
Sentinel-2 Compositing Tool

Release 0.1

Adriana Caswell, Christian Devey, and Muhammad Ba

Apr 08, 2025

CONTENTS:

1	About W25Project_Compositing	1
1.1	Sentinel-2 Compositing Tool	1
1.2	Features	1
1.3	Repository contents	1
1.4	Installation	2
1.5	Instructions	2
1.6	Example Workflow	3
1.7	Troubleshooting	3
1.8	Credits	4
2	Sentinel-2 Compositing Tool Demo	5
2.1	Demo Data	5
2.2	Running The Tool	5
2.3	Installation	6
2.4	Demo Data Setup	6
2.5	Downloading your own Sentinel-2 Imagery	6
2.6	Getting Started	8
3	s2composite module	15
	Python Module Index	19
	Index	21

ABOUT W25PROJECT_COMPOSITING

1.1 Sentinel-2 Compositing Tool

This tool creates median value composites from multiple Sentinel-2 images for B01, B02, B03, B05, B08, B11, and B12. It resamples the composites to a finer 10 m resolution and generates detailed statistical analyses of the data. By clipping the imagery to a specific area of interest (AOI), this tool allows users to focus on relevant regions and ensure that the data used is both accurate and tailored for their needs. Additionally, it resamples the composite bands to a 10m resolution, enhancing spatial resolution for more precise analysis. The tool provides in-depth statistical summaries, such as the mean, variance, and other metrics, helping users better understand the data's characteristics and variability. Developed by Adriana Caswell, Christian Devey, and Muhammad Ba, this tool is designed for people in the field of geospatial analysis looking to create median value composites of Sentinel-2 imagery.

1.2 Features

- **Preprocessing:** Clips Sentinel-2 SAFE files to your area of interest (AOI) and converts them to TIFF format for further analysis.
 - **Compositing:** Creates median value composites from multiple Sentinel-2 bands, ensuring a robust representation of the observed area over time.
 - **Resampling:** Resamples the composite bands to 10m resolution for higher spatial detail.
 - **Statistics:** Provides detailed statistical analysis for each composite band, helping to better understand the underlying data patterns.
 - **Visualization:** Displays composite bands and individual bands for further visual interpretation and analysis.
-

1.3 Repository contents

- **README.md** - provides an introduction to the tool and the repository
- **licence.txt** - file containing the MIT licence associated with this tool
- **s2compo_env.yml** - conda environment that manages the dependencies for the tool
- **s2composite.py** - Python module containing all functions needed for the tool
- **S2CompoTool.ipynb** - Jupyter Notebook allowing users to easily interact with the s2composite module
- **/docs** - folder containing Sphinx Autodocumentation files

- **/demo** - folder containing sample input data and folder structure for the demonstration (as outlined in the documentation)
- **GEOM4009_FinalPresentation.pdf** - a copy of the final presentation associated with this project
- **GEOM4009_FinalReport.pdf** - a copy of this report
- **.gitignore** - file indicating files and directories to ignore when making commits

The repository also contains a branch (pages_docs) that contains the Sphinx Autodocumentation html files structured to create the repository's GitHub Pages.

1.4 Installation

1. Clone this repository: https://github.com/GEOM4009/W25Project_Compositing.git
2. Download and install [Anaconda](#)
3. Create a conda environment by navigate to W25Project_Compositing in the Anaconda Prompt and running the following command:

```
conda env create -f s2compo_env.yml
```

4. Activate the environment

```
conda activate s2compo
```

5. Open the Sentinel-2 Composting Tool and follow the Instructions below

```
jupyter lab S2CompoTool.ipynb
```

1.5 Instructions

1.5.1 1. Download Sentinel-2 Imagery

Download the Sentinel-2 imagery for your area of interest from the Copernicus Browser.

1.5.2 2. User Input Setup

Define **User input** variables in the S2CompoTool Jupyter Notebook.

1.5.3 3. Running the Code

Once the setup is complete, you can *Run All Cells* or go through the workflow cell-by-cell.

3.1 Preprocessing

The tool will automatically preprocess the Sentinel-2 data, clip it to the AOI, and convert the bands to TIFF format. The clipped TIFF files will be saved in the user input `clippedS2` directory, ensuring they are ready for compositing.

After this step, users have the option to perform cloud masking (cloud masking functionality is not a part of the tool) or remove images with excessive cloud cover to improve the compositing results.

3.2 Compositing and Resampling

The tool will create median value composites from the clipped bands and resample them to 10m resolution for detailed analysis. The generated composites and resampled composites will be stored in the **composites** directory:

- **Median composites:** Median composites for each band. Saved as B**_composite.tif.
- **Resampled composites:** The median composites resampled to a 10m resolution for improved spatial analysis. Saved as B**_resampled10m.tif.

3.3 Statistics

Once the median value composites are created, the tool can provide statistics on the composite bands to help analyze the data further. The statistics are essential for understanding the overall composition of the data and can be used to draw insights for specific applications like vegetation analysis or land use classification. Available Statistics:

- **Mean:** Average value for each pixel across all the images in the composite.
- **Standard Deviation:** The measure of variation of pixel values in the composite.
- **Min and Max:** The range of pixel values across all the bands in the composite.

3.4 Visualization

The tool can display individual bands or RGB composites for visual inspection:

- **RGB Composite:** Uses bands B02 (blue), B03 (green), and B04 (red) for generating a RGB composite image.
 - **Individual Bands:** Displays each band separately, such as B01, B02, etc., to allow for detailed analysis of specific wavelengths.
-

1.6 Example Workflow

1. **Download Sentinel-2 data**
 2. **Define user inputs:** In the S2CompoTool, specify the input, output, and AOI paths in the user input section.
 3. **Run S2CompoTool:** Preprocess, create composites, and resample.
 4. **Visualize results:** View the composites and individual bands for analysis.
 5. **Analyze the data:** Use the statistics and visual output to gain insights about your area of interest.
-

1.7 Troubleshooting

If you encounter any issues, refer to the following troubleshooting steps:

- **Missing Files:** Ensure that all required input files, such as the Sentinel-2 SAFE files and AOI shapefile, are correctly specified in the input section.
 - **File Paths:** Make sure all paths are correct and accessible. The script may fail if paths are incorrectly specified or if files are missing.
 - **Memory Issues:** If the process takes too long or runs out of memory, try processing smaller regions or reduce the number of bands used.
-

1.8 Credits

- **Adriana Caswell, Christian Devey, and Muhammad Ba** for the tool's development
 - **Dr. Derek Mueller and Dr. Anders Knudby** for initiating the project and providing guidance throughout its development
-

SENTINEL-2 COMPOSITING TOOL DEMO

Created by **Adriana Caswell, Christian Devey, and Muhammad Ba**

This demo showcases the **Sentinel-2 Compositing Tool**, designed to streamline the process of working with Sentinel-2 imagery. The tool generates **median value composites** from multi-temporal Sentinel-2 data, resamples them to a consistent 10m resolution, and performs various statistical analyses to provide insights into the data. This can be useful for monitoring vegetation, land use, and other environmental factors over time.

2.1 Demo Data

The demo is pre-configured with a set of **demo data** that can be used to test the tool. The demo data includes:

- **StudyArea.shp**: Defines the area of interest (AOI) for analysis.
- **Sentinel-2 Imagery**: The corresponding Sentinel-2 images aligned with the study area, which are provided in the demo/S2 folder along with a text file listing the required imagery and download links.

2.2 Running The Tool

The tool automates the preprocessing, compositing, and analysis tasks, saving valuable time and ensuring reproducibility. The process includes:

- **Preprocessing**: Clipping Sentinel-2 SAFE files to the AOI and converting them to the appropriate format.
- **Compositing**: Creating median value composites to summarize multi-temporal data into a single image.
- **Resampling**: Rescaling the image resolution to 10m for consistency.
- **Statistical Analysis**: Calculating key statistics such as mean and variance over defined grid areas.

Once you run the tool, it will automatically execute all necessary steps in sequence, from data preprocessing to the final visualization and statistical analysis, making it user-friendly and efficient. However, if you prefer, you can also run each cell individually to have more control over the process and explore the results at each step.

Throughout this guide, we will provide **screenshots** for each step to help you visualize the process and ensure everything is set up correctly. This tool is especially valuable for researchers and practitioners working with satellite data who need to quickly and accurately analyze large datasets.

By the end of this demo, you will have learned how to use the tool with the provided **demo/S2**, also customize inputs, and interpret the resulting output, enabling you to apply it to your own Sentinel-2 data for environmental monitoring and analysis.

2.3 Installation

1. Clone the repository:

```
git clone https://github.com/GEOM4009/W25Project_Compositing.git
```

2. Install [Anaconda](#) if you haven't already.

3. Navigate to the cloned directory and create a Conda environment:

```
conda env create -f s2compo_env.yml
```

4. Activate the environment:

```
conda activate s2compo
```

5. Open the Sentinel-2 Compositing Tool:

```
jupyter lab S2CompoTool.ipynb
```

2.4 Demo Data Setup

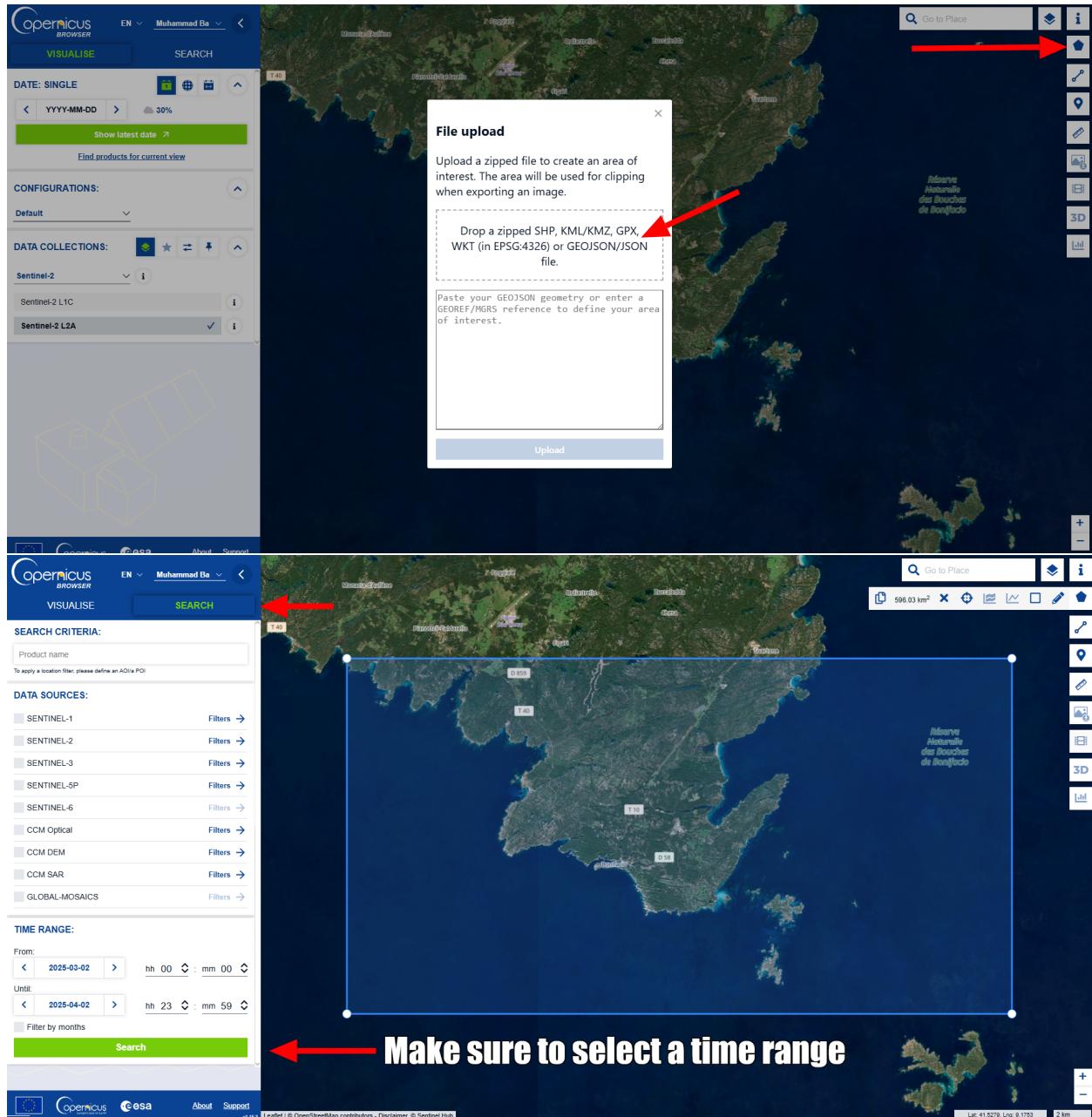
In the `demo/S2` folder, you will find a text file that outlines the specific Sentinel-2 imagery used in the demo. This file provides detailed information on the imagery aligned with the study area, ensuring that the selected satellite data matches the area of interest for your analysis. The images have already been downloaded and are pre-organized in the folder, so there is no need for you to manually download or save any additional files.

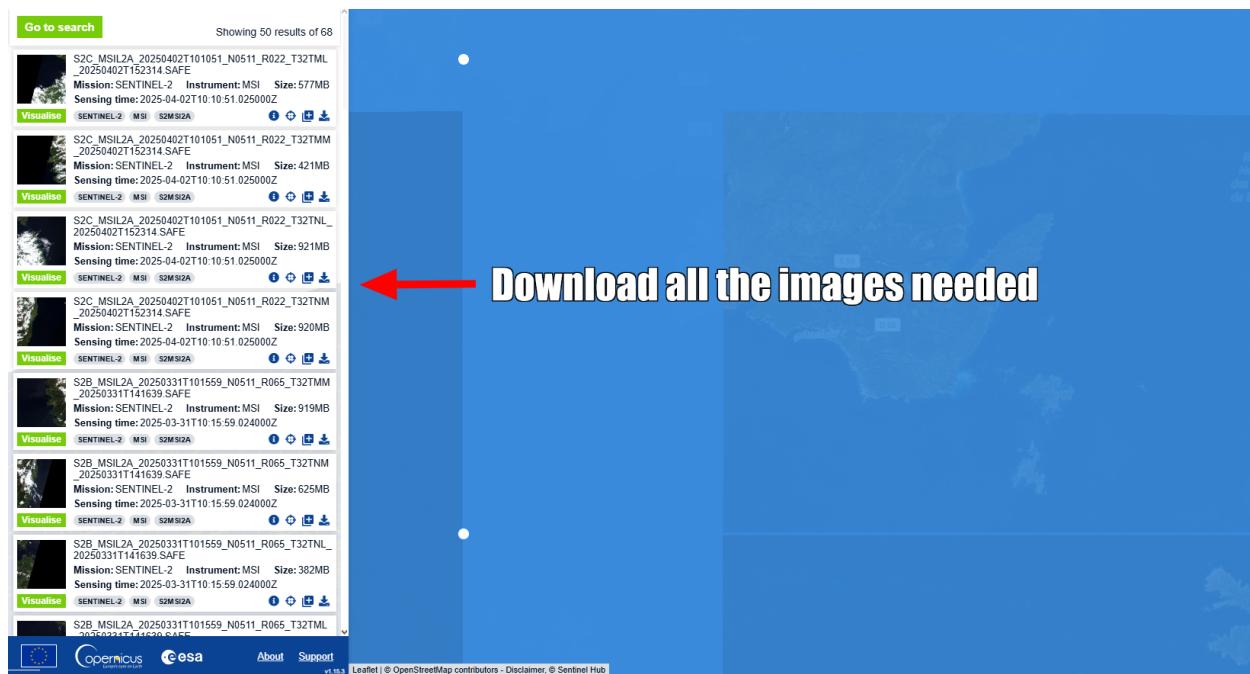
Additionally, the `StudyArea.shp` file is already included in this folder. This shapefile defines the Area of Interest (AOI) for the analysis, which is used to clip the Sentinel-2 imagery. Since the AOI is pre-configured to match the study area, you don't need to make any modifications. However, if you want to analyze a different region, you can replace this shapefile with one that corresponds to a new AOI.

The `demo/S2` folder is already set up with all the necessary files, and the tool is configured to automatically detect and use them during the preprocessing, compositing, and analysis steps. This setup ensures that you can run the demo without any additional configuration or file management.

2.5 Downloading your own Sentinel-2 Imagery

First, download Sentinel-2 data from the [Copernicus Browser](#).





2.6 Getting Started

2.6.1 Set Up User Inputs

In this section, you will define the paths and other settings necessary for the tool to run properly. These variables control the location of your Sentinel-2 data, the clipped data, and where your output files will be saved. Make sure to adjust the paths according to your system and data storage locations.

Name	Date modified	Type	Size
inputsS2	2025-03-26 12:45 PM	File folder	
Corsica	2025-03-26 12:43 PM	Compressed (zipp...)	3 KB

Make sure to created a new folder where we will keep everything needed to run the tool. You will need your zipped SHP file which in this case is "Corsica". Also create a new folder named "inputs2"

Name	Date modified	Type	Size
clippedS2	2025-03-26 1:29 PM	File folder	
composites	2025-03-26 1:30 PM	File folder	
S2A_MSIL2A_20250318T101751_N0511_R0...	2025-03-26 12:44 PM	Compressed (zipp...)	419,800 KB
S2A_MSIL2A_20250325T100701_N0511_R0...	2025-03-26 12:44 PM	Compressed (zipp...)	547,592 KB
S2A_MSIL2A_20250325T100701_N0511_R0...	2025-03-26 12:44 PM	Compressed (zipp...)	359,898 KB
S2B_MSIL2A_20250321T101629_N0511_R0...	2025-03-26 12:45 PM	Compressed (zipp...)	372,653 KB

After opening this folder. Go ahead and make two new subdirectories names "clippedS2" and "composites" which will be needed for the tool to run. You should also be saving all your sentinel 2 images into the "inpuutsS2" folder (sentinel 2 images highlighted in red). PLEASE NOTE ALL THE PATHS OF THESE FILES AND DIRECTORIES AS WE WIL NEED THEM FOR THE TOOL.

2.6.2 In the S2CompoTool.ipynb Notebook, define the necessary input variables:

```
inputS2 = "demo/S2"
clippedS2 = "demo/S2/clipped"
compS2 = "demo/S2/composites"
aoiShp = "demo/S2/StudyArea.shp"
lineSpacing = 5000
```

Sentinel-2 Compositing Tool

Created by Adriana Caswell, Christian Devey, and Muhammad Ba

This tool creates median value composites for Sentinel-2 bands, resamples them to 10 m, and provides statistics about the composite. For more details, view the documentation.

```
[1]: import s2composite as s2
import rasterio as rio
from rasterio.plot import show
```

User inputs

Access the tool using jupyter labs.

Source Sentinel-2 imagery for your area of interest from the Copernicus Browser and define the following variables:

- **inputS2**: Path to the directory where zipped Sentinel-2 SAFE files are saved
- **clippedS2**: Path to directory where clipped TIFF files will be saved after preprocessing
- **compS2**: Path to directory where the median composites and resampled composites will be saved
- **aoiShp**: Path to the shapefile representing your area of interest (AOI) which will be used to clip the Sentinel-2 files
- **lineSpacing**: Spacing for grid statistics (in meters). This will help in calculating statistics like mean, variance, etc., over a grid defined by this spacing.

```
[2]: inputS2 = "demo//S2"
clippedS2 = "demo//S2//clipped"
compS2 = "demo//S2//composites"
aoiShp = "demo//StudyArea//StudyArea.shp"
lineSpacing = 5000
```

Change the paths depending on your computer and folder structure.

Once variables have been defined, you can Run All Cells or go through the workflow cell-by-cell below.

Create composites

+ 5 cells hidden

Composite statistics

+ 1 cell hidden

Display composites

Sentinel-2 Compositing Tool

Created by Adriana Caswell, Christian Devey, and Muhammad Ba

This tool creates median value composites for Sentinel-2 bands, resamples them to 10 m, and provides statistics about the composite. For more details, view the [documentation](#).

```
[1]: import s2composite as s2
import rasterio as rio
from rasterio.plot import show
```

User inputs

Source Sentinel-2 imagery for your area of interest from the [Copernicus Browser](#) and define the following variables:

- **inputS2**: Path to the directory where zipped Sentinel-2 SAFE files are saved
- **clippedS2**: Path to directory where clipped TIFF files will be saved after preprocessing
- **composites**: Path to directory where the median composites and resampled composites will be saved
- **aoiShp**: Path to the shapefile representing your area of interest (AOI) which will be used to clip the Sentinel-2 files
- **lineSpacing**: Spacing for grid statistics (in meters). This will help in calculating statistics like mean, variance, etc., over a grid defined by this spacing.

```
[3]: inputS2 = r"C:\School\GIS\Project\tool\inputsS2"
clippedS2 = r"C:\School\GIS\Project\tool\inputsS2\clippedS2"
compS2 = r"C:\School\GIS\Project\tool\inputsS2\composites"
aoiShp = r"C:\School\GIS\Project\tool\Corsica\Corsica\StudyArea.shp"
lineSpacing = 5000
```

In this case, here are the file paths of everything needed in order to run this the tool.

These paths should be adjusted based on your own paths.

2.6.3 Run the Tool

Once the variables are set, you can **Run All Cells** or **One at a time** in the Jupyter Notebook, and the tool will automatically execute all steps from preprocessing to visualization and statistical analysis.

2.6.4 Steps Of The Tool

2.6.5 Preprocessing

Clip Sentinel-2 SAFE files to the Area of Interest (AOI) and convert them to TIFF format:

```
s2.prepS2(inputS2, aoiShp, clippedS2)
```

This step will create clipped TIFFs, which I will show in a screenshot.

2.6.6 Create Composites

The tool generates **median value composites** and resamples them to 10m resolution:

```
bands, meta10m, meta20m, meta60m = s2.sortBands(clippedS2)
s2.compositeBands(bands, meta10m, meta20m, meta60m, compS2)
composites = s2.resampleBandsTo10m(compS2, overwrite = False)
```

Create composites

Clip the Sentinel-2 SAFE files to your area of interest and output TIFF files.

```
[]: s2.prepS2(inputs2, aoiShp, clippedS2)
Clipping and converting 4 Sentinel-2 images...
Clipping and conversion complete.
```

Optional: Perform cloud masking or remove images with excessive cloud cover to improve the compositing results.

Run the following code block to create median value composites for each band and resample the composites to 10 m.

```
[]: bands, meta10m, meta60m = s2.sortBands(clippedS2)
s2.compositeBands(bands, meta10m, meta60m, compS2)
composites = s2.resampleBandsTo10m(compS2, overwrite = False)

Sorting images by band...
Sort complete.
Compositing bands...
B01 median composite created.
B02 median composite created.
B03 median composite created.
B04 median composite created.
B05 median composite created.
B08 median composite created.
B11 median composite created.
B12 median composite created.
Compositing complete.

Resampling 8 bands to 10m resolution using bilinear...
Resampled: C:\School\GIS\Project\tool\inputsS2\composites\B01_resampled_10m.tif
Resampled: C:\School\GIS\Project\tool\inputsS2\composites\B02_resampled_10m.tif
Resampled: C:\School\GIS\Project\tool\inputsS2\composites\B03_resampled_10m.tif
Resampled: C:\School\GIS\Project\tool\inputsS2\composites\B04_resampled_10m.tif
Resampled: C:\School\GIS\Project\tool\inputsS2\composites\B05_resampled_10m.tif
Resampled: C:\School\GIS\Project\tool\inputsS2\composites\B08_resampled_10m.tif
Resampled: C:\School\GIS\Project\tool\inputsS2\composites\B11_resampled_10m.tif
Resampled: C:\School\GIS\Project\tool\inputsS2\composites\B12_resampled_10m.tif
Resampling complete.
```

After running the tool, the images should start to process and follow the steps. It will first create composites and display these files in the folders we created not too long ago. If there is no output, please make sure to verify the file paths and also decompress the SHP file before using it.

2.6.7 Generate Composite Statistics & Display Composites

The tool calculates key statistics, such as mean and variance, over a grid defined by lineSpacing:

```
stats = s2.gridStats(bands, lineSpacing)
```

To visualize the composite bands:

```
s2.showBands(composites)
```

Composite statistics

```
[ ]: stats = s2.gridStats(bands, lineSpacing)
```



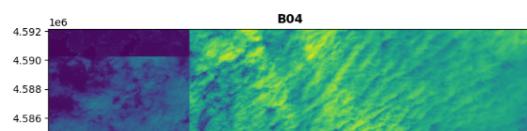
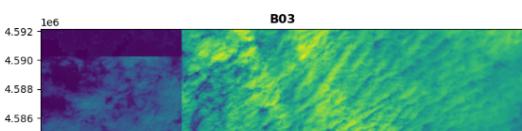
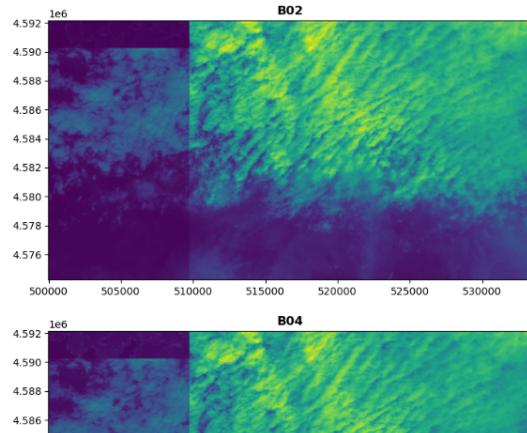
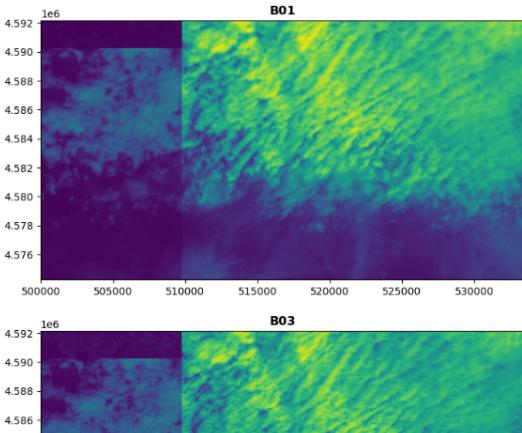
Display composites

Run the following code blocks to:

- Display the composite for each band of interest
- Display an RGB composite of the composites created
- Display the composite for a single band of interest

```
[8]: # display composite for each band of interest
s2.showBands(composites)
```

Further down in the tool, it will then get the composite statistics and also display these composites as you can see below

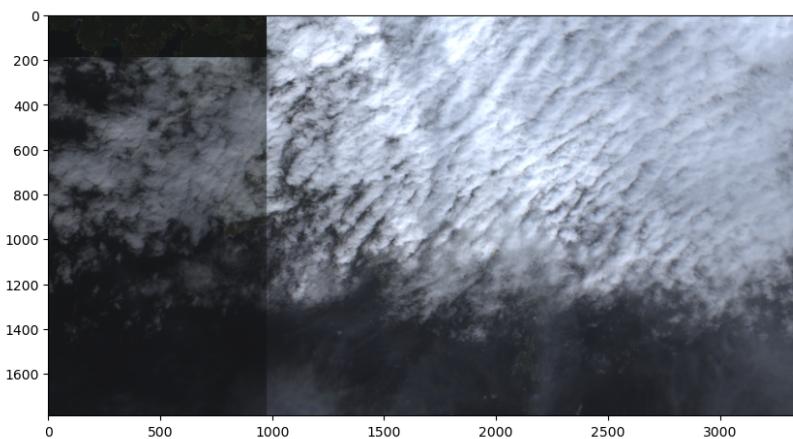


2.6.8 To generate an RGB composite:

```
R = "demo/S2/composites/B04_resampled_10m.tif"
G = "demo/S2/composites/B03_resampled_10m.tif"
B = "demo/S2/composites/B02_resampled_10m.tif"
s2.showRGB(R, G, B)
```

```
# display RGB composite
R = r"C:\School\GIS\Project\tool\inputsS2\composites\B04_resampled_10m.tif"
G = r"C:\School\GIS\Project\tool\inputsS2\composites\B03_resampled_10m.tif"
B = r"C:\School\GIS\Project\tool\inputsS2\composites\B02_resampled_10m.tif"
s2.showRGB(R, G, B)
```

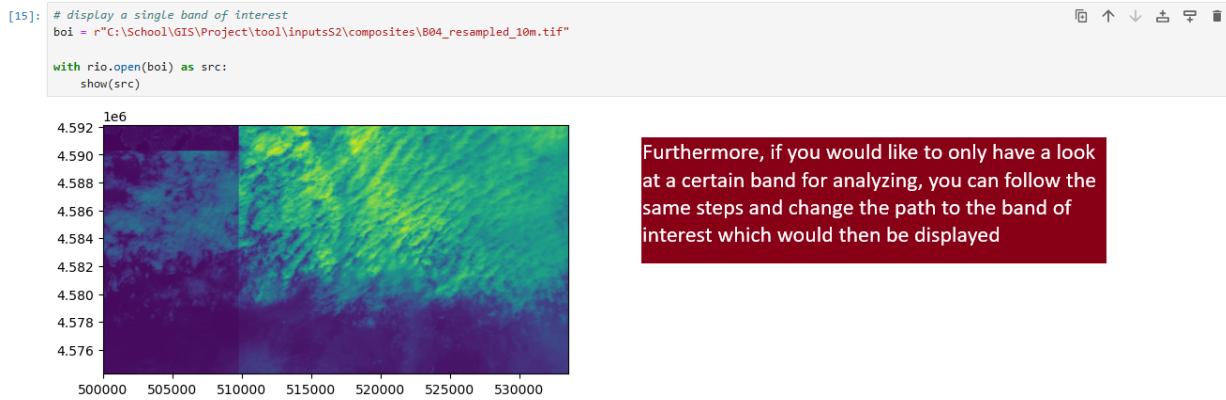
File paths can be inputted here which would then display the RGB composite



After all these steps are achieved, the major part of the tool is now completed and for further analysis you can display the RGB composite. For this step you will have to head into the "inputsS2" then into the "composites" subdirectory where you can find the bands needed which are (B04, B03 and B02). Make sure to have the file paths of the bands needed.

2.6.9 To display a single band:

```
boi = "demo/S2/composites/B01_resampled_10m.tif"
with rio.open(boi) as src:
    show(src)
```



2.6.10 Troubleshooting

2.6.11 Check Error Messages

- Review any error messages generated during the process to diagnose specific issues.

2.6.12 File Paths

- **Solution:** Ensure the paths for the Sentinel-2 data (`inputS2`) and AOI shapefile (`aoiShp`) are correct. Double-check for typos, extra spaces, and ensure you have the correct file extensions (`.SAFE` for Sentinel-2 and `.shp` for shapefiles).
- **Note:** Use double backslashes (`\\"`) or single forward slashes (`/`) for Windows paths.

2.6.13 Data Types

- **Solution:** Ensure input paths are pointing to the correct data types. Specifically, `.SAFE.zip` files for Sentinel-2 imagery and `.shp` files for shapefiles.

2.6.14 Storage

- **Solution:** Ensure your hard drive has adequate space to store output files, particularly the TIFF files generated by the tool.

2.6.15 Dependencies

- **Solution:** Ensure the tool is being used within the correct environment, specifically the `s2compo` conda environment. Do not modify the repo folder structure or the Sentinel-2 SAFE files.

2.6.16 Output Files Not Generated

- **Solution:** Verify that output directories are correctly set up and that they are writeable. If output files are not generated, check for errors during execution.
-

2.6.17 Further Assistance

If problems persist, please contact the project contributors or consult the **Credits** section for additional support.

2.6.18 Credits

- **Adriana Caswell, Christian Devey, and Muhammad Ba** for developing the tool, including the design, coding, and testing of the Sentinel-2 Compositing Tool. We contributed to the development of key features, troubleshooting, and documentation to ensure a seamless user experience.
 - **Dr. Derek Mueller** for providing valuable guidance throughout the project, offering feedback, and ensuring that the tool's functionalities align with the project's goals.
 - **Dr. Anders Knudby** for his support and insights into the project's direction, helping us refine the tool and incorporate best practices in geospatial analysis and compositing.
-

S2COMPOSITE MODULE

Functions for S2CompoTool.

s2composite.compositeBands(band_dict, meta10, meta20, meta60, out_folder)

Create a median value composite from band dictionary created using sortBands().

Author: Adriana Caswell

Parameters

band_dict

[dict] Dictionary of 3D arrays for each band of interest. Key corresponds to band of interest: B01, B02, B03, B04, B05, B08, B11, B12. Value is the 3D array containing masked arrays.

meta10

[dict] Metadata for 10 m bands.

meta20

[dict] Metadata for 20 m bands.

meta60

[dict] Metadata for 30 m bands.

out_folder

[str] Path to folder where composite TIFFs will be output.

s2composite.getStatsGrid(bandGroup, grid, cellSize)

Calculates and records stats from grid center points.

Author: Christian Devey

Parameters

bandGroup

[arr] 3D masked arrays, i.e. an array of Sentinel Bands

grid

[dict] Dictionary storing grid information

cellSize

[int] The resolution of each cell in the sample band in meters

Returns

stats

[dict] Dictionary With the collected stats. The Keys are “Mins”: The Minimum values found at each grid center point “Maxs”: The maximum values found at each grid center point “StDivs”: The Standard deviation of the values found at each grid center point “CoeffVars”: The coefficients of variation of the points around each center point “NumValidVals”: number of valid data points at the grid centers

s2composite.grabRegionStats(*imageLayer*, *x*, *y*, *rSize*)

Gets Coefficient of variation of cells surrounding the center cell. Outputs a float Coefficient of variation.

Author: Christian Devey

Parameters

imageLayer

[arr] 2D masked array, representing a Sentinel Bands

x

[int] The X coordinate of the center cell

y

[int] The Y coordinate of the center cell

rSize

[The width of the region]

Returns

cv

[float] The coefficient for variation of the cells examined

s2composite.gridStats(*sortedBands*, *lineSpace*)

Calculates and records stats from grid center points.

Author: Christian Devey

Parameters

sortedBands

[dict] Dictionary of 3D masked arrays, i.e. an array of Sentinel Bands

lineSpace

[int] The desired spacing in meters between grid lines

Returns

bandStats

[dict] Dictionary With the collected stats. The Keys are “Mins”: The Minimum values found at each grid center point “Maxs”: The maximum values found at each grid center point “StDivs”: The Standard deviation of the values found at each grid center point “CoeffVars”: The coefficients of variation of the points around each center point “NumValidVals”: number of valid data points at the grid centers

s2composite.makeGrid(*sampleBand*, *lineSpace*, *cellSize*)

Creates a Grid Over the Area of Interest. Outputs a Dictionary storing grid information.

Author: Christian Devey

Parameters

sampleBand

[arr] 3D masked array, i.e. an array of Sentinel Bands

lineSpace

[int] The desired spacing in meters between grid lines

cellSize

[int] The resolution of each cell in the sample band in meters

Returns

grid

[dict] Dictionary representing a grid over the area of interest. The Keys are “lineSpace”: The spacing in meters between lines “horzLns”: An array of the height of each horizontal line of the grid “vertLns”: An array of the height of each horizontal line of the grid “cPoinOffset”: The spacing between

`s2composite.prepS2(img_folder, shp_path, out_folder)`

Clips Sentinel-2 SAFE zip files to area of interest and outputs TIFF file.

Author: Adriana Caswell

Parameters**img_folder**

[str] Path to folder containing Sentinel-2 SAFE zip files.

shp_path

[str] Path to area of interest (polygon) Shapefile.

out_folder

[str] Path to folder where clipped TIFFs will be output.

`s2composite.resampleBandsTo10m(img_folder, out_folder=None, resampling_method='bilinear', overwrite=True)`

Resample specific Sentinel-2 bands to 10m resolution.

Author: Mumu Ba

Parameters**img_folder**

[str] Path to folder containing clipped Sentinel-2 TIFF files.

out_folder

[str, optional] Path to folder where resampled TIFFs will be saved. If None, resampled files will be saved in *img_folder* (default: None).

resampling_method

[str] Resampling method (default: ‘bilinear’).

overwrite

[bool] If True, overwrite original files; if False, save as new files with “_resampled” suffix.

Returns**resampled_files**

[list] List of resampled file paths.

`s2composite.showBands(imgs)`

Display 8 bands in one figure.

Author: Adriana Caswell

Parameters**imgs**

[list of str] List of paths to TIFF files.

Returns

None.

`s2composite.showRGB(red_band, green_band, blue_band)`

Combine red, green, and blue band TIFF files to display RGB composite.

Author: Adriana Caswell

Parameters

red_band

[str] Path to TIFF for red band.

green_band

[str] Path to TIFF for green band.

blue_band

[str] Path to TIFF for blue band.

Returns

None.

`s2composite.sortBands(img_folder)`

Sort Sentinel-2 images by band for bands of interest.

Author: Adriana Caswell

Parameters

img_folder

[str] Path to folder containing Sentinel-2 TIFF files clipped using prepS2().

Returns

band_dict

[dict] Dictionary of 3D arrays for each band of interest. Key corresponds to band of interest: B01, B02, B03, B04, B05, B08, B11, B12. Value is the 3D array containing masked arrays.

meta10

[dict] Metadata for 10 m bands.

meta20

[dict] Metadata for 20 m bands.

meta60

[dict] Metadata for 60 m bands.

PYTHON MODULE INDEX

S

s2composite, 15

INDEX

C

`compositeBands()` (*in module s2composite*), 15

G

`getStatsGrid()` (*in module s2composite*), 15

`grabRegionStats()` (*in module s2composite*), 16

`gridStats()` (*in module s2composite*), 16

M

`makeGrid()` (*in module s2composite*), 16

`module`

`s2composite`, 15

P

`prepS2()` (*in module s2composite*), 17

R

`resampleBandsTo10m()` (*in module s2composite*), 17

S

`s2composite`

`module`, 15

`showBands()` (*in module s2composite*), 17

`showRGB()` (*in module s2composite*), 17

`sortBands()` (*in module s2composite*), 18