CLUMondo-Based Land Use Modelling Workflow Manual

Overview

This Python-based land use modelling script simulates land cover changes using principles derived from the CLUMondo C++ model. It incorporates regional suitability, neighbourhood influence, conversion resistance and allowance matrices, and dynamic land use demand elasticities.

Requirements

Python Version: Python 3.8+

Required Python Packages:

- numpy
- pandas
- rasterio
- gdal
- openpyxl
- numba

Input Data & Parameters

You can either provide individual arguments via CLI or use a config file (.txt) containing `--arg=value` format lines.

Required arguments:

Argument	Type	Description	Additional Notes
--land_array	str	Path to land cover raster	Raster representing initial land cover state.
--suit_array	str	Path to location suitability stack	Raster stack with suitability scores per class.
--region_array	str	Path to region raster	Mask where value 1 indicates restricted cells.
--neigh_weights	str	Neighbourhood weights, comma-separated.	One weight per class (e.g., 0.2,0.3,0.5).
--start_year	int	Start year.	Start of simulation period.
--end_year	int	End year.	End of simulation period.

Argument	Type	Description	Additional Notes
--demand	str	Path to demand file.	Excel file with yearly demand for each service.
--dem_weights	str	Demand weights, comma-separated.	Priority of meeting each demand (e.g., 1.0,0.8).
--lus_conv	str	Path to LUS conversion file.	Excel file with conversion factors per class.
--lus_matrix_path	str	Path to LUS matrix file.	Can be fixed matrix or directory with yearly ones.
--conv_res	str	Conversion resistance, comma-separated.	Resistance to change per land cover class.
--allow	str	Path to allow file.	Matrix indicating allowed class-to-class transitions.
--out_dir	str	Output directory.	Folder to save model output and logs.
--crs	str	Output CRS.	Coordinate system of output rasters (e.g., EPSG:4326).

Optional arguments:

Argument	Type	Description	Default	Additional Notes
--max_diff_allow	float	Maximum allowed difference.	3.0	Class-wise tolerance between supply and demand.
--totdiff_allow	float	Total allowed difference.	1.0	Total % difference allowed between current and target demand.
--max_iter	int	Maximum iterations.	3000	Per-year loop cap for convergence.
--dtype	str	Output data type.	'int16'	Type of data saved in GeoTIFF (e.g., 'float').
--no_data_out	int	No data value for output.	-9999	Used to fill masked cells in output rasters.
--change_years	str	Years to change reg_suit_array as a comma-separated string.	''	Only needed if suitability maps change over time.
--change_paths	str	Paths to reg_suit_array files for change years.	''	Must match order of --change_years.
--age_array	str	Path to age array.	None	Optional input to track land use persistence.
--zonal_array	str	Path to zonal array	None	Optional input to include zonal allowance in matrix
--preference_array	str	Path to preference array	None	Optional input to include preference values for suitability
--preference_weights	str	Preference weights, comma separated	None	Optional input to weigh preference array, necessary if preference_array is given
--width_neigh	int	Window width for neighborhood analysis.	1	A value of 1 means 3x3 window (Moore neighborhood).

Argument	Type	Description	Default	Additional Notes
--demand_max	float	Maximum elasticity value for demand.	3.0	Caps how elastic demand can get during updates.
--demand_setback	float	Elasticity setback value.	0.5	Used to reset elasticity if it exceeds limit.
--no_data_value	int	No data value.	-9999	Used internally to mask invalid areas.

Running the Model

Option 1: Using Config File

python clumondo_model.py --config=config.txt

Example for config.txt file:

```
--land_array=./data/land_2020.tif
--suit_array=./data/suitability_stack.tif
--region_array=./data/region_mask.tif
--neigh_weights=0.2,0.3,0.5
--start_year=2020
--end_year=2030
--demand=./data/demand_projection.xlsx
--dem_weights=1.0,0.8,0.6
--lus_conv=./data/lus_conversion.xlsx
--lus_matrix_path=./data/yield_matrices/
--conv_res=0.5,0.2,0.3
--allow=./data/allow_matrix.xlsx
--out_dir=./output/
--crs=EPSG:4326
--max_diff_allow=3.0
--totdiff_allow=1.0
--max_iter=3000
--dtype=int16
--no_data_out=-9999
--change_years=2025,2028
--change_paths=./data/suit_2025.tif,./data/suit_2028.tif
--age_array=./data/age_2020.tif
--preference_array = .data/preference_stack.tif
--preference_weights=0,0,0.5
--width_neigh=1
--demand_max=3.0
--demand_setback=0.5
--no_data_value=-9999
```

Option 2: Command Line Arguments

python clumondo_model.py [your arguments here]

Example for input in command line

```
python clumondo_model.py \
 --land_array ./data/land_2020.tif \
 --suit_array ./data/suitability_stack.tif \
 --region_array ./data/region_mask.tif \
 --neigh_weights 0.2,0.3,0.5 \
 --start_year 2020 \
 --end_year 2030 \
 --demand ./data/demand_projection.xlsx \
 --dem_weights 1.0,0.8,0.6 \
 --lus_conv ./data/lus_conversion.xlsx \
 --lus_matrix_path ./data/yield_matrices/ \
 --conv_res 1.5,2.0,1.2 \
 --allow ./data/allow_matrix.xlsx \
 --out_dir ./output/ \
 --crs EPSG:4326 \
 --change_years 2025,2028 \
 --change_paths ./data/suit_2025.tif,./data/suit_2028.tif \
 --age_array ./data/age_2020.tif \
 --width_neigh 1 \
 --demand_max 3.0 \
 --demand_setback 0.5
```

Output Files

Each simulation year is saved in a timestamped subfolder under the --out_dir location.

- covYYYY.tif: Land cover raster.
- ageYYYY.tif: (if age tracking enabled)
- logfile.txt: Iteration logs and convergence stats.

Dynamic Suitability Support

If the suitability changes over time, provide --change_years and --change_paths.

Important specification of input data

Disclaimer: For specific descriptions of the functionalities of the single input dataset, please also refer to the original CLUMondo manual.

Land cover raster (land_array):

Class ID's in the land cover data have to start from 0 and rise in ascending order. There should be no gaps between class ID's. An example is below:

Class label	Class ID
Forest	0
Pasture	1
Cropland	2
Urban	3

Users should adjust their data to this format, otherwise correct execution of the model cannot be guaranteed.

Location suitability stack (suit_array):

The location suitability stack should have the same amount of layers in the stack as classes in the land cover raster. Also, it should have the same number of rows, columns and spatial extent as the land cover raster. Values in the location suitability stack should range from 0-1. The same applies if you have dynamic location suitability stacks (for change_years),

Region raster (region_array):

The region raster should have the same number of rows, columns and spatial extent as the land cover raster. A value of 1 should indicate a restricted pixel.

Age raster (age_array):

The age raster should have the same number of rows, columns and spatial extent as the land cover raster.

Zonal stack (zonal_array):

The zonal stack should have the same number of layers in the stack as classes in the land cover raster. Also, it should have the same number of rows, columns and spatial extent as the land cover raster. Each layer in the stack represents zones for restricted allowance for each class (see description of allowance matrix). This can be helpful to simulate conversion from natural to plantations in concession areas, where this conversion is allowed only in specific zones while prohibited outside. Zones should be flagged with the value 1, while 0 indicates no zonal restriction.

Preference stack (preference_array):

The preference stack should have the same number of layers in the stack as classes in the land cover raster. Also, it should have the same number of rows, columns and spatial extent as the land cover raster. Each layer in the stack represents preference zones for location suitability for each land cover class. Similar to the zonal stack, values should be either 1 or 0. The idea is that suitability values will be enhanced at these locations. An example would be a program which supports maintenance of mosaic agriculture in slopy areas.

Preference weights (preference_weights):

The preference weights serve to multiply the values from the preference stack. Weight values should be between 0 and 1 and should be provided for all classes. If there are no preference areas for a class, users can simply assign 0 as the weight.

Demand:

The demand file should be submitted as an excel file. Column names (names of services) should be included. Each row specifies the demand for a year in the timeseries to be modelled. The unit of the demand is not specified. Below is an example:

Milk	Wheat	Timber
2135	8058	6771
2132	8356	7316
2130	8653	7861
2127	8951	8406
2124	9249	8951

Land system services (lus_matrix):

This file should be submitted as an excel or be the path to a folder of excel files. In case it is in excel file, it should contain an extra row and column for labels. Please make sure that all land cover classes and all demands are listed in this table. Below is an example:

	Milk	Wheat	Timber
Forest	0	0	1.7
Pasture	2.3	0	0
Cropland	0	4.1	0
Urban	0	0	0

In case you provide a path to a folder with excel files (for dynamic land system services), the files should be named “yield_data_{year}.xlsx”. The function will for each year in the iteration then pick the corresponding excel file.

Allowance matrix (allow):

This file should be submitted as an excel file. It should contain an extra row and column for labels. Make sure all possible combinations of land cover class transitions are depicted. If a transition is allowed, the cell should be 1, otherwise remain 0. Rows indicate possible transitions from one land cover to another. Below is an example:

	Forest	Pasture	Cropland	Urban
Forest	1	1	1	0
Pasture	0	1	1	1
Cropland	0	0	1	1
Urban	0	0	0	1

For example, forest can remain forest or transition into pasture or cropland, but not into urban area. Conversely, urban can only remain urban and not transition into another class.

Furthermore, there are some special indicators:

2 = The conversion is allowed, yet only in specified areas. These areas are indicated in the zonal raster stack

1xx = The conversion is allowed after xx number of years. For example, 120 would mean that A can convert to B if it has been A for at least 20 years.

-1xx = After xx number of years, the class has to switch from its current class to another. For example, -115 would mean that after 15 years, class A must switch to any other class which is allowed. This sort of entry can only be put at the diagonal of the matrix.

10xx = After xx number of years, there is an autonomous change from class A to class B.

Please also refer to the original CLUMondo manual for more details.

Conversion matrix (lus_conv):

This file should be submitted as an excel file. It should contain an extra row and column for labels. This file shows the order by which land class demands should be drawn (as priorities). This file is particularly interesting for multifunctional land systems. Below is an example (no multifunctionality, therefore just 1 and 0):

	Milk	Wheat	Timber
Forest	0	0	1
Pasture	1	0	0
Cropland	0	1	0
Urban	0	0	0

Troubleshooting

- No output? Check logs for convergence. Adjust max_iter or diff thresholds.
- Mismatch in arrays? Ensure all rasters align in size, CRS, and extent.
- NoData issues? Confirm the --no_data_value matches your raster's actual values.