# **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **ClimateAF v1.10**

# **A program to generate climate normal, annual, seasonal, and monthly climate data for historical and future periods in Africa**

Tongli Wang, Andreas Hamann, and Sarah Namiiro

*October 07, 2024*

*Contact:*

*tongli.wang@ubc.ca (software issues)*

*andreas.hamann@ualberta.ca* *(data issues)*

*namiiro@ualberta.ca* *(help with usage)*

# **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **About this program**

ClimateAF is a standalone MS Windows application written in Visual Basic 6.0. It extracts and downscales climate data from built-in climate grids to estimate 75 climate variables for historical time periods (months, years, decades, and 30-year normal periods from 1901 to 2020), and for the future periods 2020s (2011-2040), 2050s (2041-2070) and 2080s (2071-2100), based on CMIP6 multi-model projections.

Climate variables generated by the software include monthly base variables (mean minimum, maximum and average temperature as well as precipitation). In addition, economically or biologically relevant variables such as growing and chilling degree days, heating and cooling degree days, Hargrave's climate moisture deficit and reference evaporation, and seasonal variables.

The climate grids were developed with a deep neural network that uses geographic and atmospheric information to model local weather patterns at medium resolution (e.g. see Fig 1).

Subsequently, the ClimateAF software employs lapse-rate based dynamic downscaling to generate gridded data at high resolution, or to provide scale-free point estimates of climate variables for user-provided latitude, longitude and elevation coordinates.



**Figure 1**. Extent of the ClimateAF data coverage (– 18° to 60° longitude, – 35° to 38° latitude). The map shows mean annual precipitation, with the inset showing precipitation induced by orographic lift on the windward side, and rain shadows on the leeward side of major mountain ranges (Rift Valley, Mt Kenya, Mt Kilimanjaro area).

# **Climate variables predicted**

**1) Annual variables:**

*Directly calculated variables:*

MAT mean annual temperature (°C),

 MWMT mean warmest month temperature (°C),

 MCMT mean coldest month temperature (°C),

TD temperature difference between MWMT and MCMT, or continentality (°C),

 MAP mean annual precipitation (mm),

*Derived variables:*

 DD\_0 / DD\_below\_0 degree-days below 0°C, chilling degree-days

 DD5 / DD\_above\_5 degree-days above 5°C, growing degree-days

 DD\_18 / DD\_below\_18 degree-days below 18°C, heating degree-days

 DD18 / DD\_above\_18 degree-days above 18°C, cooling degree-days

 Eref Hargreaves reference evaporation (mm)

 CMD Hargreaves climatic moisture deficit (mm)

**2) Seasonal variables\***

*Three-month Seasons:*

DJF December-January-February

MAM March-April-May

JJA June-July-August

SON September-October-November

*Seasonally calculated variables:*

Tmin\* Average monthly minimum temperature (°C)

Tmax Average monthly maximum temperature (°C)

Tave Average monthly mean temperature (°C)

PPT / Prec Total precipitation (mm)

**3) Monthly variables\***

Tmin01 ...12 Average monthly minimum temperature (°C)

Tmax01 …12 Average monthly maximum temperature (°C)

Tave01 …12 Average monthly mean temperature (°C)

PPT / Prec01 …12 Total precipitation (mm)

\*) All monthly and seasonal variables are calculated as defined by the World Meteorological Organization (WMO). Temperature values are measured by minimum-maximum thermometers, and Tmin and Tmax for a month are the average of daily values for each month. Tave is the average of Tmin and Tmax for each month, and PPT is the total precipitation recorded in each month. Seasonal variables are calculated as the average of monthly variables for temperature and as the sum of monthly variables for precipitation.

# **Using the ClimateAF program**

No installation is required for the software. Simply extract the downloaded zip file into a new folder. This folder should contain two files (“ClimateAF\_v1.00.exe” and “Help.rtf”) and the three subfolders (“Baselinedat”, “Perioddat”, GCMdat”). Start the program by double clicking the ClimateAF\_v1.10.exe file.

### ***1) Use the program interactively***

* Latitude and longitude can be entered in either decimal degrees (e.g. Lat: 51.542, Long: 129.333) or degree, minute and second (e.g., 51°30’15”N, 129°15’30’W). Longitude information is accepted either in positive or negative values. Elevation has to be entered in meters, or empty if no elevation data available. If "Monthly variables", "Seasonal variables" or "All variables" output variables was selected, an additional output sheet appears and annual climate variables are still calculated.
* Output data can be saved as text file and imported to spreadsheet file using space-diliminated option.

## ***2) For multi-location process***

* Most users will have their sample data information in an Excel spreadsheet or in a text file. To make it possible for the program to read this data it must be prepared in Comma Separated Values (CSV) format, and decimal point must be a period (.) not a comma.
* Create a spreadsheet with the headers “ID1, ID2, lat, long, el” as shown in the example below. ID1 and ID2 can be “Location”, “Region” or whatever. It does not matter what variable names you choose, but the file **must** have the header row, and the variables **must** be in exactly the same order as shown below. If you don’t have elevation information or a second ID, you still need a variable header, you should put a period “.” in these columns. If you have more columns in your original file, you **must** remove them. Merge them back based on your IDs after the software is done with the data processing.



* Also note that longitude values are read in decimal degrees, and western hemisphere longitudes and southern hemisphere latitude values must be negative. If you have your coordinates in degees, minutes (and seconds), you can apply the formual: DD+MM/60 (+SS/3600) to convert to decimal degrees in Excel.
* After the spreadsheet is prepared as shown, save it as “comma delimited text file” by choosing “Save as …” form the file menu, and then specifying (\*.csv) from the “Save as type …” drop down menu.
* You can also directly create a comma delimited text file in any text editor such as Notepad. If there is a missing value, you need to enter a “.” between two commas.

 

* Save this text file with a .csv extension by writing out the full file name with extension in parenthesis when saving, e. g. “test.csv” instead of test.csv or test.
* Now you are ready for processing: Click on  to read your spreadsheet and on  to specify your output file folder and file. Then, click the  button. Climate variable information will be appended as additional columns to your input file. If elevation information is provided the climate variables will be elevation adjusted.

**R code to generate custom climate grids**

You can also generate your own custom climate grids with this handy R code:

* Create a custom input file from a DEM for your study area

library(terra)

vec1 <- vect("Your\_study\_area\_in\_WGS84- EPSG4326.shp")

ras1 <- rast("Your\_DEM\_in\_WGS84- EPSG4326.tif")

ras1 <- crop(ras1, ext(vec1))

ras1 <- mask(ras1, vec1)

plot(ras1)

plot(vec1, add=T)

writeRaster(ras1, "Custom\_DEM.asc")

* The next step requires the R library SDMTools. For this, install RTools.exe for your version of R: <https://cran.r-project.org/bin/windows/Rtools/>. Then run this code:

install.packages("remotes")

remotes::install\_github("cran/SDMTools")

* Now, you are good to go to easily create an input file with a few lines of code from your custom DEM created in the previous step:

library(SDMTools)

dat1 <- asc2dataframe("Custom\_DEM.asc")

dat1$ID1 <- row.names(dat1)

dat1$ID2 <- 0

names(dat1)[1:3] = c("LAT","LONG","ELEV")

dat1 <- dat1[,c(4:5,1:3)]

head(dat1)

write.csv(dat1, "ClimateAF\_input.csv", row.names=F, na="")

* Once the software has processed the input file representing your DEM, you can convert it back into climate grids like this:

dat1 <- read.csv("ClimateAF\_output\_Normal\_1961\_1990MSY.csv")

dat1[dat1 == -9999] <- NA

dat1 <- na.omit(dat1)

dat1 <- dat1[,-c(1:2)]

library(SDMTools)

dir.create("ASCII\_files")

dataframe2asc(dat1, out="ASCII\_files")

* Lastly, you can convert them back to TIFF files (or any other format) if you wish:

library(terra)

dir.create("TIFF\_files")

 files <- list.files("ASCII\_files", pattern="asc")

 for (file in files) {

 ras1 <- rast(paste0("ASCII\_files/", file))

 path <- file.path(paste0("TIFF\_files/",

 tools::file\_path\_sans\_ext(basename(file)), ".tif"))

 writeRaster(ras1, path, overwrite=T)

 }

**References**

Note that the ClimateAF package has not undergone peer-review in a journal yet. In the interim, reference usage like this: "*Climate data has been generated with the ClimateAF v1.10 softwarepackage, available at* [*http://tinyurl.com/ClimateAF*](http://tinyurl.com/ClimateAF)*, based on methodology similar to Mahoney et al. (2022) and Wang et al. (2016).*"

Mahoney, C.R., Wang, T., Hamann, A., Cannon, A.J. 2022. A CMIP6 ensemble for downscaled monthly climate normals over North America. *International Journal of Climatology* **42**: 5871-5891.

Wang, T., Hamann, A. Spittlehouse, D.L. and Carroll, C. 2016. Locally downscaled and spatially customizable climate data for historical and future periods for North America. *PLoS One* **11**: e0156720.