

# IN24B-10 Planet Utilities: A Python Package for Efficient Processing and Analysis of PlanetScope Satellite Data



Ankur Kumar<sup>1</sup>, Aaron S Kaulfus<sup>2</sup> and Udaysankar Nair<sup>1</sup>

(1) University of Alabama in Huntsville, Huntsville, United States, (2) NASA Marshall Space Flight Center, Huntsville, United States

**What:** Planet Utilities is a Python package designed to simplify the processing of PlanetScope satellite data, offering comprehensive functions for metadata extraction, data reading, visualization, and supervised classification.

**Why:** Researchers often face the burdensome task of writing repetitive code to read and visualize PlanetScope datasets, lacking a common repository for streamlined access and analysis.

**How:** This package was created to address these challenges and is conveniently available on PyPI (Python Package Index), providing a user-friendly interface, extensive documentation, tutorials, and examples for easy integration into workflows.

## Getting Started with planet\_utils

To install the latest version of planetutils from PyPI (<https://pypi.org/project/planetutils/>), use the following command:

```
pip install planetutils
```

Alternatively, you can install the package from the source code by cloning the Git repository and using pip to install it in editable mode:

```
git clone https://github.com/NASA-IMPACT/planet_utils
cd planetutils
python3 setup.py install
```

The PlanetScope package can be installed with pip directly from the github:

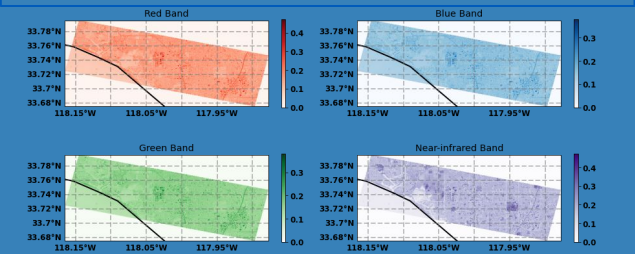
```
pip install git+https://github.com/NASA-IMPACT/planet_utils
```

## Modules and functions

Module	Function	Description
read	equalize	Applies histogram equalization to improve the contrast of an image band.
	get_coeffs	Extracts reflectance coefficients from a Planet Labs XML metadata file.
	get_extent	Extracts the bounding box (extent) from a Pandas DataFrame with longitude and latitude values.
	get_location	Extracts geographic location information from Planet Labs XML metadata files.
	get_mask	Generates a binary mask using a reference image, typically representing the Red band.
	get_reflectance	Calculates reflectance values from radiance data in a multi-band image file.
	get_rgb	Generates a true color composite (RGB) from a raster file.
plot	get_rgb_equalized_factors	Retrieves equalized factors for creating a true color composite (RGB) from a raster file.
	plot_coast	Plot natural features and gridlines on a map using Cartopy.
metadata	true_color_composite	Generates and displays a true color composite image using a Planet object.
	check_xmlfile_type	Check the type of input XML file and parse it if necessary.
	extract_element	Extract longitude and latitude coordinates for a specified corner from PlanetScope XML metadata.
	get_location	Extract geographic locations for the bounding box corners from PlanetScope XML metadata.
	get_xml_filename	Generate the metadata XML filename corresponding to a given TIFF filename.
	get_coeffs	Extract coefficients for converting digital numbers to reflectance from XML metadata.

## Reading and plotting reflectance

```
from planetutils import read
planet_filename = '20230910_175433_97_24c4_3B_AnalyticMS_8b_clip.tif'
planet = read.PlanetScope(planet_filename)
coeff = planet.coeff
coastalred = read.get_reflectance(planet, factor=20, band='Coastal Blue')
```



## Documentation

```
planetutils.metadata.get_xml_filename

get_xml_filename(tiff_filename) [source]

Generate the metadata XML filename corresponding to a given TIFF filename.

This function takes a TIFF filename and generates the corresponding metadata XML filename by replacing the ".tif" extension with "_metadata.xml". It's designed to work with PlanetScope imagery file naming conventions.

Args:
    tiff_filename (str): The name of the TIFF file for which you want to find the metadata XML file.

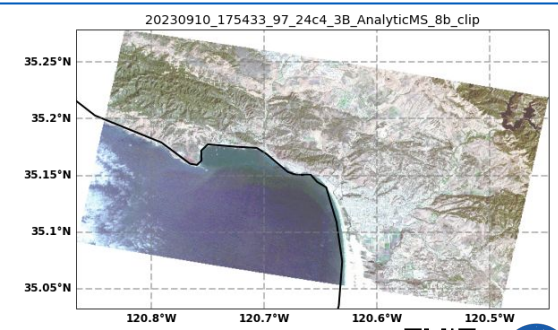
Returns:
    str: The metadata XML filename corresponding to the input TIFF filename.

Example:
    The following example demonstrates how to use the get_xml_filename function:

    tiff_filename = "your_image.tif"
    metadata_filename = get_xml_filename(tiff_filename)
    print(f"The metadata XML file for {tiff_filename} is: {metadata_filename}")
```

## Plotting true color composite

```
from planetutils import read
planet_filename = '20230910_175433_97_24c4_3B_AnalyticMS_8b_clip.tif'
planet = read.PlanetScope(planet_filename)
plot.true_color_composite(planet, factor=20)
```



Contact: [Ankur Kumar ankur.kumar@uah.edu](mailto:ankur.kumar@uah.edu)  
[Aaron S Kaulfus aaron.kaulfus@nasa.gov](mailto:Aaron S Kaulfus aaron.kaulfus@nasa.gov)

