 DEVELOP Recipe Card

Batch Processing NetCDF Files in Arc 5/1/2016

Created by: Wise Disasters

# Requirements

|  |  |  |
| --- | --- | --- |
| ArcGIS 10.2 | Python Scripter |  |
|  |  |  |

# Steps

1. Download all the needed files. NC files have a .nc extension. They cannot be opened unless the NetCDF in GIS is applied to create rasters.
2. Create four different folders
   * A folder for all NetCDF files
   * A folder for the rasters created from netCDF files
   * A folder for points
   * A folder for interpolated rasters
3. Put all NC files in the same new folder. Rename these files in a way that will not confuse you or the processor. Including in the name what the file is and the dates it covers would be a good idea. Keep the name short as it can get confusing as you progress.
4. Using Python scripter, prepare to write four different programs to process each folder.
   * If you do not have the Python Scripter (pyscript) installed, you can download it from the internet.
5. Pyscript only lets you writes scripts for python programs. Therefore you cannot overload it by writing complex programs. For example, it will only let you use one for loop.
6. You will need to write 4 different programs to process each folder.
7. In every program, remember to set the window size to the same value. This will ensure that all output maps have the same resolution, what will ensure eventual calculations will be easier and more accurate. The next code line shows how to set the window size in python:

#Cell Size

Arcpy.env.cellsize = 0.002

1. Transforming NC files into rasters:
   * In Pyscript, create new file by going to File🡪New
   * Save file with a name that clearly shows what the program will do; something like CDFtoraster
   * The first thing to do in a python program is to import system modules. Depending on what functions you will use in your program, modules can be different. However the following are imperative for any program:

*# Import System Modules*

*import arcpy*

*from arcpy import env*

*from arcpy.sa import \**

* To connect pyscript with ArcGIS, you need to check spatial analyst extension license

*# Check out the ArcGIS Spatial Analyst extension license*

*arcpy.CheckOutExtension("Spatial")*

* To be able to automatically delete files with same name from previous work, you have to activate the overwrite option by the following line:

*# Overwrite*

*arcpy.env.overwriteOutput = True*

* Create a workspace. The workspace will be the folder where the program will go to get the files.

*# Set the Workspace*

*env.workspace = ("C:\Users\DEVELOP3\Desktop \NC")*

* Create output folder. This is the folder that will contain the rasters created from the NetCDF function. This should be the folder you created to contain these rasters.

*Outfolder = ("C:\Users\DEVELOP3\Desktop\All\_MERRA\Rasters")*

* Create file list. This is getting files from the workspace into the program so that they can be processed.

*#Create a Python list*

*Rainfall\_NC\_List = arcpy.ListFiles("\*.nc")*

* Loop through the list you created

*# Loop through the list*

*for raster in Rainfall\_NC\_List:*

*print "Netcdf in process for " + raster*

*NCfiles = arcpy.env.workspace + "/" + raster #Find raster in workspace*

*TempLayerFile = "Rainfall"*

*output = Outfolder + "/" + raster[:9] #Save raster with same name as cdf file, restrictname length to 9 letters*

*#arcpy.md.MakeNetCDFRasterLayer(raster, "PRECTOT","lon","lat",output)*

*# Process: Make NetCDF Raster Layer*

*arcpy.MakeNetCDFRasterLayer\_md(NCfiles, "PRECTOT","lon","lat", TempLayerFile, "", "", "BY\_VALUE")*

*# Process: Copy Raster*

*arcpy.CopyRaster\_management(TempLayerFile, output + ".tif", "", "", "", "NONE", "NONE", "")*

print "CDF to raster done!"

1. Change rasters into points. The next step is to change rasters into points.

Create new script. Follow the same instructions as above, and change the contents of the loop. The workspace will be the raster files, the output folder will be the points folder.

# Convert rasters to points

Raster\_List = arcpy.ListRasters("\*.tif")

for raster in Raster\_List:

print "Raster to point in process for " + raster

out\_ptfile = point\_folder+ '/' + raster[:9]

arcpy.RasterToPoint\_conversion(raster, out\_ptfile , "Value")

print("Raster to point batch done!")

1. Interpolation. The next step is to interpolate the points to come up with a smooth map. You can use Kriging, Spline or IDW. If you have a lot of points in your map, use IDW. If you have a few points, use Kriging or Spline. For best results, make sure you have at least 2 or 3 points outside of the study area.

* Use same steps as above except for the loop. Make sure you have correct workspace and output folder
* For kriging, the for loop will look like this:

*# Convert rasters to points*

*pt\_List = arcpy.ListFiles("\*.shp")*

*for ptfile in pt\_List:*

*print "Kriging in process for " + ptfile*

*out\_krig\_file = krig\_folder+ '/' + ptfile[:9]*

*outKrig = Kriging(ptfile, "GRID\_CODE", KrigingModelOrdinary("CIRCULAR"))*

*#Create temporary file to hold the results of kriging*

*outKrig.save(out\_krig\_file + '.tif' ) #Save temporary file with recognizable name*

*print("Krig Complete")*