Handover Note

Name: FABIO LANA **Index number: 3065101**

Job Title: GIS Specialist

Date of Handover Note: 14th July 2017

Duration of Assignment: 1st February to July 14, 2017

Brief Description of Duties:
TOR

- Develop remote-sensing modules using Google Earth Engine
  March 2017
- Development, support, maintenance of applications built on Google Earth Engine
  August 2017
- Documentation and knowledge sharing on good practices using Google Earth Engine
  December 2017

Supervisor and reporting procedures:
Mr. Sergio Bogazzi

Regular/re-occurring meetings, reports or procedures:

- SCRUM stand-up meeting

Key Documents/reference material to read (attach when possible):

- Python code documentation
  - https://sdlc.fao.org/bitbucket/projects/WAPOR2/repos/wapor-algorithm/browse/docs/_build/html
- Code repository
  - https://sdlc.fao.org/bitbucket/projects/WAPOR2/repos/wapor-algorithm/browse?at=refs%2Fheads%2Fagbp_removed

Status of recent and current projects/reports/meetings:

1. **Automatic upload of data**
   - ***Status***
     Completed using a personal gmail account using the geebam approach but the new corporate account has not been tested for uploading.
   - ***Action needed***
     Replace the uploading with personal account with the corporate account. I am afraid that this is not possible and to use the corporate account is necessary to buy some space on Google Cloud Storage.
     The corporate account has no Google Drive attached as far as I know so exporting is only possible if the file is stored in a GEE Asset of **AGAIN** Cloud Storage.
   - ***Critical issues/challenges/priorities***
     IMHO FAO needs to purchase a proper account with Google and some space in Cloud Storage. The “geebam” approach is not sufficiently solid for uploading large dataset because the software can break anytime. A Cloud Storage will solve all the problem for uploading massive datasets and most important remove the necessity to use Google Drive for downloading voluminous data produced within GEE.
2. **Code conversion from JavaScript to Python**
   - ***Status***
     Most of the algorithms prototyped in JavaScript have been ported in Python. The only algorithm that has not been completely converted is the NET Water Productivity.

- ***Action needed***
  Complete the conversion of the NET Water Productivity if necessary.

- ***Critical issues/challenges/priorities***
  Create a GitHub repository for sharing the Python code.

3. Uploading the new L2 level files.

- ***Status***

All the dataset available until July 14th were loaded.
Namely:

- L2_EANE_AET (288 files)
- L2_WANE_AET (288 files)
- L2_EANE_PHE (42 files)
- L2_WANE_PHE (42 files)
- L2_WANE_NPP (288 files)
- L2_EANE_TFRAC (288 files)
- PROBAV_2015_LC_100m_crop_occurrence

- ***Action needed***

Complete the uploading.
Namely:

- L2_EANE_PHE
- L2_EANE_NPP
- L2_WANE_TFRAC

- ***Critical issues/challenges/priorities***

Several datasets are not completed. Livia is awaiting the release of the missing data.
To date, 14th of July, we have used more that 70% of the space on GEE.

NOTES.

At the moment, I am using a modified version of geebam for uploading the data and I have written some python scripts for generating comma delimited files containing the metadata.
The modified version, stored in Bitbucket, solves some issues of the original code.
These modifications have also been incorporated in the original version available in GitHub so now there are only minor differences between my version and the original version.
Metadata are generated using filenames so is sufficient to provide the path of the directory containing the files to be uploaded and the metadata.csv is generated automatically.
I have copied several versions of the metadata generator under the directory snippets in Bitbucket. The naming conventions of the files provided by FRAME are heterogeneous (see following table) and this can cause problems when generating metadata. The latest version named metadata_generator3.py can generate metadata for seasonal data extracting the season (1 or 2) and the time in the season (end, maximum or start).

L1_AET_0906.tif	Dekad L1
L1_PCP_090403.tif	Daily L1
L2_EANE_PHE_09s1_e.tif	Seasonal L2
L2_WANE_AET_0910.tif	Dekadal L2 for West Africa

Documentation for all the code developed is provided using the “de facto” standard Python documentation package Sphinx in https://sdlc.fao.org/bitbucket/projects/WAPOR2/repos/wapor-algorithm/browse/docs/_build/html

The following information are provided solely for non-python developer and could be inaccurate or outdated.

Water Productivity Program Parameters.

usage: wpMain.py **timeframe [-e {u,d,t}] [-i] [-a ['agbp', 'aet', 't_frac', 'wp_gb', 'wp_nb']] [-s] [-v] [-h]**

Arguments:
---

timeframe Calculate Water Productivity Annually for the chosen period and can be provided as a single year or two dates (e.g. 2015 or 2015-01-01 2015-01-31)

-h show this help message and exit
-x, --export, choices=['u', 'd', 't'],
help="Choose export to url(-u), drive (-d) or asset (-t)")

-i, --map_id help="Generate map id for the chosen dataset"

-s, --arealstat help="Zonal statistics chosen country and for the chosen dataset"

-o, --output choices=['csv', 'json'],
help="Choose format fo the annual statistics csv(-o 'csv') or json (-o 'json')"

-a, --aggregation choices=['agbp', 'aet', 't_frac', 'wp_gb', 'wp_nb'],
help="which dataset must be calculated between the chosen timeframe"

-v, --verbose, help="Increase output verbosity"

Only available to developers for testing purposes not available on server
---
-m --map, choices=['agbp', 'aet', 't_frac', 'wp_gb', 'wp_nb']
help="Show calculated output overlaid on Google Map"

###Example 1
* Calculate Gross Biomass Water Productivity between 1st of January 2015 and 30th of January 2015
* Statistics calculated for Benin
* Generate map ID

wpMain.py 2015-1-1 2015-1-30 -a wp_gb -i -s "Benin"

###Example 2
* Calculate Gross Biomass Water Productivity for 2015 (already stored from 2010-2016)
Will be possible to calculate new dataset only from 2017 onwards
* Generate map ID

wpMain.py 2015 -a wp_gb -i

###Example 3
* Calculate Gross Biomass Water Productivity between first and 31st of January 2012
(Note the position of the parameter -i -s)
* Generate map ID
* Calculate statistics for watershed named 'Rift Valley'

wpMain.py 2012-01-01 2012-01-31 -a wp_gb -i -s w 'Rift Valley'

Water Productivity Data Management
Uploading or updating data in GEE. Valid credentials (a gmail account) must be provided.

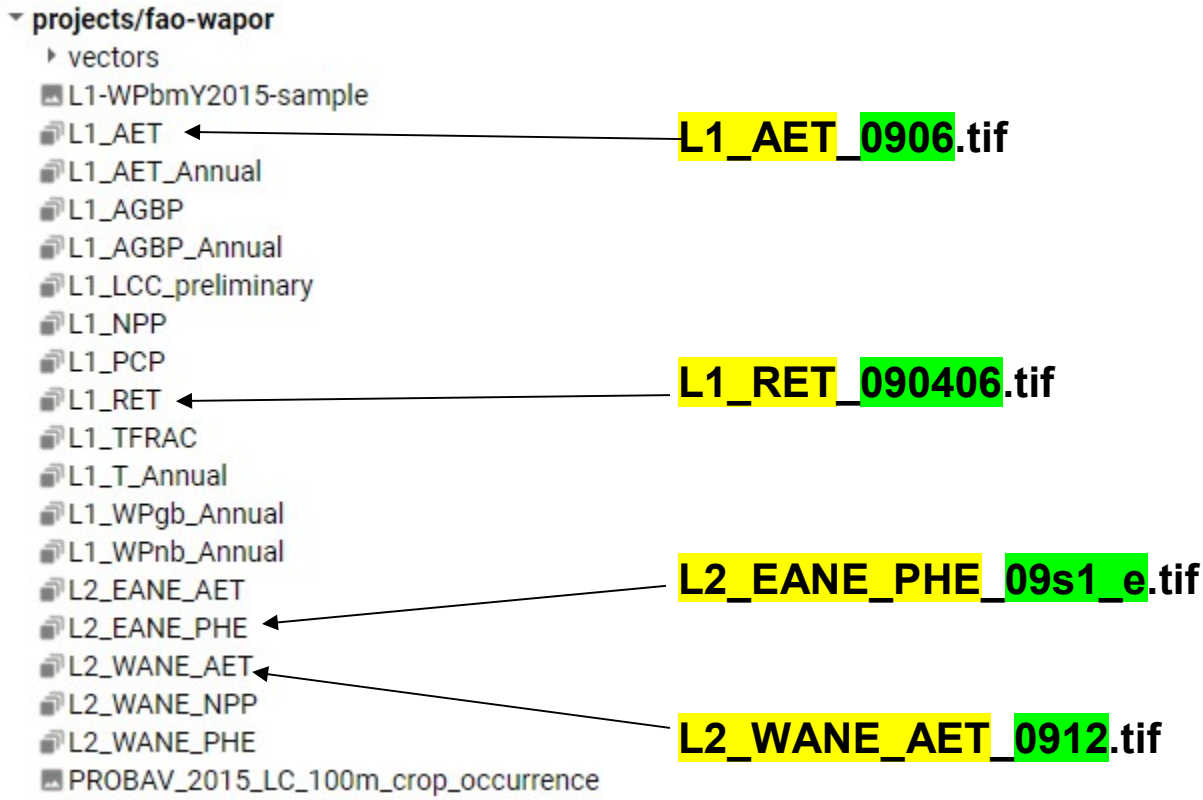
###Example 1
Load all files contained in a directory
wpDataManagement.py ../wapor_algorithms/image_test/

###Example 2
Load specific files

wpDataManagement.py ../wapor_algorithms/image_test/L2_EANE_PHE_09s1_e.tif
wpDataManagement.py ../wapor_algorithms/image_test/L1_RET_090406.tif

Metadata and upload

Files are copied in the correct GEE asset using the left part of the file name highlighted in yellow.
If the asset name is not available is created before copying the file.



The right part of the file name, in green, is used to calculate additional parameters such as the day contained in a dekad, the season, year, starting date of the dekad etc…

Properties (5) area: WANE system:time_start: 2009-02-01 01:00:00 UTC id_no: L2_WANE_AET_0904 level: 2 days_in_dk: 10	**Properties (5)** area: AfNE system:time_start: 2009-04-01 UTC id_no: L1_AET_0910 level: 1 days_in_dk: 10
Properties (4) area: AfNE system:time_start: 2009-04-09 UTC id_no: L1_PCP_090409 level: 1	**Properties (7)** area: EANE system:time_start: 2009-01-01 12:00:00 UTC id_no: L2_EANE_PHE_09s1_e level: 2 year: 2009 season: 1 season_time: e