FengYun-4B/GIIRS FYGeoAIR Carbon Monoxide (CO) retrievals from July 2022 to June 2025

Project lead: Zhao-Cheng Zeng (Peking University; zczeng@pku.edu.cn)

1 Descriptions

This dataset contains CO retrieval data observed by the Geostationary Interferometric Infrared Sounder (GIIRS) on board China's FengYun-4B satellite, which was launched in 2021. The datasets include CO profile estimates, the uncertainty, and the averaging kernel (AK) information. The spatial resolution is 2km and temporal resolution 2-hour covering both day and night. The data have been pre-filtered to ensure the high data quality. The nighttime data should be used with caution due to their low thermal contrast, which may lead to high retrieval uncertainty. Prior to using this dataset, please reach out to the project lead Zhao-Cheng Zeng for more detailed information. A Python code for reading and plotting the data is also available.

Data access [Zenodo]:

- July 2022 to June 2023: https://doi.org/10.5281/zenodo.17252092
- July 2023 to June 2024: https://doi.org/10.5281/zenodo.17258556
- July 2024 to June 2025: https://doi.org/10.5281/zenodo.17258562

Algorithm reference: Zeng, Z.-C., Lee, L., and Qi, C.: Diurnal carbon monoxide observed from a geostationary infrared hyperspectral sounder: first result from GIIRS on board FengYun-4B, Atmospheric Measurement Techniques, 16, 3059–3083, 2023. https://doi.org/10.5194/amt-16-3059-2023.

Please visit project website for future updates: https://fengyunair.github.io/

2 Variables stored in the file

[**latitude**] : latitude for the observation
[**longitude**] : longitude for the observation
[**hour_utc**] : observation decimal hour UTC+0
[**sza**] : solar zenith angle
[**vza**] : satellite viewing zenith angle
[**ap_skint**] : a priori surface skin temperature [K] from ERA5 reanalysis
[**ret_surft**] : retrieved surface skin temperature [K]
[**ap_surfp**] : a priori surface pressure [hPa] from ERA5 reanalysis
[**midlayer_pres**] : mid-layer atmospheric pressure [hPa]
[**midlayer_temp**] : mid-layer atmospheric temperature [K]
[**ap_co_profile**] : a priori CO partial column for each layer [molecules/cm2]
[**ret_co_col**] : retrieved total column CO [molecules/cm2]
[**ret_co_profile_uncertainty_percentage_bottom11layers**] : retrieval uncertainty (%) for CO partial column profile (for the bottom 11 layers)
[**ret_columnaveragingkernel_bottom11layers**] : averaging kernel matrix [molecule/molecule] for retrieval (for the bottom 11 layers)

3 Pre-filtering of the data

The retrievals have been pre-filtered using the following criteria: (1) RMSE of spectral fitting residual less than 1.5K; (2) Reduced χ^2 less than 2; (3) The absolute difference of the a priori and retrieved surface skin temperature less than 10K.

4 Changes in starting hour and location

The starting hours for the 12 measurement cycles in a day by FY-4B/GIIRS were 00:00, 02:00, 04:00,..., 22:00UTC, respectively, and were changed to 01:00, 03:00, 05:00,. . ., 23:00 UTC, respectively, after 6 September 2022. The hour information in the file names changes accordingly. In addition, FY-4B/GIIRS migrated to 105°E in Feb of 2024 from 133°E. The GIIRS coverage become wider. During the satellite migration, no observations are available (in Feb 2024).

5 Sample Python codes for reading and mapping the data

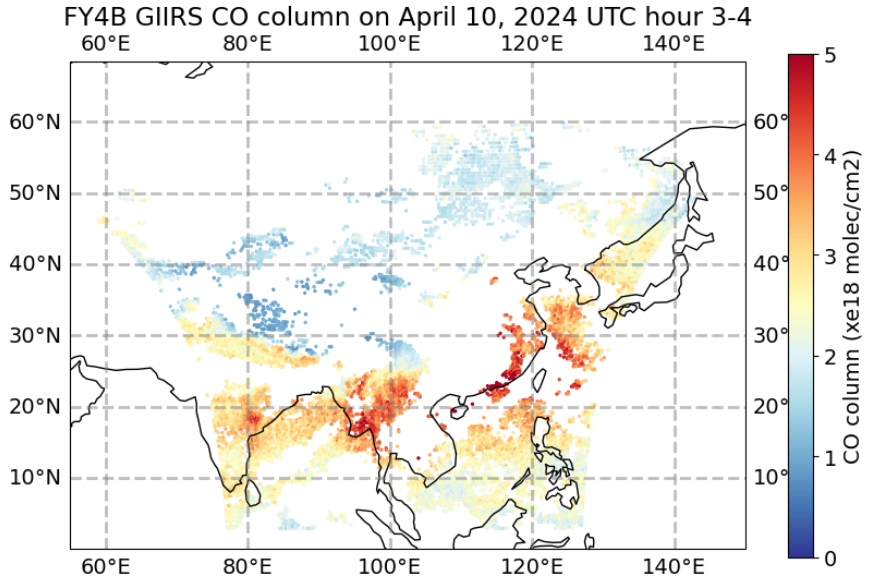
```
# import packages
import numpy as np
import numpy.matlib
import h5py as h5
import cartopy.crs as ccrs
import matplotlib.pyplot as plt

# reading file
YYYYMMDD = '20240410'
HH1Str = '03'
HH2Str = '04'

new_file  = 'FY4B_GIIRS_FYGEOAIR_CO_v1_' + YYYYMMDD + '_h' + HH1Str + '_h' + HH2Str + '.h5'
f = h5.File(new_file,'r')
latitude = f['latitude'][:]
longitude = f['longitude'][:]
hour_utc = f['hour_utc'][:]
sza = f['sza'][:]
vza = f['vza'][:]
ap_skint  = f['ap_skint'][:]
ret_surft = f['ret_surft'][:]
ap_surfp = f['ap_surfp'][:]
midlayer_pres = f['midlayer_pres'][:]
midlayer_temp = f['midlayer_temp'][:]
ap_co_profile = f['ap_co_profile'][:]
ret_co_col = f['ret_co_col'][:]
ret_co_profile_uncertainty_percentage_bottom11layers =
          f['ret_co_profile_uncertainty_percentage_bottom11layers'][:]
ret_columnaveragingkernel_bottom11layers = f['ret_columnaveragingkernel_bottom11layers'][:]
f.close()
```

```
# Mapping ammonia columns
plt.figure(figsize=(10,6))
plt.rc('font',size=14)
proj = ccrs.PlateCarree()
ax = plt.axes(projection=proj)
ax.set_extent([55, 150, 0, 60])
ax.coastlines()
ax.gridlines(crs=ccrs.PlateCarree(), draw_labels=True,linewidth=2, color='gray', alpha=0.5, linestyle='--')
x = longitude
y = latitude
color = ret_co_col/1e18
plt.scatter(x,y,s=2,c=color,vmin=0,vmax=5,cmap='RdYlBu_r')#,vmin=80.0,vmax=140)
plt.colorbar(label='CO column (xe18 molec/cm2)')
plt.title('FY4B GIIRS CO column on April 10, 2024 UTC hour 3-4')
plt.show()
```

```
# Mapping result
```



```
# Plot column averaging kernel for the observation close to [35N,115E]
lon00 = 115
lat00 = 35
idx = np.argmin(np.abs(longitude-lon00)+np.abs(latitude-lat00))
plt.figure(figsize=(5,5))
plt.rc('font', size=14)

## define the number of retrieved layer
print('Number of retrieved layers:
',np.sum(~np.isnan(ret_co_profile_uncertainty_percentage_bottom11layers[idx,:])))
NumRetLayer = 11 # number of retrieved layer, 11 here

cmap = plt.get_cmap('Spectral')
colormap = [cmap(i) for i in np.linspace(0, 1, NumRetLayer)]
for ii in range(NumRetLayer):
    tcolor = colormap[ii]
    plt.plot(ret_columnaveragingkernel_bottom11layers[idx,ii,:NumRetLayer],\
             midlayer_pres[idx,:NumRetLayer],color=tcolor,label='layer '+str(ii))
for ii in range(NumRetLayer):
    tcolor = colormap[ii]
    plt.plot(ret_columnaveragingkernel_bottom11layers[idx,ii,ii],\
             midlayer_pres[idx,ii],'.',color=tcolor,markersize=30)

plt.ylabel('Mid-layer Pressure (hPa)',color='k')
plt.xlim(-0.06,0.5)
plt.ylim(200,1000)
plt.xlabel('Averaging Kernel Rows',color='k')
plt.gca().invert_yaxis()
# add legends and set its box position
plt.legend(['943','815','710','618','538','470','408','356','310',\
            '270','235'],bbox_to_anchor = (1.0, 1.0),fontsize=12) # ,title= 'Pressure (hPa)'
print('Mid layer pressure [hPa]:',midlayer_pres[idx,:NumRetLayer])
```

result [the circles represent the matrix diagonal elements]