DirtSAT v2.1 readme

This readme document discusses how to run the “rooftop-index” v2.1 developed by DirtSAT.

The full code base requires manual changes to run, as the format of any individual client’s data (LiDAR) will take various forms (LAS vs tiled geoTIFF vs full, mosaiced geoTIFF) and obviate or require certain steps. The v2.1 code is broken into 4 major groups and are denoted by the prefix in the individual script names (for example 1_1-script.R, 1_2-script.R, 2_1-script.R, etc):

- Subgroup 1: These scripts are used to preprocess the DEM and DSM files to clip and mosaic rasters into appropriate domains. There are options for multi domain (for example NYC) and single domain (for example Miami).
- Subgroup 2: These scripts (1 R script, 1 JavaScript script) interact with Google Earth Engine (GEE) to pull down satellite derived data.
- Subgroup 3: This script is the workhorse script that serves 2 purposes, computes Flat Areas (FA) on each rooftop and assigns them unique IDs (together denoted as FAIDs) and computes the input dataset for the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), a multi-criteria decision analysis method that is used to compute the final rank.
- Subgroup 4: This script merges/filters separate TOPSIS input files and runs the TOPSIS algorithm to calculate the final rank and build the final HTML widget to display results.

Below describes each script within each subgroup to provide more detail. NOTE: rasters refer to gridded datasets such as digital elevation models (DEMs), digital surface models (DSMs), land surface temperature (LST) and the normalized difference vegetation index (NDVI - greenness). These rasters are manipulated using the terra and raster libraries in R. Vector datasets represent the footprint features. Vector datasets are manipulated using sf (simple features) library. General data manipulation follows tidy theory using the tidyverse (e.g. method chains using pipes “%>%”).

Subgroup 1:

- /R/1_1_parse_multidomain_based_on_external_index.R - When the footprint domain is too large to process all at once, we will need to break down the domain into manageable chunks (tiles). For example, there are 1,082,349 buildings in New York City (NYC), much too many to run all at once. Therefore this footprint vector needs to be clipped into smaller domains (tiles). The size of these tiles will ultimately depend on the compute environment and available resources. Generally the random access memory (RAM) of the machine will be the determining factor. Compute resources used for the NYC analysis are 32 threads (Xeon Silver 4110), with 128Gb RAM. This script breaks down the footprint file into smaller chunks. For the NYC example, we break down the domain using the index system used to clip the DEM and DSM returns provided by the city. However, one can generate any index system that suits the purpose (redefine index_file variable). THIS SCRIPT IS OPTIONAL, IF THE DOMAIN DOES NOT NEED TO BE BROKEN DOWN (E.G. MIAMI), THEN MOVE ON TO SCRIPT 1_4 USING THE

ORIGINAL FOOTPRINT DOMAIN. Note: All current naming conventions follow NYC (001, 002, 003, etc).

- /R/1_2_merge_dsm_dem_gdal.R - This script merges (mosaics) DEM and DSM tiles into a single file. This is important to do, even with pre- tiled domains such as with NYC, due to edge effects. For example, footprint domains may not be fully covered by pre- tiled rasters (e.g. there may not be full coverage for each rooftop). So far, this has been done with both the NYC and Miami runs. These rasters will then be clipped into either multiple domains (such as in 1_3 - using the tiled footprints in 1_1) or a single domain (1_4).

- /R/1_3-clip_rasters_multidomain.R - Use this script to break down the full merged rasters to match the tiled footprints extents output from 1_1. This script simply clips the rasters based on the extent of the tiled footprints. THIS SCRIPT IS ONLY REQUIRED IF 1_1 IS REQUIRED FOR COMPUTATIONAL LIMITATIONS.

- /R/1_4-clip_rasters_single_domain.R - This script clips the larger domain into a single domain matching an “original” footprint. THIS SCRIPT IS USED IF 1_1 IS NOT REQUIRED.

Summary, if multi-domains are required run 1_1, 1_2, and 1_3, if the entire domain can be run at once, run 1_2 (if original DEM and DSMs are tiled) and 1_4.

Subgroup 2:

- /R/2_1-export_ndvi_sentinel2.R - This script accesses GEE via R using the rgee package in R. Specifically, this script uses the extent of the merged raster from 1_2 to extract summertime (June - September) NDVI (or greeness) from the Sentinel 2 satellite. NOTE: Running GEE using the python API in R requires setup, this can be done with manuals online and will require differences depending on operating system. Contact Hoylman for help with Linux environments. GEE REQUIRES CREDENTIALS.

- /js/2_2-lst_gee.js - This script computes land surface temperature using GEE using the Java Script code editor. Complex expressions are used in this script and hard to emulate in rgee. Here, draw a domain big enough for the analysis, it does not need to be precise. GEE REQUIRES CREDENTIALS.

Subgroup 3:

- /R/3_compute_FAID_and_TOPSIS_input.R - This is the workforce script that computes FAIDs for each rooftop and footprint tile.FAIDs are areas within building footprints that are flat and greater than 5000 sf. To find contiguous flat areas, a raster of building height is smoothed with a gaussian filter to remove any pits, and a raster of slope is masked to 1 if slope is less than 45 degrees and 0 if slope is greater than 45 degrees. These two new rasters are multiplied and filtered to remove pixels that are smaller than 5 feet. The raster is then polygonized using GDAL and intersected with the building footprint vector to create a vector of contiguous flat areas within building footprints. Finally, this new

vector is filtered to remove flat areas smaller than 5000 square feet. Then using these FAIDs and footprints we compute all of the required input for the TOPSIS algorithm; FAID_ave_slope, ave_parapet_height, FAID_flat_area_ft2, load_volume, FAID_height_above_ground, FAID_under_100ft, FAID_ave_ndvi and FAID_ave_lst.

Subgroup 4:

- /R/4_merge_data_and_run_TOPSIS.R - This is the final script. This script merges the tiled TOPSIS input geojson files from 3_1, provides methods to filer larger domains by areas of interest (such as CUNY), compute the TOPSIS rank and generate interactive maps in HTML widget form.

Extra considerations:

Coordinate reference systems (CRS)s: Area calculations require equal area projections such as EPSG:5070 - NAD83 / Conus Albers, or more local coordinate systems. It is typical that LiDAR data is projected in an equal area projection, so it is valid to use the original coordinate system of the LAS files / processed DEM/DSM. Regardless of which coordinate system is chosen for the domain, it is critical to make sure CRS match across raster and vector datasets.

Spatial resolution:

As of now, only 1ft spatial resolution rasters have been used. Courser resolution data can be used, but may require alterations.