

# CanoPy tutorial

CanoPy is the Python module for [the Georgia Canopy Analysis 2009 project](#) sponsored by [the Georgia Forestry Commission \(GFC\)](#). For further information about this project, please refer to the [CanoPy page](#).

This document explains detail steps for assessing canopy using the [CanoPy](#) module and a provided sample data set. To learn more about this module, please read the [CanoPy user manual](#). For technical details, please read the [CanoPy technical manual](#).

## Authors

- [Owen Smith](#)

## Requirements

- ArcGIS Desktop 10.x
- ArcPy
- Python 2 standard module: os

We are currently planning on developing a fully open source solution without using ArcGIS and Feature Analyst.

## Sample data set

canopy\_sample\_data.zip

The sample data will contain 3 folders titled 'naip', 'data', and 'Outputs'. The 'data' folder contains the Georgia physiographic regions shapefile, the NAIP Quarter Quad (QQ) polygon shapefile in addition to the TIFF file which will be used for snapping. The 'naip' folder will contain 4 input QQ NAIP tiles that form a 2 x 2 area. The 'Outputs' folder contains the outputs that are created by Textron's Feature Analysis software for each NAIP QQ in the sample dataset. This is included as this tutorial does NOT go over the process of using Textrons Feature Analyst but rather the process of using the CanoPy Python module.

## Steps

1. Either clone the repository using Git in the terminal with the below code

```
git clone https://github.com/HuidaeCho/canopy.git
```

OR visit the [CanoPy GitHub repository](#) and download the zip file.

- If you download the .zip file you will first need to unzip the file. It will be named 'canopy-master', rename this folder to 'canopy'.
- 2. Download the CanoPy sample data from this page and extract it from its .zip format.
- 3. Make a copy canopy\_config-example.py in the same folder and rename it to canopy\_config.py.
- 4. Edit the new canopy\_config.py with a text editor or python ide of your choice (ie. notepad, vim, pycharm) to recognize your layers and folder structures. In this case you will set the various paramters of canopy\_config.py as such.
  1. phyregs\_layer = 'Physiographic\_Districts\_GA'
  2. phyregs\_area\_sqkm\_field = 'AREA\_SQKM'
  3. naipqq\_layer = 'naip\_ga\_2009\_1m\_m4b'
  4. naipqq\_phyregs\_field = 'PHYREGS'
  5. naip\_path = '.../CanoPy\_sample\_data/naip'
  6. spatref\_wkid = 102039
  7. project\_path = '.../CanoPy\_sample\_data'
  8. analysis\_path\_format = '%s/%d Analysis' % project\_path
  9. analysis\_year = 2009
  10. analysis\_path = analysis\_path\_format % analysis\_year
  11. snaprast\_path = '%s/data/rm\_3408504\_nw\_16\_1\_20090824.tif' % project\_path
  - This will allow CanoPy to read the required data and create the required output paths for the data to be saved into.
- 5. Open a new ArcMap 10.x document.
- 6. Add the following files from the sample data folder to ArcMap.
  1. Physiographic\_Districts\_GA.shp
  2. naip\_ga\_2009\_1m\_m4b.shp
  - You may notice these are the phyregs\_layer and the naip\_qq layer from the canopy\_config.py file, however for them to properly be utilized by CanoPy the data must be in ArcMap for the required spatial functions to be used.
- 7. Open the Python window under the geoprocessing pane located on the top bar within ArcMap.
- 8. After the Python window is open you must import the CanoPy module for ArcMap to be able to use its functions. To accomplish this use the following code.

```
import sys
sys.path.append('C:/path/to/the/canopy/module')
import canopy
```

9. Most functions in CanoPy only contain one parameter, phyreg\_ids. This is the unique identifier for each physiographic region that enables the user to decide which region to process. In this case you will be using the id [3] which is the phyreg\_id of the Blue Ridge Mountains region. However, to first be able to use the id's you have to add the id's to the naipqq\_layer. To do this simply run the following code in the Python window.

```
canopy.assign_phyregs_to_naipqq()
```

- After it is finished the naipqq\_layer will now contain the naipqq\_phyregs\_field which will be titled 'PHYREGS' in this case. Each NAIP QQ will now have the region id's of each region it is in.
- The secondary purpose of canopy.assign\_phyregs\_to\_naipqq is to calculate the area in square kilometers of each physiographic region. The phyregs\_layer will now contain the phyregs\_area\_sqkm\_field which will be titled 'AREA\_SQKM'. The 'AREA\_SQKM' field will be used with the generate\_gtpoints function.

10. Next you will assign the `phyreg_ids`, which in our case will be:

```
phyreg_ids = [3]
```

- If more regions were able to process you would assign those regions as such:

```
phyreg_ids = [3, 7, 15]
```

11. Next you will reproject all input NAIP tiles to the projection specified by the `spatref_wkid` in `canopy_config` which will be [WKID-102039: USA Contiguous Albers Equal Area Conic USGS version](#). Use the following code to use the reprojection function

```
canopy.reproject_naip_tiles(phyreg_ids)
```

- Reprojected tiles with the suffix 'rm\_' will be saved to `.../CanoPy_sample_data/2009 Analysis/Blue_Ridge_Mountains/Inputs` which will have been created by the function.
12. The reprojected tiles in the Inputs folder will be used in as the inputs with Textrons Feature Analyst, however we are skipping that step, so you will want to move the `.../CanoPy_sample_data/Outputs/` folder to the `.../CanoPy_sample_data/2009 Analysis/Blue_Ridge_Mountains` folder. This is where the classified outputs of Textrons Feature Analyst would have been saved to be used with the rest of the CanoPy functions.
- The folder structures should now look like this:



13. After the 'Outputs' have been moved. The wrapper function can be used to complete the process with the following code.

```
canopy.convert_afe_to_canopy_tiff(phyreg_ids)
```

This wrapper function, titled `convert_afe_to_canopy_tiff` contains three individual functions. These functions are

1. `convert_afe_to_final_files`
    - Ensures all tiles are TIFF files with the values of 0 & 1. Files have prefix of 'fr\_'
  2. `clip_final_files`
    - Clips all tiles to the outline of their QQ to remove all edge artifacts (errors along edge of file) and make mosaicing more accurate. Files have prefix of 'cfr\_'.
  3. `mosaic_clipped_final_files`
    - Mosaics all clipped tiles to the shape of their physiographic region and clips the mosaic to the physiographic region boundaries.
14. After running the final wrapper function the outputs folder will contain all intermediate and final 2-bit output Canopy TIFFS.

From:  
<https://clawiki.isnew.info/> - **CLAWRIM Wiki**

Permanent link:  
<https://clawiki.isnew.info/software/canopy/tutorial?rev=1722746586>

Last update: **2024-08-03 10:43 pm**

