Tool for Optimum Planting Window Selection

NG-Akilimo-SAA

Manual for Executing and Managing the Decision Support Tool for Variety Selection and Planting Window Recommendation

Co-authors: Eduardo Garcia Bendito[1], Helen Peter[2], Abdullahi Ibrahim Tofa[2], Kamaluddin Tijjani[2], Alpha Kamara[2], Christine Kreye[3]

[1]International Institute of Tropical Agriculture, Nairobi, Kenya
[2]International Institute of Tropical Agriculture, Kano, Nigeria
[3]International Institute for Tropical Agriculture, Oyo Road, Ibadan, Nigeria

Summary

Seasonal variability is a major constraint for many farmers cultivating maize under rainfed conditions in Northern Nigeria. Climate change is resulting in more erratic seasonal weather patterns and limiting the ability of farmers to rely on previous knowledge of the farming system where they operate. The Variety and Planting Window Recommendation Decision Support Tool (DST) for the EiA SAA - Nigeria use case aims to provide farmers with spatially explicit recommendations on optimal variety and planting window combinations. The recommendations are based on DSSAT v47 using a spatialization framework and consuming data from iSDA soil, CHIRPSv2 and ECMWF AgERA5. Additionally, calibrated cultivar coefficients are used to reproduce the local behavior of maize varieties in the target area. The tool is expected to be updated every cropping season (year), to account for the previous year's conditions on the recommendations.

1. Introduction......1
2. Description of the system......1
2.1 Overview......1
2.2 Environmental data......2
2.3 Crop model (DSSAT)......3
2.4 Aggregated outputs......3
3. CG Labs and computation requirements......3
3.1 CG Labs......3
3.2 SAA - Nigeria use case space......4
3.3 R routines......5
3.4 DSSAT specifics......5
4. Data......6
4.1 Inputs......6
4.1.1 Main......7
4.1.2 DSSAT......7
4.1.3 User......8
4.2 Intermediate......8
4.3 Outputs......9
5. Functions......10
5.1 Download data......10
5.2 Extract, Transform and Load......10
5.3 DSSAT......11
5.4 Aggregation......13
6. Executing the DST......14

1. Introduction

Most farmers rely on expertise and local knowledge to make most of their agronomic decisions. Access to data and analytical tools represent a viable solution to support farmer's decision-making around tactical investments such as choosing the appropriate crop variety and determining the optimal planting time. This access can play a crucial role in facilitating adaptation and mitigating uncertainties related to climate change. Maize is a staple food in most of Nigeria, primarily in the maize belt and is particularly sensitive to extreme climate conditions, impacting food security and critical for the livelihoods of millions of farmers in the region. Choosing specific crop varieties and determining optimal sowing dates are cost-effective strategies that directly enhance productivity. Advising farmers on these two components of the cropping cycle helps them to optimize their investments and reduce risks for the cropping season.

Decision Support Systems (DSS) help making organizational decisions based on scientific knowledge, and can be aided by computational processes. In agronomy, some of these DSSs are based on crop simulation models (CSM). These are dynamic process-based modeling tools which describe crop growth as a function of several environmental and management conditions. The Decision Support System for Agrotechnology Transfer (DSSAT) uses a modular approach to describe and simulate agricultural crop growth. It has been extensively used in many applications, including climate adaptation, and can be implemented in geospatial software, which provides spatially explicit crop growth predictions, with some limitations.

In the context of this DST, variety and planting windows are the focus model parameters, and are defined as the values (variety and planting dates) which maximize the yield, since this is the most important parameter for farmers and the one which determines most agronomic decisions. This tool therefore provides the final results as a combination of optimal variety and planting times which maximize the yield.

2. Description of the system

2.1 Overview

In general terms, the DST is composed of 3 main components: inputs, DSSAT and aggregations. Figure 1 presents the general overview with the 3 main components as well as the different elements and categories in the system. These 3 components are sequential, and must be executed in the appropriate order (A, B, C) to produce the expected results. Component A refers to the data input requirements to execute the process, B is related to the formatting, set-up and execution of DSSAT and C generates the final outputs with the recommendations. In this DST, external as well as internal data is used along the process, and only essential data will be persisted in the system. The entire process is publicly accessible on GitHub for partners or anyone else.

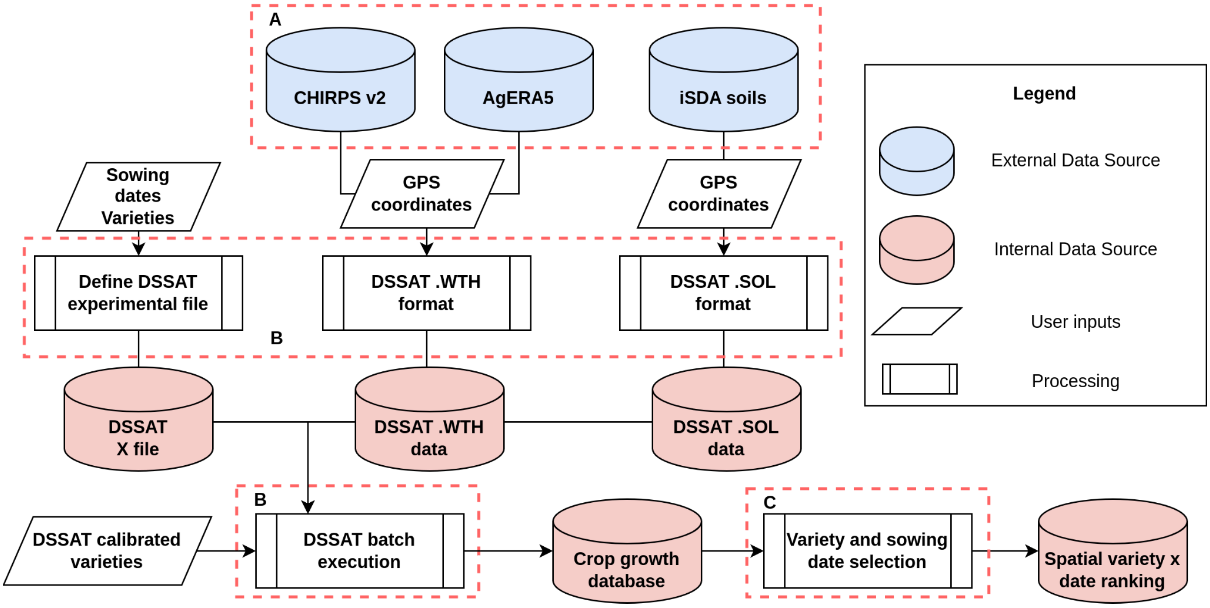

Figure 1: General overview of the variety and planting window advice DST. Component A refers to the external data sources for the environmental data. Component B contains DSSAT related processes. Component C is the final step dealing with the aggregation and formatting of the final data outputs.

2.2 Environmental data

The external data sources into the processing chain (fig. 1) refer to environmental data. This information is necessary in order to be able to execute the DST and particularly the DSSAT engine. The component A in figure 1 reflects these data sources, mainly for historical weather characterization and the soil conditions. The weather data is obtained from two external data sources: ECMWF AgERA5 (Boogard et al., 2020) and CHIRPSv2 (Funk et al., 2014). Variables such as minimum, maximum and mean daily temperature; wind speed, solar radiation and relative humidity are obtained from AgERA5. Rainfall is sourced from CHIRPSv2, since it is considered more accurate in the African context. Both of these datasets are combined into one to be loaded into DSSAT (see 6.2 Extract, Transform and Load and 6.3 DSSAT). These sources can be replaced for others depending on the needs of the user.

Soil data is extracted from iSDA soil data (Hengl et al., 2020), including all the variables available as well as the fertility capability classification based on Sanchez et al., 2013. Soil carbon, nitrogen, as well as physical characteristics of the soil are provided as per DSSAT requirements. Same as weather data, this data could be substituted by other sources depending on needs. The intermediate DSSAT files for environment (.WTH and .SOL) are also stored for user reference.

2.3 Crop model (DSSAT)

There are specific data requirements for the DSSAT software. These are generally included in the default DSSAT installation and include ecotype files (.ECO), variety coefficient files (.CUL), but, in order to fit the tool for the use case, there are some additional data that needs to be included in the .CUL file concerning variety specific coefficients. This specific information is added to the DSSAT executable directory, and the process is explained in 6.3 DSSAT.

2.4 Aggregated outputs

The outputs of the DST are the final look-up tables containing the relevant information in order to create a searchable index by location, and provide the variety and planting window recommendations to the user. The information included in the final outputs are lookup_key, var, pdate, rank.

3. CG Labs and computation requirements

3.1 CG Labs

The Collaborative GARDIAN Labs (CG Labs) is an collaborative data science platform that allows researchers to work together on the same data science project. It is maintained by Excellence in Agronomy in collaboration with SCiO partners. The platform is accessible from https://eia.scio.services:18002 and requires sign-up. Once the user has access to CG Labs it will be required to select an environment from the list of options. We encourage users to select the *Data science environment (Low)* server option indicated in figure 2, unless they are going to execute a large scale simulation and they require a high computing environment.

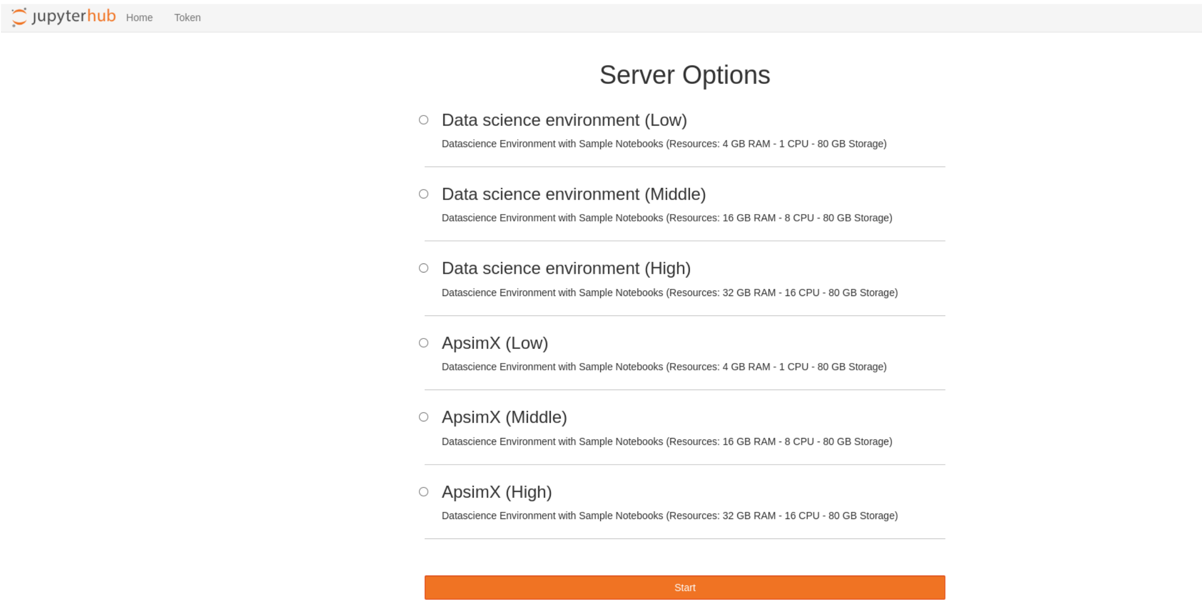

Figure 2: CG Labs Hub Control with the several server options.

The computation environment is based on a JupyterHub and includes R, Python and Julia software for data science projects. It provides access to a Linux terminal for operating system interactions. For most CGIAR users, the most interesting feature is probably the RStudio integration, where users can interact with their preferred IDE. Figure 3 describes the default panels of RStudio. Additionally to the software, CG Labs provides access to enough storage to run the necessary processes, plus the necessary data to execute the DST (see [3.2 Environmental data](#)). The system is kept on a backup to minimize data losses.

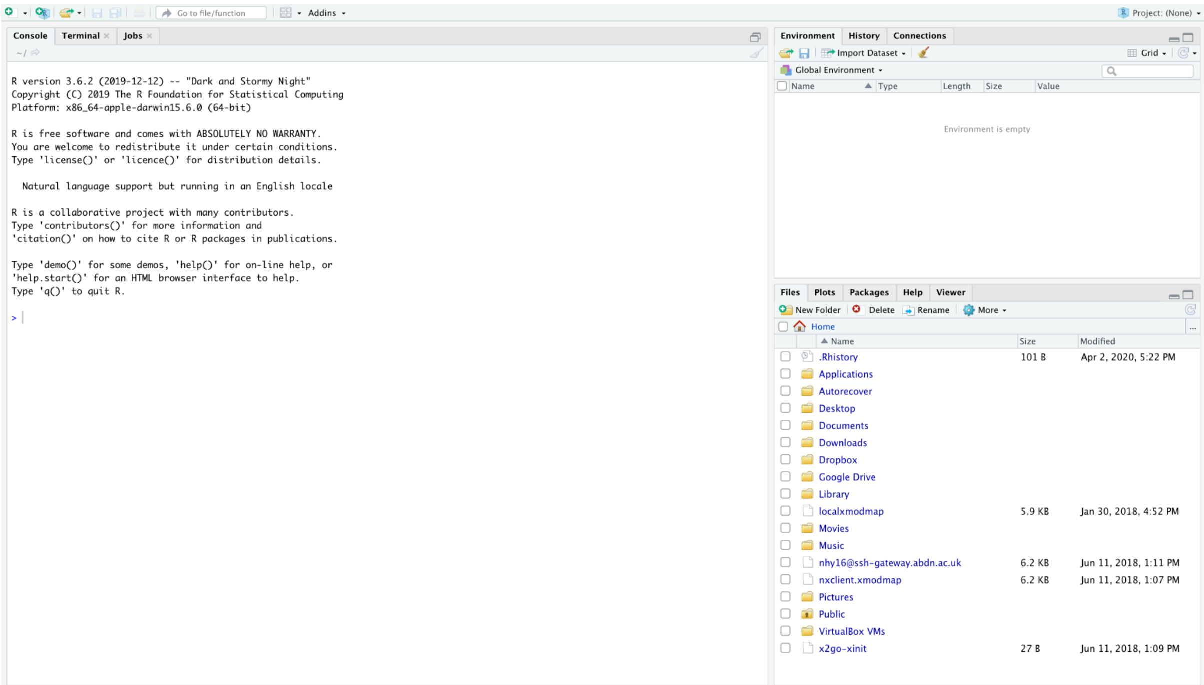

Figure 3: RStudio default IDE

3.2 SAA - Nigeria use case space

The use case has its own private storage space in CG Labs to store and share files amongst team members. The name of the shared space is "saa-use-case", and is located in the home directory of the user when logging in (figure 4). This space has a specific architecture to facilitate the use of the tool, and users are encouraged to maintain it or keep it to the extent possible. This design facilitates the separation between data and processes, providing more safety against losses of files, and ensuring security. Figure 4 describes the architecture of the system. The total size of the shared space contains approximately 10 GB, and the total available storage is approximately 100 GB. This could be adjusted as per required. It is important to observe the clear separation into 2 main sections: data and functions. The folder "*data*" is the section where the information should be stored, and it is described with detail in section 5. The folder "*functions*" contains the scripts and processes running in the DST and it is described in section 6. All users who have access to this space have full-rights, and therefore, they are encouraged to be careful. Testing new routines or additional features should be carried outside of the shared space, preferably on the private user directory.

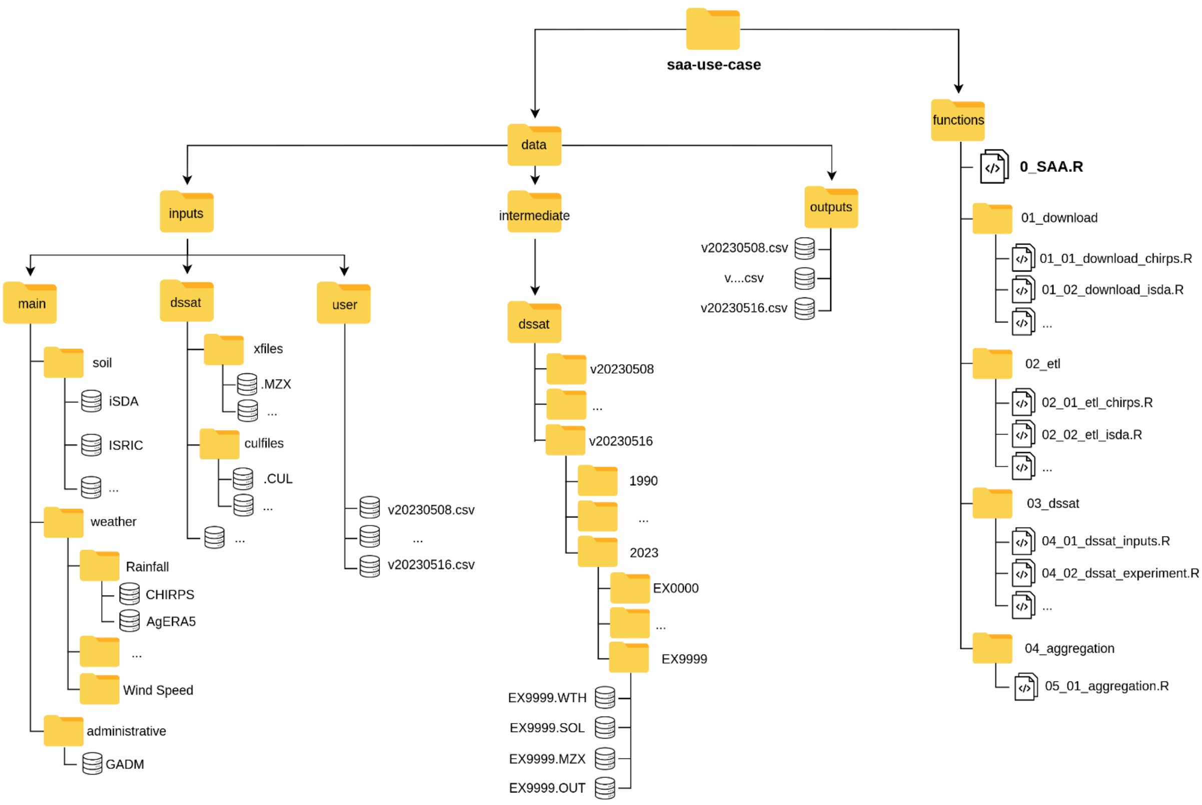

Figure 4: Architecture of the SAA DST, containing all the necessary components.

3.3 R routines

The DST is built using R software and packages to interact with the data and other software. Necessary packages are installed by default when executing the different components in the DST, but it is worth paying attention to this. The current DST works with the software requirements shown in table 1. The routines are organized around the main processes: **a)** data sourcing; **b)** ETL; **c)** DSSAT; **d)** aggregation. There is one generic script which executes the entire routine at once with the defined parameters for a simulation. The entire process is publicly accessible via a GitHub repository (https://github.com/EiA2030/mashawarar) so that partners and third parties interested can make use of the tool.

Table 1: Software requirements and versions.

3.4 DSSAT specifics

The current DST runs on the CERES model of DSSAT v47, which is written on Fortran and requires an installation of the software and is included in all CG Labs server options by default (see 4.1 CG Labs). DSSAT requires inputs in a specific format and there are minimum input requirements in order for the software to execute. The DST covers these and adds more than the minimum data requirements, although certain management conditions are still being implemented (as of the writing of this document). In order to not have to redesign the simulations everytime, certain management conditions are retrieved from template experimental files (.MZX) stored under intermediate. Anytime the user wants to adjust the management condition simulations, a new template must be stored under /home/jovyan/saa-use-case/data/inputs/dssat/xfiles/template_vYYYYMMDD.MZX where *YYYYMMDD* is the date corresponding to when the simulation is being executed, otherwise the DST will return an error. Additional templates are provided for soil (.SOL) and weather (.WTH) error files, containing -99 values (NA in DSSAT).

The DST executes DSSAT as part of the routines, producing DSSAT files which are stored under /home/jovyan/saa-use-case/data/intermediary/dssat/vYYYYMMDD, again corresponding to the date when the DST was run. These folders contain the specific files for a simulation, including the outputs (.OUT) which are used in the aggregation step (see [6.4 Aggregation]). The other files are produced by the ETL processes for soil and weather inputs and are kept for archiving the simulation.

One important consideration for the DSSAT process is that the engine and actual binary executables from DSSAT are in the root directory, under the folder /opt/DSSAT which also contains other DSSAT default data, as shown in figure 5. It is important to consider that all customized data for DSSAT (e.g.: variety coefficients in .CUL files) need to be at the DSSAT directory, in order for the simulation to succeed.

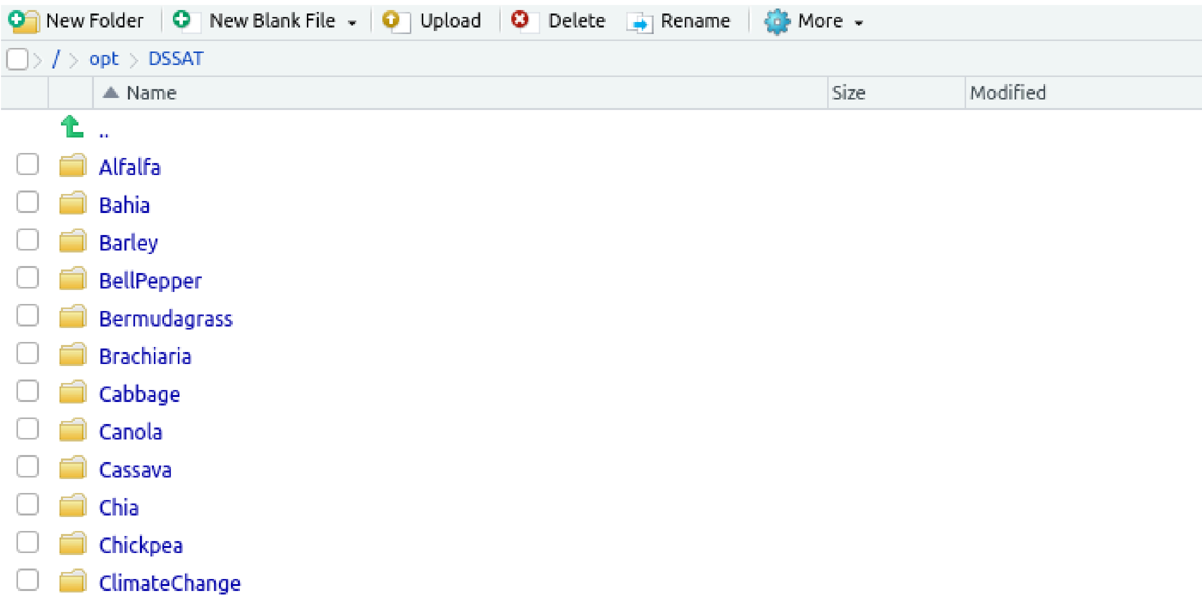

Figure 5: directory where DSSAT executables are stored in CG Labs. This contains the binary Fortran files, as well as additional default data for DSSAT.

4. Data

4.1 Inputs

There are data input requirements for the DST to execute, such as environmental information (see 3.2 Environmental Data), but also inputs required by DSSAT v47 software, or others that need to be provided by the user (such a specific target area). This data is stored under /home/jovyan/saa-use-case/data/inputs. Again, this has three sub-folders with different characteristics.

4.1.1 Main

The first sub-folder is main, and it is mainly referring to spatial or environmental datasets typically stored as NetCDF or GeoTIFF files, all referring to the extent of Nigeria. Most of the time this directory stays up to date, since there are specific functions to download and source data (see [6.1 Download data](#)). At the moment main contains 3 sub-folders: soil, weather and administrative, with a *README.md* describing the contents. Each of these folders contains available databases. Figure 6 indicates the substructure for this folder Unless the user wants to add a specific environmental database, it is better not to add things here. Please always discuss and inform users of any new data to be added.

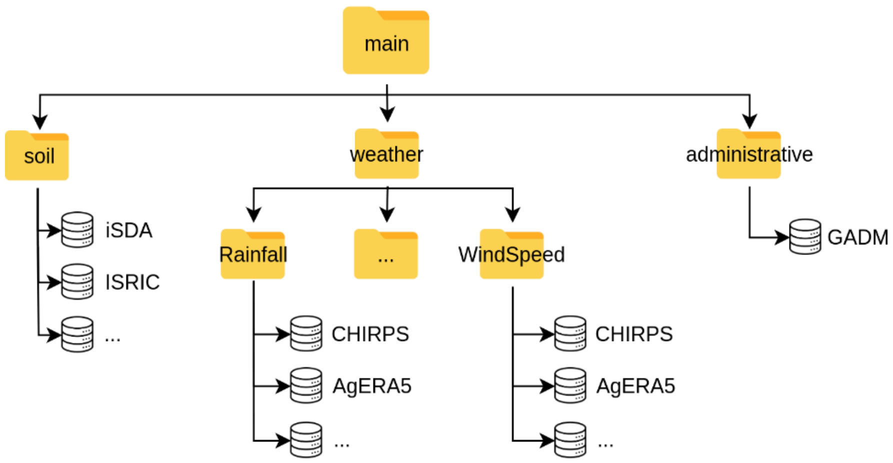

Figure 6: main folder under data/inputs containing the spatial and environmental databases.

4.1.2 DSSAT

Another sub-folder is dssat, which stores template format files and input data necessary for DSSAT (figure 7). This includes experimental files (.MZX) with template management design for the simulations. These management templates need to be kept track in order to know if the DST runs on a different scenario, and thus the files should be deposited using the labeling convention vYYYYMMDD.MZX, where:

- **v**: Indicates a version
- **YYYY**: Indicates the year of the DST execution
- **MM**: Indicates the month of the DST execution
- **DD**: Indicates the day of the DST execution
- **.MZX**: File extension for DSSAT

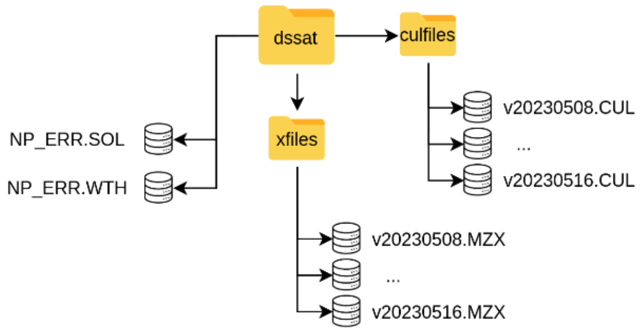

Figure 7: The dssat folder with its specific structure containing templates (xfiles, NP_ERR) and the SAA.CUL with the specific varieties part of the DST.

Cultivar files (SAA.CUL) are DSSAT files containing genotype coefficients for different maize varieties. These are calibrated coefficients for specific locations, or generic ones. In the dssat sub-folder there should only be a single .CUL file, which contains the different coefficients for the varieties used in the DST. Users should add new varieties with the relevant variables ("*VAR#*", "*VRNAME*", "*P1*", etc) to this file. The DST will only execute the varieties included in this file. It is also important to consider that the tool will execute all varieties included in this file. A new variety should be added at the end of the file, followed by a new empty line, as indicated in red below:

```
@VAR#  VRNAME.......... EXPNO   ECO#    P1    P2    P5    G2    G3 PHINT
!                                        1     2     3     4     5     6
IF0012 DT STR W             . IB0001 302.0 0.400 805.9 780.0  6.50 40.00
IF0014 COMP1 SYN NEW        . IB0001 253.3 0.424 794.9 743.3  6.25 38.90
IF0019 EVDT 99              . IB0001 199.5 0.300 789.0 720.4  6.69 40.00
```

4.1.3 User

The user folder contains data provided by the user. These could be files containing latitude (Y) and longitude (X) coordinates in decimal degrees and WGS84 (EPSG:4326) SRID. The first column should refer to the longitude (X) and the second column to latitude (Y). The files should be stored as a (.CSV) with the same labeling convention as the experimental files (.MZX).

4.2 Intermediate

Data produced in the DSSAT component of the DST (component B, figure 1) are stored in /home/jovyan/saa-use-case/data/intermediate. The folders under intermediate refer to the different DSSAT executions of the DST and are labeled with the versioning indicated in section 5.1: vYYYYMMDD. These are auto-generated folders upon execution of the DST. These folders contain a series of sub-folders labeled as EX0000, referring to each location (pair of longitude and latitude) of the simulation. Each of these represents a simulated location and contains those DSSAT files in the specific formats of the software, which include .WTH, .SOL, .MZX and .OUT. They are generated by the dssat functions (see 6.3

DSSAT) and the user does not need to interact with them. The output files comprise the crop growth database (fig. 1) and are the inputs for the aggregation functions.

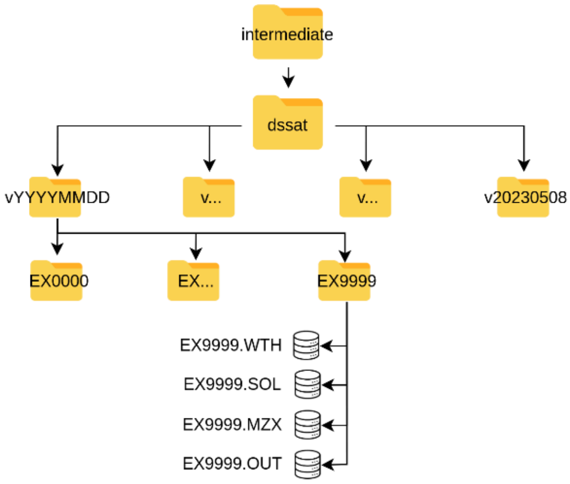

Figure 8: Structure of the intermediate data folder

4.3 Outputs

The final outputs of the DST are aggregated results from the DSSAT OUT crop growth database. These are also auto-generated files following the mentioned naming convention (see 5.1 Inputs) and are stored in .CSV tabular format under **/home/jovyan/saa-use-case/data/outputs**. The files contain the final and necessary information for the recommendations:

- **lookup_key**: Location unique identifier
- **var**: Variety recommended
- **pdate**: Planting date to be recommended
- **rank**: for each location the ranking of each variety and planting date recommendation.
-

Table 2 provides an example of an aggregated output from the DST.

5. Functions

All the processes indicated below are sub-processes and already part of the 0_saa.R mother script, which is the interface to the DST for the users. Any execution of the DST should be done via running this script, and not running the sub-processes shown below. These are a mere demonstration to provide more background and nuisance on the tool for users.

5.1 Download data

Necessary data for the execution of the DST has to be pre-populated into the system. This process is carried by the tools stored in **/home/jovyan/saa-use-case/functions/01_download**. These set of functions do not run with the rest of the DST, since these are heavy processes which take time. Instead, the data manager will need to make sure the data is updated and execute the functions as per the update schedules, or in worst case after data losses. The data sources rely on CG Labs internal data vault to access and repopulate the data inputs for environmental data. Each download process is specific for the different data sources in ***/home/jovyan/saa-use-case/data/input/main*** and needs to be executed separately from the CG Labs terminal. For example, to download the data for chirps data, the user would need to execute the following command:

```
jovyan@user:~$ cd /home/jovyan/saa-use-case/functions/01_download
jovyan@user:~/saa-use-case/functions/01_download$ Rscript 01_2_download_chirps.R 2022
```

Where *2022* is the year that needs to be updated. It is important that the user executes the script from the local script directory (*/home/jovyan/saa-use-case/functions/01_download*), as shown in the example above. The outputs will be automatically stored in the relevant folder in */home/jovyan/saa-use-case/data/input/main*

5.2 Extract, Transform and Load

The functions stored in **/home/jovyan/saa-use-case/functions/02_etl** serve to interact with the relevant data inputs in ***/home/jovyan/saa-use-case/data/input/main***. They are a typical set of ETL (*Extract, Transform and Load*) procedures that allow to provide data for the DST, and particularly for the DSSAT software. These are stand-alone procedures integrated into the DSSAT routines, so the user does not need to execute them, unless a new variable from the dataset is implemented, or a new dataset is being added, in which case an ETL will need to be developed. At the moment there are 4 ETL processes providing inputs in the DST: **02_01_isda.R** for soil variables;

02_02_agera5.R for climatic variables (except rainfall); **02_03_chirps.R** for rainfall; and **02_04_gps.R** to provide GPS coordinates in the target area if the user does not provide specific ones. These are stand-alone procedures integrated into the DSSAT routines, so the user does not need to execute them, but below are examples of how they work:

```
get.isda(X = 9.578, Y = 10.564)
```

Which would return something like:

```
iso     X      Y depth lyr_center clay sand silt bulk_density  ph
1  NG 9.578 10.564    20          10   21   55   23         1.36 6.1
2  NG 9.578 10.564    50          35   27   52   23         1.35 6.2
```

For extracting the data from CHIRPS, for example:

```
chirps(startDate = "2020-07-01", endDate = "2020-07-13",
      coordPoints = data.frame(X = 9.578, Y = 10.564))
```

Returning:

```
       X      Y      dates year month day      rain
1  9.578 10.564 2020-07-01 2020    07  01  0.000000
2  9.578 10.564 2020-07-02 2020    07  02  0.000000
3  9.578 10.564 2020-07-03 2020    07  03 20.258272
4  9.578 10.564 2020-07-04 2020    07  04 20.258272
5  9.578 10.564 2020-07-05 2020    07  05 20.258272
6  9.578 10.564 2020-07-06 2020    07  06  5.997376
7  9.578 10.564 2020-07-07 2020    07  07 11.994753
8  9.578 10.564 2020-07-08 2020    07  08  5.997376
9  9.578 10.564 2020-07-09 2020    07  09  5.997376
10 9.578 10.564 2020-07-10 2020    07  10 11.994753
11 9.578 10.564 2020-07-11 2020    07  11  0.000000
12 9.578 10.564 2020-07-12 2020    07  12 33.512047
13 9.578 10.564 2020-07-13 2020    07  13  6.702409
```

5.3 DSSAT

As mentioned, this is the engine of the DST. These functions are the processes that form component B shown in figure 1, and are stored in **/home/jovyan/saa-use-case/functions/03_dssat**. Just as the other download and ETL processes, the user does not need to interact with these scripts in order to execute the DST. They automatically generate and run the necessary steps to run and execute DSSAT, and depend on *02_etl* (see 6.2 Extract, Transform and Load) to prepare the necessary soil and weather data requirements.

The functions provide the possibility to be executed in parallel for computationally intensive runs of the DST, such as multiple years, over large number of locations (e.g.; 100). They require certain inputs to be provided by the user, such as the target area or locations to be simulated and the range of dates to cover. The functions automatically read and construct the necessary folder structure to store the intermediate DSSAT results under ***/home/jovyan/saa-use-case/data/intermediate/dssat***. The processes are sequential and need to be executed in order, but this is handled by the DST. Below are examples of these functions.

The function **03_01_dssat_inputs.R** is responsible for preparing the necessary environmental data (soil and weather) in the required DSSAT formats. It uses the set of locations and range of times to extract the relevant information as per below:

```
dssat.extdata(coords = data.frame("longitude" = c(9.578), "latitude" = c(10.564)),
              sdate = "2023-01-01", edate = "2023-12-31",
              jobs = 1,
              path.to.ex = "/home/jovyan/saa-use-case/data/intermediate/dssat/v20231231")
```

The code above will extract and write DSSAT .WTH and .SOL files to a folder */home/jovyan/saa-use-case/data/intermediate/dssat/v20231231/EXTE0000/* for the range of dates indicated and the coordinates provided. The DST is designed to simulate the entire year (1st Jan to 31st Dec), so the .WTH files will contain daily weather observations for each entire year in the date range.
DSSAT requires an Xfile or experimental file which defines the scenario to be simulated. This file is generated using the **03_02_dssat_experiment.R** script and requires the same arguments to be defined:

```
dssat.Xdata(coords = data.frame("longitude" = c(9.578), "latitude" = c(10.564)),
            sdate = "2023-01-01", edate = "2023-12-31",
            jobs = 1,
            path.to.ex = "/home/jovyan/saa-use-case/data/intermediate/dssat/v20231231")
```

This produces the .MZX file using the template *vYYYYMMDD.MZX* (see [5.1 Inputs]) provided by the user in */home/jovyan/saa-use-case/data/inputs/dssat/xfiles*. These files look something like:

```
*EXP.DETAILS:

*GENERAL
@PEOPLE
@ADDRESS
@SITE
X=7.82500000069449, Y=12.2749999676097

*TREATMENTS                        -------------FACTOR LEVELS------------
@N R O C TNAME.................... CU FL SA IC MP MI MF MR MC MT ME MH SM
 1 1 1 0 IF0012_PD_1                1  1  0  1  1  0  1  0  0  0  0  0  1
 2 1 1 0 IF0012_PD_2                1  1  0  1  2  0  1  0  0  0  0  0  1

*CULTIVARS
@C CR INGENO CNAME
 1 MZ IF0012 DT STR W
```

Once the required files (.SOL, .WTH and .MZX) are generated, it is possible to execute DSSAT. The function in 03_03_dssat_execute.R uses the same inputs as the other two dssat functions shown above, and launches a batch execution. The DSSAT output files are stored in */home/jovyan/saa-use-case/data/intermediate/dssat/v20231231/EXTE0000/*, but only the .OUT files are kept. Below is an example of how DSSAT is executed in the DST:

```
dssat.execute(coords = data.frame("longitude" = c(9.578), "latitude" = c(10.564)),
              sdate = "2023-01-01", edate = "2023-12-31",
              jobs = 1,
              path.to.ex = "/home/jovyan/saa-use-case/data/intermediate/dssat/v20231231")
```

5.4 Aggregation

The final outputs from the DST are generated through the functions in **/home/jovyan/saa-use-case/functions/04_aggregation**. There are two processes in this component (C in figure 1): ***04_01_aggregation_dssat.R*** which aggregates DSSAT outputs; and ***04_02_aggregation_rank.R*** which ranks the aggregated outputs and produces the final look-up table for the DST (see 5.3 Outputs). Both processes are again automated and part of the processing chain, so the user does not need to explicitly run the scripts. The function 04_01_aggregation_dssat.R reads the DSSAT .OUT files for all locations, varieties and planting dates in the DST run, and puts everything into a single temporary table. It is executed as follows:

```
dssat.aggregate(years = c(2020:2023),
                jobs = 1,
                path.to.ex = "/home/jovyan/saa-use-case/data/intermediate/dssat/v20231231")
```

The final ranked outputs from the DST are generated through the script 04_02_aggregation_rank.R, which produces several metrics of the yield (upper limit, mean, lower limit and coefficient of variation) to determine the most appropriate combination of variety and planting windows in each simulated location. The function in this script can be executed as follows:

```
rank.aggregate(years = c(2020:2023),
               jobs = 1,
               path.to.ex = "/home/jovyan/saa-use-case/data/intermediate/dssat/v20231231")
```

Which stores the final DST output in a CSV tabular format in */home/jovyan/saa-use-case/data/outputs/* (see 5.3 Outputs).

6. Executing the DST

To execute the DST, the user can run a simple command from the terminal. The terminal window is available in Rstudio IDE, besides the console tab (figure 3).

```
jovyan@user:~$ cd /home/jovyan/saa-use-case/functions
jovyan@user:~/saa-use-case/functions$ Rscript 0_saa.R Kano 2022-01-01 2022-12-31 6
```

It is recommended to execute the DST from the functions directory, as shown in the first line of the block above. In the example, the Rscript command is used to execute R scripts from the terminal, in this case, we are executing the 0_saa.R script, a 'mother' script containing all other scripts and putting the different processes together. After this the user needs to provide a set of 4 arguments. The first argument is the set of *GPS coordinates* corresponding to each point simulation (in the example above Kano). Optionally, the user can provide a table in CSV format with the file name convention (as indicated in 5.1.3 User) and storing it under the relevant user folder. The 2 next arguments are the *start* and *end dates* (2022-01-01 and 2022-12-31 in the example), composing the date ranges that the DST will be executed for. This can span over multiple years, and at least, the DST will execute an entire calendar year, even if the year is the same both at the start and end dates. Finally, the user needs to provide the number of parallel processes to be executed (6 in the example). This process also checks that the input requirements are met, including the .MZX file with the appropriate name, and will write the intermediate and final outputs in the relevant folders.

References:

Boogaard, H., Schubert, J., De Wit, A., Lazebnik, J., Hutjes, R., Van der Grijn, G., (2020): Agrometeorological indicators from 1979 to present derived from reanalysis. Copernicus Climate Change Service (C3S) Climate Data Store (CDS). DOI: 10.24381/cds.6c68c9bb

Funk, C.C., Peterson, P.J., Landsfeld, M.F., Pedreros, D.H., Verdin, J.P., Rowland, J.D., Romero, B.E., Husak, G.J., Michaelsen, J.C., and Verdin, A.P., 2014, A quasi-global precipitation time series for drought monitoring: U.S. Geological Survey Data Series 832, 4 p. http://pubs.usgs.gov/ds/832/

Hoogenboom, G., C.H. Porter, V. Shelia, K.J. Boote, U. Singh, J.W. White, W. Pavan, F.A.A. Oliveira, L.P. Moreno-Cadena, J.I. Lizaso, S. Asseng, D.N.L. Pequeno, B.A. Kimball, P.D. Alderman, K.R. Thorp, M.R. Jones, S.V. Cuadra, M.S. Vianna, F.J. Villalobos, T.B. Ferreira, W.D. Batchelor, J. Koo, L.A. Hunt, and J.W. Jones. 2021. Decision Support System for Agrotechnology Transfer (DSSAT)

Hengl, Tomislav, Miller, Matt, Križan, Josip, Kilibarda, Milan, Acquah, Gifty, & Sila, Andrew M. (2020). iSDAsoil: soil organic carbon for Africa predicted at 30 m resolution at 0-20 and 20-50 cm depths (v0.13)

Sanchez, P. A., Palm, C. A., & Buol, S. W. (2003). Fertility capability soil classification: a tool to help assess soil quality in the tropics. Geoderma, 114(3-4), 157-185.