Sentinel-2 Compositing Tool v1.0.0

GEOM 4009 Team Report
Adriana Caswell, Christian Devey, and Muhammad Ba
April 8, 2025

Introduction

Client background

Our client, Dr. Anders Knudby, is the Chair of the Department of Geography, Environment and Geomatics at the University of Ottawa. He is also a key member of the Shallow Water Earth Observation Lab (SWEO), which specializes in developing and applying remote sensing techniques to study coastal and aquatic environments. Dr. Knudby brings academic leadership and technical expertise to the project, offering guidance rooted in both geospatial science and applied environmental research. His work often involves interdisciplinary approaches that combine satellite data analysis with field validation and modeling techniques, particularly in the context of shallow water mapping.

Client requirements and priorities

Dr. Anders Knudby has emphasized the need for a robust and reliable compositing tool that can process large volumes of satellite imagery while minimizing the impact of visual obstructions. Key requirements include:

- The ability to process and integrate multiple satellite images across specific areas and timeframes.
- Automated filtering of unusable pixels affected by clouds, ice, foam, and sunlight.
- Improved image quality for more accurate bathymetric analysis in shallow waters (typically within 10–15 meters of depth).
- A streamlined and efficient tool that supports future research and scalability.

From the outset, the client has valued clear communication and collaboration. After an initial Zoom meeting to define the project scope and objectives, communication has continued via periodic email check-ins to share updates, address questions, and ensure the work remains aligned with expectations.

Purpose

The purpose of this project was to write a program that processes a collection of Sentinel 2 images of a specific area, and uses the collection to create a composite image and collect statistics from the images.

Scope

The scope of this project involved, clipping an image to the area of interest and converting it to the Tiff file format, creating a median composites from the tiff images, resampling the composite image to a resolution of 10m, recording descriptive statistics from the bands, and displaying the images.

Data

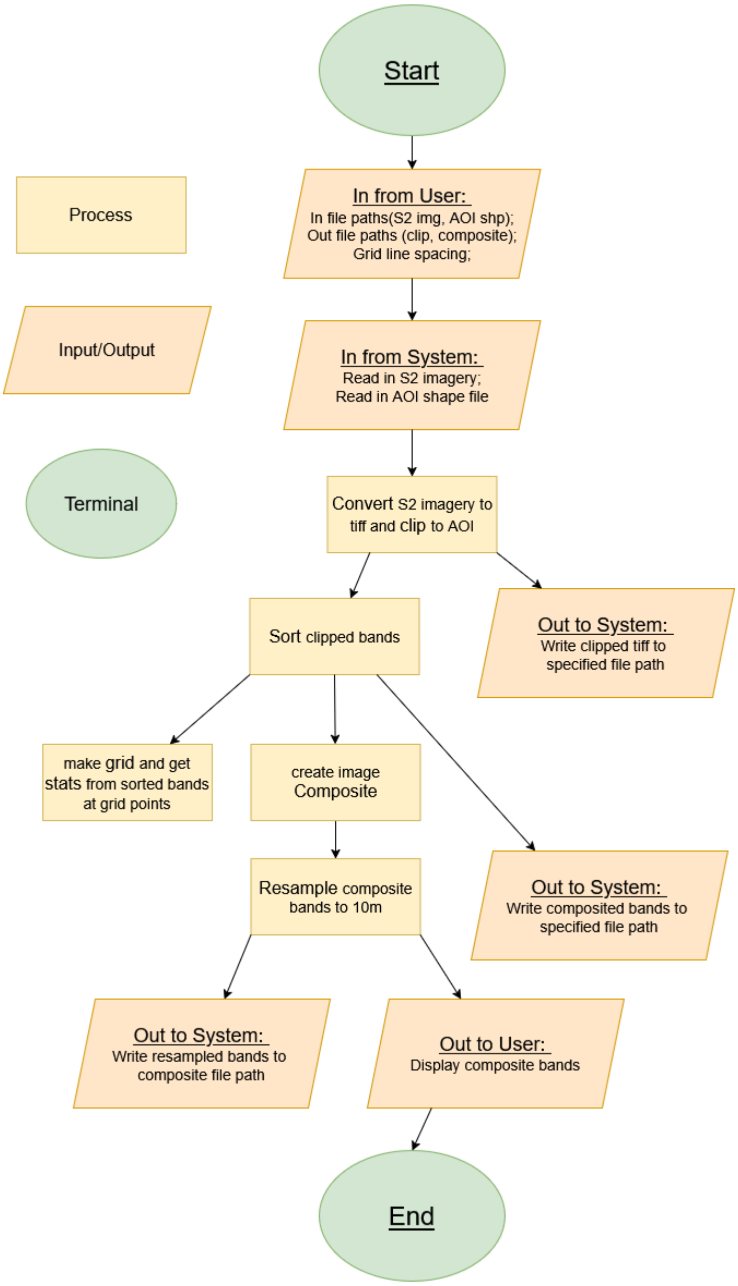
As the purpose is to create a composite of Sentinel-2 imagery for a specific area of interest, this tool requires a shapefile containing a polygon outlining the user's area of interest and zipped Sentinel-2 SAFE files covering the area. The number of SAFE files is determined by the user. The GitHub repository contains a demo folder with sample user input. More details can be found in the documentation.

Workflow

The workflow for the Sentinel-2 Compositing Tool is as follows:

1. Get file paths and other user input
2. Convert to TIFF and clip to area of interest (AOI)
3. Sort bands by band number
4. Create median composites for each band
5. Resample composites to 10 m
6. Get statistics related to the composite
7. Display the composites

This workflow is illustrated in the diagram below:



Documentation

The Sentinel-2 Compositing Tool documentation (created using Sphinx) can be found in PDF form attached to this report or online at geom4009.github.io/W25Project_Compositing/. This documentation outlines the tool's features, provides instructions for installation, outlines how to use the tool, and provides a demonstration/tutorial.

The Sentinel-2 Compositing Tool was designed to work on WindowsOS. It should also run on LinuxOS and MacOS but unexpected errors may occur. This tool consists of a Jupyter Notebook (S2CompoTool.py) which calls functions from a Python module (s2composite.py). The Python module has a number of dependencies which are managed using the s2compo conda environment which is available with the tool in the GitHub repository. A full list of the tool's dependencies can be found in the s2compo_env YAML. The key dependencies for this tool include:

- GDAL (Geospatial Data Abstraction Library) - a software library that reads and writes geospatial data used in this tool to read, clip, and convert the zipped Sentinel-2 SAFE files to GeoTIFFs and resample the composites to 10 meters
- Rasterio - a Python library for handling raster data used in this tool to read in GeoTIFFs as arrays and write arrays to GeoTIFFs
- NumPy - a Python library that provides a multidimensional array object and routines to handle the arrays used in this tool to compute statistics about to the input imagery and create the median value composite

Please refer to the Sentinel-2 Compositing Tool documentation for details about the installation and set up, a user guide which includes a tutorial and example, troubleshooting instructions, and documentation of the s2composite module.

Discussion

Challenges

While developing this tool, we encountered a variety of challenges which we were able to overcome. These included:

- **Large file size of Sentinel-2 imagery** → solved by clipping the image to the area of interest earlier in the workflow and using GDAL to create GeoTIFFs from the zipped Sentinel-2 SAFE files to avoid unzipping the files
- Clipping imagery resulting in **inconsistent dimensions and metadata** when imagery did not cover the entire area of interest → solved by using GDAL to clip the imagery which filled the excess space with a NoData value which we were able to control (NoData = 32000)
- **NoData values impacting calculations** for statistics and composite creation → solved by implementing NumPy masked arrays which masked NoData values so they would not be included in calculations.

- **Optimizing processing time** Balancing processing time became difficult, especially with large datasets. We opted for using a few select images and testing the tool to align the way. With the maximum images in each test being 10 which was 30 in total as aligned by the client.
- **Learning and adapting to new tools and libraries** Integrating new libraries and tools, such as GDAL, NumPy, and others, into the workflow posed a learning curve and required time for troubleshooting and optimization.
- **Writing functions with unknown input formats** When a function is being developed concurrently with its predecessor, the final output of the proceeding function may not be finalized, meaning the development of the subsequent function must be flexible to changing input formats.

Limitations and future work

The Sentinel-2 Compositing Tool allows users to create median value composites from Sentinel-2 imagery. However there are limitations which impact the user experience and the quality of the final composites. These limitations are outlined in this section along with suggestions for future work which can address these limitations.

Cloud masking: The tool does not include cloud masking and cloud cover negatively impacts the quality of the final composites. Users can adjust for this by selecting imagery that does not include significant cloud cover - either manually or by filtering the imagery by cloud cover in Copernicus Hub when downloading imagery. In the future, cloud masking could be incorporated into the workflow between s2prep and sortBands functions.

Imagery retrieval: Currently, users are required to retrieve Sentinel-2 imagery from the Copernicus Hub or by other means they have at their disposal. Future work could integrate imagery retrieval into the workflow using the Sentinel-2 API. This could be achieved using the Sentinel-2 API directly or by using the sentinelsat Python library.

Statistics display: The Sentinel-2 Compositing Tool calculates statistics about the input imagery that provides the user with information about the quality of the output composites. Currently, the statistics are output as a dictionary of arrays which can be challenging for the user to interpret. Future work should focus on improving how the statistics are displayed.

This could involve creating a GeoTIFF from the arrays containing information about statistics which could be visualized alongside the final composites.

Compositing methods: Currently, is only capable of creating median value composites. Future work could build upon this to create composites using a variety of methods - including using the mean or user specified quartile values which the client voiced interest in. This could be achieved by adapting the compositeBands function to include a method parameter or creating new functions.

Additional areas for future work include:

- More robust error handling
- Allow user to input a project name for naming output files
- Include a clean-up option to remove intermediate files
- Include visualization of intermediate files
- Allowing the user to select the bands of interest
- Improve existing functions by soft coding parameters where possible

Client interactions

We began the project with a Zoom meeting with Dr. Anders to discuss the scope, goals, and context of Satellite-Derived Bathymetry. He provided clear expectations and highlighted key challenges, such as the impact of image obstructions on depth accuracy. Since then, communication has continued through occasional email check-ins, where we've shared progress and received feedback. Dr. Anders has been responsive and supportive, helping guide our technical direction while allowing us the flexibility to explore solutions. His input has been valuable in keeping the project aligned with real-world research needs.

Questions asked included:

- Whether to use L1C or L2A Sentinel-2 imagery.
- How to present grid statistics (CSV, directly in the Jupyter Notebook, or another format).
- Preferred file format for the final images (TIFF, GeoTIFF, or others).
- Preferred method of accessing the tool.

Conclusion

The Sentinel-2 Compositing Tool is an open source tool that can be found in the GEOM4009/W25Project_Compositing GitHub Repository along with full documentation. The tool allows users to create median value composites resampled to 10 m for Sentinel-2 imagery and computes statistics about the composite providing users with information about the composites' quality. The tool uses the user interface of Jupyter Notebook to allow users to easily interact with the s2composite module. While the tool fulfils its primary purpose, there are still some improvements to be made to improve the quality of the output and the user interaction. Nonetheless, it provides a solid base for users to work with to create composites for Satellite Derived Bathymetry (SDB) or other purposes.

Acknowledgements

Thank you to Dr. Derek Mueller and Dr. Anders Knudby for initiating the project and providing guidance throughout its development.

We were inspired and guided by the following sources:

- Inspiration for median value composite methods: ahmad-geo-edu/median-composite and Dimo Dimov
- Clipping using GDAL: Geospatial Linux documentation by University of Edinburgh and LTS International
- Documentation from GDAL, Rasterio, Matplotlib, and NumPy
- Examples of resampling in geomatics and getting familiar with the concept resampleexamples
- Video inspiration that shows what GDAL can accomplish in the realm of resampling GDAL resampling
- Inspiration of resampling Sentinel 2 imagery and more Sentinel 2 resampling documentation
- ChatGPT

Repository files

The GEOM4009/W25Project_Compositing GitHub repository associated with this tool includes the following files:

- **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 Auto Documentation 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 Auto Documentation html files structured to create the repository's GitHub Pages.

Sentinel-2 Compositing Tool

Release 1.0.0

Adriana Caswell, Christian Devey, and Muhammad Ba

Apr 08, 2025

CONTENTS:

CHAPTER

ONE

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

CHAPTER

TWO

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 wiht 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

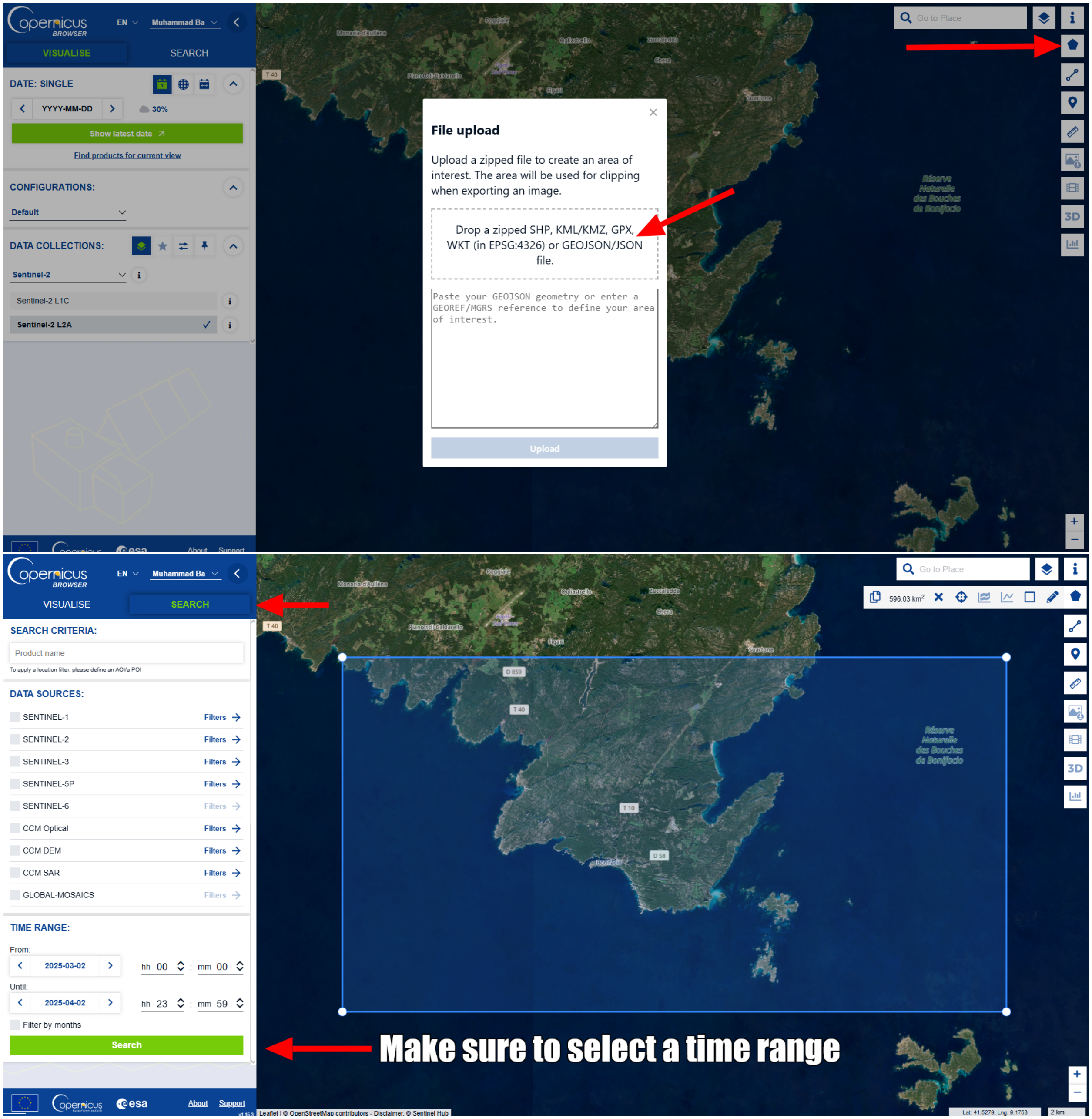
In the demo/S2 folder, you will find a text file that outlines the specific Sentinel-2 imagery used in the demo. This file provides detailed information on the imagery aligned with the study area, ensuring that the selected satellite data matches the area of interest for your analysis. The images have 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.

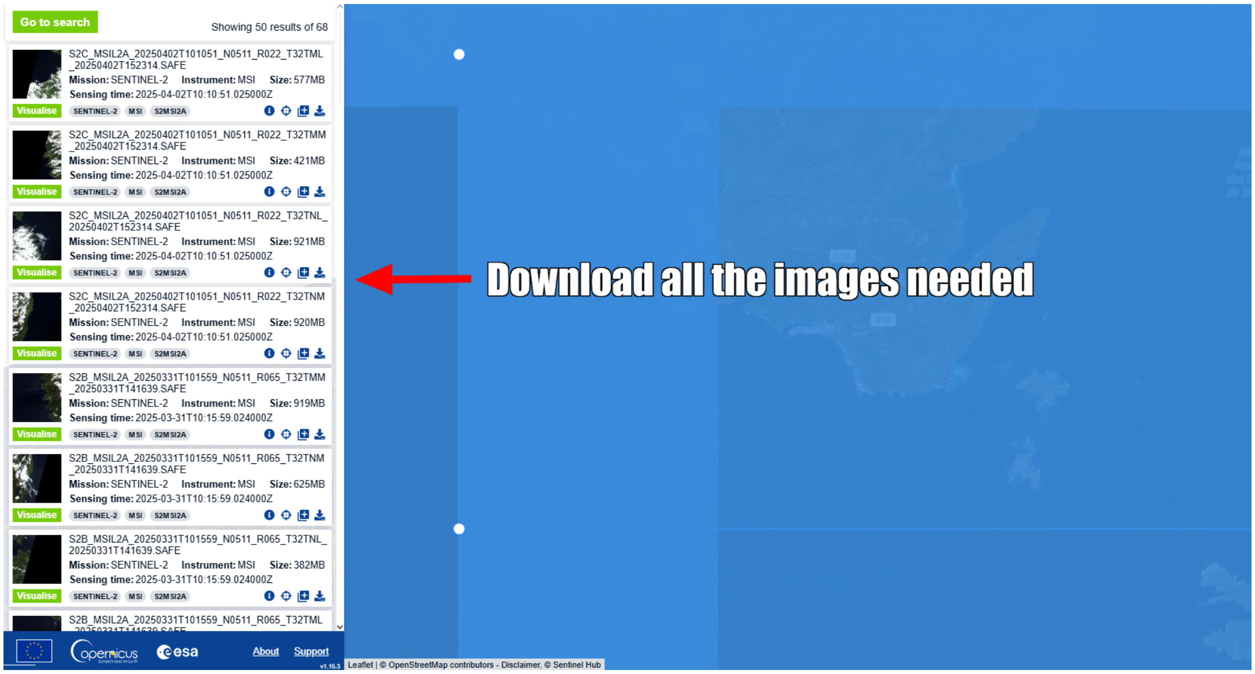
Additionally, the StudyArea.shp file is already included in this folder. This shapefile defines the Area of Interest (AOI) for the analysis, which is used to clip the Sentinel-2 imagery. Since the AOI is pre-configured to match the study area, you don't need to make any modifications. However, if you 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.

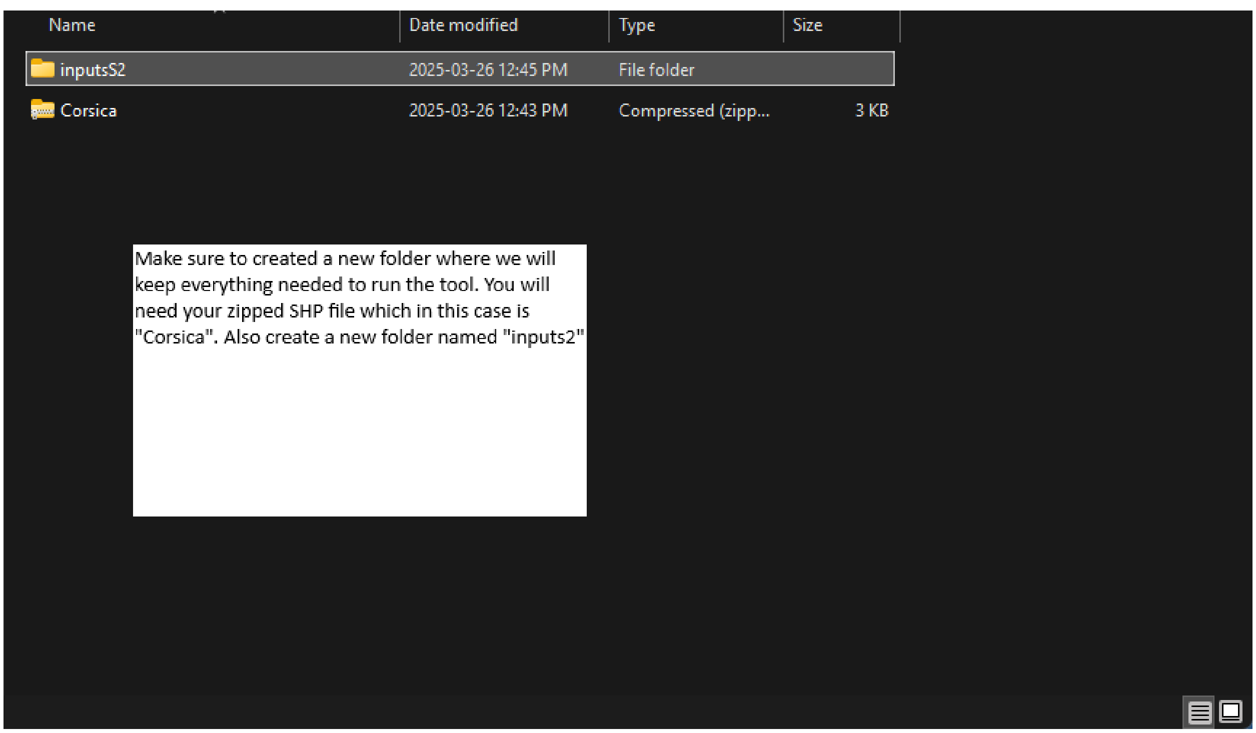

VISUALISE
SEARCH
DATE: SINGLE
YYYY-MM-DD
30%
Show latest date
Find products for current view
CONFIGURATIONS:
Default
DATA COLLECTIONS:
Sentinel-2
Sentinel-2 L1C
Sentinel-2 L2A
File upload
Upload a zipped file to create an area of interest. The area is used for clipping when exporting an image.
Drop a zipped SHP, KML/KMZ, GPX, WKT (in EPSG:4326) or GEOJSON/JSON file.
Paste your GEOJSON geometry or enter a GEOREF/MGRS reference to define your area of interest.
Upload
SEARCH CRITERIA:
Product name
DATA SOURCES:
SENTINEL-1
SENTINEL-2
SENTINEL-3
SENTINEL-5P
SENTINEL-6
CCM Optical
CCM DEM
CCM SAR
GLOBAL-MOSAICS
TIME RANGE:
From:
2025-03-02
Until:
2025-04-02
Filter by months
Search
Make sure to select a time range

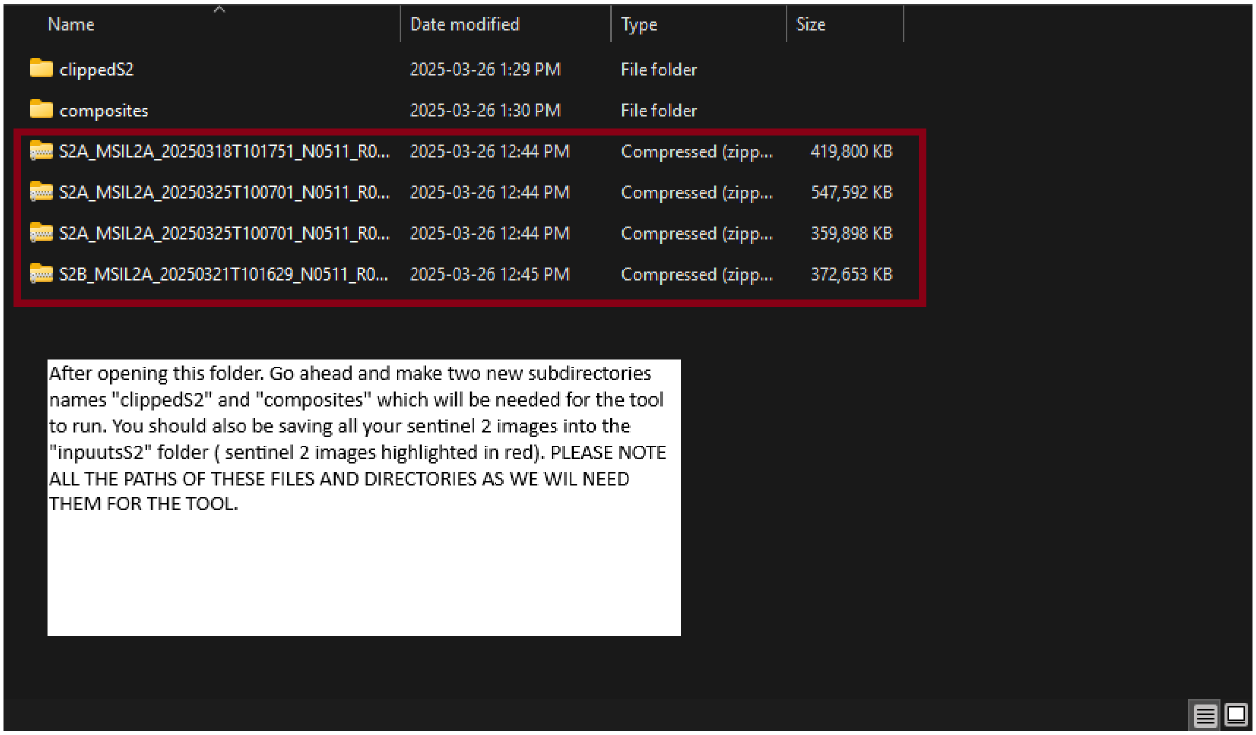


2.6 Getting Started

2.6.1 Set Up User Inputs

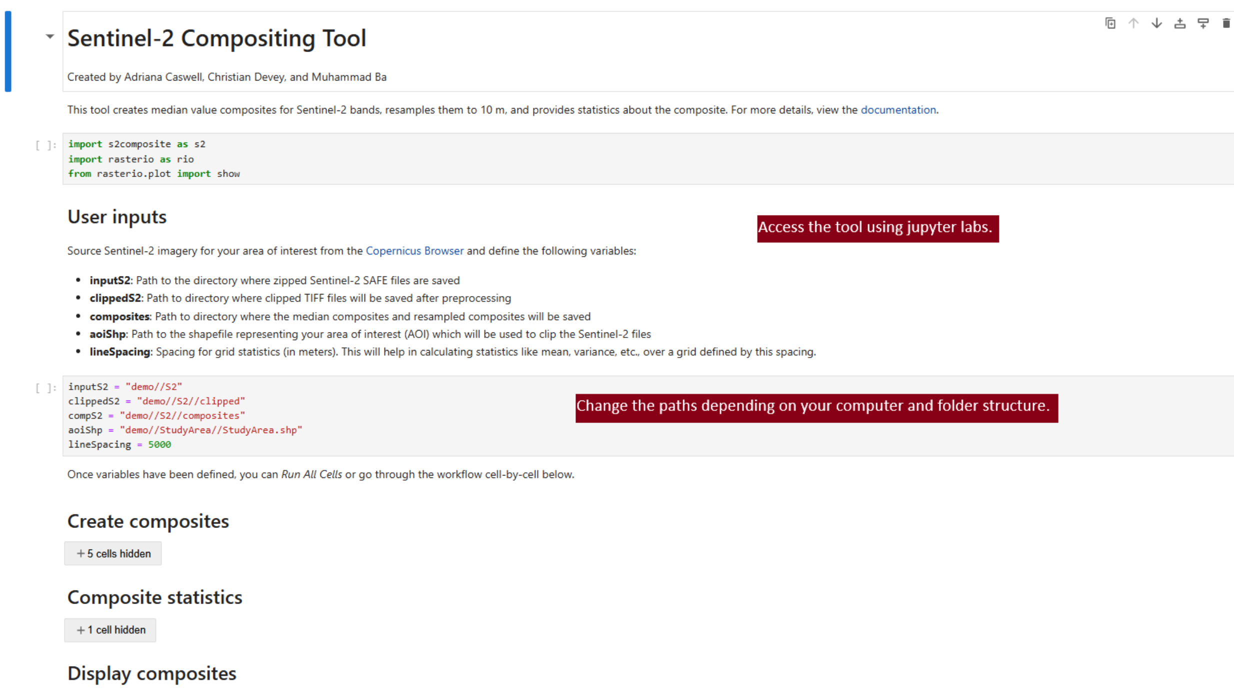
In this section, you will define the paths and other settings necessary for the tool to run properly. These variables control the location of your Sentinel-2 data, the clipped data, and where your output files will be saved. Make sure to adjust the paths according to your system and data storage locations.

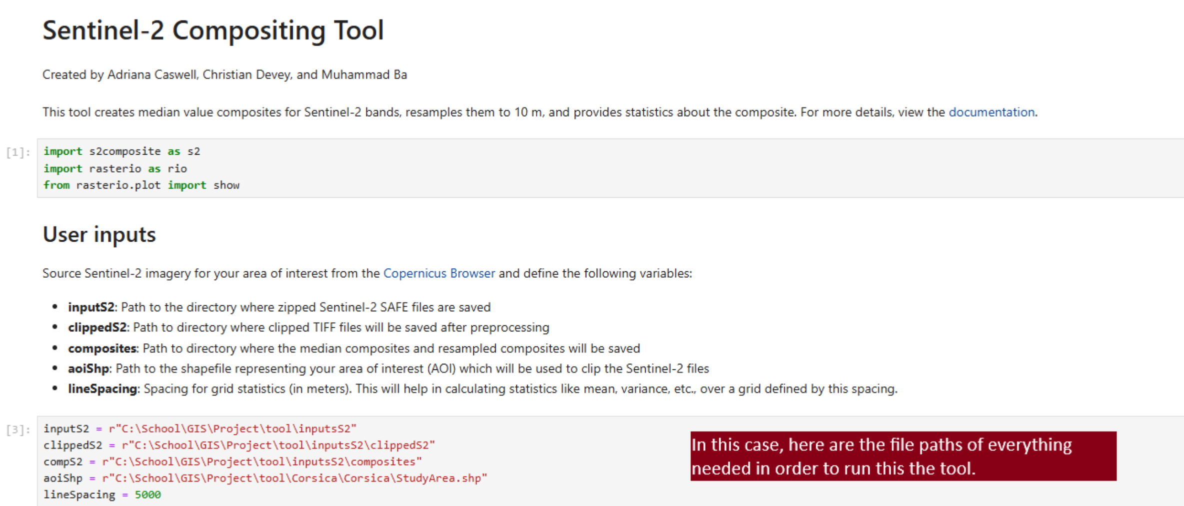




2.6.2 In the S2CompoTool.ipynb Notebook, define the necessary input variables:

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





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, meta20m, meta60m = s2.sortBands(clippedS2)
s2.compositeBands(bands, meta10m, meta20m, 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 out put, please make sure to verify the file paths and also decompress the SHP file before using it.

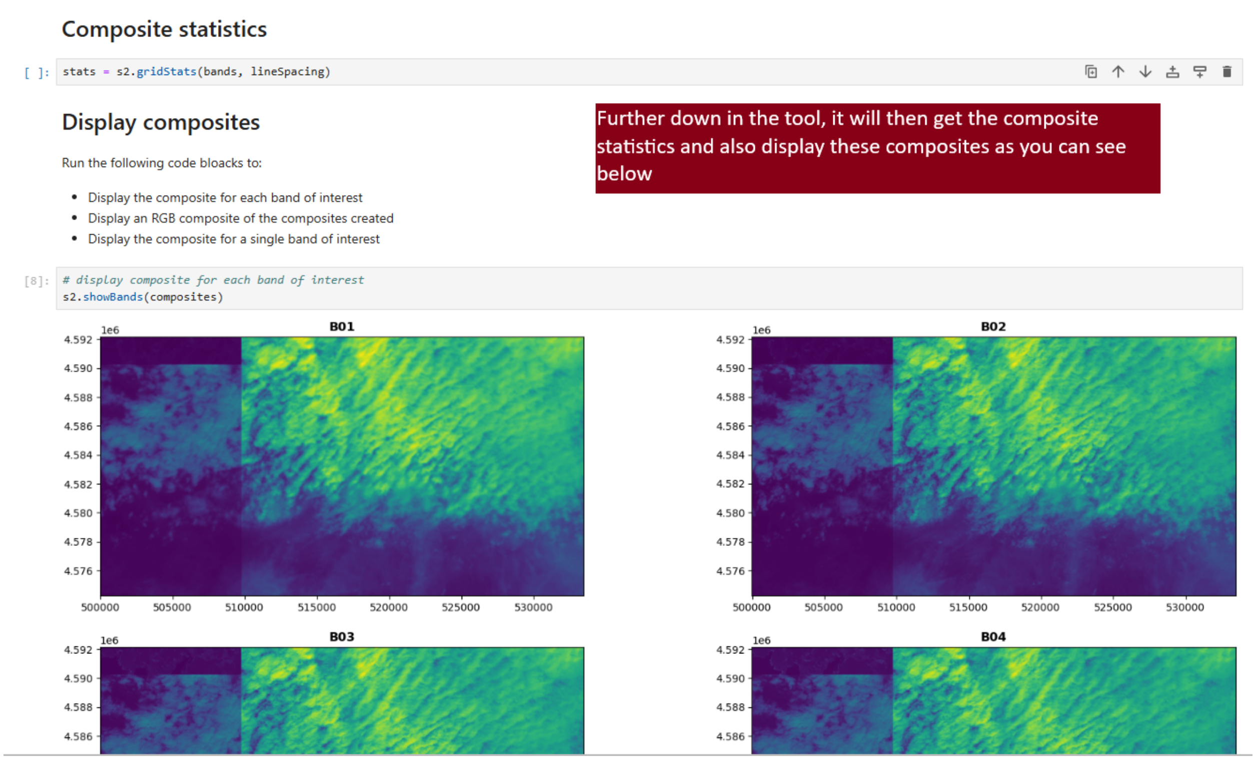
2.6.7 Generate Composite Statistics & Display Composites

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

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

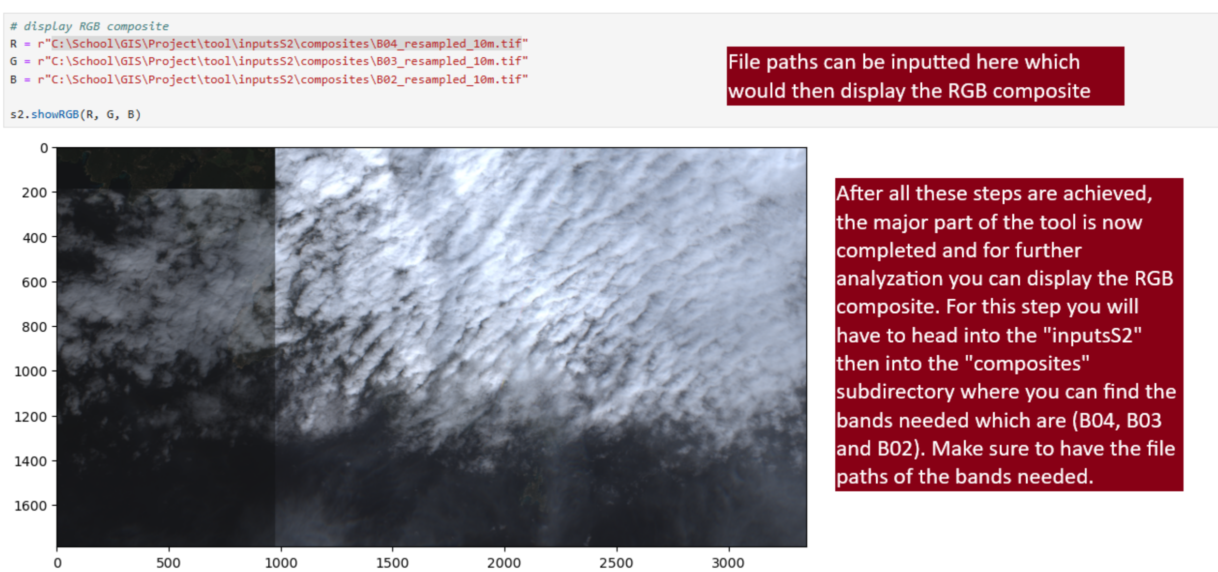
To visualize the composite bands:

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



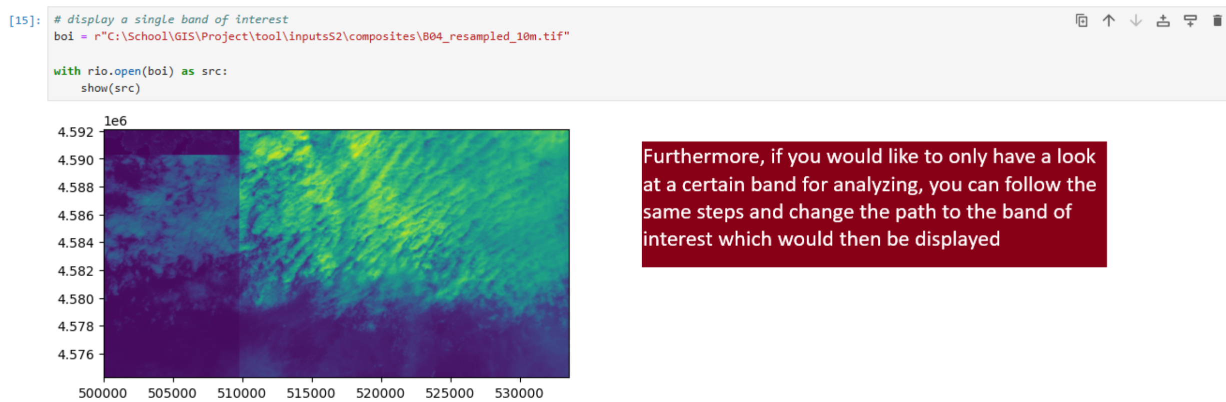
2.6.8 To generate an RGB composite:

```
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)
```



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.

CHAPTER

THREE

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 corrosponds 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] Dictionart 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 dieveation 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 Coefficiant of variation of cells surounding the center cell. Outputs a float Coefficiant 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 coeffician 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 dieveation 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 Intrest. 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 corrosponds 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

INDEX