
气象数据格式笔记

发布 *v0.0.1*

2020 年 03 月 08 日

1 内容索引	3
1.1 GRIB	3
1.1.1 数据格式简介	3
1.1.2 处理工具及方法	3
1.1.3 示例解析	7
1.2 NetCDF	7
1.2.1 数据格式简介	7
1.2.2 处理工具及方法	7
1.2.3 示例解析	8
1.3 HDF	8
1.3.1 数据格式简介	8
1.3.2 处理工具及方法	8
1.3.3 示例解析	8
1.4 GeoTIFF	8
1.5 Shapefile	8
1.6 GeoJSON	8

记录各类气象数据格式的特性及其使用方法

1.1 GRIB

1.1.1 数据格式简介

GRIB(General Regularly distributed Information in Binary form), 是由世界气象组织 (WMO) 设计和维护的一种用于存储和传输网格数据的标准数据格式, 它是一种自描述的二进制压缩格式, 通常具有扩展名.grib, .grb 或.gb。

世界气象组织一共发布了 3 各版本的 GRIB 标准:

- GRIB 版本 0: 已淘汰, 无技术支持, 目前几乎不再使用。
- GRIB 版本 1: 版本 1 是 GRIB 的历史遗留版本, 已停止开发。由于它已在国际民航组织 (ICAO) 的世界范围预报系统中使用, 因此仍得到 WMO 的认可。
- GRIB 版本 2: 版本 2 格式是 GRIB 标准的扩展和强化, 它与版本 1 相比在压缩比等性能上有更优异的表现。一些国家的数值天气预报机构 (尤其是美国和欧洲) 正在逐步采用此版本, 版本 2 不能与版本 1 兼容。

想了解更多 GRIB1 和 GRIB2 的信息, 请参考: [Introduction to GRIB Edition1 and GRIB Edition 2](#)

GRIB 数据格式是以一个被称为“报文” (Message) 的数据结构为基本单元的集合体。每个“报文”中会存储一套经纬度、变量数组以及所有描述性的属性信息, 而每个 GRIB 文件里会按顺序排列存储多个“报文”。

1.1.2 处理工具及方法

ecCodes

ecCodes 是一个由 ECMWF 开发的程序包, 它可以提供用于解码和编码 GRIB 格式的 API 和工具。我们可以使用 conda 来安装: `$ conda install -c conda-forge eccodes`

ecCodes 提供了一套处理 grib 数据的命令行工具, 你可以使用 `grib_dump`, `grib_ls` 和 `grib_get` 来查看文件内容, 也可以使用 `grib_set` 和 `grib_filter` 去修改内容, 还可以用 `grib_copy` 去把部分内容复制出来, 或者使用 `grib_get_data` 从文件中把经纬度和变量值提取出来。`grib_compare` 还可以按照键去对不同 GRIB 文件进行对比。

grib_ls

`grib_ls` 命令行主要用于查看 GRIB 文件的内容信息

1. 查看 GRIB 文件所有报文的所有参数

```
$ grib_ls ERA5_20191231.grib
ERA5_20191231.grib
edition      centre      typeOfLevel level      dataDate    stepRange    dataType
↔ shortName  packingType gridType
1            ecmf        surface     0          20191231    0            an
↔ 10u        grid_simple regular_ll
1            ecmf        surface     0          20191231    0            an
↔ 10v        grid_simple regular_ll
1            ecmf        surface     0          20191231    0            an
↔ 2d         grid_simple regular_ll
...
384 of 384 messages in ERA5_20191231.grib

384 of 384 total messages in 1 files
```

2. 仅查看报文信息中的 shortName 和 dataType 参数

```
$ grib_ls -p shortName,dataType ERA5_20191231.grib
ERA5_20191231.grib
shortName    dataType
10u          an
10v          an
2d           an
...
384 of 384 messages in ERA5_20191231.grib

384 of 384 total messages in 1 files
```

3. 筛选参数 shortName 为 tp 的报文信息

```

$ grib_ls -w shortName=tp ERA5_20191231.grib
ERA5_20191231.grib
edition      centre      typeOfLevel level      dataDate    stepRange    dataType
↪ shortName  packingType gridType
1            ecmf        surface     0          20191230    5-6          fc
↪ tp        grid_simple regular_ll
1            ecmf        surface     0          20191230    6-7          fc
↪ tp        grid_simple regular_ll
1            ecmf        surface     0          20191230    7-8          fc
↪ tp        grid_simple regular_ll
...
384 of 384 messages in ERA5_20191231.grib

384 of 384 total messages in 1 files

```

4. 查看距离 (25°N,100°E) 最近点的 paramId,name 和值

```

$ grib_ls -l 25,100,1 -p paramId,name ERA5_20191231.grib
ERA5_20191231.grib
paramId      name          value
165          10 metre U wind component -0.229126
166          10 metre V wind component -0.69986
168          2 metre dewpoint temperature 272.847
...
384 of 384 total messages in 1 files
Input Point: latitude=25.00 longitude=100.00
Grid Point chosen #2 index=39460 latitude=25.00 longitude=100.00 distance=0.00 (Km)
Other grid Points
- 1 - index=39461 latitude=25.00 longitude=100.25 distance=25.18 (Km)
- 2 - index=39460 latitude=25.00 longitude=100.00 distance=0.00 (Km)
- 3 - index=39180 latitude=25.25 longitude=100.25 distance=37.48 (Km)
- 4 - index=39179 latitude=25.25 longitude=100.00 distance=27.78 (Km)

```

更多详细的参数说明及使用方法可以执行 `grib_ls -h` 查看帮助文档或阅读 ECMWF 官方文档: [GRIB tools](#)

CDO

```
conda install -c conda-forge cdo
```

Python

cfgrib & xarray

```
conda install -c conda-forge eccodes cfgrib xarray
```

pygrib

安装: `conda install -c conda-forge pygrib`

使用方法

```
In [1]: import pygrib

In [2]: msgs = pygrib.open('./20180930-fc-sfc.grib')

In [3]: msg1 = msgs[1]

In [4]: msg1
Out[4]: 1:Surface pressure:Pa (instant):regular_ll:surface:level 0:fcst time 0 hrs:from_
↪201809301200

In [5]: msg1.keys()    # 查看所有关键字, 此处列出的关键字都可以作为 select 的过滤条件
Out[5]:
['globalDomain',
 'GRIBEditionNumber',
 'eps',
 ...
 'section5Length',
 'analDate',
 'validDate']

In [6]: msg1.shortName # 查看 shortName
Out[6]: 'sp'

In [10]: msg1.name     # 查看 name
Out[10]: 'Surface pressure'

In [17]: lats, lons = msg1.latlons() # 提取经纬坐标

In [21]: array = msg1.values # 提取变量值
```

(下页继续)

(续上页)

```
In [23]: tps = msgsg.select(shortName='tp') # 选出所有 shortName 为 tp 的报文
```

警告: 在 pygrib 中, 打开的 grib 文件无法自动回收内存, 因此在使用完毕 grib 数据以后, 一定要执行 `.close()` 操作, 否则如果打开的 grib 文件太多, 会造成内存溢出。

iris

```
conda install -c conda-forge iris
```

pynio

```
conda install -c conda-forge pynio
```

NCL

```
conda install -c conda-forge ncl
```

1.1.3 示例解析

1.2 NetCDF

1.2.1 数据格式简介

NetCDF3

NetCDF4

1.2.2 处理工具及方法

netCDF4

CDO

Python

NCL

1.2.3 示例解析

1.3 HDF

1.3.1 数据格式简介

HDF4

HDF5

1.3.2 处理工具及方法

GDAL

CDO

Python

NCL

1.3.3 示例解析

1.4 GeoTIFF

1.5 Shapefile

1.6 GeoJSON