# CropSuite

Version 1.3.1

Effective 17-04-2025

# User Manual

Matthias Knüttel
Florian Zabel

Cite As:

Knüttel, M., Zabel, F. (2025): CropSuite User Manual. DOI: 10.5281/zenodo.14196314

**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. The Fortran code was translated into Python and at the same time several extensions and improvements were made to improve usability and expand the possibilities.

## Related Works

A detailed model description entitled “CropSuite v1.0 - a comprehensive open-source crop suitability model considering climate variability for climate impact assessment” has been published and is available at the following link:

https://doi.org/10.5194/gmd-18-1067-2025

This publication presents CropSuite v1.0, a comprehensive and fully open-source modeling framework designed to assess crop suitability under both current and future climate conditions. One of the key innovations of CropSuite is its explicit integration of interannual climate variability, a factor that is often underrepresented in traditional crop suitability models. By accounting for year-to-year fluctuations in climate, CropSuite provides a more robust basis for climate impact assessments, particularly in regions with high climate variability such as sub-Saharan Africa.

The model supports the evaluation of suitability for 48 different crops and includes modules for simulating multiple cropping systems and identifying optimal sowing periods. CropSuite is intended to support researchers, policymakers, and stakeholders in making informed decisions regarding agricultural planning and climate adaptation strategies.

## 2. Automatic setup of the Python environment and starting CropSuite

To ensure compatibility and avoid version conflicts, CropSuite includes setup scripts for both Windows and macOS. These scripts automatically install the correct Python version (if needed) and all required Python packages with the exact versions compatible to CropSuite. It is recommended to use the provided setup script to install and start CropSuite. This guarantees that all dependencies are correctly configured for optimal performance and model accuracy. If the setup script does not work, the installation can also be done manually (see **Fehler! Verweisquelle konnte nicht gefunden werden.. Fehler! Verweisquelle konnte nicht gefunden werden.**).

**IMPORTANT**

Unzip the CropSuite.zip before starting for the first time!

### Windows

**IMPORTANT**

Please make sure the App Execution Alias for Python is deactivated:

Go to Settings → Apps → Advanced App Settings → App execution aliases

Look for App Installer python.exe and python3.exe and deactivate them.


**IMPORTANT**

To start, simply double-click on “run_cropsuite_win.bat”.

### MacOS

**IMPORTANT**

On MacOS, change to the unzipped CropSuite directory with ‘cd’, then start CropSuite with

```
>> bash run_cropsuite_macos.sh
```

### Starting CropSuite manually after installation

If you prefer not to use the supplied batch (.bat) or bash (.sh) script to run CropSuite, you can start it manually via Python. In this case, you must first activate the Python virtual environment.

Use the appropriate command for your operating system

On Windows (Command Prompt):

```
>> .cropsuite-venv\Scripts\activate
```

On macOS or Linux (Terminal):

```
>> source ./cropsuite-venv/bin/activate
```

After activation, run CropSuite with:

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

**IMPORTANT**

Make sure you are in the CropSuite root directory when running these commands!

### Starting CropSuite without an active network connection

A network connection is required during the installation of CropSuite. The setup script downloads necessary Python packages and data files from the internet.

If no active internet connection is detected, the setup process will automatically cancel to avoid an incomplete or faulty installation.

However, once CropSuite has been successfully installed, a network connection is no longer mandatory.

The startup script checks internet availability and, if none is found, automatically switches to offline mode. In offline mode, CropSuite remains fully functional, and all features can be used without limitations.

# 3. Manual setup of the Python environment and starting CropSuite

If the automatic setup of the Python environment required for CropSuite fails, the environment can be set up manually using the following instructions.

## 3.1. Prerequisites

To run CropSuite, your system must meet the following requirements:

Python 3.9, 3.10, 3.11, 3.12 or 3.13

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.
- pyproj: Repojecting raster datasets
- 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.

## 3.2. Tested version numbers

Python Version 3.9, 3.10, 3.11, 3.12 and 3.13.
Version 3.13 still may have incompatibilities with some of the used packages.

| *Package* | *Tested version numbers* | *Remarks* |
|---|---|---|
| *numpy* | 1.24.3, 1.25.0, 1.26.4, 2.2.4 | |
| *scipy* | 1.9.3, 1.11.2, 1.14.1, 1.15.1, 1.15.2 | RegularGridInterpolator instead of interp2d in newer versions. interp1d deprecated in newer versions. |
| *rasterio* | 1.3.9, 1.3.10, 1.3.11, 1.4.1, 1.4.3 | |
| *matplotlib* | 3.6.3, 3.9.2, 3.10.1 | |
| *xarray* | 2023.6.0, 2024.10.0, 2025.3.1 | |
| *numba* | 0.60.0, 0.61.0, 0.61.2 | |
| *rio-cogeo* | 5.3.3, 5.3.6, 5.4.1 | |
| *cartopy* | 0.23.0, 0.24.1 | |
| *dask* | 2023.3.2, 2024.10.0, 2025.3.0 | |
| *netCDF4* | 1.6.4, 1.7.2 | |
| *pillow* | 10.2.0, 10.4.0, 11.0.0, 11.1.0 | |
| *psutil* | 7.0.0 | |
| *pyproj* | 3.7.0, 3.7.1 | |
| *scikit-image* | 0.25.1, 0.25.2 | Alternative interpolation method |
| *netCDF4* | 1.6.4, 1.7.2 | |

**IMPORTANT**

macOS systems often come with pre-installed versions of certain packages, which may not be fully compatible with your specific requirements. It is therefore recommended to verify the installed version and install a compatible one if needed.

Example for scipy:

Check version:

```
>> python3 -c "import scipy; print(scipy.__version__)"
```

Uninstall existing version:

```
>> pip uninstall scipy
```

Install compatible version:

```
>> pip install scipy==1.15.1
```

Check version:

```
>> python3 -c "import scipy; print(scipy.__version__)"
```

### 3.3. 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

### 3.4. Installation on newer Linux distributions

Since Ubuntu (from version 23.04) does not allow system-wide pip installations outside the package management by default, there are two recommended ways to install the required Python packages system-wide:

First check for each package whether it is available via the Ubuntu package management:

```
>> sudo apt update
>> sudo apt install python3-[package_name]
```

If the package is available, it will be installed. If not, the package must be installed as follows:

Since some packages were installed by the Debian/Ubuntu package management, they are locked. Since some modifications must be made anyway, this lock must be circumvented as follows:

```
>> sudo pip3 install --break-system-packages --ignore-installed typing_extensions
```

The following command must be executed for each package:

```
>> sudo pip3 install --break-system-packages --ignore-installed [package-name]
```

**IMPORTANT**

pip is often not installed by default. The following command must be executed first:

```
>> sudo apt-get install python-pip
```

### 3.5. 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==0.24.1 dask==2025.3.0 "dask[distributed]" numpy scipy==1.15.2 matplotlib==3.10.1 netCDF4==1.7.2 rasterio==1.4.3 psutil==7.0.0 pyproj==3.7.1 rio-cogeo==5.4.1 numba==0.61.2 scikit-image==0.25.2 tk pillow==11.1.0 xarray==2025.3.1
```

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 pyproj rio-cogeo numba scikit-image 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.

# 4. 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:

## 4.1. Elevation Data

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. The elevation data is required for using the height regression downscaling method for temperature data.

## 4.2. 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.

## 4.3. Climate Data

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

CropSuite can generate the required input data using its built-in preprocessing tool from NetCDF4 datasets such as the ISIMIP data, which includes daily mean temperature values (tas) and daily precipitation values (pr). Alternatively, custom-made GeoTIFF or NetCDF4 files named Temp_avg.tif/nc and Prec_avg.tif/nc can be used. These files must contain temperature data in °C (tas) and precipitation data in mm (pr) for each day of the year. The climate data must have the dimensions (365, y, x), where 365 represents the days of the year, and y and x represent the spatial dimensions.

The temperature data provides the daily mean temperature in °C for every day of the year, while the precipitation data provides the daily average totals in mm for each day. It is important to note that the temperature and precipitation data must have the same dimensions and spatial resolution. As of now, a fixed file name is still expected for the climate data: the temperature data should be in ‘Temp_avg.tif’ / ‘Temp_avg.nc’, and the precipitation totals in ‘Prec_avg.tif’ / ‘Prec_avg.nc’.

### 4.4. Variability Files

The variability files are also created by the built-in preprocessing tool. For this purpose, daily mean daily temperature values and daily precipitation amounts are required as input data for a longer period (20 or 30 years are recommended).

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.

### 4.5. 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.

### 4.6. 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:

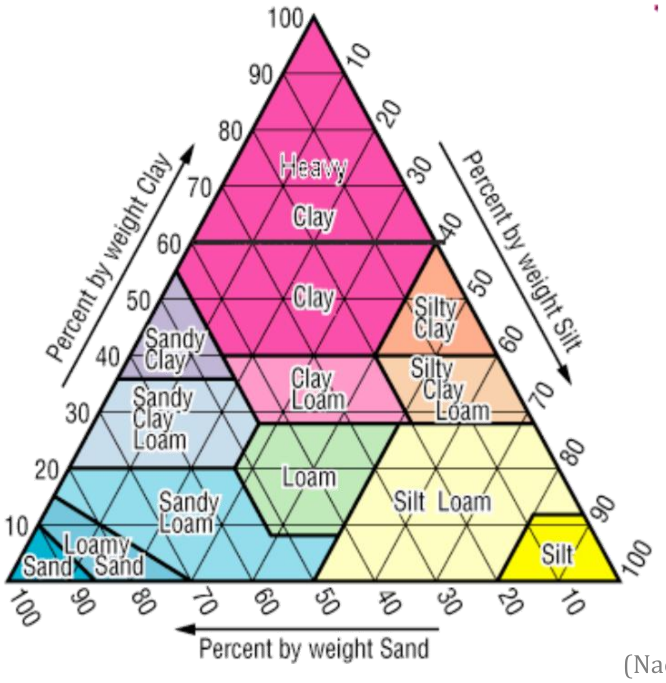
| Name | Unit | Short Description |
|---|---|---|
| Base Saturation | % | Proportion of essential nutrient ions on soil surfaces |
| Coarse Fragments | Vol.-% | Large, non-soil particles present in the soil, affecting drainage and root growth |
| Clay Content | % | Clay content of the soil, is converted 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 nearest neighbour 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! Data available in other projections (e.g. SoilGrids) must be re-projected beforehand!

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 |

## 4.7. Option Files

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

### 4.7.1. 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]<br>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 |

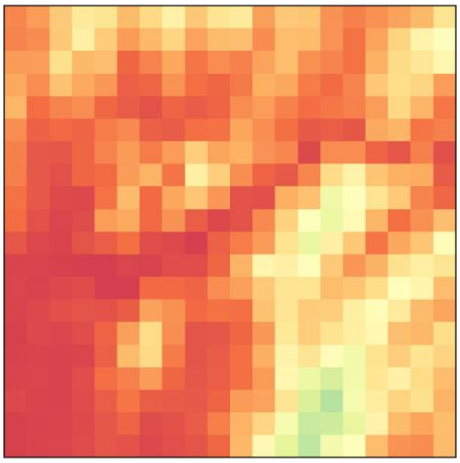
By default, CropSuite offers various options for downscaling the climate data, which can be specified in config.ini.

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

For temperature data:

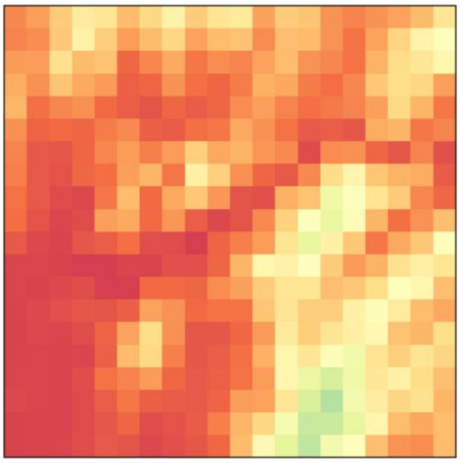
Coarse Data

-



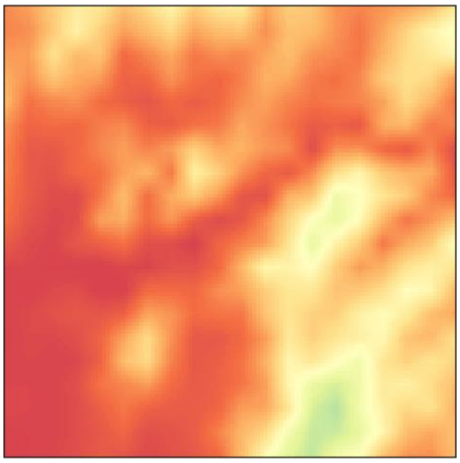
Nearest Neighbour

0



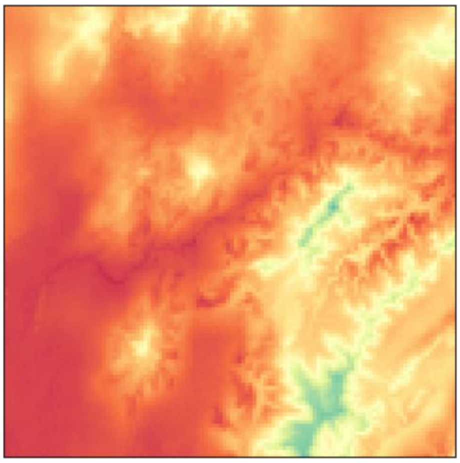
Bilinear Interpolation

1



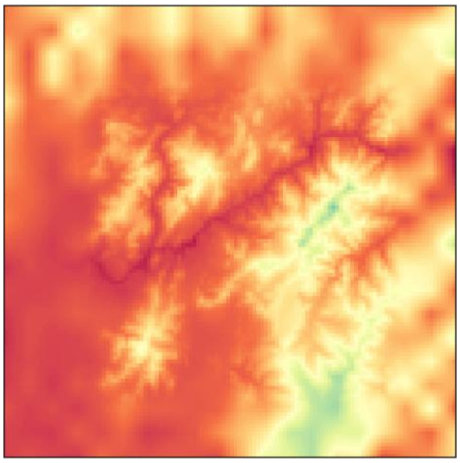
Factors based on WorldClim data

2



Height Regression

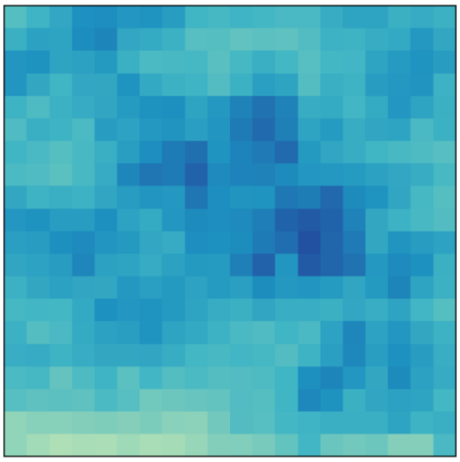
3



For precipitation data:

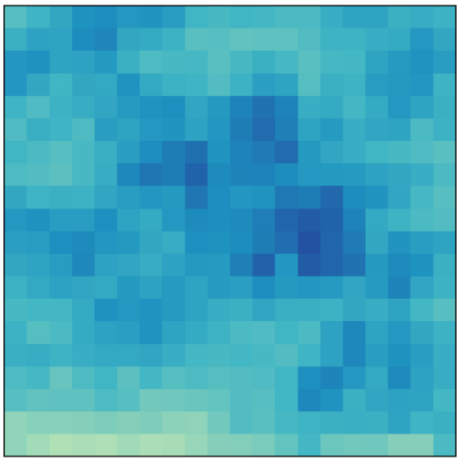
Coarse Data

\-



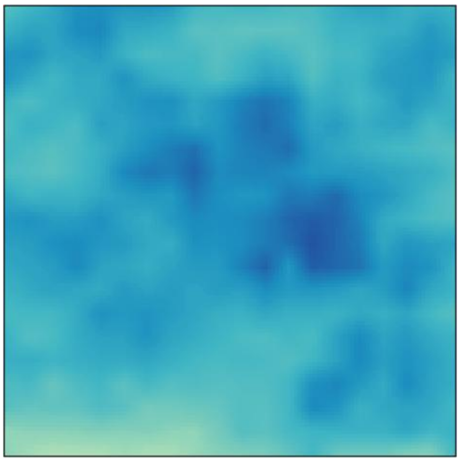
Nearest Neighbour

0



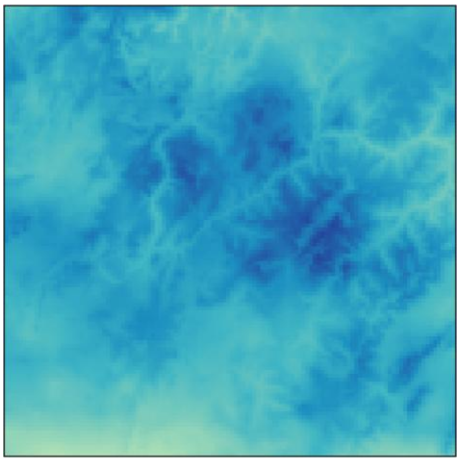
Bilinear Interpolation

1



Factors based on WorldClim data

2



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

| Section [options]<br>Name | Available Options | Description |
|---|---|---|
| use_scheduler | y<br>n | Use built-in splitting algorithm for large areas.<br>**IMPORTANT**<br>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.<br>0: Irrigation module deactivated |
| precipitation_downscaling_method | 0, 1, 2 | 0: Nearest Neighbour interpolation<br>1: Bilinear Interpolation<br>2: By using WorldClim Datasets |
| temperature_downscaling_method | 0, 1, 2, 3 | 0: Nearest Neighbour Interpolation<br>1: Bilinear Interpolation<br>2: By using WorldClim Datasets<br>3: Height Regression |
| output_format | geotiff<br>netcdf4<br>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<br>Recommended: 5-25 |
| downscaling_use_temperature_gradient | y<br>n | Downscaling: Check if slope is within physical limits. Here: Adiabatic Temperature Gradient.<br>Default: y |
| downscaling_dryadiabatic_gradient | Float | Downscaling: Dry Adiabatic Temperatur Gradient [K/m].<br>Default: 0.00976 |
| downscaling_saturation_adiabatic_gradient | Float | Downscaling: Saturation Adiabatic Temperature Gradient [K/m]<br>Default: 0.007 |
| downscaling_temperature_bias_threshold | Float | Downscaling: Temperature BIAS Threshold [K]. Higher Threshold leads to higher performance on the cost of quality<br>Default: 0.0001 |

| | | |
|---|---|---|
| downscaling_precipitation_-bias_threshold | Float | Downscaling: Precipitation BIAS Threshold [mm]. Higher Threshold leads to higher performance on the cost of quality<br>Default: 0.0001 |
| downscaling_precipitation_per_day_threshold | Float | Downscaling: Precipitation below this value will be set to 0. [mm/day]<br>Default: 0.75 |
| output_all_limiting_factors | y<br>n | Write all suitability values of the individual parameter data sets to a geotiff with several bands. |
| remove_interim_results | y<br>n | This function removes all binary intermediate results. Recommended to keep the result files as small as possible.<br>**IMPORTANT**<br>Leads to considerably longer processing times if crops are added later! |
| remove_downscaled_climate | y<br>n | Removes the downscaled climate data<br>**IMPORTANT**<br>Leads to considerably longer processing times if crops are added later! |
| output_soil_data | y<br>n | Output combined soil data |
| multiple_cropping_turnaround_time | Integer | Number of days required between harvest and reseeding with multiple cropping |
| rrpcf_interpolation_method | String | Interpolation method for rrpcf files if they are available at a lower resolution than the target resolution. Available options:<br>1: Linear (default)<br>2: nearest<br>3: cubic |
| consider_crop_rotation | y<br>n | Consideration of crop rotation by combining the data for all simulated crops |
| debug | 1 | Entry not required - in the case of problems, this entry can be added to generate a log file. |

| | | |
|---|---|---|
| simulate_calcification | 0<br>1<br>2<br>3 | Simulate liming of the soil: If the pH value of the soil is too low, lime is applied to raise the pH value.<br>0: No liming<br>1: Raise by max. 0.5<br>2: Raise by max. 1.0<br>3: Raise by max. 1<br>If soil pH exceeds 6.5 due to liming, the application is limited to reach a maximum pH of 6.5. Higher values cannot be achieved through liming in CropSuite. |
| resolution | Integer | Spatial resolution of the output data.<br>0: 0.5° (~ 60 km/px at the equator)<br>1: 0.25° (~ 30 km/px at the equator)<br>2: 0.1° (~ 12 km/px at the equator)<br>3: 5 arcmin (~ 10 km/px at the equator)<br>4: 2.5 arcmin (~ 5 km/px at the equator)<br>5: 30 arcsec (~ 1 km/px at the equator)<br>6: 7.5 arcsec (~ 250 m/px at the equator)<br>If the selected spatial resolution is finer than 30 arcseconds, climate data are interpolated using nearest neighbor. For coarser resolutions, both climate and soil data are resampled using nearest neighbor. |

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]<br>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]<br>Name | Available Options | Description |
|---|---|---|
| consider_variability = | y<br>n | Enable climate variability module |

Section [membershipfunctions]:

| Section [membershipfunctions]<br>Name | Available Options | Description |
|---|---|---|
| plot_for_each_crop = | y<br>n | Plot membership functions for control |

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

| Section [parameters.xxx]<br>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<br>0: Only the first layer is being used<br>1: The top three layers are used (topsoil)<br>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.<br>Default: 2,1.5,1,0.75,0.5,0.25 |
| conversion_factor = | Integer<br>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<br>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:<br>0: linear (Default)<br>1: cubic<br>2: quadratic<br>3: spline<br>4: poly<br>5: slinear |
| rel_member_func = | String | Designation of the membership function in the parameterisation file<br>(→ plant parameterization .inf) |

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

#### 4.7.2. 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.<br>For perennial crops use 365. |
| temp_vals = | °C List of Integers | Plant-specific temperature values<br>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<br>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_temp_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_temp_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 |
| lethal_min_prec_duration | [days] Integer | Specifies the maximum period in days in which no precipitation may fall |
| lethal_min_prec | [mm] Integer | Maximum precipitation that may fall for each day to be counted as a consecutive dry day |
| lethal_max_prec | [mm] Integer | Minimum precipitation that must fall for this day to be counted as a heavy precipitation day |
| lethal_max_prec_duration | [days] Integer | Number of consecutive days with heavy precipitation from which the suitability is set to 0 |
| consider_in_preproc | 0, 1 | Consider additional parameters in preprocessing of rrpcf files. |

In addition, up to 100 further conditions can be defined as Boolean operators in the following format:

```
AddCon:x = [Parameter],[Start Time Range],[End Time Range],[Operator],[Value]
```

With:
x: Consecutive number
[Parameter]: Temperature or Precipitation
[Start Time Range] and [End Time Range]: Day of growing cycle, in between the condition applies.
[Operator]: <=, >=, <, >
[Value]: Value, which must not be exceeded.

Example:

```
AddCon:1 = Precipitation,75,105,<=,600
```

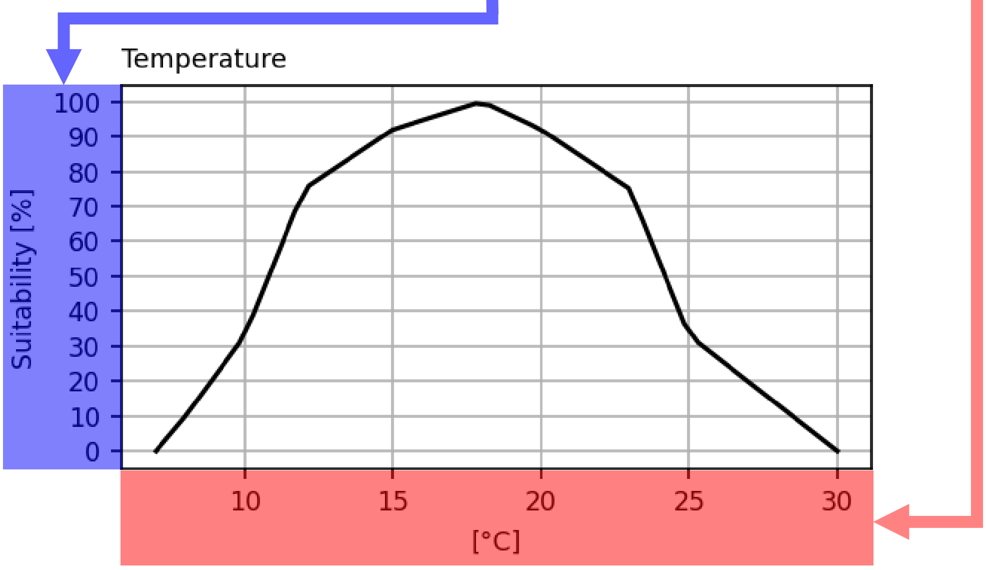
The precipitation must be below 600 between the 75th and 105th day of the growing cycle, otherwise the suitability is set to 0.

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

| | | |
|---|---|---|
| [name]_vals = | Unit of Layer<br>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.

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

| | | |
|---|---|---|
| optimal_sowing_date_first.tif<br>optimal_sowing_date_first.nc | Geotiff<br>NetCDF4<br>0-365 | The day/week of the year to start the first growing cycle when using multiple cropping<br>[day of year] |
| optimal_sowing_date_second.tif<br>optimal_sowing_date_second.nc | Geotiff<br>NetCDF4<br>0-365 | The day/week of the year to start the second growing cycle when using multiple cropping<br>[day of year] |
| optimal_sowing_date_third.tif<br>optimal_sowing_date_third.nc | Geotiff<br>NetCDF4<br>0-365 | The day/week of the year to start the third growing cycle when using multiple cropping<br>[day of year] |
| all_suitability_vals.tif | Geotiff<br>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<br>NetCDF4<br>0-300 | Climate Suitability taking multiple cropping into account.<br>[1/100] |
| soil_suitability.tif | Geotiff<br>NetCDF4<br>0-100 | Suitability only based on static soil datasets. Climate is not considered.<br>[1/100] |

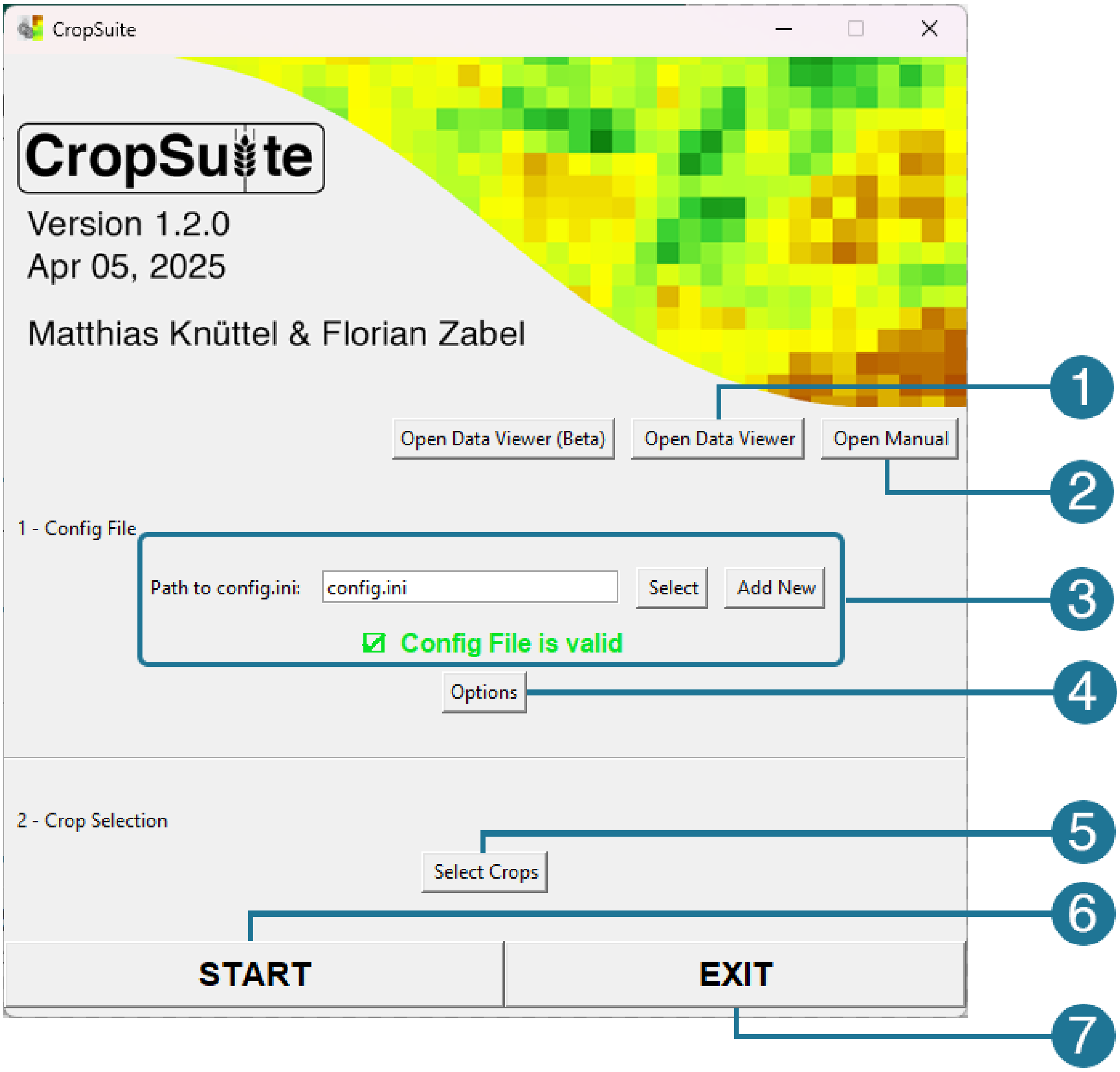
## 5. Using the Graphical User Interface (GUI)

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 figures.

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
```

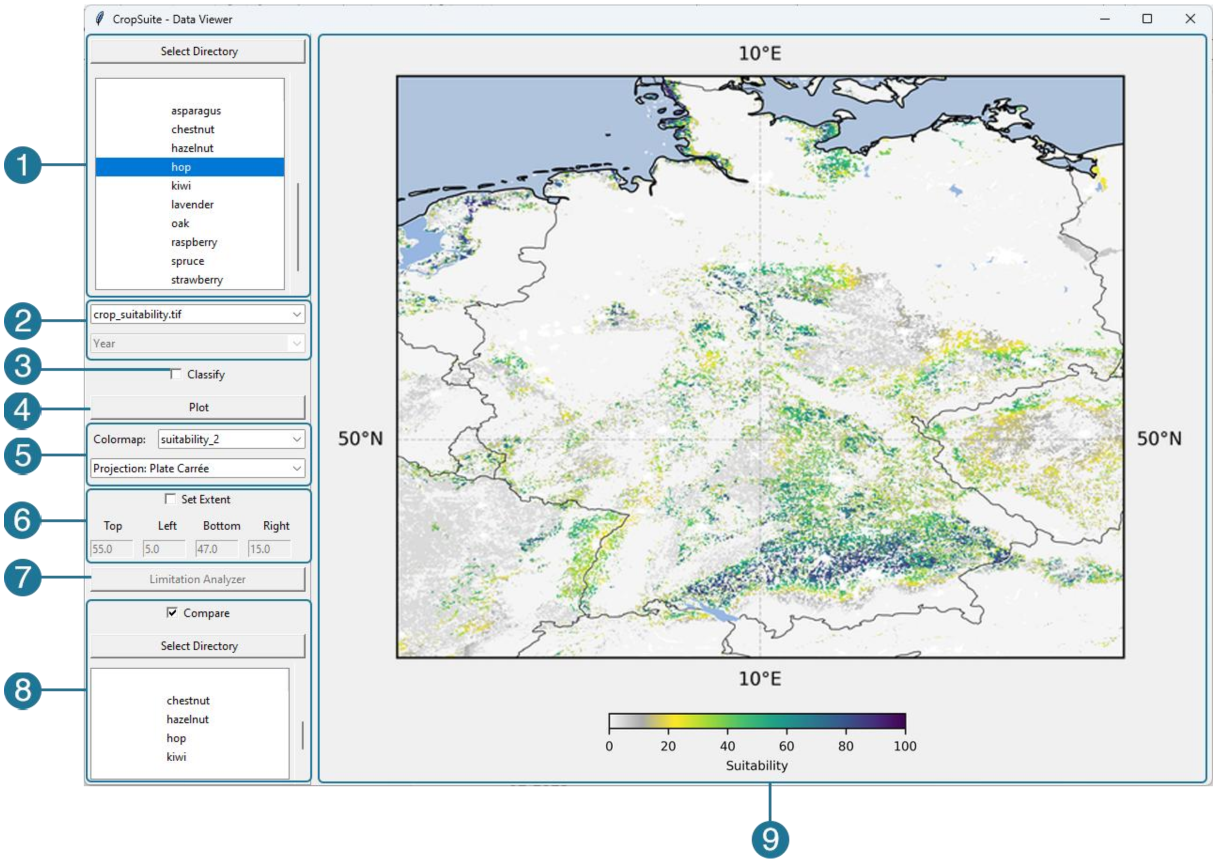
This opens the CropSuite user interface:



1. Open the data viewer (greyed out if no config.ini is specified) → See 5.1
2. Open this user manual
3. Current path to the selected configuration file, with a button to select a different file and an option to create a new configuration file.
4. Open the menu to edit the options specified in config.ini → See 5.2
5. Selecting the crops to be modelled → See 5.4
6. Start the model run
7. Quit CropSuite

## 5.1. 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.
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.

2. Selection of the available result files that can be displayed.
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.

3. Selection of whether the data should be divided into classes or whether the data should be displayed as continuous data. The classes are divided equidistantly.

4. Command to display the data with the selected settings. The map can be found as png in the selected directory.

5. Further display options: Selection of the color palette and the map projection. The map projection cannot be changed for small areas.

6 Possibility of defining the map area.
Specification of the coordinates for cutting the map in decimal degrees.

7 Open the limiting factor analyzer

8 Option to compare the data selected above with other datasets.
Selection of the directory in which the data for comparison is located
Lower treeview for selecting the comparison data. By default, the directory specified as the output directory in config.ini is also used.

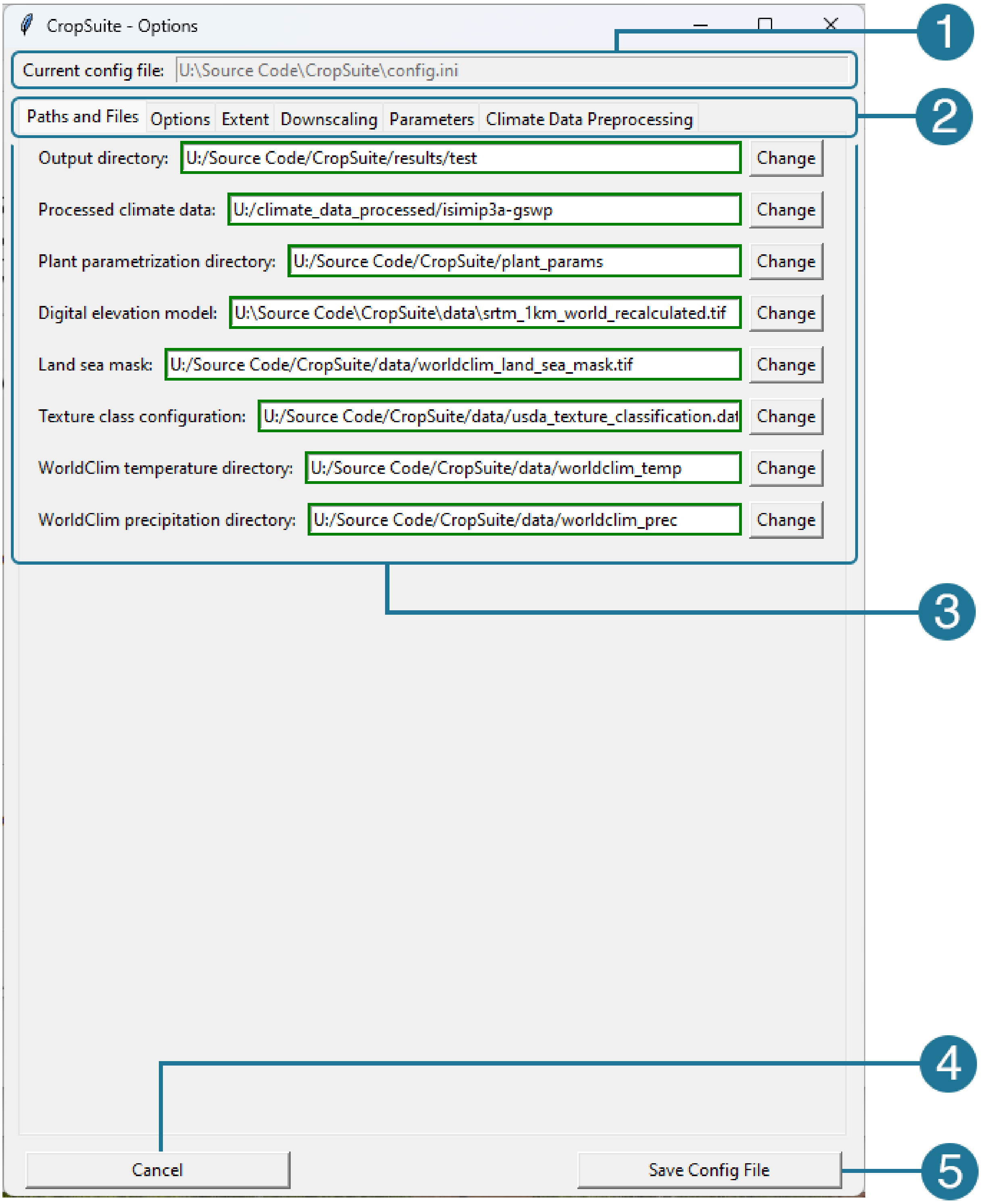
9 Map viewer

### 5.2. 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 4.7.1 regarding all settings.

#### 5.2.1. Page 'Paths and Files'

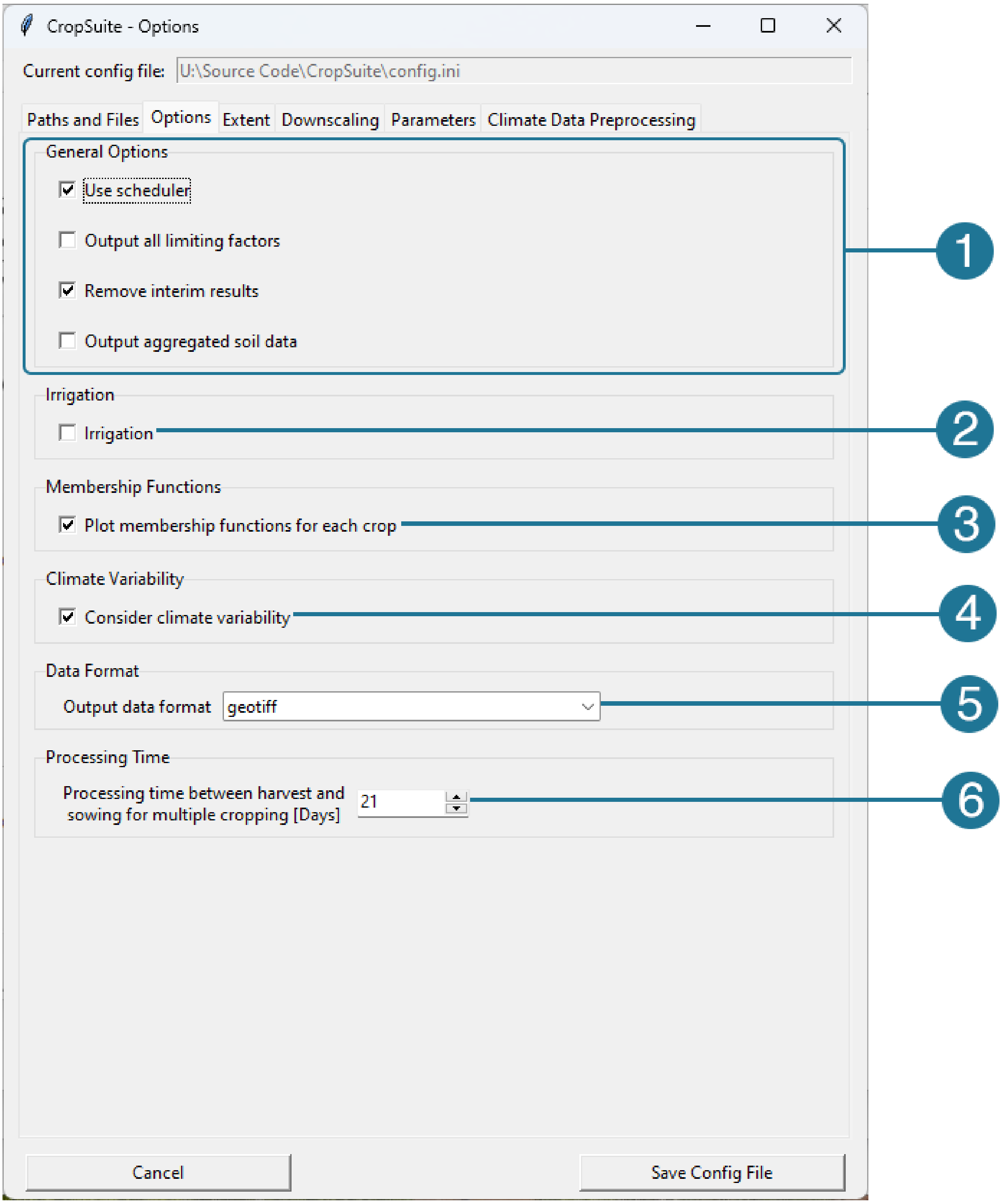


1. Current config file that is being customized
2. Tab list of the different option pages
3. Folder and file paths required for CropSuite

   **Output directory**: Folder for the result files
   **Processed climate data**: Climate data directory in which the Avg_temp.tif and Avg_prec.tif as well as the rrpcf files are located
   **Plant parametrization directory**: Directory in which the plant parameterization files are located
   **Digital elevation model**: Path to the digital elevation model (DEM)
   **Land sea mask**: Path to the land-sea mask
   **Texture class configuration**: Path to the texture class definition file
   **WorldClim temperature directory**: Directory containing the WorldClim precipitation data
   **WorldClim precipitation directory**: Directory containing the WorldClim temperature data
4. Close window without saving
5. Save the options and close the window

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

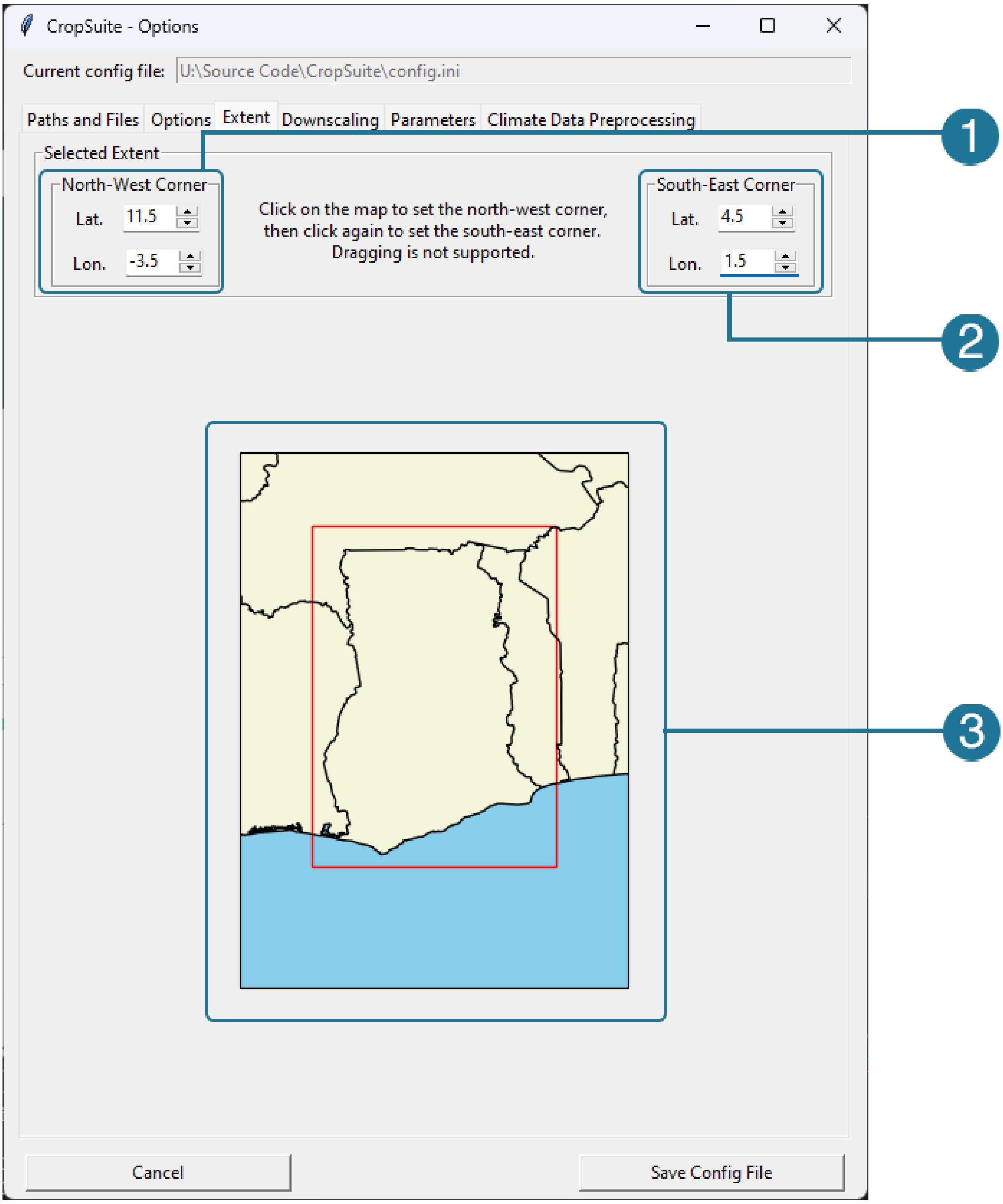
### 5.3. Page 'Options'



General Options

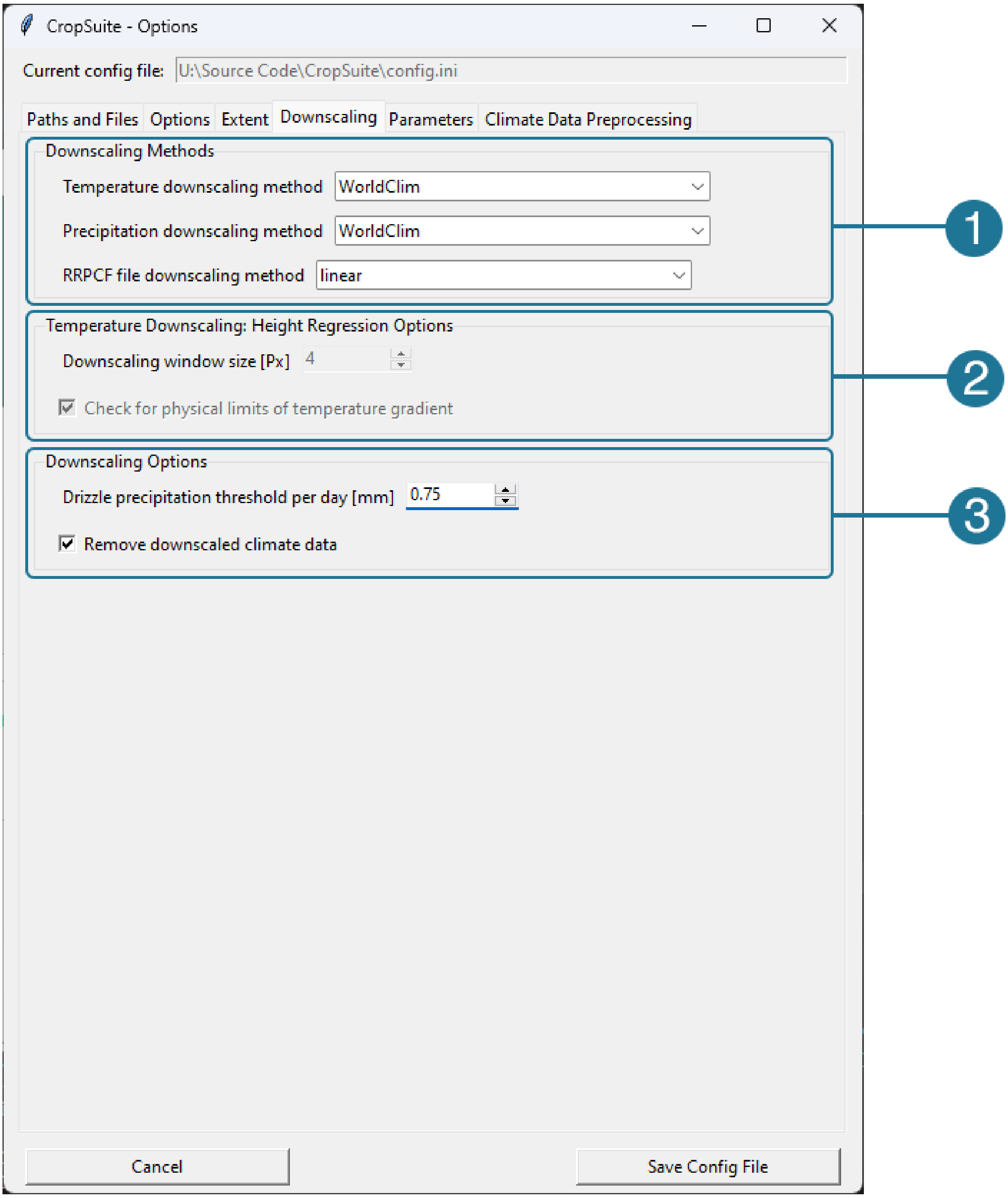
1. **Use scheduler**: Activate automatic splitting of large areas
   **Output all limiting factors**: Write all suitability values of the individual parameter data sets to a geotiff with several bands.
   **Remove interim results**: Remove unnecessary intermediate results after completion
   **Output aggregated soil data**: Output the weighted soil data as geotiff files
2. Consider irrigation (this will affect climate and crop suitability, optimal sowing dates, multiple cropping, suitable sowing days as well as climate variability)
3. Plotting of all parameterisation files
4. Option as to whether climate variability should be considered
5. Output data format
6. Number of days between harvest and reseeding with multiple cropping

### 5.3.1. Page 'Extent'



1. Selection of the coordinates of the top-left/northwest corner
2. Selection of the coordinates of the lower-right/southeast corner
3. Preview of the selected extent and selection option with mouse: first click sets northwest corner, second click sets southeast corner

### 5.3.2. Page 'Downscaling'



*Also refer to chapter 4.7.1*

Selection of downscaling methods

1

**Temperature downscaling method**: Selection of the downscaling method for the temperature data

**Precipitation downscaling method**: Selection of the downscaling method for the temperature data

**RRPCF file downscaling method**: Selection of the interpolation method for the rrpcf data, required for consideration of climate variability

Specific options when using the height regression to downscale the temperature data

2

**Downscaling window size**: Size of the moving window to determine the temperature gradient and offset. The larger the value, the more pixels are included in the calculation.

**Check for physical limits of temperature gradient**: Checks the determined temperature gradient for physical plausibility

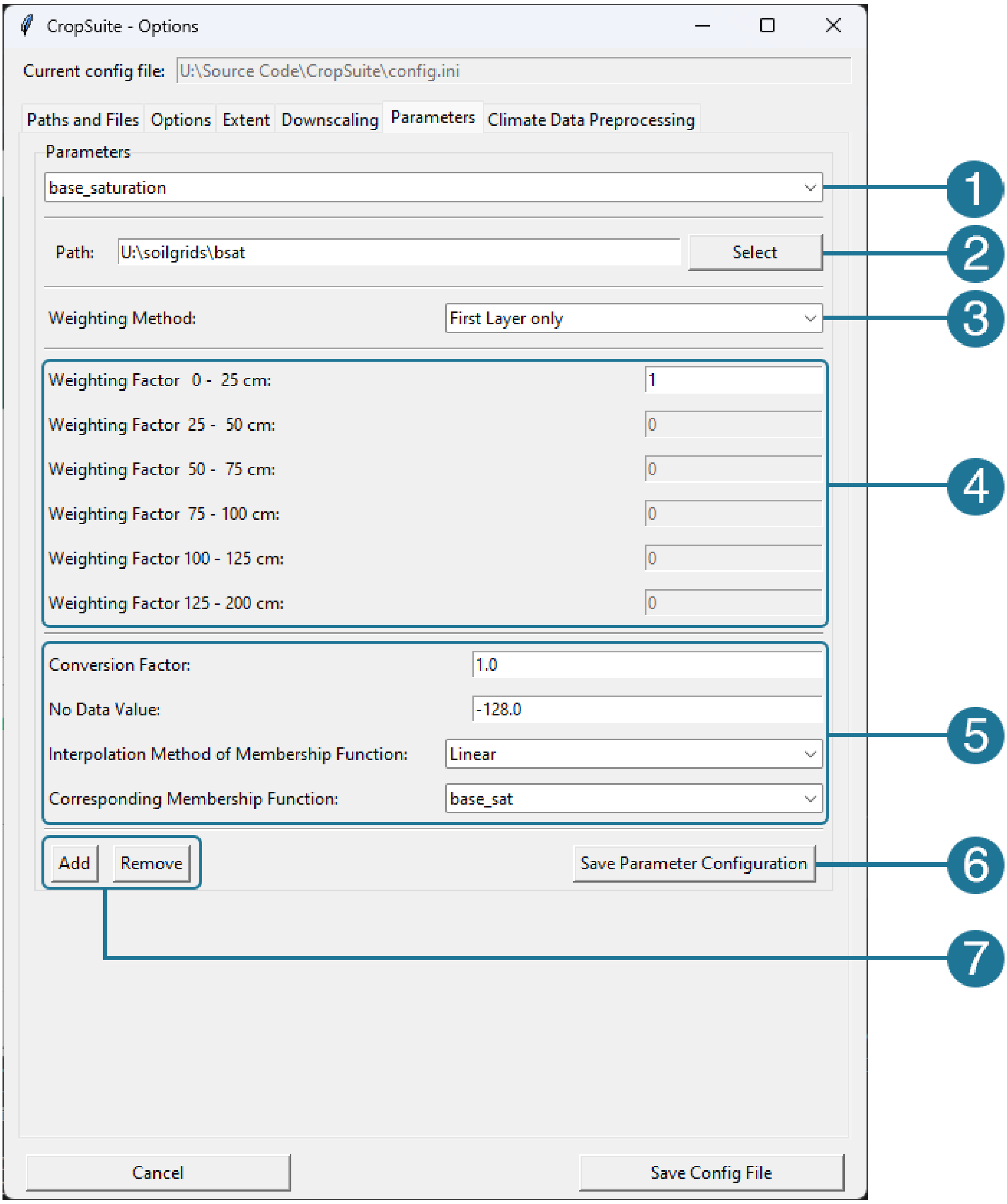
General Downscaling Options

3

**Drizzle precipitation threshold per day**: Specification of the drizzle precipitation value to correct the drizzle effect of climate models. Daily precipitation below the selected value is set to 0

**Remove downscaled climate data**: Removes the downscaled climate data files after completion to save hard disk space

### 5.3.3. Page 'Parameters'



*Also refer to chapter 4.7.1*

1. Select parameter dataset to adjust

2. Current path to dataset directory and button to change directory

3. Weighting method of layer

   **First layer only**: Only the alphabetically first GeoTIFF file in the selected directory is used
   **Top soil/First three layers**: The alphabetically first 3 GeoTIFF files are used and equally weighted
   **Full soil/Six layers**: Alle 6 soil layers are being used and weighted accordingly to the specified weighting factors

4. Weighting factors for different soil depths

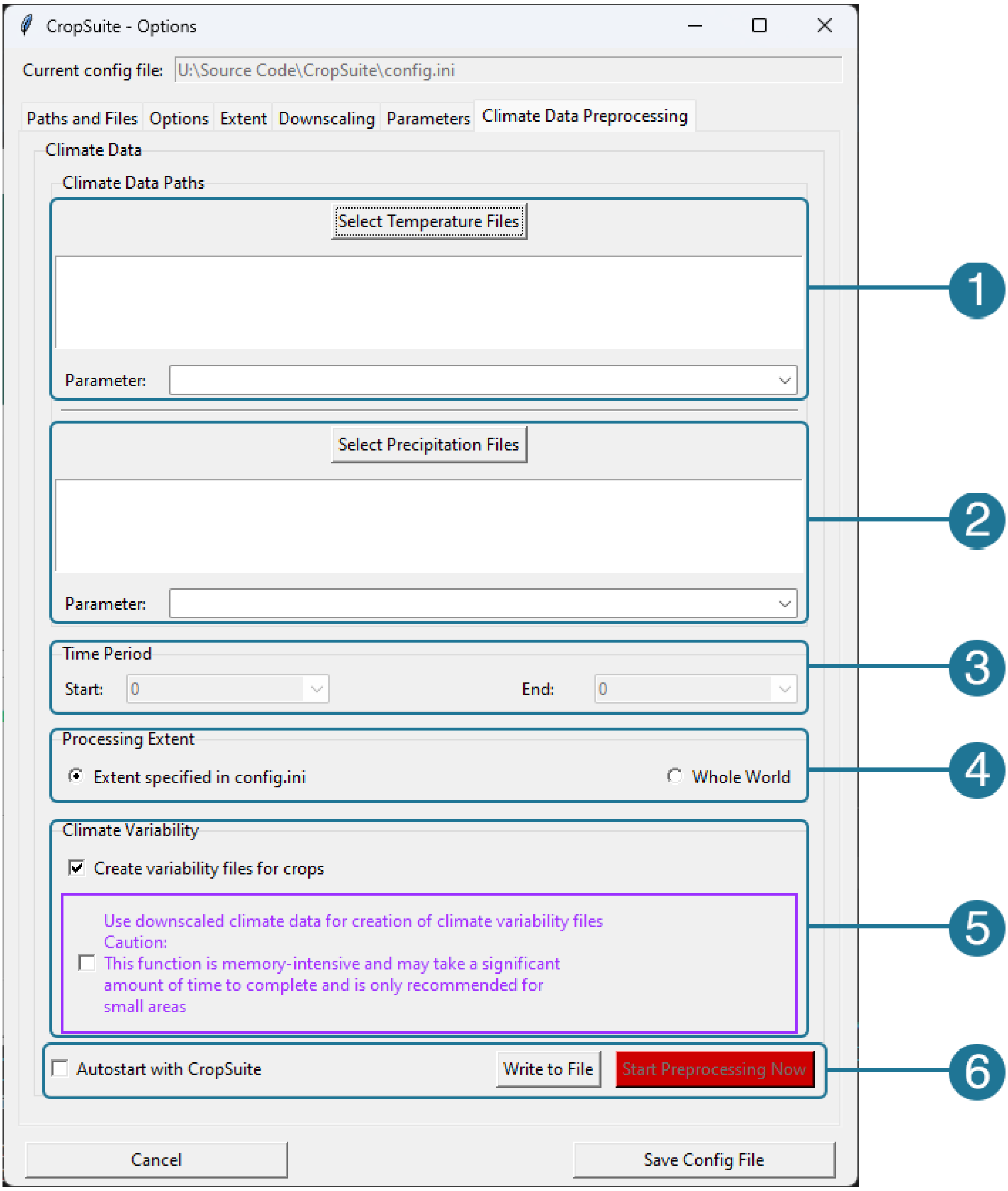
5. Parameter specified options

   **Conversion Factor**: Multiplicative conversion factor
   **No Data Value**: Specify no data value if not in metadata of dataset. By default -9999 is being used. Automatically overwritten by no data value in metadata of dataset
   **Interpolation method of corresponding membership function**
   **Corresponding membership function**: Associated membership function in the crop parameterisations

6. Save settings for current parameter

7. Add new parameter dataset or remove currently selected parameter

### 5.3.4. Page 'Climate Data Preprocessing'



*Also refer to chapter 4.4*

1 Select temperature NetCDF4 files, the selected files are listed below. Select the variable name for temperature if not automatically selected

2 Select precipitation NetCDF4 files, the selected files are listed below. Select the variable name for precipitation if not automatically selected

3 Select the period for which the preprocessing should be executed

4 Select the processing extent. 'Whole World' is not available when processing very large areas like the whole world

5 Climate variability options

**Create variability files for crops**: Calculation of climate variability files for each crop

**Use downscaled climate data for creation of climate variability files**: 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.

6 Start options

**Autostart with CropSuite**: A config file for downscaling is written, when CropSuite is started it is automatically read-in and preprocessing is carried out.

**IMPORTANT**

If the option to use downscaled climate data is selected, this can lead to a considerable increase in processing time.

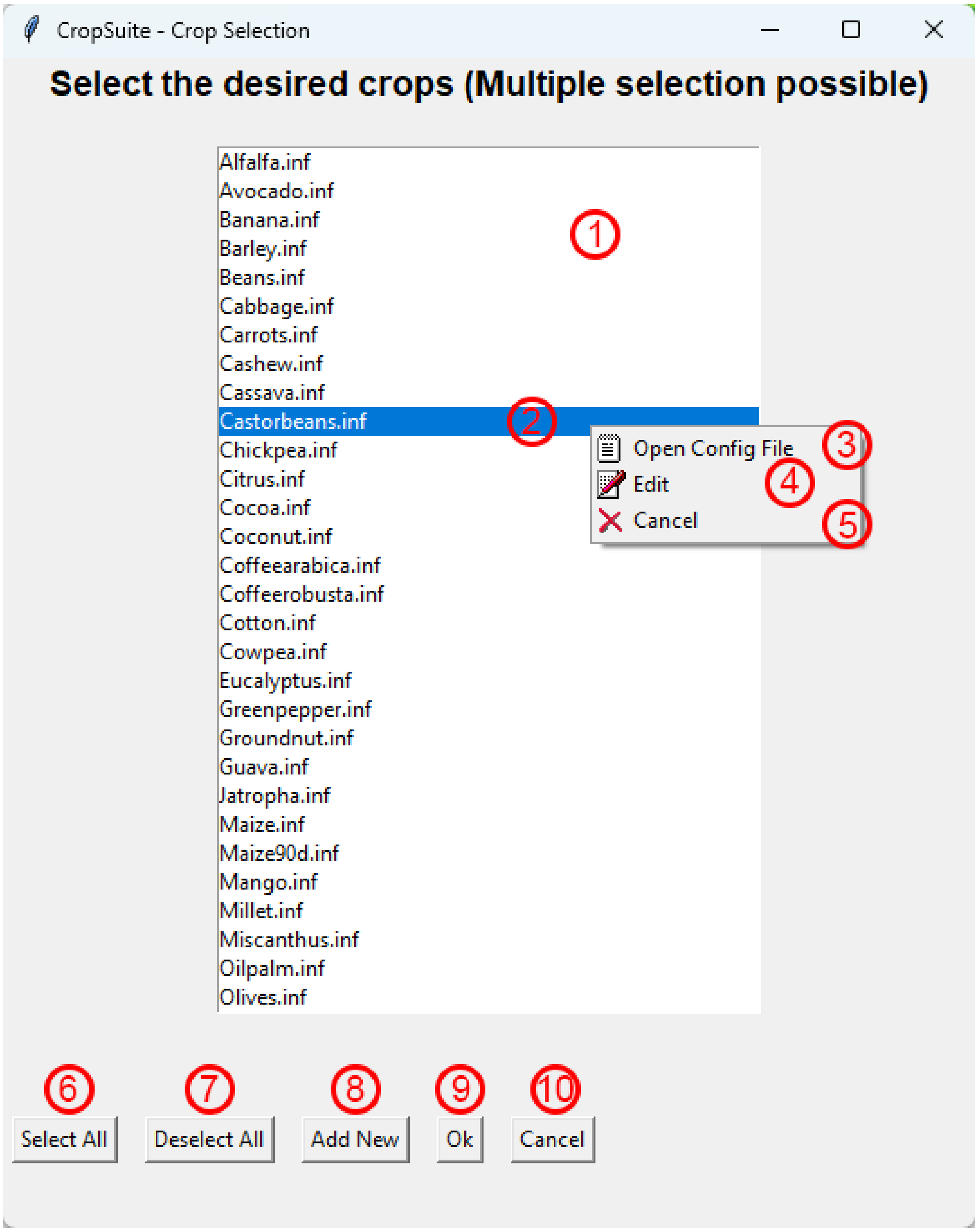
**IMPORTANT**

If a preproc.ini file exists in the CropSuite directory, the preprocessing is carried out accordingly.

**Write to file**: Write the specified settings in preproc.ini

**Start preprocessing now**: Instead of starting the preprocessing with CropSuite, the preprocessing can also be started manually immediately

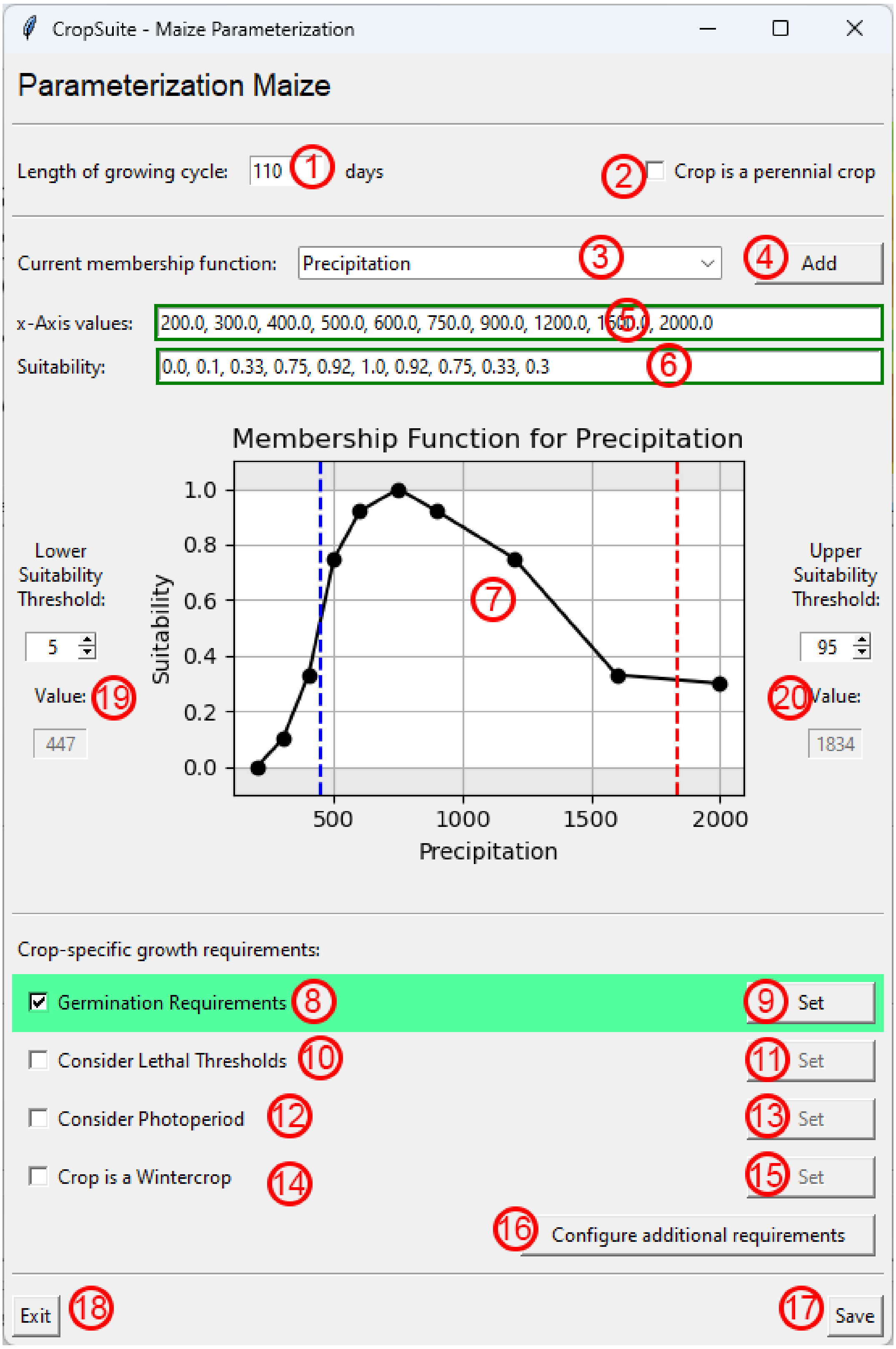
## 5.4. Window for selecting the crops to be modelled



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 5.5
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

### 5.5. Window for editing crop specific parameters

Please refer to chapter 4.7.2



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

## 6. 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).

## 7. 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

# 8. Appendix

## 8.1. 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
rrpcf_interpolation_method = linear
consider_crop_rotation = y
simulate_calcification = 0
resolution = 5

[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
```

```
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
```

## 8.2. 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_temp_duration =      3
lethal_max_temp =       40
lethal_max_temp_duration =      5
prec_req_after_sow =    50
prec_req_days =         30
lethal_min_prec_duration =      21
lethal_min_prec =  1
lethal_max_prec_duration = 3
lethal_max_prec =  100
wintercrop =            y
vernalization_effective_days = 50
vernalization_tmax =    7
vernalization_tmin =    1
frost_resistance_days = 3
frost_resistance =      -20
days_to_vernalization = 30
```

# 9. Changelog

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

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

| | | |
|---|---|---|
| | | • Corrected determination of multiple cropping optimal sowing dates<br>• Parallel processing of datasets with and without crop failure frequency<br>• Enhanced determination of limiting factor in unsuitable areas<br>• Improved Nan-handling of soil datasets<br>• Output of soil suitability<br>• Performance optimizations |
| 04.08.2024 | 0.9 | • SLURM support added<br>• Performance optimizations for MacOS systems<br>• Integrated postprocessing script to adjust shape of multiple cropping dates<br>• Enhanced file writing to decrease output file size<br>• Added high performance computing support<br>• Enhanced output of all limiting factors<br>• Improved clearness of the terminal output<br>• Added correct handling of NoData values added when all limiting values are output<br>• Downscaling performance optimizations<br>• Added different downscaling methods for temperature and precipitation<br>• Added downscaling of temperature and precipitation based on WorldClim factors<br>• Restructured Downscaling and Climate Suitability Module<br>• Added support of cloud optimized GeoTiffs (COG) as output format<br>• Improved merging of netcdf4 and cog files<br>• Support for netcdf4 climate input data added<br>• Specification of wintercrops added<br>• Vernalization enhanced<br>• Frost resistance added<br>• Separate calculation of optimum sowing date with and without vernalisation added<br>• Improved output of optimum sowing dates<br>• Improved performance of vernalization calculation<br>• User interface added<br>• Lethal temperature thresholds added<br>• Crop specific precipitation requirements after sowing added<br>• Consideration of photoperiod added<br>• Data Viewer added<br>• Crop specific number of consecutive dry days added to consider droughts<br>• Output of weighted soil datasets added<br>• 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<br>• Vernalisation: No weighting of the suitabilites, but common weighted temperature and precipitation mean value for the determination of suitability<br>• Improved determination of sowing dates for winter crops<br>• Performance improvements when writing downscaled climate data<br>• 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<br>• Free selection of the map projection in the viewer<br>• Improved map display in the viewer<br>• Possibility to set the extent in the viewer added<br>• Universal preprocessing of climate data added to CropSuite<br>• Performance improvements of daily downscaling for creation of climate variability files for each crop<br>• Winter crops: improved determination of suitability when taking climate variability into account<br>• Dynamic calculation of the slope inclination based on the terrain model added<br>• New improved Crop Parameterization window: Object-oriented implementation<br>• Limit values selectable in Crop Parameterization for the consideration of climate variability<br>• Freely selectable crop-specific temperature and precipitation limits added<br>• More robust import commands for different system structures<br>• Improved multiprocessing for downscaling during preprocessing<br>• Adaptation of the interpolation method to the RegularGridInterpolator<br>• Improved NA handling during downscaling in preprocessing<br>• Improved parameter window: object-oriented implementation<br>• Fixed crash when incorrect plant_param path was specified<br>• Added used Python package version numbers to manual<br>• Major performance and program structure enhacements<br>• Added multiprocessing to climate data downscaling<br>• Removed usage of .npy interim files<br>• Data Viewer: Colormap freely selectable<br>• Corrected output path for results without consideration of climate variability<br>• Corrected output of aggregated soil datasets<br>• Added option to remove the downscaled climate files<br>• Adjusted worldclim downscaling method<br>• Changed output directory for downscaled climate files<br>• Renamed output files for sowing if crop is a wintercrop |
| 12.12.2024 | 1.0 | • Added Limitation Analyzer<br>• Enhanced downscaling performance<br>• Adjusted preprocessing downscaling worldclim method<br>• Added multiprocessing to downscaling using the nearest neighbour, bilinear and height regression method<br>• Added multiprocessing to downscaling of rrpcf data<br>• Changed downscaling of rrpcf from stripwise to daywise<br>• Corrected bilinear and nearest neighbour downscaling method<br>• Added support for legacy created rrpcf files |
| 02.02.2025 | 1.0.1 | • Checking the plausibility of the extent during input<br>• Limit value for days without precipitation added<br>• Limit value and duration added to take heavy precipitation into account<br>• Reading of Plant Param Dir and improved handling if it is not found |

| | | |
|---|---|---|
| | | • Automatic selection of germination requirements only for non-perennial crops.<br>• Automatic selection of the paralellization method depending on the area size<br>• Improved calculation of lethal parameters and consecutive dry days for improved performance<br>• Change of interpolation method for downscaling to reduce memory usage for large areas<br>• Improved preprocessing for better performance when using large areas<br>• Improved division into strips when using the scheduler<br>• Improved data merging for better performance and lower memory utilization<br>• Automatic selection of the data type of the output data added<br>• Improved NA handling of soil data added<br>• Fixed rounding error of coordinates when selectively reading input data<br>• Fixed renewed downscaling of climate data if the results are already available<br>• Improved verification of the extents of results<br>• Added selection of interpolation method for rrpcf files |
| 16.02.2025 | 1.1.0 | • New user interface for the main window and options<br>• Preprocessing and parameter settings integrated in options<br>• Division into strips improved<br>• Consideration of limiting_factor.inf when merging several strips |
| 10.04.2025 | 1.2.0 | • Further input data structures added<br>• Automatic detection of daily highs and lows and combination to form the daily average in preprocessing<br>• Consideration of the additional conditions when creating the rrpcf files in preprocessing added<br>• Performance improvement of preprocessing<br>• Output resolution freely selectable, no longer linked to elevation model<br>• Liming of soils added<br>• Calculation of the required amount of lime added<br>• Calculation of crop rotations added<br>• Automatic adjustment of the CRS of the input data for all ground data sets, DEM and land-sea mask added<br>• Various bugfixes |
| 12.04.2025 | 1.2.1 | • Improved nan handling of soil data<br>• Deprecation of interp1d and replacement by newer functions such as Barycentric Interpolator or Akima1Dinterpolator<br>• Debug option added<br>• Automatic check of the Python environment at startup<br>• Error when adding new crops fixed<br>• Addition of crops made easier<br>• Error fixed if the plant_param-directory is not the default directory<br>• Improved window sizes on Unix systems<br>• Improved error handling when reading .ini files |

| | | |
|---|---|---|
| | | • Improved handling of unknown entries in plant parametrizations and config files |
| 16.04.2025 | 1.3.0 | • Added setup scripts for the required Python environment for Windows 11 and MacOS<br>• Fixed a bug where the length of the growing cycle was not written correctly in the parameterization inf<br>• Improved memory management for winter crops<br>• Simplified setup script for the Python environment for Windows added<br>• Added automatic installation and start script for CropSuite for MacOS<br>• Fixed a problem where the membership function is displayed twice in the crop parameterization window on MacOS |
| 17.4.2025 | 1.3.1 | • Fixed a problem in the config window where the parameter settings were not written correctly to config.ini.<br>• Fixed a crash when no valid plant-param directory was selected and a membership function is assigned to a parameter<br>• Fixed a crash where the consideration of lethal parameters is not handled correctly |

Intentionally left white