TAMSAT-ALERT v1: A new framework for agricultural decision support (Supplementary information)

Dagmawi Asfaw[1], Emily Black[1], Matt Brown[2], Kathryn Jane Nicklin[3], Frederick Out-Larbi[4], Ewan Pinnington[1], Andrew Challinor[3], Ross Maidment[1], Tristan Quaife[1]

[1]Department of Meteorology, University of Reading, Reading, UK
[2]Department of Atmospheric, Oceanic and Planetary Physics, University of Oxford, Oxford, UK
[3]School of Earth and Environment, University of Leeds, Leeds, UK
[4]Ghana Meteorological Agency, Accra, Ghana

TAMSAT-ALERT v1.0 USER MANNUAL

1. How to install TAMSAT-ALERT v1.0

TAMSAT-ALERT is comprised of a number of Python scripts. To install TAMSAT-ALERT, download the code in https://github.com/tamsat-alert/v1-0 from GitHub into a location where you have read and write permissions.

TAMSAT-ALERT requires a working installation of Python 2.7 with the following modules installed: numpy, datetime, os, matplotlib, scipy, statsmodels, seaborn, copy, glob, sys, warnings, shutil, math, string.

The instructions below assume that python can be called on a command line - i.e. that the user is using a unix or linux system/emulator. Alternatively, the scripts can be called from within a Python IDE for example by substituting execfile('prepare_driving_wrapper.py') rather than python prepare_driving_wrapper.py

2. How to run TAMSAT-ALERT

To maximize its modularity, TAMSAT-ALERT is divided into three sections, each of which is run separately. The user can choose to run all or part of the TAMSAT-ALERT framework.

2.1. Preparing driving data

The user should modify the config.py file for their environment (see below), change directory into the directory containing the TAMSAT-ALERT scripts and execute the following command:

```
python prepare_driving_wrapper.py
```

Config file: config.py

config.py is of the form: <variable> = <some_value> #Comment. The user should only edit <some_value>. Not all variables are relevant to all parts of the system. Irrelevant variables are disregarded.

The variables in config.py are as follows:

Variable	Explanation	Use
filename	full path to the ascii text file containing the driving daily time series (see below for format)	prepare_data_wrapper.py calc_cumulative_rainfall_wrapper.py forecast_metric_wrapper.py
leapremoved	1 if leap days have already been removed from the data in <filename> 0 if leap days have not already been removed from the data in <filename>	prepare_data_wrapper.py calc_cumulative_rainfall_wrapper.py forecast_metric_wrapper.py
datastartyear	year that the data in <filename> start	prepare_data_wrapper.py calc_cumulative_rainfall_wrapper.py forecast_metric_wrapper.py calcrisk_wrapper.py
dataendyear	year that the data in <filename> end	prepare_data_wrapper.py calc_cumulative_rainfall_wrapper.py forecast_metric_wrapper.py calcrisk_wrapper.py
init_year init_month init_day	year/month/day for which the forecast will be initiated (i.e. present day or hindcast equivalent)	prepare_data_wrapper.py calcrisk_wrapper.py
periodstart_year periodstart_month periodstart_day	year/month/day for which the period over which TAMSAT-ALERT will be run starts. This should be on or before init_year/month/day and should include sufficient time for spin up (if applicable)	prepare_data_wrapper.py calc_cumulative_rainfall_wrapper.py
periodend_year periodend_month periodend_day	year/month/day for which the period over which TAMSAT-ALERT will be run ends. This should be chosen to include the whole period over which the metric is calculated	prepare_data_wrapper.py calc_cumulative_rainfall_wrapper.py
climstartyear	start year for the climatology	prepare_data_wrapper.py calc_cumulative_rainfall_wrapper.py calcrisk_wrapper.py
climendyear	end year for the climatology	prepare_data_wrapper.py calc_cumulative_rainfall_wrapper.py calcrisk_wrapper.py
leapinit	1 if leap years should be included in the ensemble driving time series, prior to forecast initialization 0 if leap hears should not be included in the ensemble driving time series prior to the forecast initialization	prepare_data_wrapper.py

climatological_metric_file	full path to the file containing the climatological metric time series	prepare_data_wrapper.py calcrisk_wrapper.py
forecast_metric_file	full path to the file containing the weighting (meteorological forecast) metric time series	prepare_data_wrapper.py calcrisk_wrapper.py
ensemble_metric_file	full path to the file containing the weighting (meteorological forecast) metric values	prepare_data_wrapper.py calcrisk_wrapper.py

Input data files required

prepare_driving_wrapper.py requires one text file containing a historical daily time series of the meteorological variables required for calculation/modelling of their risk metric. The data must start on January 1st.

The file can have any number of columns, with one meteorological variable per column. The text file should not include a header row or date/time column.

The full path to the text file is assigned to the variable 'filename' in config.py

Output

The output is a zip file ensdriving.zip containing text files in the same format as the input data file, each of which contains driving data for the ensemble forecast. The data are provided from the periodstart_year/month/day to periodend_year/month/day (in config.py).

2.2. Deriving the risk metric

Examples for two different risk metrics are presented: cumulative rainfall and crop yield (simulated using GLAM). All of the code for running the cumulative rainfall scripts is available in this release. The GLAM yield assessments require a working installation of the GLAM crop model.

2.2.1 Cumulative rainfall

To derive the historical time series and the ensemble predictions for cumulative rainfall, run the command:

```
python calc_cumulativerainfall_wrapper.py
```

In addition to the relevant parts of config.py, the following inputs, specific to calc_cumulativerainfall_wrapper.py should be edited in config_cumrain.py

Variable	Explanation
intereststart_month	day/month of the start of the period over which the meteorological variable is to be cumulated/meaned
intereststart_day	
interestend_month	day/month of the end of the period over which the meteorological variable is to be cumulated/meaned
interestend_day	
intereststart_year	start/end year for which the meteorological variable is to be calculated. This is generally the same as the climatology, except where the year boundary is crossed.
interestend_year	
rainfallcolumn	column in the input array which holds the meteorological variable
calccumulative	1 if the data are to be cumulated over the period of interest 0 if the data are to be meaned over the period of interest

Input data files required

calc_cumulativerainfall_wrapper.py requires one text file of the same format as that required for prepare_driving_wrapper.py (see above). The path to the file is specified in config.py (as for prepare_driving_wrapper.py).

Output

The following files are output:

- a file containing a climatological time series of the metric. The full path is assigned to the variable climatological_metric_file in config.py. The output are of the format <year> <value> with a single value provided for each year in the range specified by intereststart_year and interestend_year
- a file containing the outputs of the ensemble forecasts. The full path is assigned to the variable ensemble_metric_file in config.py. The output is of the format <year> <value> with a single value provided for each ensemble member. The year represents the year in the climatology for which data after the initialization date are provided.

2.2.2 GLAM

To derive the historical time series and the ensemble predictions for crop yield and calculate the risk probabilities, run the command:

```
python calc_cropyield_wrapper.py
```

Run this module from Windows Power Shell or command prompt (cmd). Please make sure the glam.exe file should be compiled for Windows (using cygwin for windows) and if your system is Linux glam.exe should be compiled using Linux.

Procedure:

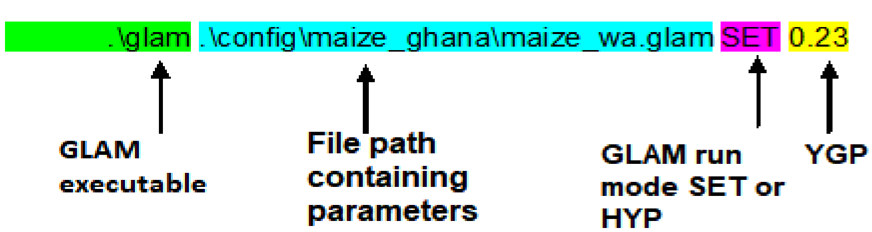
1. Download all the folder TAMSAT-ALERT-GLAM
2. Contact Andrew Challinor to get GLAM model for TAMSAT-ALERT through the following link. (http://www.see.leeds.ac.uk/research/icas/research-themes/climate-change-and-impacts/climate-impacts/glam/)
3. Compile GLAM and make sure you get the relevant parameters for your area. (Make some evaluation for the model and parameters used).
4. Put the glam.exe in the TAMSAT-ALERT-GLAM folder change the GLAM parameter files in the config folder.
5. Open the ReadVar.py and config.py scripts and change all the variables according to your requirement.
6. Go to command prompt (cmd) or Windows Power Shell and change your directory to the folder where the TAMSAT-ALERT-GLAM is found. (If you are using Linux system compile GLAM executable in the same OS and make sure the directories for the weather file and glam.exe are given properly in ReadVar.py)

7. Run the command " calc_cropyield_wrapper.py"

In addition to the relevant parts of config.py, the following inputs, specific to calc_cropyield_wrapper.py should be edited in ReadVar.py

Variable	Explanation
wth_path	path of the directory which will hold the processed meteorological inputs to GLAM
glam_command	command for running GLAM[1]
soil_tex	On this line the soil textural class is given as one of the 12 textural classes by the USDA (United Stated department of agriculture). The names are: 'clay', 'silty clay', 'sandy clay', 'silty clay loam', 'clay loam', 'sandy clay loam', 'loam', 'silt loam', 'sandy loam', 'silt', 'loamy sand', 'sand'. The names are case sensitive, so it should be written in the same way shown here.
lat	latitude of the location of interest
lon	longitude of the location of interest
weights	**Seasonal forecast [below average, average, above average].** This value represents the weather forecast probability value for below average, average and above average forecast given in fraction. This tercile probabilities are used to weight the metric we are forecasting according to the weighting metrics given.
weight_var	**Weighting variable.** This line use to switch between the weighting variable to be used 0 = rainfall sum and 1= average temperature.
wf_year:	**Weighting Forecast year.** This year represent the forecast year of the weighting variables
wf_month	**Weighting Forecast month.** This month represent the forecast month of the weighting variables.
wf_day	**Weighting Forecast day.** This day represent the forecast day of the weighting variables.
w_leadtime	**Lead time.** This represents the lead time of the weather forecast values to be used in the weighting of the ensembles it takes a value from 1 to 365. The weighting will be

[1] The GLAM command line to execute the model is given here it has a format as follows:



here only the file path and the Yield Gap Parameter (YGP) can be changed the GLAM run mode HYP is used if we want to determine the YGP first and we use SET to run the actual yield simulation. The file path show were the parameter files are saved.

	done by the sum of the rainfall starting from the weighting forecast date (obtained from wf_month and wf_day to the end of the lead time. E.g. if the forecast month is 6 (June) and the forecast date is 15, and the lead time is 90 days then the TAMSAT-ALERT will weight its forecast ensembles by the sum of rainfall from day 167 up to day 257 (167+90= 257) or the average temperature from similar days of the year.

Note that the GLAM implementation requires the following utility scripts (supplied as part of the release):

ReadVar.py

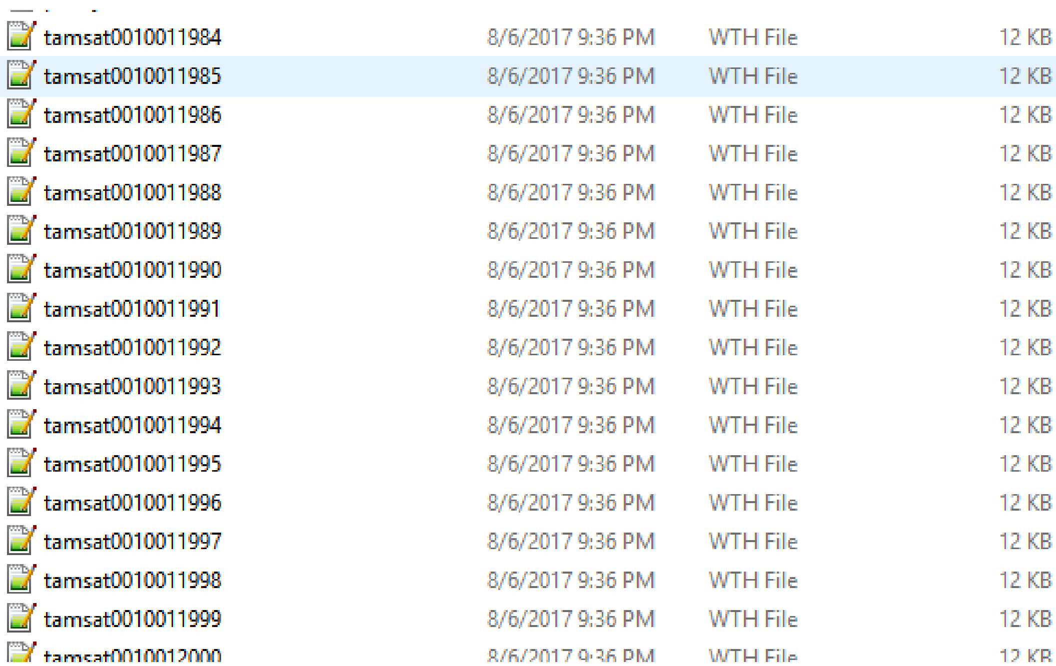
This script contains all the variables required to run the TAMSAT-ALERT-GLAM system for assessment of meteorological risk to crop yield. There are 24 variables required and all the changes should only be made within this script. Some of the variables are edited in the config.py file. Only the variables described in section 2.2.2 need to be edited from this script.

warning.py

This script checks the input variables are given properly. It checks for datatype and year orders among the data length and forecast period as well as the climatological years selected.

glam_data_prep.py

This script contains functions that will prepare the long-timeseries of weather data in the proper format for GLAM model run. GLAM requires the in a separate file for each year with specific headings and unit for the variables. This module of the TAMSAT-ALERT takes the long time series of the data from a text file and change each year's value into the required format and put the files in the path "./config/maize_ghana/ascii_input/wth/" this path must be given in the config_file.txt of the first step.



tamsat0010011984	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011985	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011986	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011987	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011988	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011989	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011990	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011991	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011992	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011993	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011994	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011995	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011996	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011997	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011998	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010011999	8/6/2017 9:36 PM	WTH File	12 KB
tamsat0010012000	8/6/2017 9:36 PM	WTH File	12 KB

each file has a name identifier, point identifier (lat,lon) and year. The name identifier 'tamsat' for example should be given in the config_file.txt as well as the glam parameter input file. This name and point identifiers can be changed but make sure the name is also changed in the parameter input file of GLAM.

ensem_glam_data_prep.py

This script is similar to that of the glam_data_prep.py but it is specifically work on the preparation of the ensemble data in GLAM format for the forecast year only. The weather driving data prepared by this script will be copied to the GLAM weather data path "./config/maize_ghana/ascii_input/wth/" every time a forecast is made by cropyield_est.py script.

hydraulic_params.py

This module contains pedotransfer functions that will provide the GLAM model with the soil water levels required (RLL, DUL, SAT) from a textural class given in the config_file.txt. The textural classes of the soil are according to the USDA classification and all names are given as shown below:

'clay', 'silty clay', 'sandy clay', 'silty clay loam', 'clay loam', 'sandy clay loam', 'loam', 'silt loam', 'sandy loam', 'silt', 'loamy sand', 'sand'

cropyield_est.py

This module contains the actual crop yield estimation using GLAM model. The TAMSAT-ALERT system where input variables are set from the forecast date onwards by climatological data sets and then prepare the changed file into a proper GLAM input file format. Then it will run the GLAM model and put the yield estimates (climatology and the ensemble predictions) in a text files for further plot and visualization.

weighting.py

weighting.py module contains the function for weighting preparing the weighting metric file. The weighting metric is based on the rainfall or temperature since both variables are given in the GLAM crop model as an input the file can be prepared according to the period specified in the configuration text file of start date for weighting and lead time. The user defines the weighing variable, weighting start date and lead time to be used in the "config_file.txt". The weightings are done by sum of forecast day plus lead time rainfall or average temperature of forecast day plus lead time. For example, if the forecast day is given as 15 and forecast month is 6 (June) and lead time of 90 days; the seasonal forecast weights are considered to be the values of June 15 + 90 days.

The weighting metric is prepared in the following format with a header, and two columns first one showing the climatological years and second one showing the weighting metric values.

WeightMetric.txt

```
# ClimaYears    WeightMetricValue
1980    26.06
1981    26.09
1982    25.98
1983    26.89
1984    26.13
1985    25.43
1986    25.66
1987    26.55
1988    25.85
1989    25.81
1990    25.98
1991    26.14
1992    25.37
1993    26.02
1994    26.29
1995    26.40
1996    26.10
1997    26.54
1998    26.36
1999    25.84
2000    26.12
2001    26.33
2002    26.42
2003    26.34
2004    25.95
2005    26.30
2006    26.61
2007    26.53
2008    26.17
2009    26.08
```

calcrisk.py

calcrisk.py module contain functions that will compare the yield estimate from GLAM with the yield of the climatology and visualize the final probabilistic estimation of climatic impact on yield for the specified location. The plots form this module are put in a folder called <plot_output> and the file outputs are in folder <data_output> please check it inside the statistical method used for the analysis (gaussian or ecdf).

GLAM parameter file

GLAM also requires the following calibration parameter files (examples included as part of the release):

GLAM–maize parameter values used for the simulation of maize yield in Ghana

Parameter	**Explanation**	**Value**	**Reference**
Leaves			
DLDTMX	Maximum daily increase in LAI	0.1	(Watson et al., 2015)
MASPA	Minimum daily increase in senesced LAI during leaf senescence.	0.03	(Bergamaschi et al., 2013)
SWF_THRESH	Critical value of soil water stress factor for leaves	0.7	(Bergamaschi et al., 2013)
NDSLA	Specific Leaf Area control	5 days	(Challinor and Wheeler, 2008)

SLA_INI	Specific leaf area control (see Challinor and Wheeler 2008)	350 $cm^2\ g^{-1}$	(Ashraf and Hafeez , 2004)
MAX_ISTG_SLA	Max development stage for SLA control	2 (until flowering)	
Evaporation and transpiration			
ALBEDO	Albedo	0.25	(Oguntunde and van de Giesen, 2004)
CRIT_LAI_T	LAI below which transpiration is physiologically limited	2.7	(Al-Kaisi et al., 1989; Bergamaschi et al., 2013)
P_TRANS_MAX	Max value of potential transpiration	0.73 cm day^{-1}	(Al-Kaisi et al., 1989)
VPD_CTE	Used to calculate vapour pressure deficit	0.7	(Tanner and Sinclair, 1983)
VPD_REF	Used to calculate Priestly–Taylor coefficient	1kPa	(Steiner et al., 1991)
SHF_CTE	Used to calculate soil heat flux	0.4	(Choudhury et al., 1987)
EXTC	Extinction coefficient	0.5	(Bergamaschi et al., 2013)
Soil and roots			
RLL	Lower limit of volumetric soil water ($m^3\ m^{-3}$)	0.083 $m^3\ m^{-3}$	
DUL	Drained upper limit of volumetric soil water ($m^3\ m^{-3}$)	0.195 $m^3\ m^{-3}$	
SAT	Saturated volumetric soil water ($m^3\ m^{-3}$)	0.409 $m^3\ m^{-3}$	
ASWS	Initial soil water (fraction of water holding capacity)	0	
NWBDAYS	Number of days water balance is run before start of planting window	30 days	
NSL	Number of soil layers	25	(Challinor et al., 2004)
ZSMAX	Maximum rooting depth	150 cm	
RKCTE	Used to calculate saturated hydraulic conductivity of the soil	75 cm day^{-1}	(Suleiman and Ritchie, 2001)
UPDIFC	Uptake diffusion coefficient	0.25 $cm^2\ day^{-1}$	(Jamieson and Ewert, 1999; Robertson and

			Fukai, 1994).
EFV	Extraction front velocity	2 cm day^{-1}	(Dardanelli et al., 1997; Bergamashi et al., 2013)
DLDLAI	Increase in root length density at surface with LAI	1 cm cm^{-3}	(Bergamaschi et al., 2013)
RLVEF	Root length density at the extraction front	0.3 cm cm^{-3}	(Watson et al., 2015)
Biomass and yield			
TE	Transpiration efficiency	7 Pa (1994–2006) 8 Pa (2007–2014)	(Walker, 1986; Adamtey et al., 2010)
TEN_MAX	Max transpiration efficiency	10 g kg^{-1}	(Walker, 1986; Adamtey et al., 2010)
RUE	Radiation Use Efficiency	3.4 g MJ^{-1}	(Kiniry et al., 1989; Lindquist et al. , 2005)
DHDT	Maximum daily increase in harvest index	0.011 day^{-1}	(Birch et al., 1999)
MAX_HI_MAIZE	Maximum harvest index for maize	0.6	(Hay and Gilbert, 2001)
Development			
FSWSOW	Fractional soil moisture for intelligent planting and emergence	0.5	(Challinor et al., 2004)
IEMDAY	Number of days from planting to emergence	6 days	(Ashagre et al., 2014)
I_TTCALC	Shape of function to calculate thermal time	1 (flat top)	(Keating et al., 1992)
TBMAI	Base temperature for maize	10 °C	(Sanchez et al., 2014)
TOMAI	Optimum temperature for maize	30 °C	(Sanchez et al., 2014)
TMMAI	Maximum temperature for maize	40 °C	(Sanchez et al., 2014)
I_PHEN	Complex (I_PHEN=0) or simple (I_PHEN=1) simulation of phenology	1	
TLIMFLW	Thermal time from emergence to silking	680 °C days	
TLIMPFL	Thermal time from silking to start of grain	145 °C days	(Tumbo et al., 2010)

	filling.		
TLIMGFP	Thermal time from start of grain filling to maturity	750 °C days	
Optional processes			
TETRS	Switch for temperature dependence of TE and RUE	1(on)	
TETR1	Temperature above which TE and RUE begin linear reduction	35 °C	(Yang et al., 2004; Carberry et al., 1989)
TETR2	Temperature above which TE and RUE are zero	47 °C	(Yang et al., 2004; Carberry et al., 1989)
TETR3	Temperature below which TE and RUE are zero	7 °C	(Yang et al., 2004; Carberry et al., 1989)
TETR4	Temperature below which TE and RUE begin linear reduction	18 °C	(Yang et al., 2004; Carberry et al., 1989)
HTS	Switch for high temperature stress around flowering	0(off)	(Gourdji et al., 2013)
TDS	Switch for terminal drought stress (TDS)	1(on)	
HIMIN	Minimum harvest index for TDS to occur	0.25	(Challinor et al., 2009)
SWC_FAC	Fraction of water holding capacity below which TDS occurs	0.05	(Challinor et al., 2009)
MIN_ISTG_TDS	Minimum ISTG for TDS to occur.	4 (start of grain filling)	
NMAXTDS	Number of days of TDS needed for the crop to be harvested.	8 days	(Jones and Thornton, 2003)
IEMER	Switch for intelligent emergence	–1 (off)	
TRKILL	Switch for lethal temperature parameterisation	–1 (off)	
WS	Switch for water stress around flowering	0 (off)	
TRLAI	Switch for temperature dependence of LAI growth	1 (on)	

TRLAIB	Base temperature below which leaf growth does not occur	7.3 °C	(Sanchez et al., 2014)
TRLAIO	Optimum temperature for leaf growth	31.1 °C	(Sanchez et al., 2014)
TRLAIM	Maximum temperature for leaf growth	41.3 °C	(Sanchez et al., 2014)

2.3. Deriving the weighting metric

As an example, a script for deriving a weighting metric based on the meteorological driving data is provided in the TAMSAT-ALERT v1.0 release. The script is run using the command:

python forecast_metric_wrapper.py

In addition to the relevant parts of config.py, the following inputs, specific to forecast_metric_wrapper.py should be edited in config_forecastmetric.py:

Variable	Explanation
intereststart_month intereststart_day	day/month of the start of the period over which the meteorological variable is to be cumulated/meaned
interestend_month interestend_day	day/month of the end of the period over which the meteorological variable is to be cumulated/meaned
intereststart_year interestend_year	start/end year for which the meteorological variable is to be calculated. This is generally the same as the climatology, except where the year boundary is crossed.
column	column in the input array which holds the meteorological variable
calccumulative	1 if the data are to be cumulated over the period of interest 0 if the data are to be meaned over the period of interest

Note that the values assigned can be different to those assigned in config_cumrain.py.

Input data files required

forecast_metric_wrapper.py requires one text file of the same format as that required for prepare_driving_wrapper.py (see above). The path to the file is specified in config.py (as for prepare_driving_wrapper.py).

Output

The following file is output:

- a file containing a climatological time series of the weighting (meteorological forecast) metric. The full path is assigned to the variable forecast_metric_file in config.py. The output are of the format <year> <value> with a single value provided for each year in the range specified by intereststart_year and interestend_year

2.4. Carrying out the risk assessment

The user should modify the config.py file for their environment (see below), change directory into the directory containing the TAMSAT-ALERT scripts and execute the following command:

python calcrisk_wrapper.py

In addition to modifying relevant variables in the config.py file, the following variables specific to calcrisk_wrapper.py should be set in config_calcrisk.py:

Variable	Explanation
stat	This should be set to 'normal' if the quintile risk assessments are calculated assuming a Gaussian distribution, and to 'ecdf' if the quintile risk assessments are calculated using an empirical cumulative distribution function.
sta_name	station name should
weights	This should be the weights of the probabilistic forecast categories in square brackets, separated by commas. For example [1,0,0] indicates a probability of 1 for the lowest tercile and 0 for each of the upper terciles; [0.1,0.2,0.4,0.3,0] indicates a quintile forecast with the probabilities 0.1,0.2,0.4,0.3,0 for the lowest to the highest quintiles.

Input data files required

The following three input files are required:

- a file containing a climatological time series of the metric. The full path is assigned to the variable climatological_metric_file in config.py. The output are of the format <year> <value> with a single value provided for each year in the range specified by intereststart_year and interestend_year. The file should have a one line header.
- a file containing the outputs of the ensemble forecasts. The full path is assigned to the variable ensemble_metric_file in config.py. The output is of the format <year> <value> with a single value provided for each ensemble member. The year represents the year in the climatology for which data after the initialization date are provided. The file should have a one line header.
- a file containing a climatological time series of the weighting (meteorological forecast) metric. The full path is assigned to the variable forecast_metric_file in config.py. The output are of the format <year> <value> with a single value provided for each year in the range specified by intereststart_year and interestend_year. The file should have a one line header.

Note that these input files can be output by forecast_metric_wrapper.py and calc_cumrafainfall_wrapper.py (see sections 2.2 and 2.4).

Output

The following is output:

- ./data_output/RiskProbability.txt an ascii text file containing probabilities of each climatological quintile

- plots in either ./plot_output/gaussian or plot_output/ecdf:
    - a histogram of the ensemble predictions (<station>_<init_date>_hist_plot.png) compared to the climatology. This plot also shows the modelled distributions (climatological, weighted and unweighted) [N.B.Modelled distribution plotting only supports Gaussian option in this release]
    - a kernel density plot of the ensemble predictions and the climatology (<station>_< init_date>_ked_plot.png). If the ensemble climatology is small this plot should be disregarded.
    - a qq plot of for the climatology and the ensemble (<station>_<init_date>_yieldprob.png)
    - a plot showing the quintile probabilities (<station>_<init_date>_pentile.png)

3. Test case

After all the files in the github repository are downloaded, the script cumrain_example.sh should be run to check that the TAMSAT-ALERT code is working properly. The expected output figures and data are included in the directory testcase_results.

References

Adamtey, N., Cofie, O., Ofosu-Budu, K. G., Ofosu-Anim, J., Laryea, K. B., and Forster, D.: Effect of N-enriched co-compost on transpiration efficiency and water-use efficiency of maize (Zea mays L.) under controlled irrigation, Agricultural Water Management, 97, 995–1005, doi:10.1016/j.agwat.2010.02.004, 2010.

Al-Kaisi, M., Brun, L. J., and Enz, J. W.: Transpiration and evapotranspiration from maize as related to leaf area index, Agricultural and Forest Meteorology, 48, 111–116, doi:10.1016/0168-1923(89)90010-5, 1989.

Ashagre, H., Hamza, I. A., Tesfaye, B., Derebachew, D., and Ayana, B.: Emergence and Seedling Growth of Corn (Zea mays L .) as Influenced by Irrigation Schedules on Vertisol, International Research Journal of Plant Science, 5, 17–22,doi:http:/dx.doi.org/10.14303/irjps.2013.061, 2014.

Ashraf, M. and Hafeez, M.: Thermotolerance of pearl millet and maize at early growth stages: Growth and nutrient relations, Biologia Plantarum, 48, 81–86, doi:10.1023/B:BIOP.0000024279.44013.61, 2004.

Bergamaschi, H., da Costa, S. M. S., Wheeler, T. R., and Challinor, A. J.: Simulating maize yield in sub-tropical conditions of southern Brazil using Glam model, Pesquisa Agropecuaria Brasileira, 48, 132–140, doi:10.1590/S0100-204X2013000200002, 2013.

Birch, C. J., Hammer, G. L., and Rickert, K. G.: Dry matter accumulation and distribution in five cultivars of maize (Zea mays): Relationships and procedures for use in crop modelling, Australian Journal of Agricultural Research, 50,513–527, doi:10.17700/jai.2015.6.1, 1999.

Carberry, P. S., Muchow, R. C., and McCown, R. L.: Testing the CERES-Maize simulation model in a semi-arid tropical environment, Field Crops Research, 20, 297–315, doi:10.1016/0378-4290(89)90072-5, 1989.

Challinor, A. J. and Wheeler, T. R.: Use of a crop model ensemble to quantify CO2 stimulation of water-stressed and well watered crops, Agricultural and Forest Meteorology, 148, 1062–1077, doi:10.1016/j.agrformet.2008.02.006, 2008.

Challinor, A. J., Wheeler, T. R., Craufurd, P. Q., Slingo, J. M., and Grimes, D. I. F.: Design and optimisation of a large-area process-based model for annual crops, Agricultural and Forest Meteorology, 124, 99–120, doi:10.1016/j.agrformet.2004.01.002, 2004.

Challinor, A. J., Wheeler, T., Hemming, D., and Upadhyaya, H. D.: Ensemble yield simulations: Crop and climate uncertainties, sensitivity to temperature and genotypic adaptation to climate change, Climate Research, 38, 117–127,doi:10.3354/cr00779, 2009.

Choudhury, B. J., Idso, S. B., and Reginato, R. J.: Analysis of an empirical model for soil heat flux under a growing wheat crop for estimating evaporation by an infrared-temperature based energy balance equation, Agricultural and Forest Meteorology, 39, 283–297, doi:10.1016/0168-1923(87)90021-9, 1987.

Dardanelli, J. L., Bachmeier, O. A., Sereno, R., and Gil, R.: Rooting depth and soil water extraction patterns of different crops in a silty loam haplustoll, Field Crops Research, 54, 29–38, doi:10.1016/S0378-4290(97)00017-8, 1997.

Gourdji, S. M., Sibley, A. M., and Lobell, D. B.: Global crop exposure to critical high temperatures in the reproductive period: Historical trends and future projections, Environmental Research Letters, 8, 0–10, doi:10.1088/1748-9326/8/2/024041, 2013.

Hay, R. and Gilbert, R.: Variation in the harvest index of tropical maize: evaluation of recent evidence from Mexico and Malawi, Annals of Applied Biology, 138, 103–109, doi:10.1111/j.1744-7348.2001.tb00090.x, 2001.

Jamieson, P. D. and Ewert, F.: The role of roots in controlling soil water extraction during drought: An analysis by simulation, Field Crops Research, 60, 267–280, doi:10.1016/S0378-4290(98)00144-0, 1999.

Jones, P. G. and Thornton, P. K.: The potential impacts of climate change on maize production in Africa and Latin America in 2055, Global Environmental Change, 13, 51–59, doi:10.1016/S0959-3780(02)00090-0, 2003.

Keating, B. A., Wafula, B. M., and Watiki, J. M.: Development of a modelling capability for maize in semi-arid eastern Kenya, in: A Search for Strategies for Sustainable Dryland Cropping in Semi-arid Eastern Kenya, edited by Probert, M. E., 41, pp. 26–32, ACIAR Proceedings No 41, Nairobi, Kenya, URL http://aciar.gov.au/files/node/2274/sustainable{_}dryland{_}cropping{_}in{_}semi{_}arid{_}eastern{_}{_}110 9.pdf{\#}page=11, 1992.

Kiniry, J. R., Jones, C. A., O'toole, J. C., Blanchet, R., Cabelguenne, M., and Spanel, D. A.: Radiation-use efficiency in biomass accumulation prior to grain-filling for five grain-crop species, Field Crops Research, 20, 51–64, doi:10.1016/0378-4290(89)90023-3, 1989.

Lindquist, J. L., Arkebauer, T. J., Walters, D. T., Cassman, K. G., and Dobermann, A.: Maize Radiation Use Efficiency under Optimal Growth Conditions, Agronomy Journal, 97, 72–78, 2005.

Oguntunde, P. G. and Van De Giesen, N.: Crop growth and development effects on surface albedo for maize and cowpea fields in Ghana, West Africa, International Journal of Biometeorology, 49, 106–112, doi:10.1007/s00484-004-0216-4,2004.

Robertson, M. J. and Fukai, S.: Comparison of water extraction models for grain sorghum under continuous soil drying, Field Crops Research, 36, 145–160, doi:10.1016/0378-4290(94)90063-9, 1994.

Sanchez, B., Rasmussen, A., and Porter, J. R.: Temperatures and the growth and development of maize and rice: A review, Global Change Biology, 20, 408–417, doi:10.1111/gcb.12389, 2014.

Steiner, J., Howell, T., and Schneider, A.: Lysimetric evaluation of daily potential evapotranspiration models for grain sorghum, Agronomy Journal, 83, 240–247, doi:10.2134/agronj1991.00021962008300010055x, 1991.

Suleiman, a. and Ritchie, J. T.: Estimating saturated hydraulic conductivity from soil porosity, Transactions of the ASAE,44, 235–239, doi:10.13031/2013.4683, 2001.

Tanner, C. B. and Sinclair, T. R.: Efficient Water Use in Crop Production: Research or Re-Search?, in: Limitations to Efficient Water Use in Crop Production, edited by Tanner, C. and Sinclair, T. R., pp. 1–27, American Society of Agronomy, Crop Science Society of America and Soil Science Society of America, Madison, Wisconsin, 1983.

Tumbo, S. D., Rwehumbiza, F. B., Kahimba, F. C., Enfors, E., and Henry, F.: Agronomic Management Strategies for Adaptation to the Current Climate Variability: The Case of North-Eastern Tanzania, http://hdl.handle.net/10625/46075, 2010.

Walker, G. K.: Transporatation efficiency of field-grown maize, Field Crops Research, 14, 29–38, doi:10.1016/0378-4290(86)90044-4, 1986.

Watson, J., Challinor, A. J., Fricker, T. E., and Ferro, C. A.: Comparing the effects of calibration and climate errors on a statistical crop model and a process-based crop model, Climatic Change, 132, 93–109, doi:10.1007/s10584-014-1264-3, 2015.

Yang, H. S., Dobermann, A., Lindquist, J. L., Walters, D. T., Arkebauer, T. J., and Cassman, K. G.: Hybrid-maize - A maize simulation model that combines two crop modeling approaches, Field Crops Research, 87, 131–154, doi:10.1016/j.fcr.2003.10.003, 2004.