```
import numpy as np
import matplotlib.pyplot as plt
import h5py
from cartopy import crs as ccrs
import cartopy.feature as cfeature
import matplotlib.pyplot as plt
from matplotlib import colors
from matplotlib.lines import Line2D
import pandas as pd
import time
import datetime
from pyhdf.SD import SD, SDC
from pyhdf import HDF, VS, V
import os
from cartopy.feature.nightshade import Nightshade
```

Extract a 2D dataset from cloudsat file

```
csat2019119path =
r'/data/debasish/cloudsatdata/cldclasslidar/2019jan/day01/201900104123
4_67528_CS_2B-CLDCLASS-LIDAR_GRANULE_P1_R05_E08_F03.hdf'

from pyhdf.SD import SD, SDC
from pyhdf import HDF, VS, V
path=
r'/data/debasish/cloudsatdata/cldclasslidar/2019jan/day01/201900104123
4_67528_CS_2B-CLDCLASS-LIDAR_GRANULE_P1_R05_E08_F03.hdf'
hdf = SD(path, SDC.READ)
csatfile = SD(path, SDC.READ)

print(csatfile.info())

#Select a particular dataset

dump=csatfile.select('CloudLayerBase')
baseheightarray=dump[:,:]
baseheightarray.shape

(13, 2)

(36950, 10)
```

Extract a 1D dataset from cloudsat file

```
path= "/home/debasish/database/HD5collection/CLOUDSATDATA/CLDCLASS-
LIDAR/2019119064142_69254_CS_2B-CLDCLASS-
LIDAR_GRANULE_P1_R05_E08_F03.hdf"
```

```
hdf = HDF.HDF(path)

latitude,longitude=oneddata(hdf,['Latitude','Longitude']) #Put either
hdf or path, hdf should be in form HDF.HDF(path), not
SD.SD(path,SDC.READ)
latitude[0:3],longitude[0:3]
```

```
([-0.0012714399490505457, -0.01097704004496336, -
0.020682640373706818],
 [-73.12510681152344, -73.12715911865234, -73.12921142578125])
```

```
path1='/home/debasish/database/HD5collection/CLOUDSATDATA/CLDCLASS-
LIDAR/2019001055106_67529_CS_2B-CLDCLASS-
LIDAR_GRANULE_P1_R05_E08_F03.hdf'
path2='/home/debasish/database/HD5collection/CLOUDSATDATA/CLDCLASS-
LIDAR/2019001023402_67527_CS_2B-CLDCLASS-
LIDAR_GRANULE_P1_R05_E08_F03.hdf'
```

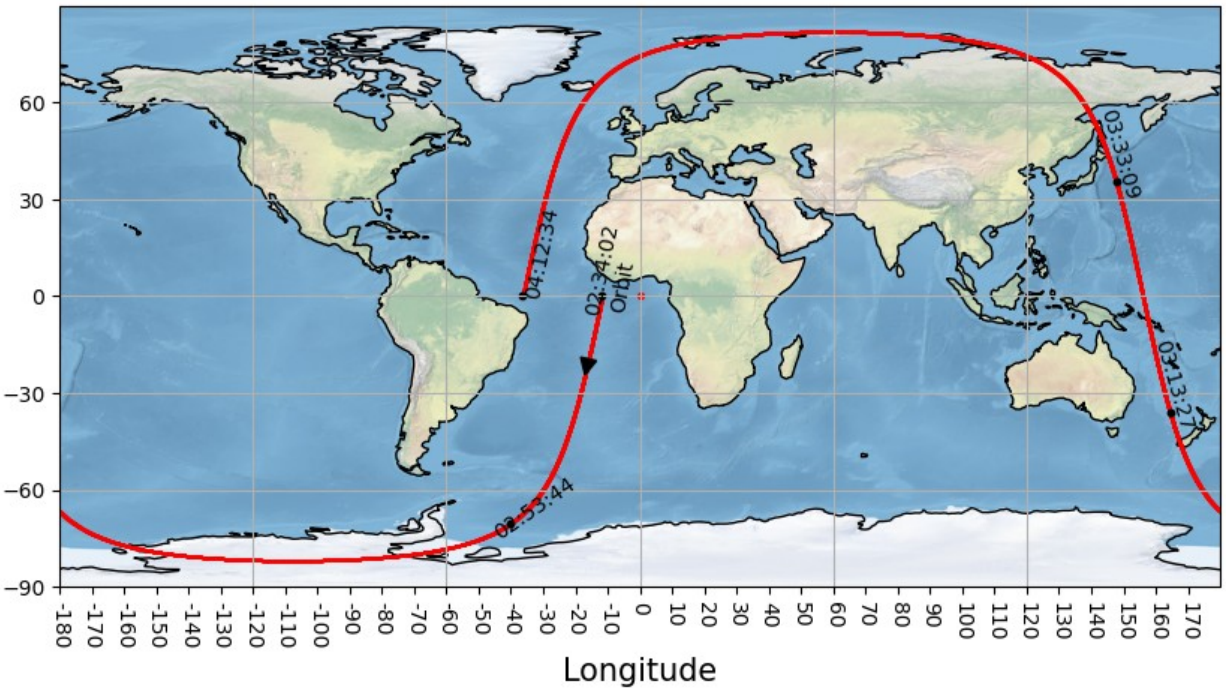
Plotting the groundtrack of CloudSat (Works)

Make a function that takes a cloudsat file and plots it groundtrack. Optionally, many more features can be added, like oribital direction arrow, text, adding timestamps at regular intervals, etc.

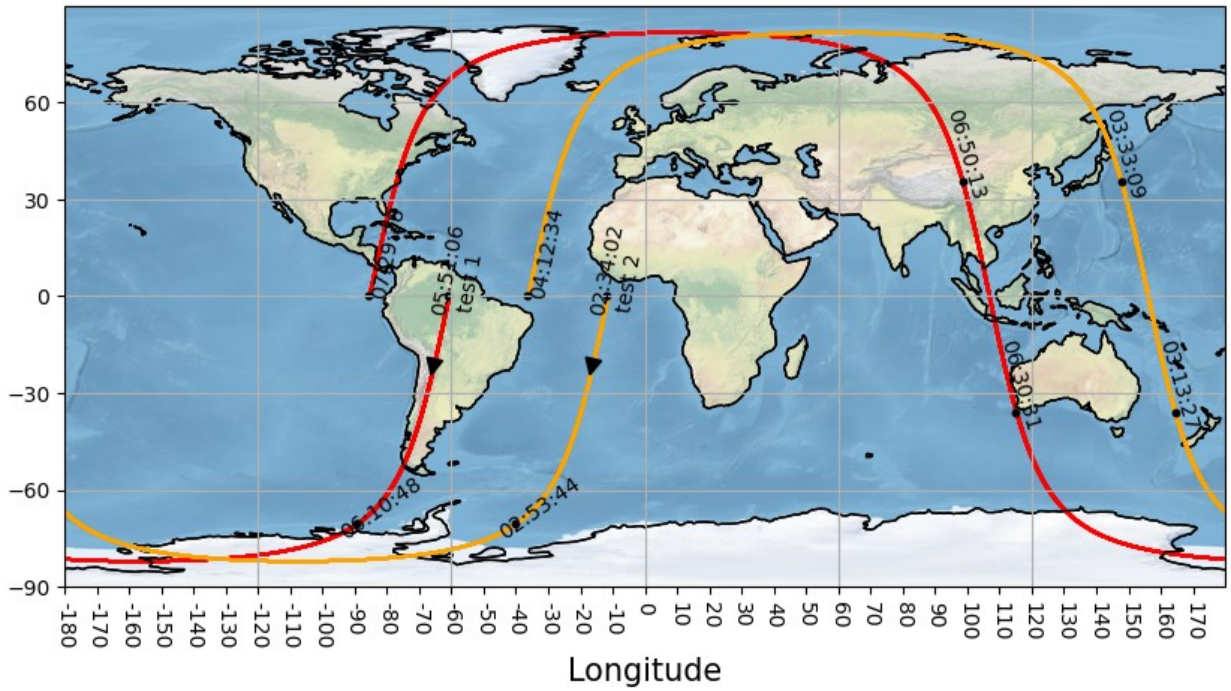
```
def groundtrackplotter(pathorfile,c='r', timestamp=True,
passedaxis=None, baseimage=True, text=None,
s=0.51,linewidths=0.51,arrow=True,arrow_color='black',arrow_width=0.00
01,arrow_head_width=5,arrow_head_length=5
,time_label=True,
time_label_color='black',time_label_fontsize=10,rotation=80,
time_label_count=3):
    if type(pathorfile)==str:
        h=HDF.HDF(pathorfile)
    else:
        h=pathorfile

    def oneddata(searchdatasetlist):
        datasetlist=[]
        for i in searchdatasetlist:
            vs=h.vstart()
            xid=vs.find(i)
            dataid=vs.attach(xid)
            dataid.setfields(i)
            nrecs,_,_,_,_=dataid.inquire()
            data=dataid.read(nRec=nrecs)
            data=list(np.concatenate(data))
            datasetlist.append(data)
            dataid.detach()
```

```
            vs.end()
        return datasetlist

    longitude,latitude=oneddata(['Longitude','Latitude'])
    fig=plt.figure(figsize=(10,10))
    if passedaxis==None:
        ax=plt.axes(projection=ccrs.PlateCarree())
    else:
        ax=passedaxis
    ax.coastlines()
    if baseimage==True:
        ax.stock_img()
    ax.gridlines()
    ax.set_xticks(np.arange(-180, 180, 10), crs=ccrs.PlateCarree())
    ax.set_xticklabels(ax.get_xticks(), rotation=-90)
    ax.set_yticks(np.arange(-90, 90, 30), crs=ccrs.PlateCarree())
    ax.set_xlabel('Longitude',fontsize=15)


ax.scatter(longitude,latitude,transform=ccrs.PlateCarree(),c=c,s=s,lin
ewidths=linewidths)

    if arrow==True:
    #Put arrow at 2000th profile, given there are 36000 profiles
        ax.arrow(longitude[2000],latitude[2000],longitude[2000+5]-
longitude[2000],latitude[2000+5]-latitude[2000],

transform=ccrs.PlateCarree(),color=arrow_color,width=arrow_width,head_
width=arrow_head_width,head_length=arrow_head_length)

    if time_label==True:
        #Put a time label along the track at #time_label_count uniform
intervals

start_time,differential_time=oneddata(['UTC_start','Profile_time'])
        start_time=start_time[0]
        #print(start_time)
        if timestamp==True:

            profile_time=[start_time+i for i in differential_time]
            profile_time_utc=[time.strftime('%H:%M:%S',
time.gmtime(i)) for i in profile_time]
            step=int(len(profile_time_utc)/(time_label_count))
            for i in range(0,int(len(profile_time_utc)-step),step):
                #Put a black dot at the time label


ax.scatter(longitude[i],latitude[i],transform=ccrs.PlateCarree(),color
=time_label_color,s=10)
                ax.text(longitude[i]-6,latitude[i]-
```

```
5,profile_time_utc[i],transform=ccrs.PlateCarree(),color=time_label_co
lor,fontsize=time_label_fontsize,
                 rotation=np.rad2deg(np.arctan((latitude[i+5]-
latitude[i])/(longitude[i+5]-longitude[i]))))

            ax.scatter(longitude[-1],latitude[-
1],transform=ccrs.PlateCarree(),color=time_label_color,s=10)
            ax.text(longitude[-1],latitude[-1],profile_time_utc[-
1],transform=ccrs.PlateCarree(),color=time_label_color,fontsize=time_l
abel_fontsize,
            rotation=np.rad2deg(np.arctan((latitude[-1-5]-latitude[-
1])/(longitude[-1-5]-longitude[-1]))))
    if text != None:
        ax.text(longitude[0]+2,latitude[0]-
4,text,transform=ccrs.PlateCarree(),color='black',fontsize=10,rotation
=80)

    if passedaxis==None:
        plt.show()
        return
    else:
        return ax

#Make a blank axis

fig=plt.figure(figsize=(10,10))
ax=plt.axes(projection=ccrs.PlateCarree())
ax.scatter(0,0,transform=ccrs.PlateCarree(),c='r',s=10.51,linewidths=0
.51)
ax=groundtrackplotter(path2,passedaxis=ax,time_label_count=5,
text="Orbit")
```




```
<Figure size 1000x1000 with 0 Axes>

fig=plt.figure(figsize=(10,10))
ax1=plt.axes(projection=ccrs.PlateCarree())

ax1=groundtrackplotter(path1,passedaxis=ax1,time_label_count=5,text="t
est 1")
ax1=groundtrackplotter(path2,passedaxis=ax1,time_label_count=5,c='oran
ge',s=0.51,linewidths=0.51,text="test 2")
#plt.show()
```



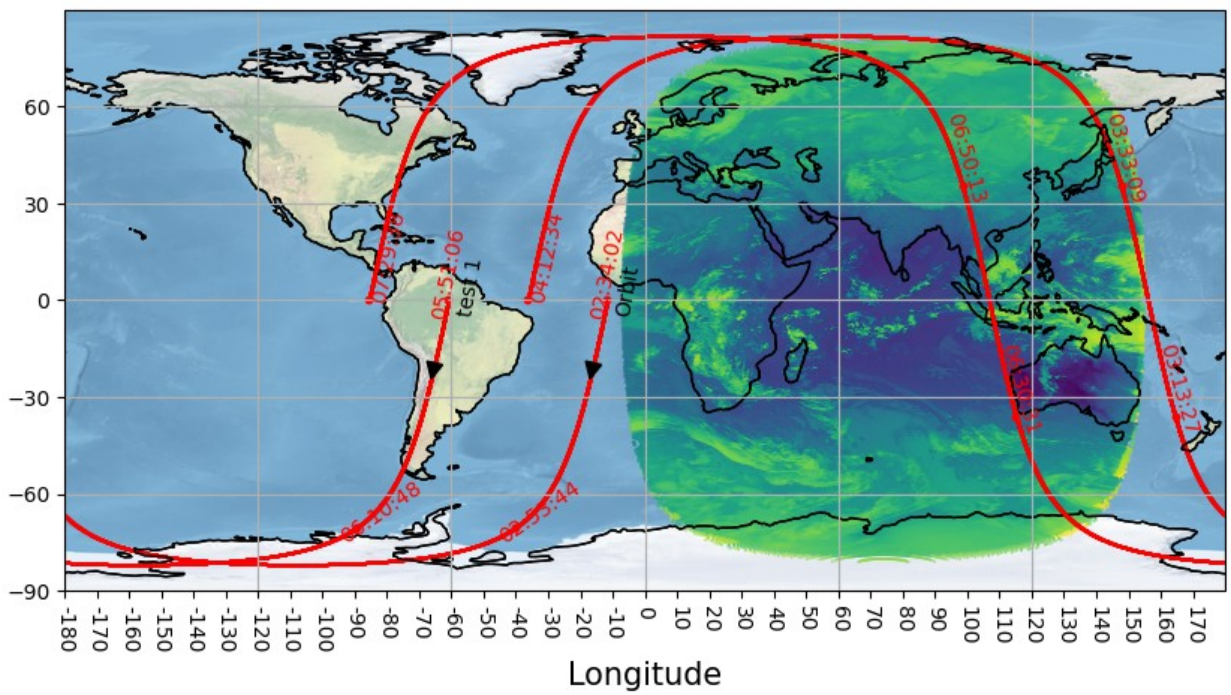

```
<Figure size 1000x1000 with 0 Axes>

<Figure size 1000x1000 with 0 Axes>
```

Plotting multiple groundtracks and INSAT-3D coverage in the same plot by using function (WORKS)

Always plot groundtracks and INSAT-3D coverage in the same plot. Otherwise the plot won't appear.

```
insatpath=r'/home/debasish/database/HD5collection/L1Bdata/INSAT-3DR/
2019 Jan/3RIMG_01JAN2019_0015_L1B_STD_V01R00.h5'
insatfile=h5py.File(insatpath,'r')

#insatfile.keys()
tircount=np.squeeze(np.array(insatfile['IMG_TIR1']))
latitudetir=np.ma.masked_equal(np.array(insatfile['Latitude']),32767)/
100
longitudetir=np.ma.masked_equal(np.array(insatfile['Longitude']),32767
)/100

np.shape(tircount),np.shape(latitudetir),np.shape(longitudetir)
```

```
#Add ax1 made above to a new figure

fig=plt.figure(figsize=(10,10))
ax=plt.axes(projection=ccrs.PlateCarree())
ax.scatter(longitudetir,latitudetir,c=tircount,transform=ccrs.PlateCar
ree(),s=0.1,label='INSAT-3DR')
ax=groundtrackplotter(path2,passedaxis=ax,time_label_count=5,
text="Orbit", baseimage=False,time_label_color='r')
ax=groundtrackplotter(path1,passedaxis=ax,time_label_count=5,text="tes
t 1", baseimage=True, time_label_color='r')
plt.show()
```




```
<Figure size 1000x1000 with 0 Axes>

<Figure size 1000x1000 with 0 Axes>

def data(file):
    file=h5py.File(file,'r')
#    acq_start=str(file.attrs['Acquisition_Start_Time'])
[0:13]+str(file.attrs['Acquisition_Start_Time'])[14:-1]
    acq_start=str(file.attrs['Acquisition_Start_Time'])[2:-1]
    acq_start=acq_start.replace('T','-')

    start_time_obj=datetime.strptime(acq_start,'%d-%b-%Y-%H:%M:%S')
    # return acq_start,start_time_obj
    #print(acq_start,start_time_obj)
    print(acq_start)
    #Print only the date from the start time in dd-mm-yyyy format
```

```
    print(start_time_obj.strftime('%d-%b-%Y'))
    print(start_time_obj.strftime('%H:%M:%S'))



#start,total=data(insatpath)
data(insatpath)
```

```
01-JAN-2019-00:15:23
01-Jan-2019
00:15:23
```

```
round(20.9334,2)
```

```
20.93
```

INSAT-Plotter function

```
from datetime import datetime
def insatplotter(insatfileorpath, figsize=9, axis=None,
baseimage=False, time_label=False, time_label_color='r',
time_label_fontsize=12, parameter='bt',channel='tir1',s=0.1,
label=None, ax=None, stock=False, colorbar=False,
coasttchikness=0.5,globalmap=True):

    if type(insatfileorpath)==str:
        insatfile=h5py.File(insatfileorpath,'r')
    else:
        insatfile=insatfileorpath

    acq_start = str(insatfile.attrs['Acquisition_Start_Time'])[2:-
1].replace('T','-') #To remove a T in the middle of string
    start_time_obj=datetime.strptime(acq_start, "%d-%b-%Y-%H:%M:%S")
    acq_end=str(insatfile.attrs['Acquisition_End_Time'])[2:-
1].replace('T','-')
    end_time_obj=datetime.strptime(acq_end, "%d-%b-%Y-%H:%M:%S")
    acq_time=start_time_obj+(end_time_obj-start_time_obj)/2 #Taking
average of start and end time


vislat=np.ma.masked_equal(np.array(insatfile['Latitude_VIS']),32767)/
1000 #Note how the scale is different for different channels

vislon=np.ma.masked_equal(np.array(insatfile['Longitude_VIS']),32767)/
1000

wvlat=np.ma.masked_equal(np.array(insatfile['Latitude_WV']),32767)/100
```

```
wvlon=np.ma.masked_equal(np.array(insatfile['Longitude_WV']),32767)/
100

tirlat=np.ma.masked_equal(np.array(insatfile['Latitude']),32767)/100

tirlon=np.ma.masked_equal(np.array(insatfile['Longitude']),32767)/100

    viscount=np.ma.masked_equal(np.array(insatfile['IMG_VIS'][0]),0)
    swircount=np.ma.masked_equal(np.array(insatfile['IMG_SWIR'][0]),0)
    mircount=np.ma.masked_equal(np.array(insatfile['IMG_MIR']
[0]),1023)
    wvcount=np.ma.masked_equal(np.array(insatfile['IMG_WV'][0]),1023)
    tir1count=np.ma.masked_equal(np.array(insatfile['IMG_TIR1']
[0]),1023)
    tir2count=np.ma.masked_equal(np.array(insatfile['IMG_TIR2']
[0]),1023)

    visalbedolut=np.array(insatfile['IMG_VIS_ALBEDO'])
    visradlut=np.array(insatfile['IMG_VIS_RADIANCE'])
    swirradlut=np.array(insatfile['IMG_SWIR_RADIANCE'])
    mirradlut=np.array(insatfile['IMG_MIR_RADIANCE'])
    mirtemplut=np.array(insatfile['IMG_MIR_TEMP'])
    wvradlut=np.array(insatfile['IMG_WV_RADIANCE'])
    wvtemplut=np.array(insatfile['IMG_WV_TEMP'])
    tir1radlut=np.array(insatfile['IMG_TIR1_RADIANCE'])
    tir1templut=np.array(insatfile['IMG_TIR1_TEMP'])
    tir2radlut=np.array(insatfile['IMG_TIR2_RADIANCE'])
    tir2templut=np.array(insatfile['IMG_TIR2_TEMP'])

    def counttodata(count,lut): #This function converts the count to
data, data can be radiance, albedo, temperature.
        return lut[count]


    if parameter=='bt' or parameter=='temperature' or
parameter=='temp':
        if channel=='vis':
            raise ValueError('VIS channel does not have BT')
        elif channel=='swir':
            raise ValueError('SWIR channel does not have BT')
        elif channel=='mir':
            plotarray=counttodata(mircount,mirtemplut)
#            plotarray=mirtemplut[mircount]
        elif channel=='tir1':
            plotarray=counttodata(tir1count,tir1templut)
#            plotarray=tir1templut[tir1count]
        elif channel=='tir2':
            plotarray=counttodata(tir2count,tir2templut)
#            plotarray=tir2templut[tir2count]
        elif channel=='wv':
```

```
            plotarray=counttodata(wvcount,wvtemplut)
#            plotarray=wvtemplut[wvcount]
        else:
            raise ValueError('Channel not found')

    if parameter=='rad' or parameter=='radiance':
        if channel=='vis':
            plotarray=counttodata(viscount,visradlut)
#            plotarray=visradlut[viscount]
        elif channel=='swir':
            plotarray=counttodata(swircount,swirradlut)
#            plotarray=swirradlut[swircount]
        elif channel=='mir':
            plotarray=counttodata(mircount,mirradlut)
#            plotarray=mirradlut[mircount]
        elif channel=='tir1':
            plotarray=counttodata(tir1count,tir1radlut)
#            plotarray=tir1radlut[tir1count]
        elif channel=='tir2':
            plotarray=counttodata(tir2count,tir2radlut)
#            plotarray=tir2radlut[tir2count]
        elif channel=='wv':
            plotarray=counttodata(wvcount,wvradlut)
#            plotarray=wvradlut[wvcount]
        else:
            raise ValueError('Channel not found')

    if  parameter=='albedo' or parameter=='alb':
        if channel=='vis':
            plotarray=counttodata(viscount,visalbedolut)
#            plotarray=visalbedolut[viscount]
        else:
            raise ValueError('Albedo is only for VIS channel')

    if parameter=='count':
        if channel=='vis':
            plotarray=viscount
        elif channel=='swir':
            plotarray=swircount
        elif channel=='mir':
            plotarray=mircount
        elif channel=='wv':
            plotarray=wvcount
        elif channel=='tir1':
            plotarray=tir1count
        elif channel== 'tir2':
            plotarray=tir2count
        else:
            raise ValueError("Channel not found")
```

```
    if channel=='vis' or channel=='swir':
        lat=vislat
        lon=vislon
    elif channel=='wv':
        lat=wvlat
        lon=wvlon
    elif channel=='mir' or channel=='tir1' or channel=='tir2':
        lat=tirlat
        lon=tirlon

    if channel=='vis':
        name='IMG_VIS'
    elif channel=='swir':
        name='IMG_SWIR'
    elif channel=='mir':
        name='IMG_MIR'
    elif channel=='wv':
        name='IMG_WV'
    elif channel=='tir1':
        name='IMG_TIR1'
    elif channel=='tir2':
        name='IMG_TIR2'


wavelength=round(float(insatfile[name].attrs['central_wavelength']),2)
    band=round(float(insatfile[name].attrs['bandwidth']),2)
    interval=str(round(wavelength-band/2,2))
+'-'+str(round(wavelength+band/2,2))+' µm'

    if axis==None:
        fig=plt.figure(figsize=(figsize,figsize))

        ax=plt.axes(projection=ccrs.PlateCarree())
        if globalmap:
            ax.set_global()
        ax.gridlines()
        ax.set_xticks(np.arange(-180, 180, 10),
crs=ccrs.PlateCarree())
        ax.set_xticklabels(ax.get_xticks(), rotation=-90)
        ax.set_yticks(np.arange(-90, 90, 30), crs=ccrs.PlateCarree())
        ax.set_xlabel('Longitude',fontsize=15)
        ax.set_ylabel('Latitude',fontsize=15)
#
ax.scatter(lon,lat,c=plotarray,transform=ccrs.PlateCarree(),s=s,label=
label)

scatterplot=ax.scatter(lon,lat,c=plotarray,transform=ccrs.PlateCarree(
```

```
),s=s,label=label, cmap='jet')

#ax.coastlines(color='black',linewidth=coasttchikness,linestyle='solid
')

ax.add_feature(cfeature.NaturalEarthFeature(category='cultural',scale=
'10m',name='admin_0_countries',edgecolor='black',facecolor='none',line
width=coasttchikness))
        if stock:
            ax.stock_img()
        ax.set_title(channel.upper()+'-'+str(parameter.upper())+' Acq.
time - Date='+str(acq_time.strftime('%d-%b-%Y'))+str(' Time=')
+str(acq_time.strftime('%H:%M:%S'))+'(UTC) \n Time interval- '
+str(start_time_obj.strftime('%H:%M:%S'))+'-
'+str(end_time_obj.strftime('%H:%M:%S'))+'\n Cenral
wavelength-'+str(wavelength)+'µm, Channel
range-'+str(interval),color='r')
        cbar=plt.colorbar(scatterplot, ax=ax, pad=0.01, shrink=0.9,
aspect=20)
        if parameter=='albedo' or parameter=='alb':
            text='Visible albedo (%)'
        elif parameter=='bt' or parameter=='temperature' or
parameter=='temp':
            text='Brightness temp. (K)'
        elif parameter=='rad' or parameter=='radiance':
            text='Radiance (mW. $cm^{-2}.sr^{-1}.\mu^{-1} $)'
        cbar.set_label(text,fontsize=15, rotation=270, labelpad=14)
        plt.show()
        return
    #Make channel string all caps

    if axis!=None:
        ax=axis
        ax.gridlines()
        ax.set_xticks(np.arange(-180, 180, 10),
crs=ccrs.PlateCarree())
        ax.set_xticklabels(ax.get_xticks(), rotation=-90)
        ax.set_yticks(np.arange(-90, 90, 30), crs=ccrs.PlateCarree())
        ax.set_xlabel('Longitude',fontsize=15)
        ax.set_ylabel('Latitude',fontsize=15)


scatterplot=ax.scatter(lon,lat,c=plotarray,transform=ccrs.PlateCarree(
),s=s,label=label,cmap='jet')

ax.add_feature(cfeature.NaturalEarthFeature(category='cultural',scale=
'10m',name='admin_0_countries',edgecolor='black',facecolor='none',line
width=coasttchikness))
        ax.set_title(channel.upper()+'-'+str(parameter.upper())+' Acq.
time - Date='+str(acq_time.strftime('%d-%b-%Y'))+str(' Time=')
```

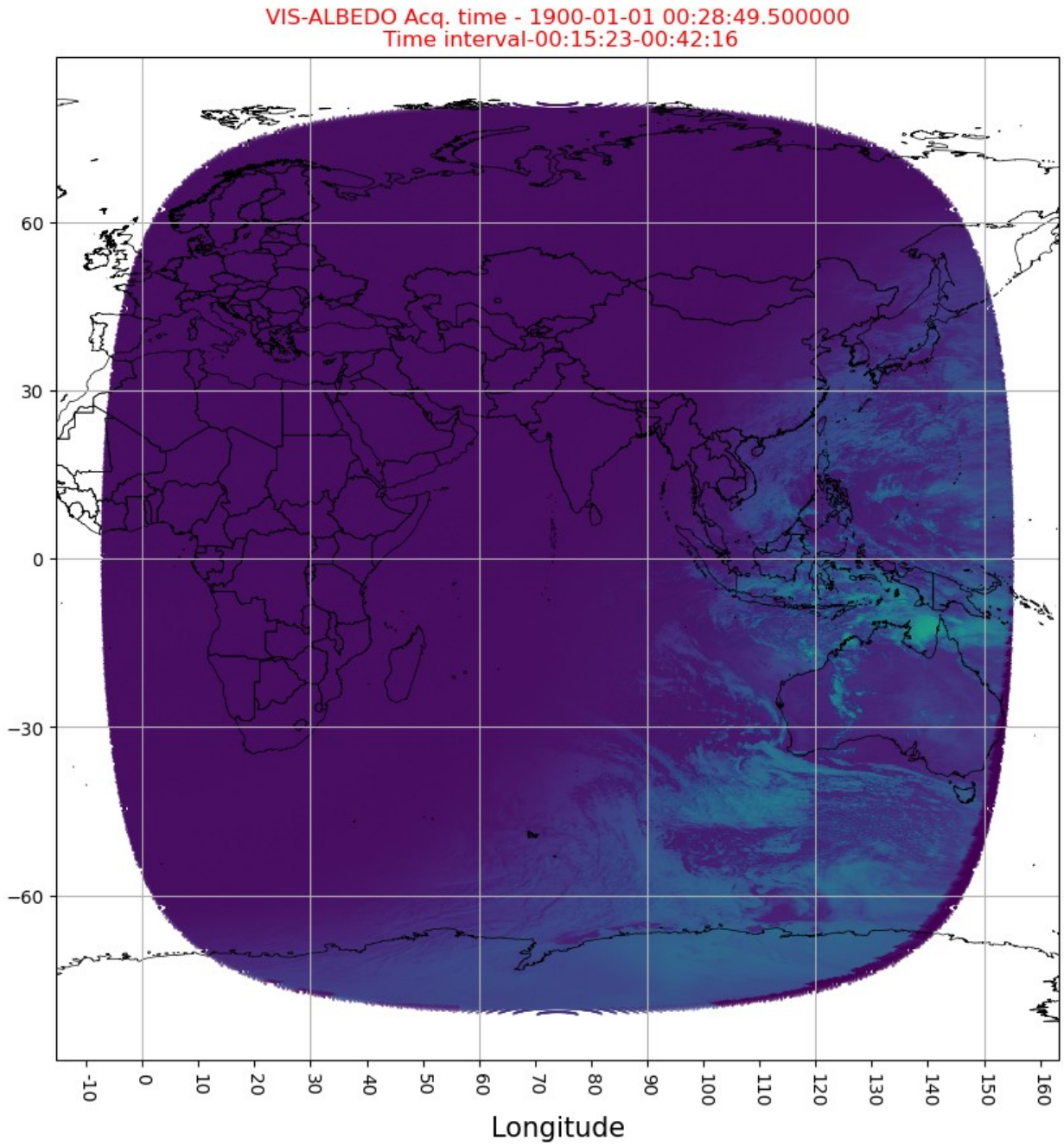
```
+str(acq_time.strftime('%H:%M:%S'))+'(UTC) \n Time interval- '
+str(start_time_obj.strftime('%H:%M:%S'))+' -
'+str(end_time_obj.strftime('%H:%M:%S'))+'\n Central
wavelength-'+str(wavelength)+'µm, Channel
range-'+str(interval),color='r')
        cbar=plt.colorbar(scatterplot, ax=axis, pad=0.01,
shrink=0.805, aspect=20)
        if parameter=='albedo' or parameter=='alb':
            text='Visible albedo (%)'
        elif parameter=='bt' or parameter=='temperature' or
parameter=='temp':
            text='Brightness temp. (K)'
        elif parameter=='rad' or parameter=='radiance':
            text='Radiance (mW. $cm^{-2}.sr^{-1}.\mu^{-1} $)'
        cbar.set_label(text,fontsize=15, rotation=270, labelpad=14)

        return ax
```

#state -> Works with Count only. Gotta check what's wrong with radiance and bt ones

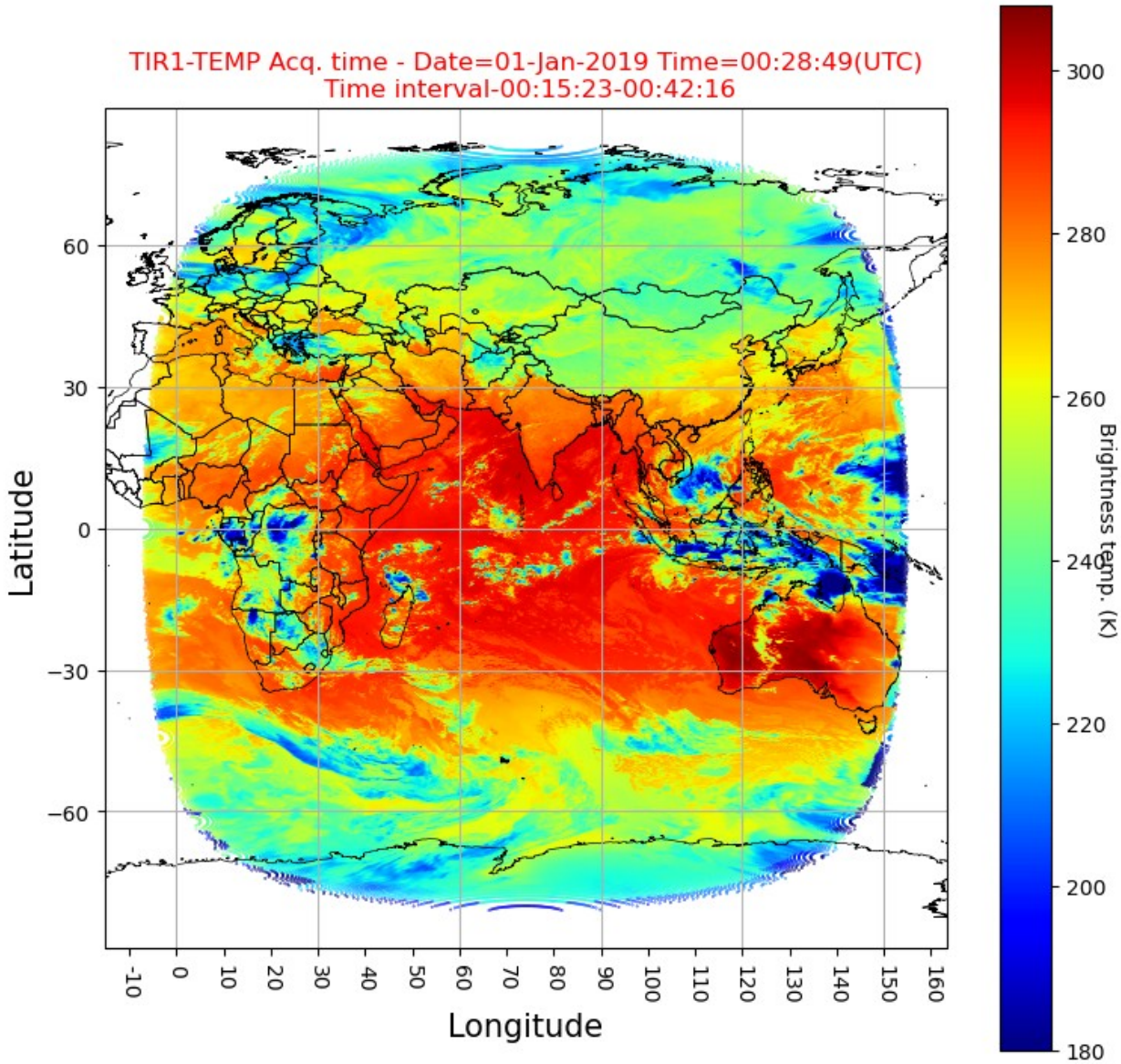
#state -> Works with Count, radiance, brightness temperature.

Average TIR plot runntime -1 min, vis plot runtime - 13 min

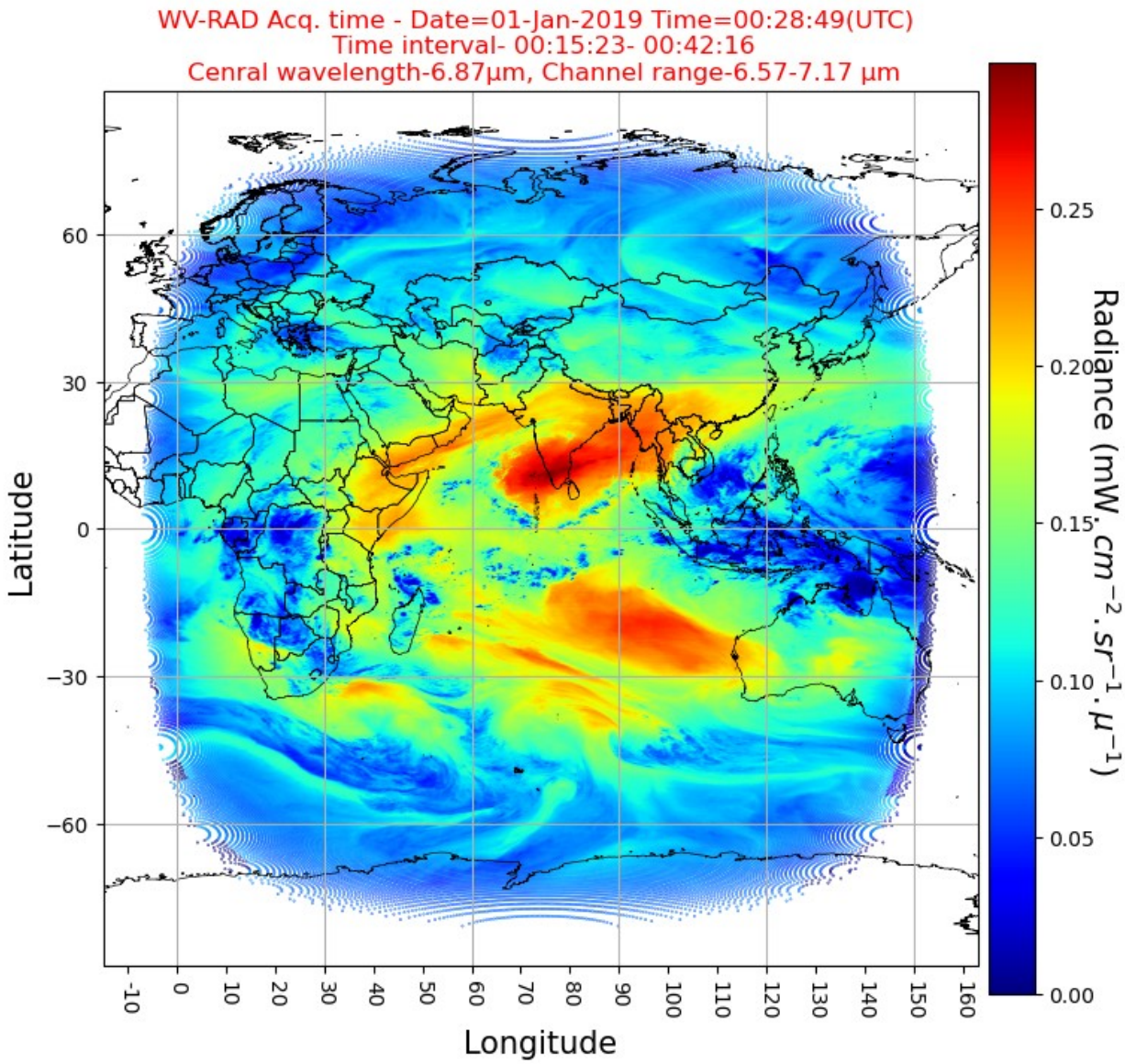
```
insatpath=r'/home/debasish/database/HD5collection/L1Bdata/INSAT-3DR/
2019 Jan/3RIMG_01JAN2019_0015_L1B_STD_V01R00.h5'
insatplotter(insatpath,channel='vis',parameter='albedo',stock=False,gl
obalmap=False)
```

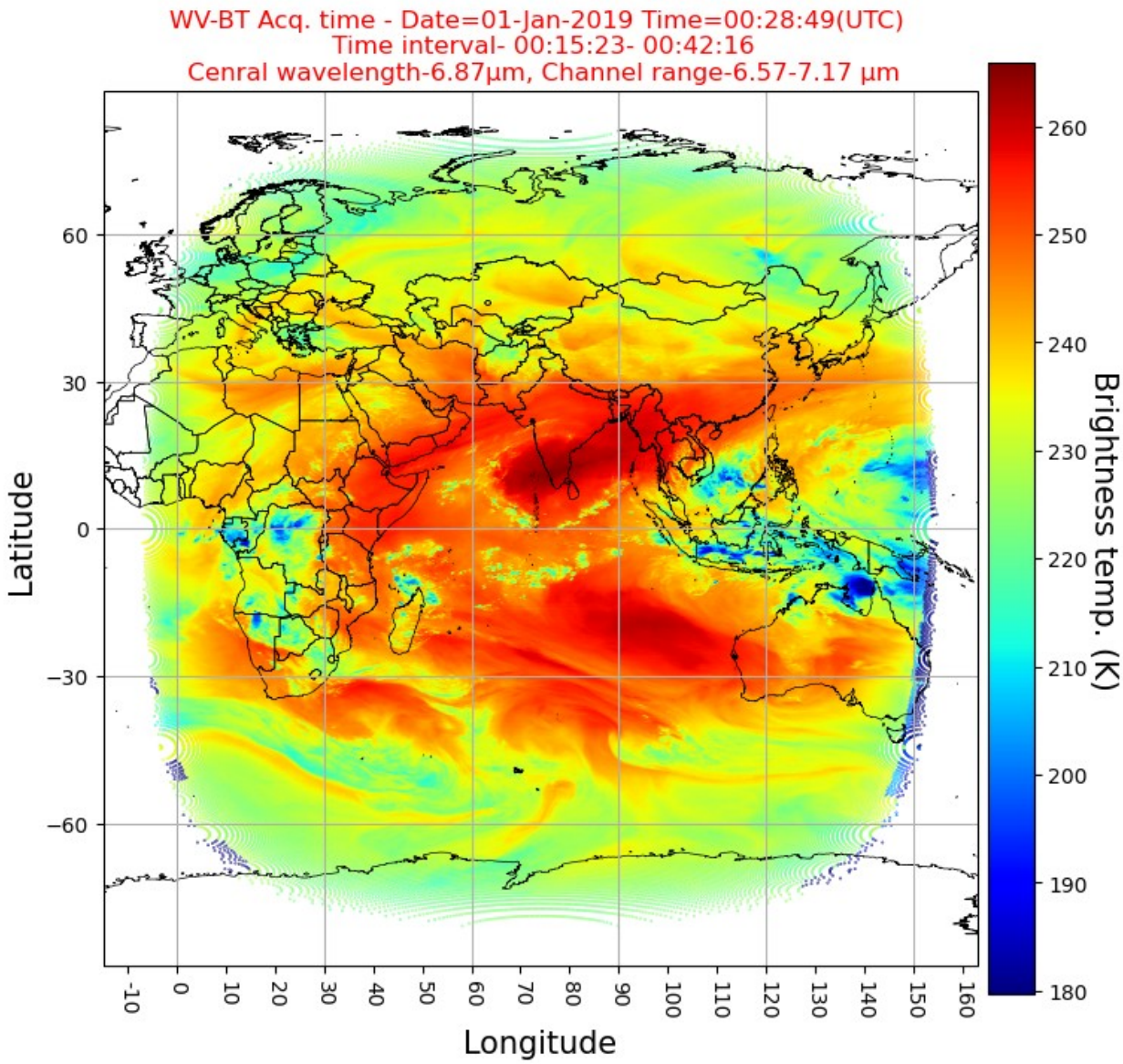



```
insatpath=r'/home/debasish/database/HD5collection/L1B
data/INSAT-3DR/2019 Jan/3RIMG_01JAN2019_0015_L1B_STD_V01R00.h5'
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False, )
```



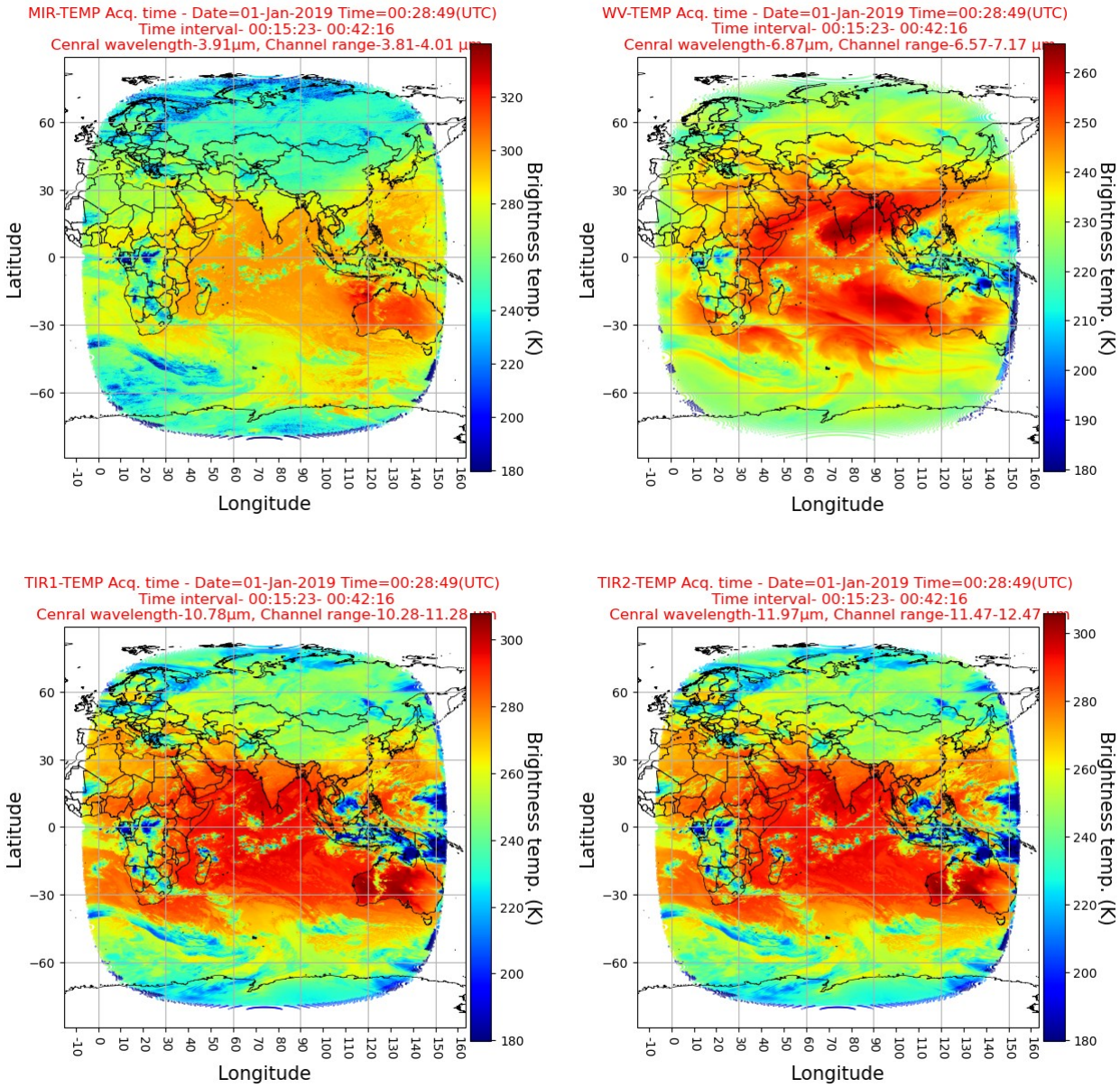

```
insatpath=r'/home/debasish/database/HD5collection/L1B
data/INSAT-3DR/2019 Jan/3RIMG_01JAN2019_0015_L1B_STD_V01R00.h5'
insatplotter(insatpath,channel='wv',parameter='rad',stock=False,global
map=False, )
insatplotter(insatpath,channel='wv',parameter='bt',stock=False,globalm
ap=False, )
```


WV-RAD Acq. time - Date=01-Jan-2019 Time=00:28:49(UTC)
Time interval- 00:15:23- 00:42:16
Cenral wavelength-6.87µm, Channel range-6.57-7.17 µm
Latitude
Longitude
Radiance (mW.cm^{-2}.sr^{-1}.µ^{-1})
60
30
0
−30
−60
-10
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
0.25
0.20
0.15
0.10
0.05
0.00



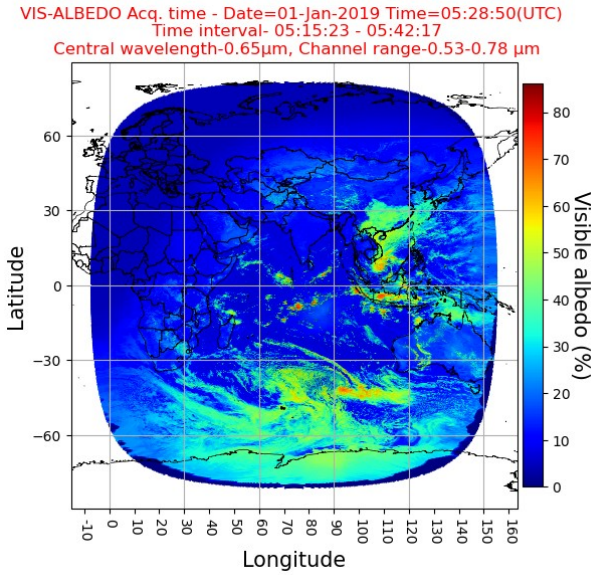

```
#For a single file, make the plots for all channels
# except SWIR and VIS, in a single figure but in different subplots

insatpath=r'/home/debasish/database/HD5collection/L1B
data/INSAT-3DR/2019 Jan/3RIMG_01JAN2019_0015_L1B_STD_V01R00.h5'

fig=plt.figure(figsize=(15,15))
ax1=plt.subplot(2,2,1,projection=ccrs.PlateCarree())
ax2=plt.subplot(2,2,2,projection=ccrs.PlateCarree())
ax3=plt.subplot(2,2,3,projection=ccrs.PlateCarree())
ax4=plt.subplot(2,2,4,projection=ccrs.PlateCarree())

insatplotter(insatpath,channel='mir',parameter='temp',stock=False,glob
almap=False,axis=ax1,figsize=6)
```

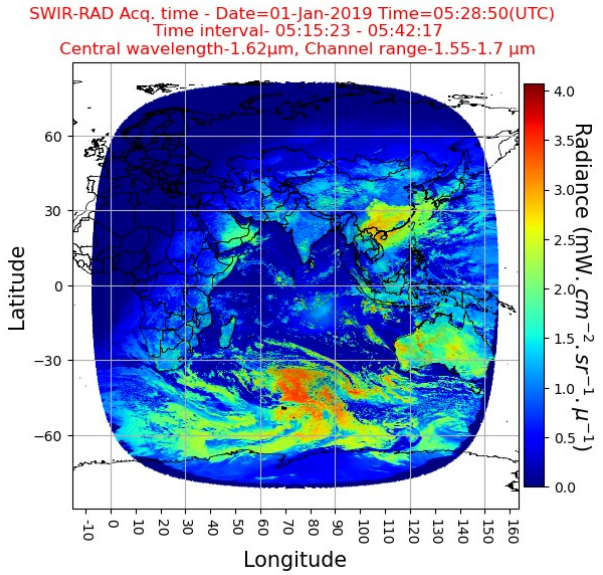
```
insatplotter(insatpath,channel='wv',parameter='temp',stock=False,globa
lmap=False,axis=ax2,figsize=6)
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False,axis=ax3,figsize=6)
insatplotter(insatpath,channel='tir2',parameter='temp',stock=False,glo
balmap=False,axis=ax4,figsize=6)
plt.show()
```

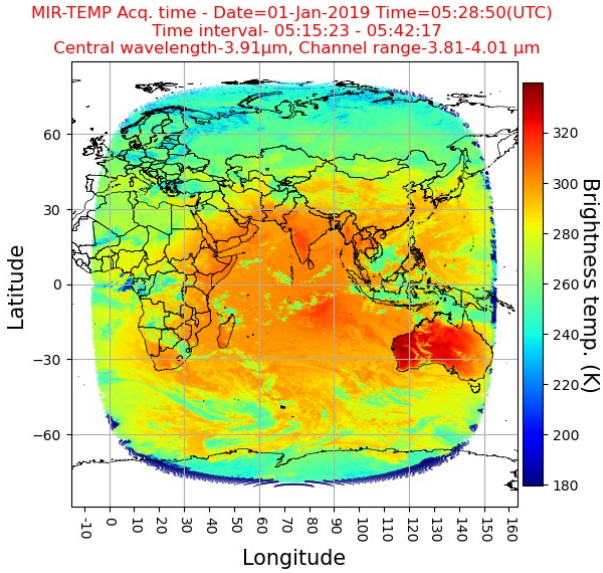


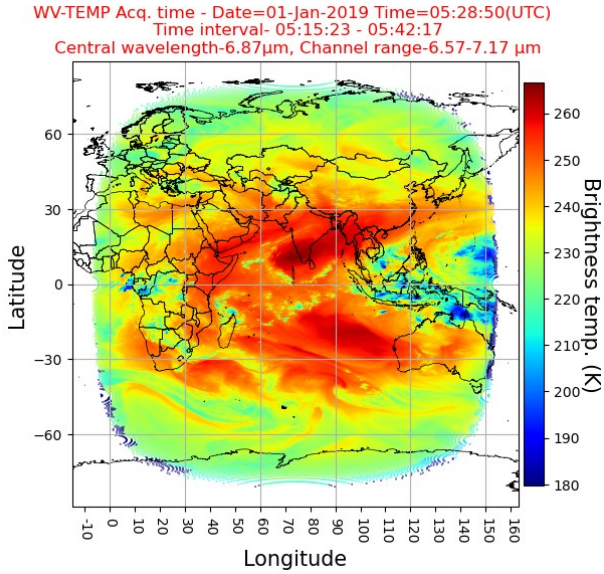

```
#Plot for all the channels in a file
insatpath=r'/home/debasish/database/HD5collection/L1B
data/INSAT-3DR/2019 Jan/3RIMG_01JAN2019_0515_L1B_STD_V01R00.h5'

fig=plt.figure(figsize=(15,22))
```

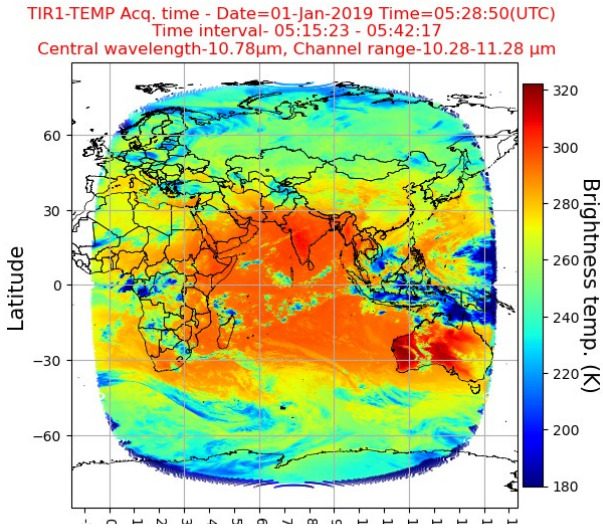
```
ax1=plt.subplot(3,2,1,projection=ccrs.PlateCarree())
ax2=plt.subplot(3,2,2,projection=ccrs.PlateCarree())
ax3=plt.subplot(3,2,3,projection=ccrs.PlateCarree())
ax4=plt.subplot(3,2,4,projection=ccrs.PlateCarree())
ax5=plt.subplot(3,2,5,projection=ccrs.PlateCarree())
ax6=plt.subplot(3,2,6,projection=ccrs.PlateCarree())

insatplotter(insatpath,channel='vis',parameter='albedo',stock=False,gl
obalmap=False,axis=ax1,figsize=6)
insatplotter(insatpath,channel='swir',parameter='rad',stock=False,glob
almap=False,axis=ax2,figsize=6)
insatplotter(insatpath,channel='mir',parameter='temp',stock=False,glob
almap=False,axis=ax3,figsize=6)
insatplotter(insatpath,channel='wv',parameter='temp',stock=False,globa
lmap=False,axis=ax4,figsize=6)
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False,axis=ax5,figsize=6)
insatplotter(insatpath,channel='tir2',parameter='temp',stock=False,glo
balmap=False,axis=ax6,figsize=6)
plt.show()
```

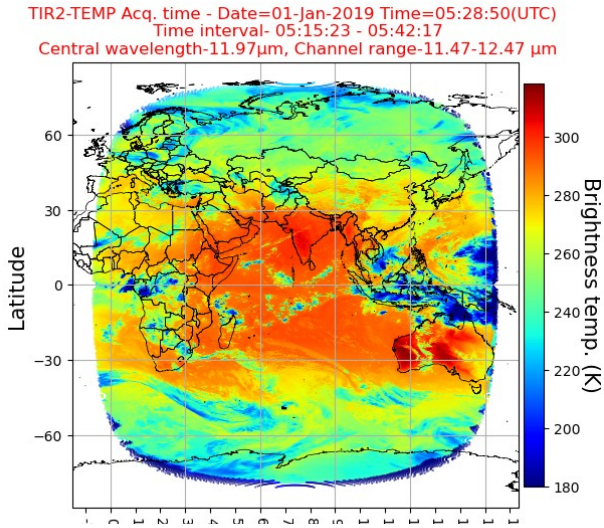

VIS-ALBEDO Acq. time - Date=01-Jan-2019 Time=05:28:50(UTC)
Time interval- 05:15:23 - 05:42:17
Central wavelength-0.65µm, Channel range-0.53-0.78 µm
Latitude
Longitude
Visible albedo (%)


SWIR-RAD Acq. time - Date=01-Jan-2019 Time=05:28:50(UTC)
Time interval- 05:15:23 - 05:42:17
Central wavelength-1.62µm, Channel range-1.55-1.7 µm
Latitude
Longitude
Radiance (mW.cm^{-2}.sr^{-1}.µ^{-1})


MIR-TEMP Acq. time - Date=01-Jan-2019 Time=05:28:50(UTC)
Time interval- 05:15:23 - 05:42:17
Central wavelength-3.91µm, Channel range-3.81-4.01 µm
Latitude
Longitude
Brightness temp. (K)

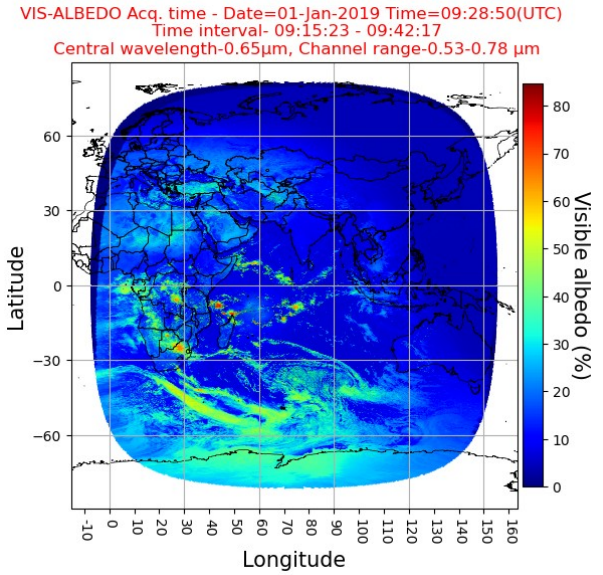

WV-TEMP Acq. time - Date=01-Jan-2019 Time=05:28:50(UTC)
Time interval- 05:15:23 - 05:42:17
Central wavelength-6.87µm, Channel range-6.57-7.17 µm
Latitude
Longitude
Brightness temp. (K)

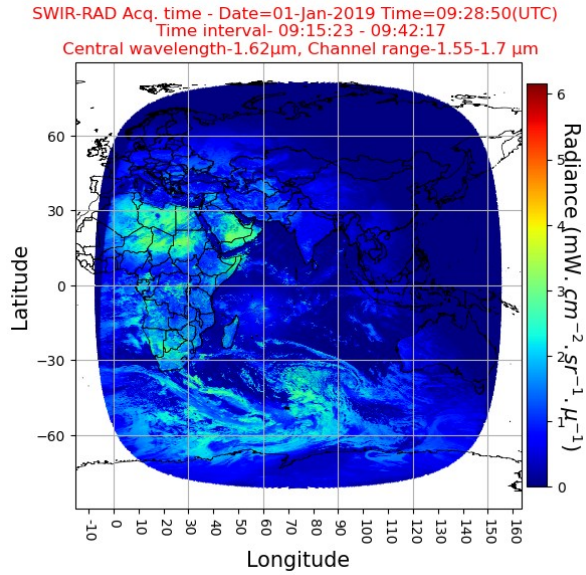

TIR1-TEMP Acq. time - Date=01-Jan-2019 Time=05:28:50(UTC)
Time interval- 05:15:23 - 05:42:17
Central wavelength-10.78µm, Channel range-10.28-11.28 µm
Latitude
Brightness temp. (K)

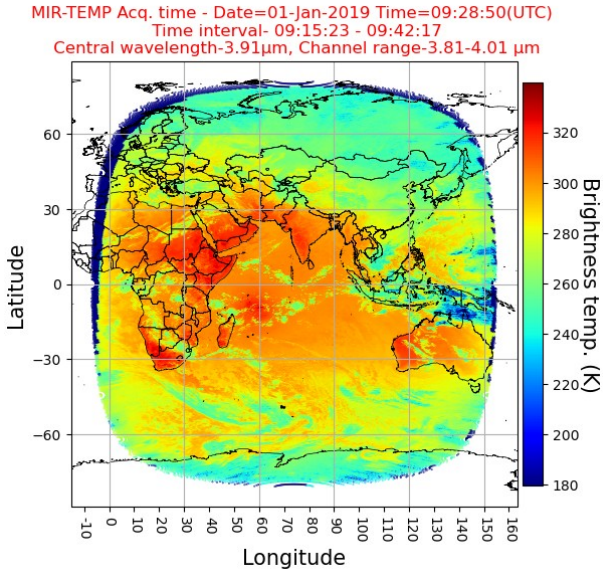

TIR2-TEMP Acq. time - Date=01-Jan-2019 Time=05:28:50(UTC)
Time interval- 05:15:23 - 05:42:17
Central wavelength-11.97µm, Channel range-11.47-12.47 µm
Latitude
Brightness temp. (K)

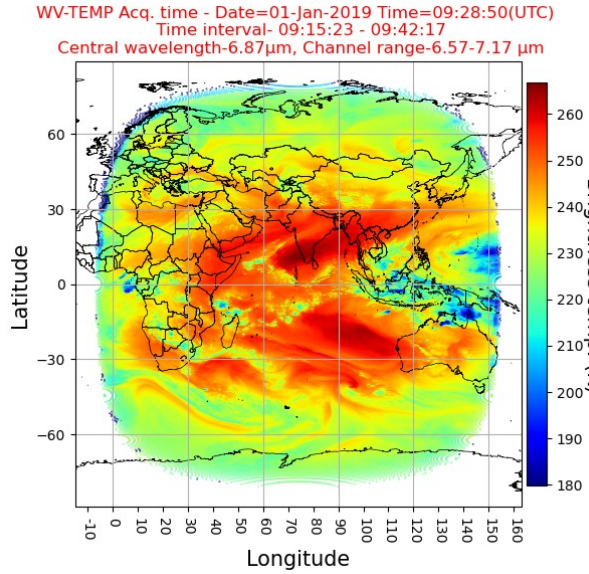
```
#Plot for all the channels in a file
insatpath=r'/data/debasish/insatdata/l1b/2019/2019jan/day01/3RIMG_01JA
N2019_0915_L1B_STD_V01R00.h5'

fig=plt.figure(figsize=(15,22))
ax1=plt.subplot(3,2,1,projection=ccrs.PlateCarree())
ax2=plt.subplot(3,2,2,projection=ccrs.PlateCarree())
ax3=plt.subplot(3,2,3,projection=ccrs.PlateCarree())
ax4=plt.subplot(3,2,4,projection=ccrs.PlateCarree())
ax5=plt.subplot(3,2,5,projection=ccrs.PlateCarree())
ax6=plt.subplot(3,2,6,projection=ccrs.PlateCarree())

insatplotter(insatpath,channel='vis',parameter='albedo',stock=False,gl
obalmap=False,axis=ax1,figsize=6)
print("Done")
insatplotter(insatpath,channel='swir',parameter='rad',stock=False,glob
almap=False,axis=ax2,figsize=6)
print("Done")
insatplotter(insatpath,channel='mir',parameter='temp',stock=False,glob
almap=False,axis=ax3,figsize=6)
print("Done")
insatplotter(insatpath,channel='wv',parameter='temp',stock=False,globa
lmap=False,axis=ax4,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False,axis=ax5,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir2',parameter='temp',stock=False,glo
balmap=False,axis=ax6,figsize=6)
print("Done")
plt.show()
```

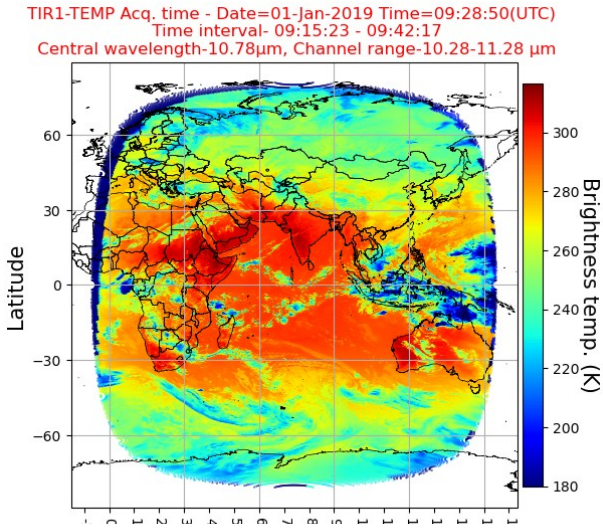
```
Done
Done
Done
Done
Done
Done
```

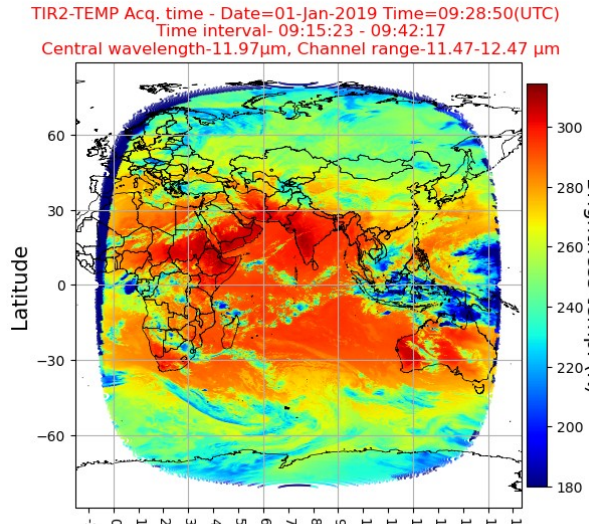

VIS-ALBEDO Acq. time - Date=01-Jan-2019 Time=09:28:50(UTC)
Time interval- 09:15:23 - 09:42:17
Central wavelength-0.65µm, Channel range-0.53-0.78 µm
Latitude
Longitude
Visible albedo (%)


SWIR-RAD Acq. time - Date=01-Jan-2019 Time=09:28:50(UTC)
Time interval- 09:15:23 - 09:42:17
Central wavelength-1.62µm, Channel range-1.55-1.7 µm
Latitude
Longitude
Radiance (mW.cm^{-2}.sr^{-1}.µ^{-1})


MIR-TEMP Acq. time - Date=01-Jan-2019 Time=09:28:50(UTC)
Time interval- 09:15:23 - 09:42:17
Central wavelength-3.91µm, Channel range-3.81-4.01 µm
Latitude
Longitude
Brightness temp. (K)

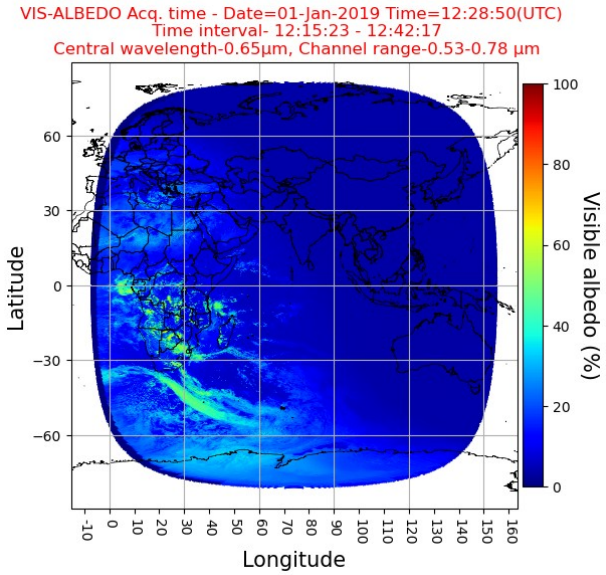

WV-TEMP Acq. time - Date=01-Jan-2019 Time=09:28:50(UTC)
Time interval- 09:15:23 - 09:42:17
Central wavelength-6.87µm, Channel range-6.57-7.17 µm
Latitude
Longitude
Brightness temp. (K)

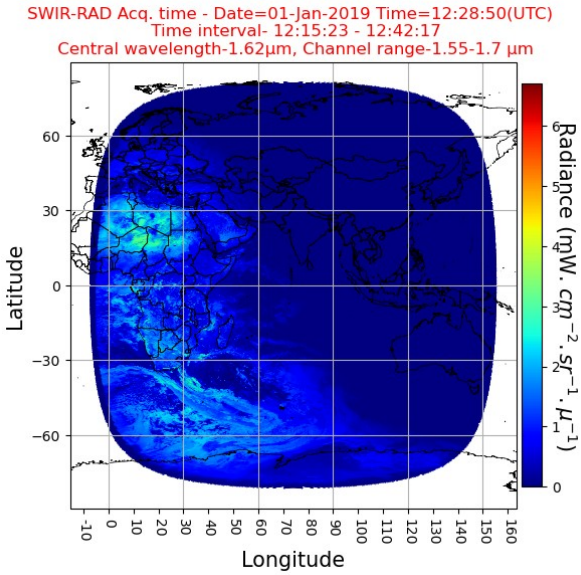

TIR1-TEMP Acq. time - Date=01-Jan-2019 Time=09:28:50(UTC)
Time interval- 09:15:23 - 09:42:17
Central wavelength-10.78µm, Channel range-10.28-11.28 µm
Latitude
Brightness temp. (K)

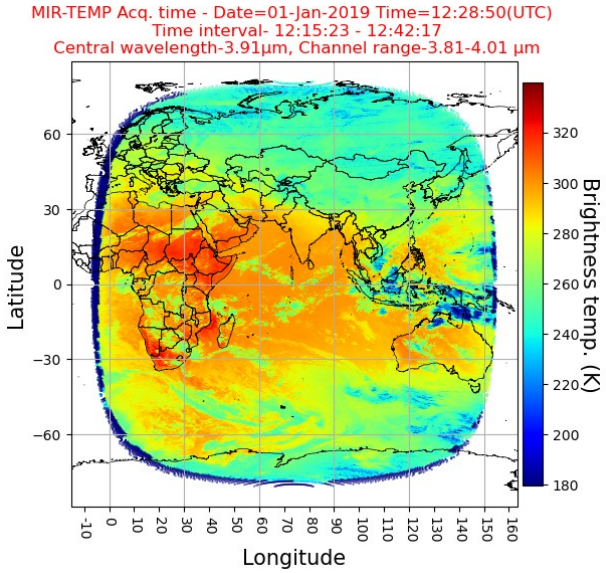

TIR2-TEMP Acq. time - Date=01-Jan-2019 Time=09:28:50(UTC)
Time interval- 09:15:23 - 09:42:17
Central wavelength-11.97µm, Channel range-11.47-12.47 µm
Latitude
Brightness temp. (K)

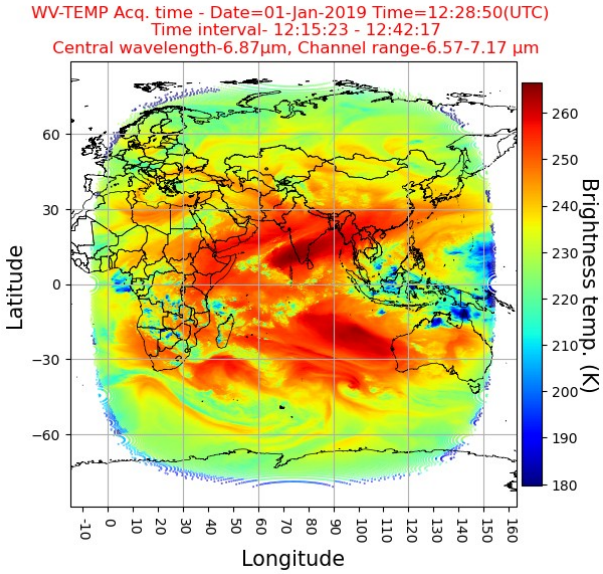
```
#Plot for all the channels in a file
insatpath=r'/data/debasish/insatdata/l1b/2019/2019jan/day01/3RIMG_01JA
N2019_1215_L1B_STD_V01R00.h5'

fig=plt.figure(figsize=(15,22))
ax1=plt.subplot(3,2,1,projection=ccrs.PlateCarree())
ax2=plt.subplot(3,2,2,projection=ccrs.PlateCarree())
ax3=plt.subplot(3,2,3,projection=ccrs.PlateCarree())
ax4=plt.subplot(3,2,4,projection=ccrs.PlateCarree())
ax5=plt.subplot(3,2,5,projection=ccrs.PlateCarree())
ax6=plt.subplot(3,2,6,projection=ccrs.PlateCarree())

insatplotter(insatpath,channel='vis',parameter='albedo',stock=False,gl
obalmap=False,axis=ax1,figsize=6)
print("Done")
insatplotter(insatpath,channel='swir',parameter='rad',stock=False,glob
almap=False,axis=ax2,figsize=6)
print("Done")
insatplotter(insatpath,channel='mir',parameter='temp',stock=False,glob
almap=False,axis=ax3,figsize=6)
print("Done")
insatplotter(insatpath,channel='wv',parameter='temp',stock=False,globa
lmap=False,axis=ax4,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False,axis=ax5,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir2',parameter='temp',stock=False,glo
balmap=False,axis=ax6,figsize=6)
print("Done")
plt.show()
```

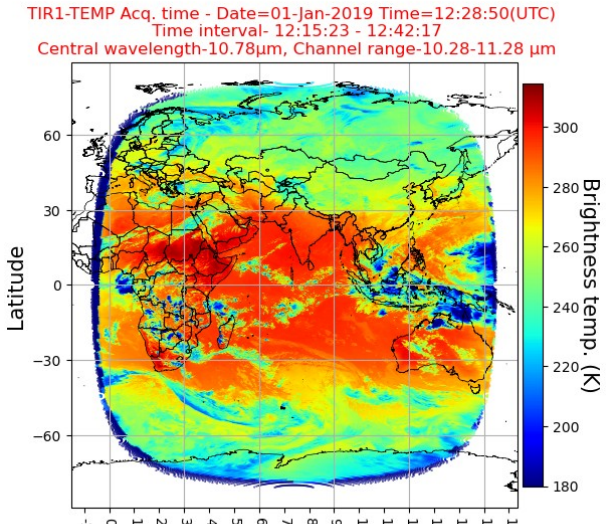
```
Done
Done
Done
Done
Done
Done
```

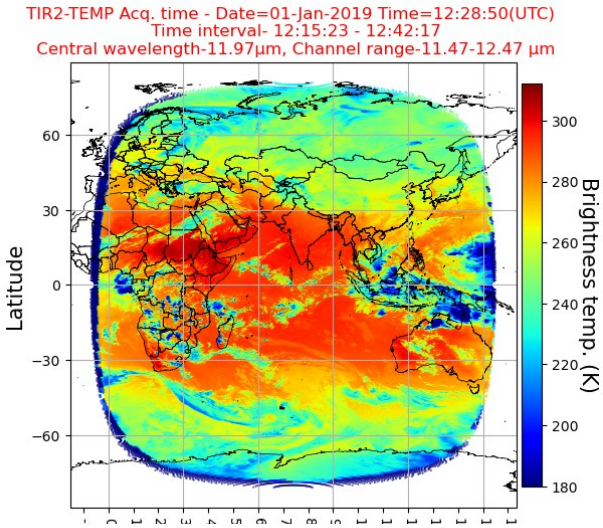

VIS-ALBEDO Acq. time - Date=01-Jan-2019 Time=12:28:50(UTC)
Time interval- 12:15:23 - 12:42:17
Central wavelength-0.65µm, Channel range-0.53-0.78 µm
Latitude
Longitude
Visible albedo (%)


SWIR-RAD Acq. time - Date=01-Jan-2019 Time=12:28:50(UTC)
Time interval- 12:15:23 - 12:42:17
Central wavelength-1.62µm, Channel range-1.55-1.7 µm
Latitude
Longitude
Radiance (mW. cm^{-2}. sr^{-1}. µ^{-1})


MIR-TEMP Acq. time - Date=01-Jan-2019 Time=12:28:50(UTC)
Time interval- 12:15:23 - 12:42:17
Central wavelength-3.91µm, Channel range-3.81-4.01 µm
Latitude
Longitude
Brightness temp. (K)

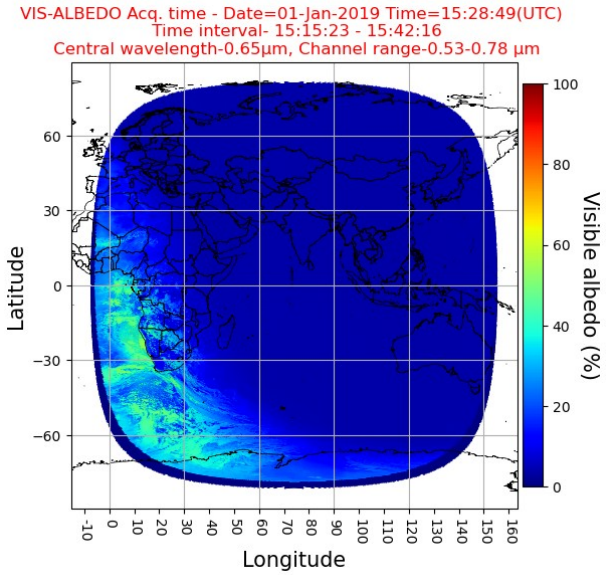

WV-TEMP Acq. time - Date=01-Jan-2019 Time=12:28:50(UTC)
Time interval- 12:15:23 - 12:42:17
Central wavelength-6.87µm, Channel range-6.57-7.17 µm
Latitude
Longitude
Brightness temp. (K)

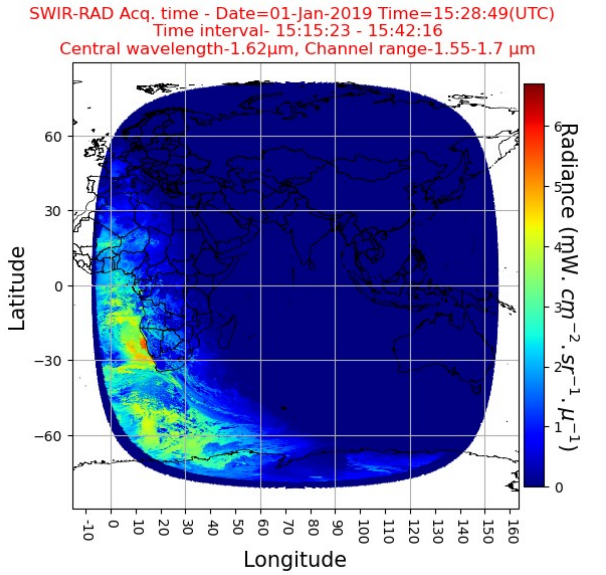

TIR1-TEMP Acq. time - Date=01-Jan-2019 Time=12:28:50(UTC)
Time interval- 12:15:23 - 12:42:17
Central wavelength-10.78µm, Channel range-10.28-11.28 µm
Latitude
Brightness temp. (K)

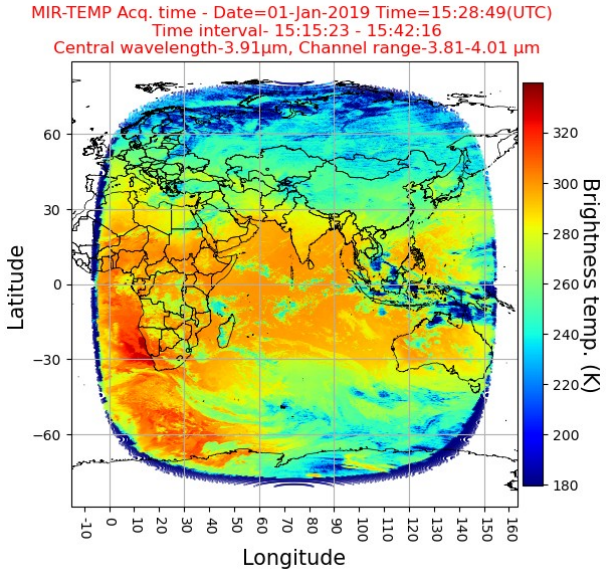

TIR2-TEMP Acq. time - Date=01-Jan-2019 Time=12:28:50(UTC)
Time interval- 12:15:23 - 12:42:17
Central wavelength-11.97µm, Channel range-11.47-12.47 µm
Latitude
Brightness temp. (K)

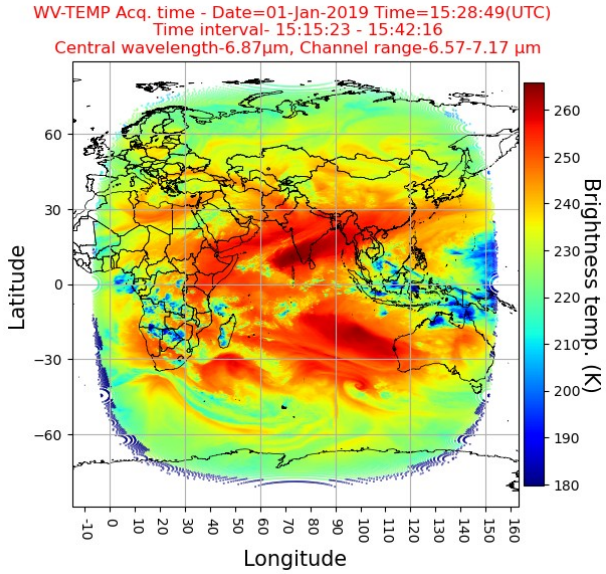
```
#Plot for all the channels in a file
insatpath=r'/data/debasish/insatdata/l1b/2019/2019jan/day01/3RIMG_01JA
N2019_1515_L1B_STD_V01R00.h5'

fig=plt.figure(figsize=(15,22))
ax1=plt.subplot(3,2,1,projection=ccrs.PlateCarree())
ax2=plt.subplot(3,2,2,projection=ccrs.PlateCarree())
ax3=plt.subplot(3,2,3,projection=ccrs.PlateCarree())
ax4=plt.subplot(3,2,4,projection=ccrs.PlateCarree())
ax5=plt.subplot(3,2,5,projection=ccrs.PlateCarree())
ax6=plt.subplot(3,2,6,projection=ccrs.PlateCarree())

insatplotter(insatpath,channel='vis',parameter='albedo',stock=False,gl
obalmap=False,axis=ax1,figsize=6)
print("Done")
insatplotter(insatpath,channel='swir',parameter='rad',stock=False,glob
almap=False,axis=ax2,figsize=6)
print("Done")
insatplotter(insatpath,channel='mir',parameter='temp',stock=False,glob
almap=False,axis=ax3,figsize=6)
print("Done")
insatplotter(insatpath,channel='wv',parameter='temp',stock=False,globa
lmap=False,axis=ax4,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False,axis=ax5,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir2',parameter='temp',stock=False,glo
balmap=False,axis=ax6,figsize=6)
print("Done")
plt.show()
```

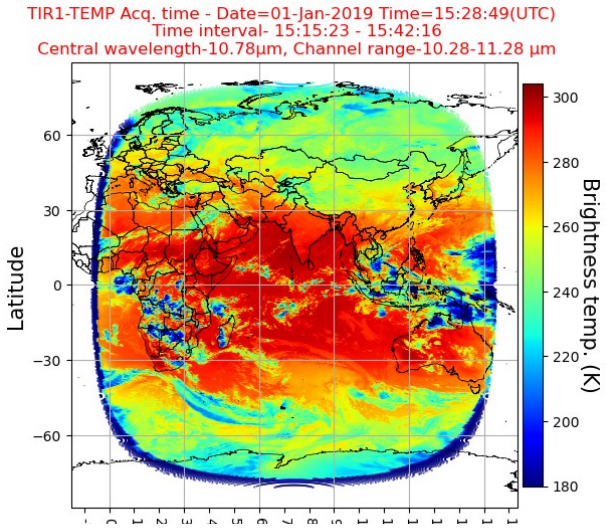
```
Done
Done
Done
Done
Done
Done
```

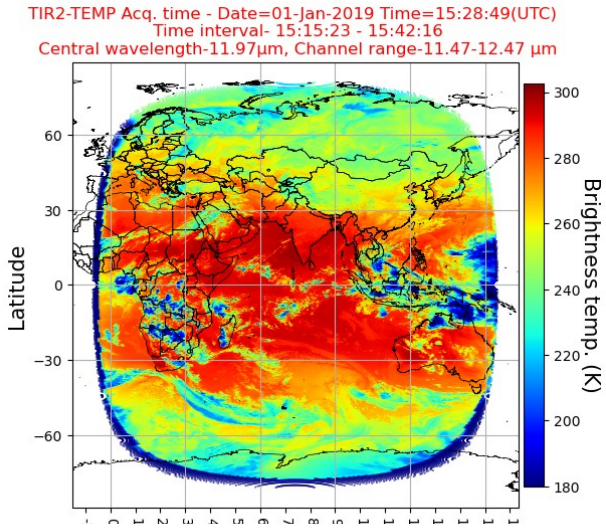

VIS-ALBEDO Acq. time - Date=01-Jan-2019 Time=15:28:49(UTC)
Time interval- 15:15:23 - 15:42:16
Central wavelength-0.65µm, Channel range-0.53-0.78 µm
Latitude
Longitude
Visible albedo (%)


SWIR-RAD Acq. time - Date=01-Jan-2019 Time=15:28:49(UTC)
Time interval- 15:15:23 - 15:42:16
Central wavelength-1.62µm, Channel range-1.55-1.7 µm
Latitude
Longitude
Radiance (mW. cm^{-2}. sr^{-1}. µ^{-1})


MIR-TEMP Acq. time - Date=01-Jan-2019 Time=15:28:49(UTC)
Time interval- 15:15:23 - 15:42:16
Central wavelength-3.91µm, Channel range-3.81-4.01 µm
Latitude
Longitude
Brightness temp. (K)

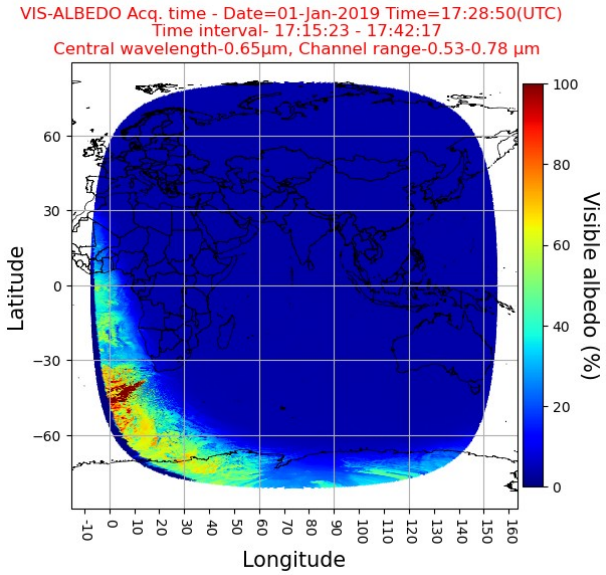

WV-TEMP Acq. time - Date=01-Jan-2019 Time=15:28:49(UTC)
Time interval- 15:15:23 - 15:42:16
Central wavelength-6.87µm, Channel range-6.57-7.17 µm
Latitude
Longitude
Brightness temp. (K)

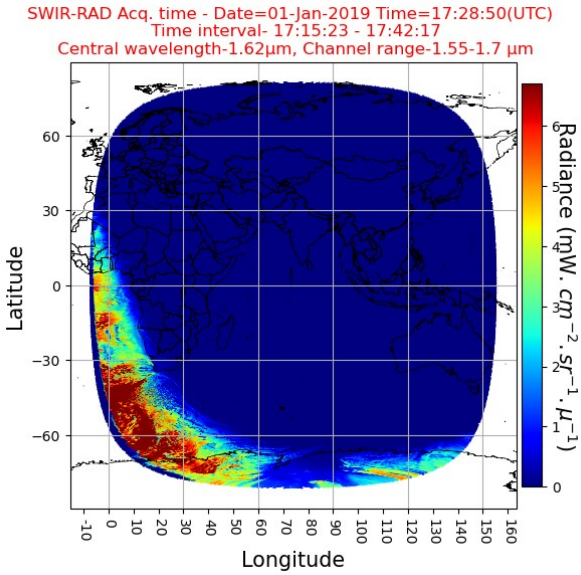

TIR1-TEMP Acq. time - Date=01-Jan-2019 Time=15:28:49(UTC)
Time interval- 15:15:23 - 15:42:16
Central wavelength-10.78µm, Channel range-10.28-11.28 µm
Latitude
Brightness temp. (K)

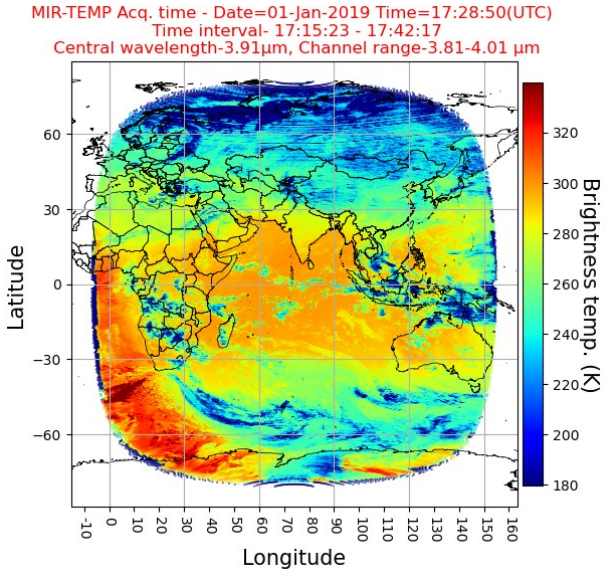

TIR2-TEMP Acq. time - Date=01-Jan-2019 Time=15:28:49(UTC)
Time interval- 15:15:23 - 15:42:16
Central wavelength-11.97µm, Channel range-11.47-12.47 µm
Latitude
Brightness temp. (K)

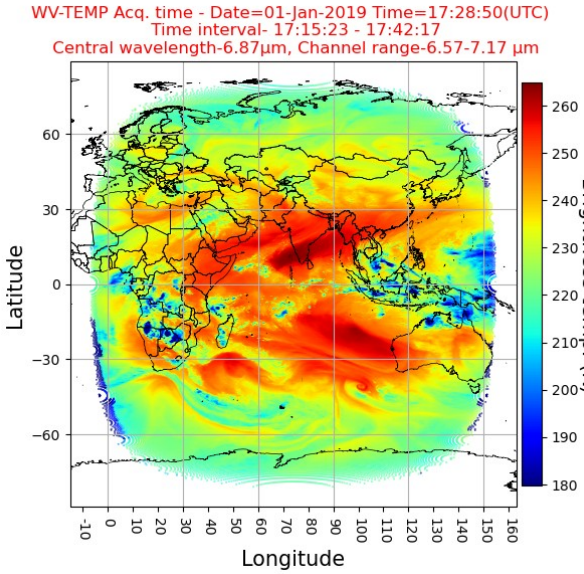
```
#Plot for all the channels in a file
insatpath=r'/data/debasish/insatdata/l1b/2019/2019jan/day01/3RIMG_01JA
N2019_1715_L1B_STD_V01R00.h5'

fig=plt.figure(figsize=(15,22))
ax1=plt.subplot(3,2,1,projection=ccrs.PlateCarree())
ax2=plt.subplot(3,2,2,projection=ccrs.PlateCarree())
ax3=plt.subplot(3,2,3,projection=ccrs.PlateCarree())
ax4=plt.subplot(3,2,4,projection=ccrs.PlateCarree())
ax5=plt.subplot(3,2,5,projection=ccrs.PlateCarree())
ax6=plt.subplot(3,2,6,projection=ccrs.PlateCarree())

insatplotter(insatpath,channel='vis',parameter='albedo',stock=False,gl
obalmap=False,axis=ax1,figsize=6)
print("Done")
insatplotter(insatpath,channel='swir',parameter='rad',stock=False,glob
almap=False,axis=ax2,figsize=6)
print("Done")
insatplotter(insatpath,channel='mir',parameter='temp',stock=False,glob
almap=False,axis=ax3,figsize=6)
print("Done")
insatplotter(insatpath,channel='wv',parameter='temp',stock=False,globa
lmap=False,axis=ax4,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False,axis=ax5,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir2',parameter='temp',stock=False,glo
balmap=False,axis=ax6,figsize=6)

#28 minutes run time.
plt.show()
```

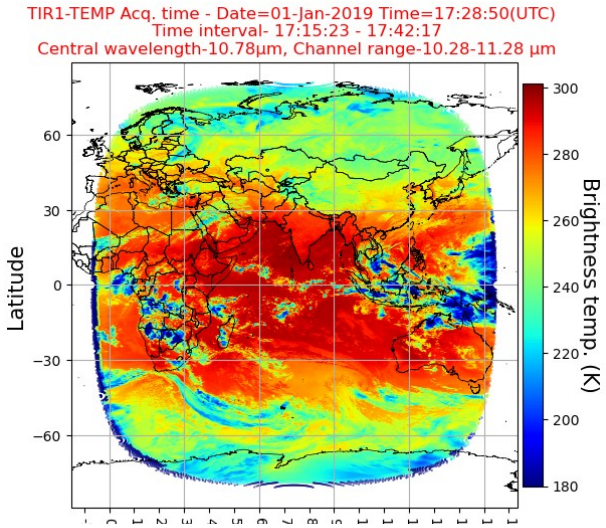
```
Done
Done
Done
Done
Done
```

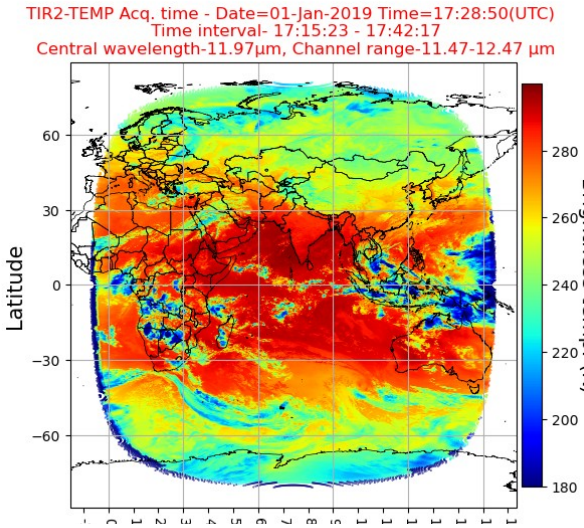

VIS-ALBEDO Acq. time - Date=01-Jan-2019 Time=17:28:50(UTC)
Time interval- 17:15:23 - 17:42:17
Central wavelength-0.65µm, Channel range-0.53-0.78 µm
Latitude
Longitude
Visible albedo (%)


SWIR-RAD Acq. time - Date=01-Jan-2019 Time=17:28:50(UTC)
Time interval- 17:15:23 - 17:42:17
Central wavelength-1.62µm, Channel range-1.55-1.7 µm
Latitude
Longitude
Radiance (mW. cm^{-2}. sr^{-1}. µ^{-1})


MIR-TEMP Acq. time - Date=01-Jan-2019 Time=17:28:50(UTC)
Time interval- 17:15:23 - 17:42:17
Central wavelength-3.91µm, Channel range-3.81-4.01 µm
Latitude
Longitude
Brightness temp. (K)

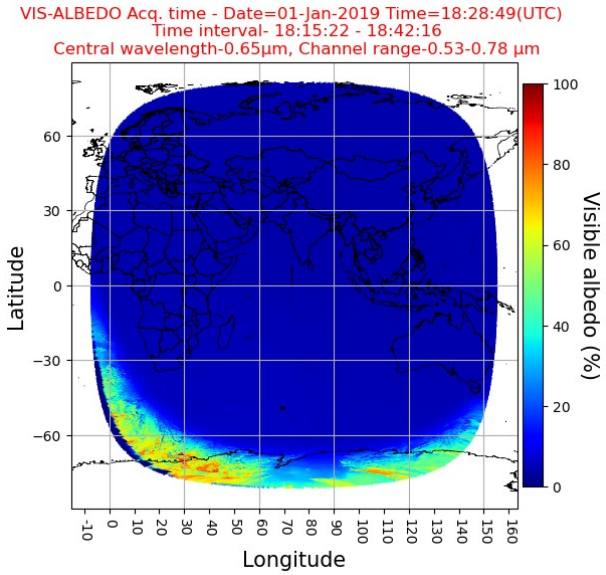

WV-TEMP Acq. time - Date=01-Jan-2019 Time=17:28:50(UTC)
Time interval- 17:15:23 - 17:42:17
Central wavelength-6.87µm, Channel range-6.57-7.17 µm
Latitude
Longitude
Brightness temp. (K)

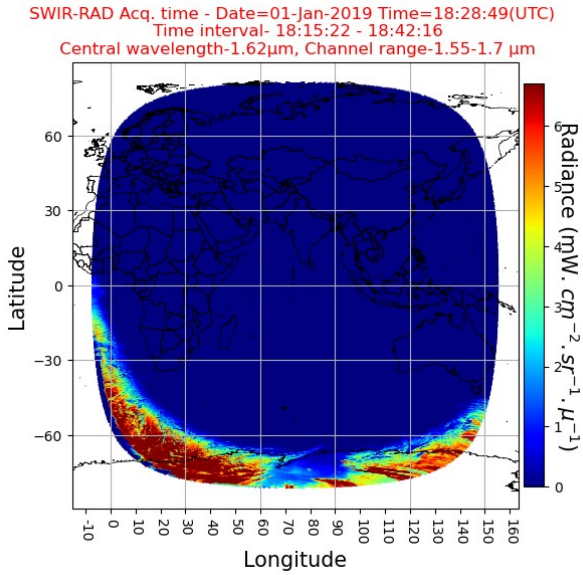

TIR1-TEMP Acq. time - Date=01-Jan-2019 Time=17:28:50(UTC)
Time interval- 17:15:23 - 17:42:17
Central wavelength-10.78µm, Channel range-10.28-11.28 µm
Latitude
Brightness temp. (K)

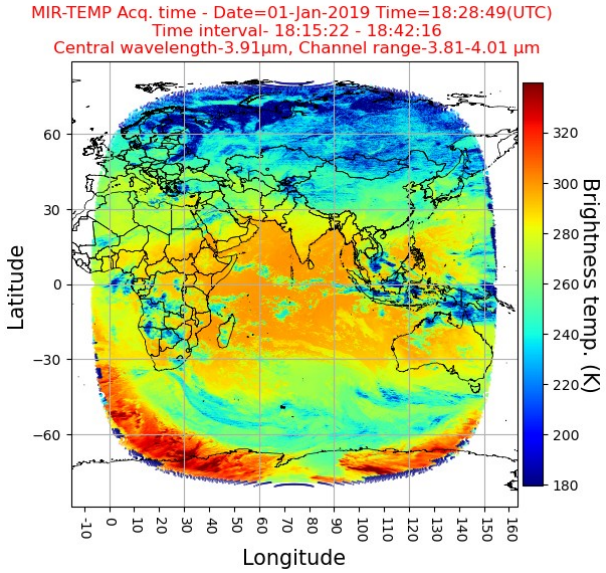

TIR2-TEMP Acq. time - Date=01-Jan-2019 Time=17:28:50(UTC)
Time interval- 17:15:23 - 17:42:17
Central wavelength-11.97µm, Channel range-11.47-12.47 µm
Latitude
Brightness temp. (K)

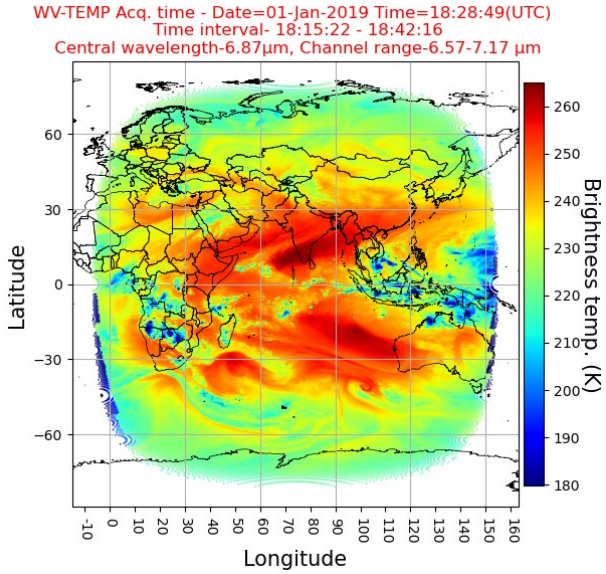
```
#Plot for all the channels in a file
insatpath=r'/data/debasish/insatdata/l1b/2019/2019jan/day01/3RIMG_01JA
N2019_1815_L1B_STD_V01R00.h5'

fig=plt.figure(figsize=(15,22))
ax1=plt.subplot(3,2,1,projection=ccrs.PlateCarree())
ax2=plt.subplot(3,2,2,projection=ccrs.PlateCarree())
ax3=plt.subplot(3,2,3,projection=ccrs.PlateCarree())
ax4=plt.subplot(3,2,4,projection=ccrs.PlateCarree())
ax5=plt.subplot(3,2,5,projection=ccrs.PlateCarree())
ax6=plt.subplot(3,2,6,projection=ccrs.PlateCarree())

insatplotter(insatpath,channel='vis',parameter='albedo',stock=False,gl
obalmap=False,axis=ax1,figsize=6)
print("Done")
insatplotter(insatpath,channel='swir',parameter='rad',stock=False,glob
almap=False,axis=ax2,figsize=6)
print("Done")
insatplotter(insatpath,channel='mir',parameter='temp',stock=False,glob
almap=False,axis=ax3,figsize=6)
print("Done")
insatplotter(insatpath,channel='wv',parameter='temp',stock=False,globa
lmap=False,axis=ax4,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False,axis=ax5,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir2',parameter='temp',stock=False,glo
balmap=False,axis=ax6,figsize=6)

#28 minutes run time.
plt.show()
```

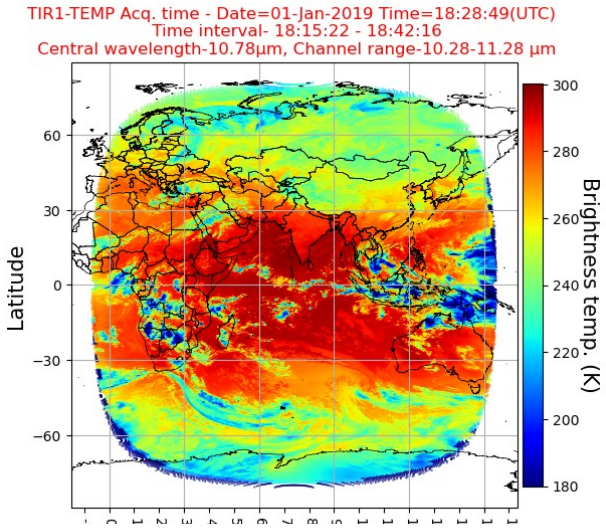
```
Done
Done
Done
Done
Done
```

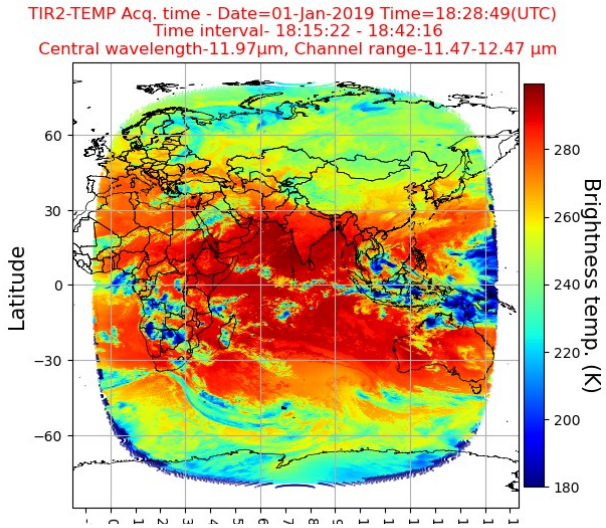

VIS-ALBEDO Acq. time - Date=01-Jan-2019 Time=18:28:49(UTC)
Time interval- 18:15:22 - 18:42:16
Central wavelength-0.65µm, Channel range-0.53-0.78 µm
Latitude
Longitude
Visible albedo (%)


SWIR-RAD Acq. time - Date=01-Jan-2019 Time=18:28:49(UTC)
Time interval- 18:15:22 - 18:42:16
Central wavelength-1.62µm, Channel range-1.55-1.7 µm
Latitude
Longitude
Radiance (mW. cm^{-2}. sr^{-1}. µ^{-1})


MIR-TEMP Acq. time - Date=01-Jan-2019 Time=18:28:49(UTC)
Time interval- 18:15:22 - 18:42:16
Central wavelength-3.91µm, Channel range-3.81-4.01 µm
Latitude
Longitude
Brightness temp. (K)

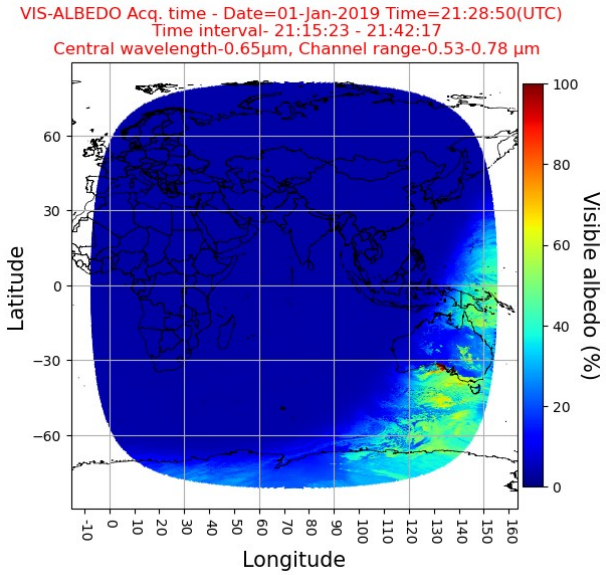

WV-TEMP Acq. time - Date=01-Jan-2019 Time=18:28:49(UTC)
Time interval- 18:15:22 - 18:42:16
Central wavelength-6.87µm, Channel range-6.57-7.17 µm
Latitude
Longitude
Brightness temp. (K)

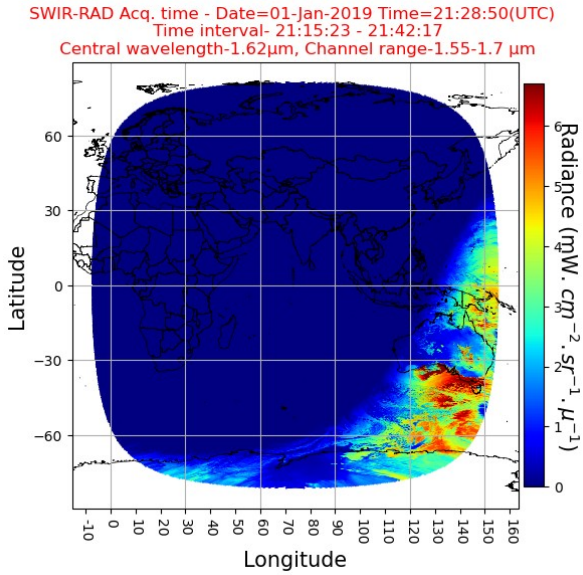

TIR1-TEMP Acq. time - Date=01-Jan-2019 Time=18:28:49(UTC)
Time interval- 18:15:22 - 18:42:16
Central wavelength-10.78µm, Channel range-10.28-11.28 µm
Latitude
Brightness temp. (K)

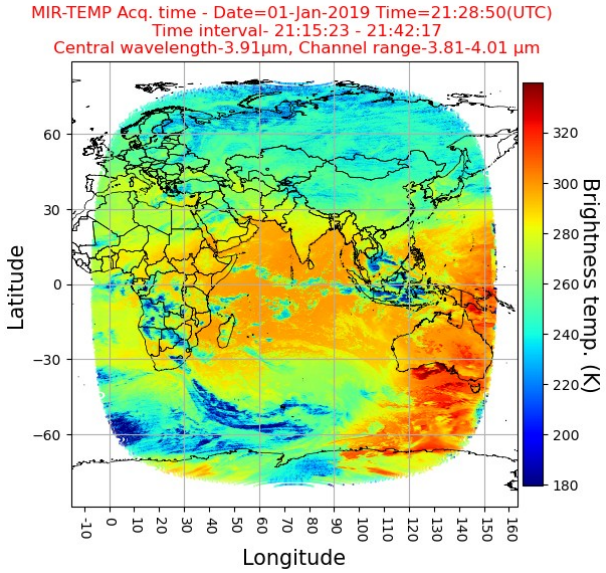

TIR2-TEMP Acq. time - Date=01-Jan-2019 Time=18:28:49(UTC)
Time interval- 18:15:22 - 18:42:16
Central wavelength-11.97µm, Channel range-11.47-12.47 µm
Latitude
Brightness temp. (K)

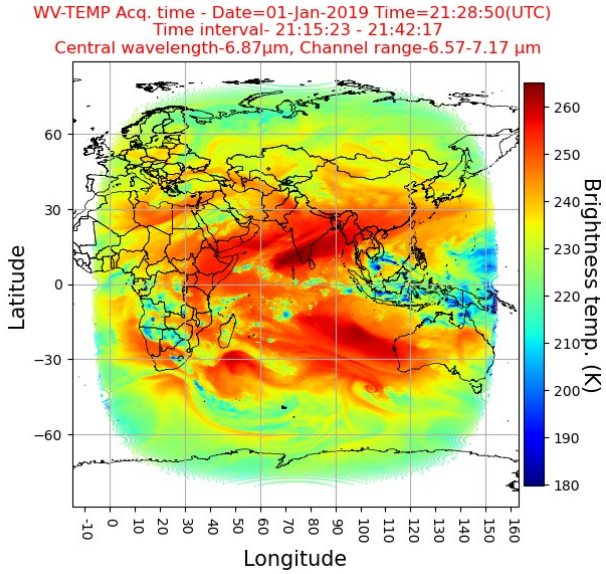
```
#Plot for all the channels in a file
insatpath=r'/data/debasish/insatdata/l1b/2019/2019jan/day01/3RIMG_01JA
N2019_2115_L1B_STD_V01R00.h5'

fig=plt.figure(figsize=(15,22))
ax1=plt.subplot(3,2,1,projection=ccrs.PlateCarree())
ax2=plt.subplot(3,2,2,projection=ccrs.PlateCarree())
ax3=plt.subplot(3,2,3,projection=ccrs.PlateCarree())
ax4=plt.subplot(3,2,4,projection=ccrs.PlateCarree())
ax5=plt.subplot(3,2,5,projection=ccrs.PlateCarree())
ax6=plt.subplot(3,2,6,projection=ccrs.PlateCarree())

insatplotter(insatpath,channel='vis',parameter='albedo',stock=False,gl
obalmap=False,axis=ax1,figsize=6)
print("Done")
insatplotter(insatpath,channel='swir',parameter='rad',stock=False,glob
almap=False,axis=ax2,figsize=6)
print("Done")
insatplotter(insatpath,channel='mir',parameter='temp',stock=False,glob
almap=False,axis=ax3,figsize=6)
print("Done")
insatplotter(insatpath,channel='wv',parameter='temp',stock=False,globa
lmap=False,axis=ax4,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False,axis=ax5,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir2',parameter='temp',stock=False,glo
balmap=False,axis=ax6,figsize=6)

#28 minutes run time.
plt.show()
```

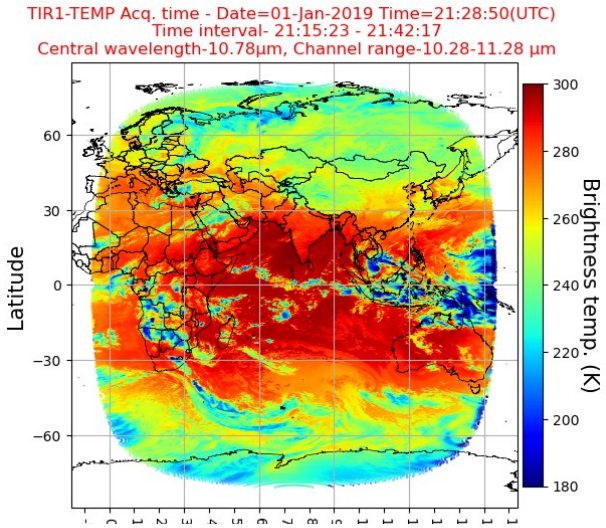
```
Done
Done
Done
Done
Done
```

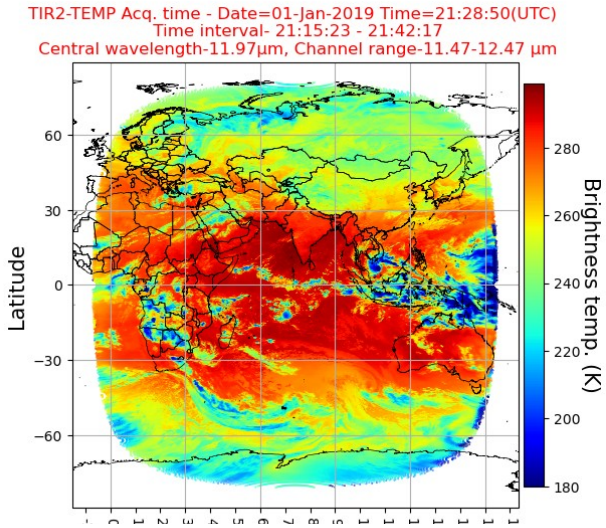

VIS-ALBEDO Acq. time - Date=01-Jan-2019 Time=21:28:50(UTC)
Time interval- 21:15:23 - 21:42:17
Central wavelength-0.65µm, Channel range-0.53-0.78 µm
Latitude
Longitude
Visible albedo (%)


SWIR-RAD Acq. time - Date=01-Jan-2019 Time=21:28:50(UTC)
Time interval- 21:15:23 - 21:42:17
Central wavelength-1.62µm, Channel range-1.55-1.7 µm
Latitude
Longitude
Radiance (mW. cm^{-2}. sr^{-1}. µ^{-1})


MIR-TEMP Acq. time - Date=01-Jan-2019 Time=21:28:50(UTC)
Time interval- 21:15:23 - 21:42:17
Central wavelength-3.91µm, Channel range-3.81-4.01 µm
Latitude
Longitude
Brightness temp. (K)

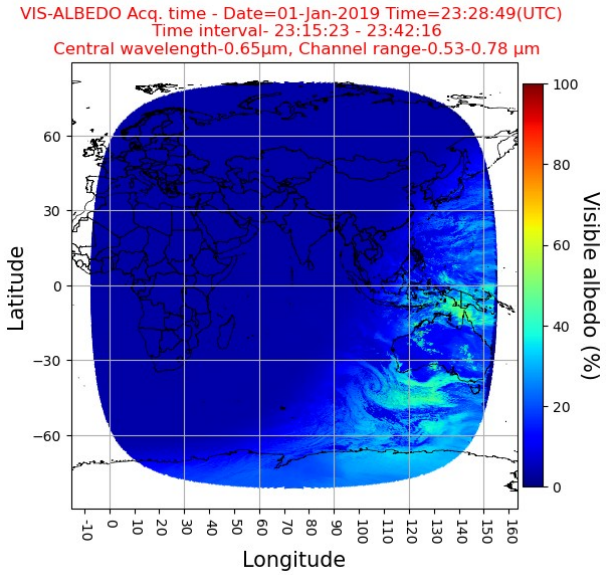

WV-TEMP Acq. time - Date=01-Jan-2019 Time=21:28:50(UTC)
Time interval- 21:15:23 - 21:42:17
Central wavelength-6.87µm, Channel range-6.57-7.17 µm
Latitude
Longitude
Brightness temp. (K)

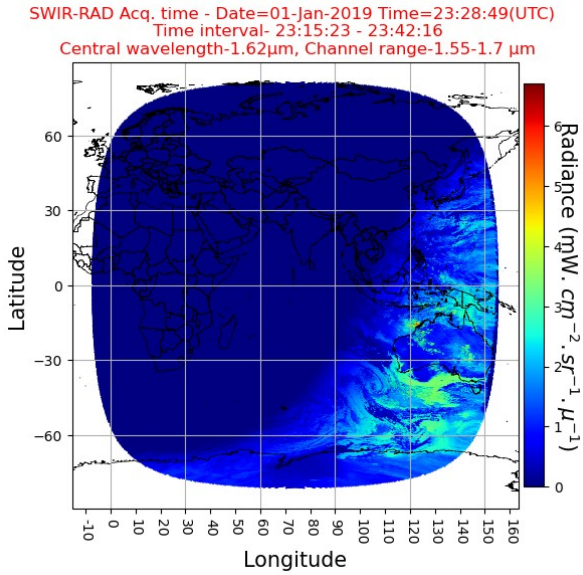

TIR1-TEMP Acq. time - Date=01-Jan-2019 Time=21:28:50(UTC)
Time interval- 21:15:23 - 21:42:17
Central wavelength-10.78µm, Channel range-10.28-11.28 µm
Latitude
Brightness temp. (K)

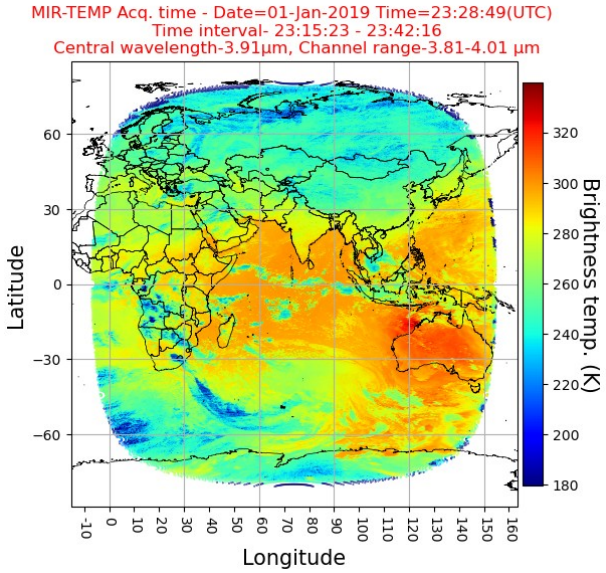

TIR2-TEMP Acq. time - Date=01-Jan-2019 Time=21:28:50(UTC)
Time interval- 21:15:23 - 21:42:17
Central wavelength-11.97µm, Channel range-11.47-12.47 µm
Latitude
Brightness temp. (K)

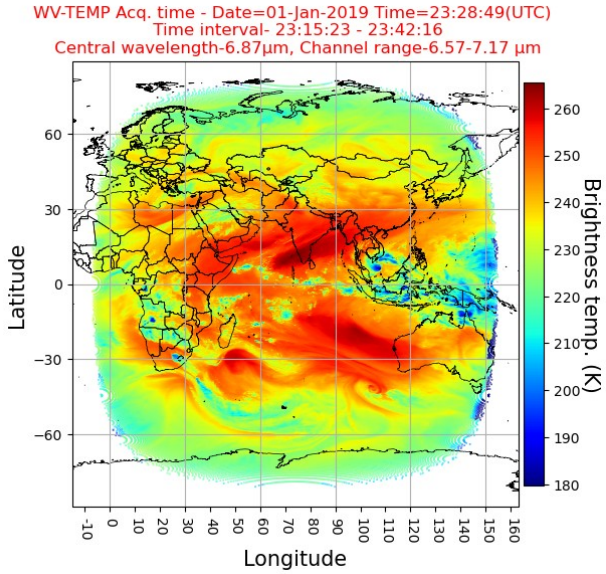
```
#Plot for all the channels in a file
insatpath=r'/data/debasish/insatdata/l1b/2019/2019jan/day01/3RIMG_01JA
N2019_2315_L1B_STD_V01R00.h5'

fig=plt.figure(figsize=(15,22))
ax1=plt.subplot(3,2,1,projection=ccrs.PlateCarree())
ax2=plt.subplot(3,2,2,projection=ccrs.PlateCarree())
ax3=plt.subplot(3,2,3,projection=ccrs.PlateCarree())
ax4=plt.subplot(3,2,4,projection=ccrs.PlateCarree())
ax5=plt.subplot(3,2,5,projection=ccrs.PlateCarree())
ax6=plt.subplot(3,2,6,projection=ccrs.PlateCarree())

insatplotter(insatpath,channel='vis',parameter='albedo',stock=False,gl
obalmap=False,axis=ax1,figsize=6)
print("Done")
insatplotter(insatpath,channel='swir',parameter='rad',stock=False,glob
almap=False,axis=ax2,figsize=6)
print("Done")
insatplotter(insatpath,channel='mir',parameter='temp',stock=False,glob
almap=False,axis=ax3,figsize=6)
print("Done")
insatplotter(insatpath,channel='wv',parameter='temp',stock=False,globa
lmap=False,axis=ax4,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir1',parameter='temp',stock=False,glo
balmap=False,axis=ax5,figsize=6)
print("Done")
insatplotter(insatpath,channel='tir2',parameter='temp',stock=False,glo
balmap=False,axis=ax6,figsize=6)
print("Done")
plt.show()
```

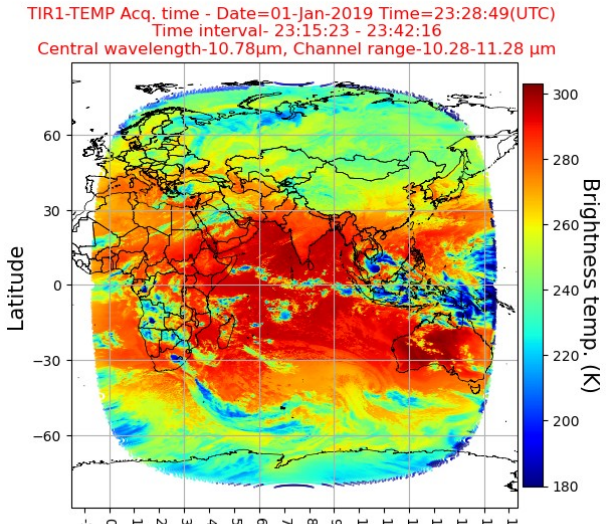
```
Done
Done
Done
Done
Done
Done
```

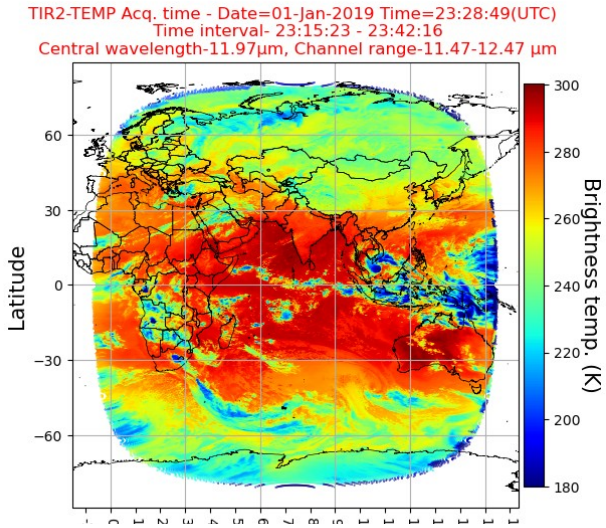

VIS-ALBEDO Acq. time - Date=01-Jan-2019 Time=23:28:49(UTC)
Time interval- 23:15:23 - 23:42:16
Central wavelength-0.65µm, Channel range-0.53-0.78 µm
Latitude
Longitude
Visible albedo (%)


SWIR-RAD Acq. time - Date=01-Jan-2019 Time=23:28:49(UTC)
Time interval- 23:15:23 - 23:42:16
Central wavelength-1.62µm, Channel range-1.55-1.7 µm
Latitude
Longitude
Radiance (mW. cm^{-2}. sr^{-1}. µ^{-1})

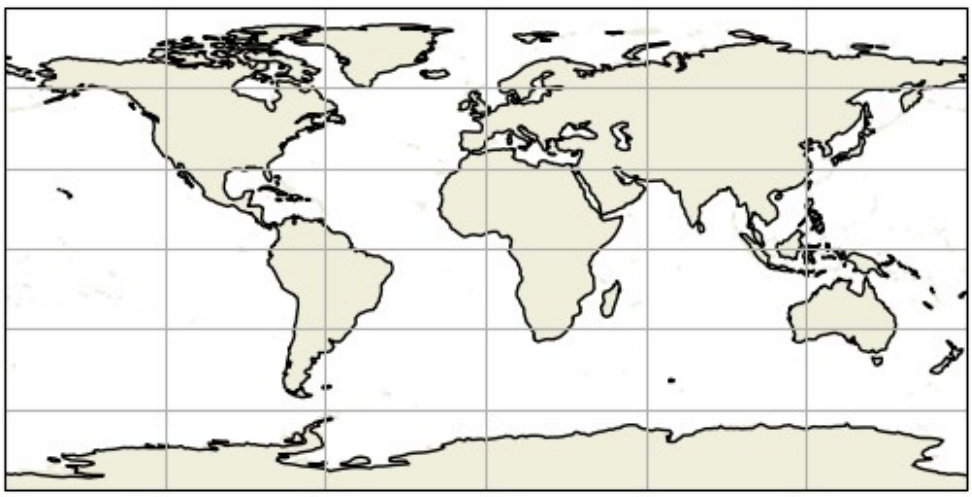

MIR-TEMP Acq. time - Date=01-Jan-2019 Time=23:28:49(UTC)
Time interval- 23:15:23 - 23:42:16
Central wavelength-3.91µm, Channel range-3.81-4.01 µm
Latitude
Longitude
Brightness temp. (K)


WV-TEMP Acq. time - Date=01-Jan-2019 Time=23:28:49(UTC)
Time interval- 23:15:23 - 23:42:16
Central wavelength-6.87µm, Channel range-6.57-7.17 µm
Latitude
Longitude
Brightness temp. (K)


TIR1-TEMP Acq. time - Date=01-Jan-2019 Time=23:28:49(UTC)
Time interval- 23:15:23 - 23:42:16
Central wavelength-10.78µm, Channel range-10.28-11.28 µm
Latitude
Brightness temp. (K)


TIR2-TEMP Acq. time - Date=01-Jan-2019 Time=23:28:49(UTC)
Time interval- 23:15:23 - 23:42:16
Central wavelength-11.97µm, Channel range-11.47-12.47 µm
Latitude
Brightness temp. (K)

```
#Find the length and breadth of a figure

fig=plt.figure(figsize=(15,21))


#Make a plot of earth with simple coastlines

ax=plt.axes(projection=ccrs.PlateCarree())
#add country borders

ax.add_feature(cfeature.NaturalEarthFeature('physical', 'land', '50m',
edgecolor='face', facecolor=cfeature.COLORS['land']))
ax.coastlines()
ax.gridlines()

plt.show()
```

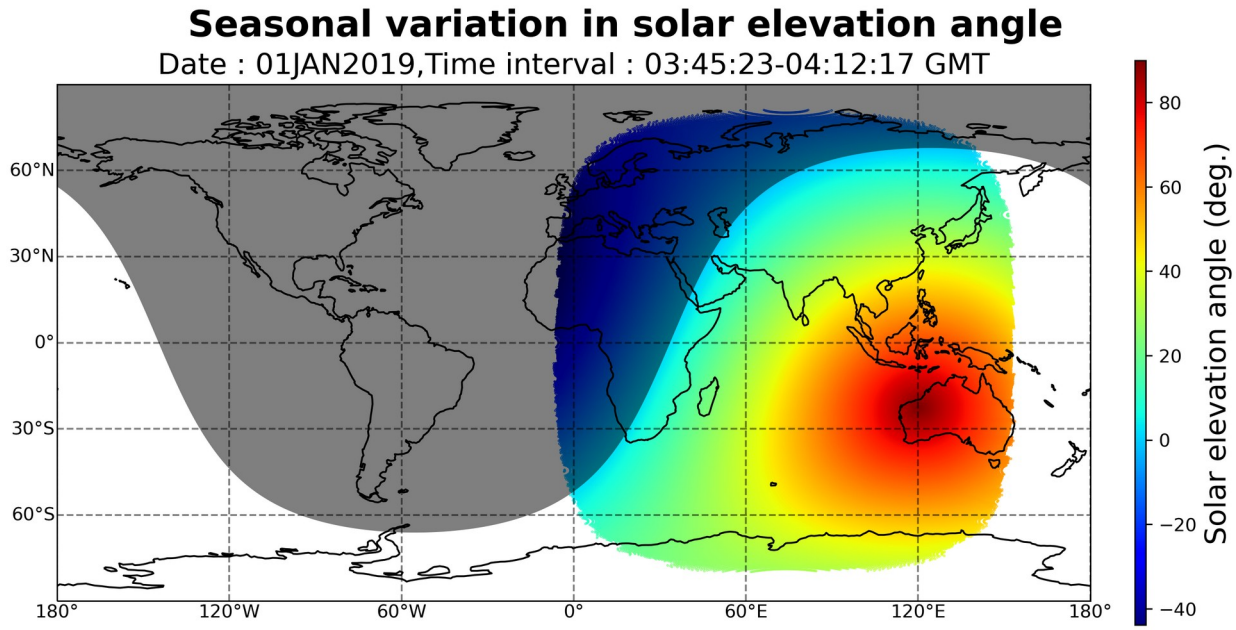


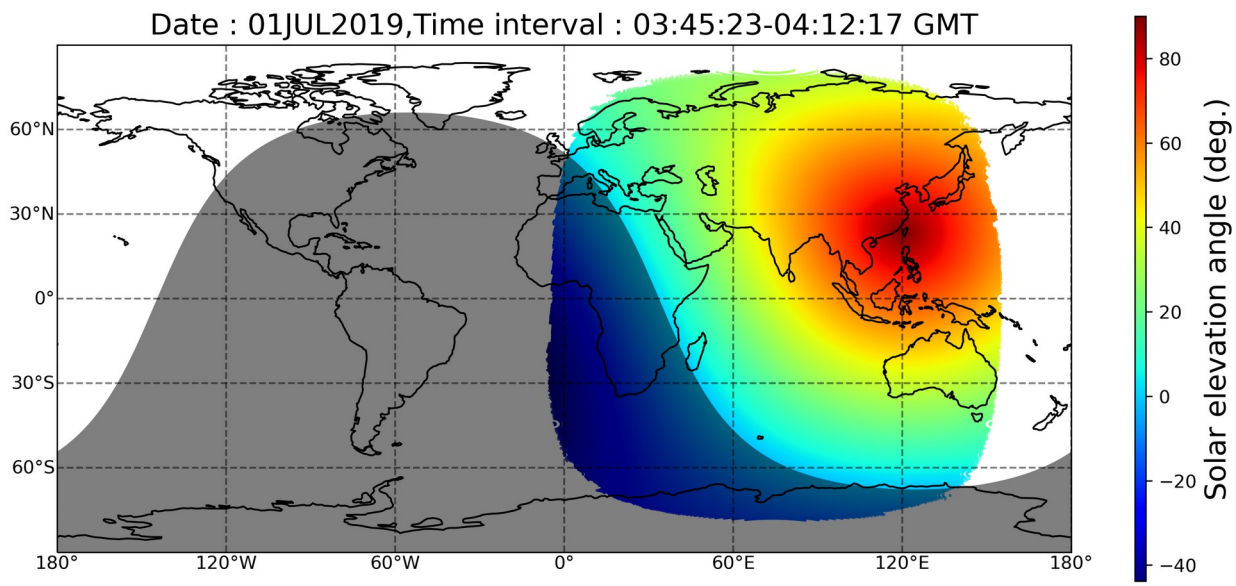
```
insatpath1 =
r'/data/debasish/insatdata/l1b/2019/2019jan/day01/3RIMG_01JAN2019_0345
_L1B_STD_V01R00.h5'
insatpath2 =
r'/data/debasish/insatdata/l1b/2019/jul2019_day1_std/3RIMG_01JUL2019_0
345_L1B_STD_V01R00.h5'

insatfile1 = h5py.File(insatpath1, 'r')
insatfile2 = h5py.File(insatpath2, 'r')

lat1 = np.array(insatfile1['Latitude'],dtype=float)
lat_fill1 = insatfile1['Latitude'].attrs['_FillValue'][0]
lat1[lat1==lat_fill1] = np.nan
lat1 = lat1/100

lon1 = np.array(insatfile1['Longitude'],dtype=float)
lon_fill1 = insatfile1['Longitude'].attrs['_FillValue'][0]
```

```
lon1[lon1==lon_fill1] = np.nan
lon1 = lon1/100

lat2 = np.array(insatfile2['Latitude'],dtype=float)
lat_fill2 = insatfile2['Latitude'].attrs['_FillValue'][0]
lat2[lat2==lat_fill2] = np.nan
lat2 = lat2/100

lon2 = np.array(insatfile2['Longitude'],dtype=float)
lon_fill2 = insatfile2['Longitude'].attrs['_FillValue'][0]
lon2[lon2==lon_fill2] = np.nan
lon2 = lon2/100


solarelevation1 = np.array(insatfile1['Sun_Elevation'],dtype=float)
solarelevation_fill1 = insatfile1['Sun_Elevation'].attrs['_FillValue']
[0]
solarelevation1[solarelevation1==solarelevation_fill1] = np.nan
solarelevation1 = solarelevation1/100

solarelevation2 = np.array(insatfile2['Sun_Elevation'],dtype=float)
solarelevation_fill2 = insatfile2['Sun_Elevation'].attrs['_FillValue']
[0]
solarelevation2[solarelevation2==solarelevation_fill2] = np.nan
solarelevation2 = solarelevation2/100

date1 = insatfile1.attrs['Acquisition_Date'].decode('utf-8')
acq_start1 = insatfile1.attrs['Acquisition_Start_Time'].decode('utf-
8')[-8:]
acq_end1 = insatfile1.attrs['Acquisition_End_Time'].decode('utf-8')[-
8:]

date2 = insatfile2.attrs['Acquisition_Date'].decode('utf-8')
acq_start2 = insatfile2.attrs['Acquisition_Start_Time'].decode('utf-
8')[-8:]
acq_end2 = insatfile2.attrs['Acquisition_End_Time'].decode('utf-8')[-
8:]

year1 = int(date1[-4:])
month_abb1 = date1[2:5]
month_no1 = datetime.datetime.strptime(month_abb1, "%b").month
day1 = int(date1[:2])
hour1 = int(acq_start1.split(":")[0])
minute1 = int(acq_start1.split(":")[1])
second1  = int(acq_start1.split(":")[2])

year2 = int(date2[-4:])
month_abb2 = date2[2:5]
month_no2 = datetime.datetime.strptime(month_abb2, "%b").month
day2 = int(date2[:2])
```

```
hour2  = int(acq_start2.split(":")[0])
minute2 = int(acq_start2.split(":")[1])
second2 = int(acq_start2.split(":")[2])
#print(date1, acq_start1, acq_end1, date2, acq_start2, acq_end2)

fig = plt.figure(figsize=(10,11),dpi=300)

ax1 = plt.subplot(2,1,1, projection = ccrs.PlateCarree())
ax2 = plt.subplot(2,1,2, projection = ccrs.PlateCarree())

extent = -1

plot1 = ax1.scatter(lon1.flatten()[0:extent], lat1.flatten()
[0:extent],
        c = solarelevation1.flatten()[0:extent], cmap = 'jet', s =
0.1,
        transform = ccrs.PlateCarree())
ax1.set_global()
gl = ax1.gridlines(crs=ccrs.PlateCarree(), draw_labels=True,
                    linewidth=1, color='black', alpha=0.5,
linestyle='--')
gl.xlabels_top = False
gl.ylabels_right = False
date1_datetime  = datetime.datetime(year1, month_no1, day1, hour1,
minute1, second1)
ax1.add_feature(Nightshade(date1_datetime, alpha=0.5))
ax1.coastlines()
cbar1 = plt.colorbar(plot1,
                     orientation = 'vertical',
                     pad=0.035,
                     #fraction=0.019,
                     aspect=50,
                     shrink = 0.815,
                     ax=ax1)
cbar1.set_label('Solar elevation angle (deg.)', fontsize=16)
ax1.set_title('Date : {},Time interval : {}-{} GMT'.format(date1,
acq_start1, acq_end1), fontsize=16)


plot2 = ax2.scatter(lon2.flatten()[0:extent], lat2.flatten()
[0:extent],
        c = solarelevation2.flatten()[0:extent], cmap = 'jet', s =
0.1,
        transform = ccrs.PlateCarree())
ax2.set_global()
gl = ax2.gridlines(crs=ccrs.PlateCarree(), draw_labels=True,
                    linewidth=1, color='black', alpha=0.5,
linestyle='--')
gl.xlabels_top = False
gl.ylabels_right = False
```

```
date2_datetime  = datetime.datetime(year2, month_no2, day2, hour2,
minute2, second2)
ax2.add_feature(Nightshade(date2_datetime, alpha=0.5))
ax2.coastlines()
cbar2 = plt.colorbar(plot2, orientation = 'vertical',
                     #pad=0.035,
                     #fraction=0.019,
                     aspect=50,
                     shrink = 0.815,
                     ax=ax2)
cbar2.set_label('Solar elevation angle (deg.)', fontsize=16)
ax2.set_title('Date : {},Time interval : {}-{} GMT'.format(date2,
acq_start2, acq_end2), fontsize=16)

fig.suptitle('Seasonal variation in solar elevation angle',
fontsize=20, fontweight = 'bold',
             y=0.94, x=0.45)
plt.tight_layout()
plt.show()
```






```
date1 = insatfile1.attrs['Acquisition_Date'].decode('utf-8')
acq_start1 = insatfile1.attrs['Acquisition_Start_Time'].decode('utf-
8').split("T")[1]
acq_end1 = insatfile1.attrs['Acquisition_End_Time'].decode('utf-
8').split("T")[1]

date2 = insatfile2.attrs['Acquisition_Date'].decode('utf-8')
acq_start2 = insatfile2.attrs['Acquisition_Start_Time'].decode('utf-
8').split("T")[1]
acq_end2 = insatfile2.attrs['Acquisition_End_Time'].decode('utf-
8').split("T")[1]
```

```
print(date1, acq_start1, acq_end1, date2, acq_start2, acq_end2)

year1 = int(date1[-4:])
month_abb1 = date1[2:5]
month_no1 = datetime.datetime.strptime(month_abb1, "%b").month
date1 = int(date1[:2])
hour1 = int(acq_start1.split(":")[0])
minute1 = int(acq_start1.split(":")[1])
second1  = int(acq_start1.split(":")[2])
print(year1, month_no1, date1, hour1, minute1, second1)
```

```
01JAN2019 03:45:23 04:12:17 01JUL2019 03:45:23 04:12:17
2019 1 1 3 45 23
```

Finding the correlation between insat channels and solar elevation angle

```
insatpath=r'/data/debasish/insatdata/l1b/2019/2019jan/
day01/3RIMG_01JAN2019_0545_L1B_STD_V01R00.h5'

insatfile = h5py.File(insatpath,'r')
print(insatfile.keys())
```

```
<KeysViewHDF5 ['GeoX', 'GeoX1', 'GeoX2', 'GeoY', 'GeoY1', 'GeoY2',
'GreyCount', 'IMG_MIR', 'IMG_MIR_RADIANCE', 'IMG_MIR_TEMP',
'IMG_SWIR', 'IMG_SWIR_RADIANCE', 'IMG_TIR1', 'IMG_TIR1_RADIANCE',
'IMG_TIR1_TEMP', 'IMG_TIR2', 'IMG_TIR2_RADIANCE', 'IMG_TIR2_TEMP',
'IMG_VIS', 'IMG_VIS_ALBEDO', 'IMG_VIS_RADIANCE', 'IMG_WV',
'IMG_WV_RADIANCE', 'IMG_WV_TEMP', 'Latitude', 'Latitude_VIS',
'Latitude_WV', 'Longitude', 'Longitude_VIS', 'Longitude_WV',
'SCAN_LINE_TIME', 'Sat_Azimuth', 'Sat_Elevation', 'Sun_Azimuth',
'Sun_Elevation', 'time']>
```

```
vislat=np.ma.masked_equal(np.array(insatfile['Latitude_VIS']),32767)/
1000 #Note how the scale is different for different channels
vislon=np.ma.masked_equal(np.array(insatfile['Longitude_VIS']),32767)/
1000
wvlat=np.ma.masked_equal(np.array(insatfile['Latitude_WV']),32767)/100
wvlon=np.ma.masked_equal(np.array(insatfile['Longitude_WV']),32767)/
100
tirlat=np.ma.masked_equal(np.array(insatfile['Latitude']),32767)/100
tirlon=np.ma.masked_equal(np.array(insatfile['Longitude']),32767)/100

viscount=np.ma.masked_equal(np.array(insatfile['IMG_VIS'][0]),0)
swircount=np.ma.masked_equal(np.array(insatfile['IMG_SWIR'][0]),0)
mircount=np.ma.masked_equal(np.array(insatfile['IMG_MIR'][0]),1023)
wvcount=np.ma.masked_equal(np.array(insatfile['IMG_WV'][0]),1023)
```

```
tir1count=np.ma.masked_equal(np.array(insatfile['IMG_TIR1'][0]),1023)
tir2count=np.ma.masked_equal(np.array(insatfile['IMG_TIR2'][0]),1023)

visalbedolut=np.array(insatfile['IMG_VIS_ALBEDO'])
swirradlut=np.array(insatfile['IMG_SWIR_RADIANCE'])
mirtemplut=np.array(insatfile['IMG_MIR_TEMP'])
wvtemplut=np.array(insatfile['IMG_WV_TEMP'])
tir1templut=np.array(insatfile['IMG_TIR1_TEMP'])
tir2templut=np.array(insatfile['IMG_TIR2_TEMP'])

def counttodata(count,lut): #This function converts the count to data,
data can be radiance, albedo, temperature.
    return lut[count]


visalbedo = counttodata(viscount,visalbedolut)
swirrad = counttodata(swircount,swirradlut)
mirbt = counttodata(mircount,mirtemplut)
wvbt = counttodata(wvcount, wvtemplut)
tir1bt = counttodata(tir1count, tir1templut)
tir2bt = counttodata(tir2count, tir2templut)

sunelevation = np.ma.masked_equal(np.array(insatfile['Sun_Elevation']
[0]),32767)/100
visalbedodownsized = visalbedo[::4,::4]
swirraddownsized = swirrad[::4,::4]
print(visalbedo.shape,swirrad.shape,mirbt.shape,wvbt.shape,tir1bt.shap
e,tir2bt.shape, sunelevation.shape)
print(visalbedodownsized.shape, swirraddownsized.shape)
```

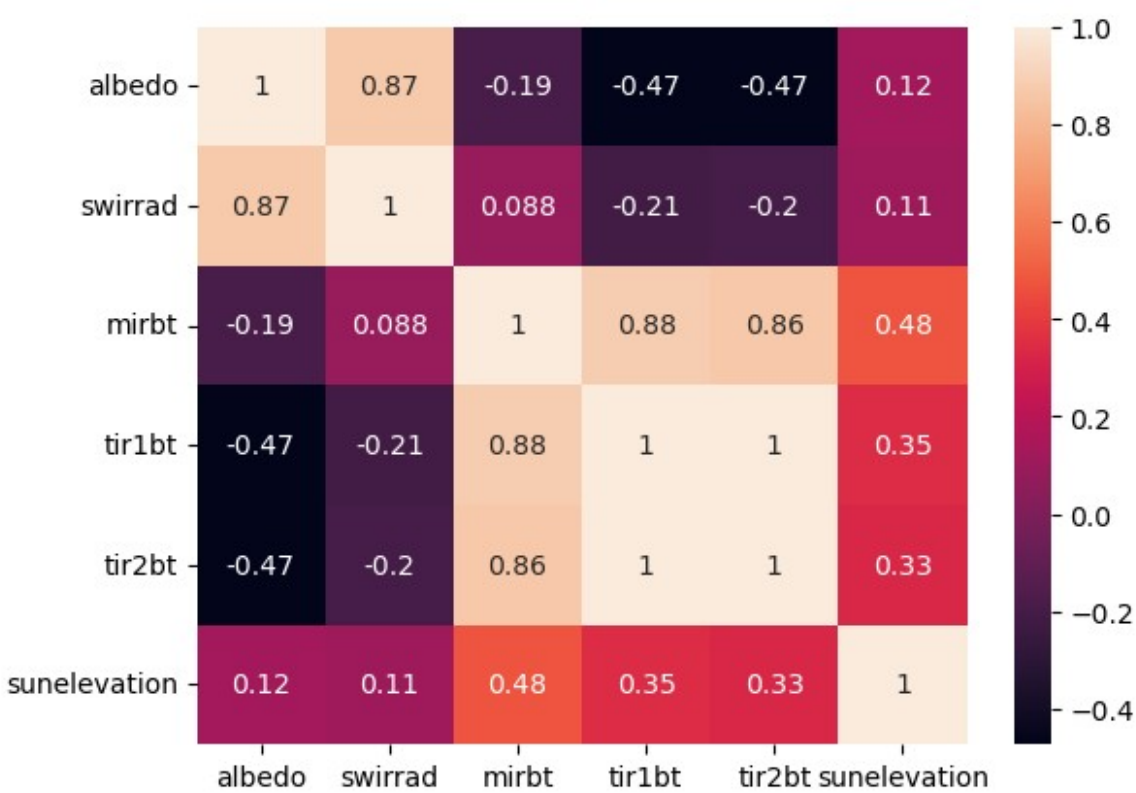
```
(11264, 11220) (11264, 11220) (2816, 2805) (1408, 1402) (2816, 2805)
(2816, 2805) (2816, 2805)
(2816, 2805) (2816, 2805)
```

```
albedoarray = visalbedodownsized.flatten()
swirradarray = swirraddownsized.flatten()
mirarray = mirbt.flatten()
tir1btarray = tir1bt.flatten()
tir2btarray = tir2bt.flatten()
sunelevationarray = sunelevation.flatten()
print(len(albedoarray),len(swirradarray),len(mirarray),len(tir1btarray
),len(tir2btarray), len(sunelevationarray))

df = pd.DataFrame({
                    'albedo':albedoarray,
                    'swirrad':swirradarray,
                    'mirbt':mirarray,
                    'tir1bt':tir1btarray,
                    'tir2bt':tir2btarray,
                    'sunelevation':sunelevationarray
```

```
                    })
print(len(df))
```

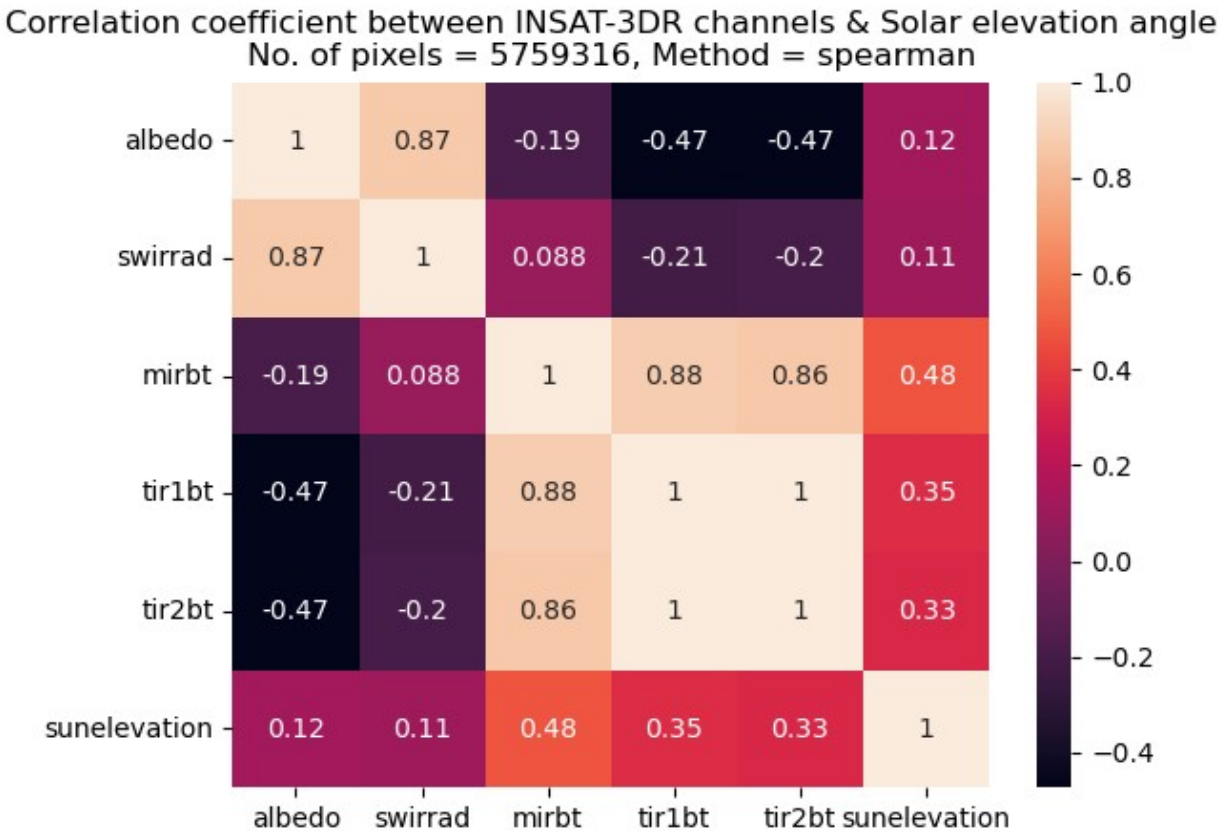
```
7898880 7898880 7898880 7898880 7898880 7898880
7898880
```

```
import seaborn as sns
df.dropna(inplace=True)
plot = sns.heatmap(df.corr(method = 'spearman'),annot=True)
```




```
method = 'spearman'
plot = sns.heatmap(df.corr(method = method),annot=True)

plot.set_title("Correlation coefficient between INSAT-3DR channels &
Solar elevation angle\nNo. of pixels = {}, Method =
{}".format(len(df),method))
```

```
Text(0.5, 1.0, 'Correlation coefficient between INSAT-3DR channels &
Solar elevation angle\nNo. of pixels = 5759316, Method = spearman')
```




```python
def generate_christmas_tree(height):
    for i in range(height):
        print(' ' * (height - i - 1) + '*' * (2 * i + 1))
    print(' ' * (height - 1) + '|')

# Example usage
tree_height = 20
generate_christmas_tree(tree_height)
```

```
                   *
                  ***
                 *****
                *******
               *********
              ***********
             *************
            ***************
           *****************
          *******************
         *********************
        ***********************
       *************************
      ***************************
     *****************************
                   |
```

```
    ******************************
   *******************************
  ********************************
 *********************************
*************************************
                    |
```

```
"Lord Debasish".replace("Lord","Moron"), "Moron
Priyo".replace("Moron","Godly")
```

```
('Moron Debasish', 'Godly Priyo')
```