GPM/TRMM Data reading program guide (IDL version)

2022/02/21

8th ed.

This document describes how to create a program (IDL) to read data from the Global Precipitation Measurement (GPM) satellite.

The sample programs described in this document have been tested with product version 07 for GPM/TRMM and with product version 5 for GSMaP.

Table of Contents

Introduction

This document explains how to read in GPM/TRMM data using IDL.
The GPM and TRMM formats have been unified since version 06 products (equivalent to TRMM version 8), and the latest algorithm is version 07 (equivalent to TRMM version 9). The latest algorithm is version 07 (equivalent to TRMM version 9), which can be read in the same way in this sample program.
In addition to IDL, there are other methods to read GPM data as shown in Table 1.1. To determine which method to use, please refer to the "Read Method Judgment Flow" on the next page.
Table 1.2 lists the operating systems on which the sample programs used in this document were tested.

Table 1.1 Data loading methods

	Data loading method	Name of material	remarks
1	Using THOR	GPM/TRMM Data Loading Program Guide (THOR Edition)	
2	Use IDL	GPM/TRMM Data Loading Program Guide (IDL version)	
3	Use C	GPM/TRMM data reading program guide (C language version)	
4	Using FORTRAN	GPM/TRMM Data Loading Program Guide (FORTRAN Edition)	
5	Using Python	GPM/TRMM data reading program guide (Python version)	

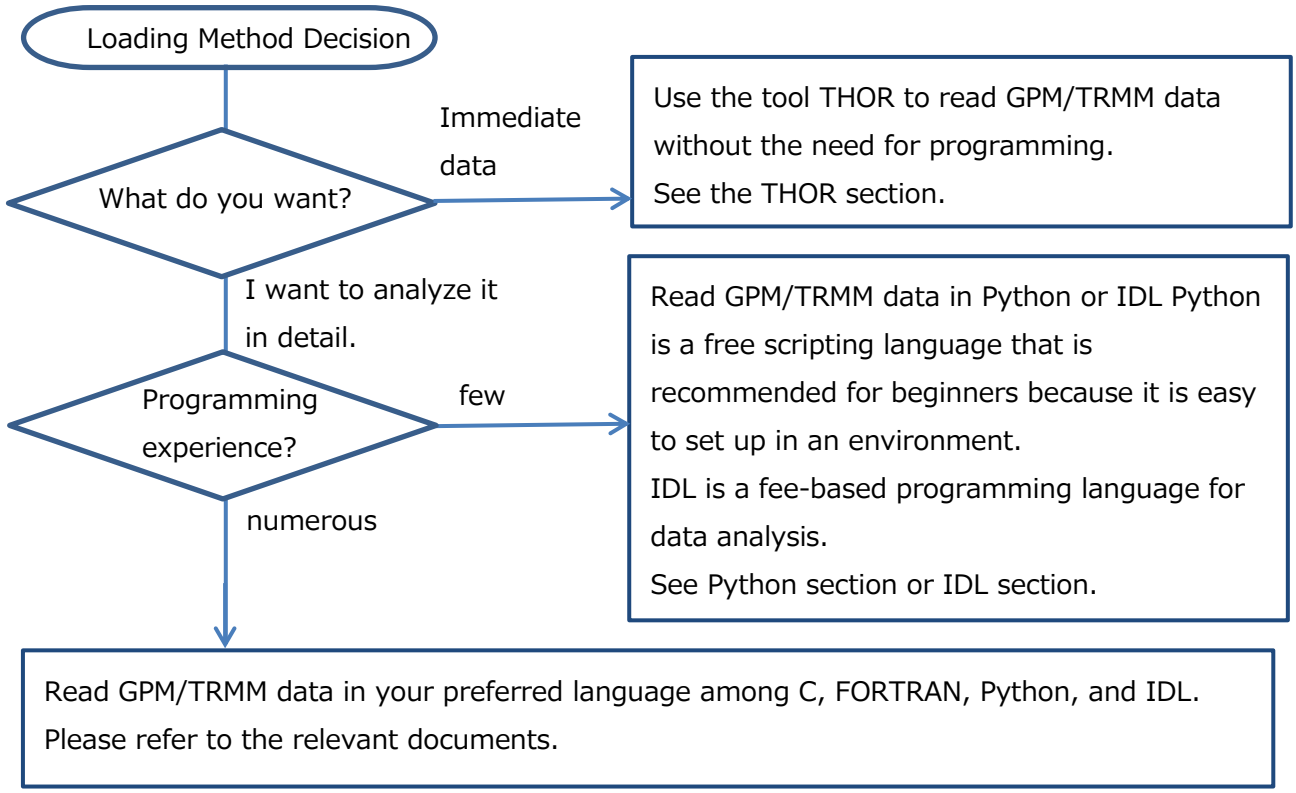


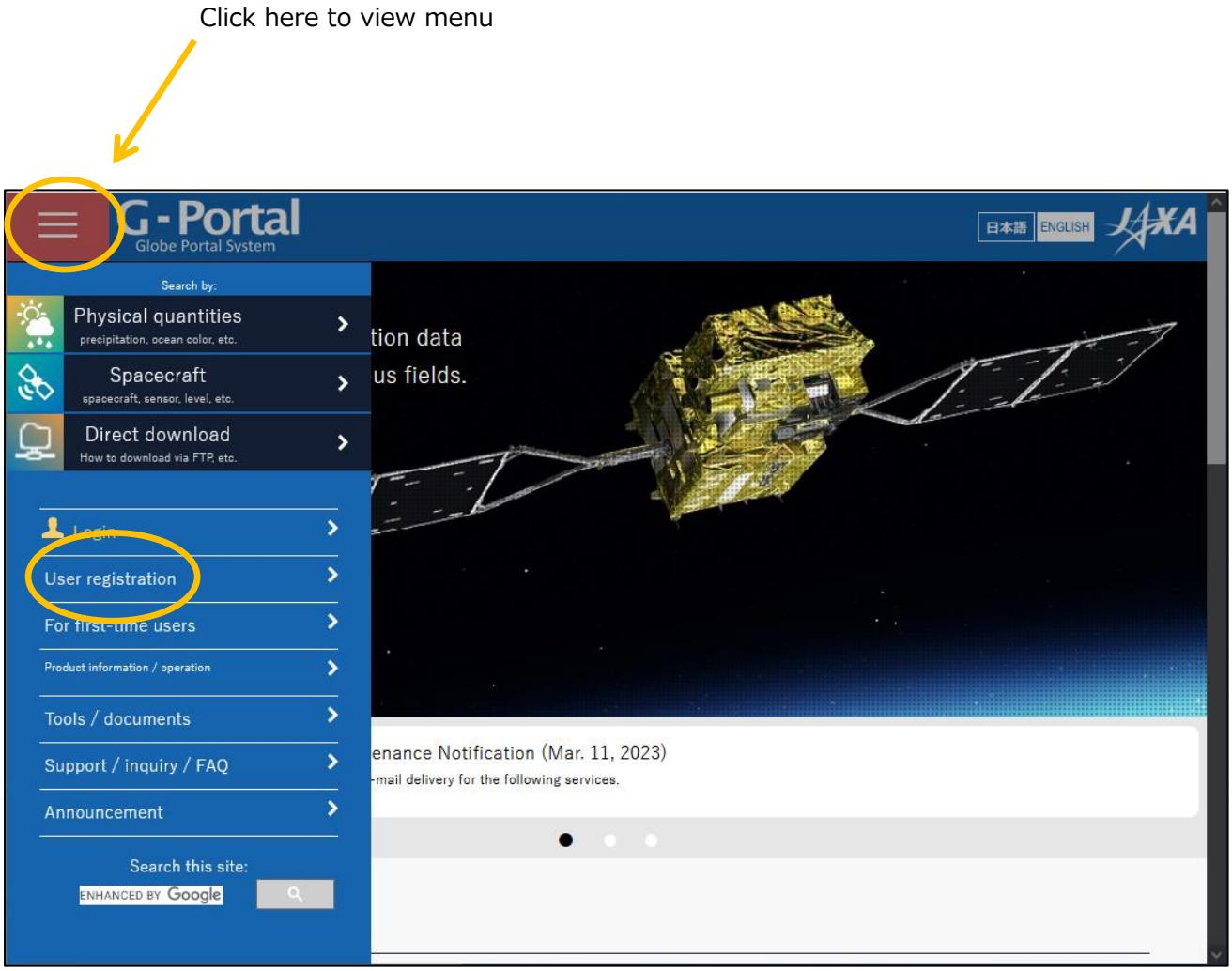
Table 1.2 Sample Program Operation Check Table

	sample program	Linux	Windows	remarks
1	c	○	-	
2	FORTRAN	○	-	
3	Python	○	○	
4	IDL	○	○	

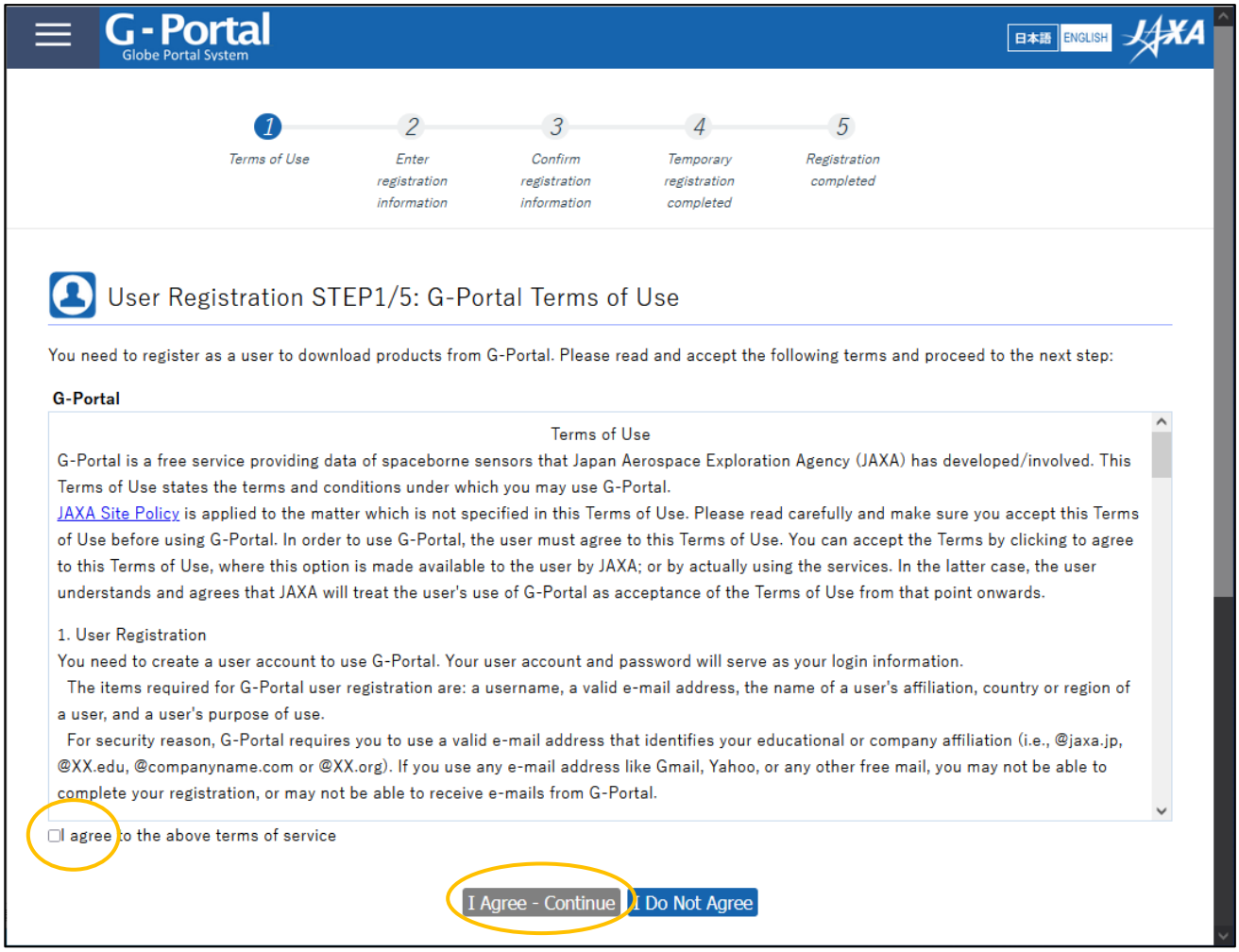
○ : Operation is confirmed. －: Operation is unconfirmed.

2. how to obtain GPM/TRMM data

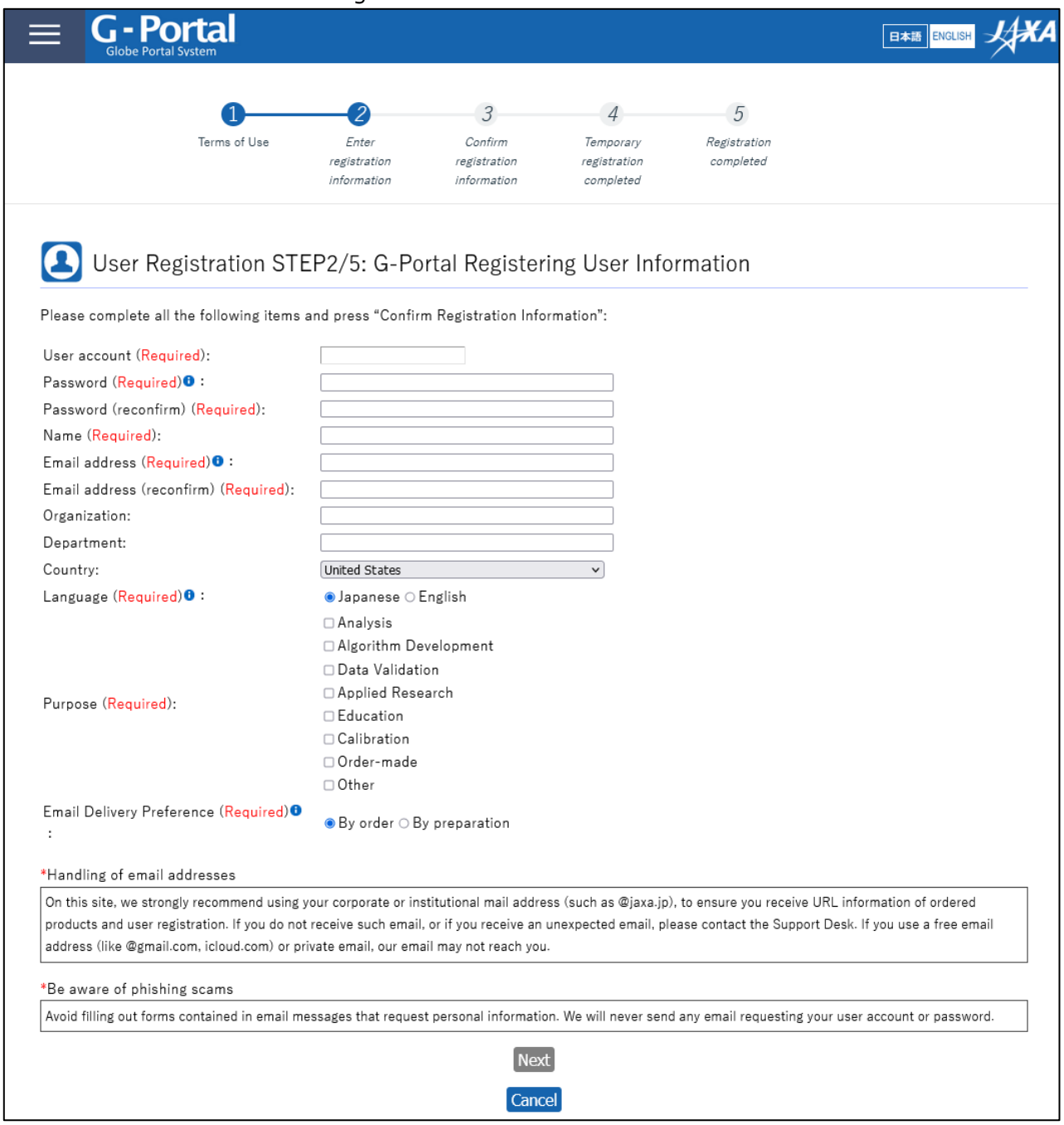
GPM/TRMM data can be obtained from the G-Portal site (https://www.gportal.jaxa.jp/gp/top.html). User registration is required to obtain the data. Please select "User Registration/Terms of Use" from the menu at the top of the G-Portal site to register as a user.



Read the terms and conditions and click "Agree and Next.



You will be taken to the user registration screen.



For the subsequent procedures and how to obtain data after user registration, please refer to "5.2 How to Use the Data Providing Service" in the "GPM Data Users Handbook. For information on how to obtain the "GPM Data Users Handbook," please refer to "3.

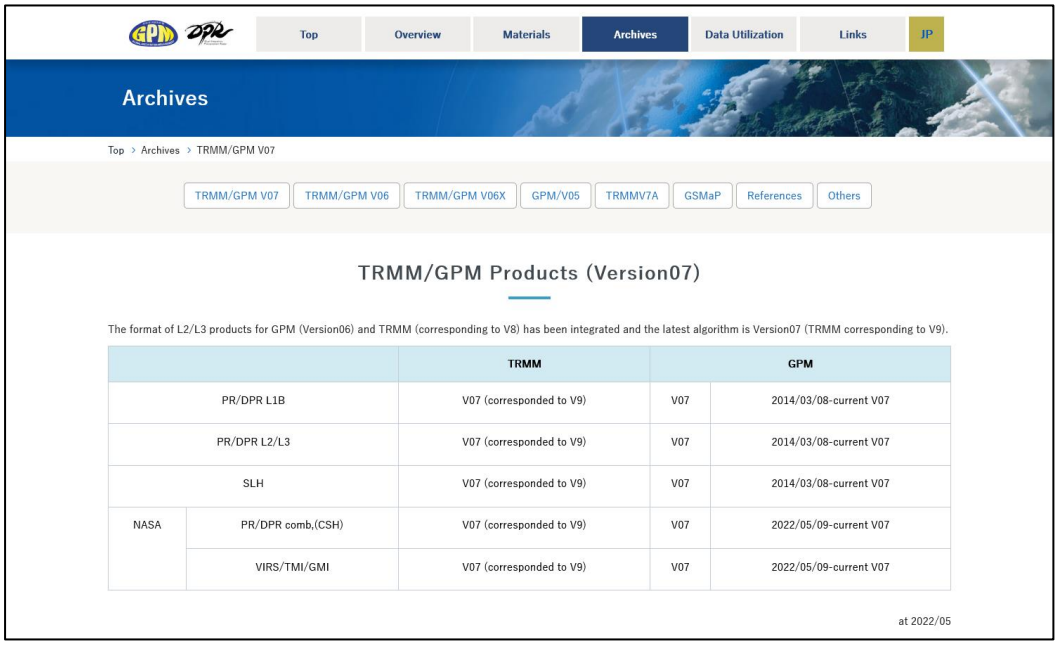
3. how to obtain related documents and sample programs

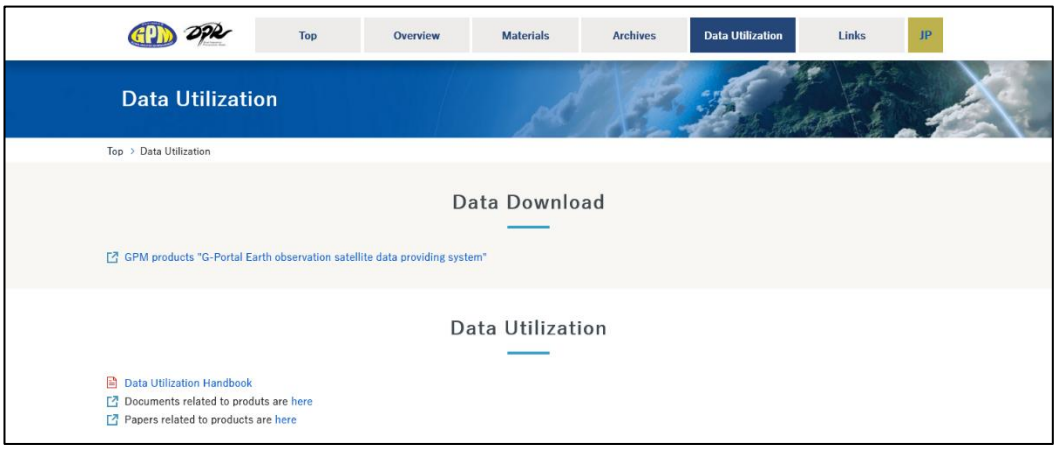
There are two types of documents related to GPM/TRMM data: documents related to data use and documents related to products. Both documents can be downloaded from the GPM website (https://www.eorc.jaxa.jp/GPM/index.html). You can also download the sample codes described in this document from Top Page > Data Utilization

Documentation for GPM data use includes

GPM Data Application Handbook

file naming convention





Click "TRMM/GPM V07" to see the list of documents for product version 07.

The products, programs, and sample data described in this document are as follows

Table 3.1 List of Sample Programs

product	sample program	sample data
L2DPR	sample_L2_DPR_IDL.pro	GPMCOR_DPR_2112070007_0140_04417_L2S_DD2_07A.h5
L3DPR	sample_L3_DPR_IDL.pro	GPMCOR_DPR_1806_M_D3M_07X.EORC.h5
GSMaP	sample_GSMaP_HDF5_IDL.pro	GPMMRG_MAP_2112010000_H_L3S_MCH_05A.h5
	sample_GSMaP_bin_IDL.pro	gsmap_gauge_now.20211201.0000.dat
	sample_GSMaP_NetCDF_IDL.pro	GPMMRG_MAP_2112010000_H_L3S_MCN_05A.nc

4. installation of library tools

When reading GPM/TRMM data in IDL, you only need to install IDL itself.
No installation of related libraries is required.

This manual has been tested in the following environments

Table 4.2 Operating Environment

(data) item	environment
calculator	Intel(R) Xeon(R) CPU ES-2665 2.4GHz
OS	Red Hat Enterprise Linux Server release 6.4
IDL	Version 8.0.1

5. GPM/TRMM data read (IDL)

IDL (Interactive Data Language) is a programming language for data analysis commonly used in scientific and technical computing.

5.1 L2 data reading

5.1.1 Source Programs

The following is an example program that reads L2DPR. It reads the data Latitude, Longitude, and precipRateESurface from the HDF5 file specified by fnL2.

```
  1:PRO sample_L2_DPR_IDL
  2:
  3: fnL2 = 'GPMCOR_DPR_2112070007_0140_044170_L2S_DD2_07A.h5'
  4:
  5: print, ' '
  6: print, '+ Input file name +'
  7: print, fnL2
  8:
  9: ;Read Dataset Sample
 10: fileID = H5F_OPEN(fnL2)
 11:
 12: dataSetLat = '/FS/Latitude'
 13: dataSetID = H5D_OPEN(fileID, dataSetName )
 14: Latitude = H5D_READ(dataSetID)
 15: H5D_CLOSE, dataSetID
 16:
 17: dataSetLon = '/FS/Longitude'
 18: dataSetID = H5D_OPEN(fileID, dataSetLon )
 19: Longitude = H5D_READ(dataSetID)
 20: H5D_CLOSE, dataSetID
 21:
 22: dataSetName = '/FS/SLV/precipRateESurface'
 33: dataSetID = H5D_OPEN(fileID, dataSetName )
 24: precipRateESurface = H5D_READ(dataSetID)
 25: H5D_CLOSE, dataSetID
 26:
 27: H5F_CLOSE, fileID
 28:
 29:;Confirmation
 30: print, ' '
 31: print, '+ Dataset +'
 32: for i=40,48 do begin
 33: print, 'Lat=',Latitude[i,1491],' Lon=',Longitude[i,1491],'
precipRateESurface=',precipRateESurface[i,1491]
 34: endfor
 35:
 36:END
```

HDF5 file name.

Open HDF5 file.

Open data sets.
fileID: fileID obtained by H5F_OPEN
dataSetLat: Specify the name of data to be read.

Data set readout
dataSetID: dataSetID obtained by H5D_OPEN

Data Set Close

HDF5 file close

A portion of the data is output to confirm that it is read correctly.

5.1.2 Execution results

The following are the results of executing the program described in 5.1.1.

```
$ idl
IDL Version 8.0.1 (linux x86_64 m64). (c) 2010, ITT Visual Information Solutions
Installation number: 70882.
Licensed for use by: JAXA

IDL> .run sample_L2_DPR_IDL.pro
% Compiled module: SAMPLE_L2_DPR_IDL.
IDL> sample_L2_DPR_IDL

+ Input file name +
.GPMCOR_DPR_2112070007_0140_044170_L2S_DD2_07A.h5
% Loaded DLM: HDF5.

+ Dataset +
Lat= -19.8801 Lon= 30.4024 precipRateESurface= 3.42141
Lat= -19.8574 Lon= 30.3586 precipRateESurface= 3.70716
Lat= -19.8345 Lon= 30.3147 precipRateESurface= 3.20117
Lat= -19.8117 Lon= 30.2709 precipRateESurface= 3.00306
Lat= -19.7884 Lon= 30.2264 precipRateESurface= 3.32433
Lat= -19.7651 Lon= 30.1819 precipRateESurface= 4.20975
Lat= -19.7416 Lon= 30.1372 precipRateESurface= 3.18332
Lat= -19.7176 Lon= 30.0917 precipRateESurface= 1.59663
Lat= -19.6936 Lon= 30.0461 precipRateESurface= 1.63730
IDL>
```

5.2 L3 data reading

5.2.1 Source Programs

The following is an example of an L3DPR reading program, reading data named precipRateESurface from a file specified by fnL3.

```
 1:PRO sample_L3_DPR_IDL
 2:
 3: fnL3 = ' GPMCOR_DPR_1806_M_D3M_07X.EORC.h5'
 4:
 5: print, ' '
 6: print, '+ Input file name +'
 7: print, fnL3
 8:
 9: fileID = H5F_OPEN(fnL3)
10:
11:;Read Dataset Sample
12: dataSetName = '/FS/G1/precipRateESurface/mean'
13: dataSetID = H5D_OPEN(fileID, dataSetName )
```

HDF5 file name.

Open HDF5 file.

Open data sets.
fileID: fileID obtained by H5F_OPEN
dataSetName: Specify the name of the data to be read.

```
14: precipRateESurface = H5D_READ(dataSetID)
15:
16: H5D_CLOSE, dataSetID
17: H5F_CLOSE, fileID
18:
19:;Confirmation
20: print, ' '
21: print, '+ Dataset +'
22: lat = (140.0/28.0) * 14 - 70.0 + (140.0/28.0/2)
23: lon = (360.0/72.0) * 63 - 180.0 + (360.0/72.0/2)
22: print, 'lat=',lat,' lon='',lon
23: print, '/FS/G1/precipRateESurface/mean[14,63,0,0,0]='
,precipRateESurface[14,63,0,0,0,0].
24:
25:END
```

Data set readout
dataSetID: dataSetID obtained by H5D_OPEN

Data Set Close

HDF5 file close

A portion of the data is output to confirm that it is read correctly.

5.2.2 Execution results

The following are the results of executing the program described in 5.2.1.

```
$ idl
IDL Version 8.0.1 (linux x86_64 m64). (c) 2010, ITT Visual Information Solutions
Installation number: 70882.
Licensed for use by: jaxa

IDL> .run sample_L3_DPR_IDL.pro
% Compiled module: SAMPLE_L3_DPR_IDL.
IDL> sample_L3_DPR_IDL

+ Input file name +
data_07A/GPMCOR_DPR_1806_M_D3M_07X.EORC.h5
% Loaded DLM: HDF5.

+ Dataset +
lat= 2.50000 lon= 137.500
/FS/G1/precipRateESurface/mean[14,63,0,0,0]= 2.13081
IDL>
```

5.3 GSMaP_HDF5 Data Image Display

5.3.1 Source prog ram

The following sample program creates an image image from a GSMaP file specified by fnL3 and displays it on the screen.

```
 1:;PRO sample_GSMaP_HDF5_IDL
 2:

 3: fnL3 = '... /... /GPMMRG_MAP_2112010000_H_L3S_MCH_05A.h5'
 4:
 5: print, ' '
 6: print, '+ Input file name +'
 7: print, fnL3
 8:
 9: ;Read Dataset Sample
10: fileID = H5F_OPEN(fnL3)

11:
12: dataSetName = '/Grid/hourlyPrecipRateGC'


13: dataSetID = H5D_OPEN(fileID, dataSetName )


14: rain_data = H5D_READ(dataSetID)
15:
16: H5D_CLOSE, dataSetID
17:
18: H5F_CLOSE, fileID
19:
20:;+++ rotate GSMaP data
21:rain_data = rotate(rain_data,4)
22:


23:;+++ convert 1-byte scale data for drawing
24:rain_byte = bytarr(3600,1800)
25:


26:tdb_rain = [0,0.1, 0.5, 1, 2, 3, 5, 10, 15, 20, 25, 1000] ; [mm/h]


27:tdb_r_elem = [255, 0, 0, 0, 51, 155, 255, 255, 255, 235, 175 ]
28:tdb_g_elem = [255, 0, 100, 180, 219, 235, 235, 179, 100, 30, 0 ]
29:tdb_b_elem = [255, 150, 250, 250, 128, 74, 0, 0, 0, 0, 0 ]
```

HDF5 file name.

Open HDF5 file.

The name of the data to be read is specified.

Open data sets.
fileID: fileID obtained by H5F_OPEN
dataSetName: Specify the name of the data to be read.

Read the data specified by dataSetID (hourlyPrecipRateGC) from the file and set it to rain_data.

The array of data read (latitude, longitude) is converted to (longitude, latitude).

It defines the data to be drawn on the map.

Defines the value of precipitation.

The color corresponding to the precipitation value is defined.

```
30:
31:num_size = size(tdb_r_elem)
32:num = num_size(1)
33:
34:for i=0L, num - 1L do begin
35: w = where( (tdb_rain[i] le rain_data) and (rain_data lt tdb_rain[i+1L]), cw )
36: if (cw ge 1) then begin
37: rain_byte(w) = i
38: endif
39:endfor ; i
40:
41:;+++ set color
42:device,retain=2,decomposed=0
43:
44:tvlct, r, g, b, /get
45:r[0:num-1L] = tdb_r_elem[*].
46:g[0:num-1L] = tdb_g_elem[*].
47:b[0:num-1L] = tdb_b_elem[*].
48:r[255]=0 & g[255]=0 & b[255]=0
49:tvlct, r, g, b
50:
51:;+++ draw on map
52:window, 1, xsize=800, ysize=400, title='GSMaP_HDF5'
53:
54:MAP_SET, 0, 0, /CYLINDRICAL, $
55: LIMIT=[-60, -180, 60, 180], pos=[0.1, 0.1, 0.9, 0.9], $
56: /noerase, /NOBORDER
57:
58:result = MAP_IMAGE(rain_byte, x0, y0, xsize, ysize, $
59: COMPRESS=1, SCALE=0.05, $
60: LATMIN=-90, LONMIN=-180, $
61: LATMAX=90, LONMAX=180)
62:
63:TV, result, x0, y0, xsize=xsize, ysize=ysize
64:
65:map_continents
66:map_grid, LABEL=1, CHARSIZE=1.0, GLINESTYLE=1, $
67: LATLAB=-15, LONLAB=-45, $
68: LONDEL=30, LATDEL=10, /BOX_AXES
69:
70:write_png,'sample_GSMaP_HDF5_IDL.png',tvrd(/true)
71:END
```

Creation of image data
For the data read in, the amount of precipitation is checked and the color (i) corresponding to the precipitation value is set to rain_byte.

Display Settings
retain=2: IDL manages drawing data
decomposed=0: Pseudo color, 0: Pseudo color, 0: Pseudo color, 0: Pseudo color

Colors are set

Map Settings
CYLINDRICAL: cylindrical equidistant projection, LIMIT: latitude and longitude of lower left and upper right
POS: lower left and upper right coordinates, noerase: do not erase the screen before drawing

Setting from image data
rain_byte: Image data, COMPRESS=1: Inverse map transformation for each pixel
LATMIN=-90: latitude corresponding to the first row of the image, LONMIN=-180: longitude corresponding to the leftmost column of the image
LATMAX=90: latitude corresponding to the last row of the image, LONMAX=180: longitude corresponding to the rightmost column of the image

Image rendering

5.3.2 Execution results

The following figure shows the results of executing the program described in 5.3.1. When the program is executed, the figure shown in Figure 5.3.1 is displayed.

```
$ idl
IDL Version 8.0.1 (linux x86_64 m64). (c) 2010, ITT Visual Information Solutions
Installation number: 70882.
Licensed for use by: jaxa

IDL> .run sample_GSMaP_HDF5_IDL_20151221.pro
% Compiled module: $MAIN$.

+ Input file name +
GPMMRG_MAP_2112010000_H_L3S_MCH_05A.h5
% Loaded DLM: HDF5.
% Compiled module: MAP_SET.
% Compiled module: MAP_IMAGE.
% Compiled module: MAP_CONTINENTS.
% Compiled module: MAP_GRID.
IDL>
```

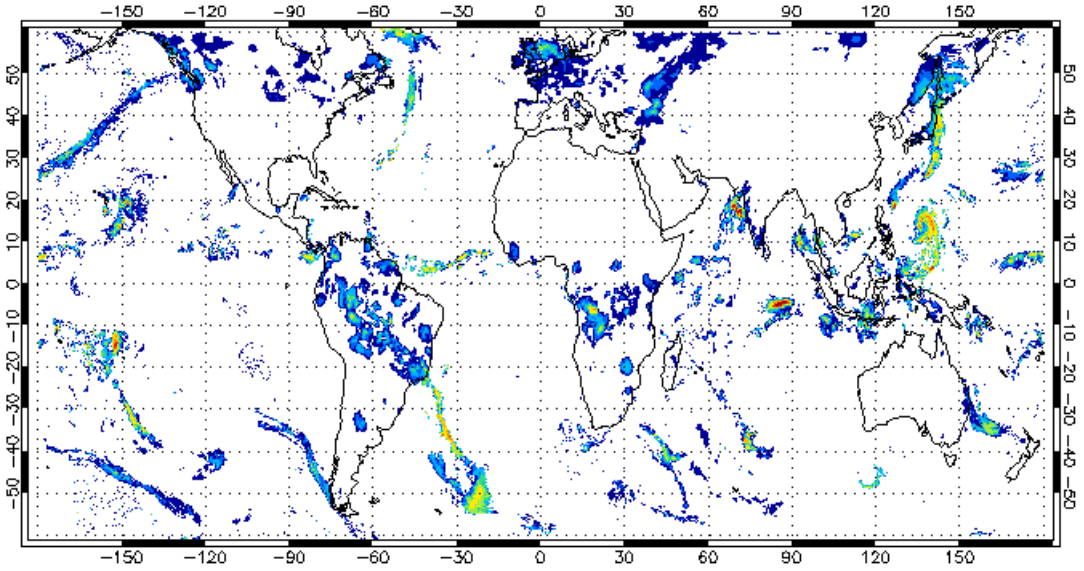


Figure 5.3.1 Execution results

5.4 GSMaP_bin data image display

5.4.1 Source Programs

The following sample program creates an image image from the GSMaP file specified in fn_bin and displays it on the screen.

```
;+++++++++++++++++++++++++++++++++++++++++++
;+++ sample code drawing GSMaP-bin data +++
;+++++++++++++++++++++++++++++++++++++++++++

;+++ input file name
fn_bin = 'gsmap_gauge_now.20211201.0000.dat'

;+++ read GSMaP_bin data
rain_data = fltarr(3600,1200)
openr, 1, fn_bin

readu, 1, rain_data
close, 1

;+++ rotate GSMaP data
rain_data = rotate(rain_data,7)

;+++ convert 1-byte scale data for drawing
rain_byte = bytarr(3600,1200)

tdb_rain = [0,0.1, 0.5, 1, 2, 3, 5, 10, 15, 20, 25, 1000] ; [mm/h]


tdb_r_elem = [255, 0, 0, 0, 51, 155, 255, 255, 255, 235, 175 ]
tdb_g_elem = [255, 0, 100, 180, 219, 235, 235, 179, 100, 30, 0 ]
tdb_b_elem = [255, 150, 250, 250, 128, 74, 0, 0, 0, 0, 0, 0 ]

num_size = size(tdb_r_elem)
num = num_size(1)


for i=0L, num - 1L do begin
 w = where( (tdb_rain[i] le rain_data) and (rain_data lt tdb_rain[i+1L]), cw )
 if (cw ge 1) then begin
 rain_byte(w) = i
 endif
endfor ; i


;+++ set color
device,retain=2,decomposed=0
```

Binary file name.

Open binary file.

Binary file loading.

The image is rotated 270 degrees (90 degrees to the right) and the top and bottom are reversed.

Defines the value of precipitation.

The color corresponding to the precipitation value is defined.

Creation of image data
For the data read in, the amount of precipitation is checked and the color (i) corresponding to the precipitation value is set to rain_byte.

Display Settings
retain=2: IDL manages drawing data
decomposed=0: Pseudo color, 0: Pseudo color, 0: Pseudo color, 0: Pseudo color

```
 37:
38:tvlct, r, g, b, /get
39:r[0:num-1L] = tdb_r_elem[*].
40:g[0:num-1L] = tdb_g_elem[*].
41:b[0:num-1L] = tdb_b_elem[*].
42:r[255]=0 & g[255]=0 & b[255]=0
43:tvlct, r, g, b
44:
45:;+++ draw on map
46:window, 0, xsize=800, ysize=400, title='GSMaP_bin'
47:
48:MAP_SET, 0, 180, /CYLINDRICAL, $
49: LIMIT=[-60, 0, 60, 360], pos=[0.1, 0.1, 0.9, 0.9], $
50: /noerase, /NOBORDER
51:
52:result = MAP_IMAGE(rain_byte, x0, y0, xsize, ysize, $
53: COMPRESS=1, SCALE=0.05, $
54: LATMIN=-60, LONMIN=0, $
55: LATMAX=60, LONMAX=360)
56:
57:TV, result, x0, y0, xsize=xsize, ysize=ysize
58:
59:map_continents
60:map_grid, LABEL=1, CHARSIZE=1.0, GLINESTYLE=1, $
61: LATLAB=-15, LONLAB=-45, $
62: LONDEL=30, LATDEL=10, /BOX_AXES
63:
64: write_png,'sample_GSMaP_bin_IDL.png',tvrd(/true)
65:END
```

Colors are set

Map Settings
CYLINDRICAL: cylindrical equidistant projection, LIMIT: latitude and longitude of lower left and upper right
POS: lower left and upper right coordinates, noerase: do not erase the screen before drawing
NOBORDER: Do not draw a border around the map

Setting from image data
rain_byte: Image data, COMPRESS=1: Inverse map transformation for each pixel
LATMIN=-90: latitude corresponding to the first row of the image, LONMIN=-180: longitude corresponding to the leftmost column of the image
LATMAX=90: latitude corresponding to the last row of the image, LONMAX=180: longitude corresponding to the rightmost column of the image

Image rendering

5.4.2 Execution results

The following figure shows the results of executing the program described in 5.4.1. When the program is executed, the figure shown in Figure 5.4.1 is displayed.

```
$ idl
IDL Version 8.0.1 (linux x86_64 m64). (c) 2010, ITT Visual Information Solutions
Installation number: 70882.
Licensed for use by: jaxa

IDL> .run sample_GSMaP_bin_IDL.pro
% Compiled module: $MAIN$.
% Compiled module: MAP_SET.
% Compiled module: MAP_IMAGE.
% Compiled module: MAP_CONTINENTS.
% Compiled module: MAP_GRID.
IDL>
```

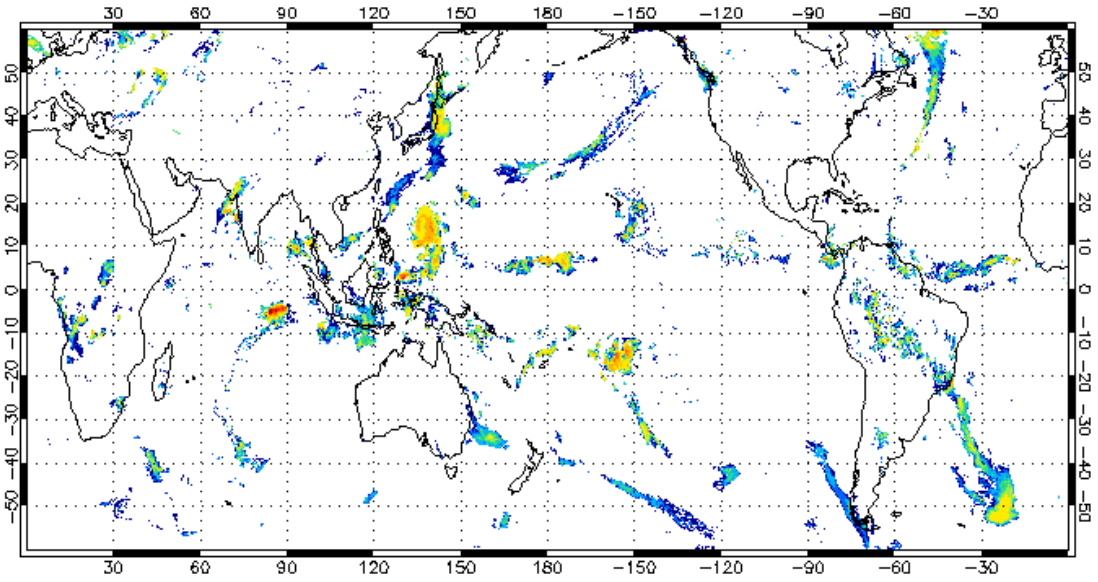


Figure 5.4.1 Execution results

5.5 GSMaP_NetCDF data image display

5.5.1 Source Programs

The following sample program creates an image image from a GSMaP file specified by fnL4 and displays it on the screen.

```
 1:PRO sample_GSMaP_NetCDF_IDL
 2:
 3:fnL4 = '... /... /data_07A/GPMMRG_MAP_2112010000_H_L3S_MCN_05A.nc'
 4:print, ' '
 5:print, '+ Input file name +'
 6:print, fnL3
 7:
 8:; Read Dataset Sample
 9:; fileID = H5F_OPEN(fnL3)
10:;
11:; dataSetName = '/Grid/hourlyPrecipRateGC'
12:; dataSetID = H5D_OPEN(fileID, dataSetName )
13:; rain_data = H5D_READ(dataSetID)
14:; H5D_CLOSE, dataSetID
15:
16:; H5F_CLOSE, fileID
17:
18:; Read Dataset Sample with netcdf
19: fileID = NCDF_OPEN(fnL4)
20:
21: dataSetName = 'hourlyPrecipRateGC'
22: dataSetID = NCDF_VARID(fileID, dataSetName )
23: NCDF_VARGET, fileID, dataSetID, rain_data
24:
25: NCDF_CLOSE, fileID
26:
27:;+++ rotate GSMaP data
28:rain_data = rotate(rain_data,4)
29:
30:; +++ convert 1-byte scale data for drawing
31:rain_byte = bytarr(3600,1800)
32:
33:tdb_rain = [0,0.1, 0.5, 1, 2, 3, 5, 10, 15, 20, 25, 1000] ; [mm/h]
34:tdb_r_elem = [255, 0, 0, 0, 51, 155, 255, 255, 255, 235, 175 ]
35:tdb_g_elem = [255, 0, 100, 180, 219, 235, 235, 179, 100, 30, 0 ]
36:tdb_b_elem = [255, 150, 250, 250, 128, 74, 0, 0, 0, 0, 0, 0 ]
37:
38:num_size = size(tdb_r_elem)
39:num = num_size(1)
40:
```

NetCDF file name.

Open NetCDF file.

NetCDF file loading.

The array of data read (latitude, longitude) is converted to (longitude, latitude).

Defines the value of precipitation.

The color corresponding to the precipitation value is defined.

```
41:for i=0L, num - 1L do begin
42: w = where( (tdb_rain[i] le rain_data) and (rain_data lt tdb_rain[i+1L]), cw )
43: if (cw ge 1) then begin
44: rain_byte(w) = i
45: endif
46: endfor ; i
47:
48:;+++ set color
49:device,retain=2,decomposed=0
50:
51:tvlct, r, g, b, /get
52:r[0:num-1L] = tdb_r_elem[*].
53:g[0:num-1L] = tdb_g_elem[*].
54:b[0:num-1L] = tdb_b_elem[*].
55:r[255]=0 & g[255]=0 & b[255]=0
56:tvlct, r, g, b
57:
58:;+++ draw on map
59:window, 1, xsize=800, ysize=400, title='GSMaP_HDF5'
60:
61: MAP_SET, 0, 0, /CYLINDRICAL, $
62: LIMIT=[-60, -180, 60, 180], pos=[0.1, 0.1, 0.9, 0.9], $
63: /noerase
64:; /noerase, /NOBORDER
65:
66: result = MAP_IMAGE(rain_byte, x0, y0, xsize, ysize, $
67:; COMPRESS=1, SCALE=0.01, $
68: COMPRESS=1, $
69: LATMIN=-90, LONMIN=-180, $
70: LATMAX=90, LONMAX=180)
71:
72:TV, result, x0, y0, xsize=xsize, ysize=ysize
73:
74:map_continents
75:map_grid, LABEL=1, CHARSIZE=1.0, GLINESTYLE=1, $
76: LATLAB=-15, LONLAB=-45, $
77: londel=30, latdel=10, /box_axes
78:
79:write_png,'sample_GSMaP_NetCDF_IDL.png',tvrd(/true)
80:
81:END
```

Creation of image data
For the data read in, the amount of precipitation is checked and the color (i) corresponding to the precipitation value is set to rain_byte.

Display Settings
retain=2: IDL manages drawing data
decomposed=0: Pseudo color, 0: Pseudo color, 0: Pseudo color, 0: Pseudo color

Colors are set

Map Settings
CYLINDRICAL: cylindrical equidistant projection, LIMIT: latitude and longitude of lower left and upper right
POS: lower left and upper right coordinates, noerase: do not erase the screen before drawing
NOBORDER: Do not draw a border around the map

Setting from image data
rain_byte: Image data, COMPRESS=1: Inverse map transformation for each pixel
LATMIN=-90: latitude corresponding to the first row of the image, LONMIN=-180: longitude corresponding to the leftmost column of the image
LATMAX=90: latitude corresponding to the last row of the image, LONMAX=180: longitude corresponding to the rightmost column of the image

Image rendering

5.5.2 Execution results

The following figure shows the results of executing the program described in 5.5.1. When the program is executed, the figure shown in Figure 5.5.1 is displayed.

```
$ idl
IDL> .run sample_GSMaP_NetCDF_IDL.pro
% Compiled module: SAMPLE_GSMAP_NETCDF_IDL.
IDL> sample_GSMaP_NetCDF_IDL

+ Input file name +
... /... /data_07A/GPMMRG_MAP_2112010000_H_L3S_MCN_05A.nc
% Loaded DLM: NCDF.
% Compiled module: MAP_SET.
% Loaded DLM: LAPACK.
% Compiled module: MAP_IMAGE.
% Compiled module: MAP_CONTINENTS.
% Compiled module: MAP_GRID.
% Loaded DLM: PNG.
```

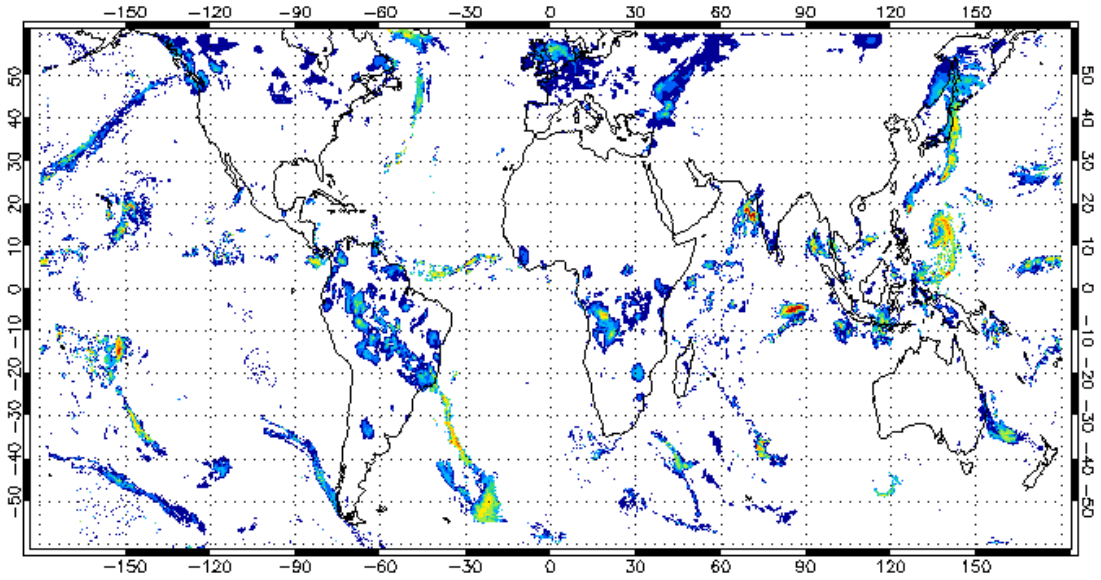


Figure 5.5.1 Execution results

revision history

version number	Date	Revised contents	remarks
1	2016/1/26		
2	2016/9/26	4. installation of libraries and tools: installation of HDF5 Delete description of	
3	2017/9/13	1. Introduction: python description added to Table 1.1, flowchart revised accordingly. Table 1.2 Sample code operation check table was added. 4. installation of library tools: Table 4.2 Product bar version and PPS Toolkit (TKIO). In Table 4.3 Operating Environment, where it says tkio-3.70.7 Changed to tkio-x.xx.x 4.2.4Edit configuration file: Change "tkio-3.70.7" to "tkio-x.xx.x".	
4	3/15/2018	3. Related documents and sample programs available: Table 3.1 sample program list added.	
5	2/5/2019	1.-3. Correction due to addition of TRMM and renewal of GPM site 5.1, 5.2 Item name change	
6	12/6/2021	1. modified to GSMaP product version 5 and GPM/TRMM product version 7. 3. revised availability of related documentation and sample programs	
7	12/24/2021	Table 3.1 Sample data updated to V7 6 Corrected code description to match V7	
8	2/21/2022	5.1, 5.2 lat/lon display added. Corrected errors. 5.5 Add GSMaP_NetCDF data display	