Biomapper User Manual

04/25/2022

Version 1.0

Team Biosphere

Project Sponsors: Dr. Christopher Doughty, Jenna M. Keany,

Team Mentor: Melissa D. Rose

Team Members: Matthew Nemmer, Brandon Warman, Teng Ao, D’Yanni Bigham

Table of Contents

Introduction

We are pleased that you have chosen Biomapper for your business needs. Biomapper is a powerful system for ecologists in Africa to have access to lidar data that has been custom-designed to meet your needs. Some of the key highlights include:

- Data access to canopy height models, above ground biomass, and digital elevation models.
- Offline mode use for when the user doesn't have a cellular network connection.
- French translation of the app, so French speaking ecologists can also use the app.

The purpose of this user manual is to help you, the client, successfully install, administer, and maintain the Biomapper product in your actual business context going forward. Our aim is to make sure you are able to benefit from our product for many years to come!

Installation

Android App Development Set up

The Biomapper Android App was developed using Android Studio, which is an easy-to-use IDE for Android development.
https://developer.android.com/studio/install

1. fd
2.

Server Creation

There might come a time when the project's web-server (currently an AWS EC2 instance) needs to be restarted. Note that before removing the current server instance, one must create a copy of the dataset tiles to be moved to the new server. If the datasets are lost, they will have to be re-created and this is a very time consuming process.

1. Create the Server Instance

How to set up an AWS EC2 instance: (packages you need and cloning the GitHub)
<link>

After setting up a new AWS instance you should be able to use the “biosphere_key.pem” to SSH into the server. Open terminal and change directories to where you have saved the pem file. Run the following commands to begin logging in to the server.

```
> chmod 600 biosphere_key.pem
> ssh -i biosphere_key.pem <SERVER IP>
```

Once successfully logged into the server we will want to make any necessary updates by running the following commands

```
> sudo yum update -y
```

2. Set up HTTPD requests

Setting up the web server is done through HTTPD and can be installed and run using the following commands.

```
> sudo yum install -y httpd
> sudo systemctl start httpd
> sudo chown -R $USER:$USER /var/www
```

We can now test to see if the application is running by creating a basic index.html file to be displayed on the website.

```
> cd /var/www/html/
> sudo nano index.html
```

Make a simple index.html file and then try accessing it by the server's ip address.

3. Install Node JS

Node JS is how our mobile app is able to make API requests to the Python script. We will configure and setup the API in the following steps.

```
> sudo yum install git -y
> git clone https://github.com/Biomapper/Server.git
> curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
> nvm install node
```

Node should now be successfully installed. You can test that it is running and which version by running the following.

```
> node -e “console.log(‘Running Node.js’ + process.version)”
```

4. Install Python

In this step we will install the filtering script and install any required dependencies.

```
> git clone https://github.com/Biomapper/PythonFilter.git
> python3 -m pip install --upgrade pip
> python3 -m pip install --upgrade Pillow
```

5. Run Node JS application

Now that we have installed all necessary dependencies we can run our node server.

```
> screen node app.js
```

You can test if the filter is working by accessing the api through your web browser. You will need to know the server IP address and can enter the following url.

6. Transfer Dataset Tiles to Server

The datasets need to be added to the server before they can be used. Assuming a copy of them was made from the previous server, they should be accessible and able to be transferred.

The file structure of the datasets should match the following:
http://<IP Address>/map-tiles/<Dataset>/<zoom>/<x-coor>/<y-coor>.png

Once the dataset tiles are on the server, the following URLs should work.
Unfiltered Tile
http://13.59.201.133/map-tiles/chm/6/34/31.png
Filtered Tile
http://13.59.201.133:3000/base-tiles/chm/6/34/31/300/600
Value from Point
http://18.223.3.121:3000/val/chm/6/34/31/140/140/40

7. Modify Mobile App to Use the Correct Server IP Address

Assuming the server IP has changed (which it very likely has), the mobile app needs to be updated to reflect these changes so the correct server will be used. Modify the app's build.gradle file and modify the <name for object> so that it contains the new IP address.

<screenshot of build.gradle file>

Note:
This step is only necessary because the IP address is currently hard-coded within the app. Purchasing a domain name for the server and using it in the app instead would be ideal. This is because the domain name is automatically associated with the correct IP address, even after it changes.

Dataset Tiling Software

A machine that includes GDAL libraries, QGIS, or some other geospatial toolset that's capable of creating map tiles is required for dataset modifications and additions. This manual uses a linux virtual machine for this purpose for performance reasons, but it might already be possible to tile datasets on your personal computer's host operating system.

1. Determine Device for Tiling

Alternative:
In order to tile datasets quickly, a large amount of processing power and storage is needed. A high-end PC or a supercomputer (like Monsoon) is recommended over a laptop

or basic PC. Another possibility to consider is trying to tile the datasets directly on Biomapper's server. This avoids the need to transfer the tiles to the server, but requires a more powerful machine than the one currently being used.

The team decided tiling datasets on a Linux machine is easier
For steps on how to install Linux Mint as a VirtualBox virtual machine, see this link:
https://itsfoss.com/install-linux-mint-in-virtualbox/

Assuming you are using VirtualBox and a guest virtual machine, you'll likely want to add Guest Additions. This allows one to transfer files from their desktop to their linux virtual machine by drag-and-drop and copy-paste.
https://www.virtualbox.org/manual/UserManual.html#additions-linux
Here are the steps:

1. Insert VBoxGuestAdditions.iso

Insert the VBoxGuestAdditions.iso "CD" file into your Linux guest's virtual CD-ROM drive. This can be done by starting the virtual machine, selecting the tab "Devices" at the top of the window, and selection "Insert Guest Additions CD image" and selecting VBoxGuestAdditions.iso

2. Run the Additions Disk

Change to the directory where your CD-ROM drive is mounted. It will likely open up after step 1. and run the following command as root:

```
sudo sh ./VBoxLinuxAdditions.run
```

Alternative:
You can also transfer tiles from your host device to your guest device (VM) via email or other online file transfer methods. However, these are slower and often more tedious than drag-and-drop and copy/paste.

2. Install Tiling Software

Source:
https://mothergeo-py.readthedocs.io/en/latest/development/how-to/gdal-ubuntu-pkg.html
In this example, we will be installing GDAL libraries on a Linux Mint machine.

1. Check Python Version

First, one must check that at least Python 3.6 is installed. This link explains in more detail how to install GDAL for Ubuntu (and also worked for Linux Mint).
Here are the commands.
Check that your Python is at least version 3.6 by using this command:

```
python3 --version
```

If it's not 3.6 or higher, run the following:

```
sudo apt-get install python3.6-dev
```

2. Get GDAL Libraries

To get the latest GDAL/OGR version, add the PPA to your sources, then install the gdal-bin package (this should automatically grab any necessary dependencies, including at least the relevant libgdal version).

```
sudo add-apt-repository ppa:ubuntugis/ppa
```

Once you add the repository, go ahead and update your source packages.

```
sudo apt-get update
```

Now you should be able to install the GDAL/OGR package.

```
sudo apt-get install gdal-bin
```

To verify the installation, you can run ogrinfo --version.

```
ogrinfo --version
```

You will need the GDAL version to install the correct python bindings.

3. Install GDAL for Python

Before installing the GDAL Python libraries, you'll need to install the GDAL development libraries.

```
sudo apt-get install libgdal-dev
```

You'll also need to export a couple of environment variables for the compiler.

```
export CPLUS_INCLUDE_PATH=/usr/include/gdal
export C_INCLUDE_PATH=/usr/include/gdal
```

'pip' is required for installing GDAL. To get pip:

```
sudo apt install python3-pip
```

Now you can use pip to install the Python GDAL bindings.

```
pip install GDAL==<GDAL VERSION FROM OGRINFO>
```

Configuration and Daily Operation

Regular User Guide

For information on how to use the application (as opposed to maintain it), see the online Biomapper User Guide:
biomapper.github.io

{ ***<To be added to user guide website:>***
As of initial release, Biomapper datasets include full coverage for more than half of all African countries. This includes all countries with significant tropical regions.

Full Coverage:
Guinea-Bissau, Guinea, Sierra Leone, Liberia, Cote D' Ivoire (Ivory Coast), Ghana, Equatorial Guinea, Gabon, Central African Republic, Congo, Democratic Republic Congo, Uganda, Rwanda, Burundi, Tanzania, Angola, Zambia, Malawai, Zimbabwe, Mozambique, Comoros, Madagascar

Near-full Coverage:
The Gambia, Togo, Benin, Cameroon, South Sudan, Kenya

Partial Coverage:
Senegal, Mali, Burkina Faso, Niger, Nigeria, Chad, Sudan, Ethiopia, Somalia
}

This website was created instead of writing the user guide here for a couple of reasons. First, a website is a very user-friendly format that can easily be shared to other users. No files need to be obtained. Instead, end users just need to be given the link to the User Guide. Additionally, this prevents typical end users from accessing the sensitive information contained in this document. Only developers and the product's clients need to know how to create a server instance, configure it, and modify the app datasets.

Accessing the Server

Connect to the server:

Set full permission for the security login file (only needs to be done once)	`chmod 600 biosphere_key.pem`
ssh into the server	`ssh -i biosphere_key.pem`

	`ec2-user@13.59.201.133`
Exit an ssh session	`~.`

Navigating the Server

Get full path to the current directory	`pwd`
List all items in the current directory -a shows hidden files/folders -l shows files/folders metadata	`ls [-a] [-l]`
Change current directory to PATH/DIR NAME	`cd <PATH/DIR NAME>`
Navigate to a new directory	`cd <DIR NAME>`
Navigate up a directory	`cd ..`
Navigate to the root directory	`cd /`
Navigate to the home directory	`cd ~`

File Managment

General	
Create a new directory/folder	`newdir <DIR NAME>`
Move a directory or file(s) to a new directory	`mv <DIR/FILE NAME> <NEW DIR/FILE NAME>`
Rename a file or directory	`mv <DIR/FILE NAME> <NEW NAME>`
Checking Size	
Check size of EC2 file systems	`df -h`
Check size of EC2 volumes	`lsblk`
Check size of a directory	`sudo du -sh <DIR NAME>`
Check size of all directories and files in the current directory	`sudo du -sh *`

Deleting	
Check specifically what files will be deleted by the command below	`find . -name "*.<FILE EXTENSION>" -type f`
Delete all files of a specific type from dir and all subdirs	`find . -name "*.<FILE EXTENSION>" -type f -delete`
Remove a directory and all of its contents	`rm -rI <DIR NAME>`

Maintenance

Adding a new Dataset to Earth Engine

Link to Earth Engine Repo:
https://code.earthengine.google.com/?accept_repo=users/mattnemmer1/BiomapperData
Link to copy of Earth Engine repo in GitHub:

Navigate to the MapDataset script on Biomapper's Earth Engine repository. Then:

1. For a dataset from within the Earth Engine Catalog, skip to the next step. If you're adding your own dataset to Earth Engine, import it as an asset.
2. Add the dataset as an image using the path to it
3. Create visualization parameters, specifying the min and max values
4. Add the dataset to the map
5. Create an export task for that dataset

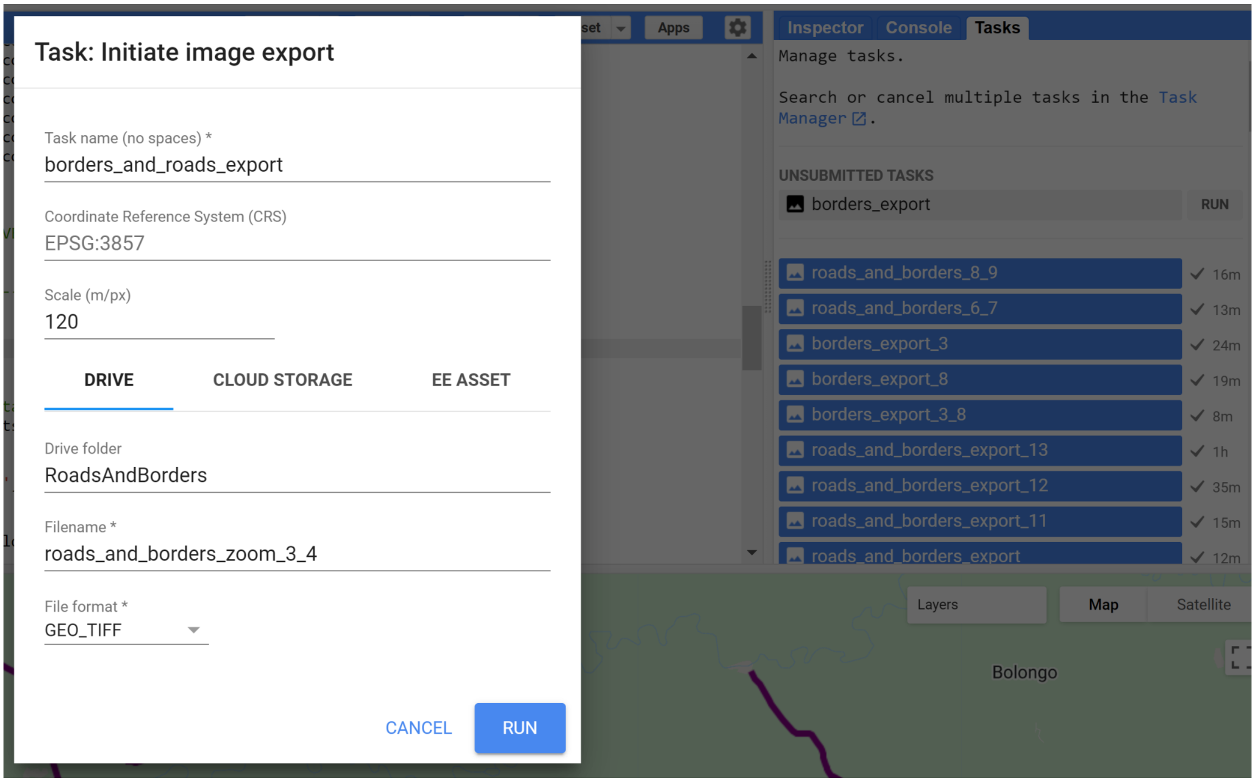
Adding or Updating Datasets to Biomapper App

This section explains how a dataset should be tiled so that it can be used by the Biomapper application.

1. Export GeoTIFFs from Google Earth Engine

Link to Earth Engine Repo:
https://code.earthengine.google.com/?accept_repo=users/mattnemmer1/BiomapperData

https://github.com/Biomapper/Google-Earth-Engine


Task: Initiate image export

Task name (no spaces) *
borders_and_roads_export

Coordinate Reference System (CRS)
EPSG:3857

Scale (m/px)
120

Drive folder
RoadsAndBorders

Filename *
roads_and_borders_zoom_3_4

File format *
GEO_TIFF

Manage tasks.

Search or cancel multiple tasks in the Task Manager.

UNSUBMITTED TASKS

borders_export RUN

roads_and_borders_8_9 ✓ 16m
roads_and_borders_6_7 ✓ 13m
borders_export_3 ✓ 24m
borders_export_8 ✓ 19m
borders_export_3_8 ✓ 8m
roads_and_borders_export_13 ✓ 1h
roads_and_borders_export_12 ✓ 35m
roads_and_borders_export_11 ✓ 15m
roads_and_borders_export ✓ 12m

Image of the export window and the task list.

Note:
It is recommended that you:

- Leave the coordinate reference system field blank.
- Set the export resolution to be the same as the dataset being exported, unless you:
    - Don't have a don't have enough processing power or storage for high-resolution tiling.
    - Are trying to save on server storage, which costs money.
    - Are seeing how rasterized vector data looks at different resolutions.
- Provide a name for a Google Drive folder to keep things organized since large exports result in multiple files being created.
- Provide a descriptive file name.

2. Combine exported GeoTIFFs

For large datasets with high resolution, there will be multiple GeoTIFFs exported from Google Earth Engine. To combine them, use "gdalbuildvrt" from GDAL or "Build Virtual Raster" from QGIS. This section assumed the use of "gdalbuildvrt." These essentially create a VRT file that tells the geospatial software *how* to treat the many GeoTIFFs as one large file. It does not actually combine them into one large image.

After navigating to the directory containing all GeoTIFFs to be combined, run the following command:

```
gdalbuildvrt combined-raster.vrt *.tif
```

A description of the command and its options can be found here:
https://gdal.org/programs/gdalbuildvrt.html

If using QGIS' "Build Virtual Raster", information can be found here:
https://docs.qgis.org/3.22/en/docs/user_manual/processing_algs/gdal/rastermiscellaneous.html#build-virtual-raster

The following is only recommended for small datasets:

```
gdal_merge.py -o combined-raster.tif *.tif
```

A description of the command and its options can be found here:

3. Tile the GeoTIFFs

Alternative:
You can also use the software MapTiler or a QGIS plugin called QTiles. Google Earth Engine also includes an option for exporting map tiles to Google Cloud. Note that MapTiler and Google Cloud are not free.

Map Tiler:
https://www.maptiler.com/desktop/
QTiles:
https://www.qgistutorials.com/en/docs/creating_basemaps_with_qtiles.html
Google Earth Engine export to Google Cloud:
https://developers.google.com/earth-engine/guides/exporting#exporting-maps

Using the previously created VRT file, use a tiling tool to create XYZ format map tiles from this for zoom levels 3 to 13 (including 3 and 13).

If using "gdal2tiles.py", run the following commands:

```
export GDAL_PAM_ENABLED=NO
```

```
gdal2tiles.py --profile=mercator --resampling=near --xyz
--zoom=3-13 --srcnodata=0 --exclude --processes=4
--tilesize=256 combined-raster.vrt chm
```

A description of the commands and options can be found here:
https://gdal.org/programs/gdal2tiles.html

3. Transfer Tiles to the Server

Transfer a directory containing the dataset you tiled to the server:

```
tar -c dir-to-transfer | ssh -i biosphere_key.pem
ec2-user@13.59.201.133 'tar -xf - -C /home/ec2-user'
```

This is recommended over an alternative file transfer method (like using ‘scp’) because it is significantly faster.

SSH into the server:

```
ssh -i biosphere_key.pem ec2-user@13.59.201.133
```

Move the tiles to the appropriate directory from home/ec2-user
(current directory after ssh):

```
mv dir-to-move ../../var/www/html/map-tiles
```

This will move the tiles from where they’re located after transferring to a directory that is accessible by the mobile application.

Change to the directory of map tiles from home/ec2-user

```
cd ../../var/www/html/map-tiles/agb
```

Note:
If you are updating a dataset rather than adding a new one, it is recommended that you rename the outdated one to something different before moving the updated tiles to var/www/html/map-tiles. This is better than deleting or replacing the old data immediately since it allows you to test the new data first.

Make sure to take note of the name of the dataset directory. This is important as these names are used within the mobile application’s code. As of writing this document, the directory names for the current datasets are “chm”, “dem”, “agb”, and “roads-and-borders”.

Updating Node JS Server

If there is ever a time you need to make changes to the API code while the server is running, the following commands will help. The active server will need to be temporarily stopped once changes are ready to be made and then restarted. Make sure you are in the “Server” directory and run the following commands:

```
screen -r
(CTRL + C)
screen node app.js
```

Trouble-shooting

Alternative:
If you experience any of the following errors on your virtual machine, you might find it's easier to simply recreate your VM entirely. First make a copy of all your important data from the VM, remove that VM and its virtual disk, and go again through the steps of installing and configuring a VM.

A copy of a starter virtual disk copy can be found at:
https://github.com/Biomapper/Other-Components

Expanding Virtual Machine Storage

If you found that your virtual machine needs more storage space for tiling, the following link describes how to switch between fixed and dynamically sized disks:
https://www.howtogeek.com/312456/how-to-convert-between-fixed-and-dynamic-disks-in-virtualbox/
This is necessary if you chose a fixed sized disk as their storage cannot be increased. It must be switched to a dynamically sized disk first.

Errors While Tiling a Dataset

There are a number of errors you might run into while tiling. In general, the following steps might help:

- Free up as much storage as possible. Not having enough space for tiles you're trying to create will cause an error.
- Tile only a few zoom levels at a time. Although tiling all zoom levels at once is possible (and relatively fast for low resolution datasets), you might find it's easier to tile a few at a time.
- Consider using a different, more powerful system for tiling.
- If using gdal2tiles, example in-browser map links are created for you to view the tiles you just created. Some no longer work, but the team found that the "openlayers.html" link worked well.

Conclusion

We hope you have found the user manual helpful in your ecology research in Africa! Team Biosphere has enjoyed working on this product and were excited to help make GEDI-derived data easier to work with for field researchers. Team Biosphere thanks Dr. Chris Doughty and Jenna Keany for all of their support in the development of this product and we are to see how it will develop with future generations of capstone developers.