Package: cdlsim (via r-universe)

May 21, 2026

Title Simulate Cropland Data Layer Products from the United States Department of Agriculture

Version 0.1.0

Description Provides simulation tools for categorical raster data (e.g., 'SpatRaster' from the terra package). Designed to simulate edge pixel values from a SpatRaster object on reclassified data derived from the United States Department of Agriculture's Cropland Data Layer (CDL) products.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports dplyr, terra, landscapemetrics, readxl, utils, Rcpp

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

LinkingTo Rcpp

VignetteBuilder knitr

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libproj-dev libsqlite3-dev

Repository https://burgerhaley97.r-universe.dev

Date/Publication 2025-07-23 20:54:33 UTC

RemoteUrl https://github.com/burgerhaley97/cdlsim

RemoteRef HEAD

RemoteSha accfd5f053cfb2a520b4da088c70ba49fde50ea6

Contents

`collapse_and_combined` *Function to resize patches in a SpatRaster using the terra and landscapemetrics packages*

Description

Function to resize patches in a SpatRaster using the terra and landscapemetrics packages

Usage

```
collapse_and_combined(patched_raster, og_raster)
```

Arguments

`patched_raster`	The SpatRaster object representing the categorical data.
`og_raster`	The values of the classes of interest, default is "all".

Value

A SpatRaster where each patch is defined as a unique value and each class has its own layer in the spatraster.

`download_cdl_mat_files`
Function to Read in and upzip confusion matrix files from the web

---

Description

Function to Read in and upzip confusion matrix files from the web

Usage

```
download_cdl_mat_files(years, temp_dir = "extracted_files")
```

Arguments

years	The years of data that you want to be downloaded.
temp_dir	The file name you want the extracted files to be stored in.

Value

The raw excel books containing the confusion matrix data for every US state's USDA Crop Land Data Layer.

---

find_patches *Function to resize patches in a SpatRaster using the terra and landscapemetrics packages*

Description

Function to resize patches in a SpatRaster using the terra and landscapemetrics packages

Usage

```
find_patches(raster)
```

Arguments

raster	The SpatRaster object representing the categorical data.

Value

A SpatRaster where each patch is defined as a unique value and each class has its own layer in the spatraster.

---

generate_transition_vectors *Function to generate transition vectors for negative cell values*

Description

Function to generate transition vectors for negative cell values

Usage

```
generate_transition_vectors(r, transition_matrix, iterations = 10)
```

Arguments

r	The SpatRaster object from get_patches with one layer for each class.
transition_matrix	The transition matrix you want to use.
iterations	The number of iterations desired.

Value

A matrix of vector for each patch in each layer.

---

get_mat_data *Function to retrieve confusion matrix data for multiple states of interest*

Description

Function to retrieve confusion matrix data for multiple states of interest

Usage

```
get_mat_data(
  state_abbreviation,
  file_path = "inst/extdata/extracted_files",
  verbose = FALSE
)
```

Arguments

state_abbreviation	A vector of two-letter abbreviations for US states.
file_path	The path to the directory where files are stored (default is "inst/extdata/extracted_files").
verbose	The stops the messages from printing to the console.

Value

A named list where each element is a list of data frames representing confusion matrices for each state.

Examples

```
# example code
# Get data from UT in 2008
ut_mat_data <- get_mat_data(c("UT"))
```

get_mat_data_dep *Function to retrieve confusion matrix data for multiple states of interest*

Description

Function to retrieve confusion matrix data for multiple states of interest

Usage

```
get_mat_data_dep(
  state_abbreviation,
  file_path = "inst/extdata/extracted_files",
  verbose = FALSE
)
```

Arguments

state_abbreviation	A vector of two-letter abbreviations for US states.
file_path	The path to the directory where files are stored (default is "inst/extdata/extracted_files").
verbose	The stops the messages from printing to the console.

Value

A named list where each element is a list of data frames representing confusion matrices for each state.

get_trans_mat *Function to format the confusion matrices as transition matrices*

Description

Function to format the confusion matrices as transition matrices

Usage

```
get_trans_mat(df_list, categories)
```

Arguments

df_list	A single data frame or a list of lists of data frames extracted using get_mat_data(). If a list, each sublist should contain two or more data frames to be summed.
categories	A list of categories defining the numbers between 1 and 256.

Value

A list of data frames where row names represent pixels that will transition and column names represent the class they will transition to.

Examples

```
# make the data frame
bl_mat_data <- get_mat_data(c("UT"))

# List of categories with their corresponding vectors
# Define the values that represent our classes of interest
non_ag <- c(61:65, 81:83, 87:88, 92, 111:112, 121:124, 131, 141:143, 152, 176, 181, 190, 195)
alfalfa <- c(36:37)
major_ag <- c(1, 2, 5, 12, 13, 22:24, 26, 225:226, 228, 234, 236, 238:241, 254)
all_numbers <- 1:256
ag <- setdiff(all_numbers, c(non_ag, alfalfa, major_ag))

cat_5 <- list(non_ag = non_ag, ag = ag, alfalfa = alfalfa, major_ag = major_ag)

# get the confusion matrix for just 2008
trans_mat_5 <- get_trans_mat(bl_mat_data, cat_5)
```

---

get_trans_mat_dep *Function to format the confusion matrices as transition matrices*

Description

Function to format the confusion matrices as transition matrices

Usage

```
get_trans_mat_dep(df_list, categories)
```

Arguments

df_list	A list of lists of data frames extracted using get_mat_data(). Each sublist should contain two or more data frames to be summed.
categories	A list of categories defining the numbers between 1 and 256.

Value

A list of data frames where row names represent pixels that will transition and column names represent the class they will transition to.

`modify_matrix` *Function to modify transition matrix based on user input.*

Description

Function to modify transition matrix based on user input.

Usage

```
modify_matrix(mat, indices = 0)
```

Arguments

`mat`	The transition matrix the you want to update.
`indices`	The vector of classes you want to remain unsimulated.

Value

A transition matrix modified to not transition some classes.

`simulate_raster_patch` *Function to simulate tagged patch values in a single layer spatraster.*

Description

Function to simulate tagged patch values in a single layer spatraster.

Usage

```
simulate_raster_patch(
  original_raster,
  transition_matrix,
  non_trans = 0,
  iterations = 10
)
```

Arguments

`original_raster`	The SpatRaster object representing the categorical data.
`transition_matrix`	Transition matrix values that define transitions based on class values.
`non_trans`	Class(es) that the user would like to remain unsimulated. Defualt is class 0 since this is the background class.
`iterations`	The number of simulations to be performed.

Value

A SpatRaster where the patches values have been simulated.

Examples

```
 # Create a skeleton for the SpatRaster example landscape
#'
r <- terra::rast(nrows=100, ncols=100, xmin=0, xmax=100, ymin=0, ymax=100)

# Initialize all cells with class 1
terra::values(r) <- 1

# Define coordinates for the two 30x30 squares of class 2
# First square (top-left corner at (10, 10))
for (i in 10:39) {  # 10 to 39 makes a 30x30 square
  for (j in 10:39) {
    r[i, j] <- 2
  }
}

# Second square (top-left corner at (50, 50))
for (i in 50:79) {  # 50 to 79 makes a 30x30 square
  for (j in 50:79) {
    r[i, j] <- 2
  }
}

# Make transition matrix with with equal probabilities
n <- 3
trans_mat <- matrix(1/3, nrow = n, ncol = n)

# Assign row and column names
rownames(trans_mat) <- c("1", "2", "3")
colnames(trans_mat) <- c("1", "2", "3")


# Simulate the input SpatRaster 5 times
patch_sim <- simulate_raster_patch(original_raster = r,
                                   transition_matrix = trans_mat,
                                   iterations = 5)
```

Index