Extracting and Standardizing Criteria to a specific Study Region

Created: January 27, 2015

This document provides step-by-step instructions for extracting large scale criteria and standardize them to a defined study region. It uses ArcPY geoprocessing tools. The script can either work using Python IDLE or the Python editor for Windows PythonWin. To load your script into Python IDLE, right click on the .py script and select “Edit using IDLE”. Make sure the parameters setting is properly filled (see instruction below). When the parameters have been filled and the script is ready to be launched, go to the Run menu and select Run Module command.

PythonWIN is a Windows IDE and GUI framework for Python. It has an [integrated debugger](http://docs.activestate.com/activepython/2.5/pywin32/html/pythonwin/doc/debugger/index.html) and a rich Python editing environment. PythonWin can be downloaded from <http://starship.python.net/crew/mhammond/win32/Downloads.html>. A shortcut can be created during the installation process to add “Edit using PythonWin” in the right clic menu. The same attention should be put on setting the parameters properly before launching the script.

## SETTING UP PARAMETERS

**Step 1 - Setting up project directories**

Before processing the criteria, a project directory and sub-directories should be created as follow: The input shapefile that delineate the study region first step in the preparation of the criteria that will serve in the representation analysis is to create a project directory along with sub-directories that will contain input data and the Python script. Note that the output directory and sub-directories are created by the script.

- StudyRegion

- shp

- grids

- proj

The “shp” folder contains the shapefile that delineate the study region. The“grids” folder contains the criteria of interest in TIFF format at large scale. Criteria don’t have to be clipped at the study region level, the script will take care of it. The “proj” folder contains the projection file used in all resulting output. To create such a file, you need to copy the .prj file from a shapefile that uses the proper projection, paste it into the proj directory and rename the file the way you want, as long as you set the name properly in the subsequent step.

**Step 2 – Specifying the location of the catchments, criteria and projection**

The second step is to specify the location of the script is in order to derive the location of the study region, criteria and projection file. To derive the study region and criteria locations, the script should be stored in the “StudyRegion” folder, with the “shp”, “grids” and proj sub-directories.

# Extract the working directory defined by the location of the script.

currentDir = os.getcwd()

# Location of the study region (shp)

studyregion = currentDir + "/shp/NWBLCC\_AoA\_fda\_diss.shp"

# Location of the criteria in TIFF format

LED = currentDir + "/grids/led\_250\_25.tif"

CMI = currentDir + "/grids/NORM\_6190\_CMI.asc"

GPP = currentDir + "/grids/MOD17A3\_Science\_GPP\_mean\_00\_13.tif"

NALC = currentDir + "/grids/nalc.tif"

## Set output coordinate system

outCS = currentDir + "/proj/NWB\_LCC\_proj.prj"

**Step 3 – Specifying the list of criteria and the output names.**

The standardization process is created to loop through a list of criteria to apply a set of operation using the same extent. A list that stores criteria paths need to be set prior to run de script. Output criteria names are set using a dictionary structure allowing the control of the output name format (criteria, source, years, etc). The output criteria names are found inside the quotes. For example, gpp output could be named “gpp\_0013”.

## Set the list of criteria

criteriaList = [CMI, GPP, LED, NALC]

## Set output names

criteriaOutput = {CMI:"cmi\_a2\_6190", GPP:"gpp\_0013", LED:"led", NALC:"nalc"}

**Step 4 – Specifying the resolution**

The resulting output grids can be created at a specified resolution. It can be more than one. To allow the script to loop through those desire resolutions, a list needs to be provided in string format.

## Set resolution

resList = ["250", "1000"]

**Step 5 – Specifying the resampling algorithms.**

Because the resulting output resolution is not necessarily the same as the input grids resolution, a resampling algorithm need to be used. The resampling algorithm can be:

* NEAREST —Nearest neighbor assignment
* BILINEAR —Bilinear interpolation
* CUBIC —Cubic convolution
* MAJORITY —Majority resampling

The NEAREST and MAJORITY methods are used for categorical data, such as a land-use classification. The BILINEAR and CUBIC methods are used appropriate for continuous data. The resampling algorithm is stored in a dictionary structure to make sure the proper algorithm will be used for the right criteria.

## Set resampling algorithm

resampleOutput = {CMI:"BILINEAR", GPP:"BILINEAR", LED:"BILINEAR", NALC:"MAJORITY"}

## GENERATING STANDARDIZED CRITERIA FUNCTIONS

ALL FUNCTION USED IN THIS SCRIPT ARE HARDCODED. THE PARAMETERS ARE TAKEN FROM THE PARAMETERS SETTING SECTION SPECIFIED IN STEP 1 TO 5. NOTHING SHOULD BE CHANGED IN THE FUNCTIONS WHILE RUNNING THE SCRIPT.

**Step 6 – Generating study region output**

In step 6, the genExtent function reprojects the study region and converts the output (ClipExtent) to grid format using the list of resolution specified in the parameters setting section.

clipExtent = tempOutputDir + "/studyregion.shp"

genExtent(studyregion,clipExtent,resList,FinalOutputDir)

**Step 7 – Generating standardized criteria to a respective grid cell resolution**

In step 7, the genCriteria function reprojects each criterion to the specified projection and extracts them using the projected study region. The resulting output used the cell size resolution given in the resList. All resulting output grids are aligned with the study region grid that uses the common resolution cell size. .

extractCriteria(criteriaList,resList,clipExtent,FinalOutputDir)