CropSuite

Version 1.0

Effective 12-12-2024

User Manual

Matthias Knüttel
Florian Zabel

Cite As:

Knüttel, M., Zabel, F. (2024): CropSuite User Manual. DOI:
https://doi.org/10.5281/zenodo.14196315

Table of Contents

1. Introduction

Due to ongoing climate change, global population growth, and increasing food pressure, it is becoming more and more important to increase the productivity of agricultural land. One approach modelling the agricultural suitability of land is CropSuite. By integrating climate and soil data through sophisticated modelling techniques, CropSuite aims to identify the most suitable crops for regions, empowering farmers and policymakers to make informed decisions and secure food production in a sustainable manner.

Soil properties such as fertility, texture, and drainage significantly influence a crop's growth and overall performance. A central feature of CropSuite is its utilization of the membership function approach to assess crop suitability. This methodology allows for the representation of complex relationships between crops and environmental parameters, assigning suitability values between 0 and 1 based on defined parameter sets.

This program represents an advance in crop suitability modelling, which builds on the work of Zabel et al. from 2014. Their Fortran code was translated into Python and at the same time several extensions and improvements were made to improve usability and expand the possibilities.

2. Short Description

2.1. Prerequisites

To run CropSuite, your system must meet the following requirements:
Python 3.9 or later

Required Python Packages: The program relies on the following Python packages. It is essential to install these packages before running the program:

- os: Enabling file and directory operations.
- configparser: Enables reading and writing configuration files in INI format.
- numpy: Offers support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
- sys: Gives access to some variables used or maintained by the interpreter and functions that interact with the interpreter.
- multiprocessing: Allows parallel processing by running multiple processes simultaneously.
- scipy: Provides additional scientific computing capabilities, such as numerical integration, interpolation, optimization, and more.
- statistics: Offers various statistical functions for data analysis.
- glob: Facilitates pattern matching for file and directory paths.
- rasterio: Enables reading and writing geospatial raster data.
- concurrent: Provides support for concurrent programming in Python.
- psutil: Allows retrieval of information about system utilization, such as CPU, memory, disks, and network.
- matplotlib: A plotting library for creating high-quality graphs and visualizations.
- math: Contains mathematical functions not included in the Python built-in math module.
- xarray: Module to read netCDF4 files to python arrays
- gc: Garbage collector, fixed Python component
- numba: Partial compilation for individual program parts for better performance
- rio-cogeo: Writing of Cloud Optimized GeoTiff files
- cartopy: Required for plotting maps
- dask: Package for parallel computing
- netCDF4: Required for reading and writing NetCDF4 files
- pillow: High-performance handling of raster files
- tkinter: Graphical User Interface

Operating System Compatibility: CropSuite is compatible with Windows, Linux and MacOS operating systems. You can run the program on any of these platforms without any modifications.

2.2. Tested version numbers

Python Version 3.9, 3.10, 3.11 and 3.12.
Version 3.13 still has incompatibilities with used packages

Package	*Tested version numbers*	*Remarks*
numpy	1.24.3, 1.25.0, 1.26.4	
scipy	1.9.3, 1.11.2, 1.14.1	RegularGridInterpolator instead of interp2d in newer versions
rasterio	1.3.9, 1.3.10, 1.4.1	
matplotlib	3.6.3, 3.9.2	
xarray	2023.6.0, 2024.10.0	
numba	0.60.0	
rio-cogeo	5.3.3, 5.3.6	
cartopy	0.23.0, 0.24.1	
dask	2023.3.2, 2024.10,0	
netCDF4	1.6.4, 1.7.2	
pillow	10.2.0, 11.0.0	

2.3. Usage of the cartopy package on MacOS

Apple's MacOS uses very restrictive protection against unsafe websites. For this reason, the cartopy package, which is used for the map display in the viewer and in the options window, is not able to download the required shapefiles. The source does not use SSL encryption, because of which an error message appears in the terminal when using the Options window or the Data Viewer, warning of the missing SSL certificate.

This can be remedied by manually downloading the required files:

Extract and place the files here:

Win: C:\Users\[username]\.local\share\cartopy\shapefiles\ natural_earth\physical
Mac: /Users/[username]/.local/share/cartopy\shapefiles\ natural_earth\ physical

https://naciscdn.org/naturalearth/10m/physical/ne_10m_coastline.zip

https://naciscdn.org/naturalearth/10m/physical/ne_10m_ocean.zip

https://naciscdn.org/naturalearth/10m/physical/ne_10m_land.zip

https://naciscdn.org/naturalearth/10m/physical/ne_10m_lakes.zip

https://naciscdn.org/naturalearth/10m/physical/ne_10m_rivers_lake_centerlines.zip

https://naciscdn.org/naturalearth/50m/physical/ne_50m_coastline.zip

https://naciscdn.org/naturalearth/110m/physical/ne_110m_coastline.zip

https://naciscdn.org/naturalearth/110m/physical/ne_110m_ocean.zip

https://naciscdn.org/naturalearth/110m/physical/ne_110m_lakes.zip

Extract and place the files here:

Win: C:\Users\[username]\.local\share\cartopy\shapefiles\ natural_earth\cultural
Mac: /Users/[username]/.local/share/cartopy\shapefiles\ natural_earth\cultural

https://naciscdn.org/naturalearth/10m/cultural/ne_10m_admin_0_boundary_lines_land.zip

https://naciscdn.org/naturalearth/50m/cultural/ne_50m_admin_0_boundary_lines_land.zip

https://naciscdn.org/naturalearth/110m/cultural/ne_110m_admin_0_boundary_lines_land.zip

3. Before running CropSuite

Please follow these installation steps:

Python Installation

Ensure that you have Python 3 installed on your system. If you don't have Python installed, you can download the latest version from the official website: https://www.python.org/downloads/

During the installation process, make sure to check the option to add Python to your system's PATH environment variable. This will allow you to run Python and pip (Python package manager) from the command line or terminal.

IMPORTANT

Depending on the type of Python installation and operating system, it may be necessary to use python3 instead of python:

```
>>python -m pip […]
```

or

```
>>python3 -m pip […]
```

In the following examples, the variant with >>python is used.

Installing Required Packages

Once you have Python installed, you need to install the necessary Python packages for CropSuite. Open a terminal or command prompt and run the following command to install the required packages:

```
>> python -m pip install cartopy dask "dask[distributed]" numpy scipy matplotlib netCDF4 rasterio psutil rio-cogeo numba tk pillow xarray
```

This command will install the packages 'numpy', 'scipy', 'matplotlib', 'rasterio', 'psutil', 'numba', 'rio-cogeo' and 'psutil', which are required for the program's functionality. If any of these packages are already installed, the command will skip those and only install the missing ones.

Verify Installation

To verify that all required packages are installed correctly, you can run the following command:

```
>> python -m pip show cartopy numpy scipy dask "dask[distributed]" matplotlib netCDF4 rasterio psutil rio-cogeo numba tk pillow xarray
```

This command will display information about the installed versions of each package.

Hardware Recommendations

To ensure optimal performance and efficient execution of CropSuite, we provide the following hardware recommendations:

	Minimum	Recommended
Processor	2 Cores with 4 Threads	24 Cores with 48 Threads
Memory (RAM)	8 GB RAM	> 128 GB RAM
Storage	> 256 GB SSD with read/write speed > 500 MB/s	

A multi-core processor will enhance the program's ability to handle multiple tasks simultaneously, improving overall responsiveness.

Sufficient RAM is crucial for handling large datasets and complex computations without slowdowns or memory-related issues.

An SSD (Solid State Drive) provides faster read/write speeds, resulting in quicker data access and reduced loading times.

Please be aware that the hardware demands are contingent on the application's scope. Lesser requirements are applicable when dealing with smaller areas. Conversely, for extensive areas and numerous crops, a substantial amount of working memory is particularly advisable. While the program includes measures to prevent out-of-memory errors, crashes might still arise, especially in Linux environments, when processing exceptionally large areas.

3.1. Required Datasets

IMPORTANT

All raster input datasets must be available in the World Geodetic System 1984 (WGS 84, EPSG 4326) in plate carree projection!

Please ensure that the following data sets are provided:

3.1.1. Elevation Data

IMPORTANT

The spatial resolution of the output files corresponds to that of the Elevation Dataset!

By default, the “srtm_1km_world_recalculated.tif” raster file is in the designated data folder. This file originates from the Space Radar Topography Mission (SRTM) project and possesses a spatial resolution of 0.008333°/pixel (30 arcsec). Certainly, it is important to acknowledge that the “srtm_1km_world_recalculated.tif” file can be replaced with an alternative file as needed. However, it is essential to bear in mind that the resolution of the elevation data directly influences the resolution of the output data.

3.1.2. Land-Sea-Mask

A land-sea mask is used to mask water or land areas. By default, the worldclim_land_sea_mask.tif is found in the data folder. Ideally, the land-sea mask has the same resolution and dimensions as the elevation data.

3.1.3. Climate Data

Essentially important are the climate data. The temperature and precipitation totals are required.

For both temperature and precipitation, CropSuite expects a geotiff file with 365 bands, one for each day. The shape of the geotiff then looks like this (days, y-axis, x-axis). For the example of Africa, we have the shape (365, 1740, 1663).

The temperature grid provides the daily mean temperature in °Celsius for every day of the year.

It is important to note that the temperature and precipitation data have the same dimensions and spatial resolution.

As of now, a fixed file name is still expected for the climate data. The temperature data are expected in ‘Temp_avg.tif’ / ‘Temp_avg.nc’, the precipitation totals in ‘Prec_avg.tif’ / ‘Prec_avg.nc’.

3.1.4. Variability Files

To take climate variability into account, specific data sets are required for each crop that describe the exceedance of climate thresholds. These data sets can be 2-dimensional (x, y) for perennial crops, or 3-dimensional (day, x, y) for non-perennial crops. Each pixel shows whether the temperature is too high or too low or the precipitation is too high or too low on each day.

A distinction is made between irrigated and rainfed cropping, whereby the variability of the lower precipitation limit is not considered for irrigated cropping. The data sets are calculated for each pixel as follows:

$$Var_{rainfed} = \max\left\{\begin{matrix} \frac{1}{noy}\sum_{y=0}^{noy}\begin{cases} 1: \frac{1}{lgc}\sum_{doy}^{doy+lgc} T_{y\,doy} > T_{high\,lim} \\ 0: \frac{1}{lgc}\sum_{doy}^{doy+lgc} T_{y\,doy} \leq T_{high\,lim} \end{cases}, \frac{1}{noy}\sum_{y=0}^{noy}\begin{cases} 1: \frac{1}{lgc}\sum_{doy}^{doy+lgc} T_{y\,doy} < T_{low\,lim} \\ 0: \frac{1}{lgc}\sum_{doy}^{doy+lgc} T_{y\,doy} \geq T_{low\,lim} \end{cases}, \\ \frac{1}{noy}\sum_{y=0}^{noy}\begin{cases} 1: \frac{1}{lgc}\sum_{doy}^{doy+lgc} P_{y\,doy} > P_{high\,lim} \\ 0: \frac{1}{lgc}\sum_{doy}^{doy+lgc} P_{y\,doy} \leq P_{high\,lim} \end{cases}, \frac{1}{noy}\sum_{y=0}^{noy}\begin{cases} 1: \frac{1}{lgc}\sum_{doy}^{doy+lgc} P_{y\,doy} < P_{low\,lim} \\ 0: \frac{1}{lgc}\sum_{doy}^{doy+lgc} P_{y\,doy} \geq P_{low\,lim} \end{cases} \end{matrix}\right\}$$

$$Var_{irrigated} = \max\left\{\begin{matrix} \frac{1}{noy}\sum_{y=0}^{noy}\begin{cases} 1: \frac{1}{lgc}\sum_{doy}^{doy+lgc} T_{y\,doy} > T_{high\,lim} \\ 0: \frac{1}{lgc}\sum_{doy}^{doy+lgc} T_{y\,doy} \leq T_{high\,lim} \end{cases}, \frac{1}{noy}\sum_{y=0}^{noy}\begin{cases} 1: \frac{1}{lgc}\sum_{doy}^{doy+lgc} T_{y\,doy} < T_{low\,lim} \\ 0: \frac{1}{lgc}\sum_{doy}^{doy+lgc} T_{y\,doy} \geq T_{low\,lim} \end{cases}, \\ \frac{1}{noy}\sum_{y=0}^{noy}\begin{cases} 1: \frac{1}{lgc}\sum_{doy}^{doy+lgc} P_{y\,doy} > P_{high\,lim} \\ 0: \frac{1}{lgc}\sum_{doy}^{doy+lgc} P_{y\,doy} \leq P_{high\,lim} \end{cases} \end{matrix}\right\}$$

A file with the name format 'rrpcf_{crop}_{ir | rf}.tif' / 'rrpcf_{crop}_{ir | rf}.nc' is therefore expected in the climate data directory for each crop.

3.1.5. WorldClim Datasets

CropSuite also supports downscaling of climate data using WorldClim data to 30 arcsec resolution. This data must be available for each month.

The required datasets can be downloaded here:

https://geodata.ucdavis.edu/climate/worldclim/2_1/base/wc2.1_30s_tavg.zip

https://geodata.ucdavis.edu/climate/worldclim/2_1/base/wc2.1_30s_prec.zip

The extracted files should be placed in the respective worldclim_temp and worldclim_prec directories.

3.1.6. Soil Datasets

Any number of data sets can be used for the soil properties. An overview of the soil properties considered by default can be found in the following table:

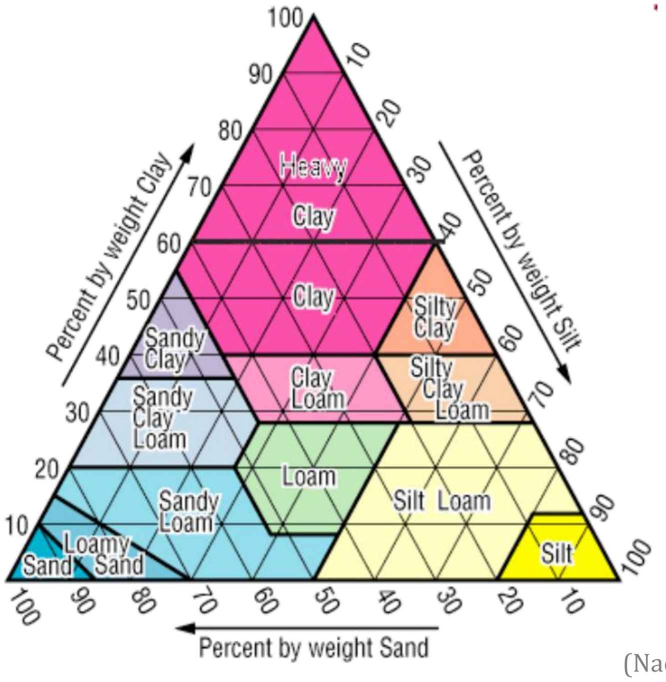
Name	Unit	Short Description
Base Saturation	%	Proportion of essential nutrient ions on soil surfaces
Coarse Fragments	Vol.-%	Large, non-soil particles present in the soil, affecting drainage and root growth
Clay Content	%	Clay content of the soil, is converted together with the sand content to the Texture Class
Gypsum Content	%	Amount of gypsum (calcium sulfate) in the soil, impacting soil structure and water retention.
pH	[1]	Measure of the soil's acidity or alkalinity, affecting nutrient availability to plants.
Salinity	dS/m	Indicator of the soil's salt content and ability to conduct electricity, influencing plant health and water absorption.
Sand Content	%	Sand content of the soil, is converted together with the clay content to the Texture Class
Organic Carbon	%	The amount of carbon stored in the soil through organic matter decomposition, influencing soil fertility and climate change mitigation.
Sodicity	%	Indicates the level of sodium (Na+) in the soil, affecting soil structure and plant growth.
Soildepth	m	The thickness or depth of the soil layer, impacting root development and nutrient availability for plants.

Soil properties data sets can vary in resolutions, dimensions, and units, if linear conversion factors are applied. These conversion factors can be conveniently defined in the config.ini file (→ config.ini), along with the option to assign varying weightings to the soil layers. In instances where output files possess a higher spatial resolution than the original soil datasets, a bilinear interpolation of the soil data is executed to ensure compatibility. If the output spatial resolution is lower than the original soil datasets, a majority resampling is used.

IMPORTANT

All soil datasets must be projected in the World Geodetic System 1984 (WGS 84, EPSG 4326) in Plate Carrée projection!

The texture class is determined from the sand content and the clay content. The USDA's triangular diagram of soil texture classes, which is also used by the FAO, is used here.



(Nachtergaele et al. 2008)

The texture classes are assigned as follows:

Code	Texture	Classes used in Sys et al. 1993
1	Heavy Clay	Cm
2	Silty Clay	SiCm
3	Clay	Co, C60
4	Silty Clay Loam	SiCL, SiCs
5	Clay Loam	CL
6	Silt	Si
7	Silt Loam	SiL
8	Sandy Clay	SC
9	Loam	L
10	Sandy Clay Loam	SCL
11	Sandy Loam	SL
12	Loamy Sand	LS, Lcs Lfs
13	Sand	S, fS

3.2. Option Files

In addition to the input data in raster format, two different types of config files are necessary.

3.2.1. config.ini

Various general settings can be made in config.ini. Paths to the input files must also be specified here and all soil layers that are to be included in the analysis must be defined.

The config.ini is an ASCII file that can be opened and edited with any text editor.

The config.ini is divided into different sections.

In the [files] section, the paths and basic file names are defined:

Section [files] Name	Type	Description
output_dir	Path	Path where the output files are to be stored
climate_data_dir	Path	Path where the 3D geotiff files containing the climate data are stored.
plant_param_dir	Path	Path where the plant parameterisation files are stored
fine_dem	Path	Path to the DEM-tif file
land_sea_mask	Path	Path to the Land-Sea-Mask-tif file
texture_classes	Path	Path to the USDA Texture classification file
worldclim_precipitation_data_dir	Path	Path to WorldClim precipitation datasets
worldclim_temperature_data_dir	Path	Path to WorldClim temperature datasets

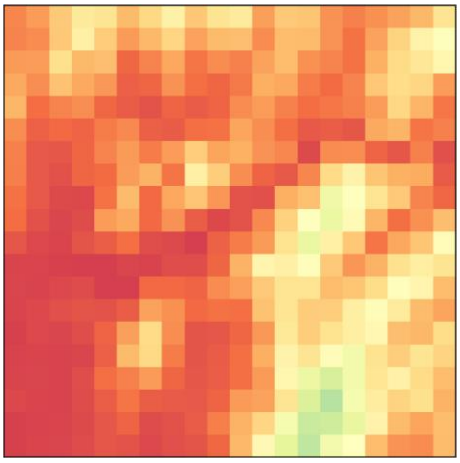
By default, CropSuite offers various options for downscaling the climate data, which can be specified in config.ini.

The following downscaling algorithms are supported, the energy conservation is ensured:

For temperature data:

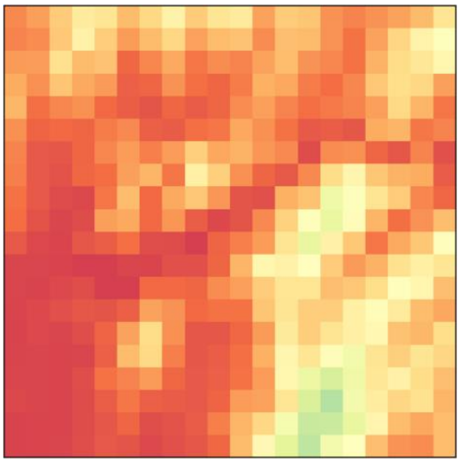
Coarse Data

-



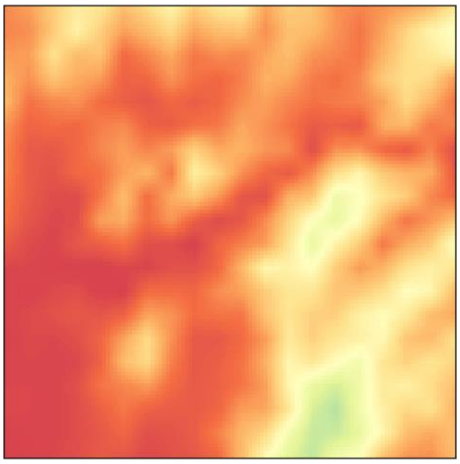
Nearest Neighbour

0



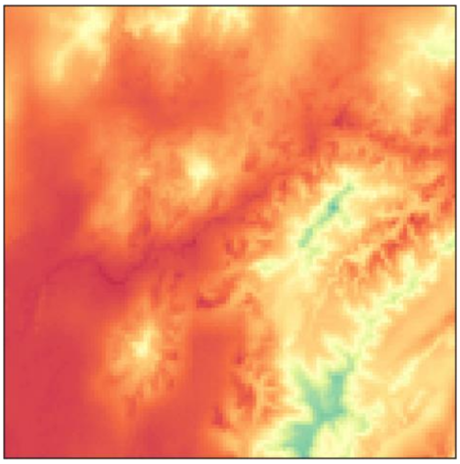
Bilinear Interpolation

1



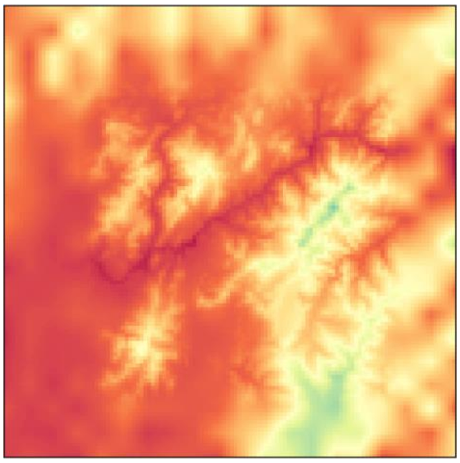
Factors based on WorldClim data

2



Height Regression

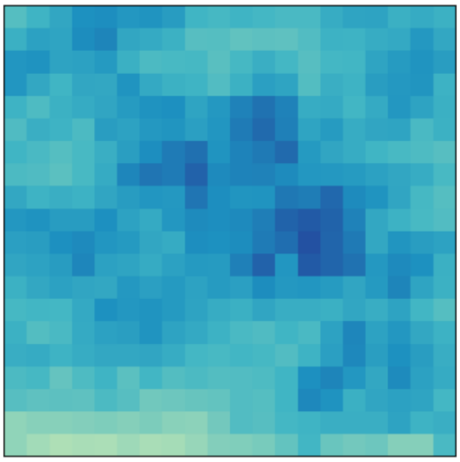
3



For precipitation data:

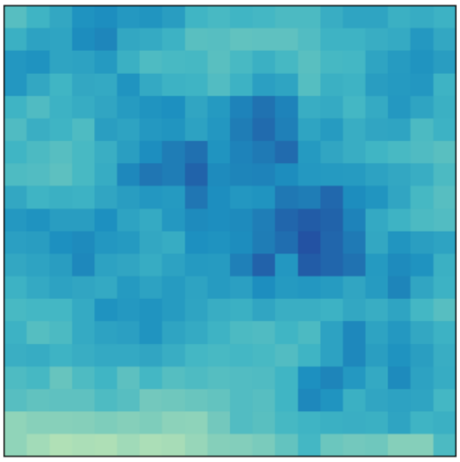
Coarse Data

\-



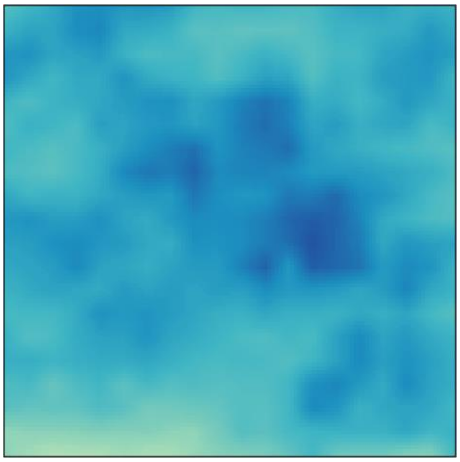
Nearest Neighbour

0



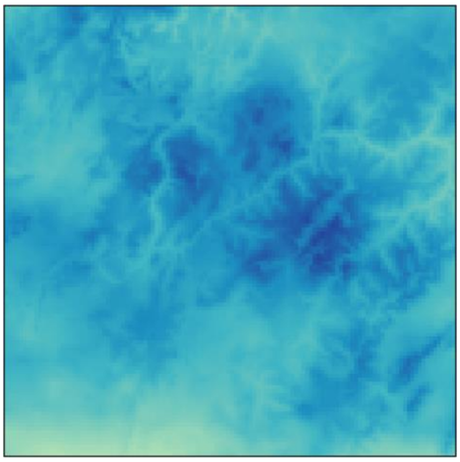
Bilinear Interpolation

1



Factors based on WorldClim data

2



In the [options] section, general settings can be made:

Section [options] Name	Available Options	Description
use_scheduler	y n	Use built-in splitting algorithm for large areas. IMPORTANT For large areas that are divided into several parts, it is essential that the output directory does not yet contain any model results!
irrigation	0, 1	1: Activate the irrigation module: Assume that all areas are Irrigated. 0: Irrigation module deactivated
precipitation_downscaling_ method	0, 1, 2	0: Nearest Neighbour interpolation 1: Bilinear Interpolation 2: By using WorldClim Datasets
temperature_downscaling_ method	0, 1, 2, 3	0: Nearest Neighbour Interpolation 1: Bilinear Interpolation 2: By using WorldClim Datasets 3: Height Regression
output_format	geotiff netcdf4 cgo	Data format of the output files: Saves the output files as geotiff, cloud-optimized geotiff (cgo) or netcdf4
downscaling_window_size	Integer	Downscaling: Size of moving window Recommended: 5-25
downscaling_use_ temperature_gradient	y n	Downscaling: Check if slope is within physical limits. Here: Adiabatic Temperature Gradient. Default: y
downscaling_dryadiabatic_ gradient	Float	Downscaling: Dry Adiabatic Temperatur Gradient [K/m]. Default: 0.00976
downscaling_saturation_ adiabatic_gradient	Float	Downscaling: Saturation Adiabatic Temperature Gradient [K/m] Default: 0.007
downscaling_temperature_ bias_threshold	Float	Downscaling: Temperature BIAS Threshold [K]. Higher Threshold leads to higher performance on the cost of quality Default: 0.0001

downscaling_precipitation_-bias_threshold	Float	Downscaling: Precipitation BIAS Threshold [mm]. Higher Threshold leads to higher performance on the cost of quality Default: 0.0001
downscaling_precipitation_per_day_threshold	Float	Downscaling: Precipitation below this value will be set to 0. [mm/day] Default: 0.75
output_all_limiting_factors	y n	Write all suitability values of the individual parameter data sets to a geotiff with several bands.
remove_interim_results	y n	This function removes all binary intermediate results. Recommended to keep the result files as small as possible. IMPORTANT Leads to considerably longer processing times if crops are added later!
remove_downscaled_climate	y n	Removes the downscaled climate data IMPORTANT Leads to considerably longer processing times if crops are added later!
output_soil_data	y n	Output combined soil data
multiple_cropping_turnaround_time	Integer	Number of days required between harvest and reseeding with multiple cropping

The [extent] section specifies the area to be modelled.

IMPORTANT
Downscaling by using the height regression method will – depending on the size of the downscaling window size – cause a strong distortion in the edge areas. It is advisable to extend the extent by 0.5 °!

Section [extent] Name	Available Options	Description
upper_left_x =	Float	Longitude of upper left corner, Format as decimal degree
upper_left_y =	Float	Latitude of upper left corner
lower_right_x =	Float	Longitude of lower right corner
lower_right_y =	Float	Latitude of lower right corner

The section [climatevariability] allows to enable the climatevariability module. The required datasets must be created first!

Section [climatevariability] Name	Available Options	Description
consider_variability =	y n	Enable climate variability module

Section [membershipfunctions]:

Section [membershipfunctions] Name	Available Options	Description
plot_for_each_crop =	y n	Plot membership functions for control

For each soil layer, a section must be included with the following settings:

Section [parameters.xxx] Name	Available Options	Description
data_directory =	Path	Path to the corresponding data set
weighting_method =	0, 1, 2	Weighting method of the layers for different soil depths 0: Only the first layer is being used 1: The top three layers are used (topsoil) 2: Where available, six soil layers are used and weighted
weighting_factors =	Six Floats	Weighting factors, if Weighting Method = 2. Compare Sys et al. 1991, page 68 cont. Default: 2,1.5,1,0.75,0.5,0.25
conversion_factor =	Integer Float	Some soil layers are not given in expected units, a conversion factor is necessary. This can be specified here. If no conversion factor is necessary, enter 1 here.
no_data =	Optional Float	If there is another no-data value in the soillayer that is not stored in the geotiff file, it can be entered here.

interpolation_method =	0, 1, 2, 3, 4, 5	Inteprolation method of the related membership function: 0: linear (Default) 1: cubic 2: quadratic 3: spline 4: poly 5: slinear
rel_member_func =	String	esignation of the membership function in the parameterisation file (→ plant parameterization .inf)

An example config.ini can be found at the end of this manual.

3.2.2. plant parameterization .inf

For each crop to be modelled, there must be an .inf file in the plant_parameterizations directory. This .inf file contains all plant-specific properties and the corresponding suitability values. An example file can be found in the appendix.

The plant parameterization files are all structured according to the following pattern:

Name	Unit/ Range	Description
name =	String	Name of the crop
growing_cycle =	Integer 1-365	Length of the growing cycle in number of days. For perennial crops use 365.
temp_vals =	°C List of Integers	Plant-specific temperature values Mean Temperature over growing cycle
temp_suit =	List of Floats 0-1	Suitability values associated with the temperature values
prec_vals =	mm List of Integers	Plant-specific precipitation totals Total Precipitation over growing cycle

prec_suit =	List of Floats 0-1	Suitability values associated with the precipitation values
hightemp_lim =	Optional Float	Set fixed upper temperature threshold of interannual variability
lowtemp_lim =	Optional Float	Set fixed lower temperature threshold of interannual variability
highprec_lim =	Optional Float	Set fixed upper precipitation threshold of interannual variability
lowprec_lim =	Optional Float	Set fixed lower precipitation threshold of interannual variability

The following optional entries are also available in the parameterisation file for winter crops:

Name	Unit/ Range	Description
wintercrop	Integer 0, 1	Flag whether the crop is a winter crop
vernalization_effective_days	Integer 1-149	Minimum number of effective vernalization days required on which the temperature must be between Tmin and Tmax
vernalization_tmax	°C Float Integer	Plant-specific maximum effective vernalization temperature
vernalization_tmin	°C Float Integer	Plant-specific minimum effective vernalization temperature
frost_resistance_days	Integer 1-149	Plant-specific number of days on which the temperature may fall below the frost resistance temperature
frost_resistance	°C Float Integer	The temperature below which the seedlings become frostbitten

IMPORTANT
Winter crops take about twice as long to calculate, as the vernalization period must be calculated separately!

The following additional entries can be used if required to define the crop:

Name	Unit/ Range	Description
photoperiod	Integer 0, 1	Flag whether the photoperiod shall be considered
minimum_sunlight_hours	Float 0-24	Minimum day length
maximum_sunlight_hours	Float 0-24	Maximum day length
lethal_thresholds	Integer 0, 1	Specify whether lethal temperature limits should be taken into account

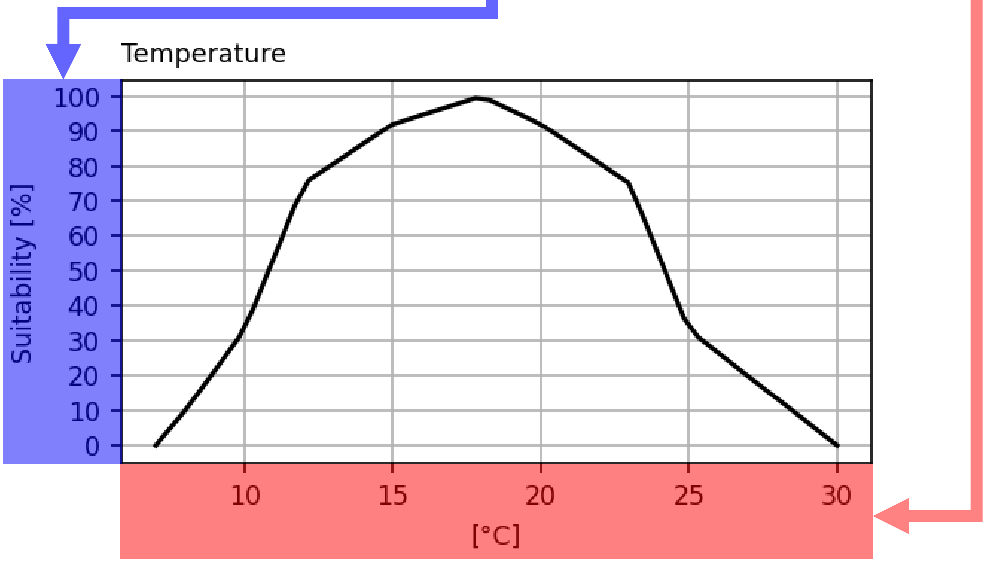
lethal_min_temp	°C Integer	Plant-specific minimum temperature
lethal_min_duration	[days] Integer	Plant-specific maximum number of consecutive days on which the temperature may fall below the minimum temperature
lethal_max_temp	°C Integer	Plant-specific maximum temperature
lethal_max_duration	[days] Integer	Plant-specific maximum number of consecutive days on which the temperature may rise above the maximum temperature
prec_req_after_sow	[mm] Integer	Minimum amount of precipitation that must fall in the period after sowing
prec_req_days	[days] Integer	Indicates the period in days after sowing in which a certain amount of precipitation must fall
temp_for_sow	[°C] Integer	Temperature for a period of time which is required for sowing
temp_for_sow_duration	[days] Integer	Time period in which a certain temperature is exceeded for sowing
consecutive_dry_days	[days] Integer	Specifies the maximum period in days in which no precipitation may fall

For each soil parameter layer a combination of the following two lines is required:

[name]_vals =	Unit of Layer List of Integers	Plant-specific values for the respective soil layer
[name]_suit =	List of Floats 0-1	Suitability values associated with the respective soil layer

Example of the temperature membership function of wheat:

```
temp_vals =        7,8,10,12,15,18,20,23,25,30
temp_suit =        0,0.1,0.33,0.75,0.92,1,0.92,0.75,0.33,0
```



Information on the change in suitability based on the frequency of crop failures is required to take climate variability into account. This is also defined as a membership function for the freqcropfail parameter:

freqcropfail_vals =	List of Floats 0-1	Plant specific crop failure frequency
freqcropfail_suit =	List of Floats 0-1	Suitability values associated with the respective crop failure frequency

Example of a full plant parameterization file for wheat:

```
name =                    wheat
growing_cycle =           115
temp_vals =               7,8,10,12,15,18,20,23,25,30
temp_suit =               0,0.1,0.33,0.75,0.92,1,0.92,0.75,0.33,0
prec_vals =               170,200,250,350,450,700,1000,1250,1500,1750
prec_suit =               0,0.1,0.33,0.75,0.92,1,0.92,0.75,0.33,0
slope_vals =              0,2,4,8,16
slope_suit =              1,0.92,0.75,0.33,0
soildepth_vals =          0.2,1.0
soildepth_suit =          0,1
texture_vals =            1,2,3,4,5,6,7,8,9,10,11,12,13
texture_suit =            0,0,1,0,1,1,1,0.92,0.92,0.75,0.33,0,0.25
coarsefragments_vals =    0,3,15,35,55
coarsefragments_suit =    1,0.92,0.75,0.33,0
gypsum_vals =             0,3,5,10,20
gypsum_suit =             1,0.92,0.75,0.33,0
base_sat_vals =           0,35,50,80,100
base_sat_suit =           0,0.33,0.75,0.92,1
ph_vals =                 5.2,5.6,6,6.5,7,7.5,8.2,8.3,8.5
ph_suit =                 0,0.33,0.75,0.92,1,0.92,0.75,0.33,0
organic_carbon_vals =     0,0.5,1.0,1.5,2
organic_carbon_suit =     0,0.33,0.75,0.92,1
elco_vals =               0,1,3,5,10
elco_suit =               1,0.92,0.75,0.33,0
esp_vals =                0,15,20,35,45
esp_suit =                1,0.92,0.75,0.33,0
freqcropfail_vals =       0,0.025,0.05,0.075,0.1,0.125,0.15,0.175,0.2,0.225,0.25
freqcropfail_suit =       1,0.98,0.95,0.88,0.73,0.5,0.27,0.12,0.05,0.02,0
```

There can be any number of plant parameterisation files in the designated folder. All crops for which files exist are modelled. Small modifications (e.g. changing the growing cycle by a few days) are also possible, but it is important to ensure that unique names are always assigned.

To check the plant parameterisations, a chart with plots for each defined membership function is automatically created for each crop in the parameterization_plots folder.

3.3. Outputs

After the program has run successfully, you will find another folder in the output directory specified in the config.ini file (→ config.ini), labelled with the Processing Extent. If the interim results are kept, various files can be found in this folder:

Name	Format	Description
downscaled_temperature.nc	netCDF4	The downscaled temperature data is represented as a 3D array with the dimensions (y, x, day). The data type used for is a 16 bit integer. Conversion factor is 1/10.
downscaled_precipitation.nc	netCDF4	The downscaled precipitation data is represented as a 3D array with the dimensions (y, x, day). The data type used for is a 16 bit integer. Conversion factor is 1/10.

If it is specified that the combined soil data should also be output, the corresponding geotiff file can also be found here for each defined soil data set.

For each modelled crop a directory is to be found in the output directory. In this folder, you will find geotiff (.tif) or netcdf4 (.nc) files, depending on the settings specified in the config.ini (→ config.ini).

The optimum sowing dates mark the beginning of the growing cycle.

The following files are created by default:

Name	Format Range	Description Unit
climate_suitability.tif climate_suitability.nc	Geotiff NetCDF4 0-100	Climate Suitability [1/100]
suitable_sowing_days.tif suitable_sowing_days.nc	Geotiff NetCDF4 0-365	Potential length of growing season [days]
limiting_factor.tif limiting_factor.nc	Geotiff NetCDF4 0, 1, 2	Climatic limiting factor. 0: Temperature limiting 1: Precipitation limiting 2: Crop failure variability limiting []
multiple_cropping.tif multiple_cropping.nc	Geotiff NetCDF4 0-3	Potential for multiple harvest [Count of Harvests]
crop_limiting_factor.tif crop_limiting_factor.nc	Geotiff NetCDF4 0, ..., n	Limiting factor, accounting climatic limitations as well as soil limitations. 0: Temperature limiting 1: Precipitation limiting 2: Crop failure variability limiting 3-n: In order of the Soil layers [ ]
crop_suitability.tif crop_suitability.nc	Geotiff NetCDF4 0-100	Suitability [1/100]
optimal_sowing_date.tif optimal_sowing_date.nc	Geotiff NetCDF4 0-365	The day/week of the year to start the growing cycle [day of year]
optimal_sowing_date_-vernalization.tif optimal_sowing_date_-vernalization.nc	Geotiff NetCDF4 0-365	The optimal sowing date if considering vernalization period for wintercrops [day of year]

optimal_sowing_date_first.tif optimal_sowing_date_first.nc	Geotiff NetCDF4 0-365	The day/week of the year to start the first growing cycle when using multiple cropping [day of year]
optimal_sowing_date_second.tif optimal_sowing_date_second.nc	Geotiff NetCDF4 0-365	The day/week of the year to start the second growing cycle when using multiple cropping [day of year]
optimal_sowing_date_third.tif optimal_sowing_date_third.nc	Geotiff NetCDF4 0-365	The day/week of the year to start the third growing cycle when using multiple cropping [day of year]
all_suitability_vals.tif	Geotiff n-Bands	Output of all suitability values for each parameter. The order of the bands is specified in limiting_factor.inf. Conversion factor is 1/100
climate_suitability_mc.tif	Geotiff NetCDF4 0-300	Climate Suitability taking multiple cropping into account. [1/100]
soil_suitability.tif	Geotiff NetCDF4 0-100	Suitability only based on static soil datasets. Climate is not considered. [1/100]

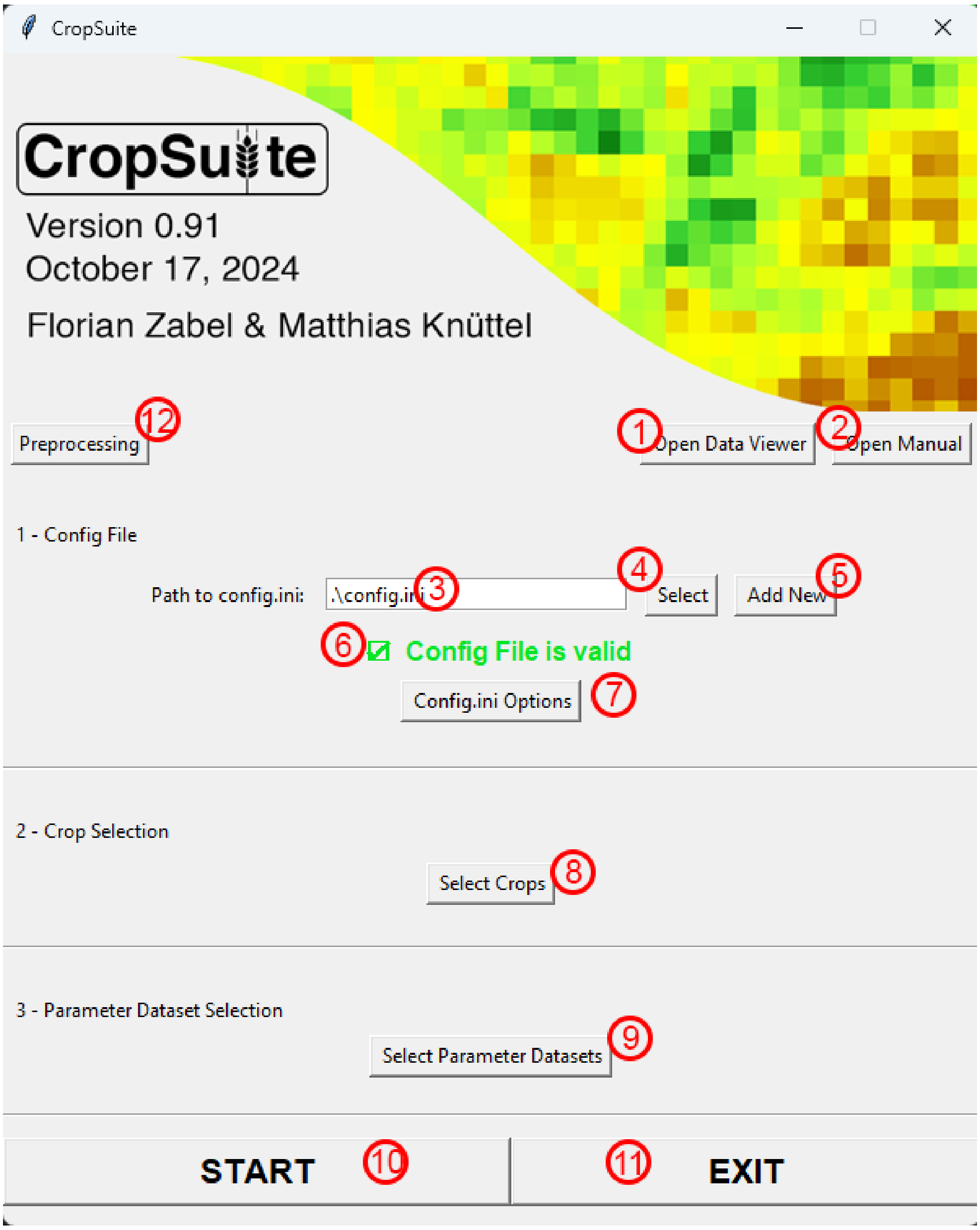
4. Using the Graphical User Interface (GUI)

CropSuite comes with a graphical user interface that greatly simplifies the process of making settings, defining plant parameters, specifying static soil data and starting the model. The GUI also offers extensive options for visualising and comparing the results and creating finished images.

The user interface is started on Windows systems by double-clicking on CropSuite_GUI.py. On Linux and MacOS systems, it is necessary to start the programme via the terminal with the following command:

```
>> python CropSuite_GUI.py
```

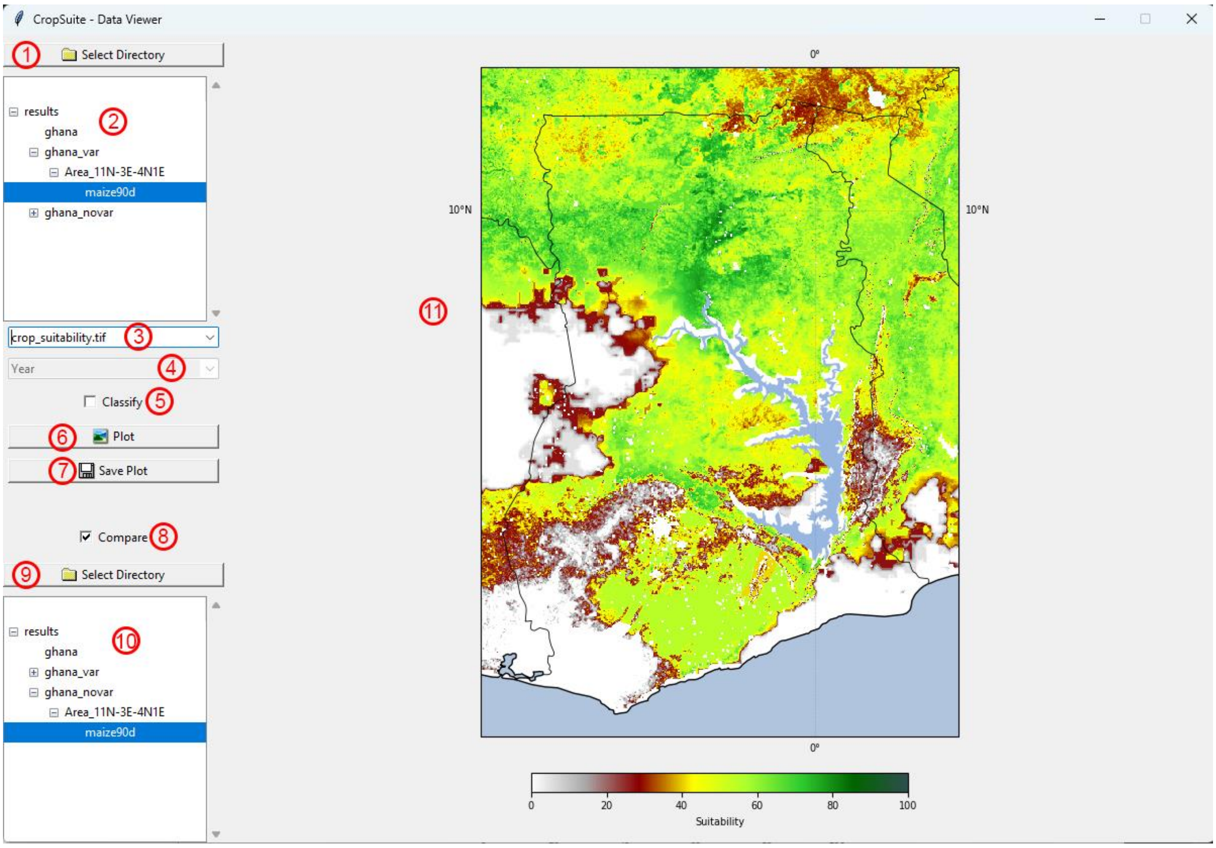
This opens the CropSuite user interface, which looks like this:



1. Open the data viewer (greyed out if no config.ini is specified) -> See 4.1
2. Open this user manual
3. Path to the currently selected config.ini file
4. Select a different config.ini file
5. Create a new config.ini file
6. Check whether the selected config.ini file matches the expected pattern and is readable
7. Open the menu to edit the options specified in config.ini -> See 4.2
8. Select crops to model -> See 4.3
9. Select static datasets -> See 4.5
10. Start the model run
11. Quit CropSuite
12. Open Preprocessing Window -> See 4.6

4.1. Data Viewer

With the integrated data viewer, all output files can be visualised and analysed and figures can be created.

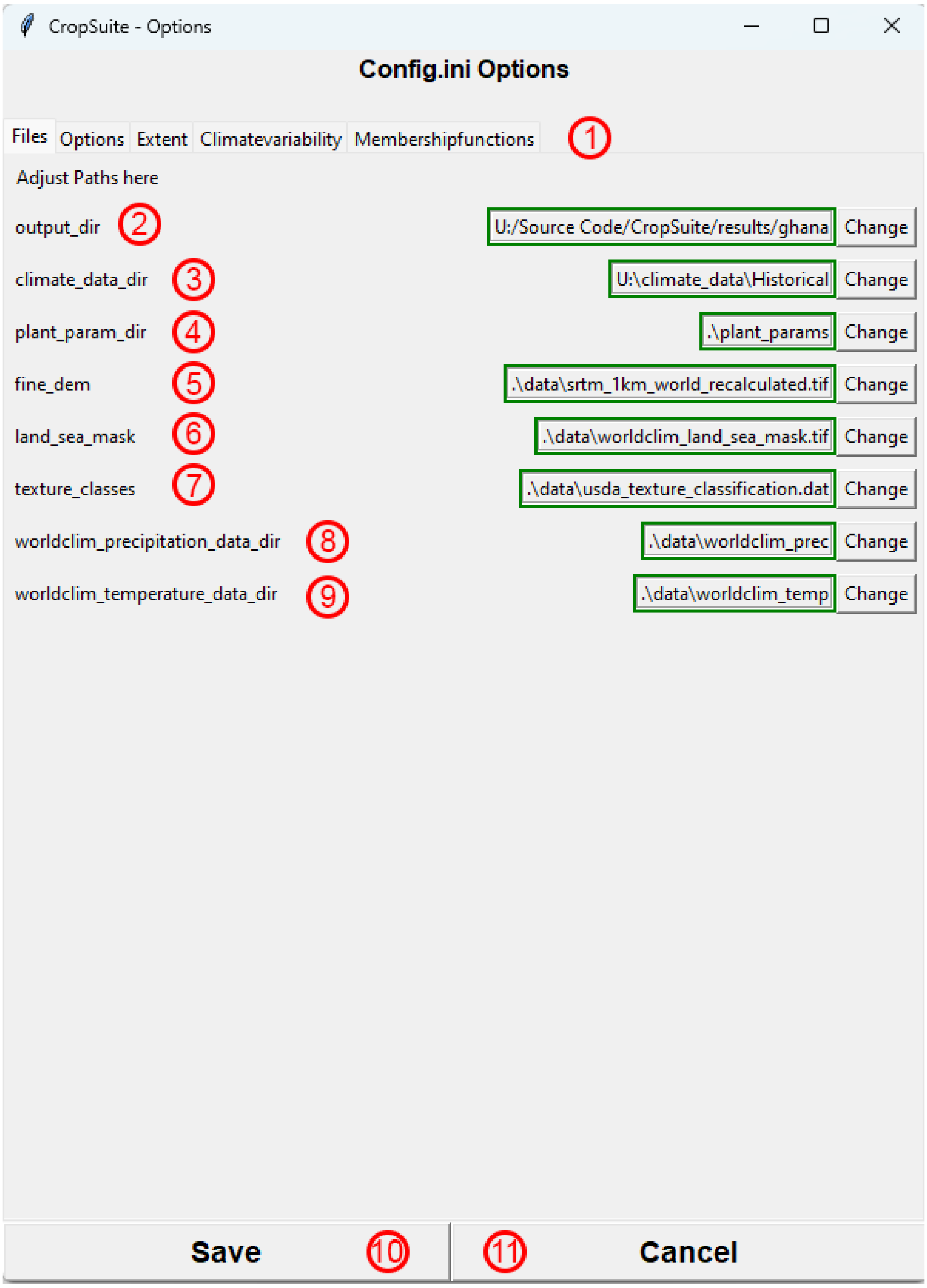


①	Select the directory in which the result files are located. The complete directory structure from the selected directory is always displayed. The output directory specified in config.ini is used by default.
②	Directory tree of the selected directory. To visualise data, the lowest level of the result directory must always be selected. Typically, these are the directories that are named like the crops, as they contain the result files.
③	Selection of the available result files that can be displayed.
④	For result files that are output for every day of the year (such as precipitation and temperature data), the time period to be displayed can be selected here.
⑤	Selection of whether the data should be divided into classes or whether the data should be displayed as continuous data.
⑥	Command to display the data with the selected settings.
⑦	Command to display the data with the selected settings and save as a .png file. The .png file can be found in the folder selected in the treeview above
⑧	Option to compare the data selected above with other datasets.
⑨	Selection of the directory in which the data for comparison is located
⑩	Lower treeview for selecting the comparison data. By default, the directory specified as the output directory in config.ini is also used.
⑪	Figure surface on which the desired figure is drawn

4.2. Config.ini Option Window

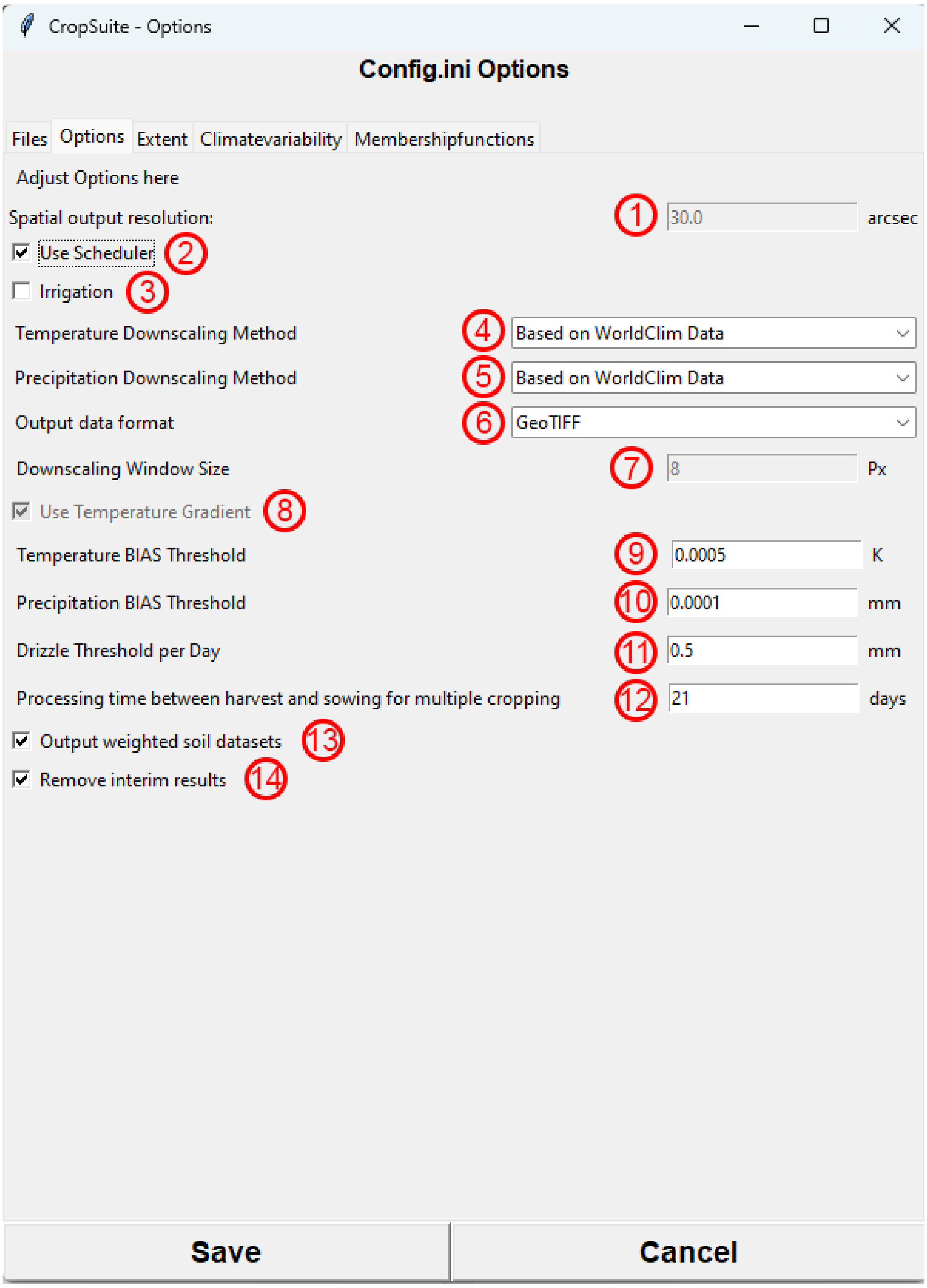
All settings that can be made in text form in config.ini can also be made via the Config window.

Please also refer to chapter 3.2.1 regarding all settings.



① Tab list of the various option pages
② Current output directory and the button to change it
③ Climate data directory in which the Avg_temp.tif and Avg_prec.tif are located
④ Directory in which the plant parameterization files are located
⑤ Path to the digital elevation model (DEM)
⑥ Path to the land-sea mask
⑦ Path to the texture class definition file
⑧ Directory containing the WorldClim precipitation data
⑨ Directory containing the WorldClim temperature data
⑩ Save the options and close the window
⑪ Close window without saving

The path entries are outlined in green if the required paths and files were found.


CropSuite - Options
Config.ini Options
Files
Options
Extent
Climatevariability
Membershipfunctions
Adjust Options here
Spatial output resolution:
30.0
arcsec
Use Scheduler
Irrigation
Temperature Downscaling Method
Based on WorldClim Data
Precipitation Downscaling Method
Based on WorldClim Data
Output data format
GeoTIFF
Downscaling Window Size
8
Px
Use Temperature Gradient
Temperature BIAS Threshold
0.0005
K
Precipitation BIAS Threshold
0.0001
mm
Drizzle Threshold per Day
0.5
mm
Processing time between harvest and sowing for multiple cropping
21
days
Output weighted soil datasets
Remove interim results
Save
Cancel
1
2
3
4
5
6
7
8
9
10
11
12
13
14

① Spatial resolution of the output files

② Activate automatic splitting of large areas

③ Consider irrigation (this will affect climate and crop suitability, optimal sowing dates, multiple cropping, suitable sowing days as well as climate variability)

④ Downscaling method for the temperature data

⑤ Downscaling method for the precipitation data

⑥ Output data format

⑦ When using height regression method: Size of the moving window

⑧ When using height regression method: Check for physical limits

⑨ Conservation of energy: Maximum sum of the temperature residuals

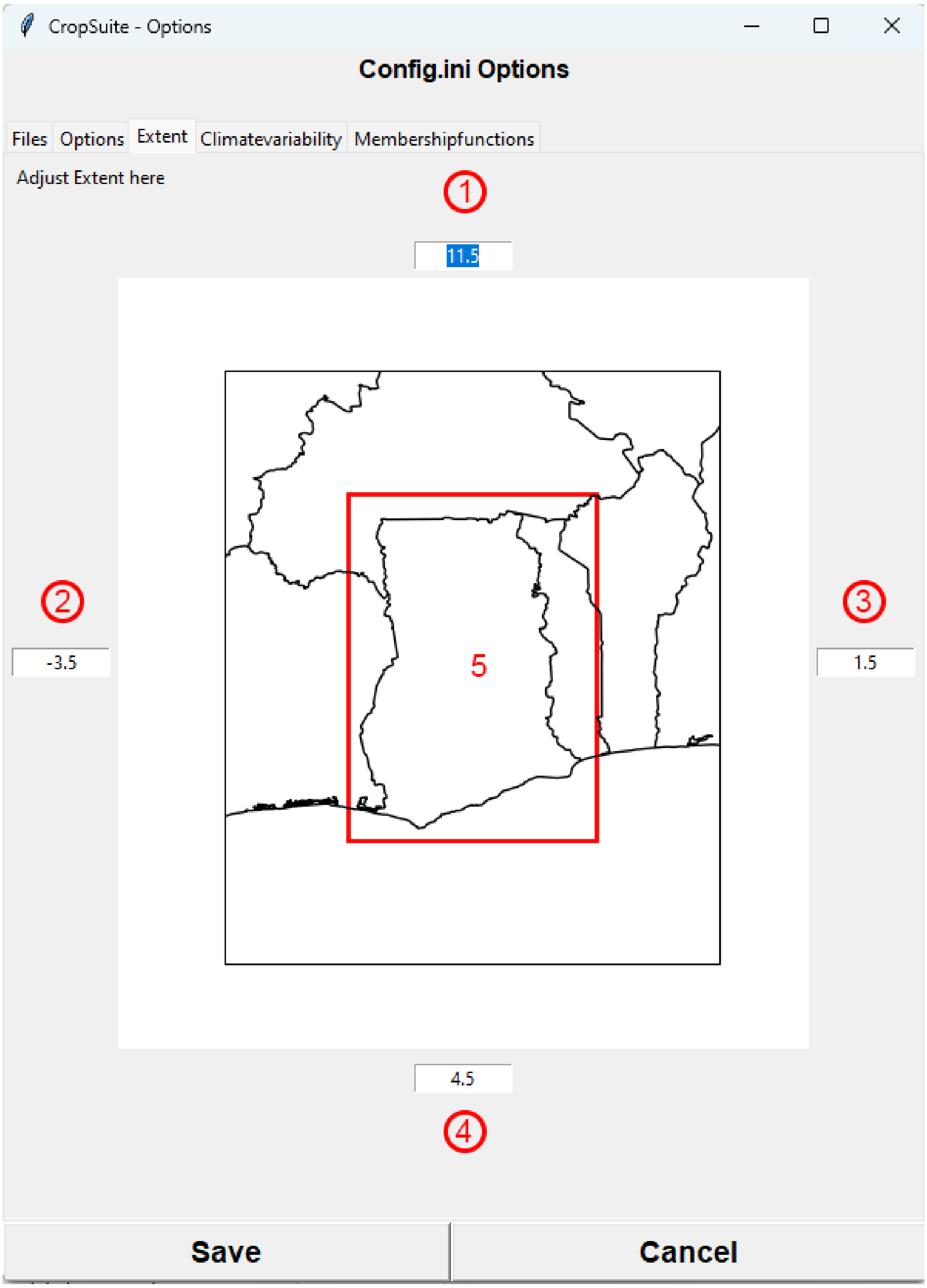
⑩ Conservation of energy: Maximum sum of the precipitation residuals

⑪ Threshold for drizzle precipitation

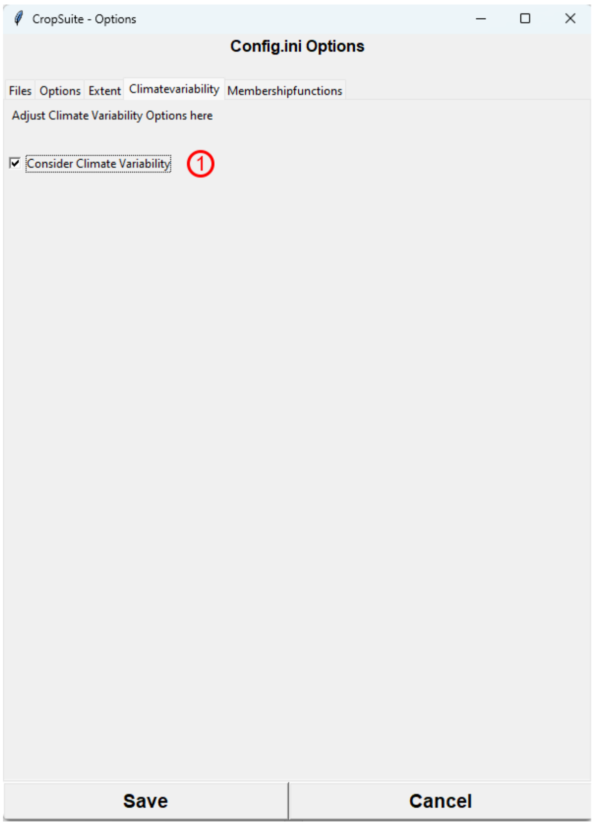
⑫ Number of days between harvest and reseeding with multiple cropping

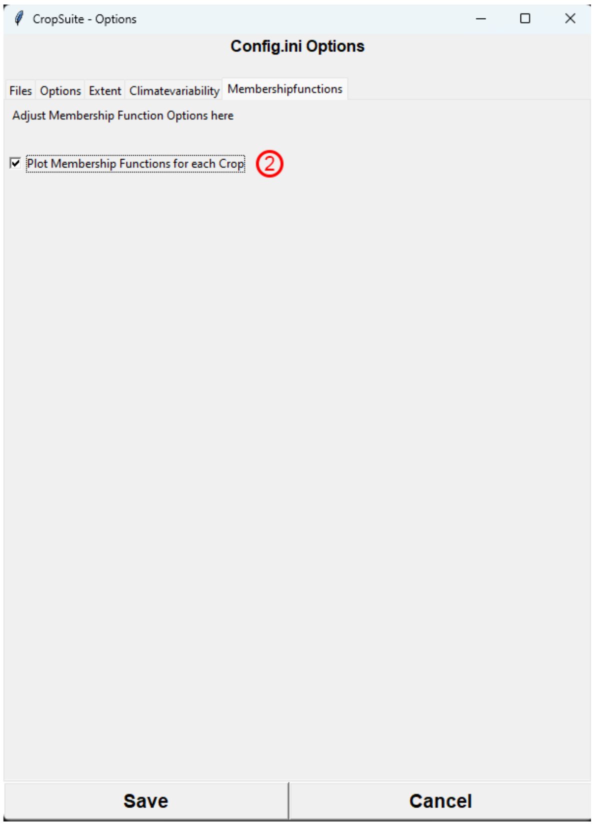
⑬ Output the weighted soil data as geotiff files

⑭ Remove unnecessary intermediate results


CropSuite - Options
Config.ini Options
Files
Options
Extent
Climatevariability
Membershipfunctions
Adjust Extent here
1
11.5
2
-3.5
3
1.5
5
4.5
4
Save
Cancel

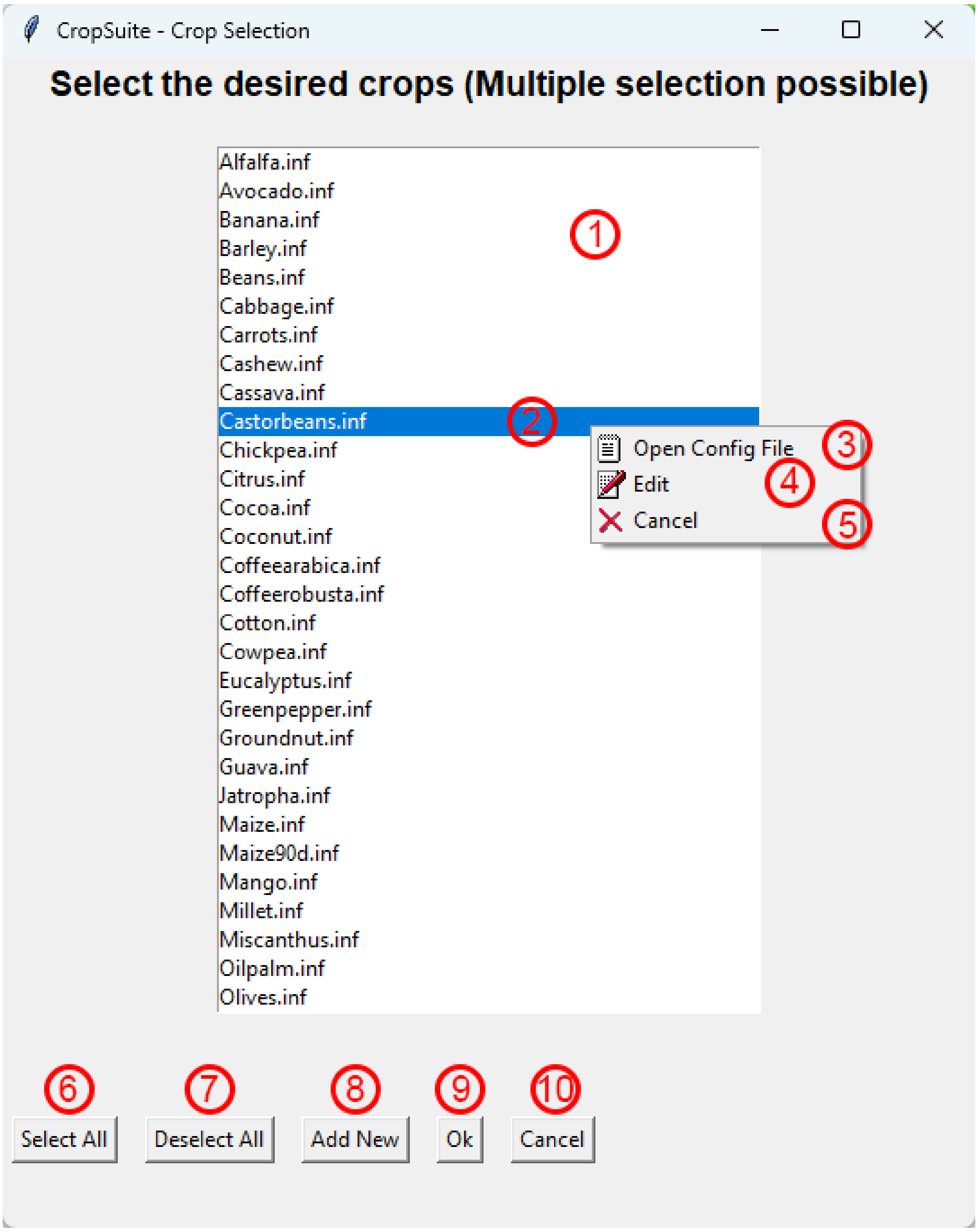
1. Extent of the modelling domain: Top Coordinate
2. Extent of the modelling domain: Left Coordinate
3. Extent of the modelling domain: Right Coordinate
4. Extent of the modelling domain: Bottom Coordinate
5. The map shows the currently selected domain





1. Option as to whether climate variability should be considered
2. Plotting of all parameterisation files

4.3. Window for selecting the crops to be modelled



① Selection list of currently available crops. Multiple selection of crops by a single left click is possible.

② Right-click on a crop to open the context menu

③ Context menu: Open parameterisation file as text file

④ Context menu: Switch to the editing dialogue for the respective crop -> See 4.4

⑤ Context menu: Close the menu

⑥ Select all available crops

⑦ Deselect all crops

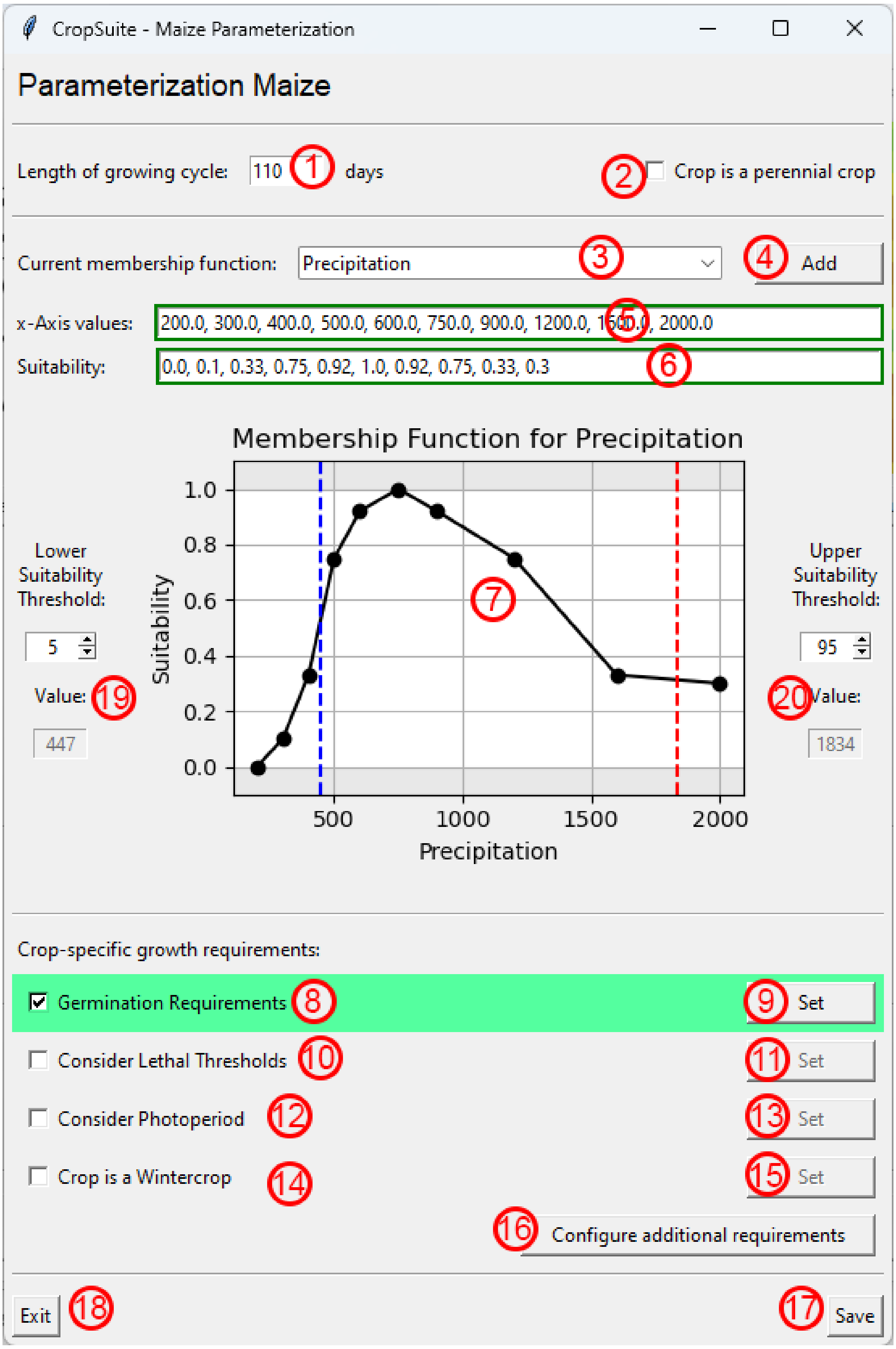
⑧ Add a new crop, not listed in the selection list

⑨ Confirm selection

⑩ Close the window

4.4. Window for editing crop specific parameters

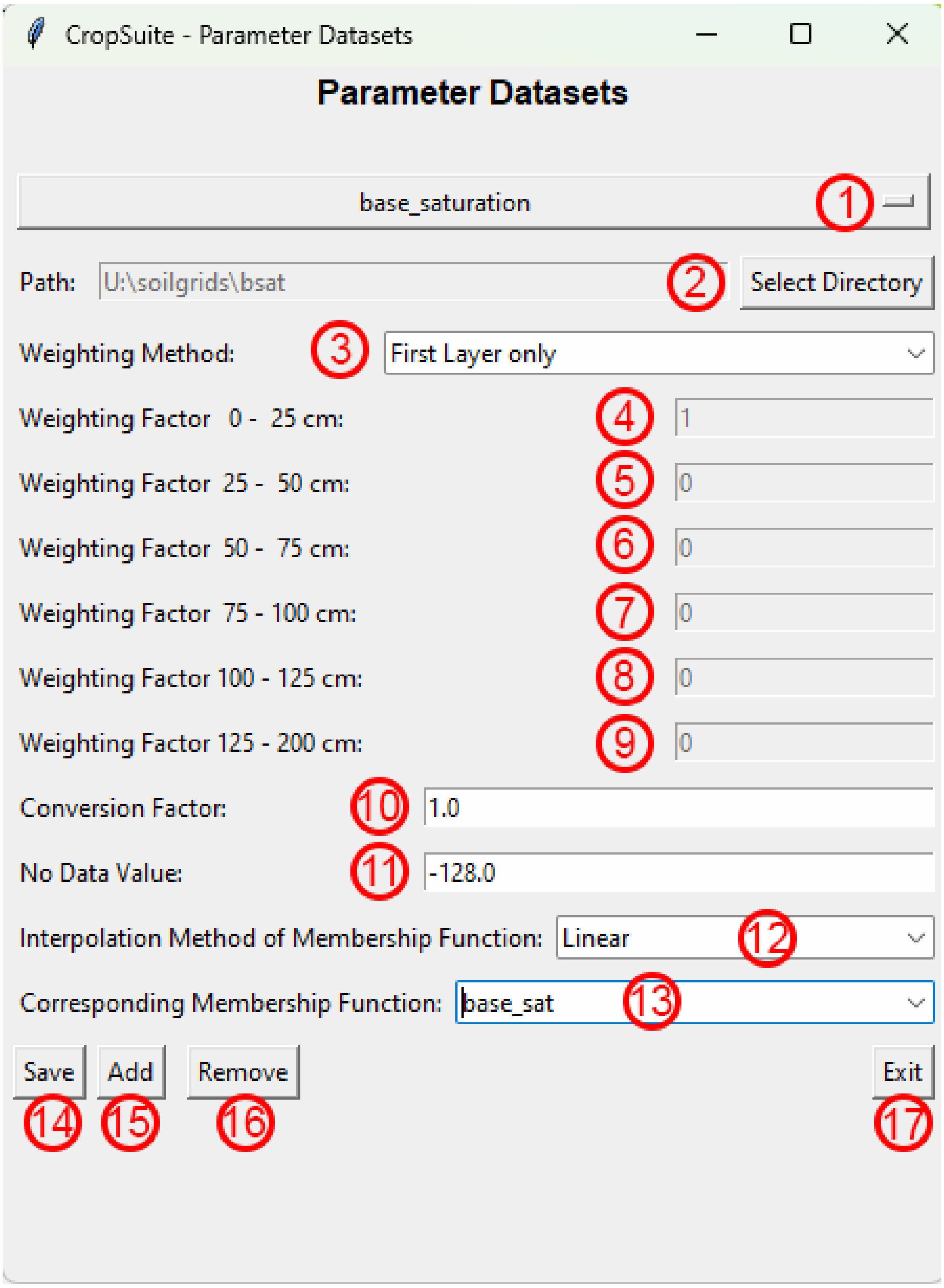
Please refer to chapter 3.2.2



1	Length of the growing cycle in days
2	Crop is a perennial crop
3	Select membership function to adjust
4	Add a new membership function
5	Membership-Function: Adjust x-axis values
6	Membership-Function: Adjust y-axis values (suitability value)
7	Control plot of the current membership function
8	Enable consideration of germination requirements
9	Set germination requirements
10	Enable Consideration of lethal
11	Set lethal thresholds
12	Enable consideration of photoperiod sensitivity
13	Set requirements for photoperiod sensitivity
14	Crop is a wintercrop
15	Set vernalization parameters required if crop is a wintercrop
16	Add additional crop requirements
17	Save the options and close the window
18	Close without saving
19	Set suitability lower variability limit. The selected value is displayed in 7 as blue vertical line
20	Set suitability of upper variability limit. The selected value is displayed in 7 as red vertical line

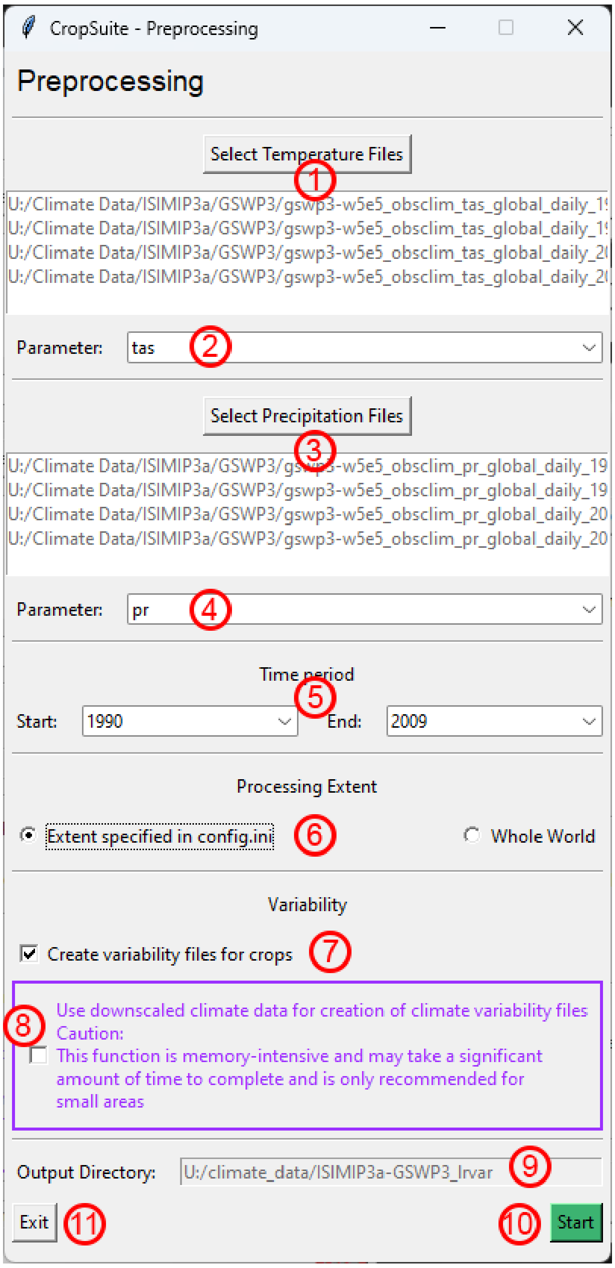
4.5. Window for editing parameter layer

Please also refer to chapter 3.2.1



①	Select parameter dataset to adjust
②	Current path to dataset directory and button to change directory
③	Weighting method of layer
④-⑨	Different weighting factors for different soil depths
⑩	Conversion factor if required by dataset
⑪	No data value if not in metadata of dataset
⑫	Interpolation method of corresponding membership function
⑬	Associated membership function in the crop parameterisations
⑭	Save settings for current parameter
⑮	Add new parameter dataset
⑯	Remove parameter dataset
⑰	Close window

4.6. Preprocessing window



1. Select temperature NetCDF4 files, the selected files are listed below
2. Select the parameter which includes the daily average temperature
3. Select temperature NetCDF4 files, the selected files are listed below
4. Select the parameter which includes the daily average temperature
5. Select the timerange for which the preprocessing should be executed
6. Select the processing extent. ‘Whole World’ is greyed out if 8 is activated
7. Calculation of climate variability files for each crop

⑧	Calculation of climate variability files based on downscaled climate data. This is only recommended for very small processing extents, as it takes up a lot of memory and computing time. Is greyed out if 'whole world' is selected as the processing extent.
⑨	Climate data directory selected in config.ini
⑩	Start the preprocessing process
⑪	Exit this window

5. Running the Model

To start CropSuite, there are various options. Once all the necessary files are defined, you can initiate the program by double-clicking on the CropSuite.py file.

Alternatively, especially on Linux systems, it is recommended to launch the program through the command line. To do this, navigate to the program folder and execute the following command in the console:

```
>> python CropSuite.py
```

For better usability, CropSuite also comes with an easy-to-use user interface. All necessary settings can be made here, new config files created, plant parameterisation files created, checked and adjusted and additional static layers defined. The user interface also offers an analysis tool, the Data Viewer. This can be used to visualise and compare results and generate finished figures.

On Windows systems, a double-click on the CropSuite_GUI.py file is sufficient to open the user interface. Once all options have been set, the model is started by clicking on Start.

On Linux and MacOS systems, it makes sense to also start the GUI via the terminal by using:

```
>> python CropSuite_GUI.py
```

IMPORTANT

Depending on the type of Python installation, it may be necessary to call the program in the command line with one of the following commands:

```
>>python CropSuite.py
```

or

```
>>python3 CropSuite.py
```

In the following examples, the variant with `>>python` is used.

Both CropSuite and the 'climate_data_preprocessing.py' script both offer support for a range of command line commands.

The CropSuite.py presents a silent mode feature, accessed by the command line parameter '-silent'. Moreover, the option to specify any 'config.ini' file is available using the '-config' parameter. These options can be used independently of each other.

```
>> python CropSuite.py -silent
>> python CropSuite.py -silent -config “.\config.ini”
>> python CropSuite.py -config “.\config.ini”
```

Additionally, the pre-processing script 'climate_data_preprocessing.py' provides the options to directly continue processing by calling CropSuite.py. Furthermore, it is also possible to handover a custom config.ini file:

```
>> python climate_data_preprocessing.py
>> python climate_data_preprocessing.py -continueprocessing
>> python climate_data_preprocessing.py -continueprocessing -config “.\config.ini”
>> python climate_data_preprocessing.py -config “.\config.ini”
```

For Unix systems, it is often beneficial to suppress terminal output, redirecting it to a file, and defining the Python process as a background task. This can be achieved, for instance, as shown below:

```
>> nohup python CropSuite.py -silent &
```

IMPORTANT

CropSuite also supports the use of a slurm environment for task scheduling and has also been optimised for use on high performance computers (HPC).

6. Troubleshooting

Comprehensive error handling has been implemented, as well as a log file that provides further information. In case of an error, the following files are relevant:

- config.ini
- Error Message
- Plant parameterization files

7. Appendix

7.1. Example config.ini

```
[files]
output_dir = \results\output
climate_data_dir = ..\climate_data\Historical
plant_param_dir = .\plant_params
fine_dem = .\data\srtm_1km_world_recalculated.tif
land_sea_mask = .\data\worldclim_land_sea_mask.tif
texture_classes = .\data\usda_texture_classification.dat
worldclim_precipitation_data_dir = .\data\worldclim_prec
worldclim_temperature_data_dir = .\data\worldclim_temp

[options]
use_scheduler = y
irrigation = 0
precipitation_downscaling_method = 2
temperature_downscaling_method = 2
output_format = geotiff
output_all_startdates = y
output_grow_cycle_as_doy = y
downscaling_window_size = 8
downscaling_use_temperature_gradient = y
downscaling_dryadiabatic_gradient = 0.00976
downscaling_saturation_adiabatic_gradient = 0.007
downscaling_temperature_bias_threshold = 0.0005
downscaling_precipitation_bias_threshold = 0.0001
downscaling_precipitation_per_day_threshold = 0.5
output_all_limiting_factors = y
remove_interim_results = y
remove_downscaled_climate = n
output_soil_data = y
multiple_cropping_turnaround_time = 21

[extent]
upper_left_x = -3.5
upper_left_y = 11.5
lower_right_x = 1.5
lower_right_y = 4.5

[climatevariability]
consider_variability = y

[membershipfunctions]
plot_for_each_crop = y

[parameters.base_saturation]
data_directory = ..\soilgrids\bsat
weighting_method = 0
weighting_factors = 1.0,0.0,0.0,0.0,0.0,0.0
conversion_factor = 1.0
no_data = -128.0
interpolation_method = 0
rel_member_func = base_sat

[parameters.coarse_fragments]
```

```
data_directory = ..\soilgrids\cfvo
weighting_method = 2
weighting_factors = 2,1.5,1,0.75,0.5,0.25
conversion_factor = 10
interpolation_method = 0
rel_member_func = coarsefragments

[parameters.clay_content]
data_directory = ..\soilgrids\clay
weighting_method = 2
weighting_factors = 2,1.5,1,0.75,0.5,0.25
conversion_factor = 10
interpolation_method = 0
rel_member_func = texture

[parameters.gypsum]
data_directory = ..\soilgrids\gyps
weighting_method = 0
weighting_factors = 2,1.5,1,0.75,0.5,0.25
conversion_factor = 10
interpolation_method = 0
rel_member_func = gypsum

[parameters.pH]
data_directory = ..\soilgrids\ph
weighting_method = 2
weighting_factors = 2.0,1.5,1.0,0.75,0.5,0.25
conversion_factor = 10.0
interpolation_method = 0
rel_member_func = ph

[parameters.salinity]
data_directory = ..\soilgrids\sal
weighting_method = 0
weighting_factors = 2,1.5,1,0.75,0.5,0.25
conversion_factor = 1
interpolation_method = 0
rel_member_func = elco

[parameters.sand_content]
data_directory = ..\soilgrids\sand
weighting_method = 2
weighting_factors = 2,1.5,1,0.75,0.5,0.25
conversion_factor = 10
interpolation_method = 0
rel_member_func = texture

[parameters.soil_organic_carbon]
data_directory = ..\soilgrids\soc
weighting_method = 1
weighting_factors = 2,1.5,1,0.75,0.5,0.25
conversion_factor = 100
interpolation_method = 0
rel_member_func = organic_carbon

[parameters.sodicity]
data_directory = ..\soilgrids\sod
weighting_method = 0
weighting_factors = 2,1.5,1,0.75,0.5,0.25
conversion_factor = 1
interpolation_method = 0
```

```
rel_member_func = esp

[parameters.soildepth]
data_directory = ..\soilgrids\soildepth
weighting_method = 0
weighting_factors = 2,1.5,1,0.75,0.5,0.25
conversion_factor = 100
interpolation_method = 0
rel_member_func = soildepth
```

7.2. Example parameterization file

```
name =                      winterwheat
growing_cycle =             115
temp_vals =                 7,8,10,12,15,18,20,23,25,30
temp_suit =                 0,0.1,0.33,0.75,0.92,1,0.92,0.75,0.33,0
prec_vals =                 170,200,250,350,450,700,1000,1250,1500,1750
prec_suit =                 0,0.1,0.33,0.75,0.92,1,0.92,0.75,0.33,0
slope_vals =                0,2,4,8,16
slope_suit =                1,0.92,0.75,0.33,0
soildepth_vals =            0.2,1.0
soildepth_suit =            0,1
texture_vals =              1,2,3,4,5,6,7,8,9,10,11,12,13
texture_suit =              0,0,1,0,1,1,1,0.92,0.92,0.75,0.33,0,0.25
coarsefragments_vals =      0,3,15,35,55
coarsefragments_suit =      1,0.92,0.75,0.33,0
gypsum_vals =               0,3,5,10,20
gypsum_suit =               1,0.92,0.75,0.33,0
base_sat_vals =             0,35,50,80,100
base_sat_suit =             0,0.33,0.75,0.92,1
ph_vals =                   5.2,5.6,6,6.5,7,7.5,8.2,8.3,8.5
ph_suit =                   0,0.33,0.75,0.92,1,0.92,0.75,0.33,0
organic_carbon_vals =       0,0.5,1.0,1.5,2
organic_carbon_suit =       0,0.33,0.75,0.92,1
elco_vals =                 0,1,3,5,10
elco_suit =                 1,0.92,0.75,0.33,0
esp_vals =                  0,15,20,35,45
esp_suit =                  1,0.92,0.75,0.33,0
freqcropfail_vals =         0,0.025,0.05,0.075,0.1,0.125,0.15,0.175,0.2,0.225,0.25
freqcropfail_suit =         1,0.98,0.95,0.88,0.73,0.5,0.27,0.12,0.05,0.02,0
photoperiod =               1
minimum_sunlight_hours =  10.0
maximum_sunlight_hours =  15.0
lethal_thresholds =         1
lethal_min_temp =           0
lethal_min_duration =       3
lethal_max_temp =           40
lethal_max_duration =       5
prec_req_after_sow =        50
prec_req_days =             30
consecutive_dry_days =      21
wintercrop =                y
vernalization_effective_days = 50
vernalization_tmax =        7
vernalization_tmin =        1
frost_resistance_days =     3
frost_resistance =          -20
days_to_vernalization =     30
```

8. Changelog

Date	to Version	Changes
07.06.2023	0.1	Initial Development Version
27.07.2023	0.2	Development Version • Added Error Handling • Added Irrigation Module • Added NetCDF4 Support • Restructured config.ini. • Possibility to freely select the number of parameters sets. • Plots of the membership functions improved. • Use of dictionaries instead of lists for formulas and plant parameterisations. • Added further memory usage optimisation for the Climate Suitability module. • Converted basic input files from binary rasters to geotiff • Added free selection of conversion factors. • Added selection option of dataset and membership function. • Logfile added. • Extensive program output added. • Parts of the code restructured. • Added automatic splitting of the processing area for OOM prevention. • Fixed several bugs • Added verification for writing intermediate results • Silent mode for Background processing added • Fixed incorrect texture class determination • Optimized the memory usage of the climate suitability module • Improved handling of NaN values in soil layers • Generally applicable lethal temperature of 40 °C over the Growing Period added
04.10.2023	0.3	Development Version • Converted Climate Suitability Module for Integer 8-bit computing for better performance and memory usage • Added climate variability module to consider extremely high and low temperatures as well as precipitation • Fixed path error • Rebuilding npy-files when adding crops • Added option to output all individual suitability values of the respective input layers as geotiff with n bands • Added option to calculate best start dates for multiple cropping
05.11.2023	0.5	Development Version • Algorithm for finding multiple optimal sowing dates when using multiple cropping improved • Bugs fixed • Output files renamed for better understandability • Plot function of parameterization files improved

		• Added combined crop failure dataset consideration instead of four single datasets • Added preprocessing script for creating combined crop failure dataset • General performance improved • Improved splitting algorithm for large areas • Post-processing script for merging split areas added • Plant parameterizations added • Optimizations for reduced memory usage • Improved file handling • Performance-optimized hard disk access
25.12.2023	0.6	Development Version • Added cleanup option • Changed consideration of climate variability • Added preprocessing script adjusted to EiA Server structure • Added performance value • Added option to use read-only climate data directories • Added possibility of defining custom config.ini files • Added possibility for using Bash Scripts • Improvement of the method for reading soil properties data
14.01.2024	0.7	1. Production Version • Crop failure variability added as a limiting factor • Conversion problems between climate and crop suitability fixed • Unique value range for suitability • Intermediate results changed from numpy binary format to tif • Lempel-Ziv-Welch algorithm for intermediate results added • Various changes to improve performance • Improving the integration of crop failure frequency • Crop failure frequency added as a separate limiting factor • Change of the data type of the downscaling to 16-bit integers algorithm for better performance • Improved checking of input files for defective files • Multiprocessing improved to avoid out-of-memory errors • Added compression to all output files • Performance enhancements • Various bug fixes
30.01.2024	0.8	1. Production Version • Fixed error when reading soil parameter data • Performance improvements • Config file check added • Improved checking of the required Python libraries • Removed no_cols and no_rows from config.ini • Multiprocessing improved • Improved workload distribution across different cores • Memory optimisation for climate suitability • Adaptation of processing sequence for improved performance • Use of multithreading for bilinear interpolations • Length of growing cycle on the parameterisation plots
03.02.2024	0.85	1. Production Version • Enhanced multiple cropping calculations • Corrected determination of multiple cropping optimal sowing dates

		• Parallel processing of datasets with and without crop failure frequency • Enhanced determination of limiting factor in unsuitable areas • Improved Nan-handling of soil datasets • Output of soil suitability • Performance optimizations
04.08.2024	0.9	• SLURM support added • Performance optimizations for MacOS systems • Integrated postprocessing script to adjust shape of multiple cropping dates • Enhanced file writing to decrease output file size • Added high performance computing support • Enhanced output of all limiting factors • Improved clearness of the terminal output • Added correct handling of NoData values added when all limiting values are output • Downscaling performance optimizations • Added different downscaling methods for temperature and precipitation • Added downscaling of temperature and precipitation based on WorldClim factors • Restructured Downscaling and Climate Suitability Module • Added support of cloud optimized GeoTiffs (COG) as output format • Improved merging of netcdf4 and cog files • Support for netcdf4 climate input data added • Specification of wintercrops added • Vernalization enhanced • Frost resistance added • Separate calculation of optimum sowing date with and without vernalisation added • Improved output of optimum sowing dates • Improved performance of vernalization calculation • User interface added • Lethal temperature thresholds added • Crop specific precipitation requirements after sowing added • Consideration of photoperiod added • Data Viewer added • Crop specific number of consecutive dry days added to consider droughts • Output of weighted soil datasets added • Crop specific number of days for time range between harvest and sowing added for multiple cropping
14.11.2024	0.91	• Correction of the WorldClim downscaling method to additive factors for temperature • Vernalisation: No weighting of the suitabilites, but common weighted temperature and precipitation mean value for the determination of suitability • Improved determination of sowing dates for winter crops • Performance improvements when writing downscaled climate data • Improved downscaling performance • Restructuring of the downscaling algorithm: No more strip-by-strip calculation, but day-by-day calculation to avoid errors at the joints of the strips

		• Free selection of the map projection in the viewer • Improved map display in the viewer • Possibility to set the extent in the viewer added • Universal preprocessing of climate data added to CropSuite • Performance improvements of daily downscaling for creation of climate variability files for each crop • Winter crops: improved determination of suitability when taking climate variability into account • Dynamic calculation of the slope inclination based on the terrain model added • New improved Crop Parameterization window: Object-oriented implementation • Limit values selectable in Crop Parameterization for the consideration of climate variability • Freely selectable crop-specific temperature and precipitation limits added • More robust import commands for different system structures • Improved multiprocessing for downscaling during preprocessing • Adaptation of the interpolation method to the RegularGridInterpolator • Improved NA handling during downscaling in preprocessing • Improved parameter window: object-oriented implementation • Fixed crash when incorrect plant_param path was specified • Added used Python package version numbers to manual • Major performance and program structure enhacements • Added multiprocessing to climate data downscaling • Removed usage of .npy interim files • Data Viewer: Colormap freely selectable • Corrected output path for results without consideration of climate variability • Corrected output of aggregated soil datasets • Added option to remove the downscaled climate files • Adjusted worldclim downscaling method • Changed output directory for downscaled climate files • Renamed output files for sowing if crop is a wintercrop
12.12.2024	1.0	• Added Limitation Analyzer • Enhanced downscaling performance • Adjusted preprocessing downscaling worldclim method • Added multiprocessing to downscaling using the nearest neighbour, bilinear and height regression method • Added multiprocessing to downscaling of rrpcf data • Changed downscaling of rrpcf from stripwise to daywise • Corrected bilinear and nearest neighbour downscaling method • Added support for legacy created rrpcf files

Intentionally left white