DAYMET DOCUMENTATION

By

Siddhisanket Raskar
Vargas Lab
Plant and Soil Department
University of Delaware

The data from DAYMET is available in form of 2*2 degree tiles in netcdf(.nc, .nc4) format in daily resolution for entire north America from 1980 to 2013. In order to use this data for any kind of processing using matlab or any other tool we have to rasterize it.

The process of getting data ready for statistical analysis has **three stages. –**

- Download data from appropriate server -
- Rasterize it - 2_NetcdfToRaster.py
- Mosaic it - 3_MosaicRasters.py

The scripts need Python 2.7 and ArcGis10.1 to work. They have been tested on windows 7 and 8 and 8.1 I have run them on linux with some modification.

Here is the detailed information about how each stage works.

Stage 1 : Download Data

DAYMET has several options using which you can download data. Here is the Link - http://daymet.ornl.gov/dataaccess.html for all available options.

PART 1 :

If you decide to use FTP server. Then follow these instructions.

- Download FTP client like filezilla for your operating system.
  Filezilla - https://filezilla-project.org/
- Here are the server details to download the MODIS Data for variables – GPP, ET, NPP. Put following information in the relevant fields in filezilla and click on Quickconnect.

  Host : ftp://daac.ornl.gov/data/daymet/Daymet_mosaics/data/
  Username :
  Pasasword :
  Port : 21
- Once connected you can browse to the appropriate index of variable you want to download and simply drag it to the relevant location on your computer.

PART 2 :

If you decide to use HTTP server then use **1_DownloadTiles.py** script. This script need wget (http://gnuwin32.sourceforge.net/packages/wget.htm) installed on your system. We will use THREDDS Server to download data. The file paths on the server and the url for the server may change over time. Make necessary changes in the script or contact me for assistance.

Stage 2 : Rasterize Data

Use script **2_NetcdfToRaster.py** for this task. Here is basic description of what this script is doing –

- It reads each .nc file from the download folder. Each .nc file contains data for 365 days of year.
- It rasterizes the .nc file and creates a temporary raster file in memory. We use MakeNetCDFRasterLayer_md module of ArcGis for this purpose.
- The inner loop in the script goes through each day of the in memory raster and creates raster file which gets written to disk. We use SelectByDimension_md tool for selecting data related to each day and CopyRaster_management tool to write that raster to filesystem.

Stage 3 : Mosaic Data

Use script 3_MosaicRasters for this task. In this stage, we will be stitching together all the smaller tiled rasters together to create a huge map which can be used for the statistical analysis. Here is the brief description of what script is doing –

- Creates geodatabase and sets up other environmental parameters needed to use arcgis tools.
- Reads all the raster tiles for each day from respective folders.
- We use AddRastersToMosaicDataset_management tool of arcgis to mosaic rasters. We use SetMosaicDatasetProperties_management to set other properties and CopyRaster_management to write the mosaicked raster back to file system.

Alternate Method :

Download mosaic .NC files from the DAYMET. Link - http://daymet.ornl.gov/mosaics.html Each file available here contains 365 layers each representing a day in that particular year. For this file to be used for the statistical analysis it is necessary to separate each day and store it as raster file. For this purpose, make use of 1_mosaic_to_raster.py script. Here is what this script does –

- We will make use of MakeNetCDFRasterLayer to convert NetCDF file into 3D raster file. The output raster contains 365 layers each depicting the day in a year.
- Now, we will give this file as input to SelectByDimention function. This function isolates individual layers.

- We make use of CopyRaster_management function to store each layer isolated by above step.
- As a result of this, the output of this script will be raster files for each day of the year.

Create monthly Summaries

Use script 2_summarize.py for this task. In this stage, we will be taking mean (except PRCP) of each months files and creating monthly summary. Here is more detailed information regarding what this script is doing –

- This script groups together files for each month of the year and then takes their mean and save the resulting raster file for further use.
- We make use of CellStatistic function provided by ArcGis to take the mean or sum of the grouped files. Please refer to the ArcGis documentation for detailed information regarding how to use this function.
- In 2_summarize.py script, we will use parameter "NO DATA" in CellStatsitic function.

Use 22_summarize.py script to calcualte monthly summaries along with the statistics. To calculate statistics which gives us better idea about confidentiality of data we will use following method.

- We know that : mean = sum/count. We will calculate mean and sum of our datasets and plug them into this formula to calcuate count. The count here will tell us how many days worth data is missing in each monthly summary.
- The rasters for sum and mean are calcuated using cellstatistic function.
- The division of rastes is done using divide function of arcgis.
- We use "DATA" parameter for CellStatistic function in this script.

Clip Output for Mexico

We make use of two scripts – 3_clip_daily.py and 4_clip_monthly.py to clip the north America rasters.

3_clip_daily.py

This script is used on the daily raster files created by 1_mosaic_to_raster.py script. The output of this script are the rasters for mexico in daily resolution. We make use of clip_management function of ArcGis to specify parameters of clipping geometry and clip the rasters to the shape of mexico. Then we use CopyRaster_management function to save the output.

4_clip_monthly.py

This script is same as the 3_clip_daily.py script except this operates on the monthly resolution data rather than daily resolution. The only change is the path of the input and output files which are modified for the

Calculating of Tmean

Use script 5_tmean.py for this task. Here, we are basically taking mean of Tmin and Tmax variables and saving it as Tmean. Here is more detailed information regarding what this script is doing -

- We are reading a raster file from Tmin and a raster file from Tmax and adding them to the array which will be processed in the next phase.
- The array which contains a file from Tmin and Tmax each is feed into CellStatistics function which takes mean of these two and creates new raster.
- Then, we use Copy Rastermanagement fucntion to save the raster into the appropriate directory.

DATASET

The datasets calculated using above scipt can be found at - D:\DATASETS\DAYMET\RASTERS. The readme.txt file present in the folder contains more information regading the folder structure of the dataset.

The backup copy of the dataset can be found on the farber cluter accessible at farber.hpc.udel.edu. Path - / lustre/scratch/SPAC/raskar/siddhi