Africa Biomass Challenge – Dataset Preparation

Yuchang Jiang, Alexander Becker

December 2022

1 Introduction

This technical report includes three parts:

1. Preprocess raw data, including Sentinel-2 and GEDI;
2. Combine GEDI and Sentinel-2 data to generate training and validation dataset (Section 2.3);
3. Combine GIZ biomass and Sentinel-2 data to generate a test dataset (Section 2.4);

2 Dataset preparation

2.1 Sentinel-2

The preprocessing of Sentinel-2 images includes downloading the L1C projects with external server and processing them to L2A projects with the official toolbox.

Download Sentinel-2 L1C products We query and download L1C products from an external server, peps (https://peps.cnes.fr/rocket/#/home) via a Python package, eodag (https://github.com/CS-SI/eodag). With a list of Sentinel-2 tile names, we query available products based on geoloations, cloud threshold (20), and date (leaf-on season of year 2020). We also use a threshold to filter out products with few valid pixels (only keep products with at least 50% valid pixels). We download the least cloudy image among all the responded products for each tile. As all old products (older than half a year) stored on the server are offline products so requesting them to be online is a necessary step before downloading them, which may take a long time (8 minutes).

Listing 1: Code to query Sentinel-2 L1C products

```
longitude_min, latitude_min, longitude_max, latitude_max = s2_tile['geom'].bounds
search_results, total_count = dag.search(
    productType='S2_MSI_L1C',
    geom={'lonmin': longitude_min, 'latmin': latitude_min,
```

```
    'lonmax': longitude_max, 'latmax': latitude_max},
    cloudCover=20,
    start='2020-05-01',
    end='2020-09-30',
)
```

Convert L1C to L2A projects We used the offical toolbox, Sen2Cor (https://step.esa.int/main/snap-supported-plugins/sen2cor/sen2cor-v2-9/) to generate L2A products. After setting up this toolbox, we use the command $Sen2Cor-02.09.00-Linux64/bin/L2A]\_Process --output\_dir \$MY\_OUTPUT\_DIR \$MY\_L1C\_FILE$. The expected data format of L1C file is in SAFE format and the output L2A file is also in SAFE format. We convert the L2A product in SAFE format into zip format for further processing.

2.2 GEDI

For GEDI preprocessing, we first reduce the size of raw data by throwing out non-useful variables, then perform simple filtering and combine results into one single file. Then we split the processed file into smaller files based on the Sentinel-2 tiles to enable further parallel processing.

Reduce the data size The original biomass products contain many different variables (see the full list of all variables here: [1]). We only keep the variables of interest: "agbd", "agbd_se", "degrade_flag", "l4_quality_flag", "lat_lowestmode", "lon_lowestmode", "selected_algorithm", "predict_stratum", "shot_number", "sensitivity". The reduced files are saved in the same format (HDF5) as the original files.

Filter and combine all GEDI files into one single file We need to filter out noisy GEDI data and also combine the large number of files into one single file. The filtering is based on the following rule, we only keep the data that satisfy:

- L4 quality flag is 1
- Not based on the selected algorithm 10
- biomass value is non-negative
- there exists valid geolocations for the captured biomass value
- data are captured by the full power beams only

Then we combine the filtered data in HDF5 format into one single CSV file.

Divide the single file based on Sentinel-2 tiles To help the parallel processing in later stage, we need to split the large CSV file into smaller CSV files based on Sentinel-2 tiles. We loop through all Sentinel-2 products downloaded, check its geolocation boundaries, filter out the GEDI data captured inside this

[1] https://daac.ornl.gov/GEDI/guides/GEDI_L4A_AGB_Density.html

geolocation boundary, and same them in a new small file with the Sentinel-2 tile name as file name.

2.3 Combination of GEDI and Sentinel-2 images

At this stage, we have the Sentinel-2 L2A products in zip format and processed GEDI data in CSV format. Both Sentinel-2 and GEDI data are organized based on the Sentinel-2 tiles. Now we need to co-register the two data sources, crop the large image into patches and save co-registered patches into HDF5 files. To speed up with parallel processing, we run the same task for each tile: we find the L2A product and GEDI CSV file correponded to a certain tile name and match together. The detailed process is as the following, we input L2A product and GEDI CSV file of a certain tile and output patched dataset with "agbd", "agbd_se" from GEDI, all channels from Sentinel-2 products, and latitude, longitude:

1. We define some parameters: the size of the resulted image patch should be $15 * 15$.

2. We convert the latitude, longitude of biomass data into x, y pixel location corresponded to Sentinel-2 images. We treat the resulted pixel location (pixel with a valid biomass value) as a center pixel and check if the image patch of size $15 * 15$ can fall inside the Sentinel-2 image fully. If yes, we define this biomass data point as valid target point. If multiple biomass value fall into the same pixel, we take the average.

3. We rasterize the valid target point into the same format of Sentinel-2 product so the resulted output are raster images with the same width, length, coordinate system as the Sentinel-2 data. This rasterization is performed for each variable independently, including "agbd", "agbd_se", latitude, longitude.

4. Now we have raster images for all variables and Sentinel-2 images so we can crop the image into $15 * 15$ patches. Instead of looping through the whole image, we only take the patch when the center pixel is a valid target point. Patches from different variables are saved in one HDF5 file.

In the end, we get HDF5 files containing patches of data. The number of the output HDF5 files should be the same as the number of Sentinel-2 tiles used as the output HDF5 files are organized based on Sentinel-2 tiles. We re-organize them by variable name so the number of the re-organized HDF5 files should be the same as the number of variables ("agbd", "agbd_se", Sentinel-2 products, latitude, longitude). Then, we filter the resulting dataset such that all resulting datapoints

1. are geographically located within the country of Ivory Coast

2. have a center pixel with cloud probability ≤ 10 %

3. have a center pixel that contains cocoa according to ETH Zurich's cocoa map[2]

Using a breadth-first search, we group all data points such that there is no inter-group geographical overlap. Finally, we split the dataset based on this group membership into train, validation and test sets in a 70:15:15 ratio.

2.4 Combination of GIZ biomass data and Sentinel-2 images

The given GIZ biomass data (20220912_GIZ_biomasse.xlsx) covers four Sentinel-2 tiles (29NNH, 29NPH, 30NVN, 30NVM). We perform the same preprocessing method as described in the Section 2.3 (replace the GEDI CSV file with the GIZ biomass data) and produce the test dataset in HDF5 format. The resulted test dataset includes biomass and corresponding Sentinel-2 images (bands, cloud cover, semantic class label), vegetation height, and geolocations. Notice that we loosen the cloud cover threshold from 20% to 50% in order to include more data in the cloudy test sites. The overview of GIZ biomass data and corresponding Sentinel-2 tiles are shown in Figure 1.

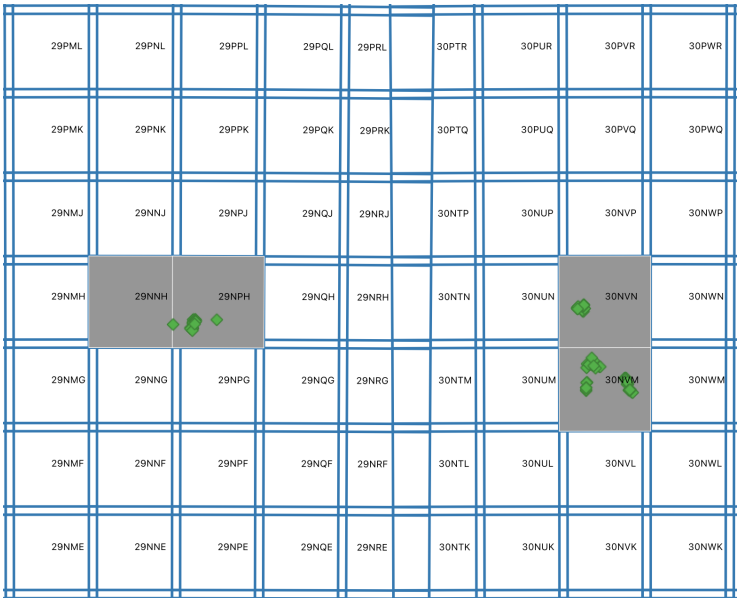

Figure 1: Visualization of GIZ biomass data in test site. Blue grids are Sentinl-2 tiles and gray ones are Sentinel-2 tiles that contain valid GIZ biomass samples. Green diamonds are GIZ biomass samples. In the end, 89 out of 90 biomass samples are used.

[2]https://nk.users.earthengine.app/view/cocoa-map