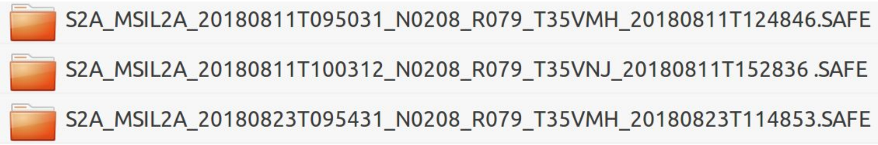
Cropyield updated algorithm

written by Samantha Wittke on 21.04.2020

Updates based on discussions and changes discussed on slack-channel cropyield.
This update is also based on experience when using Puhti array scripts.
The main script will be usable on any system. A few extra scripts will be provided to make use of Puhti array jobs and to get going towards ML ready datasets.

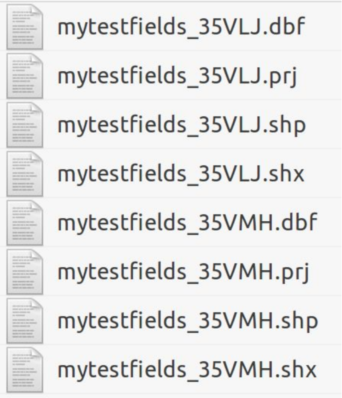
Main Inputs:

- Sentinel-2 data folder (with SAFE ‘files’, no subfolders, there can be several different tiles in it), assumption:all files shall be processed, eg. /home/myname/documents/project/S2/



```
S2A_MSIL2A_20180811T095031_N0208_R079_T35VMH_20180811T124846.SAFE
S2A_MSIL2A_20180811T100312_N0208_R079_T35VNJ_20180811T152836 .SAFE
S2A_MSIL2A_20180823T095431_N0208_R079_T35VMH_20180823T114853.SAFE
```

- folder with shapefiles (named xxx_tilename.extension (eg shp)), result of splitshp.py (which is a provided script to split a big shapefile based on tiles), eg /home/myname/documents/project/shapefiles/



```
mytestfields_35VLJ.dbf
mytestfields_35VLJ.prj
mytestfields_35VLJ.shp
mytestfields_35VLJ.shx
mytestfields_35VMH.dbf
mytestfields_35VMH.prj
mytestfields_35VMH.shp
mytestfields_35VMH.shx
```

Other inputs will be limited to as little as possible (limitation through fixed subfolders in results folder, fixed naming,etc.) .

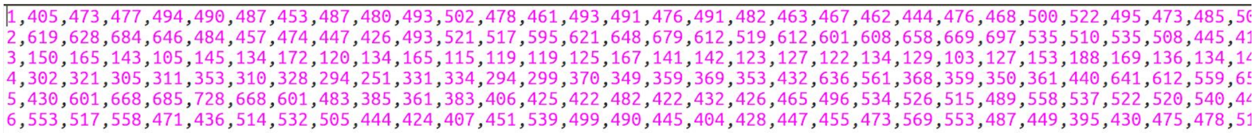
Processing steps:

1. make a list of files (.jp2) to be processed based on content of S2 folder, save list to txt file (pathfinder.py)
2. go through txt file, for each jp2 file (run main.py):
    a. find right shapefile, reproject if necessary

b. extract all pixel values from within (all_touched=False) each polygon into numpy array which is collected per tile, date and band and saved into .csv with ID as first column, following flattened array without nodata, if all array values for one ID are 0, ID is left out (arrayextractor.py)
c. extract metadata per ID (parcelID,year,DOY,tilefilename,missionID,count, with count being the number of non nodata and non 0 pixelvalues) and save into metadatafile per tile, date and band (metaextractor.py)

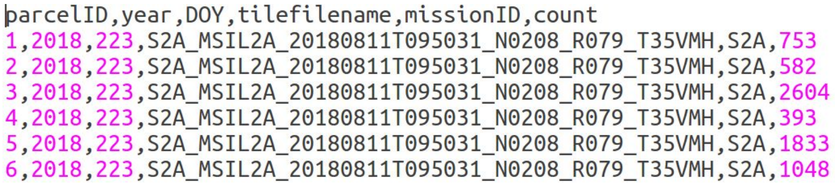
Results:

- one arrayfile per tile,date and band with all IDs of polygons within tile -> given_results_path/arrays/array_tile_date_band.csv (eg array_T35VLH_20180418_B04.csv), looking like this:



```
1,405,473,477,494,490,487,453,487,480,493,502,478,461,493,491,476,491,482,463,467,462,444,476,468,500,522,495,473,485,5
2,619,628,684,646,484,457,474,447,426,493,521,517,595,621,648,679,612,519,612,601,608,658,669,697,535,510,535,508,445,41
3,150,165,143,105,145,134,172,120,134,165,115,119,119,125,167,141,142,123,127,122,134,129,103,127,153,188,169,136,134,14
4,302,321,305,311,353,310,328,294,251,331,334,294,299,370,349,359,369,353,432,636,561,368,359,350,361,440,641,612,559,65
5,430,601,668,685,728,668,601,483,385,361,383,406,425,422,482,422,432,426,465,496,534,526,515,489,558,537,522,520,540,44
6,553,517,558,471,436,514,532,505,444,424,407,451,539,499,490,445,404,428,447,455,473,569,553,487,449,395,430,475,478,51
```

- one metadatafile per tile, date and band with all IDs of polygon within tile -> given_results_path/metadata/meta_tile_date_band.csv (eg meta_T35VLH_20180418_B04.csv), looking like this:



```
parcelID,year,DOY,tilefilename,missionID,count
1,2018,223,S2A_MSIL2A_20180811T095031_N0208_R079_T35VMH,S2A,753
2,2018,223,S2A_MSIL2A_20180811T095031_N0208_R079_T35VMH,S2A,582
3,2018,223,S2A_MSIL2A_20180811T095031_N0208_R079_T35VMH,S2A,2604
4,2018,223,S2A_MSIL2A_20180811T095031_N0208_R079_T35VMH,S2A,393
5,2018,223,S2A_MSIL2A_20180811T095031_N0208_R079_T35VMH,S2A,1833
6,2018,223,S2A_MSIL2A_20180811T095031_N0208_R079_T35VMH,S2A,1048
```

Additional scripts:

- splitshp.py, split big shapefile into separate per tile shapefiles
- mlready.py read in separate csv files for generation of ML ready dataset

Puthi array job notes:

Processing step 1 will run from sbatch script, processing step 2 can be started as several array jobs. Guidelines will be provided (array job script was not wished). Info: Due to array job preparations some non-straightforward design choices (based on experience) had to be made (like splitting process in 1 and 2, one metadata file per tile,date and band, etc...).