LGHAP aerosol dataset user guide (version 1)

Prepared by

Kaixu Bai and Ke Li

Key Laboratory of Geographic Information Science
School of Geographic Sciences, East China Normal University
Shanghai 200241, China

2021. 11

1. Descriptions of LGHAP aerosol dataset

LGHAP, the acronym of Long-term Gap-free High-resolution Air Pollutants concentration dataset, was generated via a big data analytics framework to provide gap free and high-resolution AOD, $PM_{2.5}$, PM_{10}, NO_2, O_3, and SO_2 concentrations to advance environment management and earth system science analysis. The LGHAP aerosol dataset in China is currently available and publicly accessible. The dataset was generated via a seamless integration of the tensor flow based multimodal data fusion with ensemble learning based knowledge transfer in statistical data mining. The proposed method transformed a set of data tensors of AOD and other related datasets such as air pollutants concentration and atmospheric visibility that were acquired from diversified sensors or platforms via integrative efforts of spatial pattern recognition for high dimensional gridded data analysis toward data fusion and multiresolution image analysis, as well as knowledge transfer in statistical data mining. The proposed method consists of three major procedures in general, including multisensory data homogenization, tensor flow based AOD reconstruction, and ensemble learning for PM concentration estimation. For more details of the analytical framework of the big data analytics to generate the LGHAP aerosol dataset, please refer to Bai et al. (2021).

In the current release of LGHAP aerosol dataset (LGHAP.v1), we provide a 21-year-long (2000–2020) gap free AOD, $PM_{2.5}$, and PM_{10} concentration with daily 1-km resolution covering the land area of China. Ground validation results indicate that the LGHAP AOD data are in a good agreement with *in situ* AOD observations from AERONET, with R of 0.91 and RMSE equaling to 0.21. Meanwhile, $PM_{2.5}$ and PM_{10} estimations also agreed well with ground measurements, with R of 0.95 and 0.94 while RMSE of 12.03 and 19.56 μg m^{-3}, respectively. Overall, the generated LGHAP aerosol dataset has a great potential to trigger multidisciplinary applications in earth observations, climate change, public health, ecosystem assessment, and environmental management.

Data users are encouraged to cite both the data set and the scientific publication given below when making use of these datasets:

Bai, K., Li, K., Tan, Z., Han, D., and Guo, J.: Daily 1-km gap-free AOD grids in China, v1 (2000–2020). https://doi.org/10.5281/zenodo.5652257. Accessed DAY MONTH YEAR.

Bai, K., Li, K., Tan, Z., Han, D., and Guo, J.: Daily 1-km gap-free PM2.5 grids in China, v1 (2000–2020). https://doi.org/10.5281/zenodo.5652265. Accessed DAY MONTH YEAR.

Bai, K., Li, K., Tan, Z., Han, D., and Guo, J.: Daily 1-km gap-free PM10 grids in China, v1 (2000–2020). https://doi.org/10.5281/zenodo.5652263. Accessed DAY MONTH YEAR.

Bai, K., Li, K., Ma, M., Li, K., Li, Z., Guo, J., Chang, N.-B., Tan, Z., and Han, D.: LGHAP: a Long-term Gap-free High-resolution Air Pollutants concentration dataset derived via tensor flow based multimodal data fusion. Earth System Science Data, 2021. Under review

2. Data format and naming conventions

The daily gap free AOD, $PM_{2.5}$, and PM_{10} concentration map was provided separately in the NetCDF format, while data in each individual year were archived in a zip file.

The zip file was named after in a format such as ***LGHAP.AOD.D001.Y2000.zip***. **LGHAP** is the dataset name. **AOD** is the product name, which can also be $PM_{2.5}$, PM_{10}, or other data product. **D001** indicates the resolution of data product as **D** means daily (temporal resolution) and **001** is the grid resolution of 0.01° (spatial resolution). **Y2000** is the time of the year (2000 in the case).

Similar naming convention is also applied to the daily product. For illustration, here we take the file of ***SCHAP.AOD.D001.A20200101.nc*** as an example. The only difference is the time for the given file. For daily product, the time was given in a format of *Ayyyymmdd*. yyyymmdd is the date for the given data, in which yyyy is the calendar year, *mm* is the month and *dd* is the day-of-month number.

3. Example codes to read and visualize the LGHAP data

Below gives illustrative codes to help users read and visualize AOD data in LGHAP dataset using MATLAB, Python, R, and IDL programming language. Figure 1 shows the map of gap free AOD distribution in China on November 23, 2014 that was created with the LGHAP AOD dataset. Demo codes to read and visualize another two products ($PM_{2.5}$ and PM_{10}) were also provided.

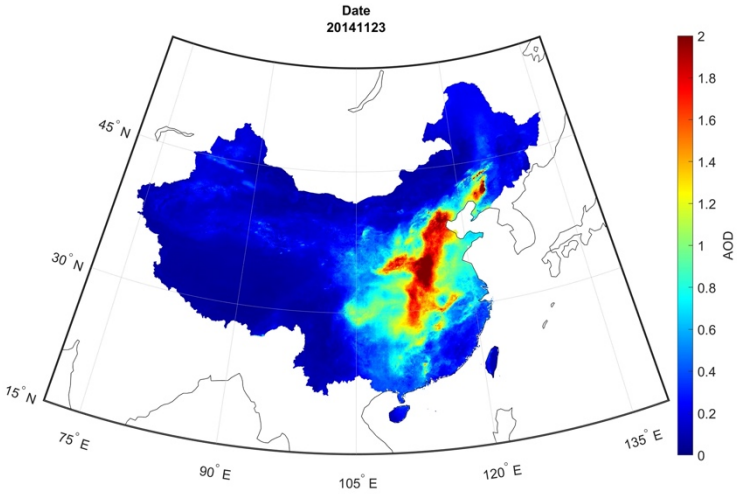

Fig. 1. The map of gap free AOD distribution in China on November 23, 2014 that was created with the LGHAP AOD dataset.

MATLAB code:

```
% This demo provides an illustration on how to read and visualize AOD in
% LGHAP dataset and write data into GeoTIFF in MATLAB.
%
% The demo was prepared with MATLAB R2020b.
%
% Last updated: 2021-11-01

%% (a) read AOD data from NetCDF file

% the nc file to be read
filename = 'LGHAP.AOD.D001.A20141123.nc';

% read longitude and latitude from NetCDF file
lon = ncread(filename,'lon');
lat = ncread(filename,'lat');

% read AOD data
AOD = ncread(filename,'AOD');

%% (b) visualization

% create georeference struct to map AOD data on projected datum
R = georefcells([14.995,56.005],[70.995,139.005],[4100,6800]);
latlim = R.LatitudeLimits;
lonlim = R.LongitudeLimits;
```

```
% load coastlines
load coastlines

% plot data
pos = get(0,'ScreenSize');

figure('color','w','position',[50,50,round(pos(3)*0.5),round(pos(4)*0.5)]);

worldmap(latlim,lonlim)

% plot coastline
geoshow(coastlat,coastlon,'Color','k')

% plot AOD
geoshow(AOD,R,'DisplayType','surface');

% set colormap limits
caxis([0,2]);

% set colors used to map data
colormap('Jet');

% add colorbar
h = colorbar();
h.Label.String = 'AOD';
h.FontSize = 14;
setm(gca,'FontSize',14);

% add title to the plot
title(['Date',string(regexp(filename,'\d{8}','match'))],'FontSize',14);

% save plot in jpg format with resolution of 300 DPI
savename = ['LGHAP.AOD.D001.A' char(string(regexp(filename,'\d{8}','match'))) '.jpg'];
print(savename,'-djpeg','-r300');

%% (c) save as GeoTIFF

% define fillvalue
Fillvalue = -999;
AOD_tiff = AOD;
AOD_tiff(isnan(AOD_tiff)) = Fillvalue;

% write GeoTIFF
```

```
savename = ['LGHAP.AOD.D001.A' char(string(regexp(filename,'\d{8}','match'))) '.tif'];
geotiffwrite(savename,AOD_tiff,R);
```

Python code:

```
# -*- coding: utf-8 -*-
"""
This demo provides an illustration on how to read and visualize AOD in
LGHAP dataset and write data into GeoTIFF in Python.

netCDF4 is needed to read netCDF file
You can install this library by command: pip install netCDF4

The demo was prepared with Python3.7.

Last updated: 2021-11-01

"""

# import supporting library
import netCDF4 as nc
import numpy as np
import matplotlib.pyplot as plt
from osgeo import gdal,osr
import re

## (a) read AOD data from NetCDF file
# the nc file to be read
filename = 'LGHAP.AOD.D001.A20141123.nc'
f = nc.Dataset(filename)

# extract date from filename
date = re.compile(r'\d{8}').findall(filename)

# read longitude and latitude from NetCDF file
lon = np.array(f['lon'][:])
lat = np.array(f['lat'][:])

# read AOD data
AOD = np.array(f['AOD'][:])

# replace missing value with nan
AOD[AOD==65535] = np.nan

## (b) visualization
# create canvas
fig, ax = plt.subplots(figsize=(17,10))
```

```
# plot AOD
im = ax.imshow(np.rot90(AOD,k=1),vmin = 0,vmax = 2,cmap = 'jet',extent =
[min(lon),max(lon),min(lat),max(lat)])

# set ticks and label
ax.set_xlabel('Longitude(°)',size=20)
ax.set_ylabel('Latitude(°)',size=20)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)

# add title
plt.title('Date:'+date[0],size=20)

# add colorbar
fig.subplots_adjust(right=0.9)
position = fig.add_axes([0.90,0.15,0.015,0.70])
cb = fig.colorbar(im,cax=position)
cb.ax.tick_params(labelsize=20)
cb.set_label('AOD',size=20)

# save plot in jpg format with resolution of 300 DPI
plt.savefig('LGHAP.AOD.D001.A'+date[0]+'.jpg',dpi=300)
plt.show()

## (c) save as GeoTIFF
# define fillvalue
AOD_tiff = AOD
AOD_tiff[np.isnan(AOD_tiff)] = -999

# write GeoTIFF
driver = gdal.GetDriverByName('GTiff')
dataset = driver.Create('LGHAP.AOD.D001.A'+date[0]+'.tif',len(lon),len(lat),1,gdal.GDT_Float64)
dataset.SetGeoTransform([min(lon),0.01,0,max(lat),0,-0.01])
sr = osr.SpatialReference()
sr.SetWellKnownGeogCS('WGS84')
dataset.SetProjection(sr.ExportToWkt())
dataset.GetRasterBand(1).WriteArray(np.rot90(AOD_tiff,k=1))

# release memory
del dataset
f.close()
```

R code:

```
# This demo provides an illustration on how to read and visualize AOD in
# LGHAP dataset and write data into GeoTIFF in R.
#
# four libraries (ncdf4,raster,rgdal and ggplot2) are needed.
#
# The demo was prepared with R 3.6.1.
#
# Last updated: 2021-11-01
# install.packages

library(ncdf4) # package for netcdf manipulation
library(raster) # package for raster manipulation
library(rgdal) # package for geospatial analysis
library(ggplot2) # package for plotting

# set your path
setwd('...')

## (a) read AOD data from NetCDF file
# the nc file to be read
filename = "LGHAP.AOD.D001.A20141123.nc"
ncdata = nc_open(filename)

# extract date from filename
date = substr(filename,regexpr('\\d{8}',filename),regexpr('\\d{8}',filename)+7)

# read longitude and latitude from NetCDF file
lon = ncvar_get(ncdata,'lon')
lat = ncvar_get(ncdata,'lat')

# read AOD data
AOD = ncvar_get(ncdata,'AOD')

# close nc file
nc_close(ncdata)

## (b) visualzation
# convert AOD to raster in memory
r = raster(AOD, xmn=min(lon), xmx=max(lon), ymn=min(lat), ymx=max(lat),
crs=CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs+ towgs84=0,0,0"))
r = flip(r, direction='y')

# plot AOD and save plot in jpg format with resolution of 300 DPI
```

```
savename = paste(c('LGHAP.AOD.D001.A',date,'.jpg'),collapse = '')
jpeg(file = savename,width = 5000,height = 4000,res = 300)
r2 = r
r2[r2>2]=2
plot(r2,col=topo.colors(100),main=paste(c("AOD Date: ",date),collapse = ''),zlim=c(0,2),xlab =
"Longitude", ylab = "Latitude",cex.axis=1.5,cex.lab=1.5,cex.main=2)
dev.off()

## (c) save as GeoTIFF
# write GeoTIFF
nc2raster = stack(r)
savename = paste(c('LGHAP.AOD.D001.A',date,'.tif'),collapse = '')
writeRaster(nc2raster,savename,format = 'GTiff',overwrite = TRUE)
```

IDL code:

```
;This demo provides an illustration on how to read and visualize AOD in
; LGHAP dataset and write data into GeoTIFF in IDL.

; The demo was prepared with IDL8.5.

; Last updated: 2021-11-01
PRO read_LGHAP_AOD

 ;Determine compilation rules
 compile_opt idl2

 ;Set your path
 envi, /restore_base_save_files
 envi_batch_init
 CD, '...'
 outpath = '...'
 ; read AOD data from NetCDF file


 filearr = dialog_pickfile(/multiple_files,  title = 'Open the original nc file')

 num = n_elements(filearr)


 FOR i=0,num-1,1 DO BEGIN

  file_ID = ncdf_open(filearr, /nowrite )

  latid = NCDF_VARID(file_ID,'lat')      ;read latitude from NetCDF file
  NCDF_VARGET, file_ID, latid, lat
  nlat = N_ELEMENTS(lat)

  lonid = NCDF_VARID(file_ID,'lon')      ;read  longitude from NetCDF file
  NCDF_VARGET, file_ID, lonid, lon
  nlon = N_ELEMENTS(lon)

  AODid = NCDF_VARID(file_ID,'AOD')       ; read AOD data
  NCDF_VARGET, file_ID, AODid, AOD
  nAOD = N_ELEMENTS(AOD)
  ;Obtain the gain, offset, and fill values of AOD data, and perform calibration
  ncdf_attget,file_ID,AODid,'scale_factor',a
  ncdf_attget,file_ID,AODid,'add_offset',b
  ncdf_attget,file_ID,AODid,'_FillValue',fv
```

```
   fv_index = where(AOD eq fv)
   output_AOD = (AOD ne fv)*AOD*a+b
   ; define fillvalue
   output_AOD[fv_index] = -999

   ;visualization

   loadct, 33
   TVLCT, r, g, b, /get
   color_table = BYTARR(3, 256)
   color_table[0, *] = r
   color_table[1, *] = g
   color_table[2, *] = b
   color_table[*, 0] = [255, 255, 255]  ;Custom colorbar


img=image(transpose(output_AOD),rgb_table=color_table,title=file,grid_units=2,POSITION=[0.
1,0.15,0.9,0.95],map_projection='geographic', image_dimensions=[max(lon)-min(lon),max(lat)-
min(lat)] ,$
     IMAGE_LOCATION=[min(lon),min(lat)],DIMENSIONS=[691,512])

   ;set colormap limits
   img.MAX_VALUE=2
   img.MIN_VALUE=0
   ;Change the figure title, grid type, axis text direction, and color bar form
   img.title = 'Date:'+strmid(file_basename(filearr),16,8)

   img.mapgrid.label_position=0
   img.mapgrid.font_name='Palatino'

   img.mapgrid.linestyle=6
   img.mapgrid.horizon_thick=1

   lons=img.mapgrid.longitudes
   lats=img.mapgrid.latitudes
   for lons_i=0,n_elements(lons)-1 do begin
     lons[lons_i].label_angle=0
     lons[lons_i].label_align=0
   endfor

   ; add colorbar
   c = COLORBAR(TARGET=img,
ORIENTATION=1,TITLE='AOD',POSITION=[0.905,0.25,0.925,0.85])
   c.RANGE=[0,2]
```

```
   c.BORDER=0
   c.TICKDIR= 1
   c.TEXTPOS = 1

   ;save plot in jpg format with resolution
   Img.save,outpath+'LGHAP.AOD.D001.A'+strmid(file_basename(filearr),16,8)+'.jpg'
   ; save as GeoTIFF
   ;Write geographic information structure
   geo_info={$
     MODELPIXELSCALETAG:[0.01,0.01,0.0],$
     MODELTIEPOINTTAG:[0.0,0.0,0.0,min(lon),max(lat),0.0],$
     GTMODELTYPEGEOKEY:2,$
     GTRASTERTYPEGEOKEY:1,$
     GEOGRAPHICTYPEGEOKEY:4326,$
     GEOGCITATIONGEOKEY:'GCS_WGS_1984'}

   ;write GeoTIFF

WRITE_TIFF,outpath+'LGHAP.AOD.D001.A'+strmid(file_basename(filearr),16,8)+'.tif',reverse(tr
anspose(output_AOD),2),/float, geotiff=geo_info
   ;Close nc file ID
   NCDF_CLOSE, file_ID
 endfor

END
```