# HydroDataPro

Version 1.0

Nicholas Giles

CE 513

3/16/2019

http://web.engr.oregonstate.edu/~gilesn/

# Table of Contents

## Introduction

Due to the spatially explicit characteristics of hydrology, the field of hydrologic (hydrology) modeling has for many years been centered on GIS analysis (Vieux, 2001). While the application and scale of hydrologic models vary greatly based on the needs and requirements of the end user, the majority of models today require large amounts of spatial data. This manuscript presents HydroDataPro, an automated approach to preparing spatial data for hydrologic modeling using arcPy. For the time being, HydroDataPro is only available for the state of Oregon, and the exact bounds of the dataset are shown in blue in Figure 1 (coverage map).

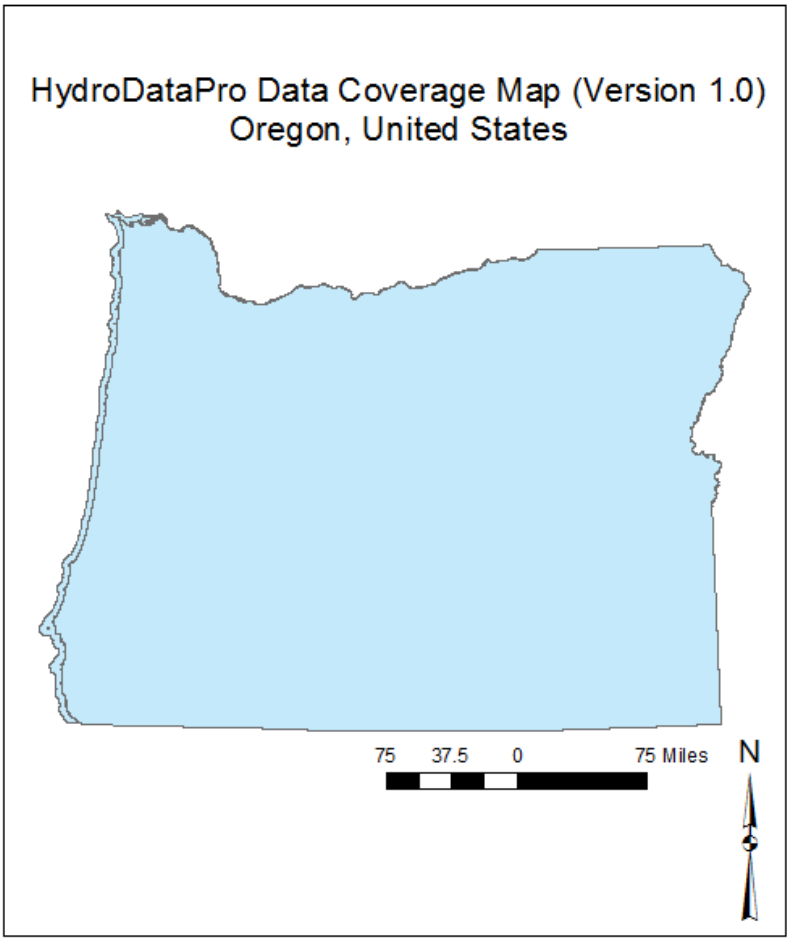


*Figure 1: Data Coverage Map for HydroDataPro (V1).*

## Site Description

The state of Oregon is located in the Pacific Northwest, a region of special research interest to hydrologists, economists, and biologists alike. Often, the hydrology of Oregon is of interest to researchers, and a hydrology model is required. For example, a 2007 study by Dr. David Graves and Dr. Heejun Chang assessed the impact climate change is having on the snowpack driven Upper Clackamas River Basin (Graves & Chang,

2007). They found that because many of the streams are fed directly from melting run off in the spring and summer, flows have, and will continue to decrease. Their model in which they used to come to this conclusion was created with GIS data. This is of course just one example of many, which demonstrate the wide reaching and powerful impact spatial data can have in Oregon. I will not go into further detail on the site description, as the essence of HDP is that it can be utilized anywhere in Oregon, not one specific site.

## Data

*Raster Data:*

- ***DEM –*** *National Elevation Dataset (NED)*
  - *Source: Geospatial Data Gateway*
  - *Native Coordinate System/Projection: GCS North American 1983*
  - *Native Resolution: 30 meter*
- ***Land Use Land Cover –*** National Land Cover Database (NLCD 2011)
  - *Source: Geospatial Data Gateway*
  - *Native Coordinate System/Projection: NAD 1983 UTM Zone 10 North*
  - *Native Resolution: 10 meter*
- ***Soil –*** Gridded Soil Survey (gSSURGO)
  - *Source: Geospatial Data Gateway*
  - *Native Coordinate System/Projection: USA Contiguous Albers Equal Area Conic*
  - *Native Resolution: 10 meter*

*Shapefile Data:*

- ***Watershed Boundary –*** *National Watershed Boundary Dataset (WBD)*
  - *Source: Geospatial Data Gateway*
  - *Native Coordinate System/Projection: GCS North American 1983*
- ***Stream –*** *Flowline (NHDPlus)*
  - *Source: Geospatial Data Gateway*
  - *Native Coordinate System/Projection: GCS North American 1983*
- ***Observed Time Series Data –*** *United States Geological Survey (USGS) Observation Gages*

## GIS/Python Methodology

This section is broken up into two parts: data preparation, and the workflow. The data preparation section will present the steps taken to acquire the data and prepare it for HDP, while the workflow section will cover the GIS methodology that is automated within HDP.

*Data Preparation:*

The most time consuming aspect of this project was the data preparation, specifically the DEM data. To create the statewide raster which is included in HDP, every DEM corresponding to a NED grid (roughly 16) were downloaded individually, unzipped, and then a mosaic operation was used to combine them. The soil, land use/land cover, flowline, and WBD datasets were simply downloaded and clipped (or masked) to the state boundary, as they were available for download at a much larger extent than the DEM data.

*Workflow:*

HDP has a fairly simple and standard workflow, outlined in Figure 2 below. HDP reprojects and clips each input layer to the user's desired projection and boundary. For all data types, the input boundary is buffered by 150 meters, and then the statewide data is clipped (or masked) to the buffer. The reason the boundary layer is buffered is to ensure data within the boundary is not skewed, as the statewide data has yet to be projected. This greatly reduces the amount of processing time required. In addition to being masked and reprojected, each raster layer is resampled to a uniform resolution (specified by the user, with a default of 30 meters if not specified) before the final mask

operation. Once each input layer (both shapefiles and rasters) have been clipped to the

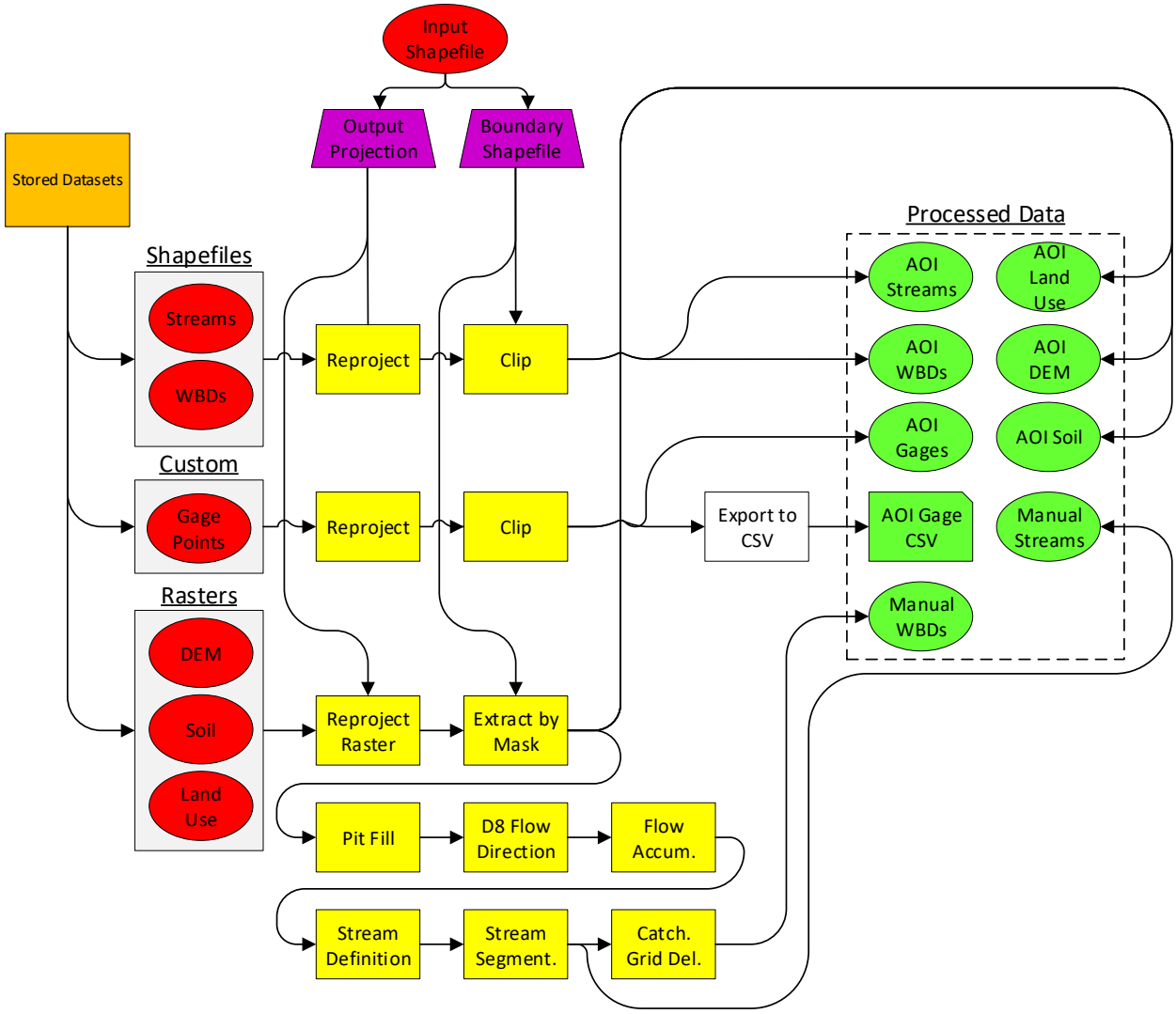


*Figure 2: General GIS Workflow for HydroDataPro (Version 1.0).*

non-buffered boundary, they are stored in the output folder and ready for use.

NHDPlus data, especially stream data can often be inaccurate and messy. For this reason the raster data goes through a second round of processing, which yields a stream raster, and a catchment grid. This methodology follows that of (Maidment & Morehouse, 2002):

1. Fill DEM
    - *Inputs: Fully processed (reprojected, resampled, masked) DEM*
2. Flow Direction
    - *Inputs: Filled DEM*
3. Flow Accumulation
    - *Inputs: Flow Direction Raster*
4. Stream Definition

- *Inputs: Flow Accumulation Raster*

5. Steam Segmentation
    - *Inputs: Stream Definition Raster, Flow Direction Raster*
6. Catchment Grid Delineation
    - *Inputs: Stream Segmentation Raster, Flow Direction Raster*

In step 4, a default stream threshold value of 10,000 cells is used, and is not currently an input parameter of HDP. All of the outputs for the manual delineation are stored in a geodatabase.

## Results

The first version of HydroDataPro provides users a helpful, and open source tool to quickly prepare hydrological data in Oregon. One of the greatest challenges in developing HDP was the size of the data, specifically the statewide DEM data. This not only took a large amount of time to mosaic all of the rasters into one large raster, but this also caused the reprojection and resampling operations to take hours for each run of the tool. Generally speaking, to avoid skewing of spatial data, it is necessary to first reporject before resampling and masking (or clipping). When clipping data prior to reprojecting, the spatial location of data is skewed along the boundary. To overcome both of these issues, it was determined to buffer the input boundary by 150 meters, and clip all statewide data to this prior to any further processing. The 150 meter buffer preserves the data of interest, as the skewed data will fall within the buffer zone. Model performance for a single raster layer improved from approximately 2 hours per layer, to approximately 10 seconds per layer.

While HDP works exclusively with Arc GIS, one of the goals was to make the tool a standalone python script. While the model builder tool can export a script which can be ran in the ArcPy window, this script is not truly a standalone script. To make HDP standalone, the script was modified to check for the spatial analyst license prior to beginning the geoprocessing, allowing the script to be ran from any environment, such as the command line.

Version 1 of HDP works very simply, as the user simply provides one projected project boundary shapefile into the “python_inputs” folder, and then runs the python script (located in the same folder) from any python environment. Each time the script is run, it automatically erases all outputs from the previous run, so the user does not have to. As of right now, the standalone script defaults to a resampling size of 30 meters, which is hard-coded into the script (line 41). The value for flow accumulation defining a stream is also hard-coded into the script, set at 10,000 cells (line 204). However, these two hard coded parameters will be more easily changed in future versions of the tool. HydroDataPro comes with an example HUC 10 watershed boundary, which is used as an example input to the tool. Please refer to the “readme.txt” file for more instructions on running HDP.

*If you would like a copy of HydroDataPro, please contact Nick Giles, at gilesn@oregonstate.edu*

## Appendix:

*References:*

Graves, David, and Heejun Chang. "Hydrologic impacts of climate change in the Upper Clackamas River Basin, Oregon, USA." Climate Research 33, no. 2 (2007): 143-158.

Vieux, Baxter E. "Distributed hydrologic modeling using GIS." In Distributed Hydrologic Modeling Using GIS, pp. 1-17. Springer, Dordrecht, 2001.

Maidment, David R., and Scott Morehouse. Arc Hydro: GIS for water resources. ESRI, Inc., 2002.

*HydroDataPro source code*

```
# -*- coding: utf-8 -*-
# ---------------------------------------------------------------------------
# HDPV1.py
# Created on: 2019-03-17 18:11:16.00000
#   (generated by ArcGIS/ModelBuilder)
# Usage: testpaths4 <hu10out_shp> <Output_Cell_Size>
<Output_Coordinate_System>
# Description: See readme.txt
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy, os, shutil

#Erase existing outputs
outfiles = os.listdir("..\\Output")
scratchfiles = os.listdir("..\\scratch")

if len(outfiles) > 0:
    for f in outfiles:
        if '.' in f and 'gdb' not in f:
            os.remove("..\\Output\\" + f)
        else:
            shutil.rmtree("..\\Output\\" + f)

if len(scratchfiles) > 0:
    for f in scratchfiles:
        if '.' in f and 'gdb' not in f:
            os.remove("..\\scratch\\" + f)
```

```
        else:
            shutil.rmtree("..\\scratch\\" + f)

#Check SA extention
arcpy.CheckOutExtension("Spatial")

# Script arguments
hu10out_shp = arcpy.GetParameterAsText(0)
if hu10out_shp == '#' or not hu10out_shp:
    hu10out_shp = "boundary_proj.shp" # provide a default value if unspecified

Output_Cell_Size = arcpy.GetParameterAsText(1)
if Output_Cell_Size == '#' or not Output_Cell_Size:
    Output_Cell_Size = "30 30" # provide a default value if unspecified

Output_Coordinate_System = arcpy.Describe(hu10out_shp).spatialReference
print "Initializing local variables..."

# Local variables:
soilclip = "..\\Data\\soilclip"
buff_shp = "..\\scratch\\buff.shp"
inclip = "..\\scratch\\inclip"
reproj = "..\\scratch\\reproj"
Resamp = "..\\scratch\\resamp"
AOI_soil = "..\\Output\\AOI_soil"
hu8_shp = "..\\Data\\hu8.shp"
inclip2_shp = "..\\scratch\\inclip2.shp"
reproj2_shp = "..\\scratch\\reproj2.shp"
```

AOI_huc8_shp = "..\\Output\\AOI_huc8.shp"

hu10_shp = "..\\Data\\hu10.shp"

inclip3_shp = "..\\scratch\\inclip3.shp"

reproj3_shp = "..\\scratch\\reproj3.shp"

AOI_hu10_shp = "..\\Output\\AOI_hu10.shp"

hu12_shp = "..\\Data\\hu12.shp"

inclip4_shp = "..\\scratch\\inclip4.shp"

reproj4_shp = "..\\scratch\\reproj4.shp"

AOI_huc12_shp = "..\\Output\\AOI_huc12.shp"

nlcd_or_utm10_tif = "..\\Data\\nlcd_or_utm10.tif"

inclip5 = "..\\scratch\\inclip5"

reproj5 = "..\\scratch\\reproj5"

resamp5 = "..\\scratch\\resamp5"

AOI_lulc = "..\\Output\\AOI_lulc"

dem_or1_tif = "..\\Data\\dem_or1.tif"

inclip6 = "..\\scratch\\inclip6"

reproj6 = "..\\scratch\\reproj6"

resamp6 = "..\\scratch\\resamp6"

AOI_dem = "..\\Output\\AOI_dem"

nhd24kst_l_or_shp = "..\\Data\\nhd24kst_l_or.shp"

inclip7_shp = "..\\scratch\\inclip7.shp"

reproj7_shp = "..\\scratch\\reproj7.shp"

AOI_streams_shp = "..\\Output\\AOI_streams.shp"

usgs_gages_shp = "..\\Data\\usgs_gages.shp"

inclip8_shp = "..\\scratch\\inclip8.shp"

reproj8_shp = "..\\scratch\\reproj8.shp"

AOI_USGS_gage_shp = "..\\Output\\AOI_USGS_gage.shp"

usgs_gages_xls = "..\\Output\\usgs_gages.xls"

```
print "Undergoing data processing..."

# Process: Buffer

arcpy.Buffer_analysis(hu10out_shp, buff_shp, "150 Meters", "FULL", "ROUND",
"NONE", "", "PLANAR")


# Process: Extract by Mask (2)

arcpy.gp.ExtractByMask_sa(soilclip, buff_shp, inclip)


# Process: Project Raster

arcpy.ProjectRaster_management(inclip, reproj, Output_Coordinate_System,
"NEAREST", "10 10", "", "",
"PROJCS['NAD_1983_Albers',GEOGCS['GCS_North_American_1983',DATUM['D_Nort
h_American_1983',SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Gre
enwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Albers'],PARAMETE
R['False_Easting',0.0],PARAMETER['False_Northing',0.0],PARAMETER['Central_Merid
ian',-
96.0],PARAMETER['Standard_Parallel_1',29.5],PARAMETER['Standard_Parallel_2',45.
5],PARAMETER['Latitude_Of_Origin',23.0],UNIT['Meter',1.0]]")


# Process: Resample

arcpy.Resample_management(reproj, Resamp, Output_Cell_Size, "NEAREST")


# Process: Extract by Mask

arcpy.gp.ExtractByMask_sa(Resamp, hu10out_shp, AOI_soil)


# Process: Clip (2)

arcpy.Clip_analysis(hu8_shp, buff_shp, inclip2_shp, "")


# Process: Project

arcpy.Project_management(inclip2_shp, reproj2_shp, Output_Coordinate_System,
"'WGS_1984_(ITRF00)_To_NAD_1983 +
```

```
WGS_1984_(ITRF08)_To_NAD_1983_2011'",
"GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROI
D['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.
0174532925199433]]", "NO_PRESERVE_SHAPE", "", "NO_VERTICAL")


# Process: Clip (3)

arcpy.Clip_analysis(reproj2_shp, hu10out_shp, AOI_huc8_shp, "")


# Process: Clip (4)

arcpy.Clip_analysis(hu10_shp, buff_shp, inclip3_shp, "")


# Process: Project (2)

arcpy.Project_management(inclip3_shp, reproj3_shp, Output_Coordinate_System,
'"WGS_1984_(ITRF00)_To_NAD_1983 +
WGS_1984_(ITRF08)_To_NAD_1983_2011'",
"GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROI
D['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.
0174532925199433]]", "NO_PRESERVE_SHAPE", "", "NO_VERTICAL")


# Process: Clip (5)

arcpy.Clip_analysis(reproj3_shp, hu10out_shp, AOI_hu10_shp, "")


# Process: Clip (6)

arcpy.Clip_analysis(hu12_shp, buff_shp, inclip4_shp, "")


# Process: Project (3)

arcpy.Project_management(inclip4_shp, reproj4_shp, Output_Coordinate_System,
'"WGS_1984_(ITRF00)_To_NAD_1983 +
WGS_1984_(ITRF08)_To_NAD_1983_2011'",
"GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROI
D['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.
0174532925199433]]", "NO_PRESERVE_SHAPE", "", "NO_VERTICAL")
```

# Process: Clip (7)

arcpy.Clip_analysis(reproj4_shp, hu10out_shp, AOI_huc12_shp, "")

# Process: Extract by Mask (3)

arcpy.gp.ExtractByMask_sa(nlcd_or_utm10_tif, buff_shp, inclip5)

# Process: Project Raster (2)

arcpy.ProjectRaster_management(inclip5, reproj5, Output_Coordinate_System, "NEAREST", "30 30", "", "", "PROJCS['NAD_1983_UTM_Zone_10N',GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],PROJECTION['Transverse_Mercator'],PARAMETER['False_Easting',500000.0],PARAMETER['False_Northing',0.0],PARAMETER['Central_Meridian',-123.0],PARAMETER['Scale_Factor',0.9996],PARAMETER['Latitude_Of_Origin',0.0],UNIT['Meter',1.0]]")

# Process: Resample (2)

arcpy.Resample_management(reproj5, resamp5, Output_Cell_Size, "NEAREST")

# Process: Extract by Mask (4)

arcpy.gp.ExtractByMask_sa(resamp5, hu10out_shp, AOI_lulc)

# Process: Extract by Mask (5)

arcpy.gp.ExtractByMask_sa(dem_or1_tif, buff_shp, inclip6)

# Process: Project Raster (3)

arcpy.ProjectRaster_management(inclip6, reproj6, Output_Coordinate_System, "NEAREST", "8.73935802822987 8.73935802822987", "", "", "GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROID['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.0174532925199433]],VERTCS['Unknown

```
VCS',VDATUM['Unknown'],PARAMETER['Vertical_Shift',0.0],PARAMETER['Direction',1
.0],UNIT['Meter',1.0]]")

# Process: Resample (3)

arcpy.Resample_management(reproj6, resamp6, Output_Cell_Size, "NEAREST")

# Process: Extract by Mask (6)

arcpy.gp.ExtractByMask_sa(resamp6, hu10out_shp, AOI_dem)

# Process: Clip (9)

arcpy.Clip_analysis(nhd24kst_l_or_shp, buff_shp, inclip7_shp, "")

# Process: Project (4)

arcpy.Project_management(inclip7_shp, reproj7_shp, Output_Coordinate_System,
'"WGS_1984_(ITRF00)_To_NAD_1983 +
WGS_1984_(ITRF08)_To_NAD_1983_2011"',
"GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROI
D['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.
0174532925199433]]", "NO_PRESERVE_SHAPE", "", "NO_VERTICAL")

# Process: Clip (8)

arcpy.Clip_analysis(reproj7_shp, hu10out_shp, AOI_streams_shp, "")

# Process: Clip (10)

arcpy.Clip_analysis(usgs_gages_shp, buff_shp, inclip8_shp, "")

# Process: Project (5)

arcpy.Project_management(inclip8_shp, reproj8_shp, Output_Coordinate_System,
'"WGS_1984_(ITRF00)_To_NAD_1983 +
WGS_1984_(ITRF08)_To_NAD_1983_2011"',
"GEOGCS['GCS_North_American_1983',DATUM['D_North_American_1983',SPHEROI
D['GRS_1980',6378137.0,298.257222101]],PRIMEM['Greenwich',0.0],UNIT['Degree',0.
```

```
0174532925199433]],VERTCS['Unknown
VCS',VDATUM['Unknown'],PARAMETER['Vertical_Shift',0.0],PARAMETER['Direction',1
.0],UNIT['Meter',1.0]]", "NO_PRESERVE_SHAPE", "", "NO_VERTICAL")


# Process: Clip (11)

arcpy.Clip_analysis(reproj8_shp, hu10out_shp, AOI_USGS_gage_shp, "")


# Process: Table To Excel

arcpy.TableToExcel_conversion(AOI_USGS_gage_shp, usgs_gages_xls, "NAME",
"CODE")


print "Begining Manual Deliniation..."


# Input DEM

Output_drop_raster = ""


arcpy.CreateFileGDB_management("..\\Output", "BasinDel.gdb")

last_folder = os.getcwd().rfind('\\')

rel_path = os.getcwd()

rel_path = rel_path[0:last_folder]


# Process: Fill

Fill_tif1 = rel_path + "\\Output\\BasinDel.gdb\\Fill_tif1"

arcpy.gp.Fill_sa(AOI_dem, Fill_tif1, "")


# Process: Flow Direction

Output_flow_direction_raster = rel_path + "\\Output\\BasinDel.gdb\\FlowDir_Fill1"

arcpy.gp.FlowDirection_sa(Fill_tif1, Output_flow_direction_raster, "NORMAL",
Output_drop_raster)
```

```
# Process: Flow Accumulation

Output_accumulation_raster = rel_path + "\\Output\\BasinDel.gdb\\FlowAcc_Flow1"

arcpy.gp.FlowAccumulation_sa(Output_flow_direction_raster,
Output_accumulation_raster, "", "FLOAT")


# Process: Basin

Output_raster = rel_path + "\\Output\\BasinDel.gdb\\Catchments"

arcpy.gp.Basin_sa(Output_flow_direction_raster, Output_raster)


# Local variables:

Input_true_raster_or_constant_value__2_ = "1"

Con_FlowAcc_2 = rel_path + "\\Output\\BasinDel.gdb\\Streams"


# Deliniate Streams based on FlowAcc > user_input Condition

arcpy.gp.Con_sa(Output_accumulation_raster,
Input_true_raster_or_constant_value__2_, Con_FlowAcc_2, "", "VALUE > 10000")


print "HDP run complete!"
```