MSPP

### Introduction

This document is a description of the process of preparing data from agMERRA for use as input in the MSPP tool and the process of interpolating data from the MSPP tool to provide a raster file output. The MSPP tool is an agronomy tool for calculating the approximate days it takes for maize to flower and mature.

MSPP is written in Visual Basic (VBA) programming language that is accessible in excel via the developer tab. The MSPP tool utilizes various mathematical models that model maize phenology based on the daily maximum, minimum and average temperature of a location, for several consecutive years between 1980 and 2010. The computation starts with the sowing date as day zero.

This document provides steps that should precede the execution of this tool - create data for the MSPP and follow the execution of this tool - interpolate the MSPP results. The implementation of these steps in R is simplified by the agMERREnc4ToCSV package that is provided alongside the MSPP tool.

### Section 1: Create data for the MSPP

The temperature data used in MSPP is available on the agMERRA website (link) in the netCDF format. The spatial extent of the data is the whole world. Each NetCDF file covers either the minimum, maximum or average temperature in one year over the period between 1980 and 2010. The first step is thus to download this data for all the years (or a required period) and have the data in three folders: minimum, maximum and average temperature. The file paths to these folders are then used to get the file paths of all the NetCDF files in them.

library(devtools)

## Loading required package: usethis

# Load "agMERRAnc4ToCSV" package  
path2package <- "../Rpackage/agMERRAnc4ToCSV"  
devtools::load\_all(path2package)

## Loading agMERRAnc4ToCSV

# Place inputs in the data folder  
netCDF\_TminFolder <- "../data/netCDF/AgMERRA\_tmin"  
netCDF\_TmaxFolder <- "../data/netCDF/AgMERRA\_tmax"  
netCDF\_TavgFolder <- "../data/netCDF/AgMERRA\_tavg"

# Get netCDf filepaths  
Tmin.nc4files <- agMERRAnc4ToCSV::nc4fullpaths(netCDF\_TminFolder)  
Tmax.nc4files <- agMERRAnc4ToCSV::nc4fullpaths(netCDF\_TmaxFolder)  
Tavg.nc4files <- agMERRAnc4ToCSV::nc4fullpaths(netCDF\_TavgFolder)

MSPP uses point data in its computation. To use it in a given set of location, you need to extract the data. You need to have the location data as a “SpatialPointDataFrame” object or in a CSV file, which can be converted to a “SpatialPointDataFrame” using the “wgsSpatialPointsDataFrame” function. This function assumes that the coordinates are in WGS84 if no coordinate system is provided.

# Provide a path to the locations CSV  
inputCSV <- "../data/tzaLocations.csv"  
  
# Create a SpatialPointsDataFrame Object from the CSV  
shpfle <- agMERRAnc4ToCSV::wgsSpatialPointsDataFrame(inputCSV,  
 LongitudeColumn = "Lon",   
 LatitudeColumn = "Lat")

Extracting data from the NetCDF files and providing the output in a data frame involves a few steps which are all covered in the “xtrctagMERRAnc4” function from “agMERRAnc4ToCSV” package. The “xtrctagMERRAnc4” function requires a vector of file paths to NetCDF files, a SpatialPointsDataFrame of locations to be extracted and the names of two columns in the SpatialPointsDataFrame with geographical names. It then returns a data frame of extracted values, which is organized in the format required in the MSPP tool.

The data should be saved in CSV files that can either be pasted in the MSPP tool or loaded via links to Excel sheets. The teps for connecting an Excel sheet to an CSV file is provided below - Appendix.

# Write to CSV  
dir.create("temperatureCSVs", showWarnings = FALSE)  
write.csv(Tmin, "temperatureCSVs/Tmin.csv", row.names = FALSE, col.names = FALSE)  
write.csv(Tmax, "temperatureCSVs/Tmax.csv", row.names = FALSE, col.names = FALSE)  
write.csv(Tavg, "temperatureCSVs/Tavg.csv", row.names = FALSE, col.names = FALSE)

### Section 2: Interpolating the results.

The MSPP tool’s output is an Excel workbook. The workbook has 3 worksheets. The first is a summary sheet, which gives the mean and standard deviation of the number of days between user-selected sowing dates of user-selected months (mean and standard deviation for all selected years) and the flowering dates, and between the flowering dates and the maturity dates. The second gives the sowing dates, in all selected months and all selected years, and their corresponding flowering dates while, the third provides the flowering dates, in all selected months and all selected years, and their corresponding maturity dates.

Interpolating the Mean values from the summary sheet can be done using the “interpolateMeanDays” function (agMERRAnc4ToCSV library). The function takes the coordinates of the locations and the mean values as input parameters and returns a raster of the interpolated values over an area covering the extent of the input coordinates.

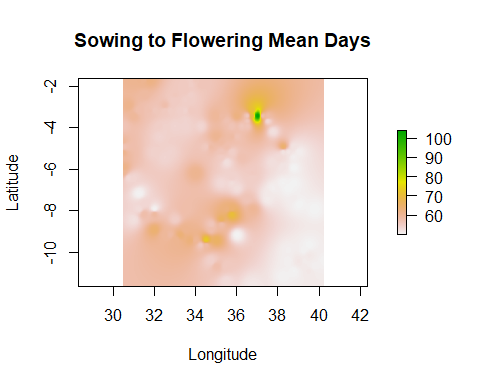
data.df <- xlsx::read.xlsx("../data/Briere\_3\_1980\_2009\_02\_01.xlsx", 1)  
  
Flowering.raster <- agMERRAnc4ToCSV::interpolateMeanDays(Lon = data.df$Lon,   
 Lat = data.df$Lat,   
 meandays = data.df$Flowering.Mean)

## [inverse distance weighted interpolation]

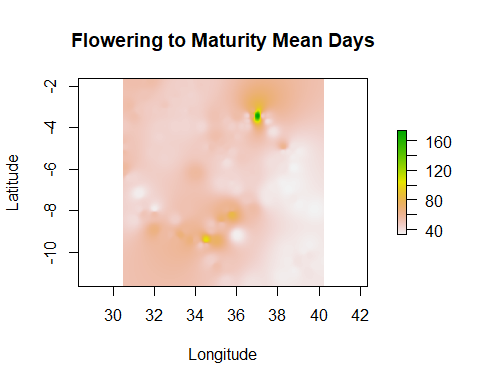
Maturity.raster <- agMERRAnc4ToCSV::interpolateMeanDays(Lon = data.df$Lon,   
 Lat = data.df$Lat,   
 meandays = data.df$Maturity.Mean)

## [inverse distance weighted interpolation]

plot(Flowering.raster,   
 main="Sowing to Flowering Mean Days",  
 xlab="Longitude",  
 ylab="Latitude")



plot(Maturity.raster,   
 main="Flowering to Maturity Mean Days",  
 xlab="Longitude",  
 ylab="Latitude")



## Appendix

### Appendix 1 Connecting an Excel sheet to the source CSV

Steps for updating/connecting an Excel sheets (Tmin. Tmax and Tavg) to the source CSV generated from the steps above. NB: The Sheets are protected wit “TAMASA” as the password \* 1. Create an empty Excel sheet. \* 2. Click on “From Text/CSV” ( “Data” tab > “Get & Transform Data” section > “From Text/CSV” icon). \* 3. A popup for selecting the CSV file will appear. Use it to select the CSV file. \* 4. This is followed by a popup for transforming the data before importing it. Use it make changes if any then click on load. \* 5. The CSV file will be loaded into the blank Excel sheet and a side panel of “Queries and connections” will popup. The panel provides options for refreshing the Excel sheet when the CSV changes. \* 6. Delete the Excel sheet and rename the new Excel sheet to replace it (for example, delete Tmin and rename new sheet Tmin to replace it).