "num_process"** variable (**6.4**). Each parallel cycle will process the products as configured on **showcast_config.py**, in the **sss_pppppp_rrr_process** variable (**6.2**).
- One terminal for the **showcast_cleaner.py** deletion routines (**6.3**).

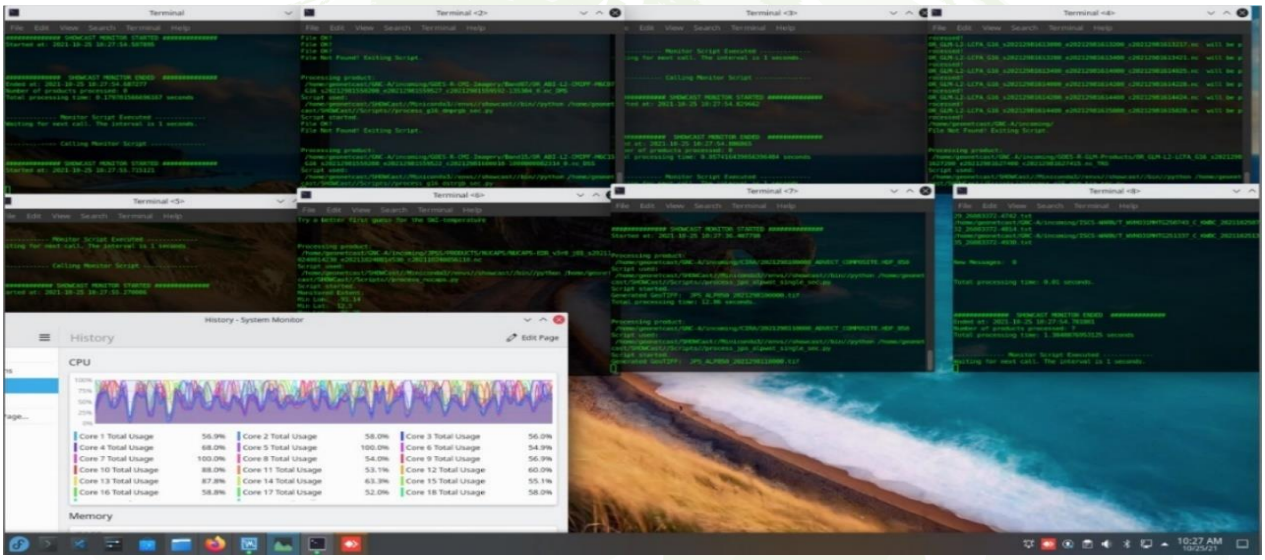


Fig. 26: SHOWCast processing module being executed (credits: William Abarca - MARN El Salvador)

Each product enabled in the **showcast_config.py** (set as **"True"**) will be processed by its parallel process and if the processing is successful for a given product, the thumbnails will start to appear in the SHOWCast main interface and the visualization and animation interfaces will be populated.

Each plot will be saved on the **SHOWCast_v_X_X_X\Output** folder. The plots found on this folder are the historical plots and they are not the plots used by the visualization and animation interfaces. These interfaces use the plots found on the **SHOWCast_v_X_X_X\HTML\Output** folder, which uses plots copied from the **SHOWCast_v_X_X_X\Output** folder.

To summarize, in the **SHOWCast_v_X_X_X\HTML\Output** we have plots from a certain period of time, and on the **SHOWCast_v_X_X_X\Output** folder we have everything.

# 9 THE SHOWCAST IMAGERY AND HTML STRUCTURE

In SHOWCast, we have basically two directories where the plots are stored:

The **Output directory**: Located at **SHOWCast_v_X_X_X/Output/**, this is where all the historical plots are stored. If, for example, you do not want to use the SHOWCast animation interface, and just want to use the plots in your own server, this is the folder to look at.

The sub folders have the following naming convention:

**SSS/PPPPPP_RRR/**

Where:

- **SSS** – Three letter representing the satellite our main category (e.g.: G16).
- **PPPPPP** – Six letter representing the product name (e.g.: BAND13).
- **RRR** – Three letters representing the region (FDK or SEC)

Inside each subfolder, the plots have the following naming convention:

**SSS_PPPPPP_RRR_YYYYMMDDHHMN.webp**

**Where: YYYYMMDDHHMN -** Year, Month, Day, Hour and Minutes

The **HTML directory**: Located at **SHOWCast_v_X_X_X/HTML/Output/**, this is where all the HTML and animation files are stored. The sub folders have the following naming convention:

**SSS/PPPPPP_RRR/**

Where:

- **SSS** – Three letter representing the satellite our main category (e.g.: G16).
- **PPPPPP** – Six letter representing the product name (e.g.: BAND13).
- **RRR** – Three letters representing the region (FDK or SEC)

Inside each subfolder, the plots have the following naming convention:

**SSS_PPPPPP_RRR_N.webp**

Where **"N"** is the animation frame (e.g.: From **"1"** to **"20"**).

## 10 THE SHOWCAST PRODUCT SELECTION INTERFACE

When the processing module is started, the plots will start to appear in the SHOWCast main interface. In the left side of the interface, we have the product categories and on the right side, the thumbnais showing the product “quicklooks”. These quicklooks show the most recent plot a a given product.

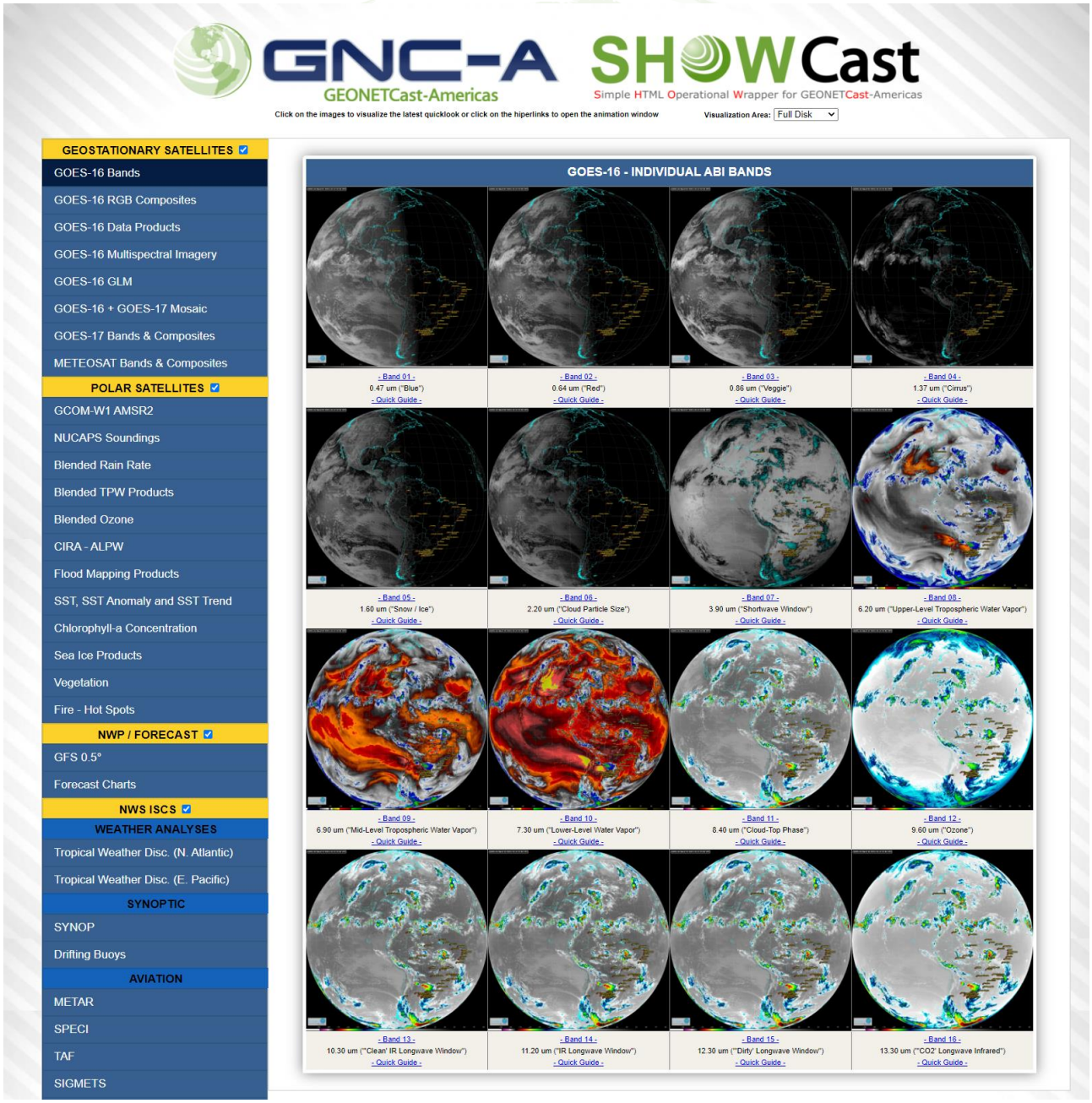



Fig. 27: The SHOWCast main interface showing the processed products (credits: Gustavo Rodriguez - CSPU / Uruguayan Air Force)

## 10.1 Selecting a product category

There are four main product categories in the SHOWCast product selection menu: **GEOSTATIONARY SATELLITES**, **POLAR SATELLITES**, **NWP / FORECAST** and **NWS ICSC** (The US National Weather Service "International Services and Communication Systems"). To facilitate the navigation, you may hide and show each category by clicking at the checkbox near each category title:

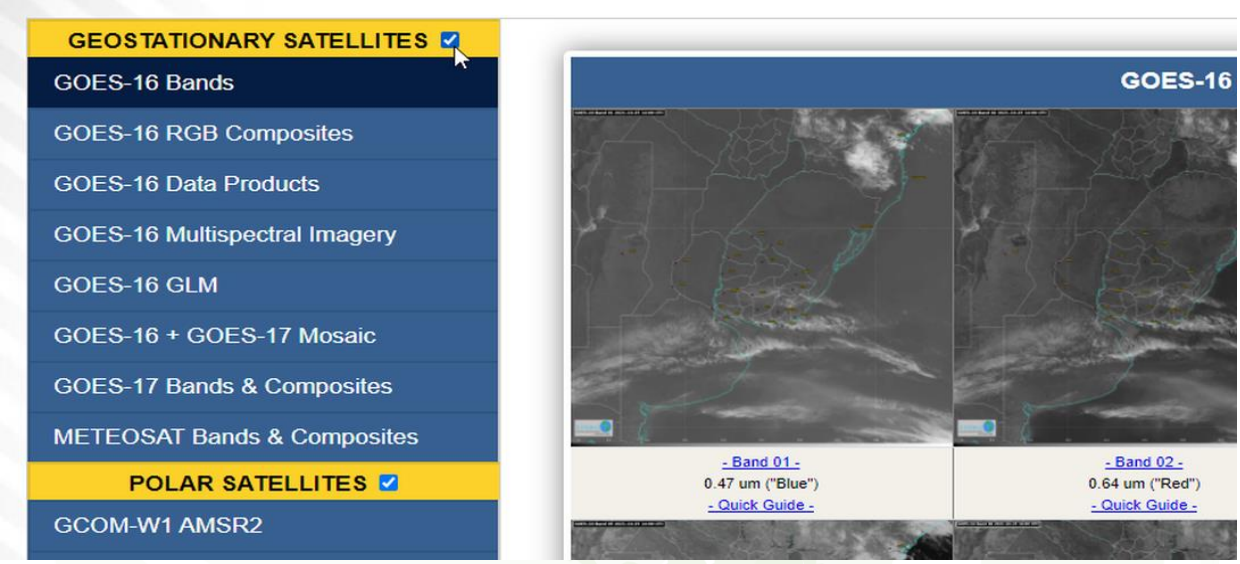



Fig. 28: Hiding / showing product categories to facilitate the navigation

To select a product set, just click at the product name on the left menu. The selected product set box will change to dark blue, and a white arrow will appear. By clicking it, the visualization preview in the right side will be changed, according to the product selection.

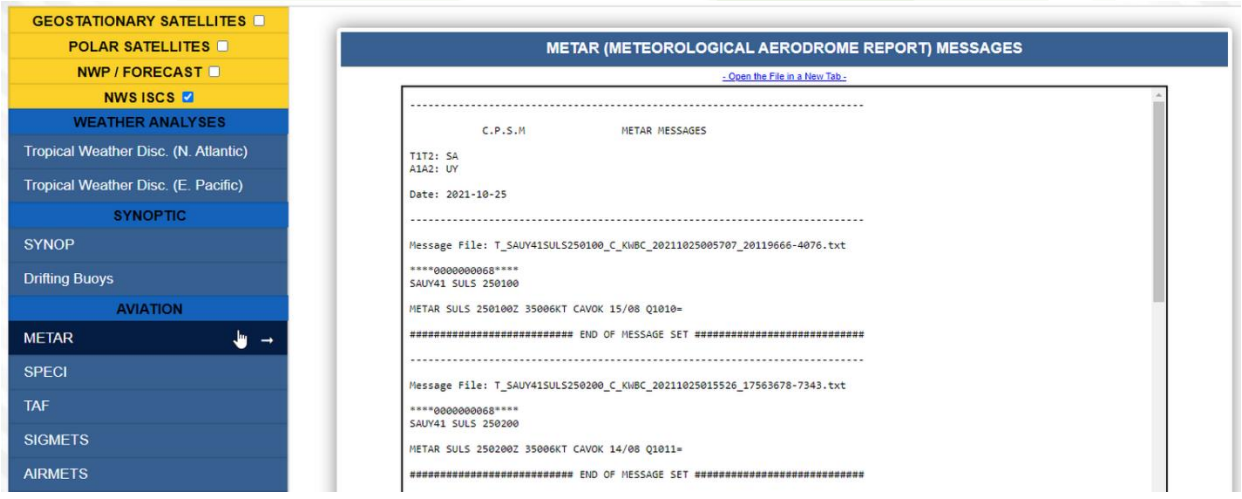



Fig. 29: Selecting a product set in the SHOWCast interface menu

## 10.2 Opening a quicklook

The interface thumbnails show the last plot generated for a particular product. If you click on it, the quicklook will be visualized in a new browser tab.

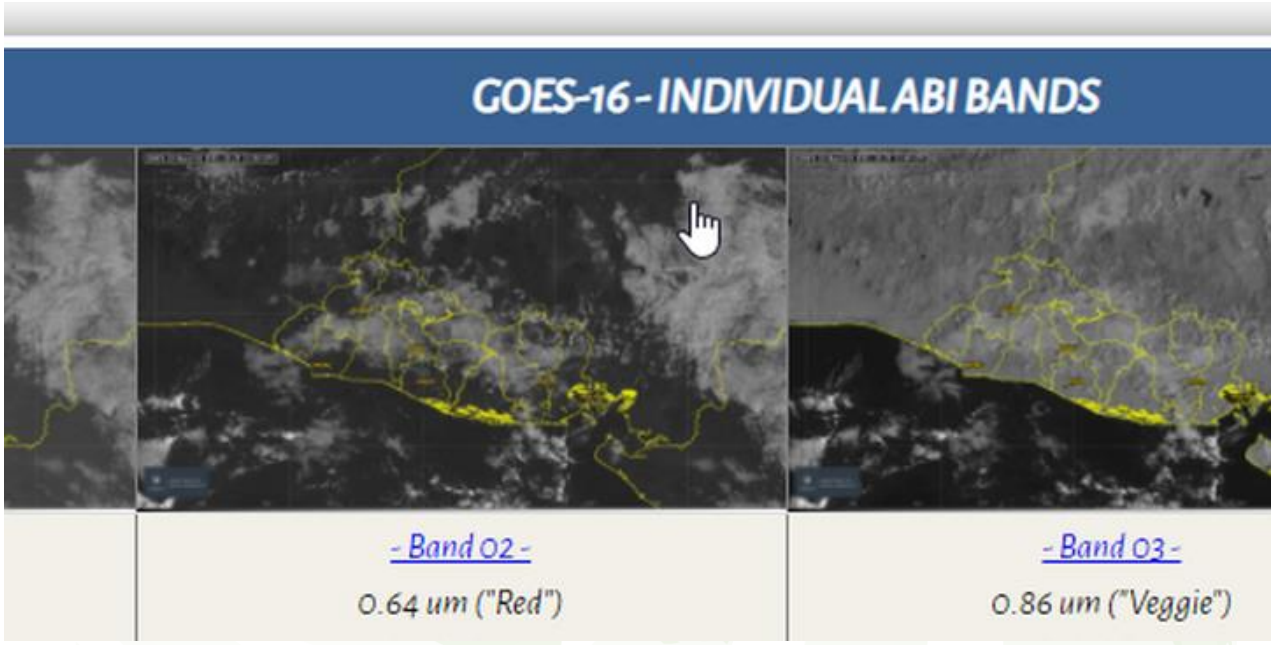



Fig. 30: Opening a quicklook

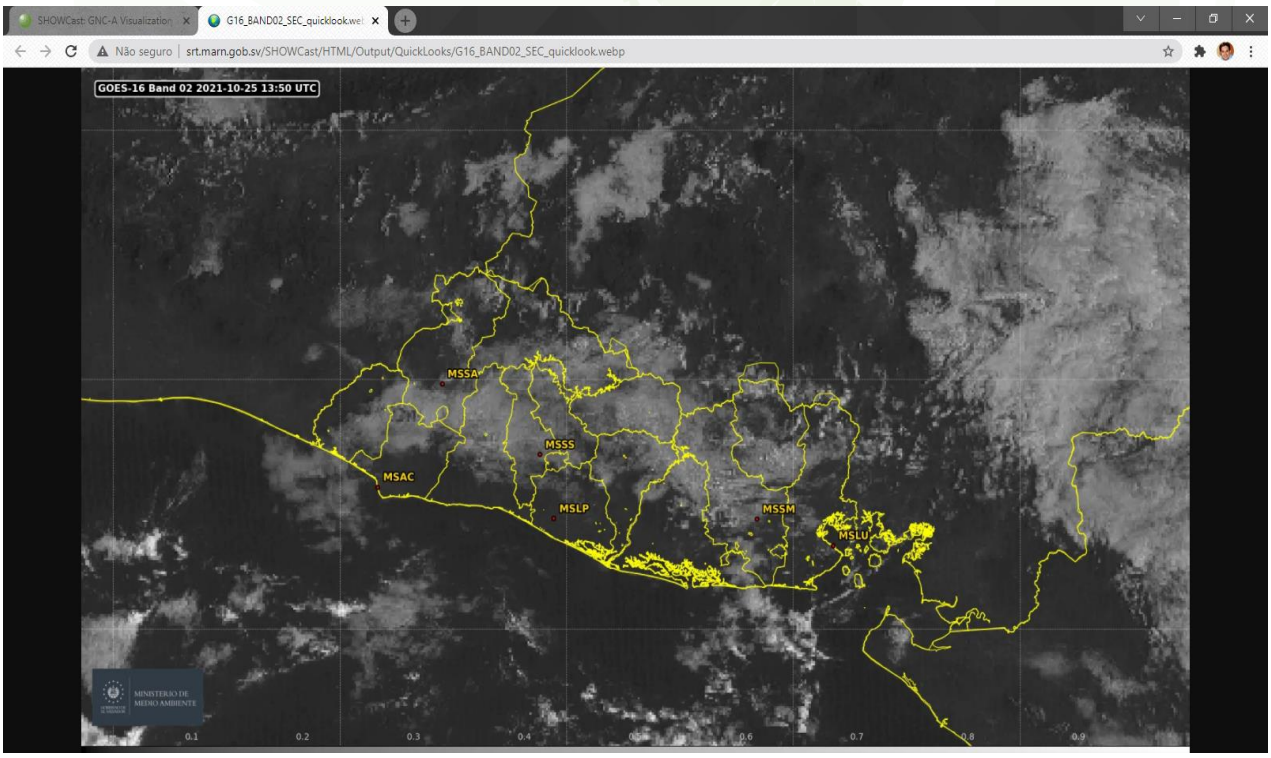



Fig. 31: Quicklook being visualized in a new browser tab

### 10.3 Changing from the "Full Disk" interface to the "User Sector" interface

The combobox in the upper-side part of the interface allows users to change from the SHOWCast's "Full Disk" visualization window to the "User Sector" (products on the cylindrical equidistant projection) visualization window.

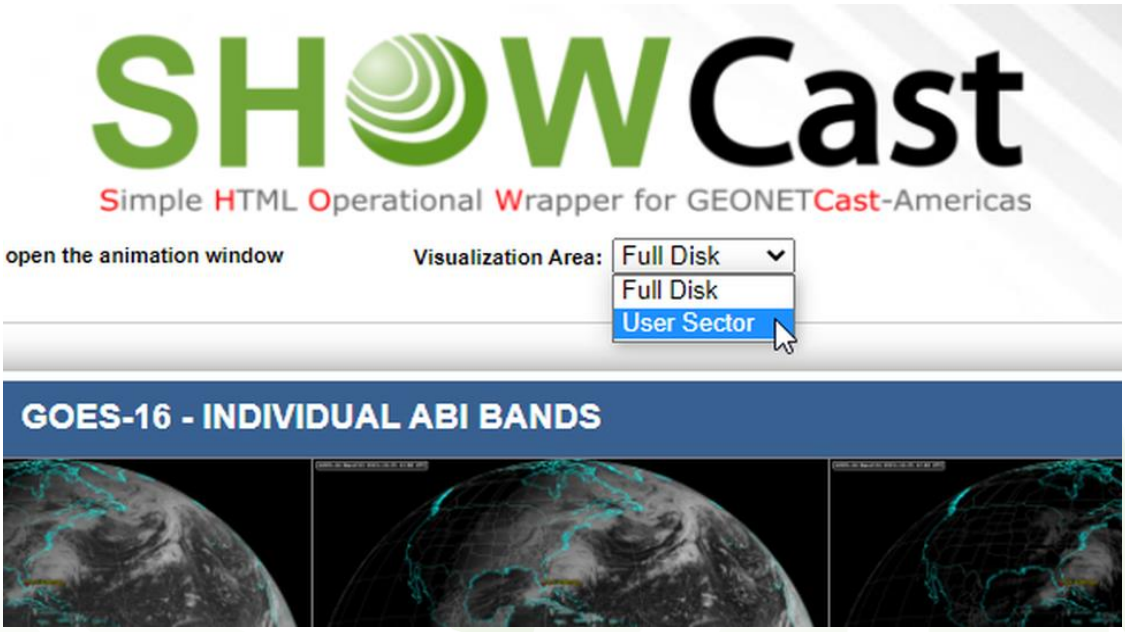



Fig. 32: Changing from the "Full Disk" interface to the "User Sector" interface

When the interface is changed, we see the same product thumbnails, but for the sectorized region, as configured for each product on the **showcast_config.py**. The "Fulls Disks" are available for the products from the **"GEOSTATIONARY SATELLITES"** menu category.

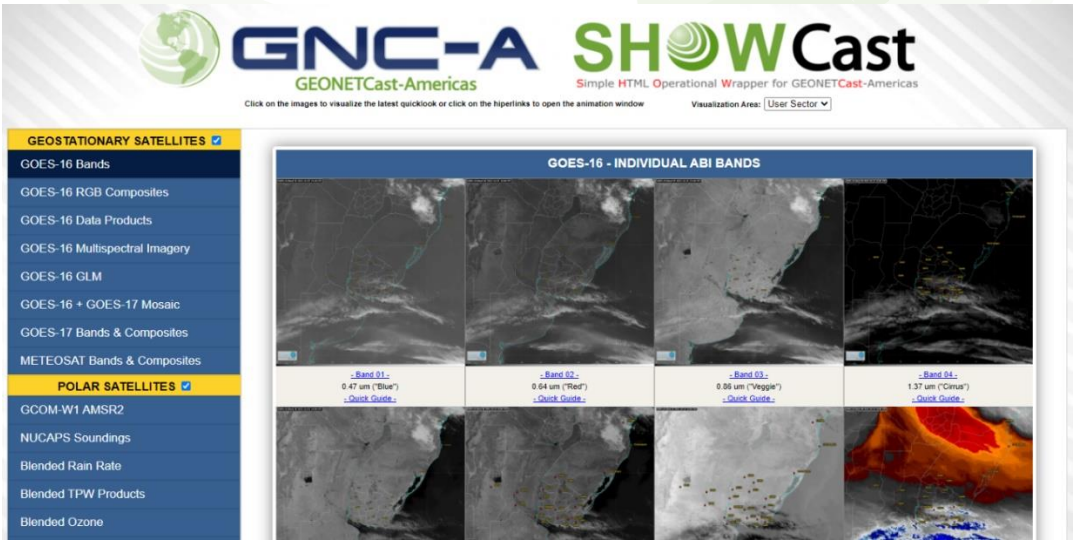



Fig. 33: SHOWCast's "User Sector" interface

## 10.4 Visualizing a Product Quick Guide

For some products, there are **"Quick Guides"** available. These guides are developed by the satellite community (CIRA, NASA SPORT, etc.) and have basic information related to a given product (applications, limitations, importance, interpretation, etc.). To visualize a Quick Guide, click at the **"Quick Guide"** hyperlink below a product quicklook.

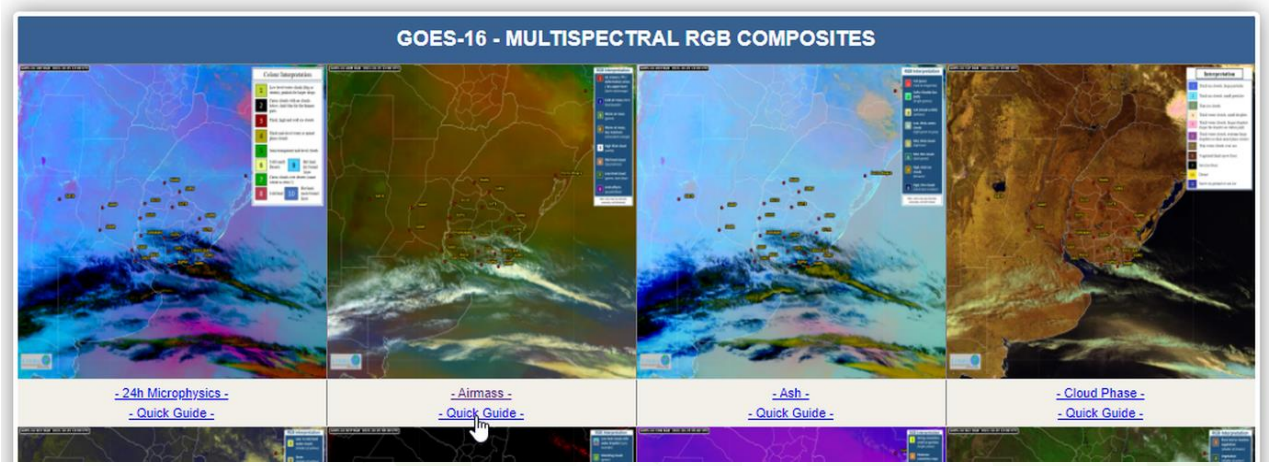



Fig. 34: Opening a product "Quick Guide"

The Quick Guide will be opened in a new browser tab:


Fig. 35: Visualizing a product "Quick Guide"

## 11 THE SHOWCAST ANIMATION INTERFACE

To open the SHOWCast animation window for a given product, just click at the hyperlink right below the product quicklook.

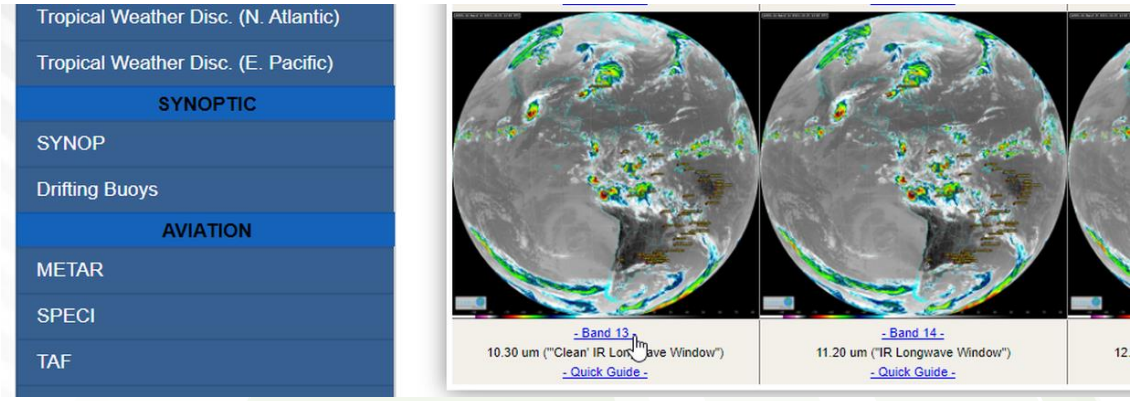



Fig. 36: Opening the SHOWCast animation interface for a given product

The SHOWCast animation window will be opened:

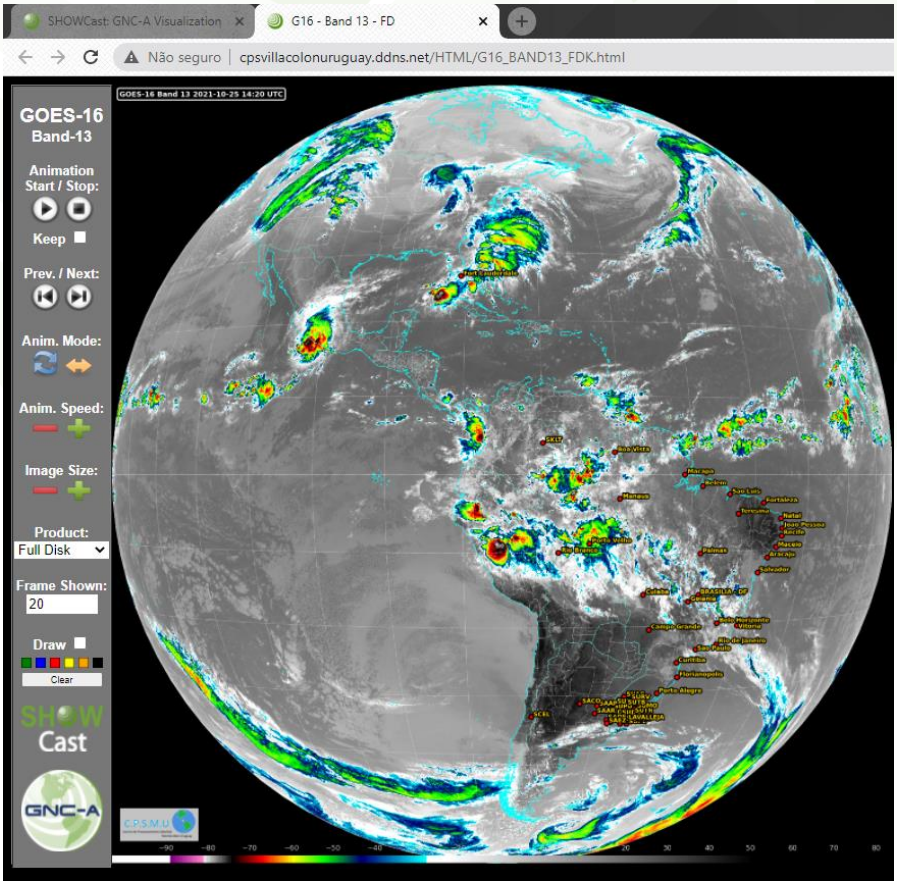



Fig. 37: SHOWCast animation window

## 11.1 Animation interface commands

These are the commands of the SHOWCast animation interface:


Fig. 38: The SHOWCast animation window.

 : Start the animation.

 : Stop the animation.

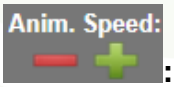
 : Keep animating even if the animation window is refreshed. If you want an animation loop to run, always being updated with the latest imagery available automatically, keep this checked.

 : Previous animation frame.

 : Next animation frame.

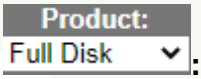
 : Looping Animation Mode. The animated frames go from the first to the last and restart from the beginning.

 : "Boomerang" Animation Mode. The animated frames go from the first to the last and from the last to the first.

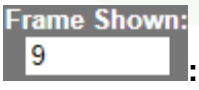


: Decrease or increase the animation speed.

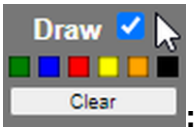
: Zoom out or zoom in.



: Select the product to visualize.



: Frame currently being shown.



: Enable / Disable drawing.

# 12 CUSTOMIZING THE PLOTS

## 12.1 Using your own logo

In order to use your own logo in the plots, simply put your logo (PNG format) inside the **"Logos"** folder in the SHOWCast main directory, and call it **"my_logo.png"**. By default, the logo from **INPE** will be added to the plots.

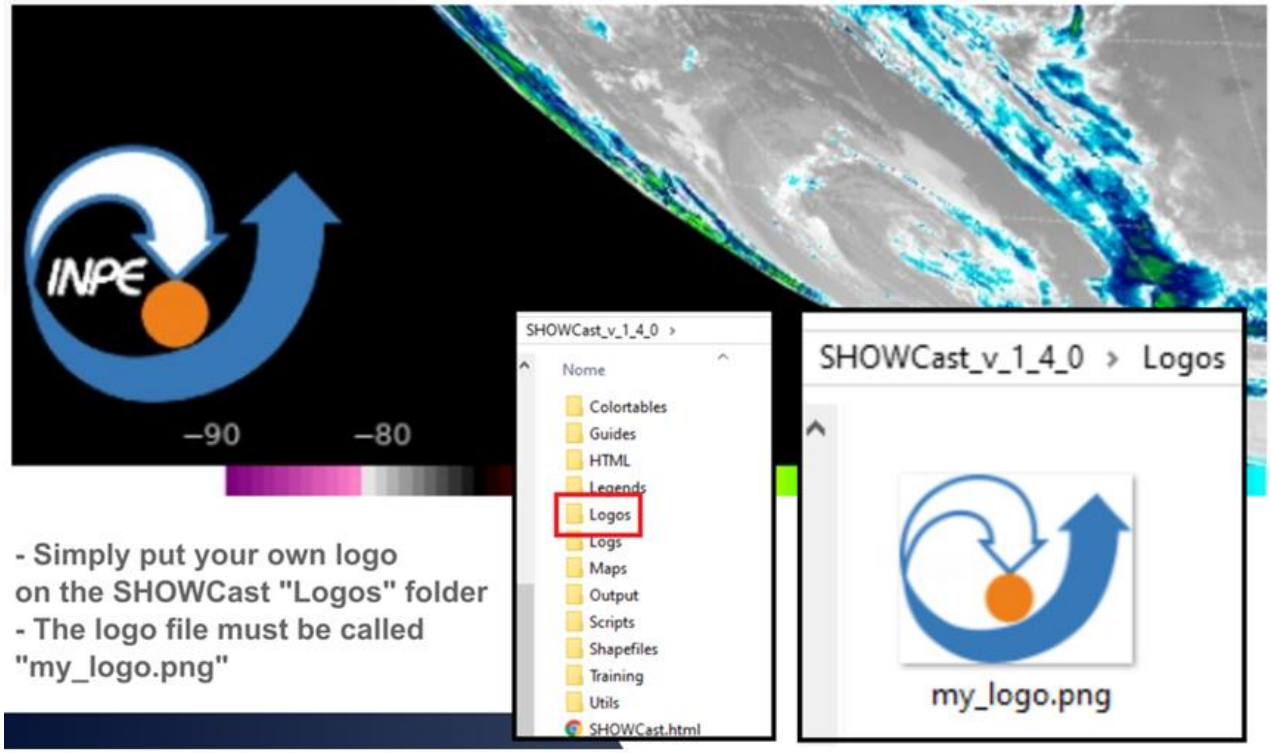



Fig. 39: Adding your own logo to the SHOWCast plots

## 12.2 Using your own labels

Inside the **SHOWCast_v_X_X_X/Utils/Labels** folder there are some **".ini"** files with labels definitions. These files are called:

**labels_example.ini:** Example label configuration file with the Brazilian capitals.
**labels_g16.ini:** Label configuration file for the GOES-16 plots.
**labels_g17.ini:** Label configuration file for the GOES-17 plots.
**labels_gfs_crb.ini:** Label configuration file for the GFS plots (C. America + Caribbean).
**labels_gfs_sam.ini:** Label configuration file for the GFS plots (South America).
**labels_msg.ini:** Label configuration file for the METEOSAT plots.

Inside each **".ini"** file, you have the label definitions. You man change it according to your needs.

**label:** What will be written on this label.
**lon:** Longitude to put this label.
**lat:** Latitude to put this label.
**x_offset:** offset (in lon. degrees) where the label will be shown.
**y_offset:** offset (in lat. degrees) where the label will be shown.
**size:** Label size.
**color:** Label color.
**marker_type:** They way the marker will be represented (https://matplotlib.org/stable/api/markers_api.html).
**marker_color:** The color inside the marker.
**marker_size:** The marker size.

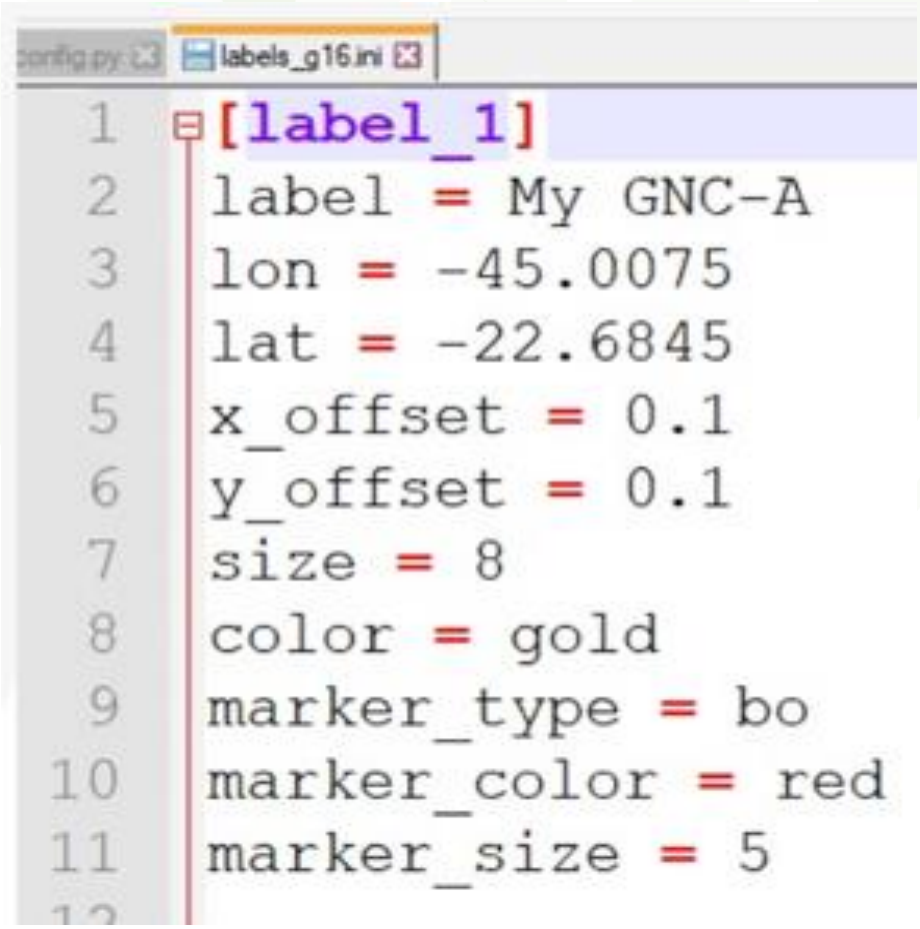



Fig. 40: Label definitions

The **labels_example.ini** is an example label configuration file with the Brazilian capitals.

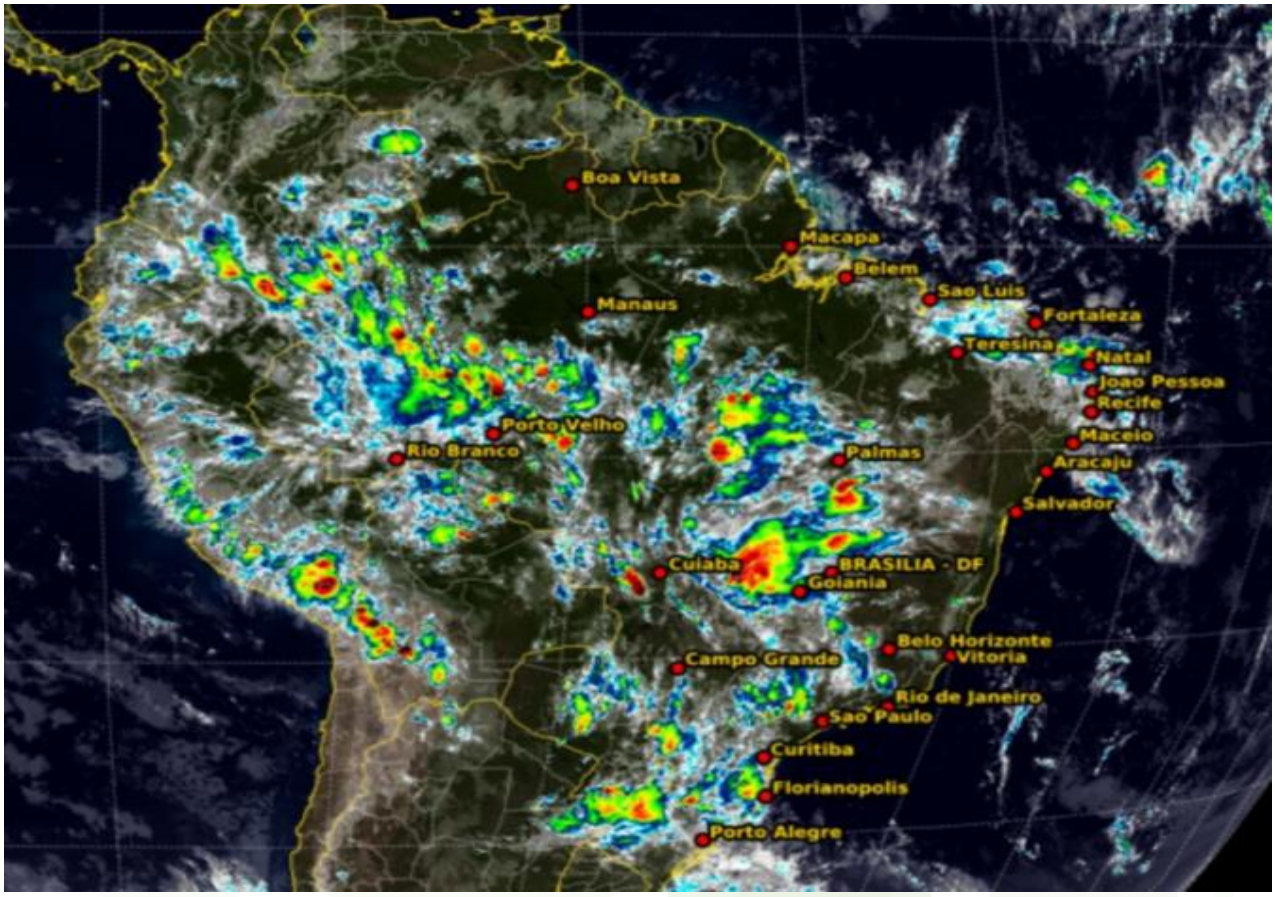



Fig. 41: Plot created with the example label configuration file

## 13 OPTIMIZING SHOWCAST ACCORDING TO THE AVAILABLE HARDWARE

It is possible to adapt SHOWCast to the available hardware, changing the following configurations **for each product** in the **showcast_config.py**:

- Select which products will be processed (**True**) or not (**False**).
- Parallel processing: Select in which process cycle each product will be processed.
- Select which region we want to plot. Smaller the region, less computer power is required.
- Select the plot's final resolution. Lesser the resolution, less computer power is required.
- Select the plot minute interval (for GOES-R products).
- Use multiple servers to produce the imagery.

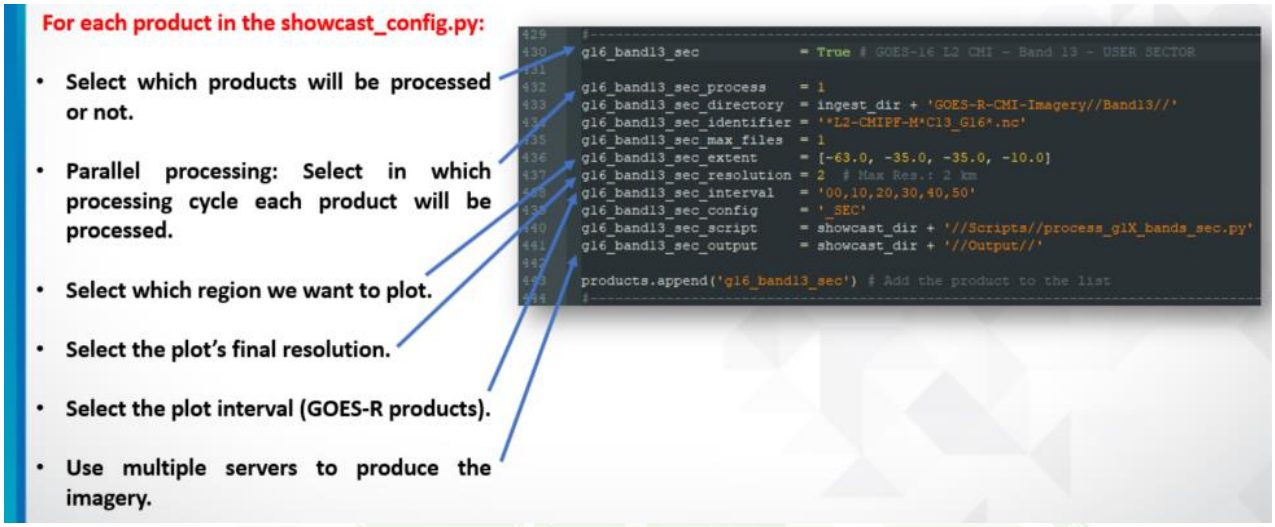



Fig. 42: Optimizing SHOWCast according to the available hardware

## 14 THE SHOWCAST "CLOUD" MODULE

The SHOWCast **"Cloud"** module gives the user the possibility of downloading data from **Amazon (AWS)** or **UNIDATA (THREDDS)**. This is very useful if a user wants to complement their GEONETCast-Americas data (or other receive mechanisms they have), or if they would like to use SHOWCast without having a physical receive station.


Fig. 43: The SHOWCast "Cloud" module

It is important to mention that the data source for Amazon (AWS) is NOAA's PDA (Product Distribution and Access). If PDA is going through a maintenance, it is possible that there's no data available from Amazon. In the other hand, the data source for UNIDATA are different GRB stations, so the data is always available, unless we have a maintanence in the GOES-R satellites.

### 14.1 Configuring the Cloud module

All the files related to the Cloud module are inside the **SHOWCast_v_X_X_X/Cloud/** folder. These are the steps required to use the Cloud module:

**1-) Define where the downloaded data is going to be saved:** In the **SHOWCast_v_X_X_X/Cloud/Scripts/grb_unidata_download_config.py** or **SHOWCast_v_X_X_X/Cloud/Scripts/pda_aws_download_config.py**, configure the variable **"ingest_folder"** with the directory you want to save the data.

```
73  # Main ingest folder - Where your dow
74  ingest_folder = "C://VLAB//Cloud//"
```

Fig. 44: Configuring where we want to store the data

**2-) Define which product you want to download:** In the **SHOWCast_v_X_X_X/Cloud/Scripts/grb_unidata_download_config.py** or **SHOWCast_v_X_X_X/Cloud/Scripts/pda_aws_download_config.py**, configure the products you want to download from Amazon or UNIDATA.

Note: In the current SHOWCast release, from Amazon it's possible to download all GOES-R Bands (L1b or L2), all Level 2 derived products, GLM and SUVI data. From UNIDATA, it is possible to download all GOES-R Bands (L1b only). For both services, the GOES-R imagery and products are available for the CONUS, MESOSCALE and FULL-DISK domains. SHOWCast currently processes FULL-DISKS only. With little modification, it could easily process CONUS and MESOSCALES, but this is not in the scope of this document.

```
#-----------------------------------------------------------------------------------------------
# ABI L2 BANDS
#-----------------------------------------------------------------------------------------------

# ABI L2 Cloud and Moisture Imagery - CONUS
ABI_L2_CMIPC          = False
ABI_L2_CMIPC_Product  = 'ABI-L2-CMIPC'
ABI_L2_CMIPC_Channel  = ['C01', 'C02', 'C03', 'C04', 'C05', 'C06', 'C07', 'C08', 'C09', 'C10', 'C11', 'C12', 'C13', 'C14', 'C15', 'C16']
ABI_L2_CMIPC_Minutes  = ['01', '06', '11', '16', '21', '26', '31', '36', '41', '46', '51', '56']
ABI_L2_CMIPC_Folders  = 'GOES-R-CMIPC-Imagery//'

# ABI L2 Cloud and Moisture Imagery - FULL DISK
ABI_L2_CMIPF          = True
ABI_L2_CMIPF_Product  = 'ABI-L2-CMIPF'
ABI_L2_CMIPF_Channel  = ['C01', 'C03', 'C04', 'C05', 'C06', 'C07', 'C08', 'C09', 'C10', 'C11', 'C12', 'C13', 'C14', 'C15', 'C16']
ABI_L2_CMIPF_Minutes  = ['00', '10', '20', '30', '40', '50']
ABI_L2_CMIPF_Folders  = 'GOES-R-CMI-Imagery//'

# ABI L2 Cloud and Moisture Imagery - MESOSCALE
ABI_L2_CMIPM          = False
ABI_L2_CMIPM_Product  = 'ABI-L2-CMIPM'
ABI_L2_CMIPM_Channel  = ['C01', 'C02', 'C03', 'C04', 'C05', 'C06', 'C07', 'C08', 'C09', 'C10', 'C11', 'C12', 'C13', 'C14', 'C15', 'C16']
ABI_L2_CMIPM_Mesoscl  = ['M1', 'M2']
ABI_L2_CMIPM_Minutes  = ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21',
 '22', '23', '24', '25', '26', '27', '28', '29', '30', '31', '32', '33', '34', '35', '36', '37', '38', '39', '40', '41', '42', '43', '44', '45', '46'
, '47', '48', '49', '50', '51', '52', '53', '54', '55', '56', '57', '58', '59']
ABI_L2_CMIPM_Folders  = 'GOES-R-CMIPM-Imagery//'
```



Fig. 45: Configuring which products we want to download

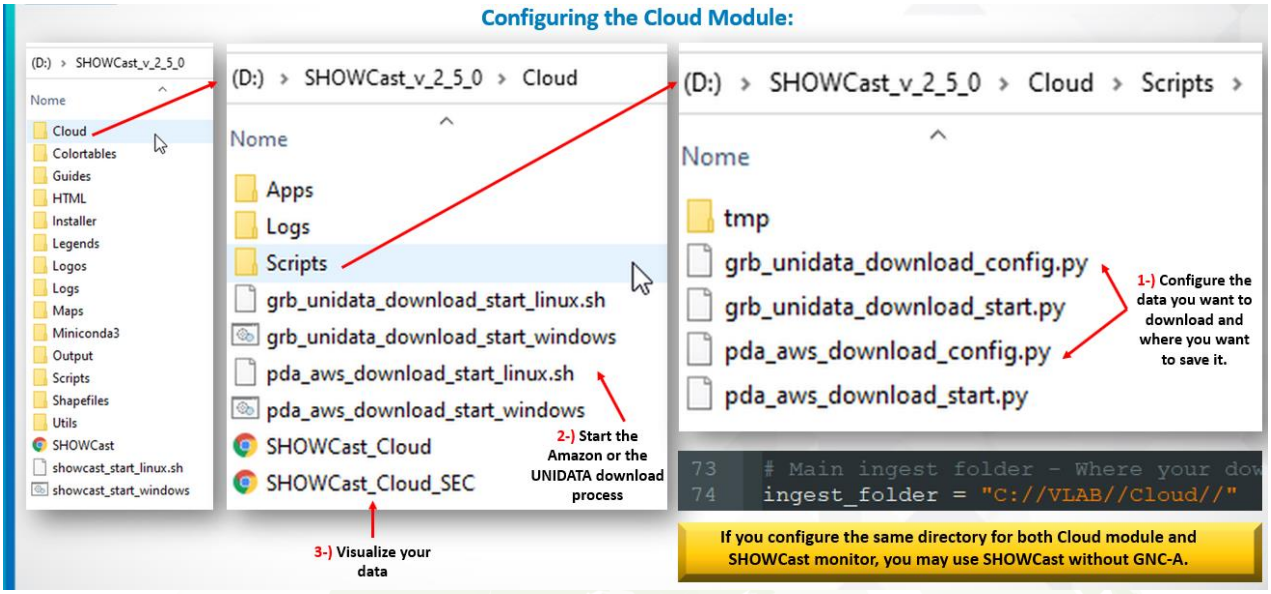



Fig. 46: Configuring the Cloud module (summary)

## 14.2 Starting the Cloud module

After configuring which data we want to download and where to store the data, in order to run the Cloud module, if you are using Windows, just double click the "pda_aws_download_start_windows.bat" or "grb_unidata_download_start_windows.bat", and if you ware using Linux, execute the "pda_aws_download_start_windows.sh" script or the "grb_unidata_download_start_windows.sh". The Cloud module terminal will be open.

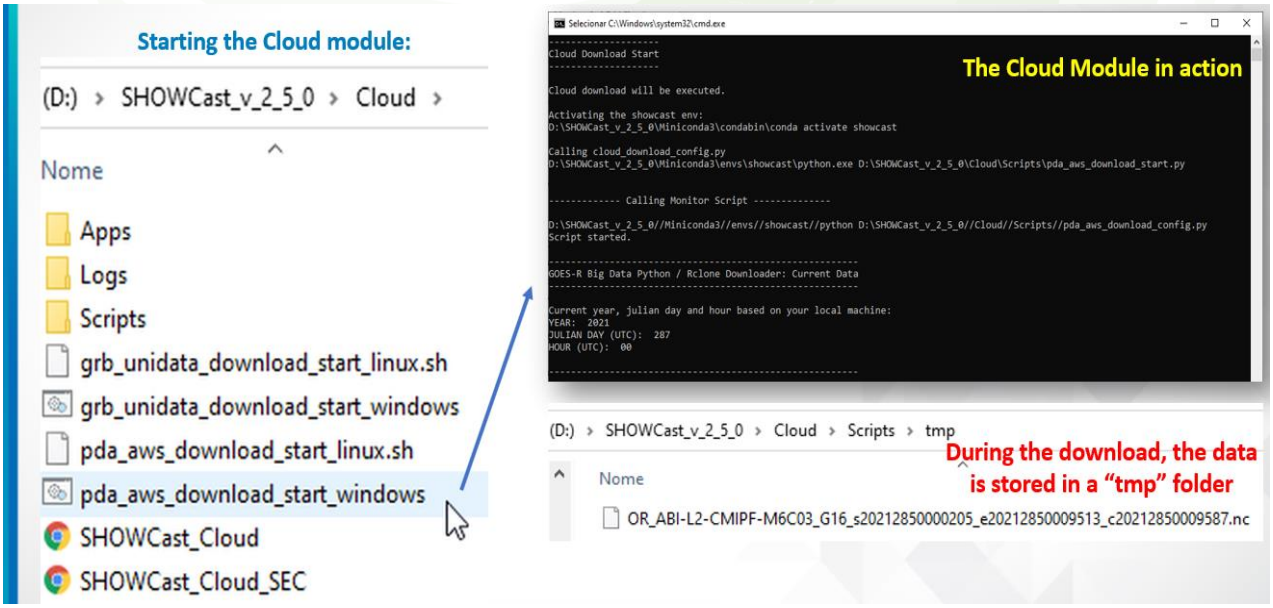



Fig. 47: Starting the Cloud module (summary)

**During** the download, the data is stored in a **"tmp"** folder, located at:

**SHOWCast_v_X_X_X/Cloud/Scripts/tmp/**

When the download is finished, the files are moved to the directory configured, and the files may be processed and visualized.

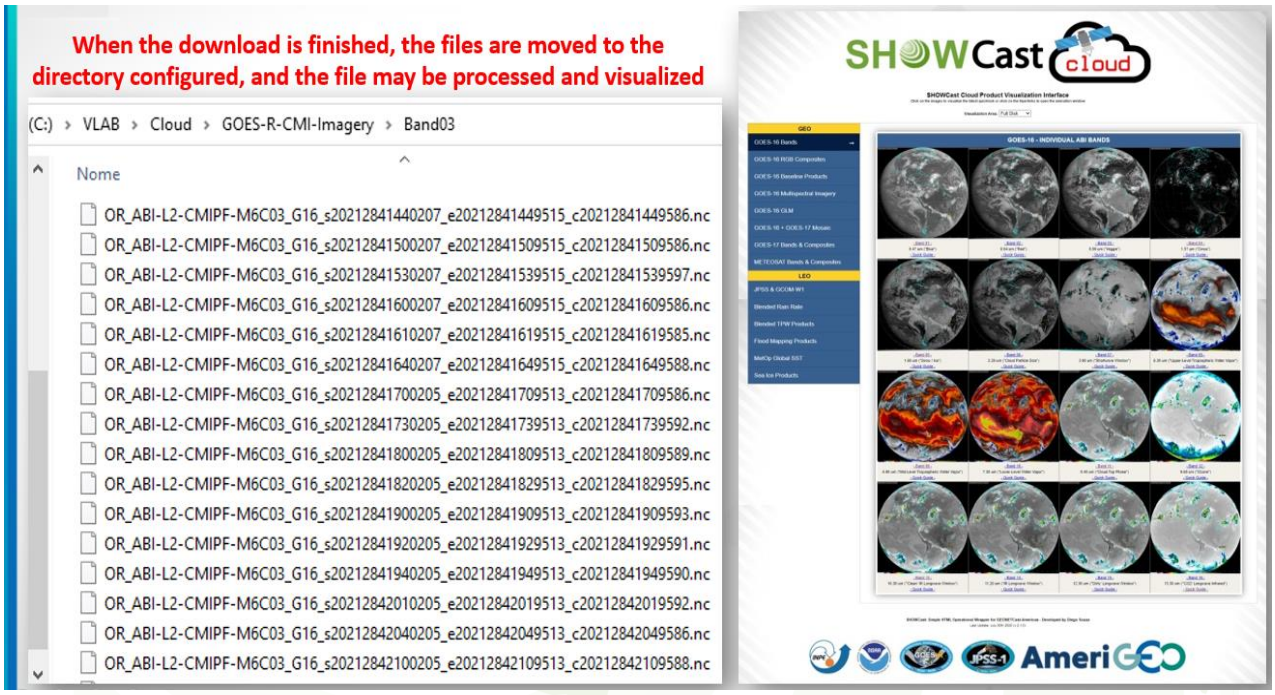



Fig. 48: Downloading and visualizing data with the SHOWCast Cloud module

**Note:** If you configure the same directory for both the Cloud module (**"ingest_folder"** variable) and the SHOWCast processing module (**"ingest_dir"** variable in the **"showcast_start.py"** script), you may use SHOWCast without having a receive station like GNC-A.

## 15 SHOWCAST RELEASE HISTORY

The following SHOWCast versions have been release until the time this document was written:

- **V 1.0 (Nov 06 2019):** Initial version (57 products)
- **V 1.1 (Nov 14 2019):** New products
- **V 1.2 (Nov 27 2019):** Region and resolution configuration
- **V 1.3 (Jan 20 2020):** Logos, Labels, Annotations and Legends
- **V 1.4 (Feb 04 2020):** New products

- **V 2.0 (Jul 8 2020):** New interface, Easier installation, New Products
- **V 2.1 (Jul 30 2020):** “Cloud” module
- **V 2.2 (Nov 23 2020):** Parallel processing / New Products (121 products)
- **V 2.3 (Mar 23 2021):** 20 s GLM, GFS, ISCS, Ozone (140+ products)
- **V 2.4 (Jul 1 2021):** ALPW, SST, SST-A, SST-T, OC, New Features (150+ products)
- **V 2.5 (Oct 8 2021):** 16 Bands, RGB’s, WebP, Interval

## 16 SHOWCAST USER EXAMPLES

We have basically three kinds of SHOWCast users:

- Users who want to use **both** processing and visualization interfaces, adapting it according to their needs:

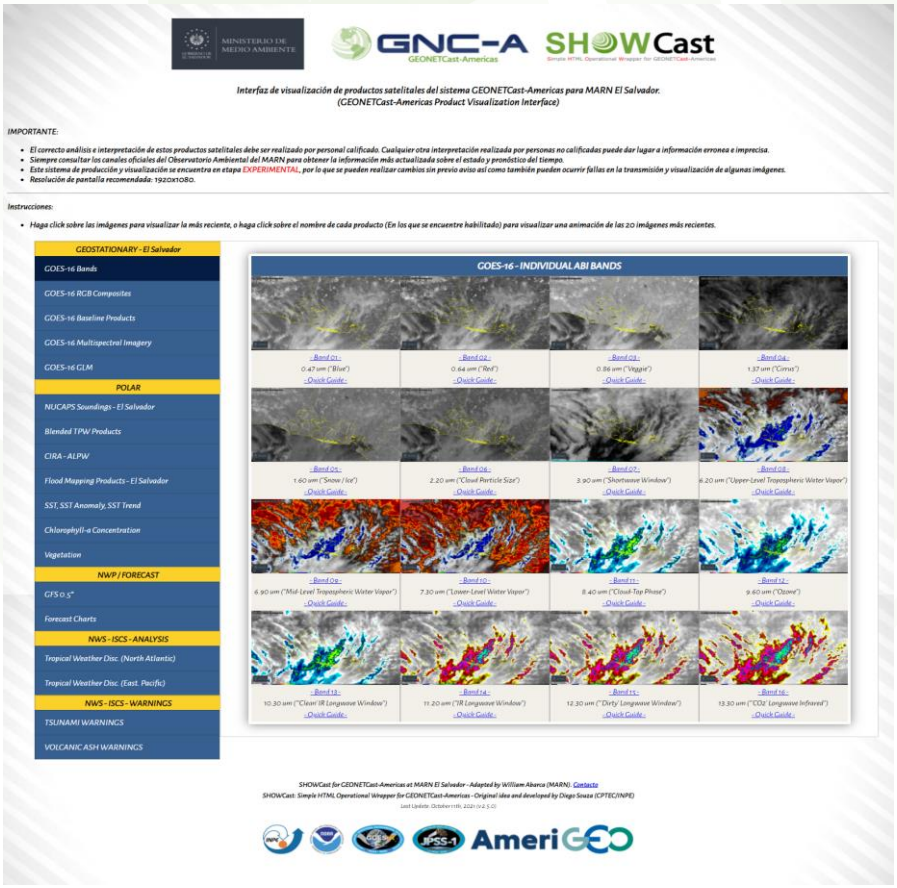



Fig. 49: Using both SHOWCast processing and visualization interfaces (credits: William Abarca [MARN El Salvador])

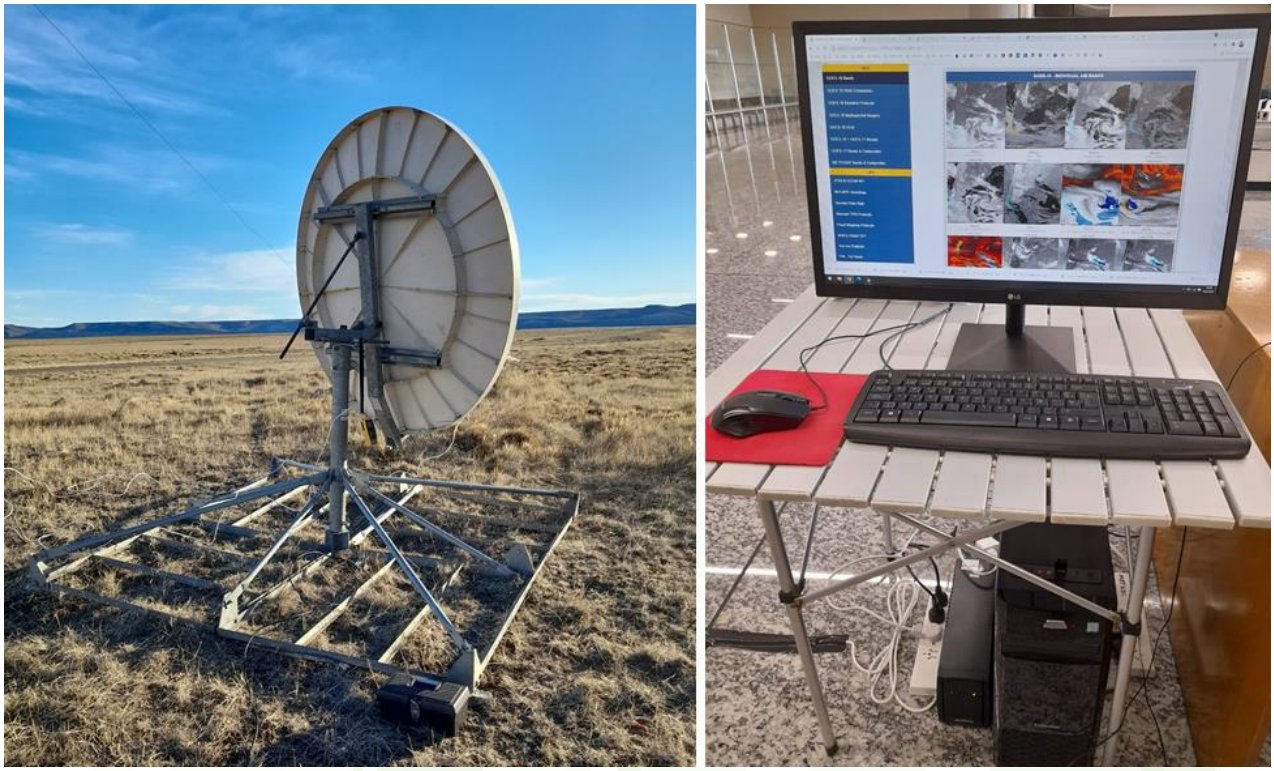


Fig. 50: Using both SHOWCast processing and visualization interfaces (credits: Ricardo Valenti [Argentine Air Force])

Fig. 51: Using both SHOWCast processing and visualization interfaces (credits: Gustavo Rodriguez [Uruguayan Air Force])

- Users who already have means for visualization and just would like to use the script examples found at **SHOWCast_v_X_X_X/Scripts**:

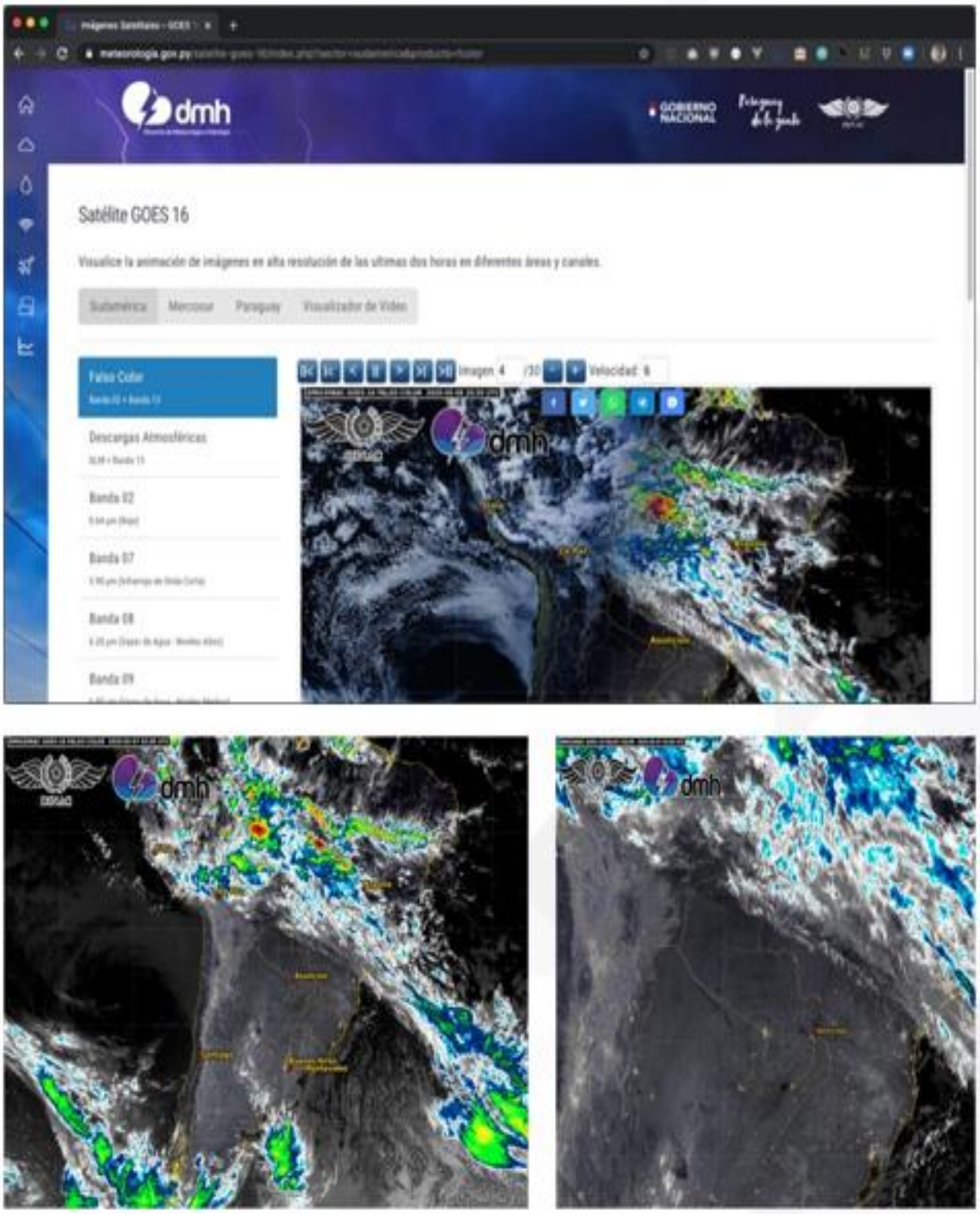



Fig. 52: Using only the SHOWCast example scripts (credits: Ever Barreto [Asunsíon Catholic University] / Wilson Caballero [DINAC] - Paraguay)

- Users who create big SHOWCast adaptations, like the **"SHOWCast.GR"** solution created by **HNMS - Hellenic National Meteorological Service (Greece)**:

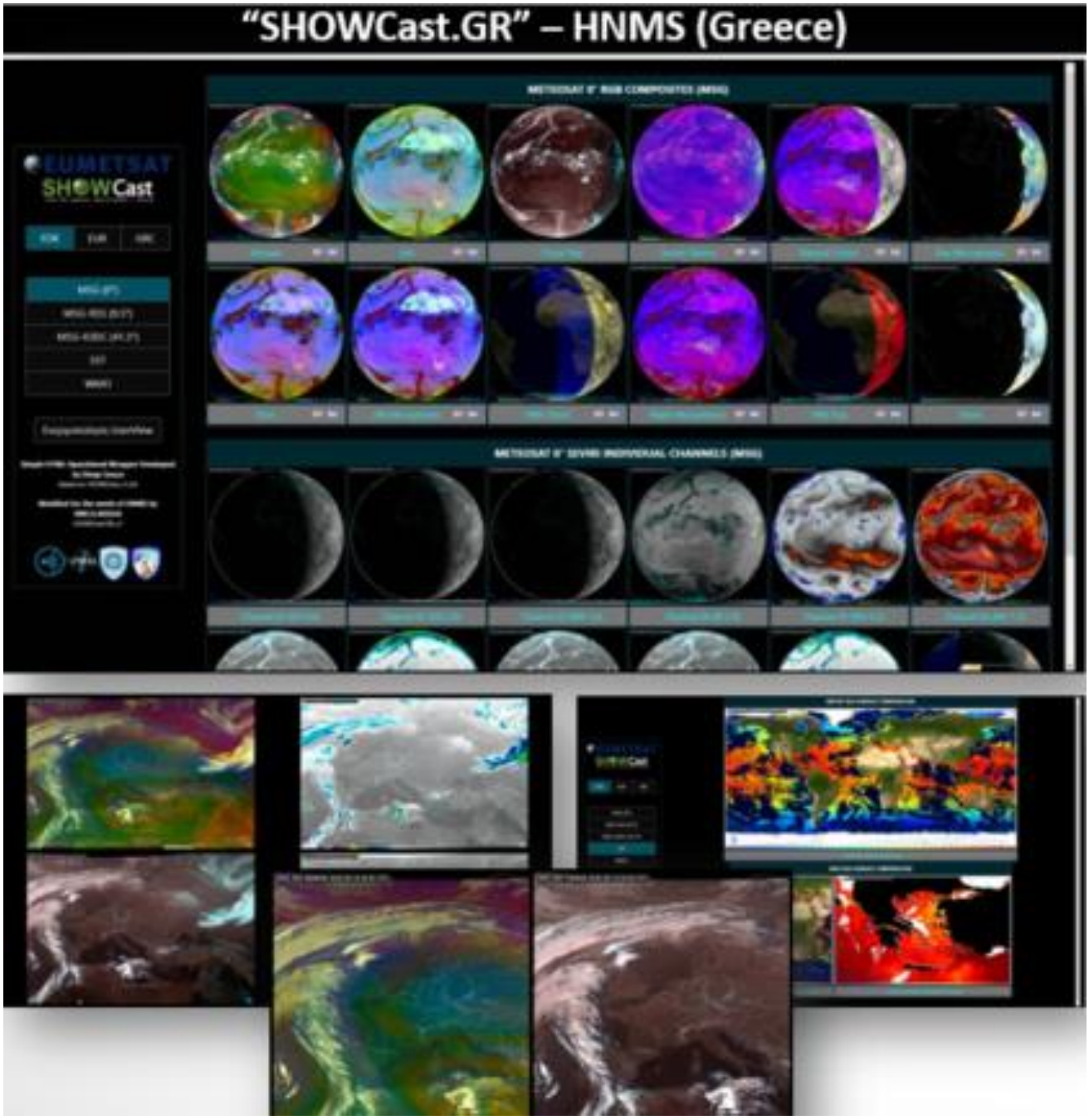



Fig. 53: "SHOWCast.GR" (credits: Dimitrious Papanastasiou [HNMS - Greece])

In this example, SHOWCast was adapted to process the content from the EUMETCast-Europe service, like 15-minute METEOSAT Second Generation data, among other new features like a multi panel visualization interface.

Or the INMET SEPIS solution created by **INMET (Brazilian National Weather Service):**

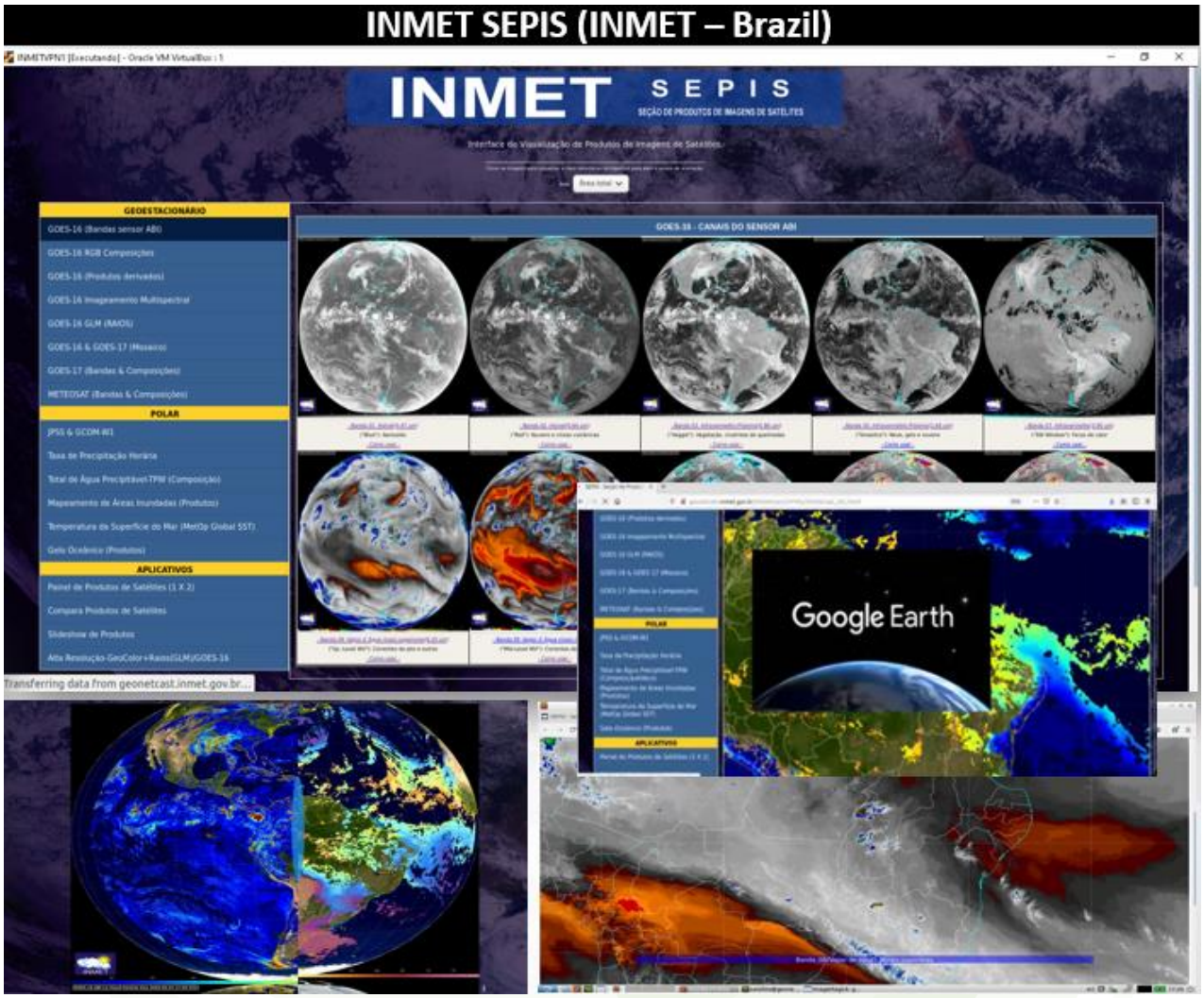



Fig. 54: "INMET SEPIS" (credits: Kleber Ataíde [INMET - Brazil])

In this example, SHOWCast was adapted and new features have been created, like a product comparison interface, a slide show, or exporting to Google Earth.

SHOWCast has also been used to teach students (satellite data access and processing).

Fig. 55: Using SHOWCast in the classroom (credits: Demilson Quintão [IPMET - Brazil])

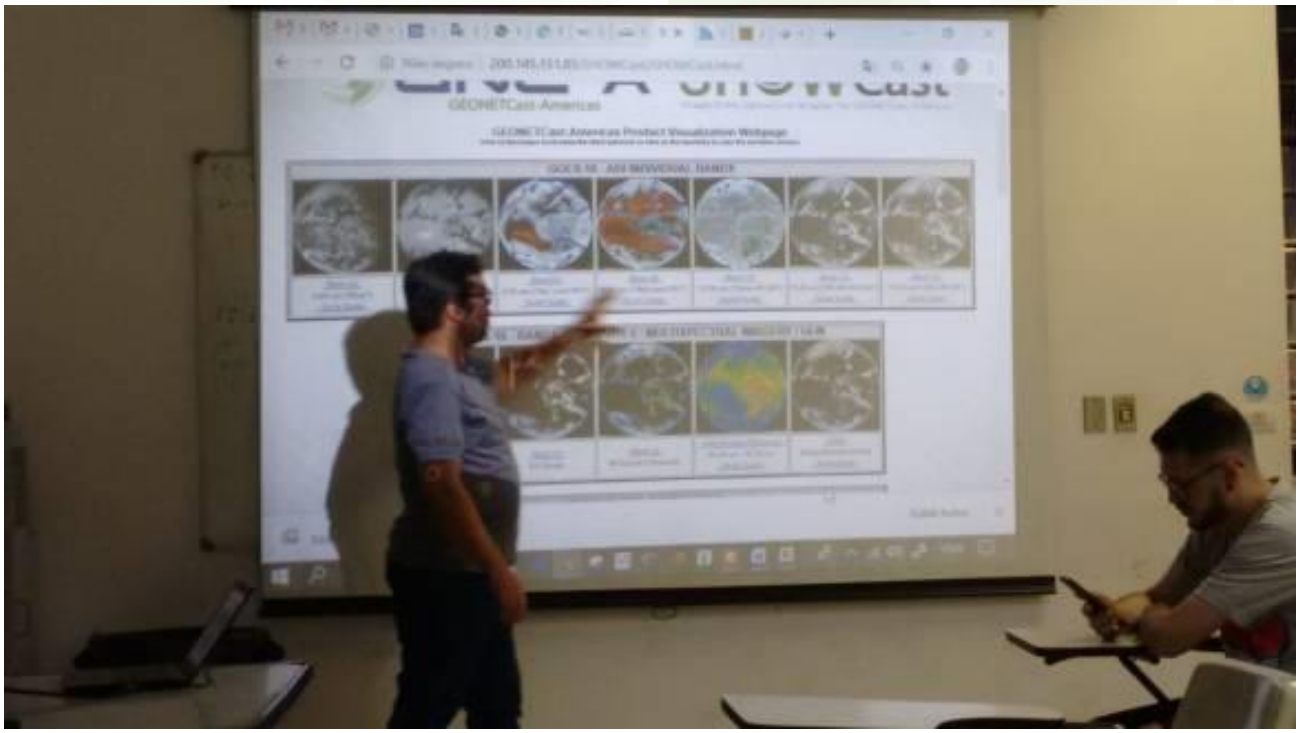


Fig. 56: Using SHOWCast in the classroom (credits: Demilson Quintão [IPMET - Brazil])

## 17 ADVANCED CONFIGURATION VIA SCRIPTS

Some SHOWCast configuration aspects are still available only via scripts. Making these changes easily configurable is on the plans for future releases.

These are the parameters:

- **Number of animation frames:** The default number of frames for the animations are **20** for most products (30 for GFS and 60 for NUCAPS Soundings). You may change the number of frames in the scripts found at SHOWCast_v_X_X_X/Scripts/, by changing the parameter **"nfiles"**.

```
# Update the animation
nfiles = 20
update(satellite, product, nfiles, sys.argv[7], sys.argv[8])
```

Fig. 57: Changing the number of animation frames via scripts

**Note:** If you change the **"nfiles"** parameter in the scripts, it is also necessary to change the **"last_image"** variable in the correspondent **HTML** file for that particular product. The **"nfiles"** from the Python script and te **"last_image"** must be equal.

```
image_name = "Output//G16//BAND13_FDK//";
image_type = "webp";
channel = "G16_BAND13_FDK_";
first_image = 1;
last_image = 20;
```

Fig. 58: Changing the number of animation frames in the HTML files

- **GFS plot interval configuration:** In order to configure the GFS plots start hour, end hour and interval, it is necessary to edit the GFS python scripts found at SHOWCast_v_X_X_X/Scripts/. You need to configure the following variables: "hour_ini", "hour_end" and "hour_inc". In the example below, plots will be created between 0 and 120 hours, with a 3-hour interval, totalizing 40 plots per GFS run.

```
# Data you want to process
# (to process only the analisys, end and inc should be equal).
hour_ini = 0   # Init time
hour_end = 3   # End time
hour_inc = 120 # Increment
```

Fig. 59: Configuring the GFS plot interval

## 18 CONCLUSION

SHOWCast is a simple yet powerful data processing and visualization package developed mainly using Python.

It works with GEONETCast-Americas, GOES-R GRB, Amazon AWS, UNIDATA THREDDS and other satellite data reception mechanisms ingesting the same files it's configured to process.

As seen on this manual, it is relatively easy to get started and it may be used as a starting point for your own processing scheme. If your institution does not have a processing mechanism or visualization interfaces, you may use SHOWCast and adapt it according to your needs. If you already have a visualization interface available (e.g.: your own webpage), you may use only the plots that are generated routinely (available at the "Output" folder) or use the example scripts as a reference. If you are an experienced programmer or web developer, you may make it bigger and better!

SHOWCast provides easy data access for those who doesn't have a receive station through the "Cloud" module and it also can be used to teach students on satellite data access and processing (it currently has +120 Python example scripts).

New features and optimizations are being added over time based on user suggestions, so do not hesitate to contact me!

diego.souza@inpe.br

I'm very happy with the knowledge gained, lessons learned and friends made throughout the SHOWCast development process.

## 19 ACKNOWLEDGEMENTS

I would like to thank some colleagues that contributed during the development of SHOWCast:

**William Abarca (MARN - El Salvador):** Helped with scripting adjustments, bug corrections, troubleshooting, testing different hardware setups, optimizations and suggestions.

**Gustavo Rodriguez (FAU - Uruguay):** For patiently testing new releases, helping with bug corrections and making great suggestions.

**Ricardo Valenti (FAA - Argentina):** For helping understanding the ISCS messages and the processing development, also testing SHOWCast with an awesome portable setup and making great suggestions.

**Demilson Quintão (IPMET - Brazil), Henry Ramírez (FAP - Peru), Kleber Ataíde (INMET - Brazil):** For testing new releases and making great suggestions.

**Diego Enoré (INPE - Brazil):** For helping in the development of the (hyper complicated) NUCAPS Sounding scripts.

**Ester Regina Ito (INPE - Brazil):** For the great help in the development of the GFS plots.

**Juan José Amides Figueroa (MARN - El Salvador):** For helping in the development of the GFS plots (color palletes suggestions and GDI processing example).

**José Galvez (NOAA):** Troubleshooting the GDI processing scripts.

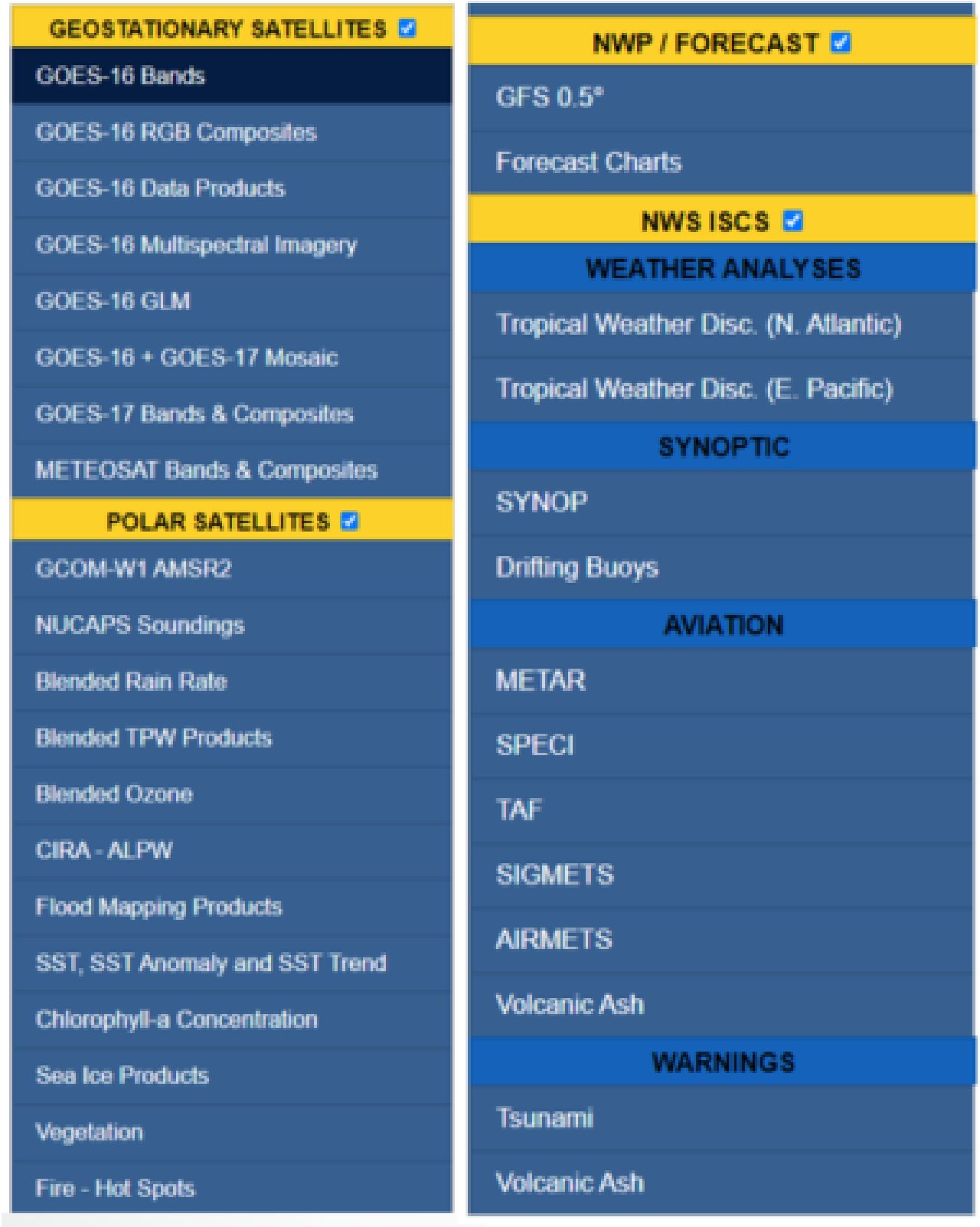
**Douglas Uba, Renato Galante, Rogerio Batista (INPE - Brazil) and Marcial Garbanzo (UCR - Costa Rica):** Great Python advices from these experts!

**Seth Clevenstine (NOAA) [2021], Natalia Donoho (NOAA) [2018-2020], Hongming Qi [2017], and Paul Seymour (NOAA) [2008-2017]:** GEONETCast-Americas Broadcast Managers.

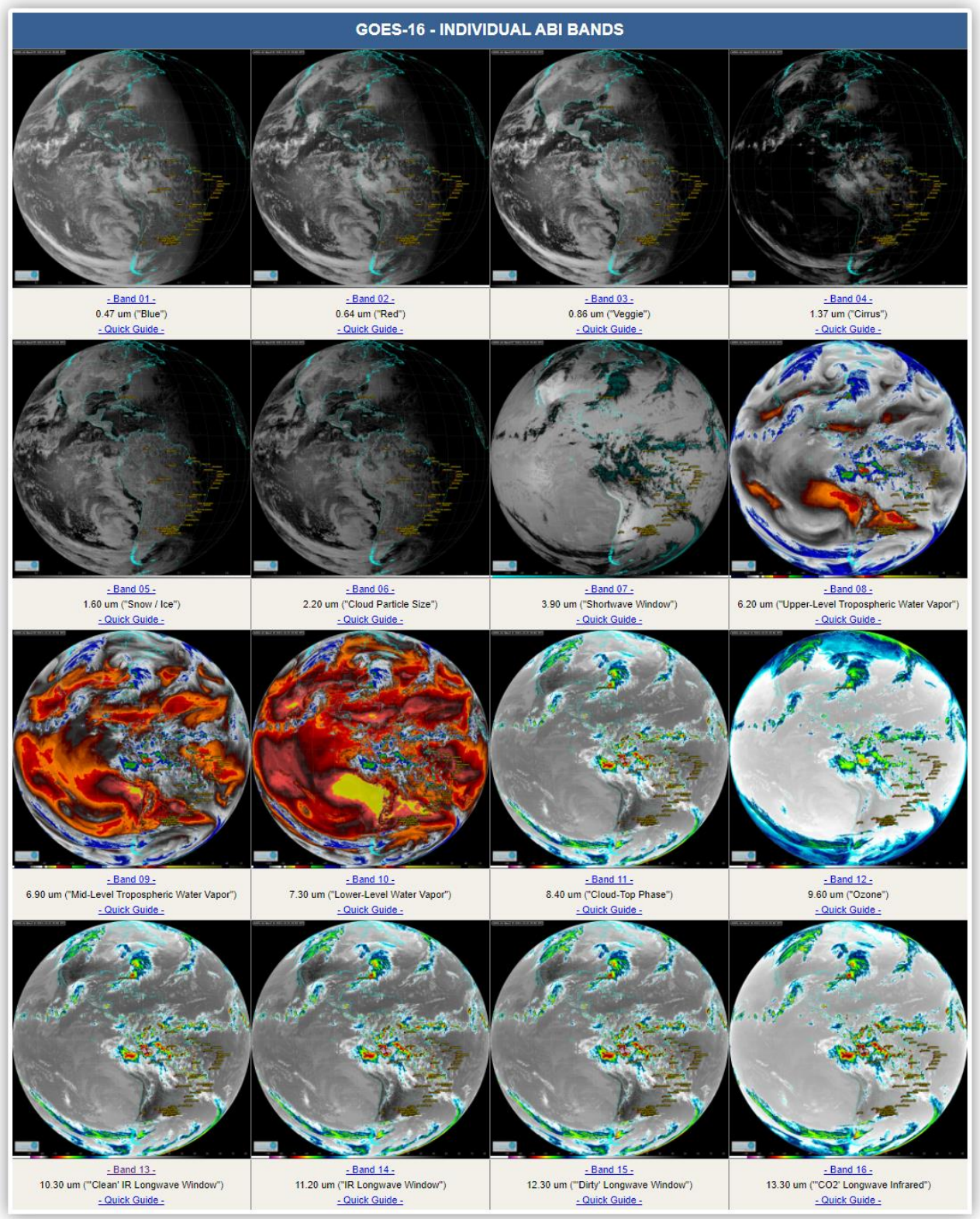
**Eric Madsen (NOAA) - NESDIS International and Interagency Affairs:** For the great contributions to the GNC-A community and the INPE / NOAA Cooperation.

# 20 APPENDIX I: VISUALIZED PRODUCTS

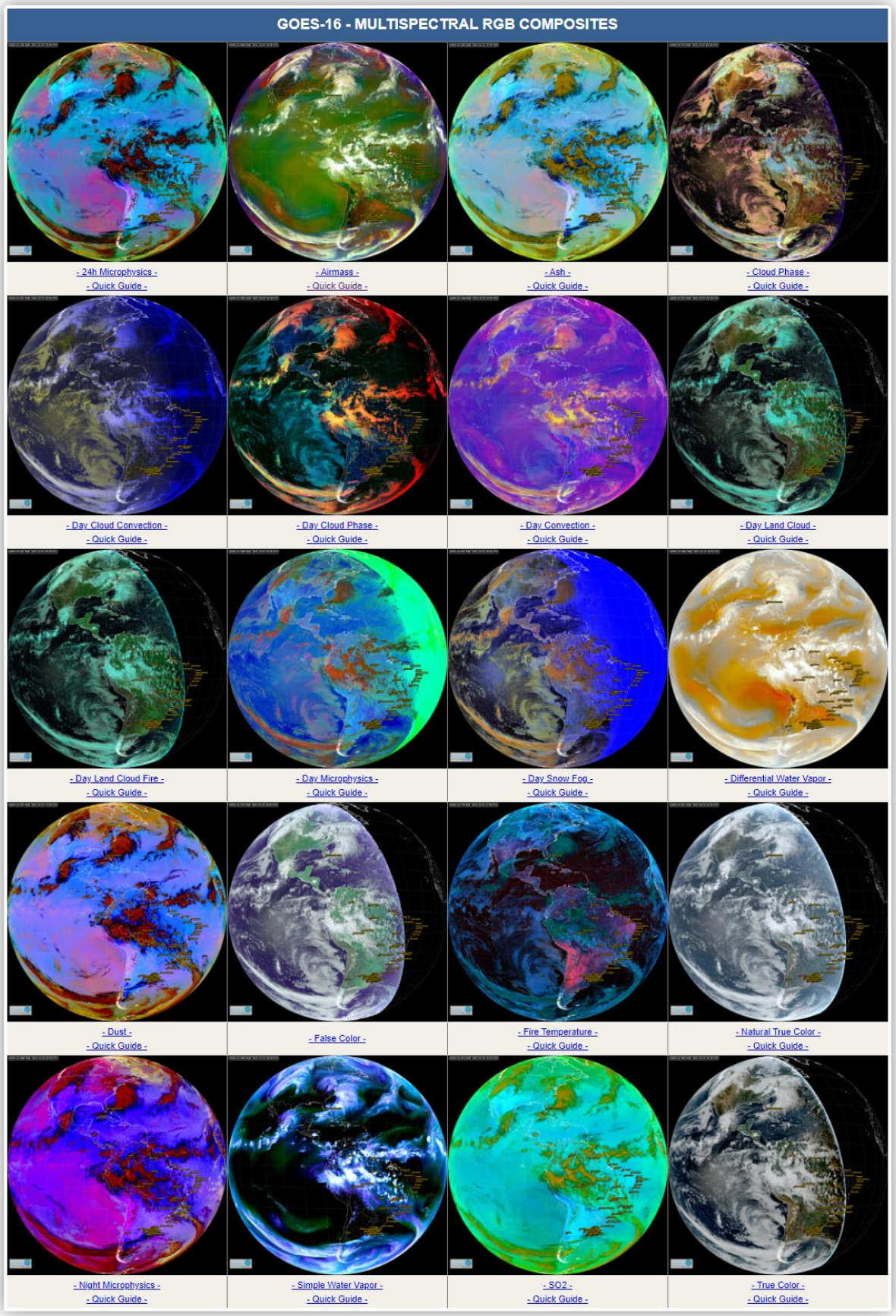
The following product categories are available for visualization in the latest release.



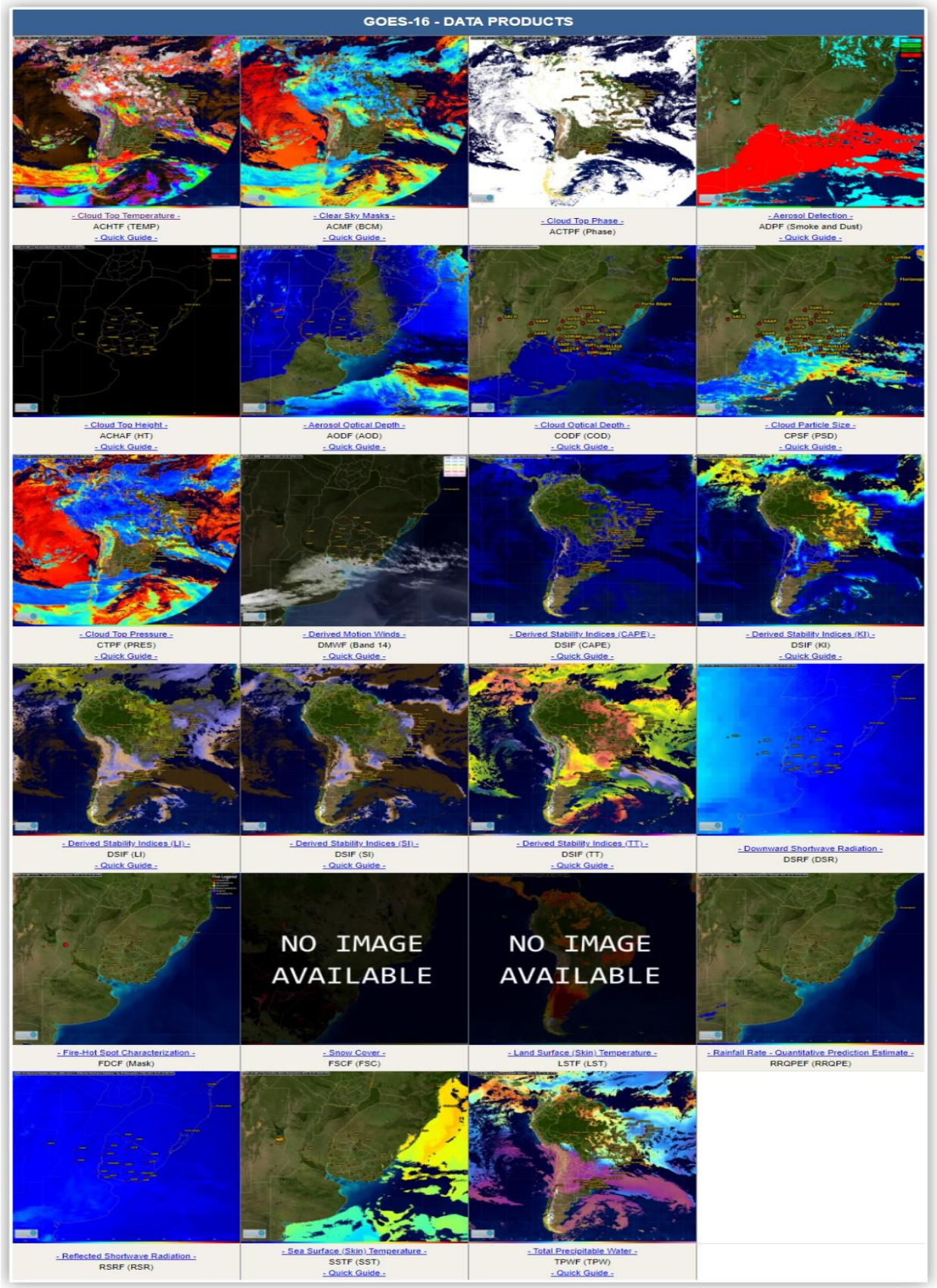
- **GOES-16: Individual ABI Bands (Full Disk and Sectorized)**



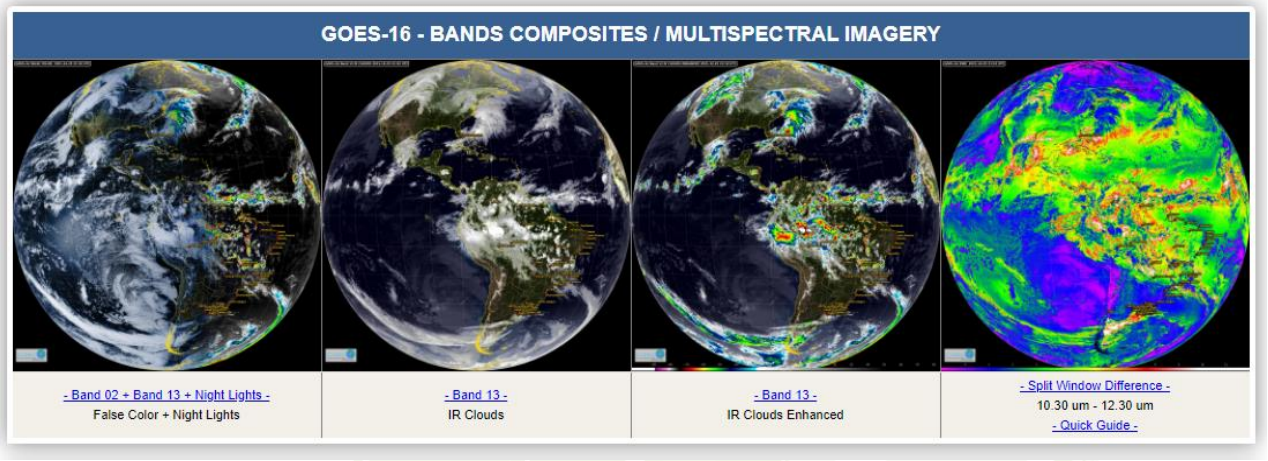
- **GOES-16: RGB Composites (Full Disk and Sectorized)**



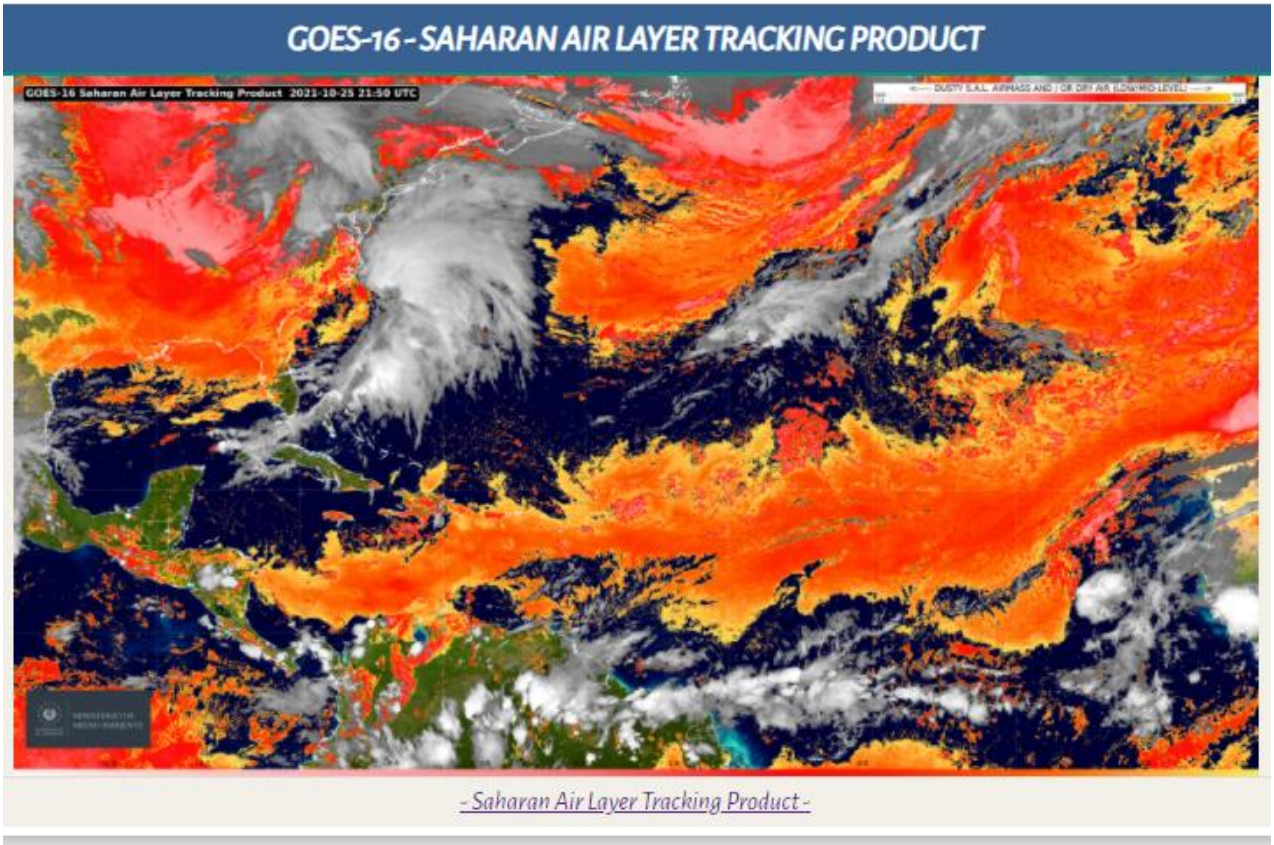
- **GOES-16: Data Products (Full Disk and Sectorized)**



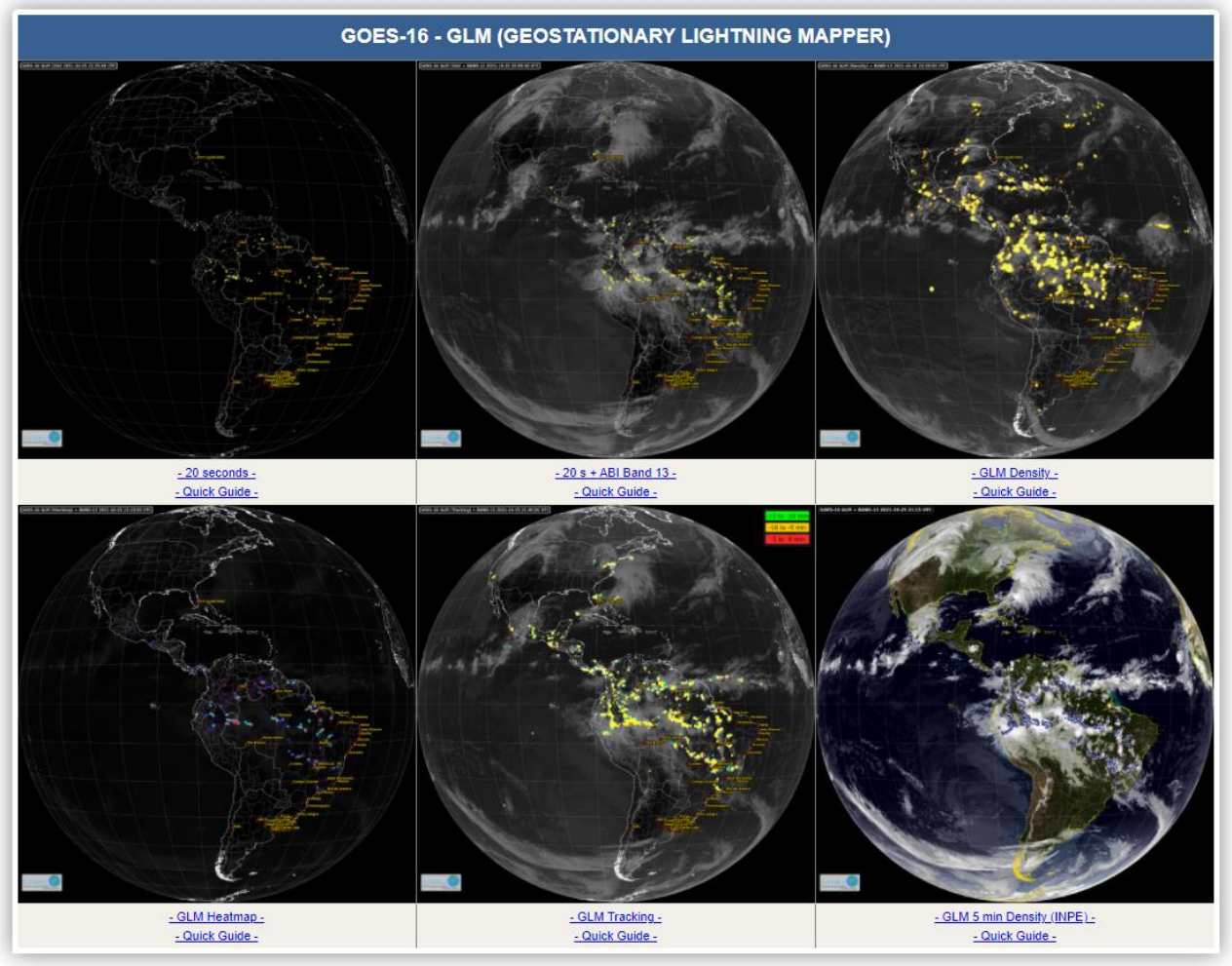
- **GOES-16: Multispectral Imagery (Full Disk and Sectorized)**



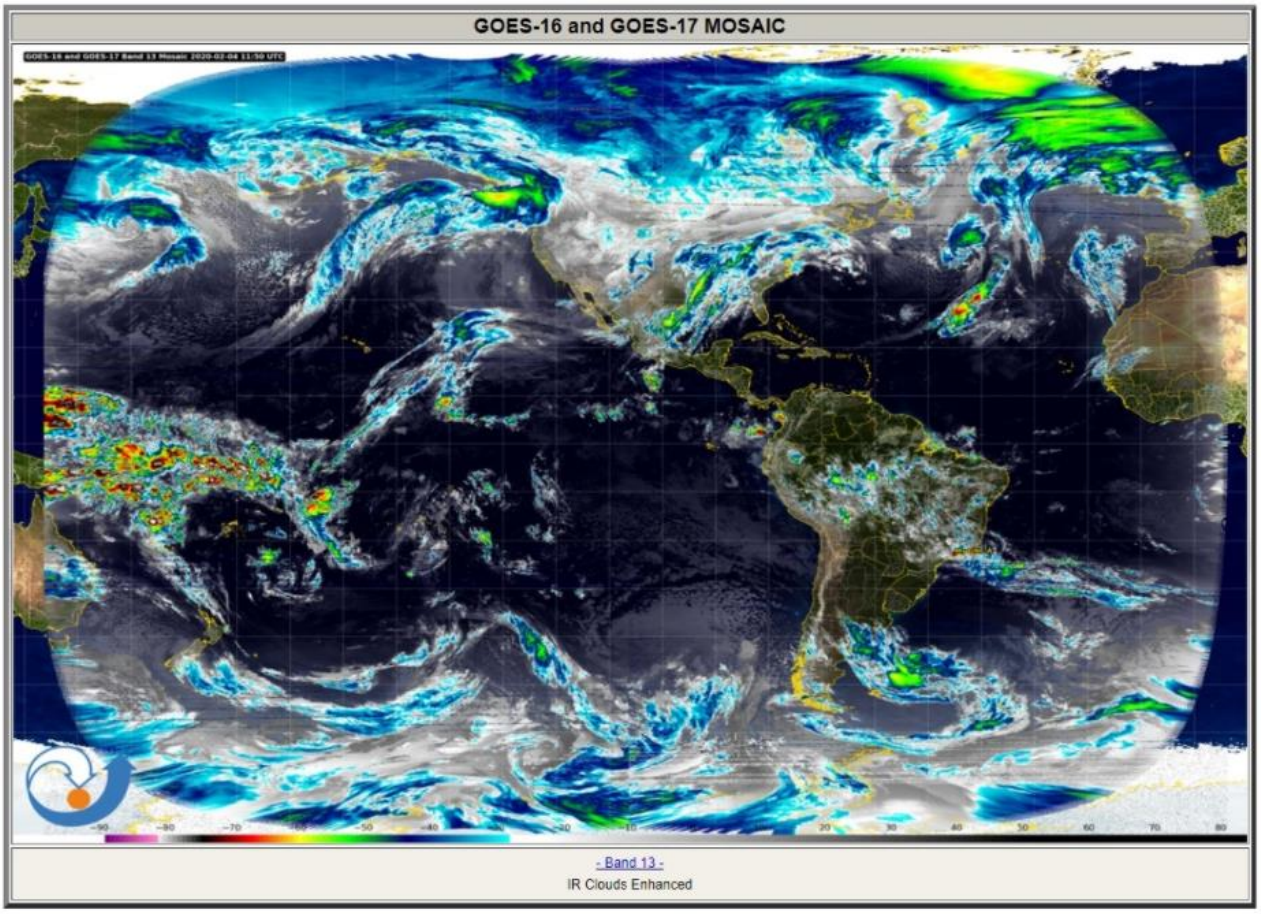

- **GOES-16: Multispectral Imagery (Sectorized)**



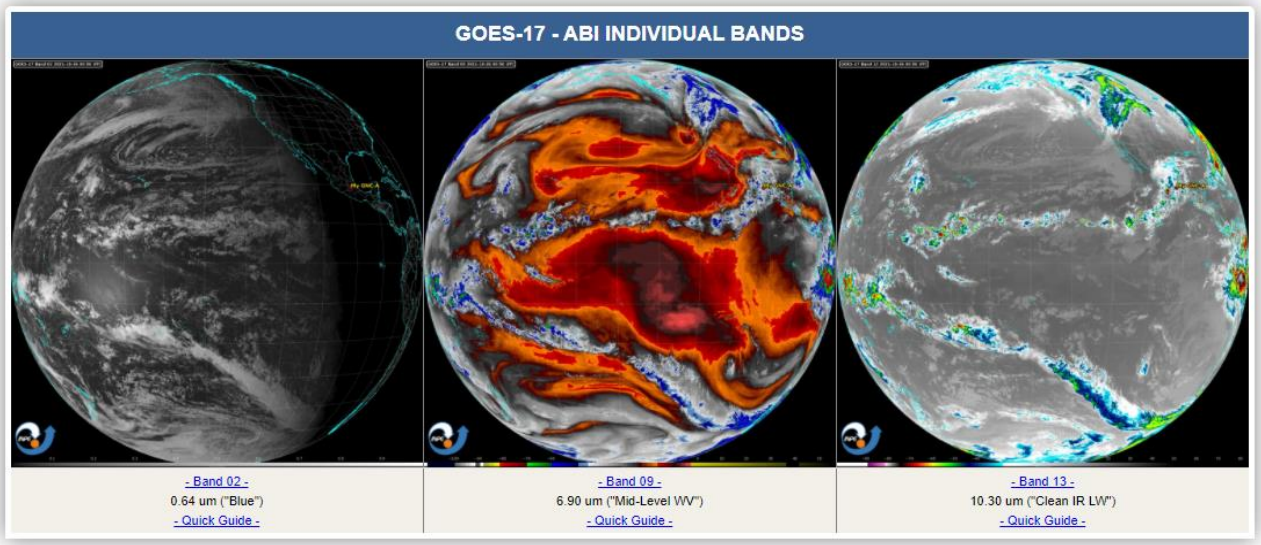
- **GOES-16: GLM (Full Disk and Sectorized)**

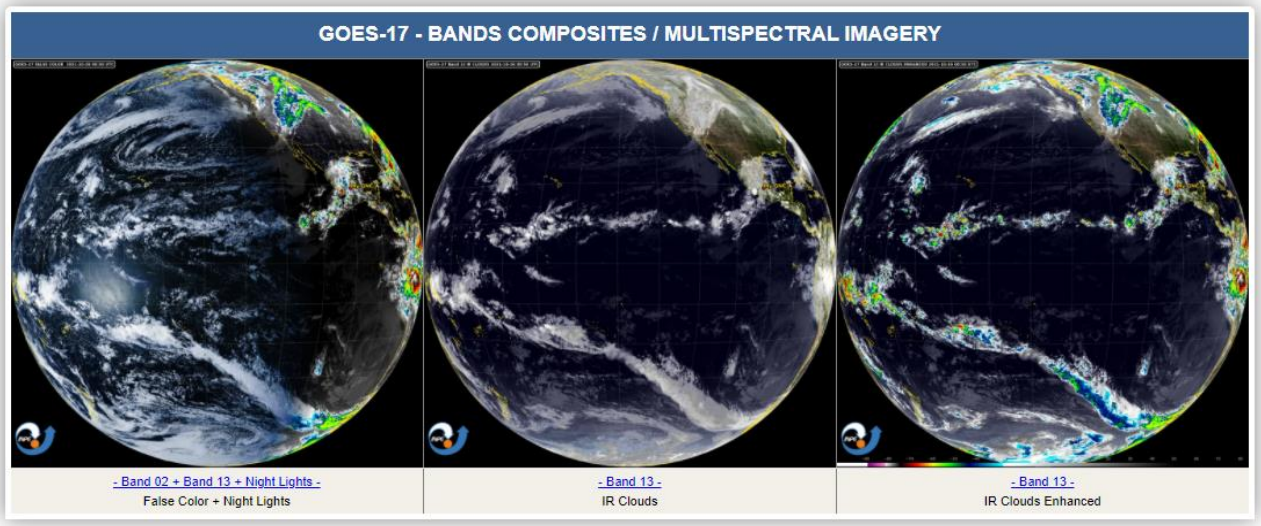


- **GOES-16 + GOES-17 Mosaic (Sectorized)**

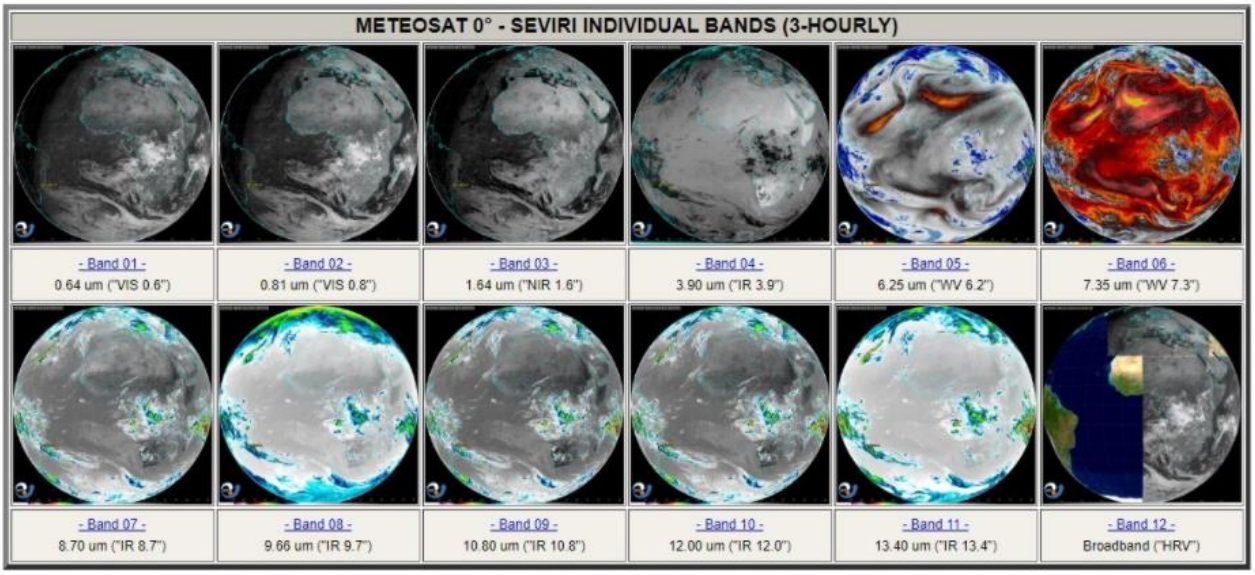


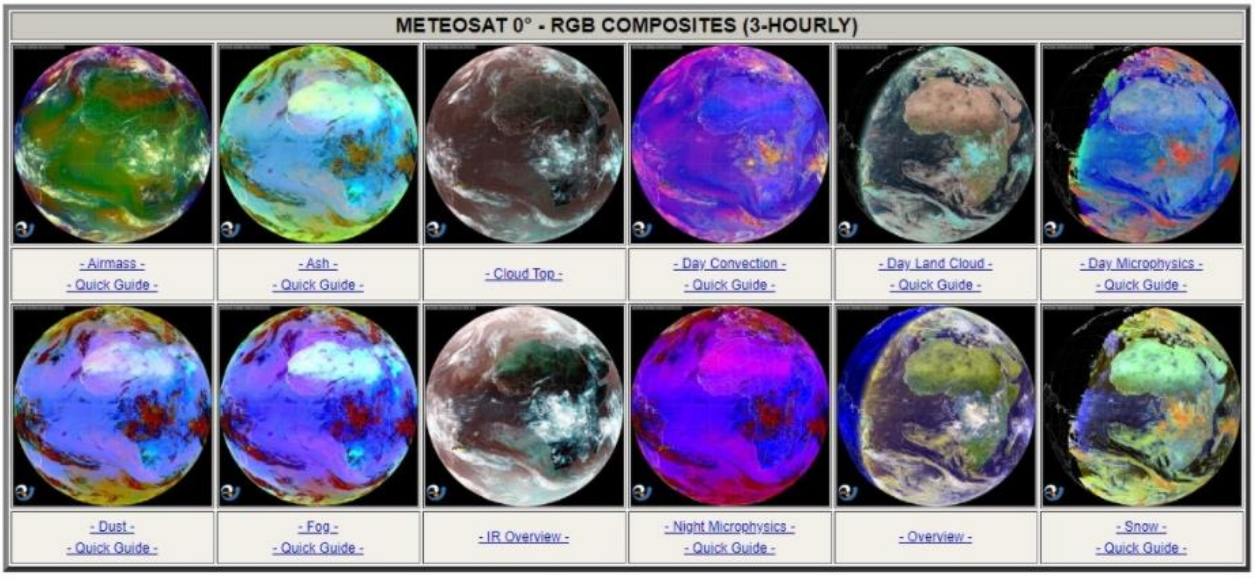
- **GOES-17 Bands and Composites (Full Disk and Sectorized)**



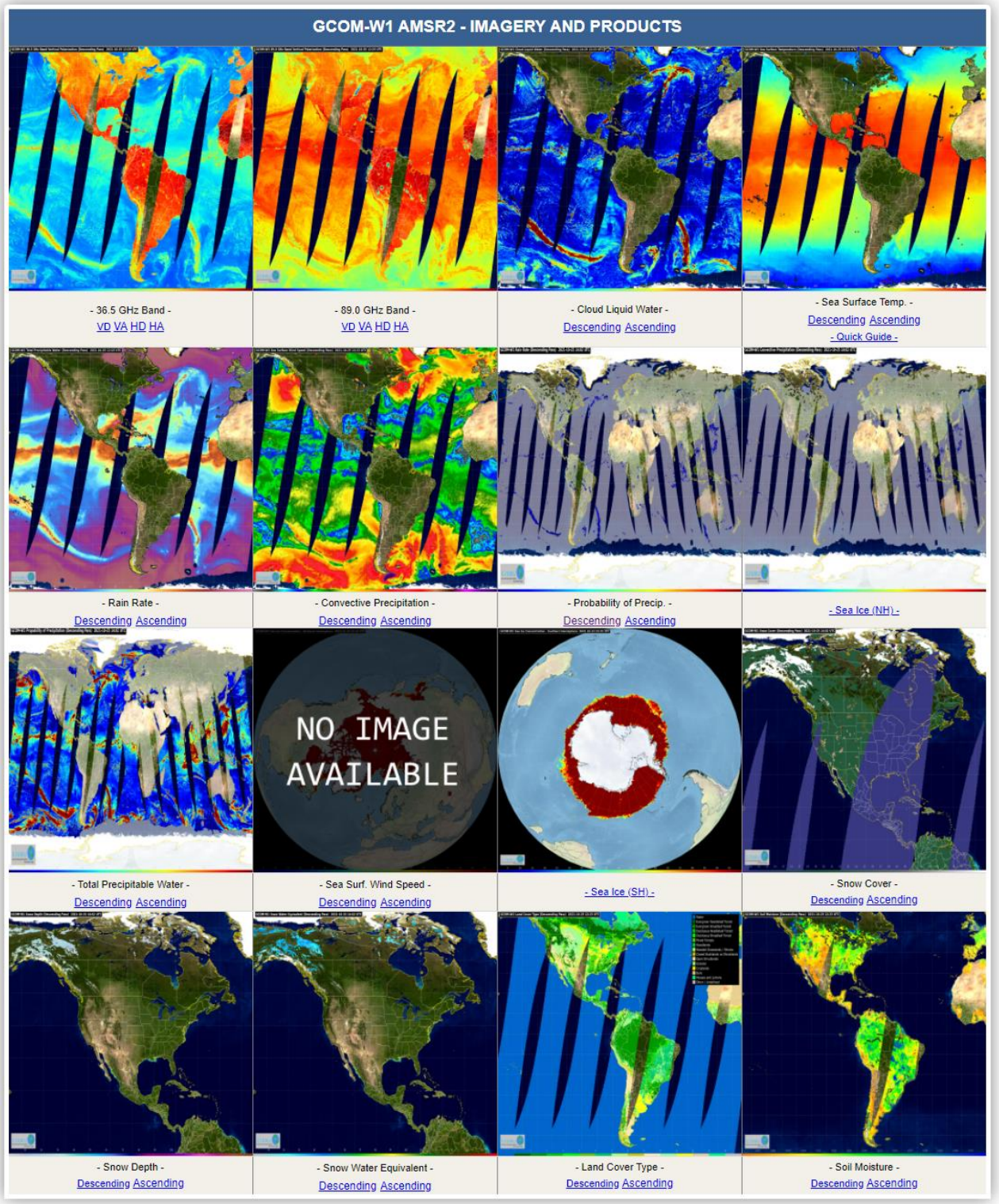



- **METEOSAT Bands and Composites (Full Disk and Sectorized)**

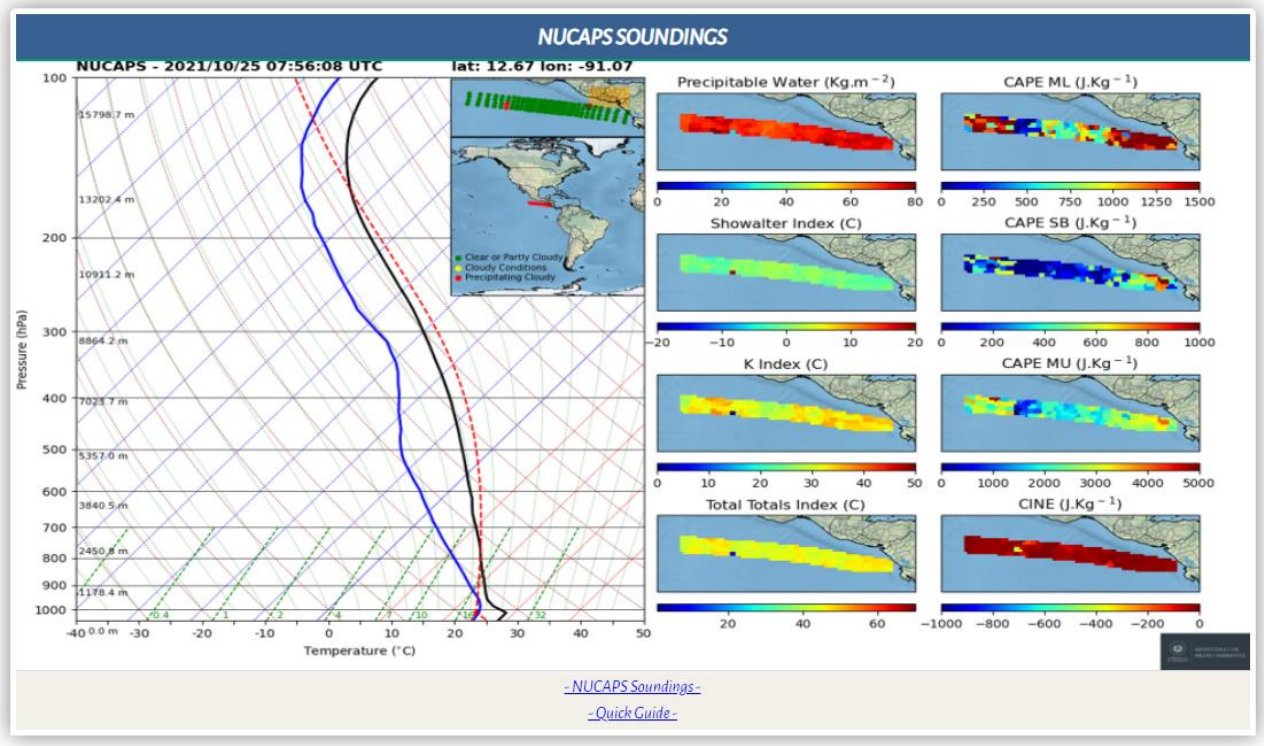


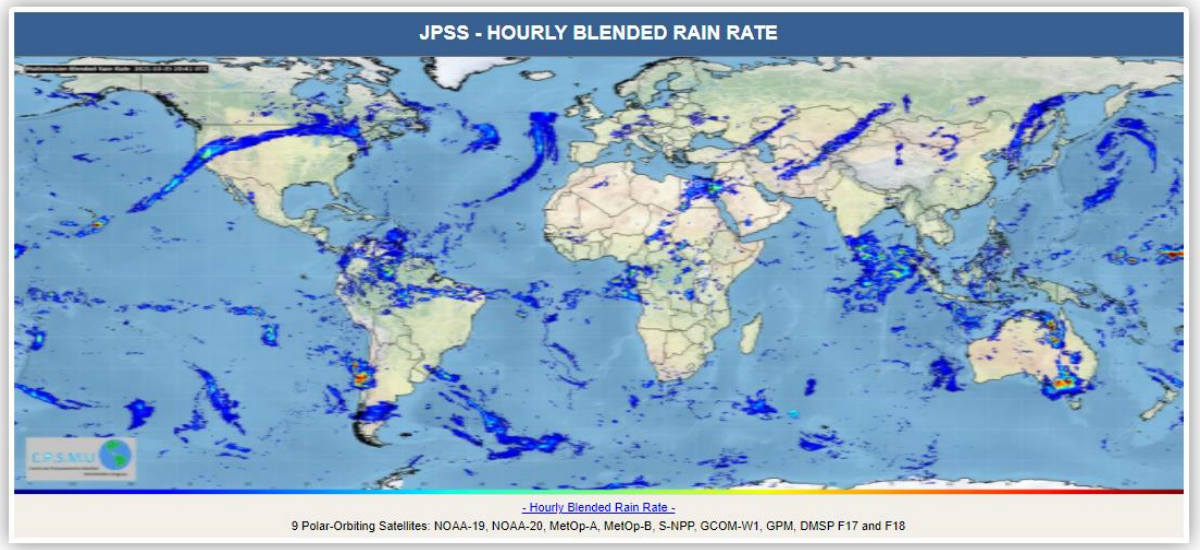
- **GCOM-W1 AMSR2 Imagery and Products**



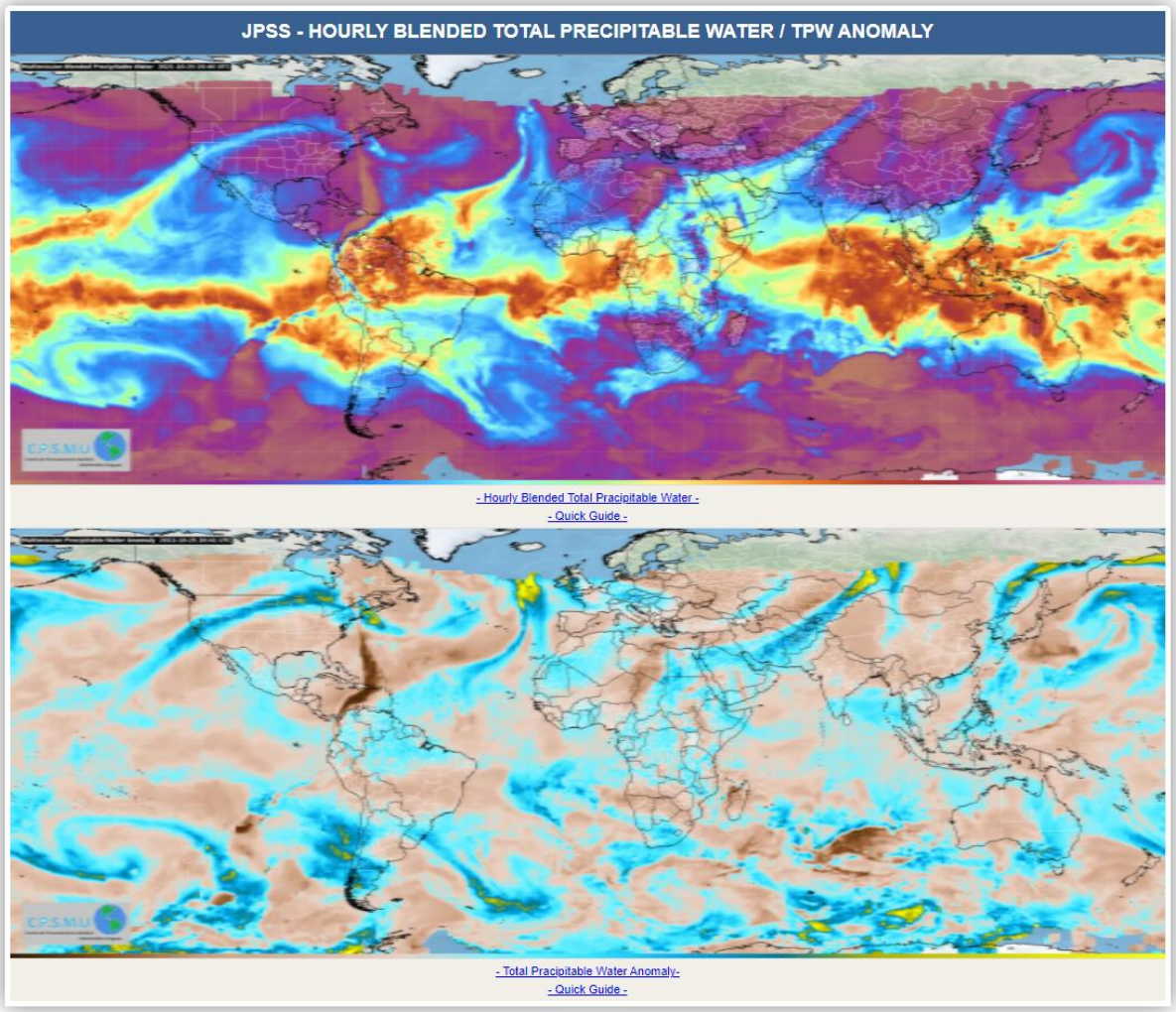
- **NUCAPS Soundings**



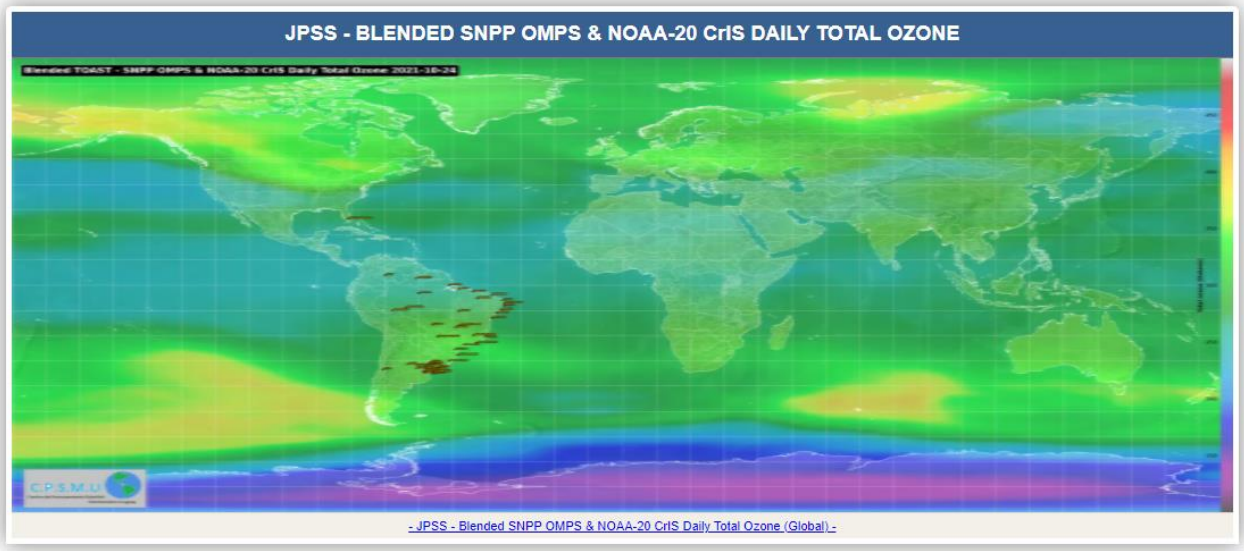
- **Blended Rain Rate**

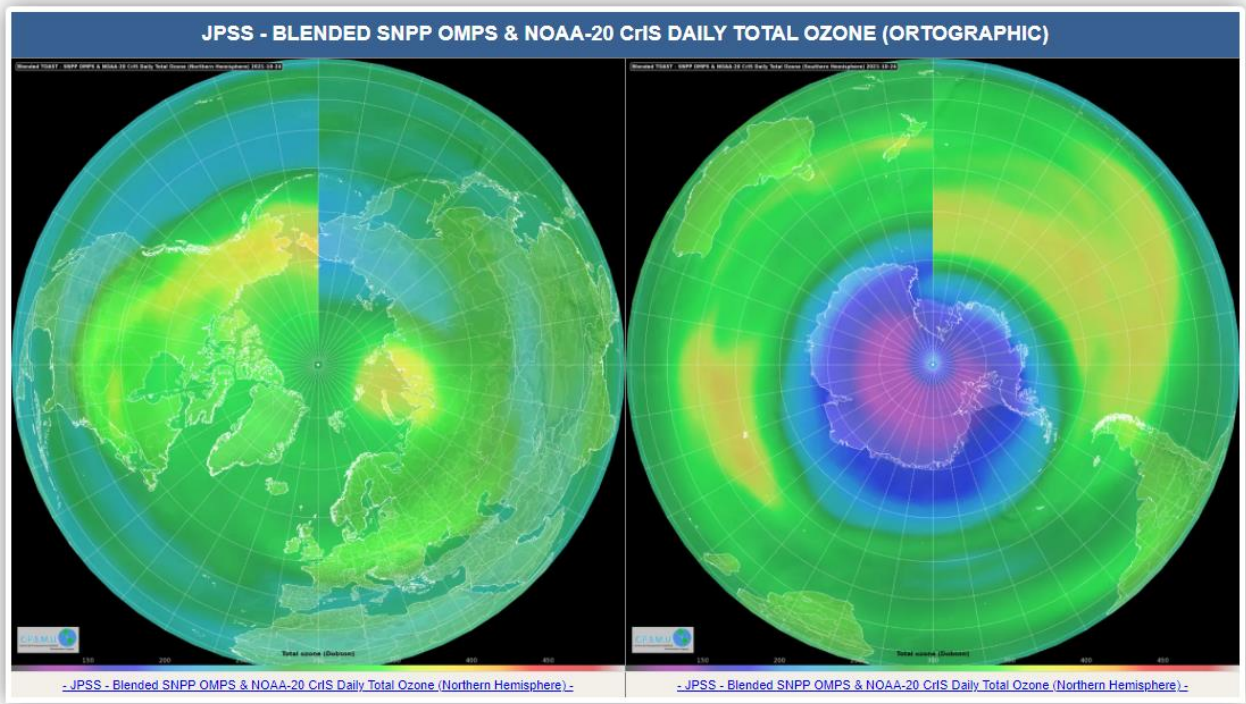



- **Blended TPW and TPW Anomaly**

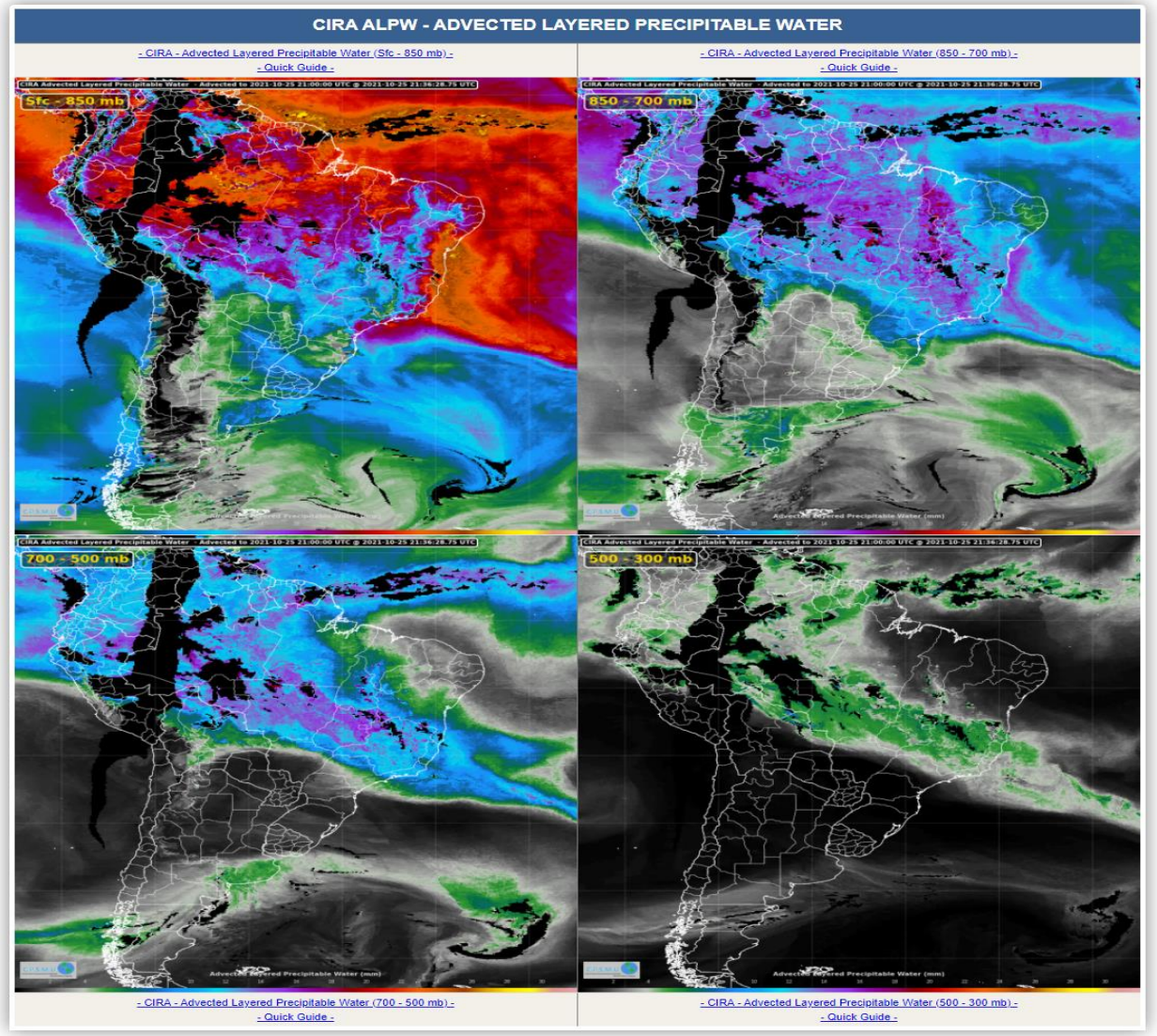


- **Blended Ozone**

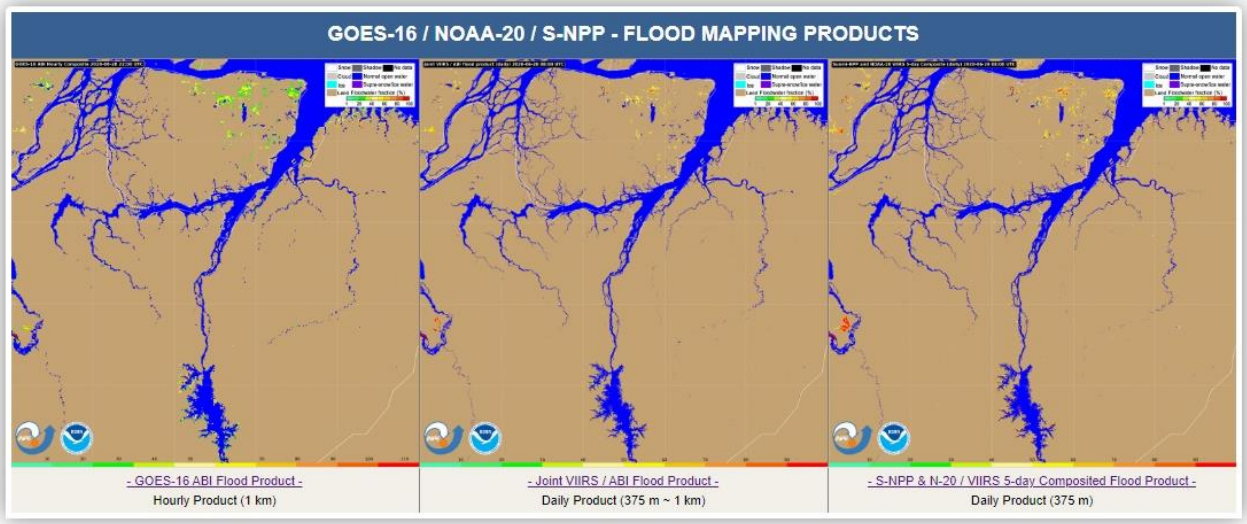


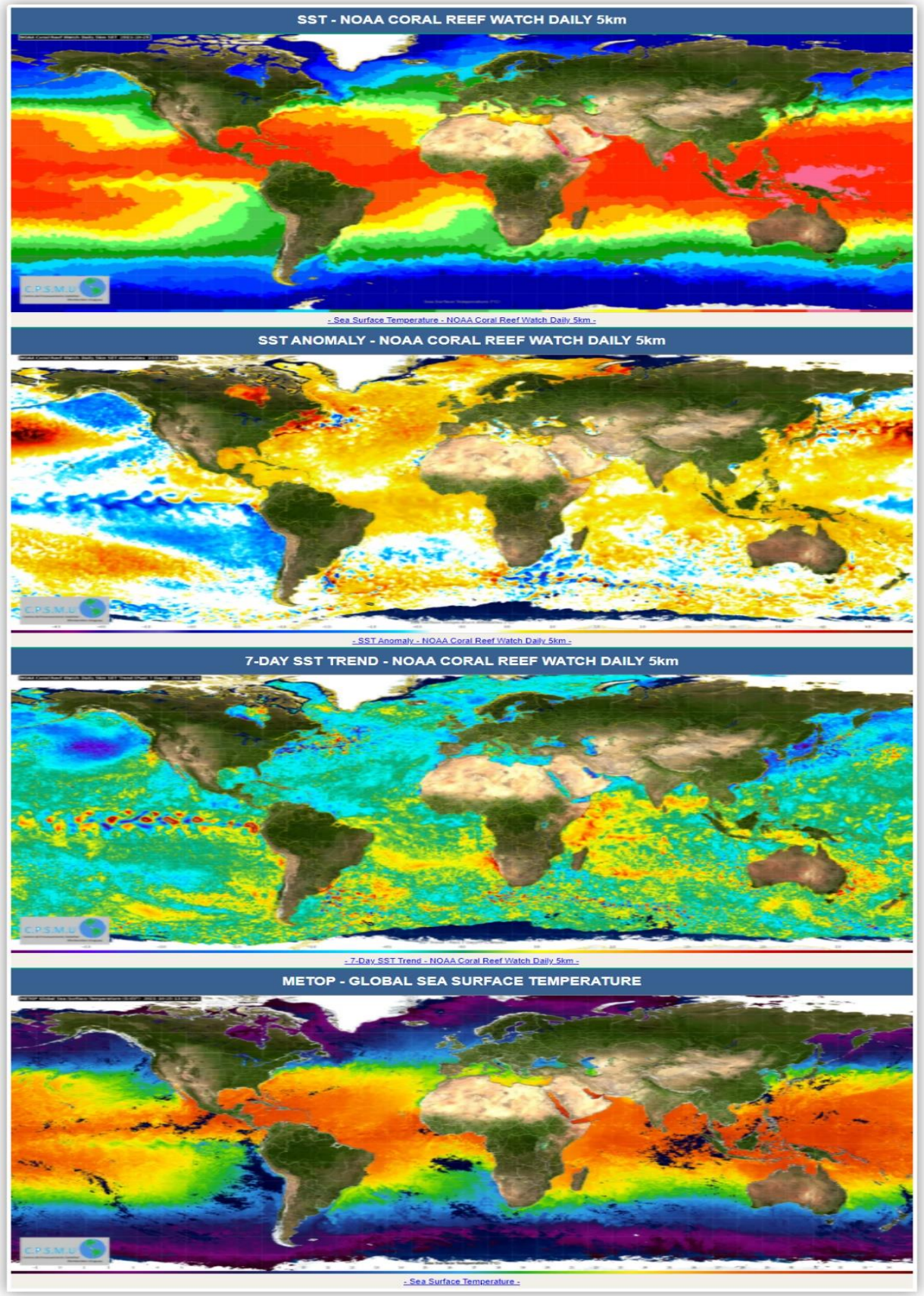
- **CIRA Advected Layered Precipitable Water**



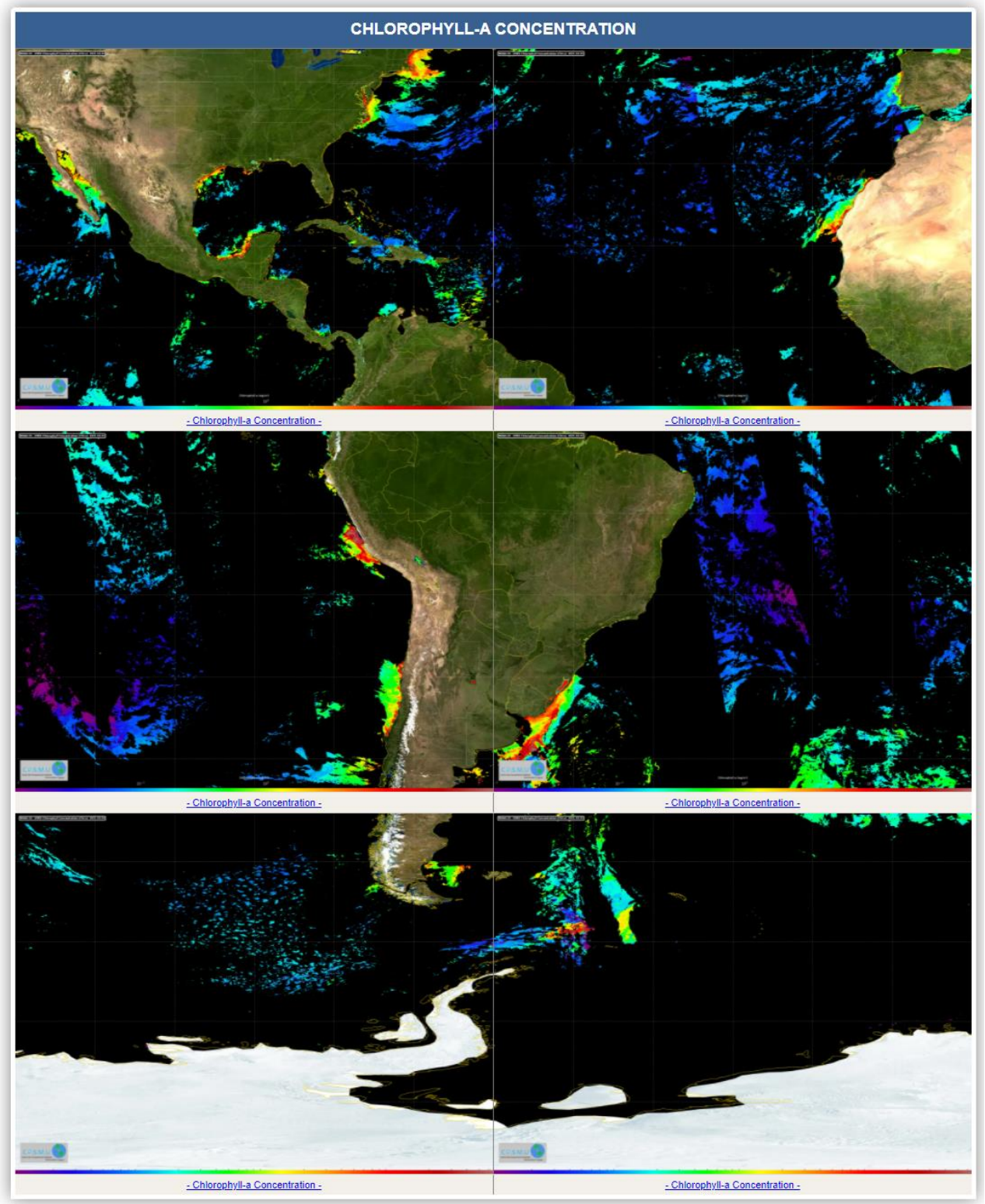

- **Flood Mapping Products**



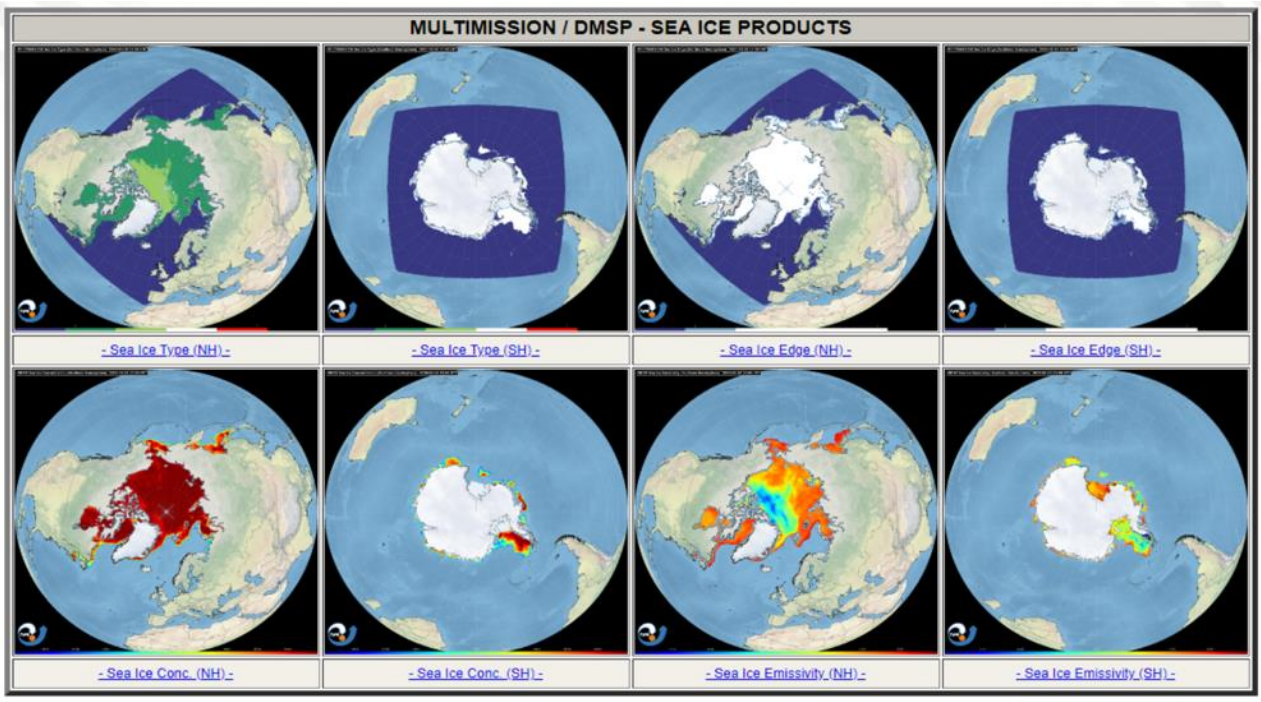
- **SST, SST Anomaly and SST Trend**



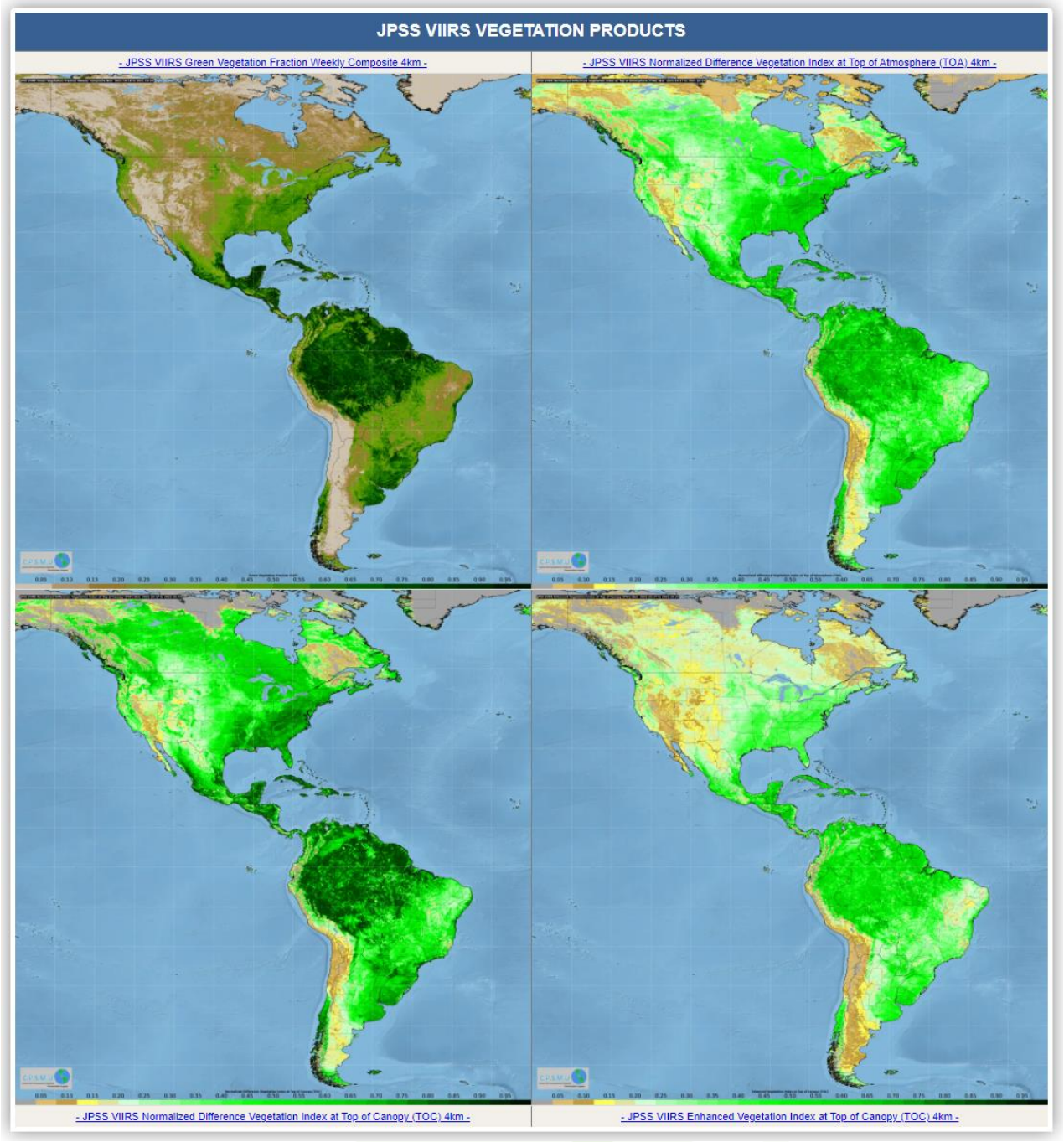
- **Ocean Color**


CHLOROPHYLL-A CONCENTRATION
- Chlorophyll-a Concentration -
- Chlorophyll-a Concentration -
- Chlorophyll-a Concentration -
- Chlorophyll-a Concentration -
- Chlorophyll-a Concentration -
- Chlorophyll-a Concentration -

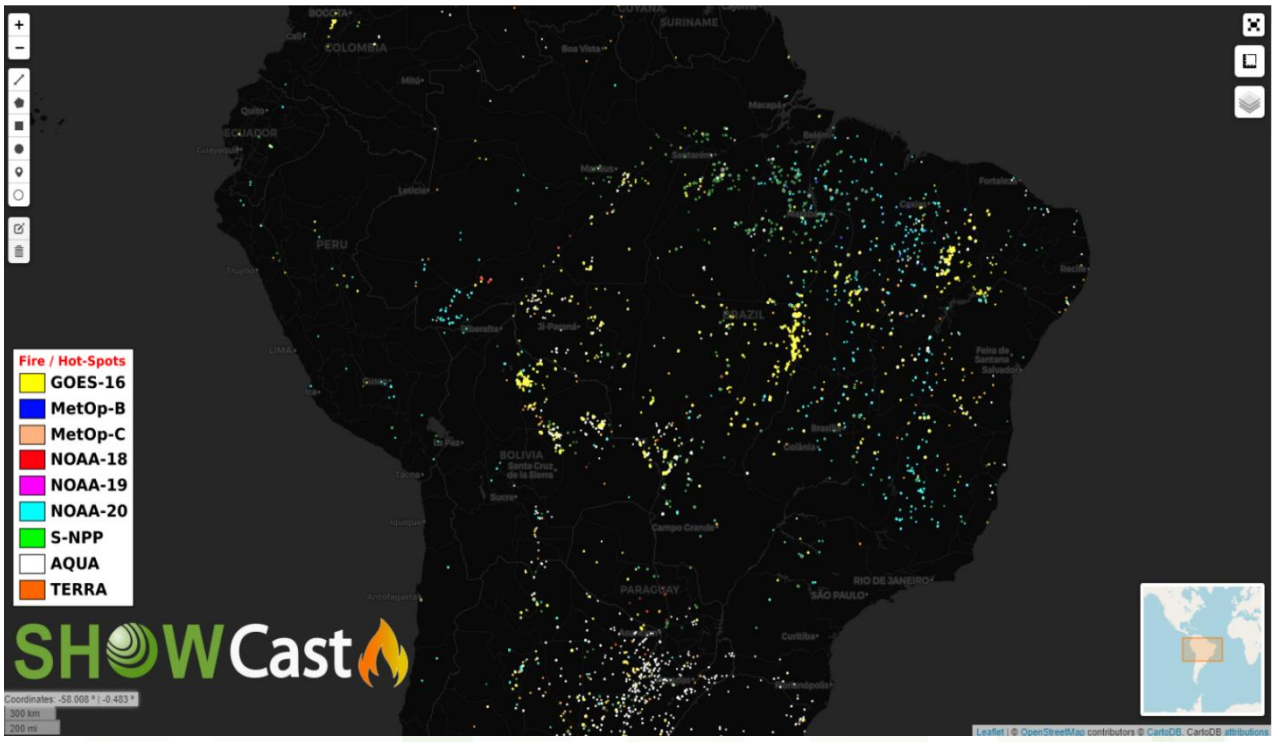
- **Multimission / DMSP Sea Ice Products**

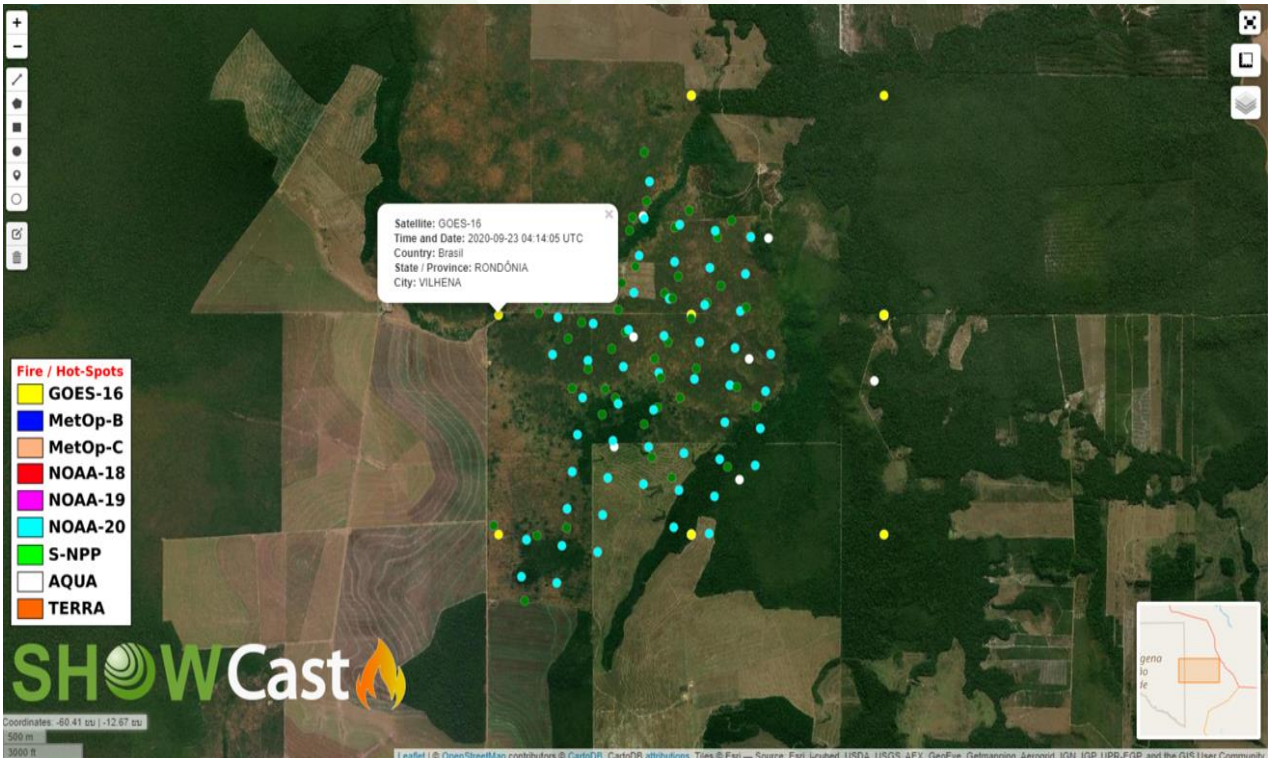


- **JPSS VIIRS Vegetation Products**

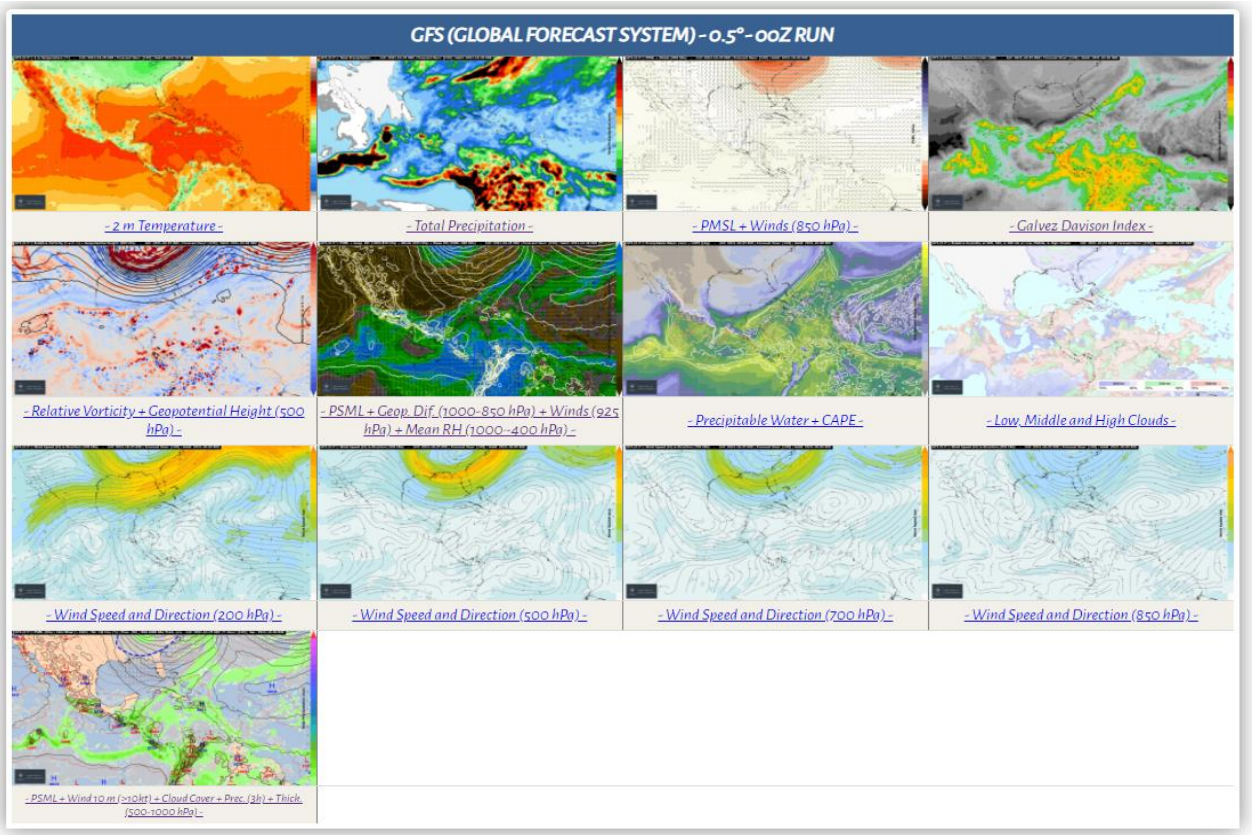


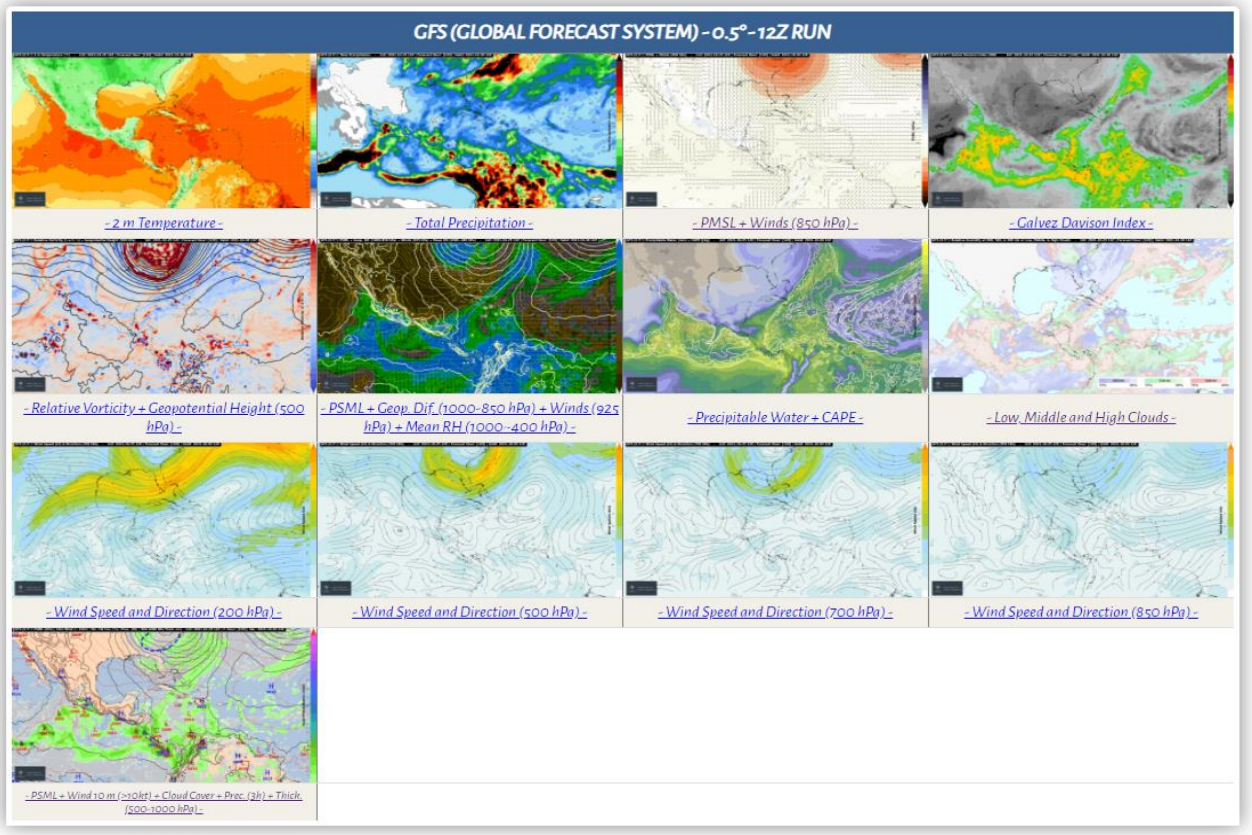
- **Multimission Fire / Hot-Spots**



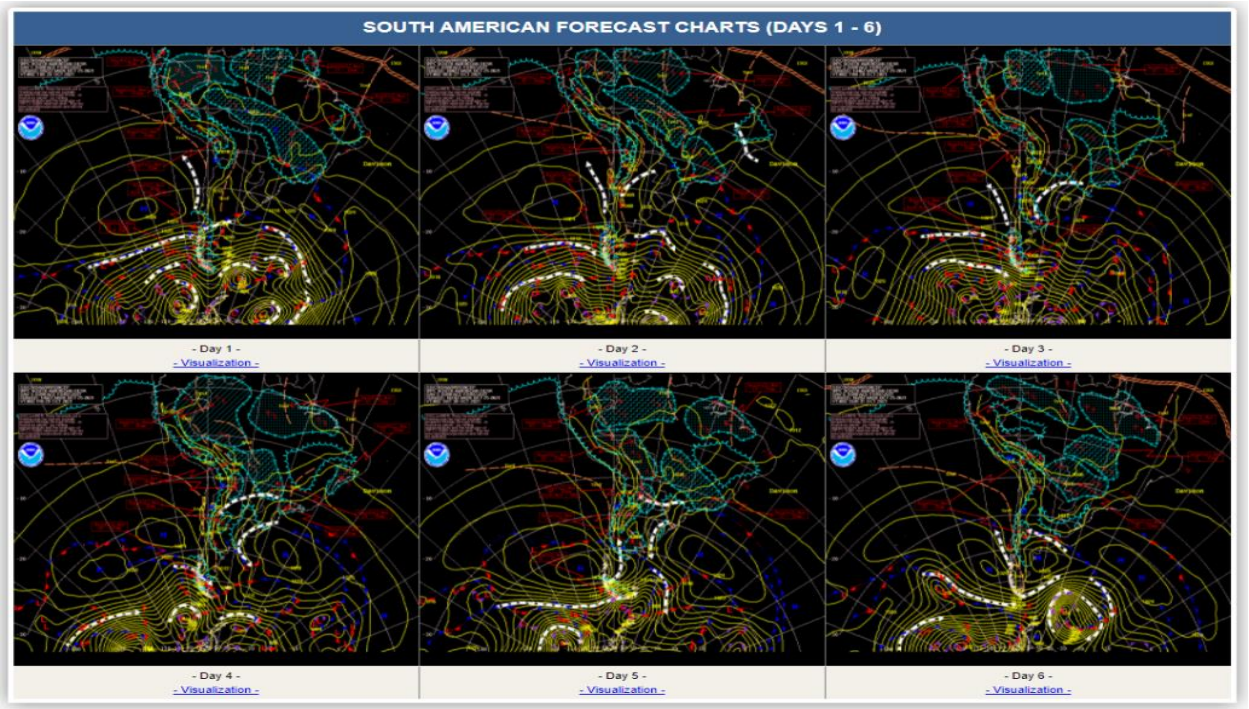



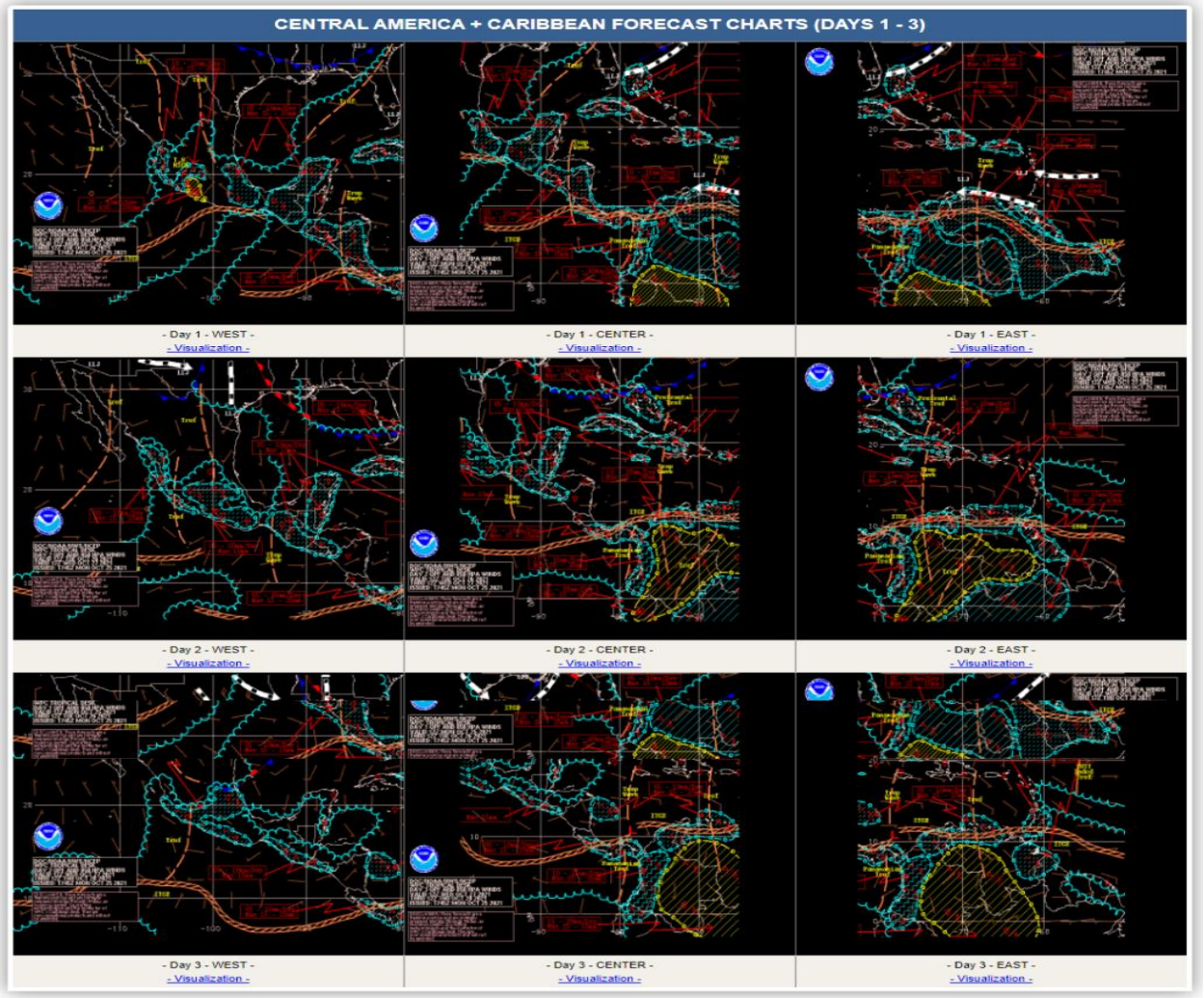
- **GFS Plots (Central America + Caribbean / South America) - 00Z and 12Z Run**



- **Forecast Charts (Central America + Caribbean / South America)**



# 21 APPENDIX II: READER NOTES