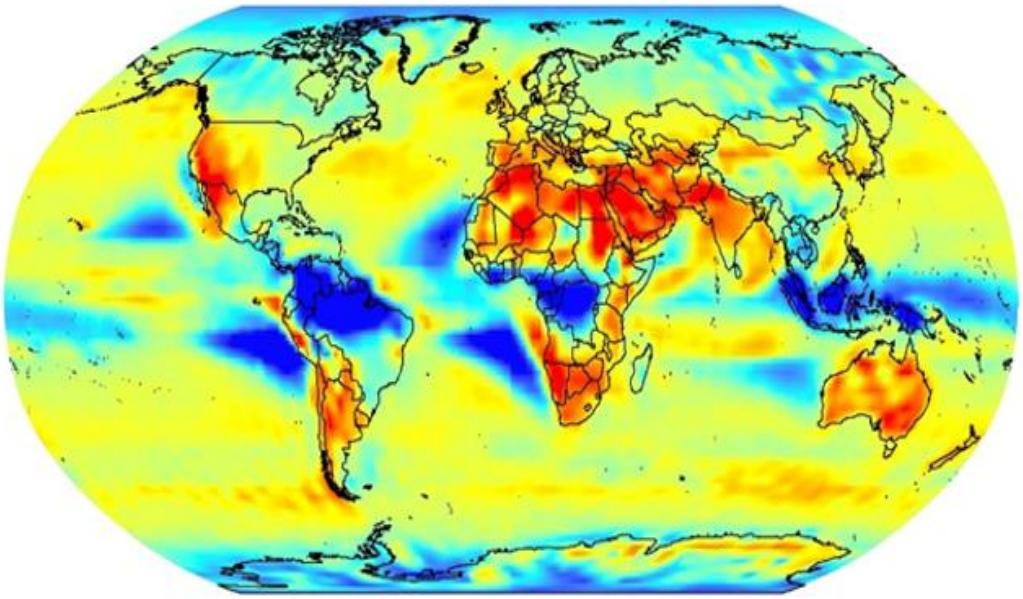
Creating Weather Frequency Maps in ArcMap: A Quick Guide for ERA Data Extraction Toolbox

— Mozhou Gao

Overview

The tool was customized by using the several side packages of python, which are arcpy, pandas, numpy, netCDF4, and ecmwfapi. The scripts are interfacing with arcpy Python GUI in ArcMap. The objective of creating this tool is help people to understand and map NetCDF data efficiently.

There are many weather datasets that use NetCDF to store data. The recommended one is ECMWF Forecasts datasets since it is free for everyone, and it has high resolution both spatially and temporally, which perfectly fit for the objective of this tool. More detail about ECMWF can be found on (https://www.ecmwf.int/). This User Manual will guide you step by step through all necessary sections and learn how to use this toolbox.



The whole toolbox was coded based on the Python library netCDF4. However, there is a bug for ArcMap when applying the function inside this library. Sometimes, the ArcMap cannot read the file from data (Figure 14). The first solution for this bug is to right-click the script on Catalog and uncheck the 'Run Python script in Process' in the source option (Figure 15). Otherwise, you have to close the ArcMap and open it again.

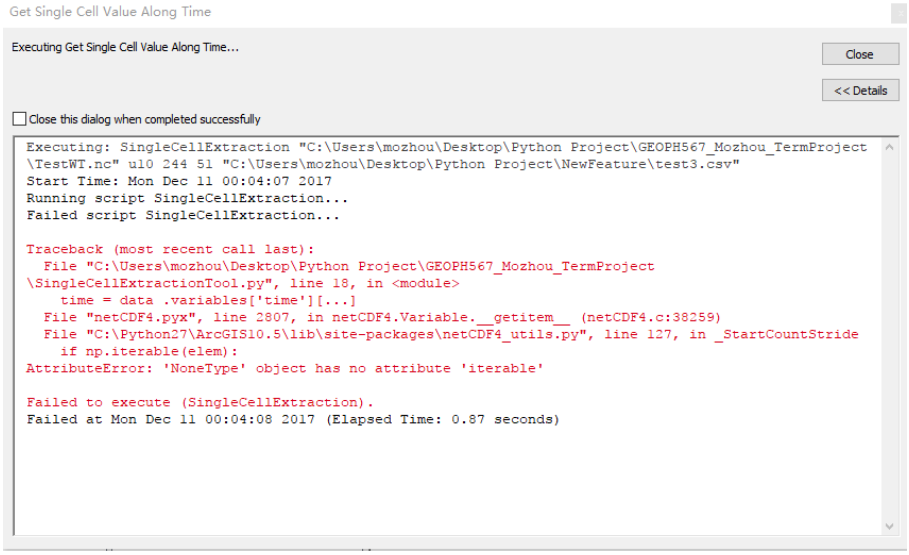


Figure 1. Error Message for bug

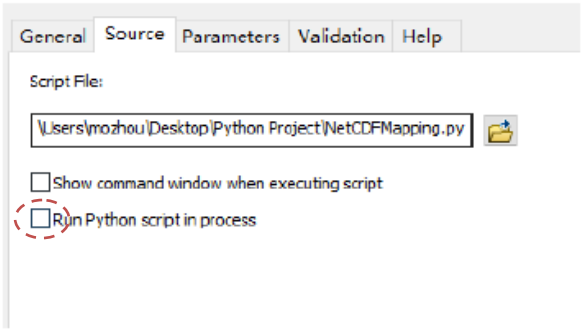
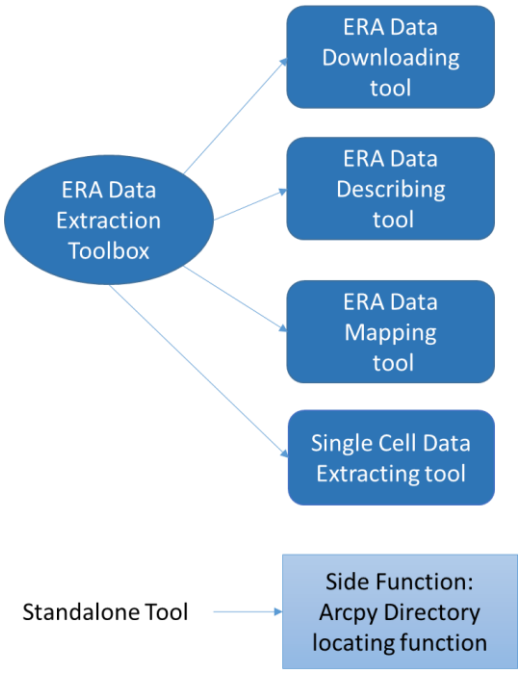


Figure 2. Bug Solution

Table of Contents

1.1. Toolbox Structure



There are four tools and a standalone python script included in this toolbox.

1. ERA Data Downloading tool
2. NetCDF Data Describing tool
3. Weather parameters mapping tool
4. Single Cell Data Extracting tool
5. A standalone python script that can locate the directory/Python Path of your arcpy in your computer (windows system).
   The reason to add this tool in the toolbox is we used a third-party python side package (ArcMap excluded library) in this toolbox. Knowing your arcpy directory can help you to set up python environment efficiently.

1.2. NetCDF Data Structure and netCDF4 module

NetCDF (Network Common Data Form) is one of the popular data formats for people with multidimensional and array-oriented scientific data. Many programming languages and programming software have the packages or functions to support this data format. The filename extension of the NetCDF is '.cn'. More detail about NetCDF can be found on https://en.wikipedia.org/wiki/NetCDF. Each NetCDF data can store more than one variable. The basic hierarchy of the data expressed as follow:

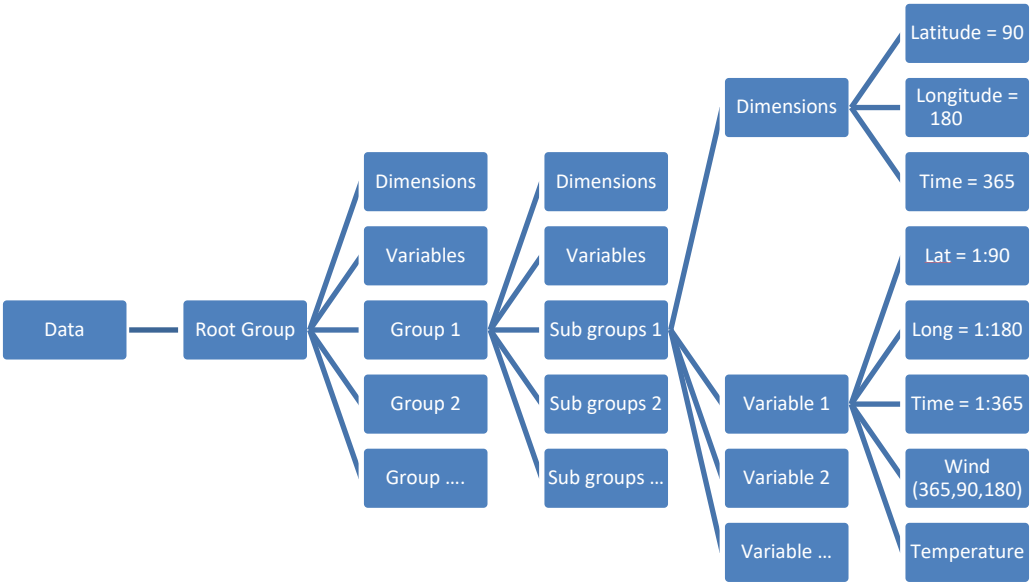


Figure 3. The NetCDF Data Structure

The highest hierarchy is Root Group, and the lowest hierarchy is variables. The variables can be stored in root group (highest), group (second highest), and sub groups (third highest). The dimension variables can create within all of them. The dimensions of the weather variables have to match the length of dimension variables (latitude, longitude and time). In Figure 3, we have a variable Wind, and the dimension of Wind is (365, 90,180), which matches the length of the time (365), latitude (90) and longitude (180).

```
<type 'netCDF4.Variable'>
int16 u10(time, latitude, longitude)
    scale_factor: 0.000236663537786
    add_offset: 2.23379604354
    _FillValue: -32767
    missing_value: -32767
    units: m s**-1
    long_name: 10 metre U wind component
unlimited dimensions: time
current shape = (31, 89, 81)
filling off
```

Figure 4. NetCDF variable object

Each variable has several attributes such as variable units, the full name of variables, variable shape, etc., which shows in Figure 4. All the variables and groups follow the OOP (object-oriented programming). Consequently, we can use .shape(), .units(), and .long_name() to check attributes.

Side package netCDF4 can easily read, append and write the NetCDF data. More detail can be found on (http://unidata.github.io/netcdf4-python/)

2.1. ECMWF API

The ERA data downloading tool was built by using the ECMWF API. Before using the tool, we have to download ECMWF API and move the package to your ArcMap Python Path. ECMWF also has a step-by-step guide, which can lead you to understand how to use this API on Python.
(https://software.ecmwf.int/wiki/display/WEBAPI/How+to+retrieve+ECMWF+Public+Datasets)

If you don't want to read the official guide, please follow my brief ECMWF-Web API setup guide:

- Sign in an account at ECMWF (https://apps.ecmwf.int/registration/) and log in your account (https://apps.ecmwf.int/auth/login/)
- Retrieve your key from ECMWF (https://api.ecmwf.int/v1/key/)
- Copy your key and paste it in a new .txt file

  Your key should look like this:

  ```
  {
      "url"      : "https://api.ecmwf.int/v1",
      "key"      : "XXXXXXXXXXXXXXXXXXXXXXX",
      "email" : "mozhou.gao@example.ca"
  }
  ```

  Open the .txt file with NotePad++, and save as a new file with name .ecmwfapirc in your computer home directory (e.g. my home directory is C:\users\mozhou.gao).

After finishing key setup, we can start to install ECMWFAPI python library. The best way to do it is to manually download the library. By doing that you can copy and paste this link in your web browser:
https://software.ecmwf.int/wiki/download/attachments/56664858/ecmwf-api-client-python.tgz

After downloading this library, you can unzip everything in your ArcMap Python Path. If you don't know your ArcMap Python Path, you can run the standalone script 'Find Your Arcpy Directory.py' on any python IDE. This script returns your arcpy package path. The arcpy library should locate in your ArcMap Python Path.

Here shows the path of arcpy in my computer.

```
C:\Program Files (x86)\ArcGIS\Desktop10.5\ArcPy\arcpy
```

Consequently, my ArcMap Python Path is C:\Program Files(x86)\ArcGIS\Desktop10.5\ArcPy and I need to move ECMWF API to this directory.

2.2. Download ERA Data

Unzip the toolbox to any folder, and connect the folder in ArcMap catalog.

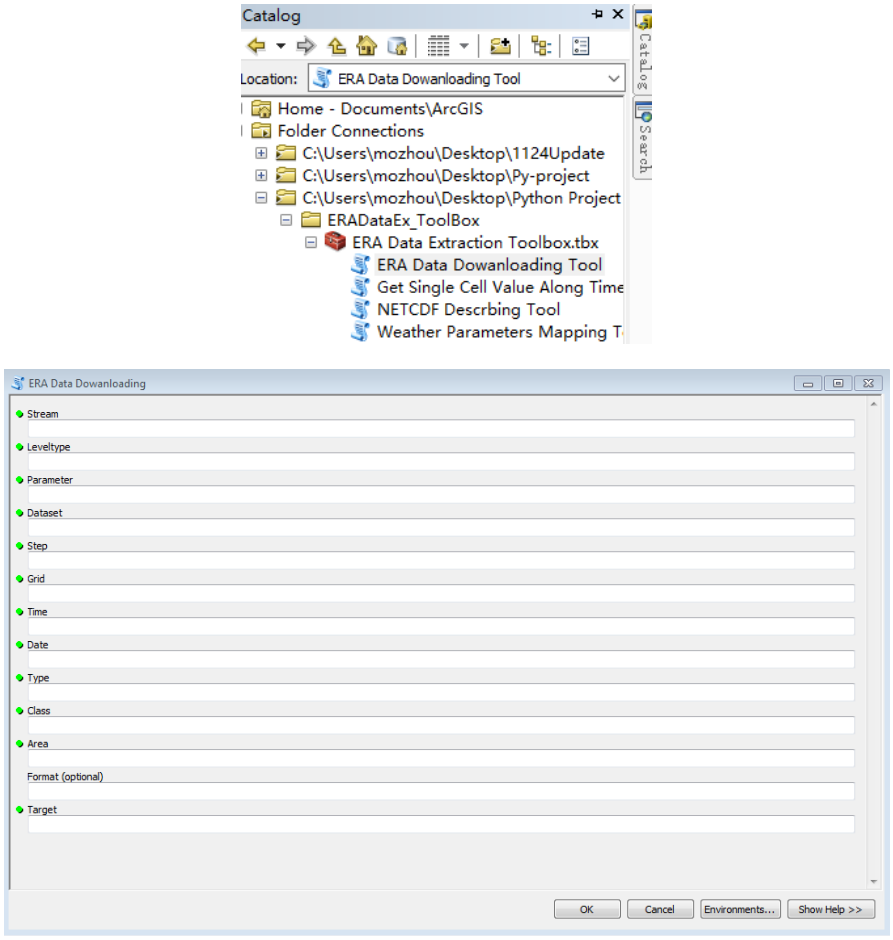


Figure 5. User Interface of ERA Data Downloading Tool

Before using ERA Downloading Tool, you must know the meaning of the input keys.
Stream: It is labels the version of the operational forecast system for seasonal forecast related products. Common types of Stream include 'oper','gfas'. More detail can be found on http://apps.ecmwf.int/codes/grib/format/mars/stream/

Leveltype: Denotes type of level. Common values are: model level (ml), pressure level (pl), and surface (sfc). More detail can be found on https://software.ecmwf.int/wiki/display/UDOC/Identification+keywords#Identificationkeywords-levtype

Parameter: Indicates the meteorological parameter that you want download such as 10 meters wind vectors, 2 meters temperature, high cloud cover, and snow density etc. Each parameter can be represented by using the individual ID. The parameter list can be found on ECMWF parameter database (http://apps.ecmwf.int/codes/grib/param-db/).

Datasets: Datasets indicates the dataset's name. The 6 most popular datasets are ERA5, ERA-Interim, ERA-Interim/Land, ERA-20C, ERA-20CM, and CERA-20C

Step: Specifies the forecast time step from forecast base time. The common steps include 0, 3, 6, 9, and 12.

Grid: Grid indicates the spatial resolution or cell size of the downloaded data. It can either be Gaussian grid or a latitude/longitude grid. The detail can be found in https://software.ecmwf.int/wiki/display/UDOC/Post-processing+keywords#Post-processingkeywords-grid

Time: Specifies the time of the data in hours and minutes. For analysis time, forecast base time and first guess verification time (we can input 00, 06, 12, and 18). You can either input one time per day or you can input multiple times.
https://software.ecmwf.int/wiki/display/UDOC/Date+and+time+keywords#Dateandtimekeywords-time

Date: Specifies the Analysis date, the Forecast base date or Observations date. The format can be found in
https://software.ecmwf.int/wiki/display/UDOC/Date+and+time+keywords#Dateandtimekeywords-time

Type: Type indicates the type of fields to be retrieved. More detail can be found in
http://apps.ecmwf.int/codes/grib/format/mars/type/

Class: Specifies the ECMWF classification given to the data. List of class type can be found on http://apps.ecmwf.int/codes/grib/format/mars/class/

Area: Specifies the desired sub-area of data to be extracted. ECMWF has an Area Examples that can guide you to select the study area you want.
https://software.ecmwf.int/wiki/display/UDOC/Post-processing+keywords#Post-processingkeywords-area

Format: This is an optional input, so if you want to NetCDF data just use keyword 'netcdf'

Target: Specifies a Unix file into which data is to be written after retrieval or manipulation. In another word, it is the data output path. The path names should always be enclosed in double quotes:
e.g. " C:\Users\mozhou.gao\Desktop\Python Project "

If you want to know more details about the ECMWF API keywords you can visit https://software.ecmwf.int/wiki/display/UDOC/MARS+keywords

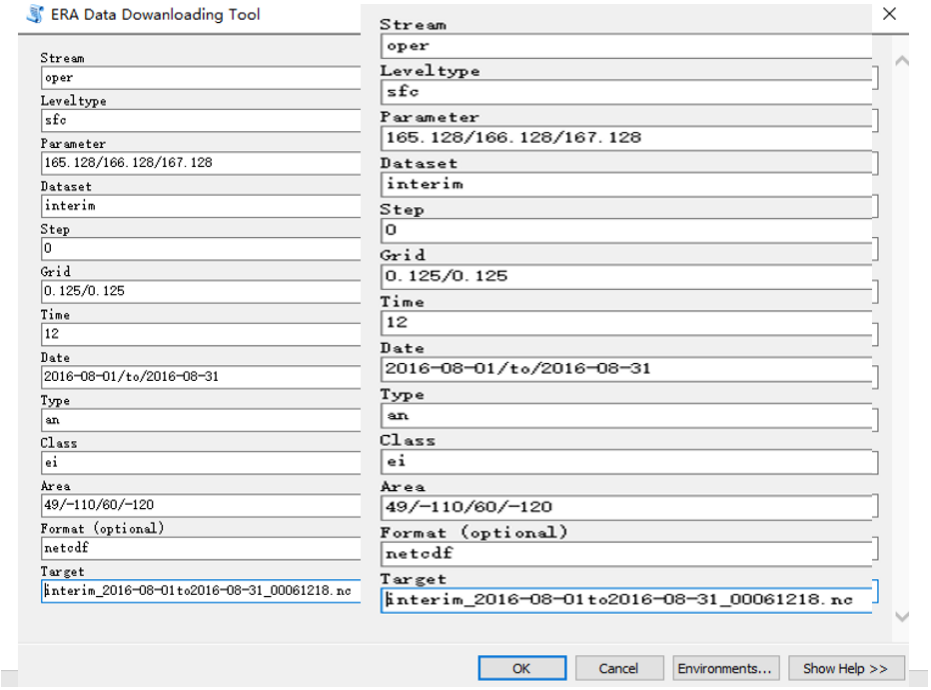


Figure 6. Sample inputs for downloading a daily average of temperature, u/v wind components data in Alberta on August 2016 with spatial resolution 0.125 x 0.125 from ERA-Interim dataset.

The reason to download precipitation separately is that precipitation data has completely different leveltype, type, and time steps. Total Precipitation data is a forecast data. For matching the wind and temperature data temporally, the time should be 0, and the step should be 12, which means I am trying to access 12 pm's based on 0 am's measurement.

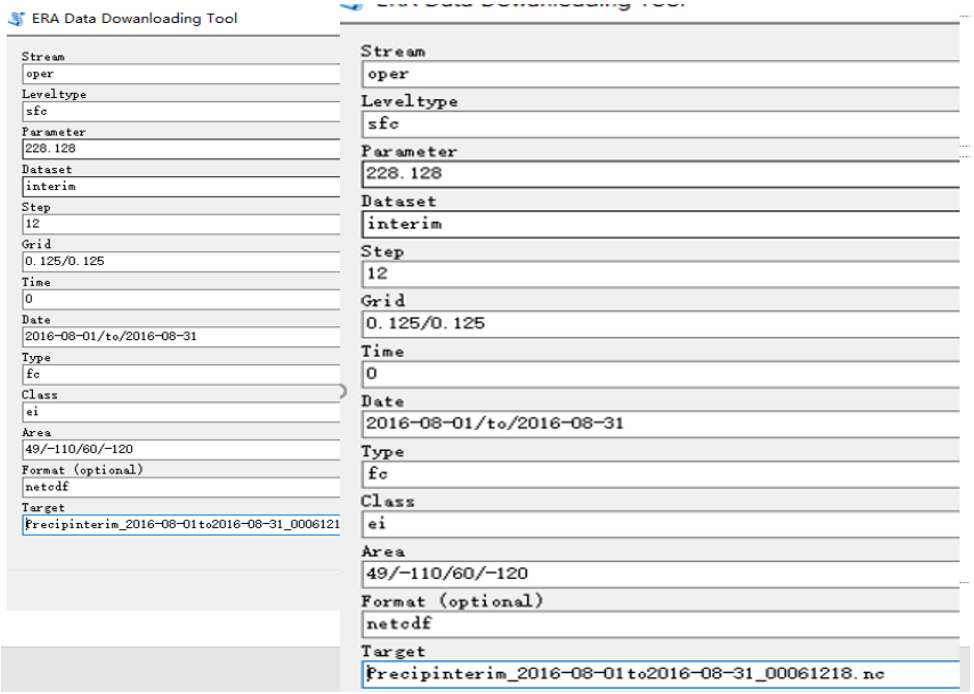


Figure 7. Sample inputs for downloading total precipitation data

3. NetCDF Data Describing Tool

The NetCDF data Describing Tool can help user to understand data structure and the attributes of variables in data file.

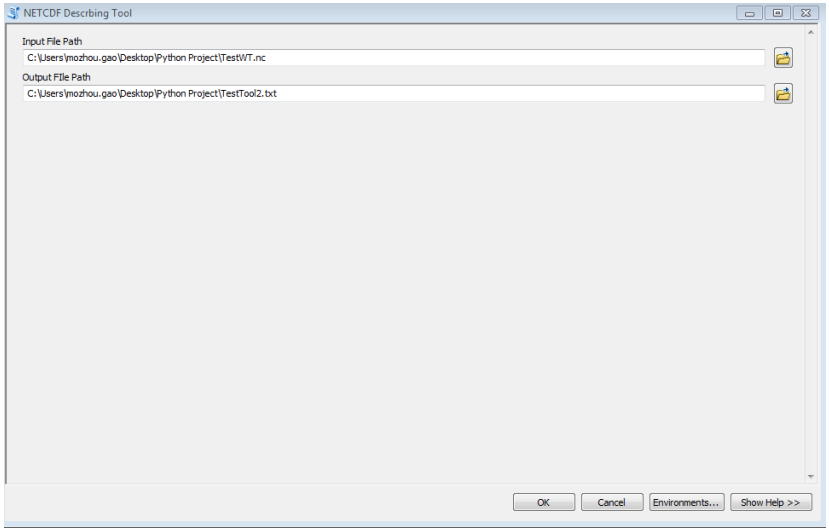


Figure 8. User Interface of NetCDF Data describing tool

This tool only requires the input data and the output text file path. Moreover, it will return data table to the arcpy message window. In Figure 8, the inputted data is sample Wind and Temperature data (TestWT.nc), which can be found in an unzipped file.

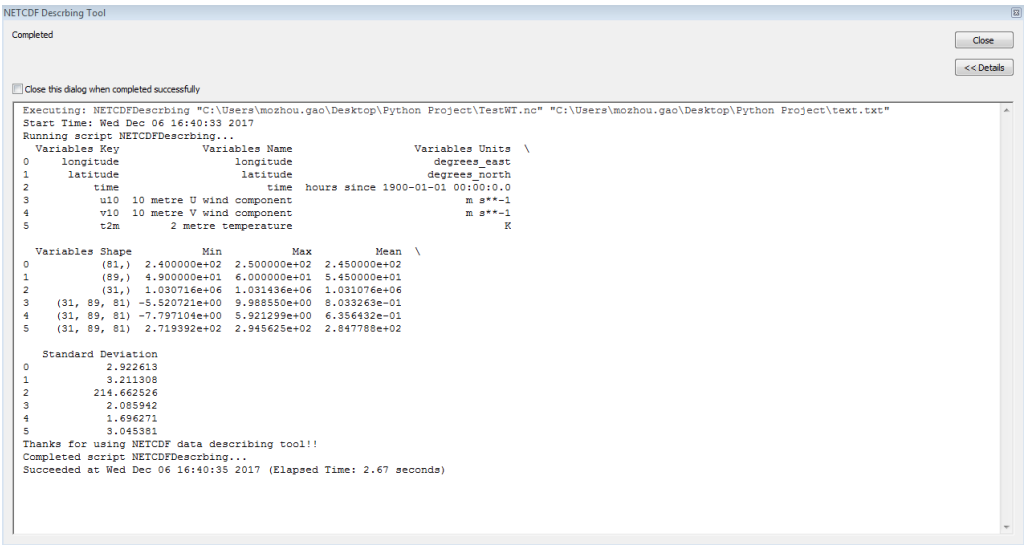


Figure 9. The Output arcpy message

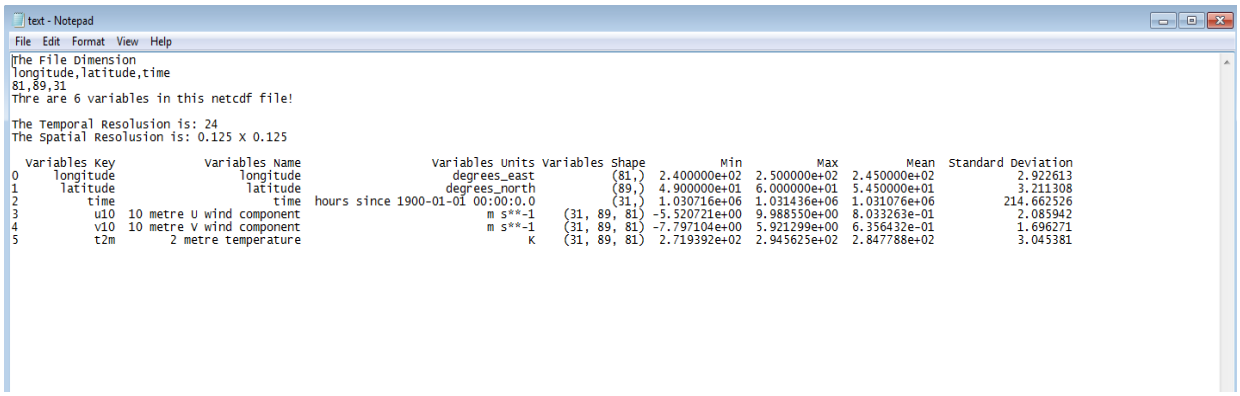


Figure 10. The preview of output text file

The text file includes the value of dimension, the number of variables, temporal resolution, spatial resolution, variables key, variables full-name, variables units, variables shape, minimum value of each variable, maximum value of each variable, mean of each variable and standard deviation of each variable.

4. Weather Range Mapping Tool

4.1. Logic Flow Chart

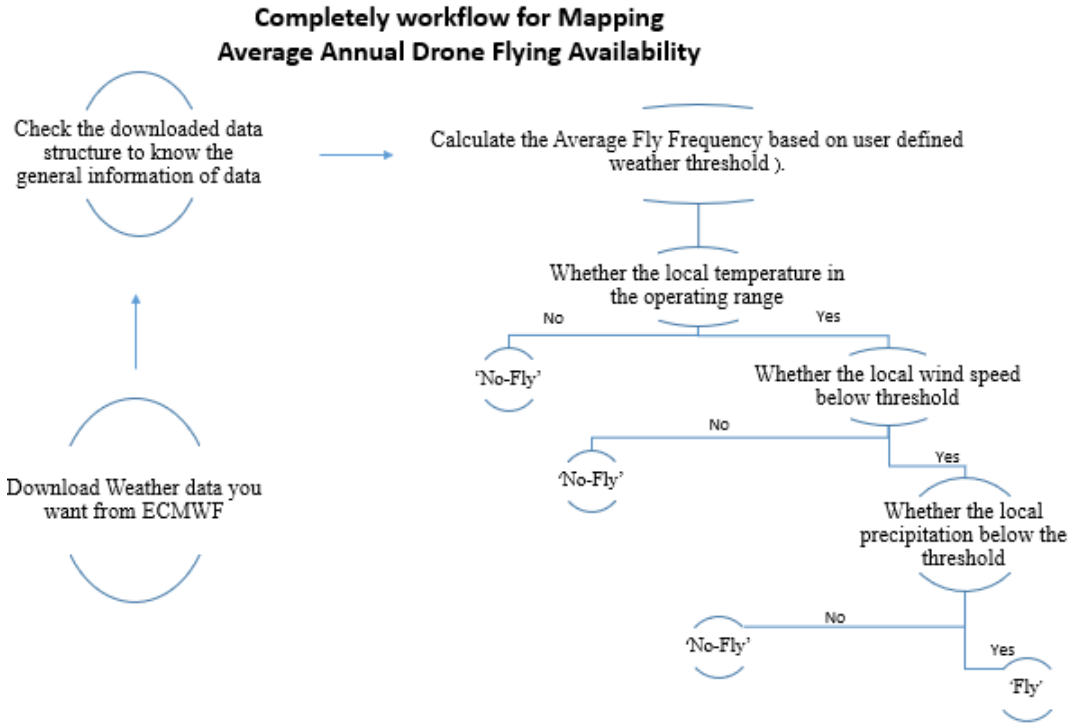


Figure 11. The general working follow for mapping the Average Drone Flying Availability

4.2. Create the Frequency Raster

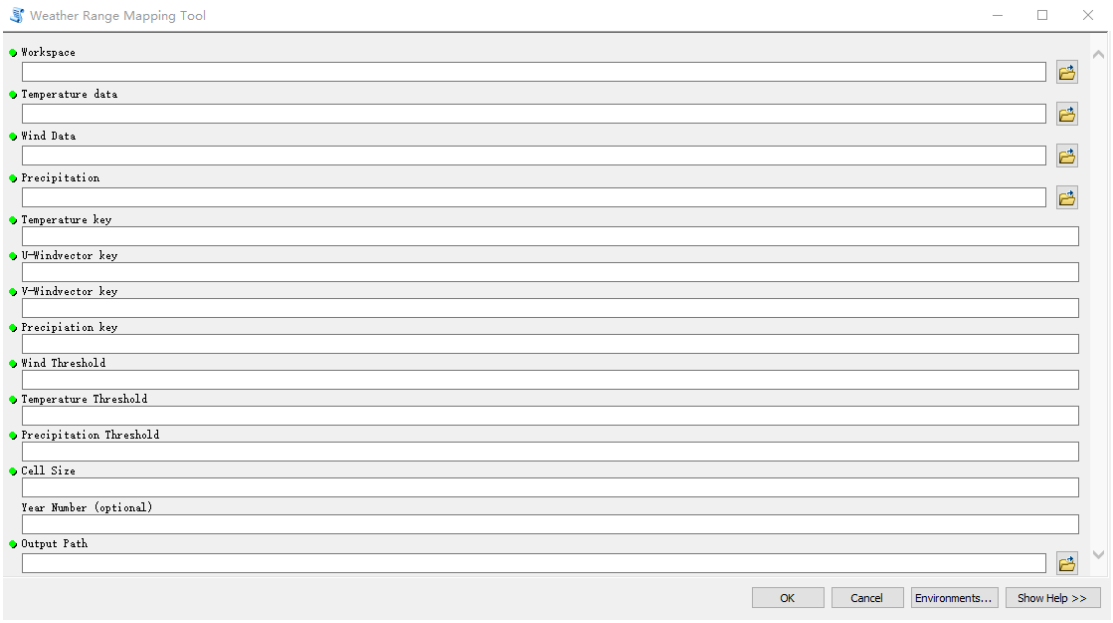


Figure 12. The User Interface of Weather Mapping Tool

There are 14 input parameters required for this tool. Before using this tool, you should already know the variable keys in your NetCDF file.

Workspace: Set the workspace path for this tool.

Temperature data: Select your temperature data (It is better to move your data files to your workspace path). In this example, my temperature data is TestWT.nc

Wind Data: Select your wind data. In this example, my wind data is TestWT.nc

Precipitation data: Select your precipitation data. In this example, my temperature data is TestPrecip.nc

Your wind data, temperature data, and precipitation data should have the same dimensions.

Temperature key: temperature variable key (If you don't know your variable keys, you can find it in the output text file of describing tool). In this example, I used 2-meter temperature. So, my temperature key is 't2m'.

U component wind vector key: I used 10 meter wind data, so the key should be 'u10'

V component wind vector key: Similarly, the key should be 'v10'

Precipitation key: My sample data is total precipitation data, so the key should be 'tp'

Wind threshold: Input the wind threshold. The format should be numerical value 'upper limit, lower limit'. (e.g. 10,0)

Temperature threshold: Input the temperature threshold.

Precipitation threshold: Input the precipitation threshold (for no rain, you can input: -1,0).

Cell size: the output raster file's cell size, it should be similar to your input data's cell size.

Year Number: If you want to calculate the annual average you can just type the number of years. If you are only interested in the total fly frequency, you can just leave it to a blank space (e.g., If you want to calculate 15 years average, and your data also includes 15 years of time series. You can input 15 for Year Number.

Output Path: enter the output path of your frequency raster.

The sample data is weather data of Alberta in August 2016, which includes wind data, temperature data, and precipitation data. Consequently, I will set wind, temperature, and precipitation based on DJI Phantom 4's operating temperature range (40,0), wind speed resistance 10m/s (10,0), and no rain (0,-1) to create our sample raster.

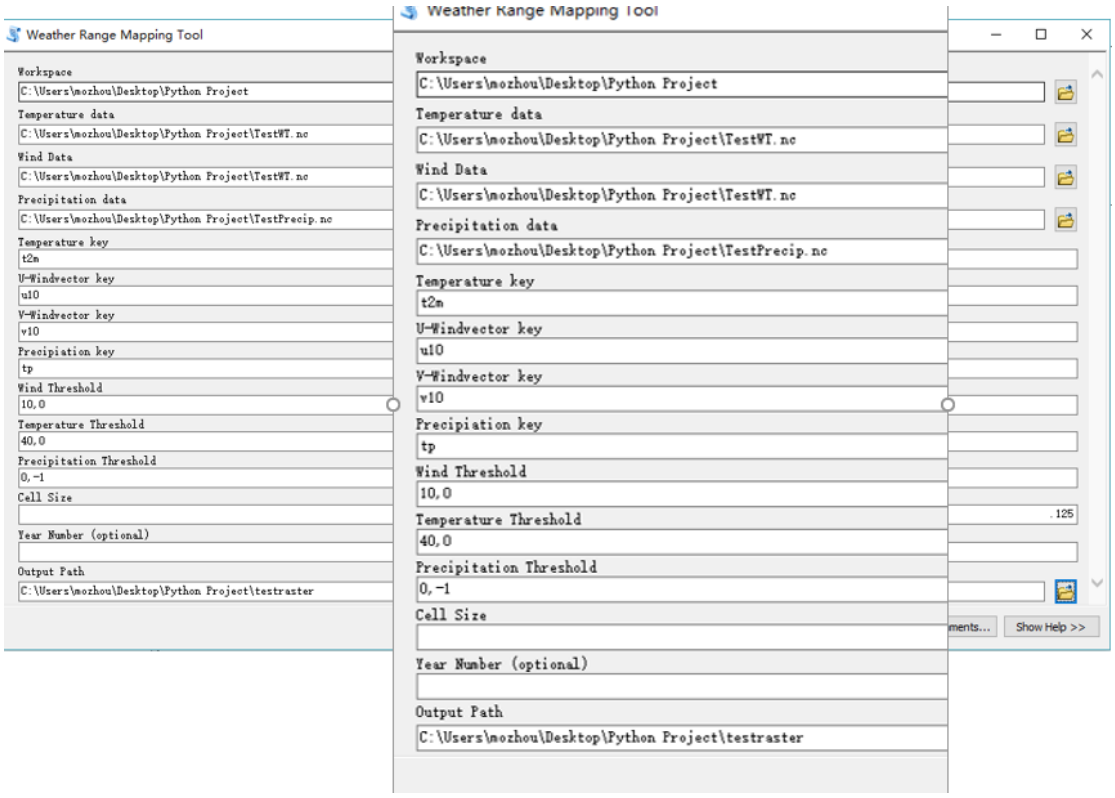
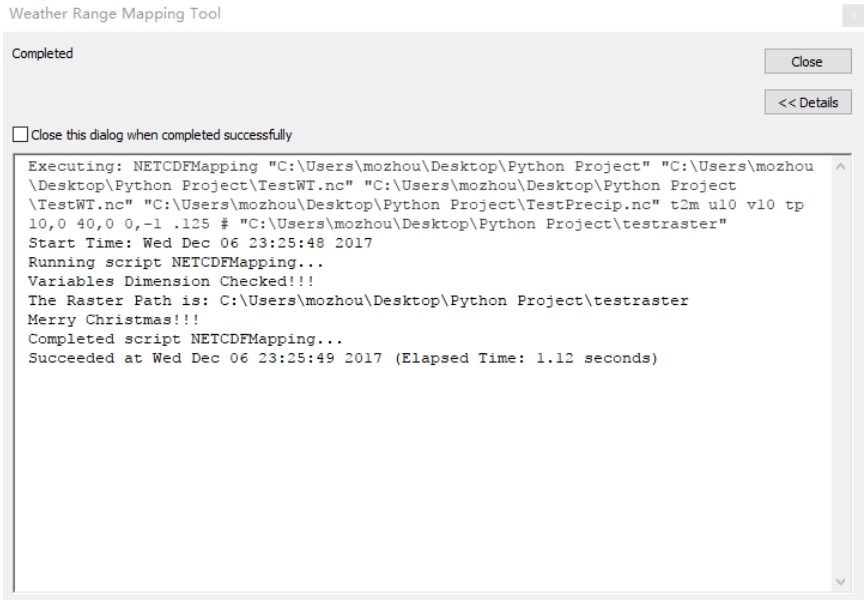


Figure 13. Sample of Inputs



The arcpy message window will print ‘Merry Christmas!!!!’ if the script rans successfully.

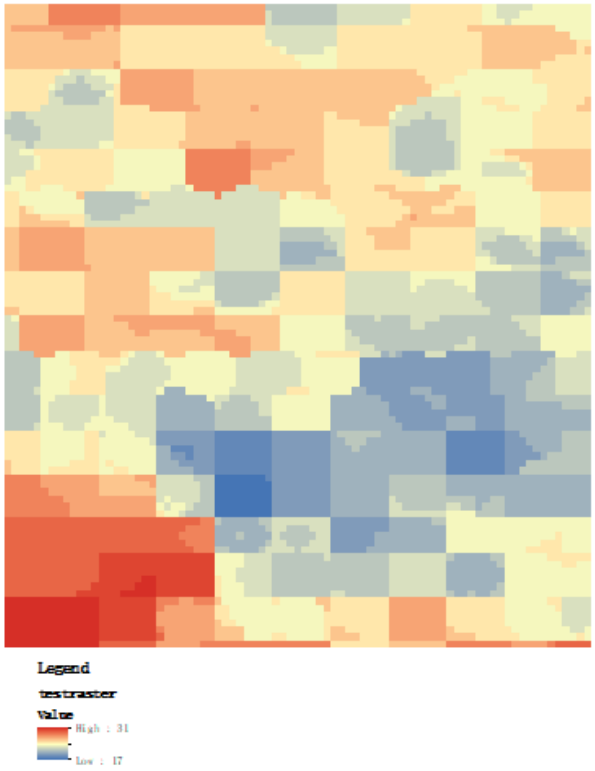


Figure 14 shows the number of days that DJI phantom can fly in August 2016 across the Alberta.

I organized some popular products' weather SPECs, if you want, you can plug in these parameters and play with it.

Manufacturer	Products	Temperature	WindSpeed Resis	Precipitation	IPrating	
SensFly	ebee	(-15,51)	12m/s	0		
	albris	(-10,40)	Automatic Flight: 8m/s Manual Flight: 12m/s			
DJI	matrice 200	(-20,45)		0	IP43	
	Phantom 4 Spec	(0,40)	10m/s	0		
	Phantom 4 Pro	(0,40)	10m/s	0		
	Inspire 1	(0,40)	10m/s	0		
	Inspire 2	(-20, 40)	10m/s	0		
Kespry	drone 2.0	NaN	NaN	NaN		
	DJI Phantom 4 pro	(0,40)	10m/s	0		
Precision Hawk	matric 100	(-10,40)	10 m/s	0		
	Lancaster 5	(?,40)		0		
Yuneec	Breeze	(0,40)		0		
	Typhoon H	(-10, 40)	5.36m/s 12MPH	0		
	Typhoon Q500 4k	(-5,50)	5.36m/s	0		
	H920 plus	(-10,50)	10m/s	0		
	H520 coming soon					
C-astral	Bramor ppx	(-25,45)	60km/h 17m/s	Rainproof		
	Bramor C4EYE	(-25,45)	15m/s			(Night Observation)
	Atlas	(-25,45)	60km/h	Rainproof		
	Bramor RTK	(-25,45)	60km/h	Rainproof		
	Bramor Geo					1.5 hour
3D robotics	pixhawk 2	(0,45)	11m/s	0-85% RH		
Dragonfly	Guardian		50km/h			
	X4-P	-25° to 38°C	50km/h vary based on payload			
	Comandar					
	Tango2 coming soon					
Aeryon labs	Skyranger	(-30,50)	25m/s		IP53	

5. Single Cell Data Extracting tool

This tool can extract the value of specific cell along the time series based on the user-selected longitude and latitude. Before using this tool, you should already know the variable keys, thecell size of the NetCDF dataset. The inputted longitude should be given from 0 to 360.

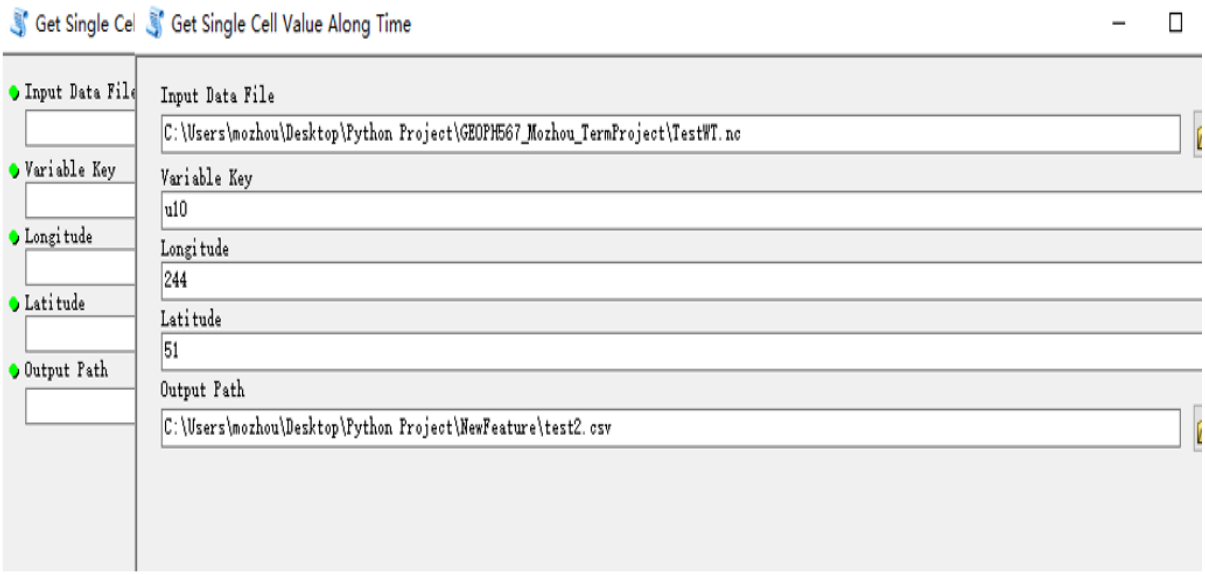


Figure 15. The User Interface of tool and the sample inputs.

The Tool will output the csv file, which contains 3 columns (Index, Time, Value).

	A	B	C	D
1		Time	u10 Value	
2	0	0	-2.24151	
3	1	1	-1.57128	
4	2	2	-0.08361	
5	3	3	-0.97465	
6	4	4	-0.96542	
7	5	5	-0.53114	
8	6	6	-2.07135	
9	7	7	-1.77457	
10	8	8	-1.30622	
11	9	9	-0.75881	
12	10	10	-2.1362	
13	11	11	-1.11428	
14	12	12	1.664857	
15	13	13	1.190347	
16	14	14	2.304558	

Figure 16. Sample output

This is the end of this guide, and you are good to use this toolbox now! Have fun! Please contact me if you have any question, my email is mozhou.gao@ucalgary.ca.