HIMAWARI-8 DOCUMENTATION

1 PREPARATION 3

1.1 INSTALLING GEOCAT 3
1.2 PREPARING ANCILLARY DATA FOR GEOCAT 4
1.2.1 DOWNLOADING YOUR OWN ANCILLARY DATA 5
1.3 MAKING A WORK DIRECTORY 5

2 EXTRACTING THE HIMAWARI-8 RAW DATASET 6

2.1 WRITE SCRIPT TO EXTRACT DATASET. 6

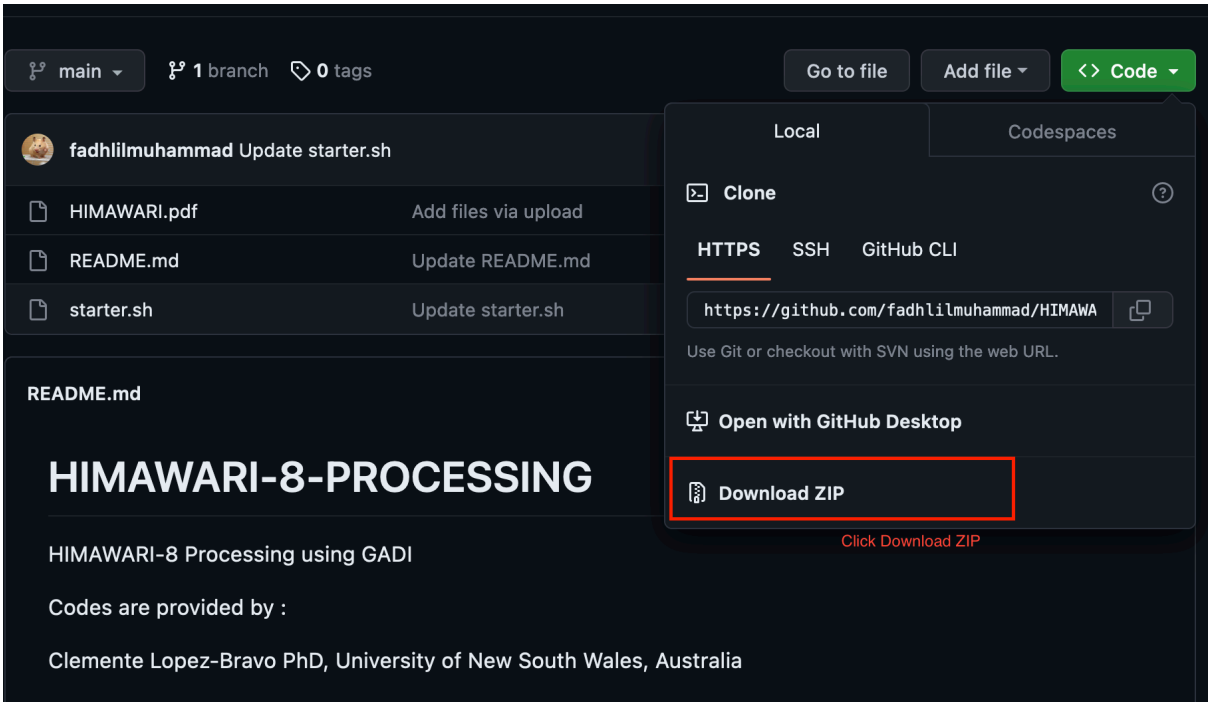
3 PROCESSING THE LEVEL-1 AND LEVEL-2 VARIABLES 10

3.1 LEVEL-1 VARIABLES 10
3.2 LEVEL-2 VARIABLES 12

1 Preparation

1.1 Downloading all the datasets

First, using VDI, go to the github page https://github.com/fadhlilmuhammad/HIMAWARI-8-PROCESSING/tree/main and click download zip as in the figure below:



Then, extract the zip and run the “starter.sh” file using terminal in VDI.

This will download all the datasets you needed for this tutorial.

1.2 Installing GEOCAT

Make your preparation folder for all your dataset. If you are using NCI, type your respective project number (e.g. v46, w40, etc) and your username (e.g. ty7605, gr5690, etc) and type in your terminal:

```
1. mkdir -p /g/data/v46/xx1234/HIMAWARI #make a folder named HIMAWARI
```

Then, copy the preparation datasets, This include the sample data, ancillary data, and GEOCAT.

```
1. cp -r /g/data/v46/xx1234/H8_TUTORIAL/* /g/data/v46/xx1234/HIMAWARI
```

Go to your directory..

```
1. cd /g/data/v46/xx1234/HIMAWARI/
```

Type in your terminal:

```
1. tar xf cspp-geo-geocat-1.0.3b.tar.gz
```

This command will extract the tar.gz file into cspp-geo-geocat-1.0.3b folder.

Then, type:

```
1. export CSPP_GEO_GEOCAT_HOME=$PWD/cspp-geo-geocat-1.0.3b
2. source $CSPP_GEO_GEOCAT_HOME/geocat_env.sh
```

This will set your GEOCAT environment for later use.

Type this line in your terminal to check whether you have done the steps successfully.

```
1. geocat_l2.sh --help
```

The output should be like this:

```
<<code snipped>>

usage: geocat_l2.py [-h] --satellite {goes,him8} [-W work_dir]
[--tmp_dir TMP_DIR] [--interrogate] [-d] [-v] [-V] [-x]
[inputs [inputs ...]]
Run GEOCAT level-2 algorithms on GOES area files, or Himawari-8 HSD or HimawariCast files.
positional arguments:
inputs One or more input files or directories.
optional arguments:
-h, --help show this help message and exit
--satellite {goes,him8}
The satellite to run geocat on. Possible values are
{'goes','him8'}. This option is mandatory.
-W work_dir, --work-dir work_dir work directory which all activity will occur in,
defaults to current dir
--tmp_dir TMP_DIR The directory where the Level 1 and 2 intermediate HDF4 file(s) are written.
--interrogate List the file metadata, and exit. [default: False]
-d, --debug always retain intermediate files. [default: False]
-v, --verbosity each occurrence increases verbosity 1 level from ERROR: -v=WARNING -vv=INFO -
vvv=DEBUG [default: 2]
-V, --version Print the CSPP Geo package version
-x, --expert Display all help options, including the expert ones.
```

1.3 Preparing ancillary data for GEOCAT

Now, we prepare the ancillary data for GEOCAT. The ancillary data can be downloaded from http://geodb.ssec.wisc.edu/ancillary/ . The ancillary data consist of Global Forecasting System datasets (GFS-NCEI/NOAA), sea surface temperature (OISST-NOAA), and snow map. In this example, we will use the ancillary data in the folder ANCILLARY. The ancillary folder **MUST** be in **YYYYMMDD** format.

First, copy all of your ancillary data to cache in GEOCAT folder. Type this in your terminal:

```
1. cp -r /g/data/v46/xx1234/H8_TUTORIAL/ANCILLARY/2016* /g/data/v46/xx1234/HIMAWARI/cspp-geo-
geocat-1.0.3b/anc/cache/
```

Then, type:

```
1. ls /g/data/v46/xx1234/HIMAWARI/cspp-geo-geocat-1.0.3b/anc/cache/
```

to check if your ancillary data is there. Then, you will see something like this:

```
1. [xx1234@gadi-login-04 xx1234]$ ls /g/data/v46/xx1234/HIMAWARI/GEOCAT/cspp-geo-geocat-
1.0.3b/anc/cache/
```

```
2016_10_09_283 2016_10_10_284  geocat_hdf
```

"YYYY_MM_DD_xxx" indicates your ancillary data folder.

Then, we have to rename the data folder to YYYYMMDD format. Type in your terminal:

```
1. mv /g/data/v46/xx1234/HIMAWARI/cspp-geo-geocat-1.0.3b/anc/cache/2016_10_09_283/
/g/data/v46/xx1234/HIMAWARI/cspp-geo-geocat-1.0.3b/anc/cache/20161009/
2. mv /g/data/v46/xx1234/HIMAWARI/cspp-geo-geocat-1.0.3b/anc/cache/2016_10_10_284/
/g/data/v46/xx1234/HIMAWARI/cspp-geo-geocat-1.0.3b/anc/cache/20161010/
```

Check if your renaming is successful:

```
1. ls /g/data/v46/xx1234/HIMAWARI/cspp-geo-geocat-1.0.3b/anc/cache/
```

You will see something like this:

```
1. [xx1234@gadi-login-04 xx1234]$ ls /g/data/v46/xx1234/HIMAWARI/cspp-geo-geocat-
1.0.3b/anc/cache/
2. 20161009 20161010 geocat_hdf
```

1.3.1 Downloading your own ancillary data

There will be times where you want to use your own dataset and use the ancillary data from http://geodb.ssec.wisc.edu/ancillary/. One way to do this is by using 'wget'. For example, we going to use data from 28-03-2023. Look at the website and find the date that you want. In this case, the date is "2023-03-28".

First, access your ancillary folder:

```
1. cd /g/data/v46/xx1234/HIMAWARI/ANCILLARY
```

Type this in your terminal:

```
1. wget --no-parent -r http://geodb.ssec.wisc.edu/ancillary/2023_03_28_087/
```

This command will download all the ancillary dataset in the web page for the given date. Then, you can rename copy and rename this folder to your GEOCAT folder.

Type:

```
1. cp -r /g/data/v46/xx1234/HIMAWARI/ANCILLARY/2023_03_28_087/ /g/data/v46/xx1234/HIMAWARI/cspp-
geo-geocat-1.0.3b/anc/cache/
2. mv /g/data/v46/xx1234/HIMAWARI/cspp-geo-geocat-1.0.3b/anc/cache/2023_03_20_087/
/g/data/v46/xx1234/HIMAWARI/cspp-geo-geocat-1.0.3b/anc/cache/20230320/
```

1.4 Making a work directory

Now, we make a work directory for our process. If you are using NCI you could put the working directory on your 'g/data/'. For example:

```
1. #go to your g/data in project v46 for user xx1234(the project number and user id is according
to you)
2. cd /g/data/v46/xx1234/HIMAWARI/
3. mkdir -p work. #make a directory named 'work'
```

This concludes the preparation part of HIMAWARI dataset processing. In the next section we will explain how to process HIMAWARI dataset and extract the level-1 and level-2 variables.

2 Extracting the HIMAWARI-8 Raw Dataset

2.1 Write script to extract dataset.

First, access your folder.

```
1. cd /g/data/v46/xx1234/HIMAWARI/
```

Then, write a script using vim, type:

```
1. vi
```

Then you will get screen like this in your terminal:

```
~                              VIM - Vi IMproved
~
~                               version 9.0.639
~                           by Bram Moolenaar et al.
~                 Vim is open source and freely distributable
~
~                           Sponsor Vim development!
~                type  :help sponsor<Enter>    for information
~
~                type  :q<Enter>               to exit
~                type  :help<Enter>  or  <F1>  for on-line help
~                type  :help version9<Enter>   for version info
~
```

Type **i** to start writing. Then copy and paste the code below:

```
#PBS -l wd

#PBS -lstorage=gdata/v46+gdata/w40                          #Your storage (e.g., scratch/w40)'

#PBS -q normalbw                                            #process (e.g., normalbw,hugemembw)'

export LC_ALL=C.UTF-8                                       #debugger'
export LANG=C.UTF-8

VAR='/g/data/v46/xx1234/GEOCAT_2/cspp-geo-geocat-1.0.3b'  #Your variables'
export CSPP_GEO_GEOCAT_HOME=$VAR                            #export environment'

source $CSPP_GEO_GEOCAT_HOME/geocat_env.sh

HFLDKD='/g/data/v46/lb5963/2023/HMW_2023/DATA_HMW/2016_10_10_0000/'
                                                            #locate the himawari dataset'
module purge
module load openmpi/2.1.6

geocat_l2.sh --no_rap --satellite him8 --num_cpu 96 --viewport -0.5 -0.4 0.25 0.2  -W work
$HFLDKD                                                     #run'
```

Then you will get something like this:

```
 #PBS -l wd

 #PBS -lstorage=gdata/v46+gdata/w40                    #Your storage (e.g., scr
atch/w40)'

 #PBS -q normalbw                                      #process (e.g., normalbw
,hugemembw)'

 export LC_ALL=C.UTF-8                            #debugger'
 export LANG=C.UTF-8

 VAR='/g/data/v46/fm6730/GEOCAT_2/cspp-geo-geocat-1.0.3b'  #Your variables'
 export CSPP_GEO_GEOCAT_HOME=$VAR                          #export environment'

 source $CSPP_GEO_GEOCAT_HOME/geocat_env.sh

 HFLDKD='/g/data/v46/lb5963/2023/HMW_2023/DATA_HMW/2016_10_10_0000/'
                                                  #locate the himawari dataset'
 module purge
 module load openmpi/2.1.6

 geocat_l2.sh --no_rap --satellite him8 --num_cpu 96 --viewport -0.5 -0.4 0.25 0
.2  -W work  $HFLDKD                                                 #run'
~
~
~
-- INSERT --
```

Type "**:wq runtest.sh**" to save and finish (without the quotation marks).

Notes before continuing:

This script will get you a non-projected data of himawari-8. Non-projected means that it is a raw image captured by the satellite that includes the earth curvature. The last line in the script extracts the himawari dataset. You can change this line to suits your needs.

Some optional arguments (e.g., --no_rap,--num_cpu) in the "geocat_l2.sh" are:

```
optional arguments (taken from GEOCAT installation guide):

-h, --help show this help message and exit
--satellite {goes,him8} The satellite to run geocat on. Possible values are {'goes','him8'}.
This option is mandatory.
-W work_dir, --work-dir work_dir work directory which all activity will occur in,
defaults to current dir
--tmp_dir TMP_DIR The directory where the Level 1 and 2 intermediate HDF4 file(s) are written.
--ancillary_only Only retrieve and process ancillary data, do not run geocat. [default: False]
--no_rap Do not use Mesoscale model (RAP) files.
[default: False]
--cache_window CACHE_WINDOW Limit product cache to hold no more that this number of hours
preceding the target time.
[default: 6. hours]
```

```
--preserve_cache Do not flush old files from the product cache.
[default: False]
--viewport LLCRNRX LLCRNRY URCRNRX URCRNRY Lower-left and upper-right coordinates
[*llcrnrx*, *llcrnry*, *urcrnrx*, *urcrnry*] of the projection viewport, in the range
[-0.5,+0.5]. [default: None]
--viewport_xy YSTART YEND XSTART XEND Starting and ending lines [*ystart*, *yend*],and starting
and ending elements with stride
[*ystart*, *yend*, *xstride*], of the projection viewport. [default: None]
--xstride XSTRIDE The number of elements *xstride* to stride in the x-direction.
--num_cpu NUM_CPU The number of CPUs to try and use. [default: 1]
--geocat_nscans GEOCAT_NSCANS Geocat will partition its processing tasks into chunks of
'geocat_nscans'. [default: 200]
--line_segments LINE_SEGMENTS Partition the input image into *line_segments*
rows, which will be processed separately by geocat. Should not exceed 20. [default: 2]
--element_segments ELEMENT_SEGMENTS Partition the input image into
*element_segments* columns, which will be processed separately by geocat. Should not exceed 20.
[default: 2]
--interrogate List the file metadata, and exit. [default:
False]
-d, --debug always retain intermediate files. [default:
False]
-v, --verbosity each occurrence increases verbosity 1 level from ERROR: -v=WARNING -vv=INFO -
vvv=DEBUG
[default: 2]
-V, --version Print the CSPP Geo package version
-x, --expert Display all help options, including the expert ones.
```

The most important arguments are --satellite, --no_rap, --num_cpu, --W and --viewport. Here, we will explain what do these 4 arguments do:

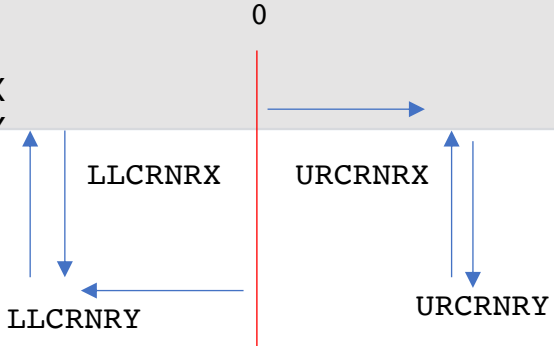
1. --satellite
   This argument indicates what kind of dataset you are using. If it is himawari-8 then it is **{him8}.** This argument is **MANDATORY.**

2. --no_rap
   Set {--no_rap True} if you plan to use mesoscale model to derive your Level-1 or Level-2 variables. Remember that this option may take lots of calculation, so if you do not really need it just set this to {--no rap False}.

3. –num_cpu
   Set the number of CPU that we will use. Do not forget this argument to set the number of CPU or this will default to one, which can cause shortage of computing power.

4. --W
   Set the working directory.

5. --viewport `LLCRNRX LLCRNRY URCRNRX URCRNRY`
   Set the region range of the extracted data. Remember that this is NOT a Lat Lon coordinates. It is based on the projection view of the satellite.
   - `LLCRNRX` : lower-left corner X
   - `LLCRNRY` : lower left corner Y
   - `URCRNRX`: upper-right corner X
   - `URCRNRY`: upper-right corner Y



For Australia, the viewport used is:

- `LLCRNRX` : -0.5
- `LLCRNRY` : -0.4
- `URCRNRX` : 0.25
- `URCRNRY` : 0.2

End of notes

If you have saved and finish the script. Now, you can submit the job to GADI.

```
1. chmod +rwx runtest.sh
2. qsub runtest.sh
```

Use command “qstat” to see if the job has finished yet.

The output will be the raw satellite data from HIMAWARI-8 consisting of 16 band imagers with different wavelength and spatial resolution. The details are given as follows:

Taken from www.en.wikipedia.org/wiki/HIMAWARI_8

Wavelength (μm)	Band number	Spatial resolution at SSP (km)	Central wavelength (μm)
0.47	1	1	0.47063
0.51	2	1	0.51000
0.64	3	0.5	0.63914
0.86	4	1	0.85670
1.6	5	2	1.6101
2.3	6	2	2.2568
3.9	7	2	3.8853
6.2	8	2	6.2429
6.9	9	2	6.9410
7.3	10	2	7.3467
8.6	11	2	8.5926
9.6	12	2	9.6372
10.4	13	2	10.4073
11.2	14	2	11.2395
12.4	15	2	12.3806

13.3	16	2	13.2807

3 Processing the Level-1 and Level-2 Variables

3.1 Level-1 Variables

Now, we are going to process the Level-1 (L1) and Level-2 (L2) variables from Himawari. First, we copy the sample dataset to your working directory. On your terminal, execute:

Go to your TUTORIAL directory:

```
1. /g/data/xx1234/HIMAWARI
```

Execute:

```
1. cd L1_AND_L2
```

Those commands copy the sample data to directory “L1_AND_L2”. Inside the folder, there should be 1 subfolder named “FILE” and four additional files, named “HMW_L1.sh”, “HMW_L2.sh”, “list.txt”, and “list_L2.txt”. These are the script to process Level-1 and Level-2 variables from HIMAWARI dataset. Now, we will explain about these files.

The “HMW_L1.sh” contains script to process the Level-1 variables. This script attempts to create python script to process the L1 variables using bash command. This script extracts the *brightness temperature* captured by the HIMAWARI. The script is as follows:

```
 1. #!/bin/bash
 2.
 5. #PBS -q normal
 6. #PBS -l walltime=05:00:00
 7. #PBS -l mem=64GB
 8. #PBS -l ncpus=1
 9. #PBS -l wd
10. #PBS -lstorage=scratch/v46+gdata/w40+scratch/w40+gdata/hh5+gdata/v46
14.
15. varg='CH8'                                                  #select your variables (TB)
16. HWV='himawari_8_ahi_channel_8_brightness_temperature'       #Himawari channel
17. DDIR='/g/data/v46/lb5963/HIMAWARI/08_V46_SUMM_2021-22/L1/'  #select input directory
18. WDIR=`pwd`'/'                                               #working directory
19. nr='55'                                                     #identifier
20. ODIR=$WDIR'DATA/'                                           #output directory
21. ol='list.txt'                                               #list of raw datafiles
22.
23. # LIST TOTAL ls $DDIR > $ol                                 #create directory
24.
25. if [ -d $ODIR ]; then
26.        echo "YES"
27. else
28.         mkdir -p $ODIR
29. fi
30.
31.
32. while read lst; do
33.
34. module purge
```

```
module use /g/data3/hh5/public/modules
module load conda/analysis3
module load gdal


ifl=$DDIR$lst
echo $ifl
pyf=$varg'_'$nr'.py'                                          #make py script on-the-fly

sed -e "14s?XXX?"$ifl"?g"  -e "37s/XXX/"$lst"/g" -e "17s/XXX/"$HWV"/g" -e "46s/XXX/"$HWV"/g" 'FILE/guia.py' > $pyf
#echo $lst
python  $pyf


if [ -f $pyf ]; then
rm  $pyf
else
echo "NOP"
fi

echo '############### STEP 1 ################################'
echo '############### STEP 1 ################################'

gdal_translate -a_srs "+proj=geos +lon_0=140.7 +h=35785863 +x_0=0 +y_0=0 +a=6378137 +b=6356752.3 +units=m +no_defs" \
-a_ullr -5500500 -5500500 5500500 5500500  \
-of netCDF  NETCDF:$input:$HWV \
$file    -unscale

data_path=$WDIR
short_file_name=$file

gdal_file_name=NETCDF:"$data_path/$short_file_name":$HWV

# CLM gdalinfo $gdal_file_name

if [ -f $lst ]; then
rm  $lst
else
echo "NOP"
fi



if [ -f $out ]; then
rm  $out
else
echo "NOP"
fi


echo '############### STEP 2 ################################'
echo '############### STEP 2 ################################'

gdalwarp -overwrite -r cubic \
-t_srs '+proj=latlong +a=6378137 +b=6356752.3' \
-te 105 -45 160 -7 -tr 0.018 0.018 \
-of netCDF  $gdal_file_name  $ODIR$out  \
-multi -wo "NUM_THREADS=4" -co "COMPRESS=DEFLATE"

if [ -f $file ]; then
rm  $file
else
echo "NOP"
fi



done <$ol
```

```
#cat $PBS_NODEFILE

```

Command explanation:

1. **“gdal_translate”** command assigns a projection in the HIMAWARI dataset. This is important since the raw dataset of HIMAWARI does not have any projection. Thus, if we plot the dataset without first projecting it, it will become inaccurate (see https://gdal.org/programs/gdal_translate.html for details).
2. **“gdalwarp”** command reproject the projection into regional projections. In this script, the command is already customised for Australia. One might wants to change some of the details for other region (see https://gdal.org/programs/gdalwarp.html).

3.2 Level-2 Variables

For the L2 variables, the script is as follows:

```
#!/bin/bash

#PBS -q normal
#PBS -l walltime=05:00:00
#PBS -l mem=64GB
#PBS -l ncpus=1
#PBS -l wd
#PBS -lstorage=scratch/v46+gdata/w40+scratch/w40+gdata/hh5+gdata/v46

varg='CTT'                                                      #variable
HWV='ACHA_mode_8_cloud_top_temperature'                         #varname in HIMAWARI
DDIR='/g/data/v46/lb5963/HIMAWARI/08_V46_SUMM_2021-22/L2/'      #input directory
WDIR=`pwd`'/'                                                   #working directory
nr='55'                                                         #identifier
ODIR=$WDIR'DATA/'                                               #output directory
ol='list_L2.txt'                                                #data list

# LIST TOTAL ls $DDIR > $ol                                     #create dir

if [ -d $ODIR ]; then
        echo "YES"
else
         mkdir -p $ODIR
fi


while read lst; do


module purge
module use /g/data3/hh5/public/modules
module load conda/analysis3
module load gdal


ifl=$DDIR$lst
echo $ifl
pyf=$varg'_'$nr'.py'
                                                        #line below creates python file
sed -e "14s?XXX?"$ifl"?g"  -e "37s/XXX/"$lst"/g" -e "17s/XXX/"$HWV"/g" -e
"46s/XXX/"$HWV"/g" 'FILE/guia.py' > $pyf
```

```
#echo $lst
python  $pyf


if [ -f $pyf ]; then
rm  $pyf
else
echo "NOP"
fi


############## GDAL ##################

input=$lst
file='R_'$lst
out=$varg'_'$lst

if [ -f $file ]; then
rm  $file
else
echo "NOP"
fi

echo '############## STEP 1 ################################'
echo '############## STEP 1 ################################'

gdal_translate -a_srs "+proj=geos +lon_0=140.7 +h=35785863 +x_0=0 +y_0=0 +a=6378137
+b=6356752.3 +units=m +no_defs" \
-a_ullr -5500500 -5500500 5500500 5500500  \
-of netCDF  NETCDF:$input:$HWV \
$file    -unscale

data_path=$WDIR
short_file_name=$file

gdal_file_name=NETCDF:"$data_path/$short_file_name":$HWV

# CLM gdalinfo $gdal_file_name

if [ -f $lst ]; then
rm  $lst
else
echo "NOP"
fi



if [ -f $out ]; then
rm  $out
else
echo "NOP"
fi

echo '############## STEP 2 ################################'
echo '############## STEP 2 ################################'

gdalwarp -overwrite -r cubic \
-t_srs '+proj=latlong +a=6378137 +b=6356752.3' \
-te 105 -45 160 -7 -tr 0.018 0.018 \
-of netCDF  $gdal_file_name  $ODIR$out  \
-multi -wo "NUM_THREADS=4" -co "COMPRESS=DEFLATE"

if [ -f $file ]; then
rm  $file
else
echo "NOP"
fi


```

```
done <$ol
#cat $PBS_NODEFILE
```

The script is pretty much the same as the L1 script. To change the variables, one could change the “varg” and “HMW”. Full list of the L1 and L2 variables can be seen in https://www.eorc.jaxa.jp/ptree/documents/README_HimawariGeo_en.txt.

After this, you can plot the dataset using the “plotbook.ipynb” in the tutorial folder.