



Python in Earth Science Applications

Ali L. Yagci

Post-Doctoral Fellow

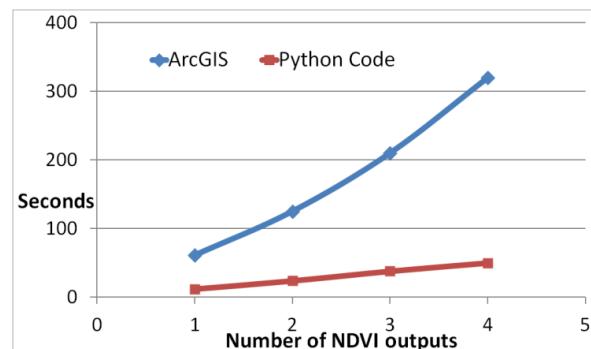
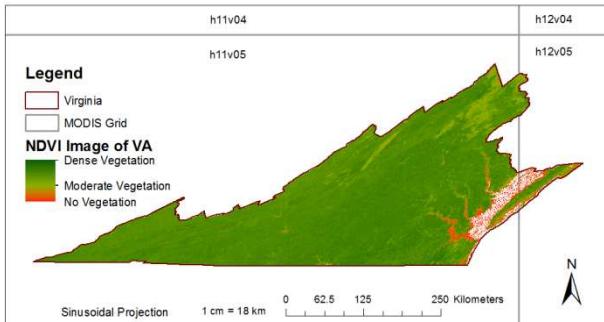
Hydrological Sciences Laboratory (617)

First Acquaintance

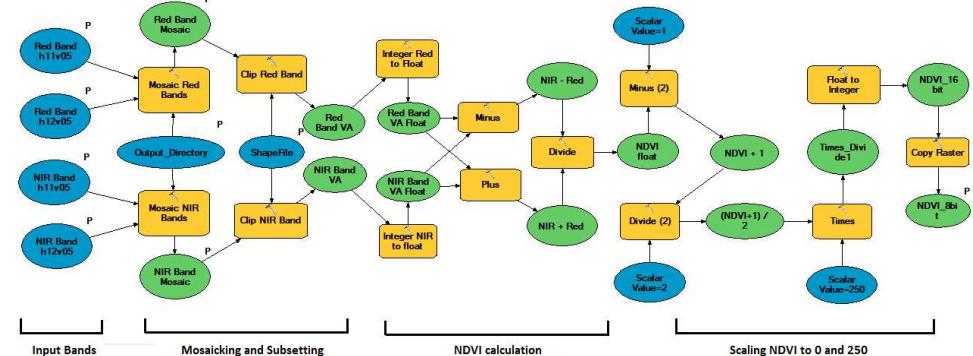
- Background
 - Geographical Information Systems (GIS)
 - Remote Sensing
- Introduction to GIS programming back in the graduate School
 - Introduction to Python
 - Implement a method to read shape file and visualize it on Tkinter.
 - Python's de-facto standard GUI (Graphical User Interface) package.
 - ESRI shape file specifications
- Introduction to Object-Oriented Programming
 - JAVA
- MATLAB
 - Heavy and hungry for memory

Simple Applications

- Simple NDVI calculation for Virginia.
 - Two MODIS sensor tiles
 - Calculate NDVI,
 - Mosaic two tiles,
 - Clip the results by VA
- Per run
 - 61 seconds (ArcGIS per run)
 - 12 second with Python
- Why Python?
 - No Intermediate file clutter
 - Passing results as an array to functions
 - Fast
 - Customizable
 - Easily automated for time series analysis



ArcGIS model builder



Python Script

```
generatBatchNDVI.py -l /home/sly/myPythonLib/generatBatchNDVI.py
File Edit Format Run Options Windows Help
remove necessary modules
from osgeo import gdal
import os, time
import agibrought as ag
import gdalmerge as gm
# Initial time
e=time.time()
# Input Variables
inFolders="/media/5B707D58707D37BB/Downloads/GIS-day_poster_2012/Data"
shpFile="/media/Seagate_Expansion_Drive/Personal_Data/US_Census_Bureau/TIGERLINE_Shapefiles/"
outFolder="/media/5B707D58707D37BB/Downloads/GIS-day_poster_2012/Python_output"
nodata=-28672
# Collect red and nir bands into lists
redList=[]; nirList=[]
for i in os.listdir(inFolders):
    if i.find("Red") > -1 and i[i.rfind(".")+len(i)] == ".tif":
        redList.append(i)
    elif i.find("Red") < -1 and i[i.rfind(".")+len(i)] == ".tif":
        redList.append(i)
# sort the lists
redList.sort(); nirList.sort()
# Mosaic and Clip by shape and NDVI calculation
for i in xrange(0,int(len(redList)/2)):
    # Mosaic
    outMosaicRed=outFolder+"/"+redList[i].replace("Red","Mosaic.Red")
    outMosaicNIR=outFolder+"/"+nirList[i].replace("NIR","Mosaic.NIR")
    gm.mosaic([inFolder+"/"+redList[i],inFolder+"/"+redList[i+1]],outMosaicRed,nodata)
    gm.mosaic([inFolder+"/"+nirList[i],inFolder+"/"+nirList[i+1]],outMosaicNIR,nodata)
    # Clip by shape file
    redVA=ag.clipByShpToMemory(outMosaicRed,shpFile)
    nirVA=ag.clipByShpToMemory(outMosaicNIR,shpFile)
    # Calculate NDVI
    outNDVIFile=outFolder+"/"+redList[i].replace("Red","NDVI")
    ag.ndviFromMemory(redVA,nirVA,outNDVIFile)
    # Release memory
    redVA=None; nirVA=None
# Final Time
elapsedTime=time()-e
print '\nWhole operation took', round(elapsedTime,4), 'seconds'
```

3

Ln: 21 Col: 9

Applications

- Drought Monitoring
 - Based on the Earth observing satellite products
 - Vegetation Indices
 - Normalized Vegetation Difference Index (NDVI)
 - Enhanced Vegetation Index (EVI)
 - Land Surface Temperature (LST)
- Evapotranspiration Modeling
 - NDVI, LST and maximum/minimum air temperature
- Time series extraction from satellite data for validation and analysis.

How I do it?

- Ubuntu operating system
 - 14.04.4 LTS
- Libraries and their dependencies handled by Anaconda.
- Integrated Development Environment
 - Spyder and IDLE
- Old way downloading libraries from Ubuntu GIS stable/unstable repository
 - <https://launchpad.net/~ubuntugis/+archive/ubuntu/ppa>
 - Not convenient, dependency problems.
 - All libraries not available.
 - Not latest release libraries.

Spyder IDE

The screenshot illustrates the Spyder IDE environment, featuring two main windows:

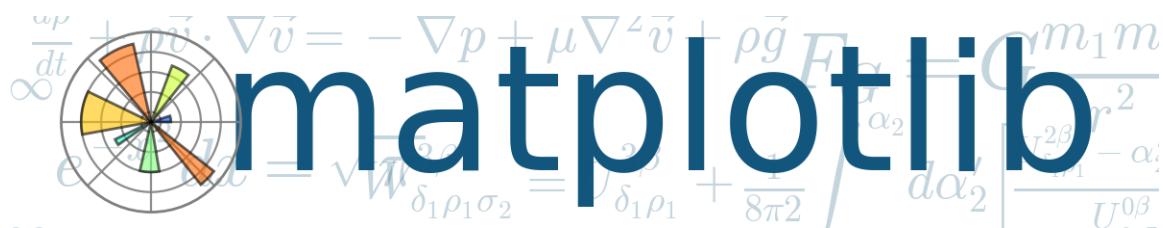
- Top Window:** Displays a Python script named `functionsArray.py`. The code implements various geospatial operations, including reading shapefiles, extracting values from rasters, and writing output files. It uses libraries like `gdal` and `osgeo`. The code includes comments explaining its functionality, such as calculating the number of features in a layer or extracting pixel values by points.
- Bottom Window:** Shows the Spyder interface with the following components:
 - Console:** Displays the Python interpreter output, including the Python version (2.7.11), copyright information, and a help message.
 - Object Inspector:** Shows the current state of variables in the kernel.
 - File Browser:** Shows the directory structure of the project, including files like `functionsBatch.py`, `functionsTextFile.py`, and `functionsUtilities.py`.

Libraries

- Using existing libraries, rather than implementing from scratch.
 - Handling file input/output. Read files and passing it to libraries.
- GDAL/OGR
 - GDAL: Geospatial Data Abstraction Library
 - OGR: OpenGIS Simple Features Reference Implementation.
 - Reading geospatial data in raster and vector format.
 - inFile=gdal.Open(raster format like HDF, GeoTiff and NetCDF)
 - inFile=ogr.Open(raster format like HDF, GeoTiff and NetCDF)

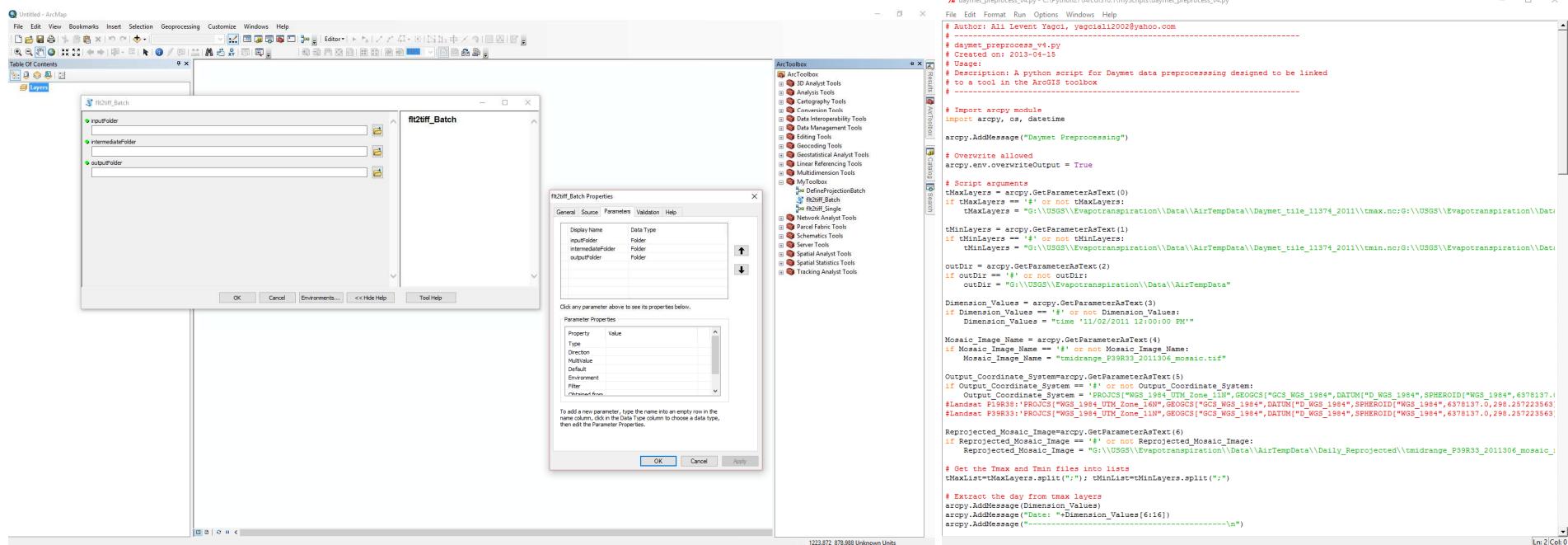
Libraries

- Scientific Libraries for Array computation
 - NumPy: Reading 2-D data into numpy arrays
 - SciPy: some utility functions for arrays
 - Matplotlib: plotting



Libraries from Commercial GIS Software

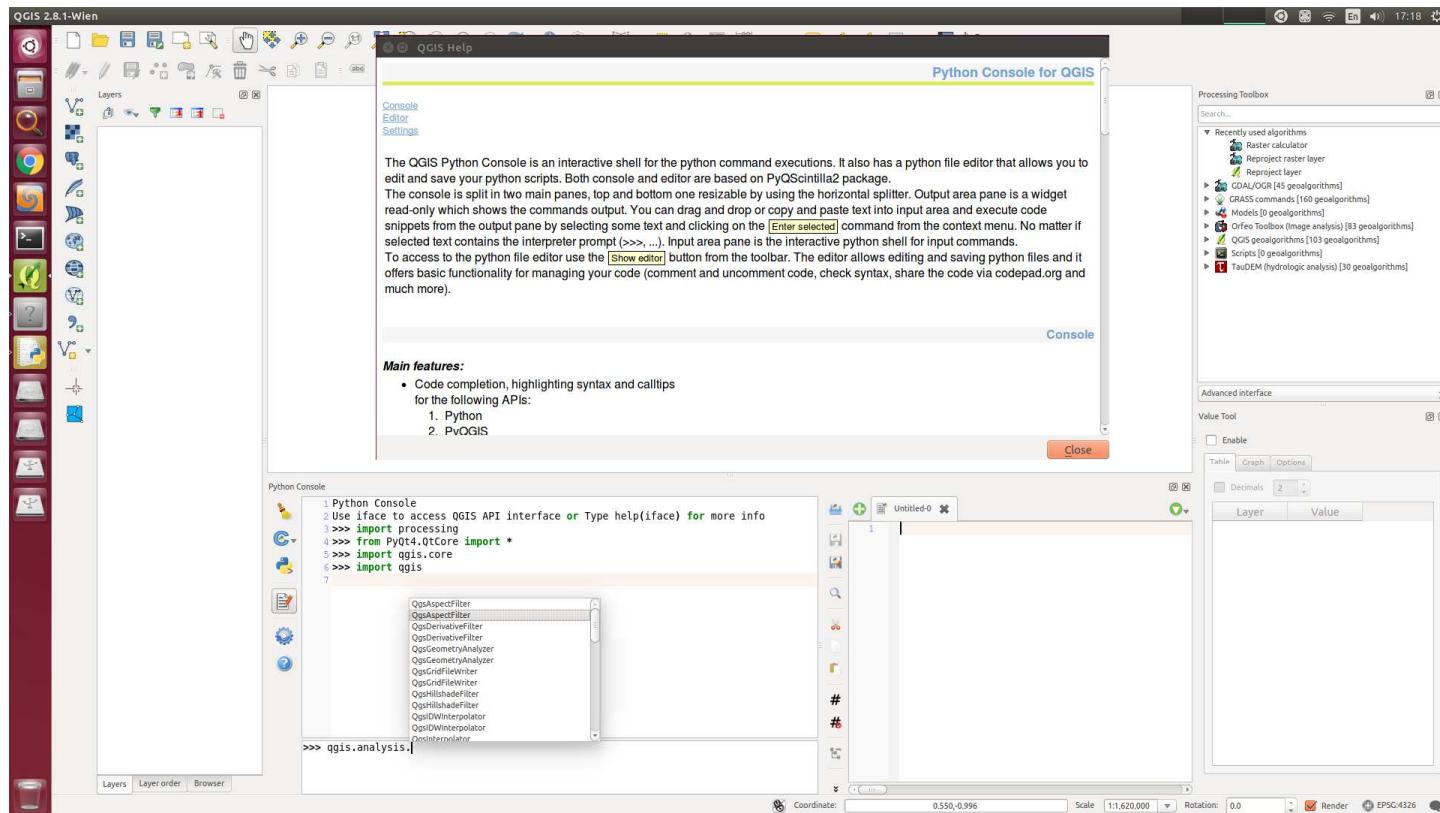
- ArcGIS
 - Built-in python console or,
 - Stand-alone python script
- ArcPy
 - Associate Python script to a tool and save it in toolbox.



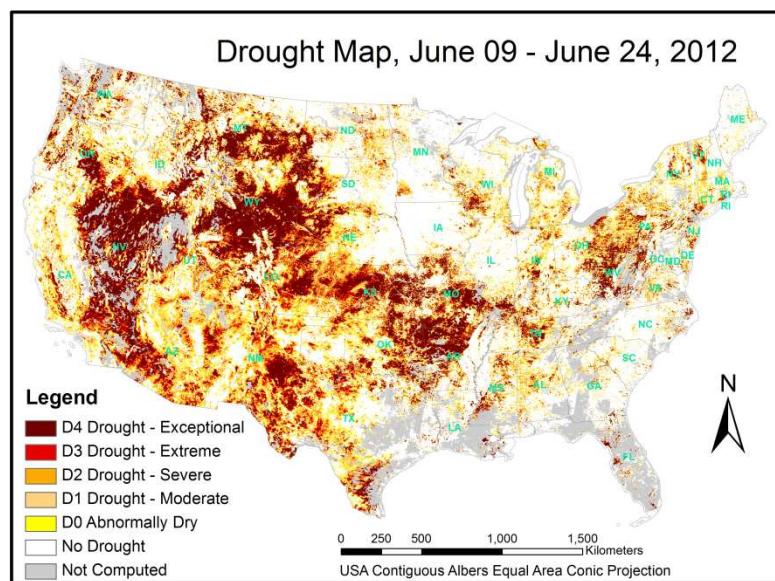
- `inputFolder = arcpy.GetParameterAsText(0)`

Libraries from Free GIS Software

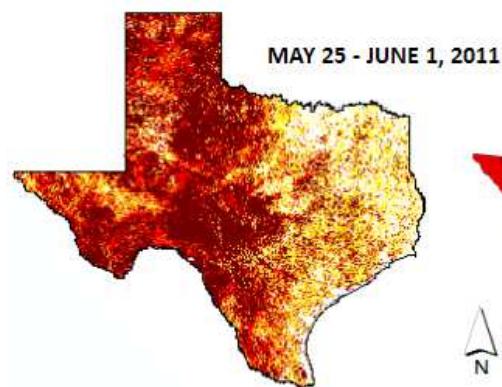
- Quantum GIS (QGIS)
 - No license issues
- Built-in Python console and editor
 - Stand-alone python scripts
 - Associate Python script to a tool and save it in toolbox.



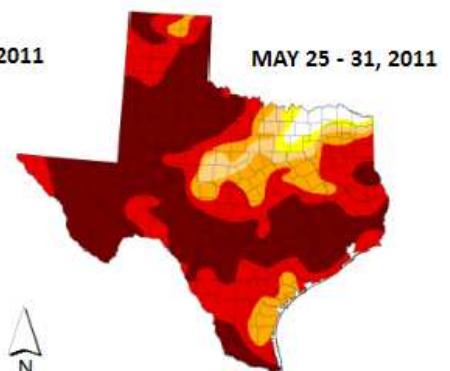
Drought Monitoring



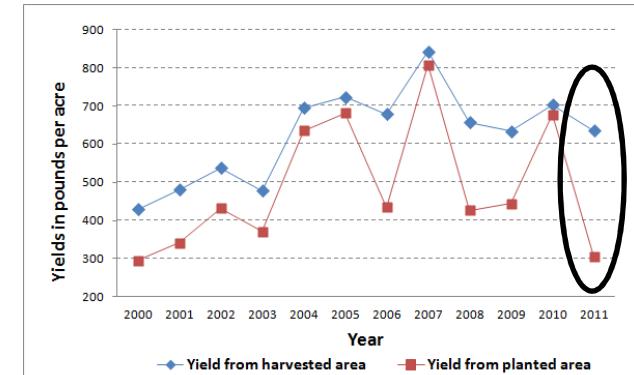
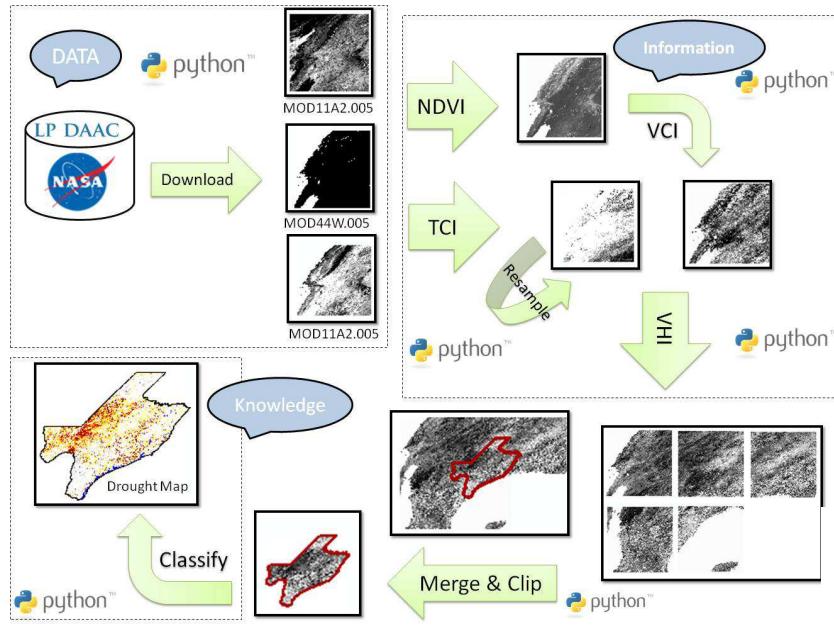
MODEL OUTPUT



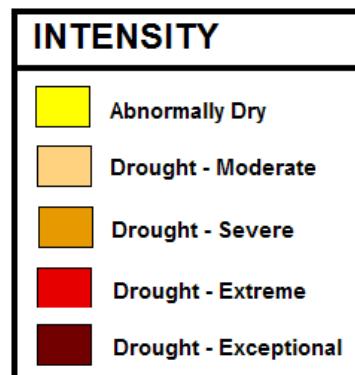
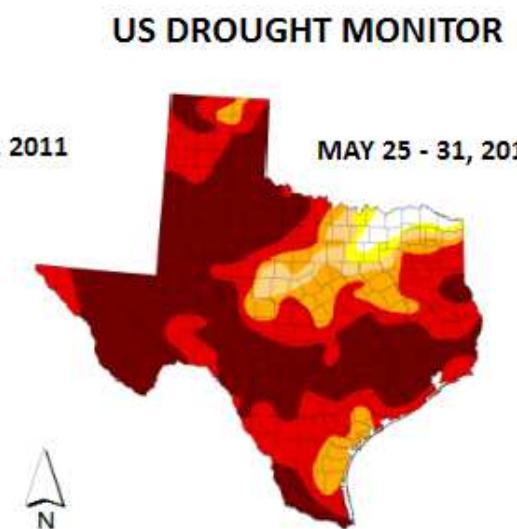
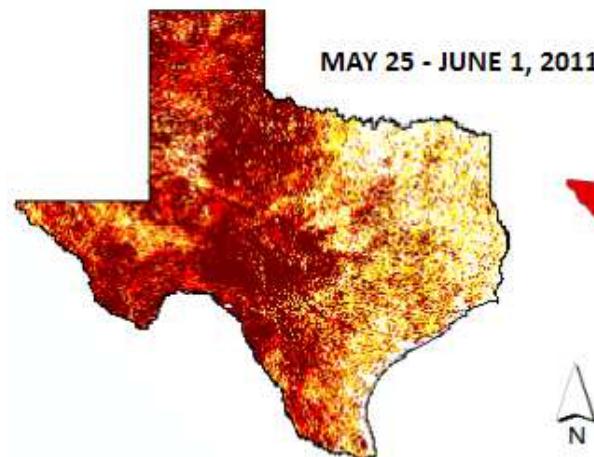
US DROUGHT MONITOR



Texas Drought in 2011



MODEL OUTPUT



Texas Drought in 2011

U.S. Drought Monitor

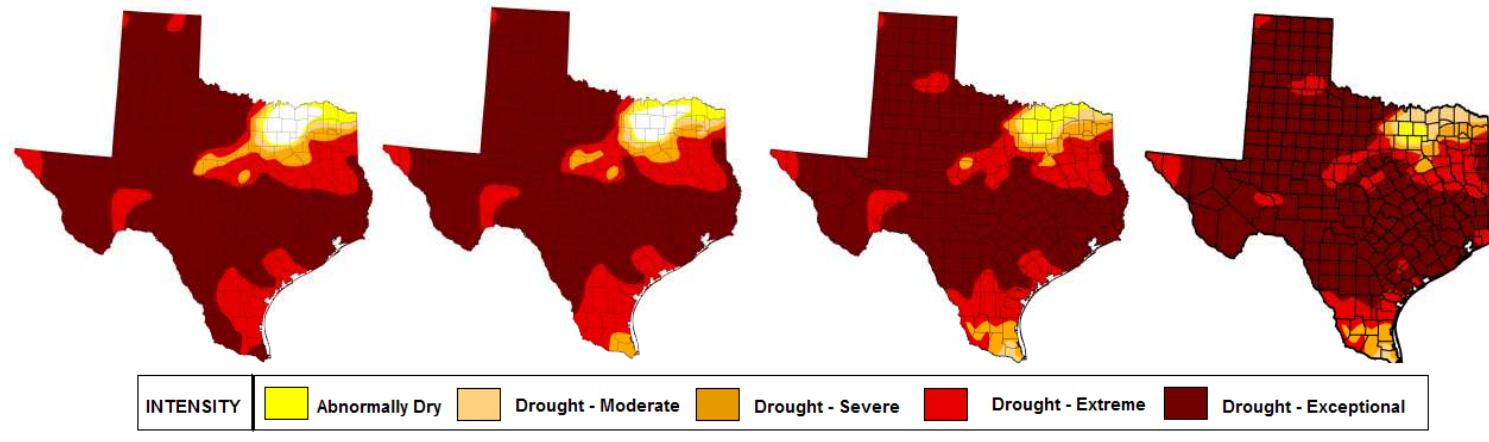
TEXAS

June 28, 2011

July 5, 2011

July 12, 2011

July 19, 2011

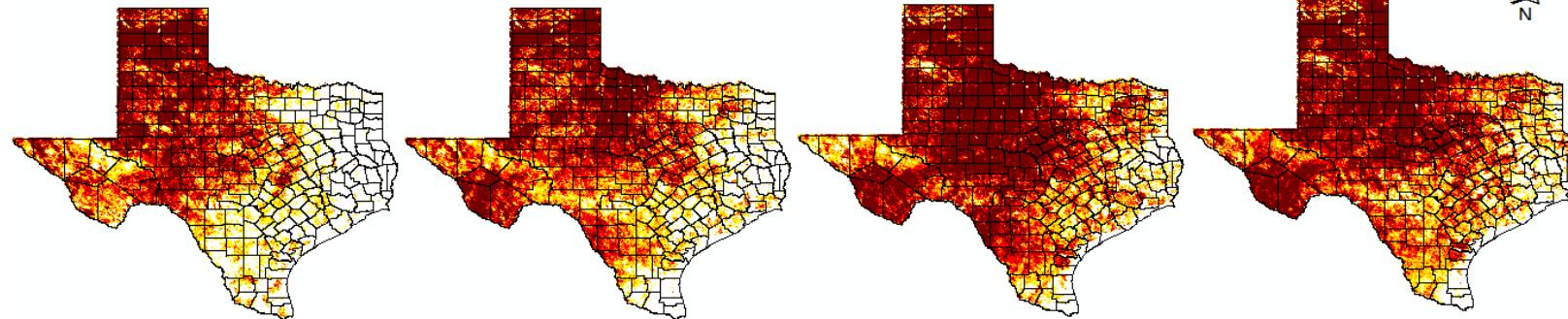


June 26, 2011

July 04, 2011

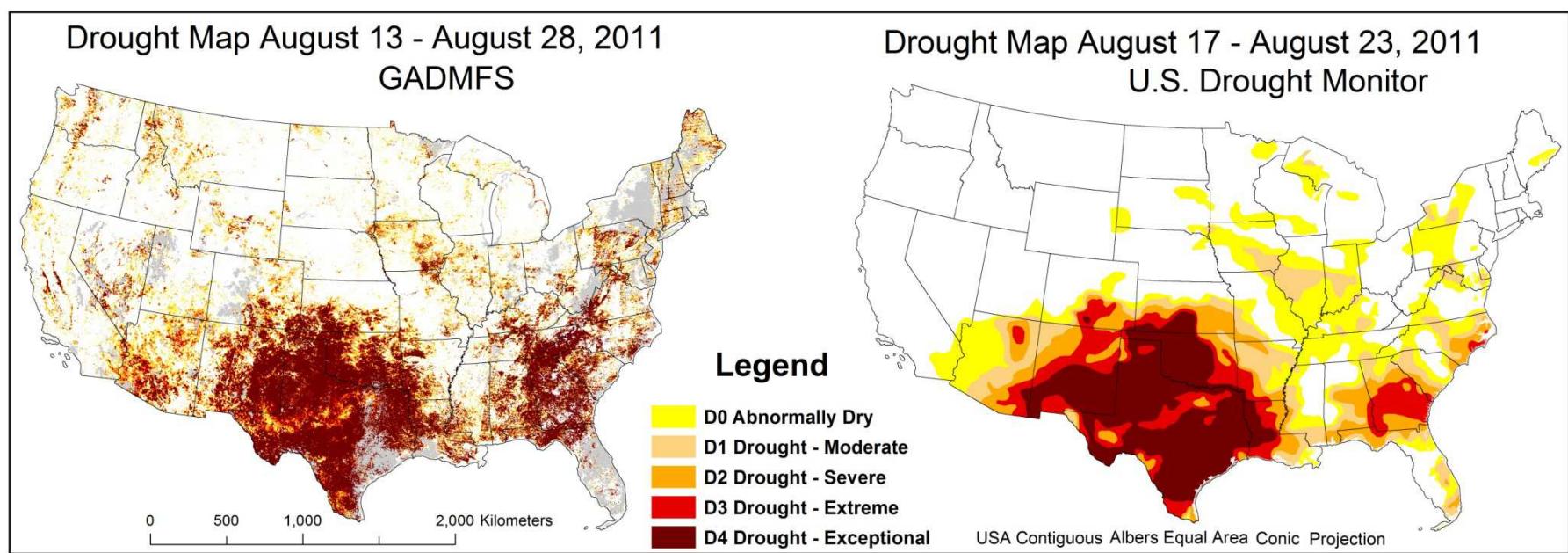
July 12, 2011

July 20, 2011

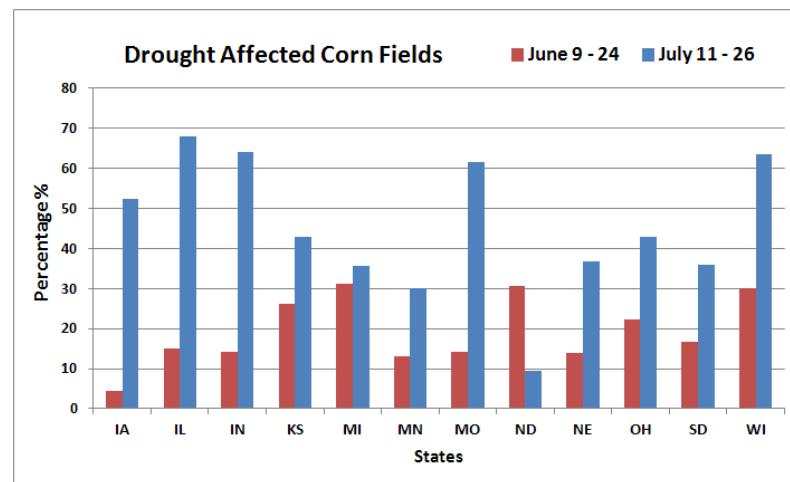
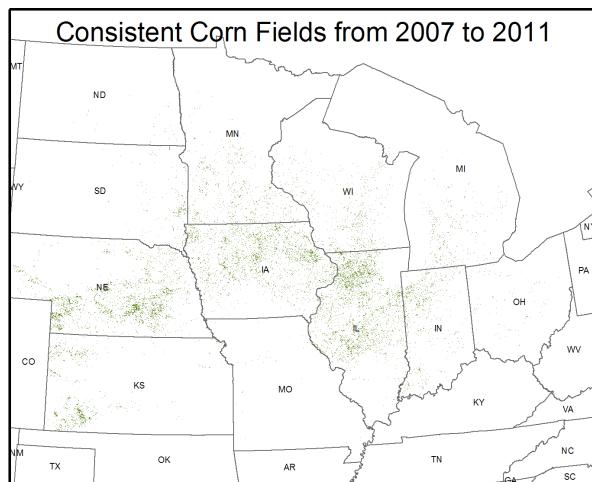
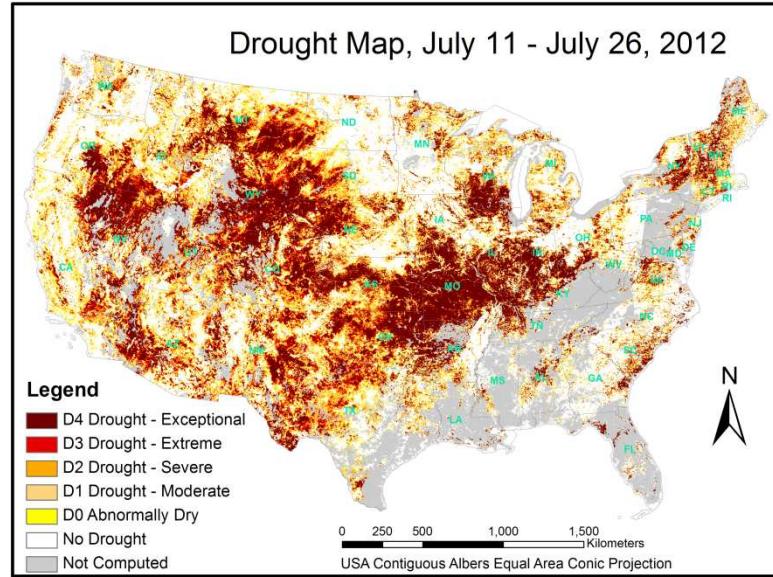
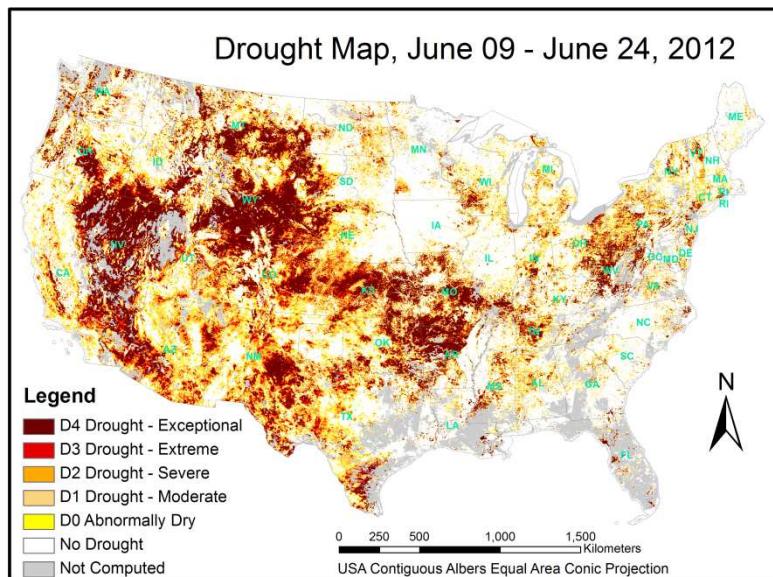


North American 1983 Projection

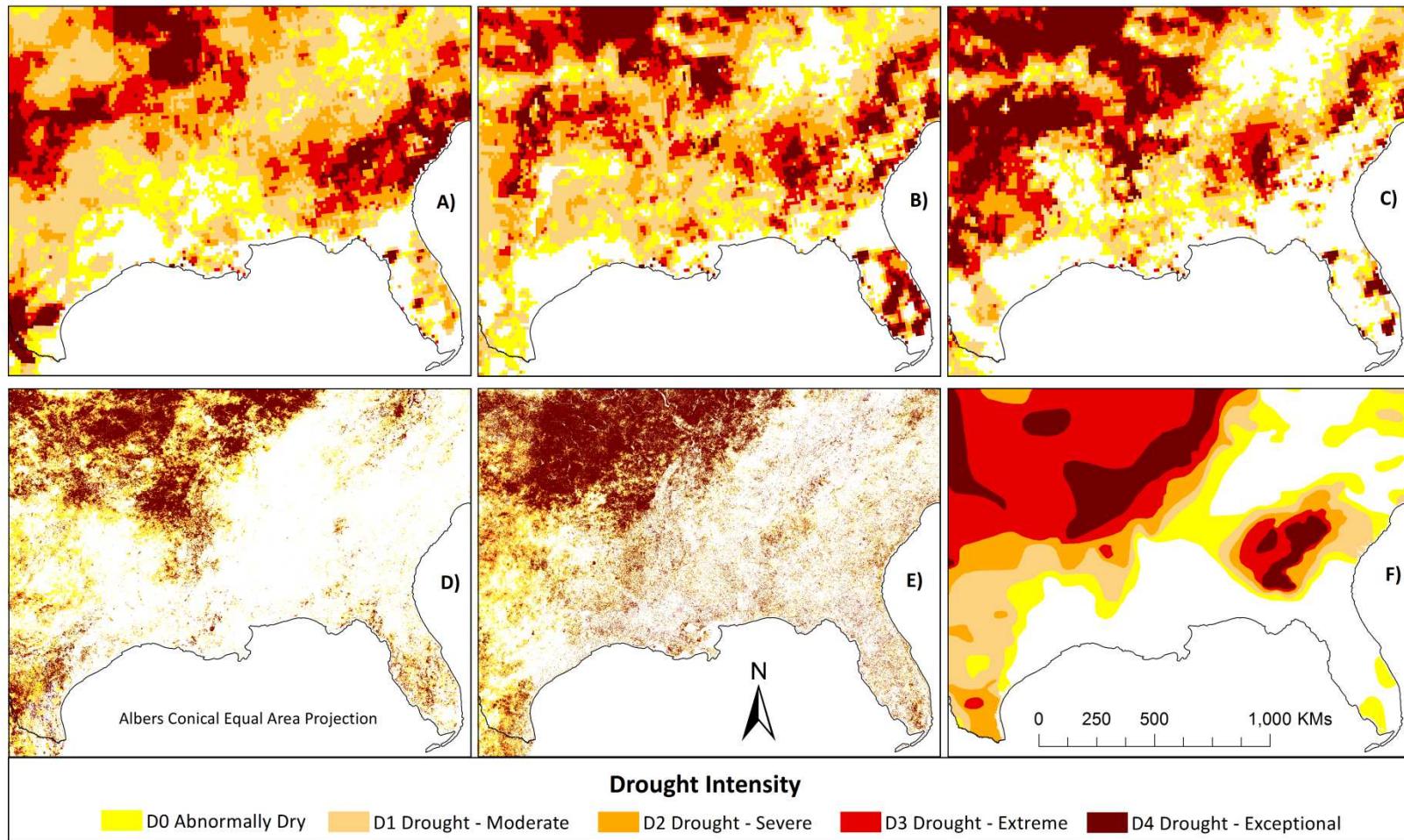
Drought in 2011



Drought-affected corn fields in 2012

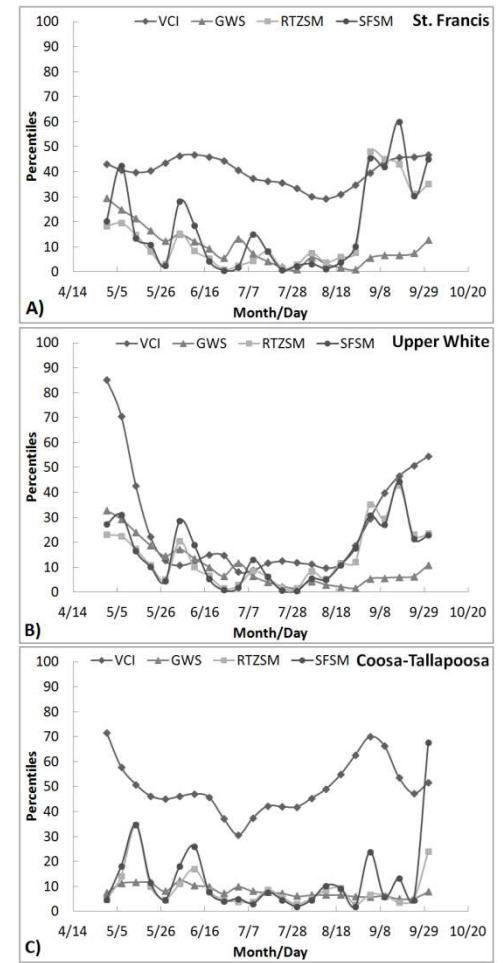
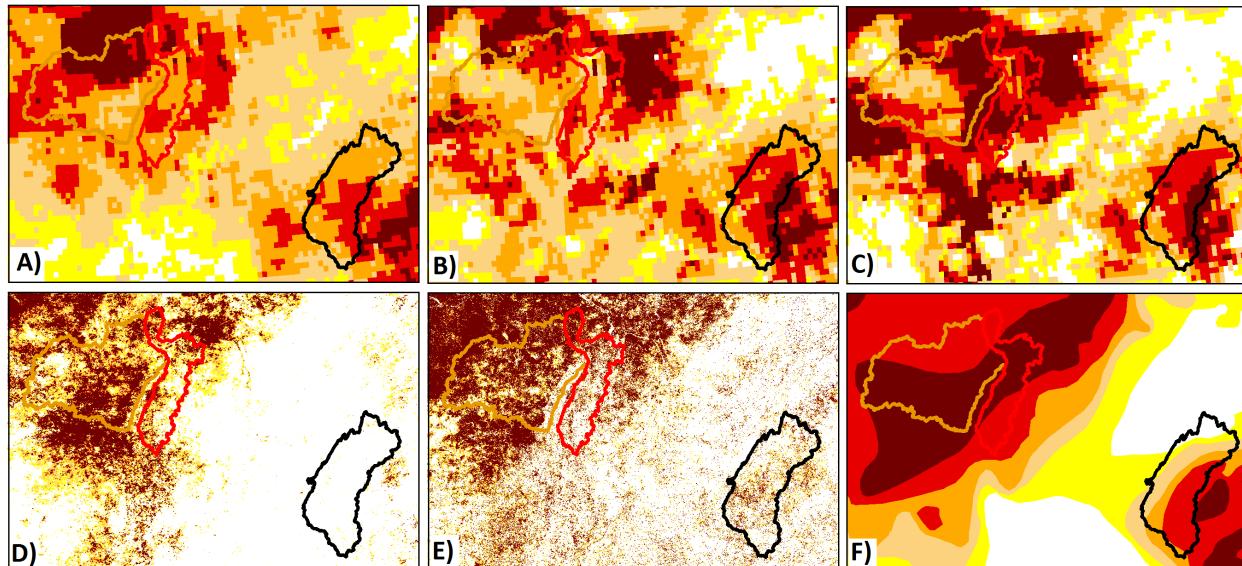


2012 Drought in Southeast US

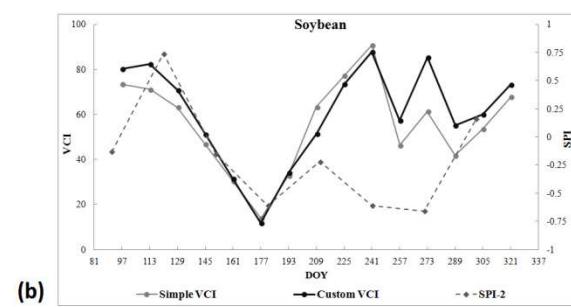
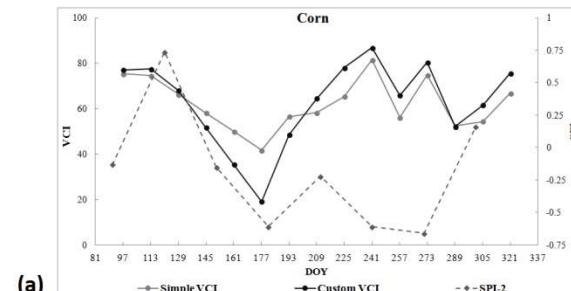
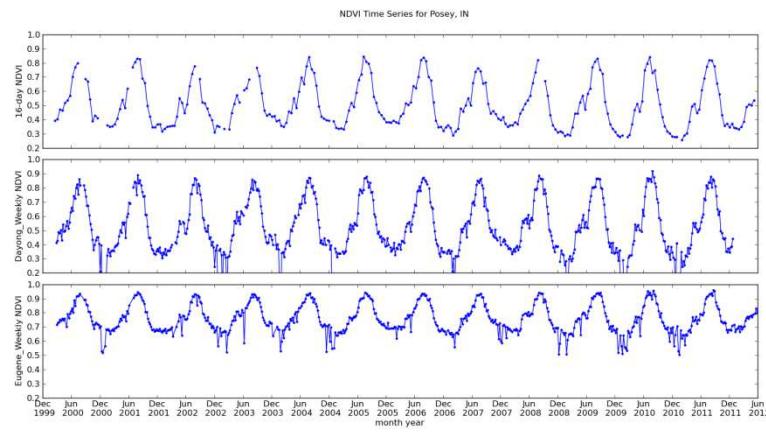


2012 Drought in Southeast US

- Creating time series of basin averages of drought indices by boundary files of Basins and writing them into text file.

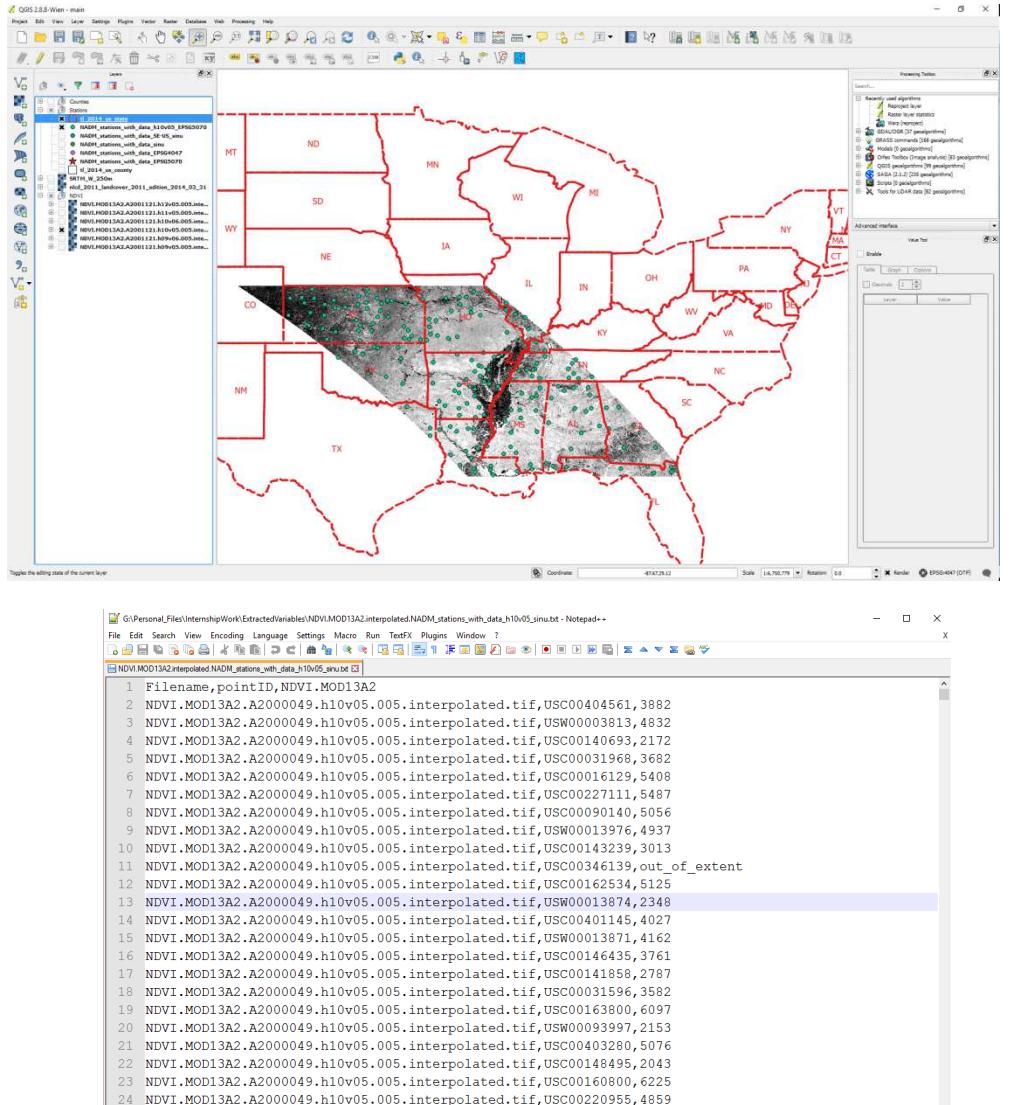


Time Series Generation

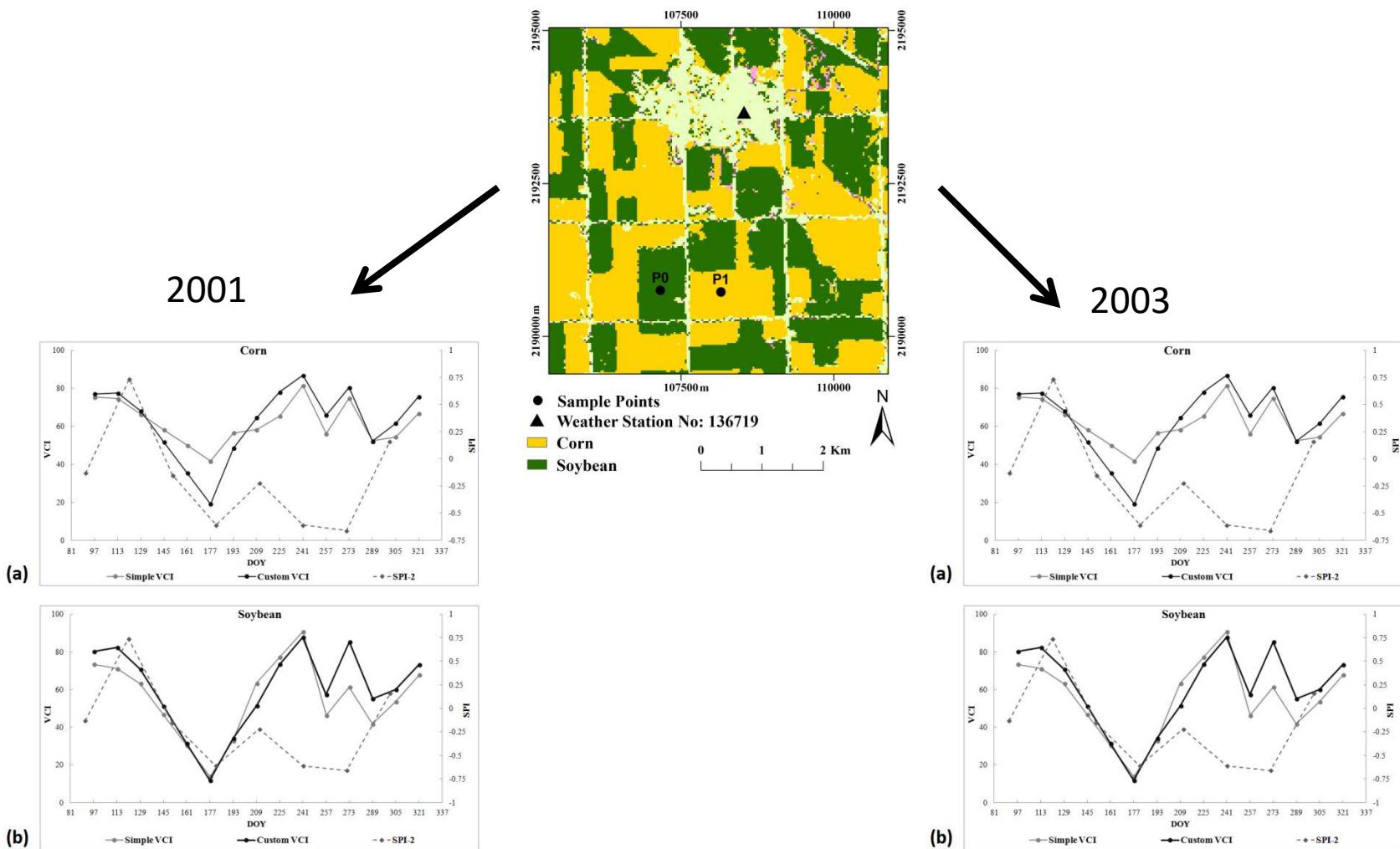


Time series extraction by points

- 196 weather stations in a MODIS tile.
- Extracting overlapping NDVI and EVI pixels with stations between 2000 and 2016 (391 MODIS layers).

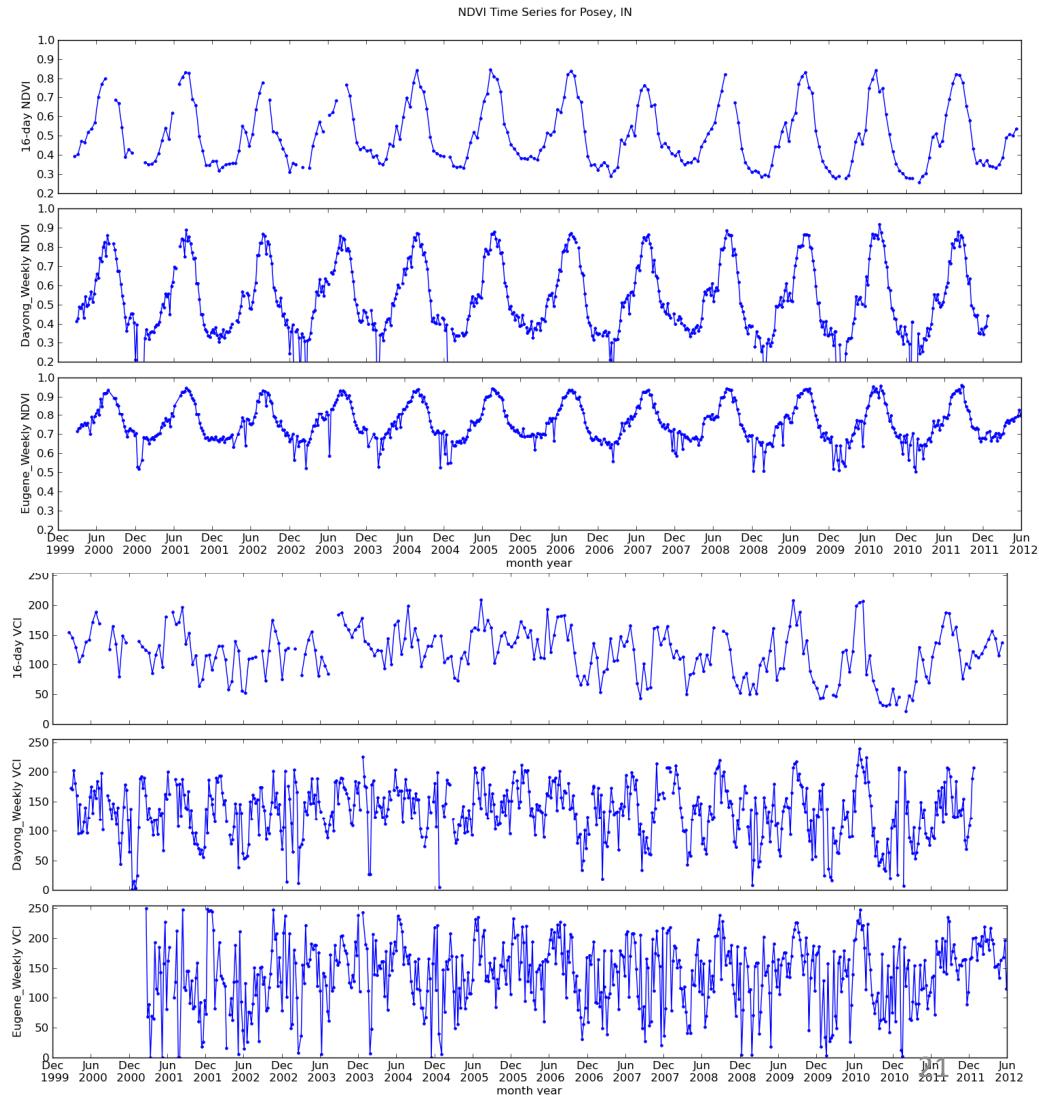


Time series extraction by points



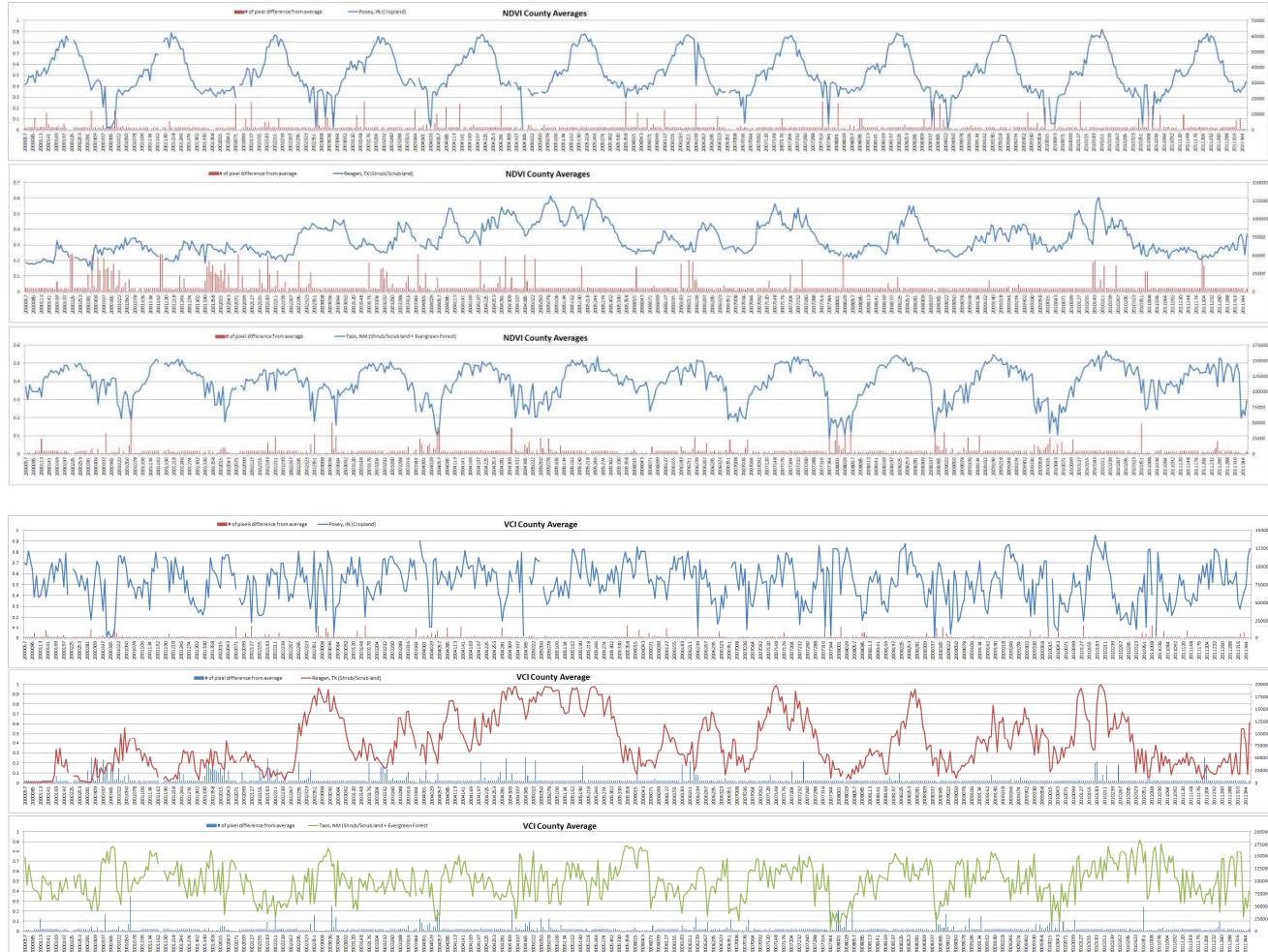
Time Series generation

- Generating Time series of NDVI and VCI by county boundaries.
- No intermediate files, everything done on memory
- Reusable functions
- Plotted in Matplotlib

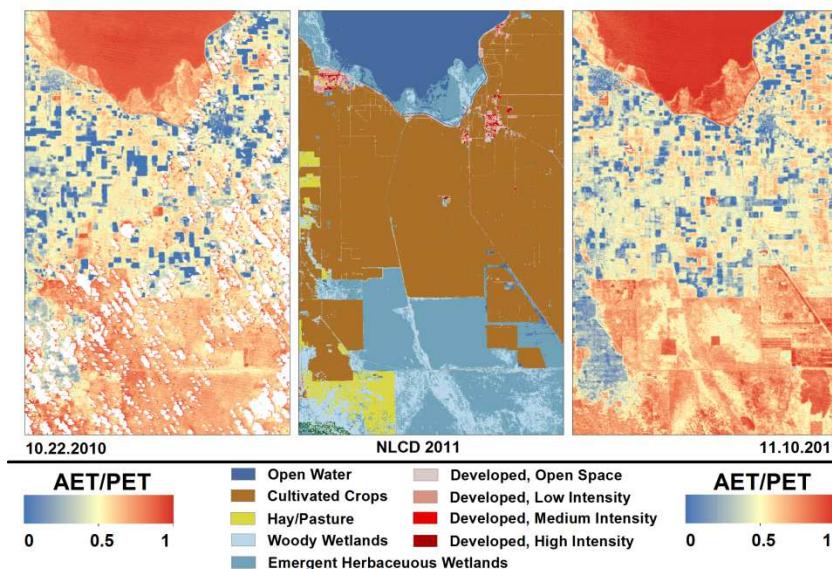


Time Series generation 2. way

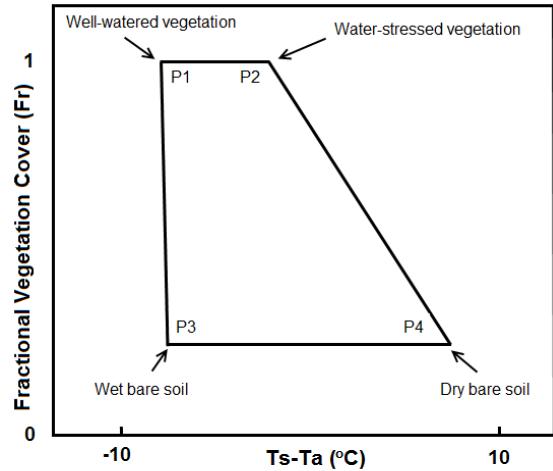
- Extract time series of indices with counts into text files.
- Import the text files into excel and plots them.
- More interactive



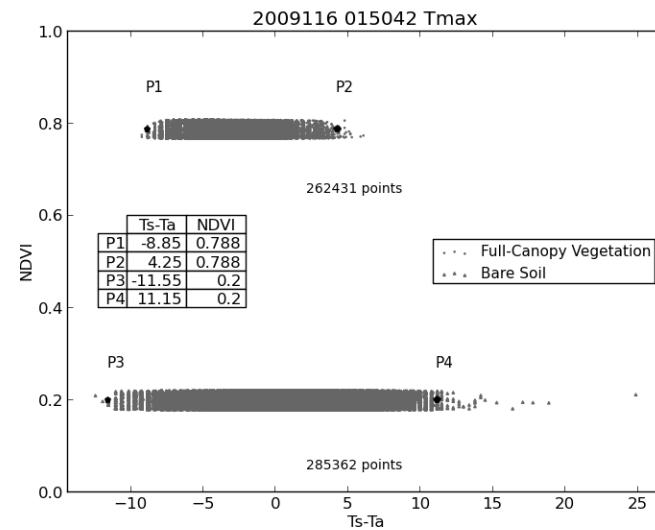
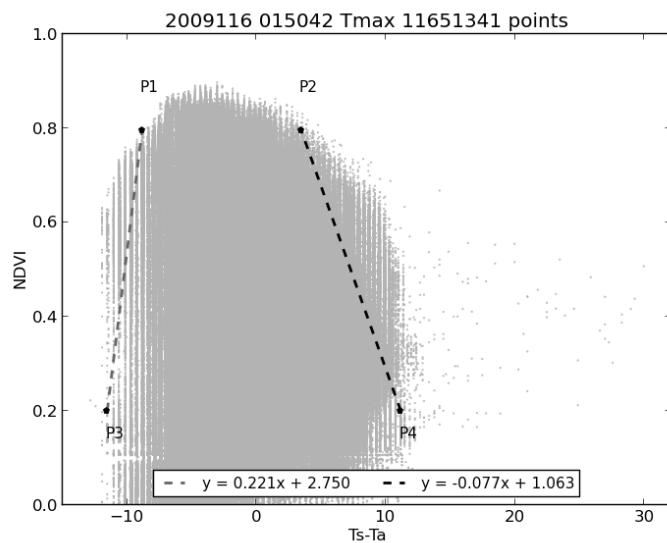
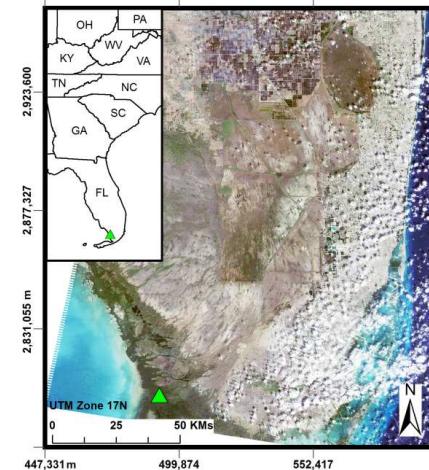
Evapotranspiration



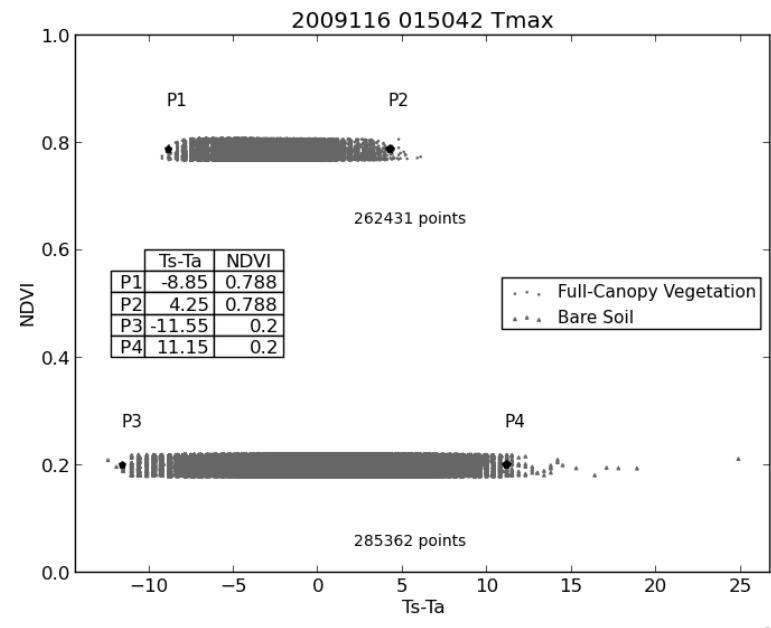
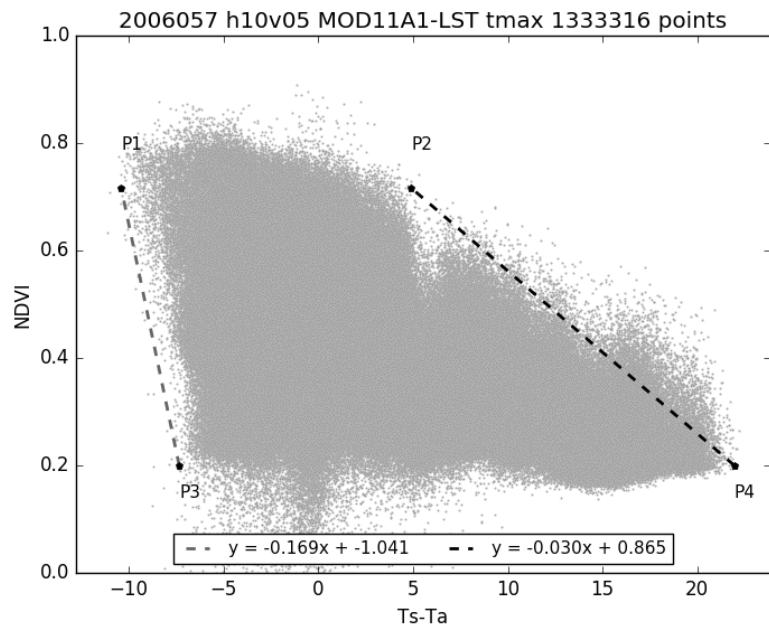
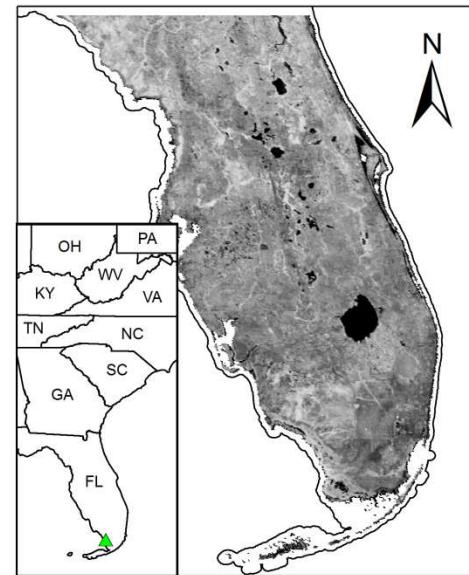
Extracting end-members from Landsat data



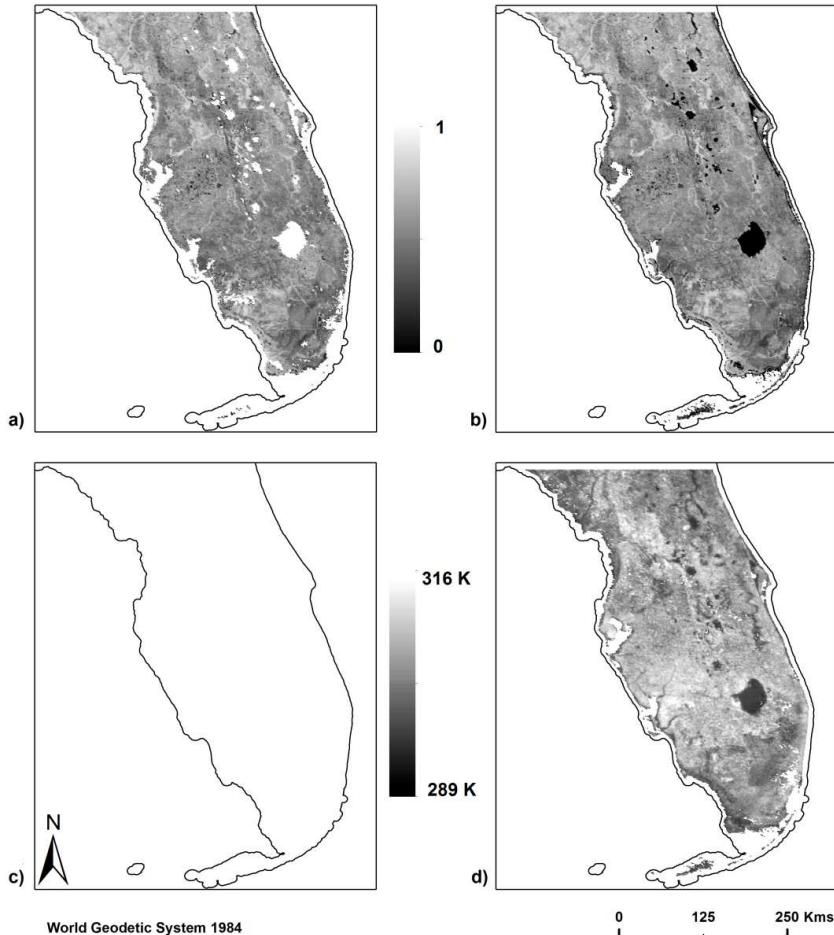
Flux tower located
at Everglades
National Park in
Florida, US



Extracting end-members from MODIS data

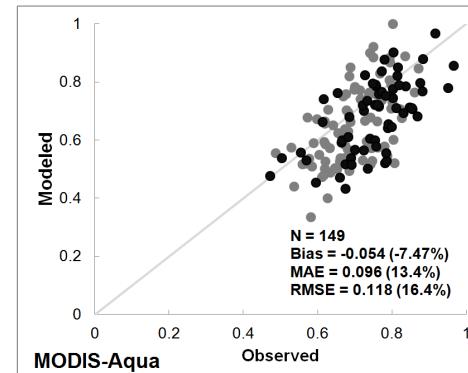
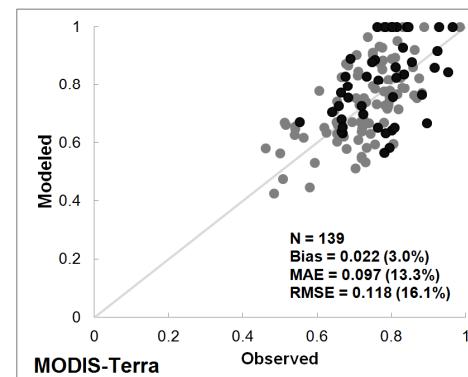
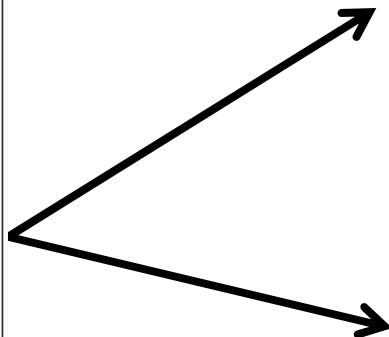
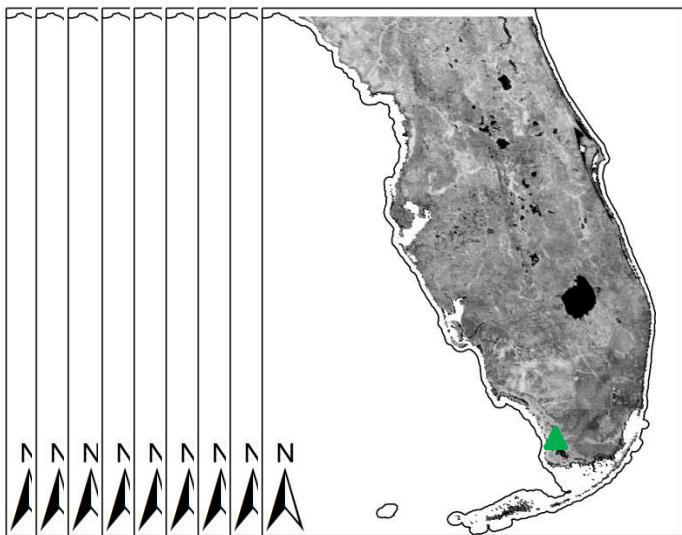
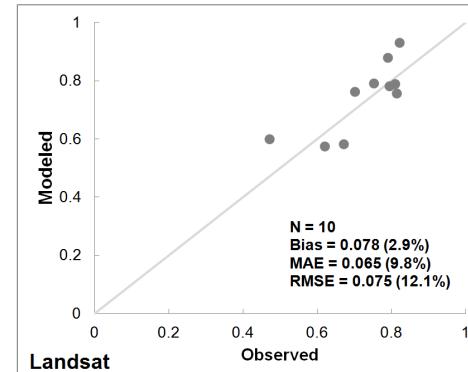
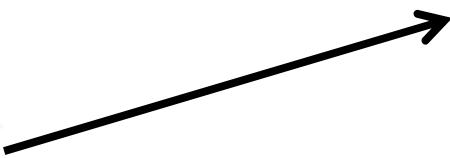
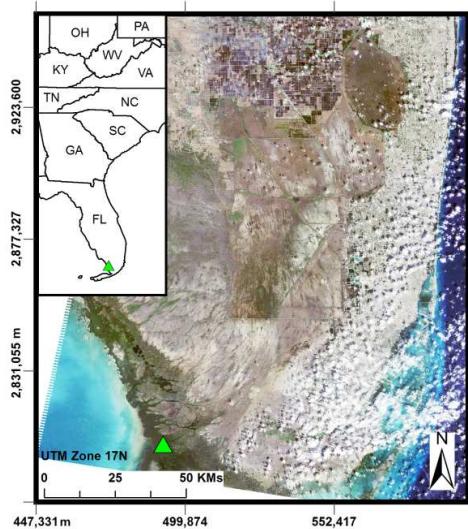


Interpolation

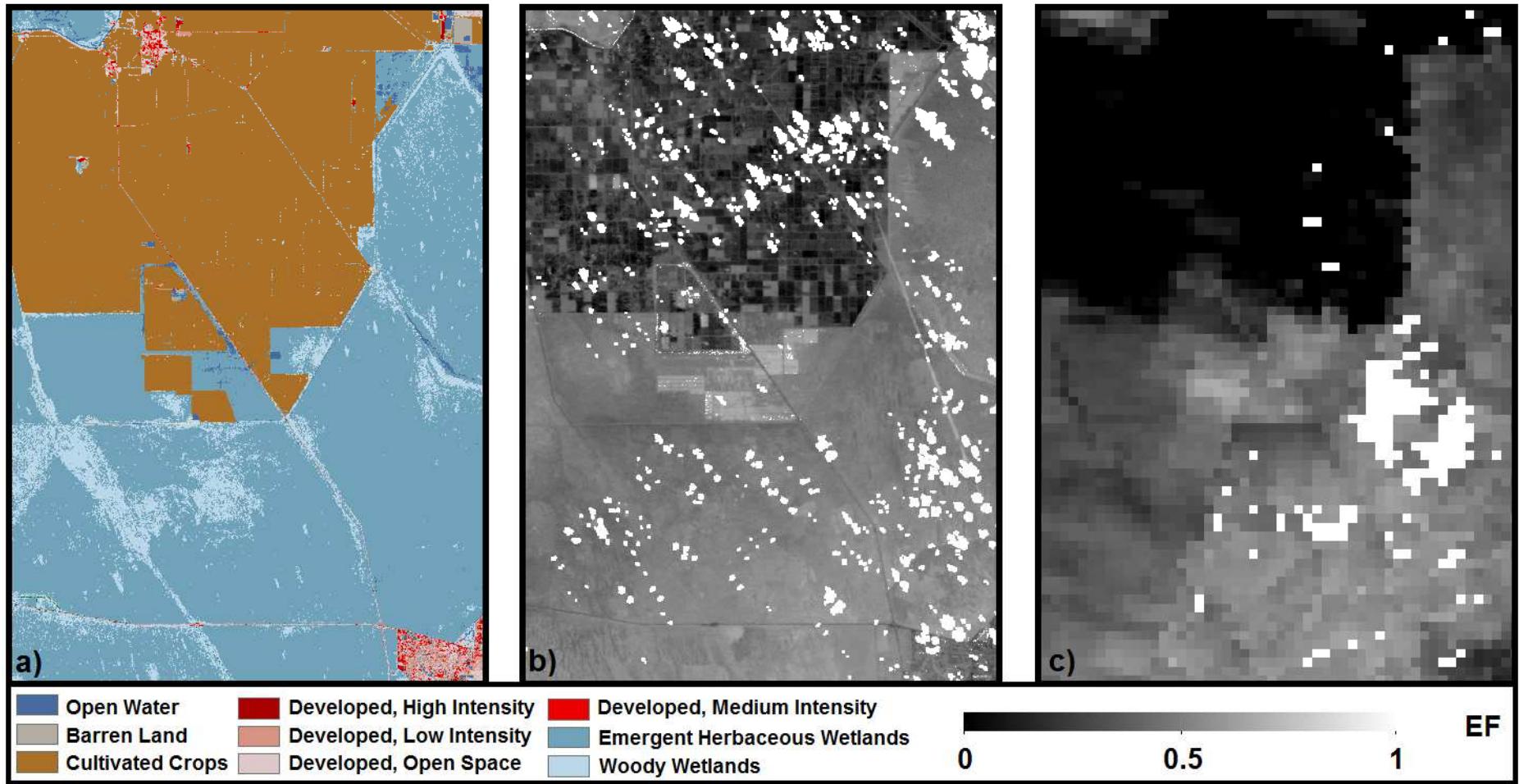


- NDVI gap-filling
 - 32 days before/after
 - Spatially auto-correlated
- LST gap-filling
 - 4 days before/after
- Analysis of Quality Bands
 - Cloudy pixels
 - Cloud adjacent pixels
 - Low-quality
 - High errors

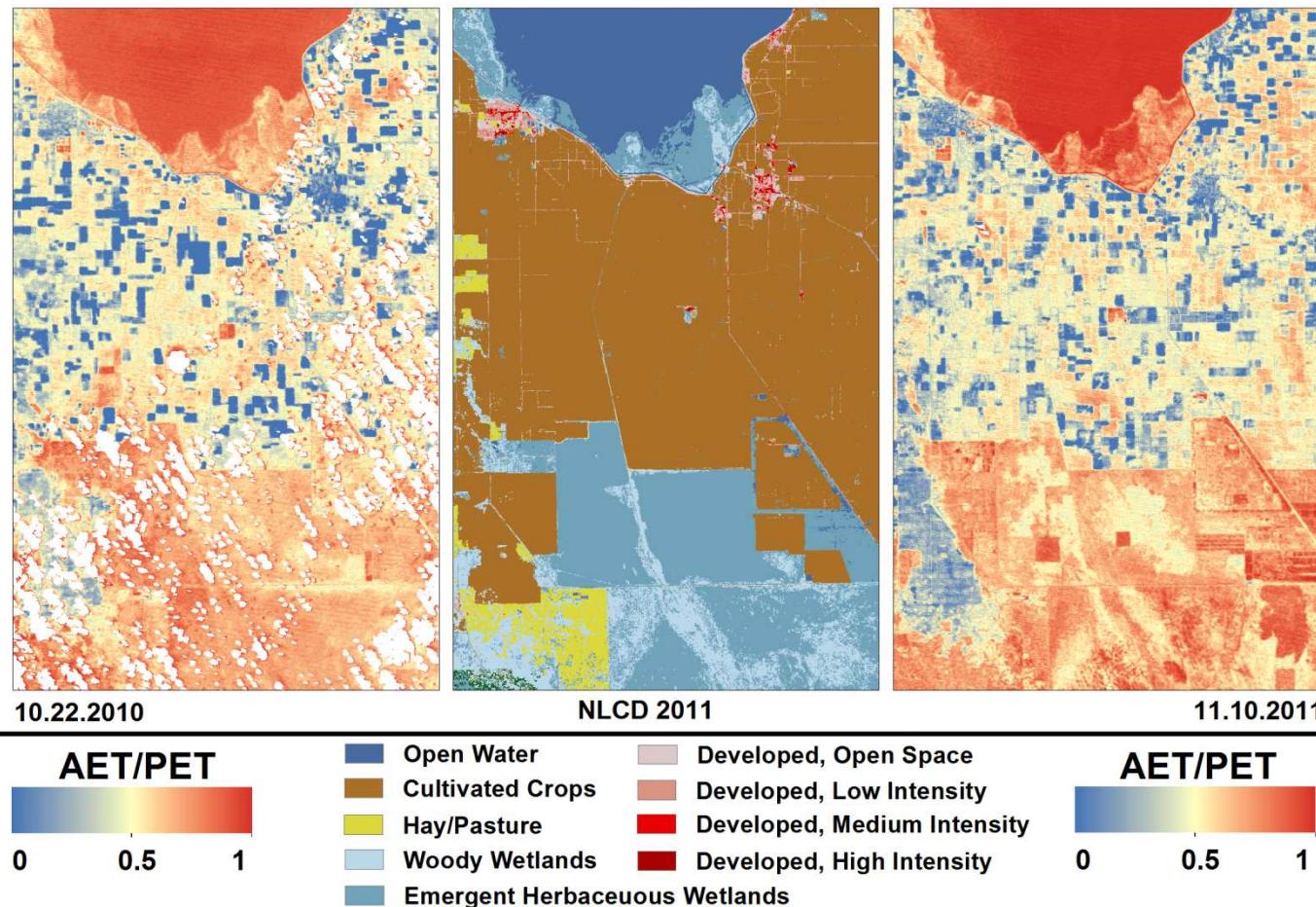
Validation of model inputs



ET maps of Everglades National Park



ET maps of Florida



Important Points

- Reusable functions
 - NDVI functions takes arrays as inputs
 - NDVI for MODIS, NDVI for Landsat or any sensor product
 - Handle which bands you need to pull
 - Drought index for a compositing period 171st day of 2012.
- Setting unused variables (e.g., arrays) to none to release memory.
 - ndvi=None

Questions?