

Pythonによるひまわり8号データの可視化

誌名	日本リモートセンシング学会誌 = Journal of the Remote Sensing Society of Japan
ISSN	02897911
著者名	豊嶋, 紘一
発行元	日本リモートセンシング学会
巻/号	41巻4号
掲載ページ	p. 493-499
発行年月	2021年9月

農林水産省 農林水産技術会議事務局筑波産学連携支援センター
Tsukuba Business-Academia Cooperation Support Center, Agriculture, Forestry and Fisheries Research Council
Secretariat



事例紹介

Python によるひまわり 8 号データの可視化

豊嶋紘一^{*1†}

Visualization of Himawari-8 data using Python

Koichi TOYOSHIMA^{*1†}

Keywords : Geostationary Meteorological Satellite, Python, Satellite Data, Himawari-8

1. はじめに

本紹介記事では、千葉大学でアーカイブ・公開を行っている静止気象衛星データのひまわり 8 号データを使って、Python による取得から可視化までの手順を紹介する。ひまわり 8 号は従来のひまわり 7 号と比べて性能が向上して、より詳細に地球を観測できるようになり、気象現象のみならず“地球環境をモニタリングする衛星”としても利用の幅が広がっている。本記事で紹介する実習の目的は、衛星データファイルの取り扱い方を理解して、目的に応じて取得・処理できるようにすることである。ここでは近年人気の高い Python を使って衛星データの可視化を行う。Python を用いる利点は C や Fortran と比べるとシンプルで少ないコードで実行できる点や、コンパイル作業が不要ですぐにコードを実行できる点が挙げられる。この実習は各自 web ブラウザを立ち上げてオンライン上の Python 環境ですぐに始めることができるようになっており、PC 環境のスペックにとらわれず手軽に実行可能である。本記事の内容は 2021 年に千葉大学で行われた講習会「第 14 回地球気候系の診断に関わるバーチャルラボラトリー形成」で実施した実習内容をアレンジしたもので、詳細については VL 講習会演習テキスト¹⁾に説明があるので参考にされたい。

2. 実習環境の準備

実習を行うにあたり Python プログラムを実行する環境を用意する。一般に Python を使用する際には、最初に PC へのインストール作業が必要となるが、今回は Google にログインするだけで Python 環境が利用可能な Google Colaboratory (以下 Colab) を利用する。Colab は Google が

機械学習の教育研究を目的として開発したツールで、Ubuntu システム上で Jupyter Notebook といわれるノートブック形式で作成したプログラムを実行し、その結果を確認しながらデータ解析を行うことができる。Jupyter Notebook では「Python のコード」と「システムコマンド」の両方が使用可能であるが、後者はシステムの OS の種類に依るので、Colab の場合 Linux の一つである Ubuntu のコマンドを用いる。例えば、コードの先頭に「!」を付加すると Ubuntu コマンドを実行することができる。Colab に関する詳細は Google の説明ページ²⁾を参照されたい。

Google にログインした上で Colab³⁾ にアクセスすると Fig. 1 に示すような画面を確認することができる。中央のグレーのボックス (セル) が Python コードを入力する箇所である。ここに任意のコードを入力して左隣の三角の「実行ボタン」を押す (若しくは Shift+Enter キーでも可) ことでコードを実行することができる。実行結果はセルの下部に表示される。新たにセルを追加する場合はセルの下側にマウスカーソルを近づけると「+コード」と表示されるので、クリックすると新しいセルが追加される。Colab の操作に関しては VL 演習テキスト¹⁾に掲載しているので参照されたい。使用する web ブラウザによって挙動がやや異

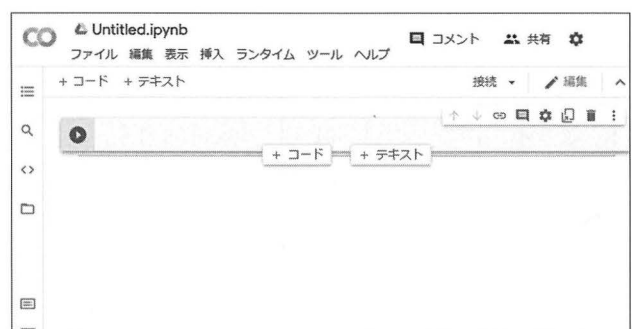


Fig. 1 The screen of Colab interface page.

(2021. 7. 6 受付, 2021. 8. 30 改訂受理)

^{*1} 千葉大学 環境リモートセンシング研究センター
〒265-8522 千葉市稲毛区弥生町 1-33

[†] 連絡著者 (Corresponding author)
E-mail: toyoshima@chiba-u.jp

^{*1} Center for Environmental Remote Sensing (CEReS), Chiba University, 1-33 Yayoi, Inage, Chiba, Japan

なることがあるが、推奨 web ブラウザには Google Chrome, Firefox, Safari が指定されている。以降に示す Python のサンプルコードは Firefox 上で実行することを想定して作成している。

3. 千葉大学 CEReS のひまわり 8 号データアーカイブについて

静止気象衛星は定常観測を行うことで、広範囲に高い時間分解能を持つ可視・赤外域の観測データを得ることができ、長期にわたる一貫したデータのアーカイブは気候変動解析を行う上でも重要である。千葉大学 環境リモートセンシング研究センター (CEReS) では、ひまわり 1 号からひまわり 8 号に至るまでの静止気象衛星のデータ・アーカイブを行っている。ひまわり 8 号データは気象庁による受信・データ処理の後、リアルタイムで東京大学, JAXA, NICT, 千葉大学の 4 機関に提供され、利用者はここを経由して取得することができる。CEReS では、気象庁からリアルタイムに取得したデータを精密幾何補正し、緯度経度直交座標に変換した全球 Gridded データとしてアーカイブしており、FTP⁴⁾にてデータの取得が可能である。また利用者には Gridded データの取得から読み込み、物理量変換、可視化までの一連の処理を自動で行う便利なスクリプトを含むサンプルコード (Python, Fortran, C 言語) のセット (count2tbb_v102.tgz⁵⁾) の提供も行っている。

Gridded データには、これまで CEReS でアーカイブを行ってきた静止気象衛星データとの関連性から、気象庁とは異なるルールでバンド名が付けられている。Table 1 にひまわり 8 号のオリジナルの観測バンド名と Gridded デー

タのバンド名の対応を示す。例えば、最も高い分解能を持つ可視バンド (Band03) は、Gridded データでは EXT01 というバンド名に対応する。CEReS Gridded データを利用される際には Table 1 を利用して読み替えて利用していただきたい。Gridded データのファイル名は yyyyymmddhhnn.BBB.CC.fld.geoss のような規則で決められており、「年, 月, 日, 時, 分」と続いた後、BBB と CC には先に説明した Gridded データのバンド名と番号、fld は全球を意味する観測領域名で構成されている。ここで使用される時刻は世界協定時刻 (UTC) である。ひまわり 8 号は東経 140.7 度の赤道上に位置し、データの範囲は東経 85 度から 205 度、北緯 60 度から南緯 60 度である。データの並びは西から東向きの带状で、北から南の順である。データ型は 2 バイト符号なし整数型、ビッグエンディアンバイトオーダーで記録されている。データを利用する際には、DN 値を物理量 (可視反射率や赤外輝度温度) に変換する必要がある、そのためルックアップテーブル (LUT) は Gridded データのサンプルプログラムセット (count2tbb_v102.tgz⁵⁾) に含まれている。LUT には DN 値に対応する物理量が記されている。

4. CEReS 公開の Gridded データ処理

ここでは Gridded データの取得と可視化の方法を紹介する。Table 2 にデータ取得と物理量変換のサンプルコードを示す。サンプルコードは紙面の都合上 2 行にわたって書かれている場合があるが、Colab のセルには、同じ行番号であれば改行せず入力していただきたい。冒頭 (2 から 6 行目) に「import」と書かれている部分では今回の処理で必

Table 1 List of Himawari-8 bands.

	CEReS gridded data 対応バンド		気象庁ひまわり8/9バンド 中心波長	階調	Pixel x Line	空間解像度
可視	EXT	01	Band 03 (0.64 μm)	2048/11bit	24000 x 24000	0.005 degree (approx. 500 m)
		01	Band 01 (0.47 μm)		12000 x 12000	0.01 degree (approx. 1km)
	VIS	02	Band 02 (0.51 μm)			
近赤外		SIR	03		Band 04 (0.86 μm)	6000 x 6000
	01		Band 05 (1.6 μm)			
	02	Band 06 (2.3 μm)				
赤外	TIR	01	Band 13 (10.4 μm)	4096/12bit		
		02	Band 14 (11.2 μm)			
		03	Band 15 (12.4 μm)			
		04	Band 16 (13.3 μm)	2048/11bit		
		05	Band 07 (3.9 μm)	16384/14bit		
		06	Band 08 (6.2 μm)	2048/11bit		
		07	Band 09 (6.9 μm)			
		08	Band 10 (7.3 μm)	4096/12bit		
		09	Band 11 (8.6 μm)			
		10	Band 12 (9.6 μm)			

Table 2 Sample code of Himawari data download and conversion to tbb.

1	!pip install wget
2	import wget
3	import bz2
4	import os
5	import numpy as np
6	import tarfile
7	wget.download("ftp://hmwr829gr.cr.chiba-u.ac.jp/gridded/FD/V20151105/201507/TIR/201507070000.tir.01.fld.geoss.bz2")
8	fname = "201507070000.tir.01.fld.geoss.bz2"
9	dbuf = bz2.BZ2File(fname).read()
10	dataDN = np.frombuffer(dbuf, dtype='>u2').reshape(6000,6000)
11	os.remove(fname)
12	dataDN
13	wget.download('ftp://hmwr829gr.cr.chiba-u.ac.jp/gridded/FD/support/count2tbb_v102.tgz')
14	with tarfile.open('count2tbb_v102.tgz', 'r:gz') as t:
15	t.extractall(path='src')
16	%cd src
17	DN,tbb = np.loadtxt('tir.01', unpack = True)
18	print(DN,tbb)
19	dataTBB = tbb[dataDN]
20	dataTBB

要なモジュールを含むライブラリの読み込みを行っている。ライブラリの読み込みは一般にコードを実行する前に一度記述すれば良い。Python がシンプルなコードで様々な処理を行うことができるのは様々な機能を持つライブラリが利用できるためである。ライブラリはいくつかの機能をまとめたパッケージ群であり、Python に標準で導入されているものと、必要な場合にオプションとして導入するものがある。例えば「wget」ライブラリはファイルをダウンロードするために使用するが、Colab の Python には標準で含まれていないライブラリであるため、Python パッケージ管理システムである「pip」コマンドを使用して追加している。「bz2」ライブラリは bz2 形式で圧縮されたファイルを解凍するために使用し、「os」ライブラリはファイル操作、「NumPy」ライブラリはデータを配列に代入して数値計算を行うために使用している。では早速ひまわりデータの取得を行う。「wget.download ()」を用いて、取得したいファイルパスを指定することでダウンロードすることができる。ここではひまわり 8 号が正式運用を開始した日である 2015 年 7 月 7 日の 00UTC の時刻における熱赤外 (Band 13: tir.01) データを取得・可視化する例を示す。データ公開サーバ⁶⁾ には、「年月」、「バンド名」のサブディレクトリ

に階層化されており、上述の命名規則で定義されたファイル名のデータが配置されている。サンプルコードを参考にファイルパスを指定してダウンロードを行う。データファイルは bz2 形式で圧縮されているため、まず「bz2」モジュールで解凍を行い、2 バイト整数型ビッグエンディアンとして配列「dataDN」に代入している。NumPy には「np.frombuffer」関数があり、メモリ上のデータを直接読み込むことができることから、大容量のデータをファイル書き出しすることなく処理できるため、高速化につながる。NumPy における符号なし 16 ビット整数型のデータ型・型コードは「uint16」または「u2」のように指定し、「<」はビッグエンディアンとして読むことを意味している。その他の型の例として、32 ビット単精度浮動小数点型のデータ型・型コードは「float32」または「f4」などがある。「reshape ()」を用いているのは、読み込んだ一連のデータを 2 次元配列に「変換」するためである。今回扱う赤外バンドの場合、経度緯度方向にそれぞれ 6000 のデータが並んでいるため、「6000,6000」と指定している。次に「os.remove ()」としてダウンロードしたデータファイルを削除しているが、この作業は任意である。ひまわり 8 号データはファイルサイズが大きいため複数のデータ処理を行うと作業ディ

Table 3 Sample code to visualize a NumPy array using matplotlib.

1	%matplotlib inline
2	import matplotlib.pyplot as plt
3	import matplotlib.cm as cm
4	plt.imshow(dataTBB,cmap="gray_r")
5	plt.xlabel('X')
6	plt.ylabel('Y')
7	plt.title("Himawari-8 Band13 201507070000", size=10)
8	plt.colorbar().set_label('DN',size=15)

スクの容量を圧迫するので、今回は配列に代入した後に削除している。ここで配列「dataDN」に正常にひまわりデータが入っていることを確認する。12 行目に書かれている通り「dataDN」と変数名をセルに入力して実行ボタンを押すと、2 次元配列となったひまわりデータ（物理量変換前の値である DN 値）を表示することができる。（確認作業が不要であればスキップする）

次に物理量変換を行うため、ルックアップテーブル (LUT) を取得する。「wget.download ()」を用いて、13 行目のようにファイルパスを指定してファイルを取得の後、圧縮ファイルを「tarfile」モジュールで gzip 圧縮ファイルの読み込み用にオープンして、「extractall」メソッドで「src」ディレクトリを指定して展開している。この処理を行う 15 行目では先頭にインデント（字下げ）を入れている。Python ではこのインデントに重要な意味があり、open 文の範囲を示す。半角スペースやタブを用いるが全角スペースは使用できないので注意が必要である。Colab ではカレントディレクトリ（現在作業しているディレクトリ）を変更する場合、「%cd」を用いる。バンド名に対応する LUT、ここでは「tir.01」ファイルを開いて、テキストファイルを読み込む関数である「np.loadtxt ()」を用いて「DN」、「tbb」のそれぞれの配列に代入している。「print (DN, tbb)」として、データが正常に代入されているかを表示して確認している。続いて LUT をもとに DN 値のデータ（配列「dataDN」）を輝度温度のデータ（配列「dataTBB」）へ置き換えを行う。NumPy 配列は配列番号を指定することで要素を取り出すことができる。Python の配列番号は 0 から始まる。例えば、ある変数の配列番号に 0 を指定すると、1 つ目のデータを取り出すことができる。LUT から読み込んだ輝度温度値の配列「tbb」は、0 から始まる DN 値に対応した輝度温度値が並んでおり、配列番号と DN 値が一致している。したがって配列「tbb」の配列番号に任意の DN 値を指定すると、それに対応する輝度温度値を返す。サンプルコードの例では「tbb」の配列番号にデータ配列「dataDN」を入れることで、対応する物理量（dataTBB）へ変換を行っている（tbb [1000] とすると、DN 値が 1000 のときの輝度温度を返す）。変換がうまく行われたことを確認するには、セルに「dataTBB」と入力して実行ボタンを押すと、輝度温度値

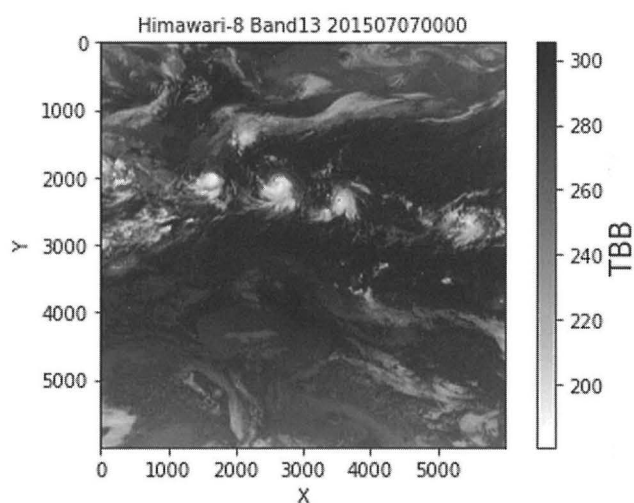


Fig. 2 Visualization of Himawari-8 data using matplotlib.

に変換されたひまわりデータが確認できる。

データを簡単に可視化するには Python のグラフ描画や画像表示のための標準ライブラリである「matplotlib」を利用するのが一般的である。配列「dataTBB」を表示する例を Table 3 に示す。サンプルコードでは「matplotlib」ライブラリをインポートして、カラーマップに（cmap = "gray_r"）としてグレースケールを指定し、さらに x 軸、y 軸のラベル指定と図のタイトル設定を行うシンプルな方法を示している。Fig. 2 はデータを可視化した例を示している。カラーバーや表示範囲を任意に変更することで、各自の目的に応じた図を作成することができる。今回は Band 13 の熱赤外データを例に取り扱ったが、可視や近赤外など他のバンドの可視化も試していただきたい。ひまわり画像に地図を重ねて表示する方法の一つとして、地球物理分野で多く用いられている GMT⁷⁾ による方法を紹介したい。Python インターフェースでも利用できるようになったので、次のセクションで取り上げる。

5. GMT を用いたデータの可視化

マップ描画ツールの一つに GMT⁷⁾ (Generic Mapping

Tool) があり, PostScript (EPS) として出力するためきれいな図が得られるのが特徴である。Python でマップを描画するには Basemap や Cartopy などを用いる方法があるが, GMT の新しいバージョンでは機能強化がなされ, Python インターフェース (PyGMT⁸⁾) でも図の描画が可能となり, データの読み込みから可視化までの一連の作業が行いやすくなった。詳細なチュートリアルは PyGMT⁸⁾ のサイトに掲載されているので参考にされたい。Colab 環境でも拡張ライブラリをインストールすることで PyGMT を利用することができる。ここでは Colab 環境への導入方法と PyGMT による描画の例を紹介する。

Table 4 で紹介するサンプルコードは GMT Ver.6 を Colab にインストールする例を示しており, 海岸線データは Colab の Ubuntu パッケージマネージャである「apt」コマンドを使用して導入した後に, Python のパッケージ管理ツール「pip」コマンドで「PyGMT」を導入している。本来ならばインストールについての解説も必要であるが紙面の都合により詳細は割愛する。

「PyGMT」でデータを描画するためには, 先に読み込んだ NumPy 配列データに緯度経度情報を付加する必要がある。Table 5 のようにして地理情報が付加された xarray 形式の配列に変換する。ひまわり 8 号データは東経 85 度から 205 度まで 0.02 度間隔, 南緯 60 度から北緯 60 度まで

0.02 度間隔であるため, 間隔を指定して連番を生成するコマンドである「np.arange」を用いて一連の緯度経度のタグを作成し, 4 行目から 8 行目では, このタグを 2 次元のひまわりデータに付加して, 配列「xr_dataTBB」としている。

次に PyGMT でひまわりデータを描く例を Table 6 に示す。最初にインストールした「PyGMT」ライブラリをインポートして, 地図の描画指定を行っていく。「MAP_FRAME_TYPE = "plain"」では地図の枠線を実線に指定し, 「region = [122.5, 154, 20, 48]」の部分は描画する緯度経度範囲を指定している。「cmap = "gray"」はカラーバーの色をグレースケールに指定して, 「series = [190, 300, 10]」は描画するデータのレンジ (ここでは表示する輝度温度値の幅を指定しており, 190 K から 300 K まで 10 K 間隔) を指定, 「fig.basemap (frame =)」以降はそれぞれタイトル, ラベルを指定している。(10 行目の「x+」や「y+」の後の文字は小文字の「l (エル)」で「ラベル」を意味する) このコードを実行すると Fig. 3 のようなひまわり画像に海岸線が描かれた図が得られる。他にも描画領域や海岸線の色などのパラメータを変更して試してみたい。

6. おわりに

ここまでオンラインの Python 環境を利用して比較的シ

Table 4 Installation code for GMT and PyGMT.

1	!sudo apt install -y gmt-gshhg gmt-dcw ghostscript build-essential cmake
2	!sudo apt install -y libcurl4-gnutls-dev libnetcdf-dev gdal-bin libgdal-dev
3	!sudo apt install -y libfftw3-dev libpcre3-dev liblapack-dev libblas-dev
4	!sudo apt install -y libgl2.0-dev xdg-utils
5	GMT_VERSION = "6.2.0"
6	!wget "https://github.com/GenericMappingTools/gmt/releases/download/{GMT_VERSION}/gmt-{GMT_VERSION}-src.tar.gz"
7	!tar xzvf gmt-{GMT_VERSION}-src.tar.gz
8	%cd gmt-{GMT_VERSION}
9	!cat ./cmake/ConfigUserTemplate.cmake > ./cmake/ConfigUser.cmake
10	!mkdir build
11	%cd build
12	!sudo cmake -DCMAKE_INSTALL_PREFIX=/usr/local -DCMAKE_BUILD_TYPE=RelWithDebInfo ..
13	!sudo make -j2
14	!sudo make -j2 install
15	!pip install pygmt

Table 5 Code to attach the geolocation information to Himawari data array.

1	import xarray as xr
2	lon1d = np.arange(85.0, 205.0, 0.02).astype(np.float64)
3	lat1d = np.arange(60.0, -60.0, -0.02).astype(np.float64)
4	xr_dataTBB = xr.DataArray(np.float32(dataTBB), name='Himawari-8',
5	coords={
6	'lat':('lat',lat1d,{ 'units':'Degrees_North'}),
7	'lon':('lon',lon1d,{ 'units':'Degrees_East'}),
8	dims=['lat','lon'])
9	xr_dataTBB

Table 6 Himawari data visualization code using PyGMT.

1	import pygmt
2	fig = pygmt.Figure()
3	pygmt.config(MAP_FRAME_TYPE="plain")
4	fig.basemap(region=[122.5, 154, 20, 48], projection="Q15c", frame=True)
5	with pygmt.config(COLOR_FOREGROUND="red", COLOR_BACKGROUND="black"):
6	pygmt.makecpt(7 cmap="gray", series=[190, 300, 10], continuous=True, M=True, reverse=True, 8)
9	fig.grdimage(grid=xr_dataTBB, projection="Q15c")
10	fig.colorbar(frame=["a10", "x+ITBB", "y+I(K)"])
11	fig.coast(resolution="h", shorelines="1/1p,green")
12	fig.basemap(frame=["ag", "+tHimawari-8_Band13"])
13	fig.text(text="20150707 00:00", x=128, y=22, font="12p,Helvetica- Bold,blue", angle=0, fill="gray")
14	fig.savefig("map.png")
15	fig.show()

ンプルな方法で、ひまわり 8 号 Gridded データの取得から変換、可視化までの流れを説明した。これを応用して各自の研究に役立てて貰えれば幸いである。紙面の制約により Python の各コマンドの説明が十分でない部分も多く、Web の情報や VL テキスト¹⁾ も参照して補足していただきたい。また Colab を用いた実習環境は無料のオンラインサービスであるため、手軽に使える反面、計算リソースなど制限がある。さらに多くのデータを使用して計算処理を行う場合には、自身の PC に Python 環境を構築することをお勧めしたい。また、可視化ツールである PyGMT・GMT Ver.6 は比較的高い頻度で更新リリースが行われている。「PyGMT」のインストールの際には最新版の GMT Ver.6 を

必要とするため、Table 4 に紹介しているサンプルコードを参照して GMT をインストールするには GMT の公式サイト⁷⁾ を参照の上、Table 4 の 5 行目部分「GMT_VERSION = "6.2.0"」を最新のバージョンに置き換えて導入していただきたい。

謝辞：ひまわり 8 号 Gridded データは千葉大学環境リモートセンシング研究センター（CEReS）から取得し、ハワイ大学のマップ描画ツール「GMT」を使わせていただきました。

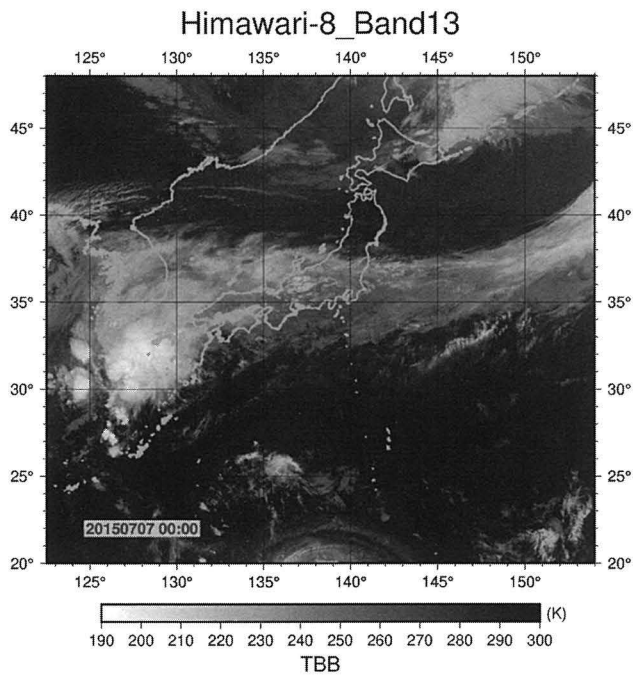


Fig. 3 Visualization of Himawari-8 data using PyGMT.

引用文献

- 1) 千葉大学環境リモートセンシング研究センター：「地球気候系の診断に関わるパーチャラボラトリー (VL) の形成」2020 年度講習会 Python によるひまわり 8 号衛星データの可視化, <https://drive.google.com/file/d/1KOs8cSRk7ZVWUQGJhKSwQFzQfUDDsUnK/view?usp=sharing> (Accessed 2021. 8. 1)
- 2) Google LLC: Colaboratory とは, <https://colab.research.google.com/notebooks/intro.ipynb?hl=ja> (Accessed 2021. 8. 1)
- 3) Google LLC: Colaboratory, <https://colab.research.google.com/notebooks/> (Accessed 2021. 8. 1)
- 4) 千葉大学環境リモートセンシング研究センター：「ひまわり 8/9 号 フルディスク (FD) gridded data (緯度経度直交座標系精密幾何補正済データ) Version 02 (V20190123) 公開について」, http://www.cr.chiba-u.jp/databases/GEO/H8_9/FD/index_jp.html (Accessed 2021. 8. 1)
- 5) 千葉大学環境リモートセンシング研究センター：ひまわり 8 号物理量変換サンプルプログラム, ftp://hmwr829gr.cr.chiba-u.ac.jp/gridded/FD/support/count2tbb_v102.tgz (Accessed 2021. 8. 1)
- 6) 千葉大学環境リモートセンシング研究センター：「ひまわり 8/9 号」全球スキャンモード (FD) gridded data 公開サーバ, <ftp://hmwr829gr.cr.chiba-u.ac.jp/gridded/FD/V20190123/> (Accessed 2021. 8. 1)
- 7) University of Hawai'i:「GMT official site」, <https://www.generic-mapping-tools.org> (Accessed 2021. 8. 1)
- 8) The PyGMT Developers:「PyGMT ユーザガイド」, <https://www.pygmt.org/latest> (Accessed 2021. 8. 1)

〔著者紹介〕

●豊嶋 紘一 (トヨシマ コウイチ)



千葉大学環境リモートセンシング研究センター特任研究員。2015 年名古屋大学環境学研究科修了, 博士 (理学)。同年より現職。専門は衛星気候学。現職では同センターの衛星データアーカイブ維持とひまわり 8 号データ可視化に尽力。日本地球惑星科学連合, 日本気象学会の正会員。
E-mail: toyoshima@chiba-u.jp