Package: LITAP (via r-universe)

August 24, 2024

Type Package

Title Landscape Integrated Terrain Analysis Package

Version 0.6.0

Description Terrain analysis and landscape and hydrology models built on terrain attributes. A major component of LITAP is founded on R. A. (Bob) MacMillan's LandMapR suite of programs for flow topology and landform segmentation analyses with extended new parameters and methodologies, as well as with new calculations and uses of directional terrain attributes.

License GPL-3

Encoding UTF-8

LazyData true

URL https://github.com/FRDC-SHL/LITAP

Depends R (>= 3.5.0)

Imports assertr (>= 2.7), dplyr (>= 0.7.0), DT (>= 0.2), ggplot2 (>= 3.2.0), glue (>= 1.4.2), gridExtra (>= 2.2.1), magrittr (>= 1.5), progress (>= 1.1.2), purrr (>= 0.2.2.2), raster (>= 2.5.8), readr (>= 1.1.0), rlang (>= 0.4.10), rmarkdown (>= 1.5), stringr (>= 1.2.0), tibble (>= 2.1.3), tidyselect (>= 1.1.0), tidyr (>= 1.0.0), writexl (>= 1.4.0)

Suggests gt (>= 0.3.1), foreign (>= 0.8.67), knitr, microbenchmark, readxl, rgdal, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.1.2

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

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

RemoteUrl https://github.com/FRDC-SHL/LITAP

RemoteRef HEAD

RemoteSha 763c565df3ec610cb8f8a17da24cd81c2a07e659

Contents

`facet_mapper` *Calculate landform facets using LSM logic*

Description

This function takes backup data frame output from `flow_mapper()` and `form_mapper()` calculates fuzzy attributes (among other metrics). Based on FacetMapR by R. A. (Bob) MacMillan, LandMapper Environmental Solutions.

Usage

```
facet_mapper(
  folder,
  arule = NULL,
  crule,
  edge_row = NULL,
  edge_col = NULL,
  procedure = "lsm",
  zone = NULL,
  clean = FALSE,
  resume = NULL,
  log = TRUE,
  verbose = FALSE,
  quiet = FALSE,
  debug = FALSE
)
```

Arguments

`folder` Character. Location of `flow_mapper()` output

`arule` Character. Location of ARULE file. If NULL, A Rules are derived from the dem file (see Details).

`crule` Character. Location of CRULE file `edge_row` Numeric. Number of rows to remove around the edge of the dem before deriving the A Rules. Default (NULL) results in removing 5% of the rows per side (total of 10%).

`edge_col` Numeric. Number of cols to remove around the edge of the dem before deriving the A Rules. Default (NULL) results in removing 5% of the cols per side (total of 10%).

`procedure` Character. Which procedure to use. One of `lsm` (Original LandMapR program) or `bc_pem` (newer BC-PEM Direct-to-Site-Series program).

`zone` file. If `procedure = "bc_pem"`, zones must either be defined for each seqno in the form dem file, OR must be provided as an index file here. With a `zone` defined for each `seqno`. `zone` file can be either dem (.dem), Excel (.xlsx, .xls), or text (.txt, .csv, .dat)

`clean` Logical. Remove all output files from previous runs in this folder?

`resume` Character. From which stage should the run be resumed? (see

`log` Logical. Create log file recording progress?

`verbose` Logical. Output extra progress messages.

`quiet` Logical. Suppress all messages.

`debug` Logical. If TRUE, output files contain intermediate columns useful for debugging (e.g., 'buffer', 'seqno_buffer', etc.) Default FALSE.

Details

Based on the technique described in Li et al. 2011, if no `arule` file is provided, the ARULE cutoffs are calculated from the `form_mapper()` dem files. These A Rules are saved as `afile_derived.csv` in the `folder` provided. The topographic derivative percentiles are stored to `topographic_derivatives.csv`, also in the `folder`.

Procedure `lsm` refers to the landform segmentation model (LSM) offered in the original LandMapR. Procedure `bc_pem` refers to calculating variables required for the British Columbia Predictive Ecosystem Mapping Direct-to-Site-Series program (BC-PEM DSS).

For resuming a run, `resume` must be one of the following:

- attributes
- classes

@references Sheng Li, David A. Lobb, Brian G. McConkey, R. A. MacMillan, Alan Moulin, and Walter R. Fraser. 2011. Extracting topographic characteristics of landforms typical of Canadian agricultural landscapes for agri-environmental modeling. I. Methodology. Canadian Journal of Soil Science 91(2), 251-266. doi:10.4141/CJSS10080.

Examples

```
# First need to run flow_mapper()
flow_mapper(file = system.file("extdata", "testELEV.dbf", package = "LITAP"),
           out_folder = "./testELEV/", nrow = 90, ncol = 90, grid = 5)


# And form_mapper()
```

```
form_mapper(folder = "./testELEV/")


crule <- system.file("extdata", "crule.dbf", package = "LITAP")

# Run facet_mapper() - Derive A Rules
facet_mapper(folder = "./testELEV/", arule = NULL, crule = crule)

# Derive A Rules, omitting rows and cols from the calculation
facet_mapper(folder = "./testELEV/", arule = NULL, crule = crule,
             edge_row = 3, edge_col = 1)

# Run facet_mapper() - supply A Rules
arule <- system.file("extdata", "arule.dbf", package = "LITAP")
crule <- system.file("extdata", "crule.dbf", package = "LITAP")
facet_mapper(folder = "./testELEV/", arule = arule, crule = crule)

# Clean up (remove all output)
unlink("./testELEV/", recursive = TRUE)
```

flow_mapper *Map flow through the landscape*

Description

Run an elevation file through all functions to calculate watershed flow and fill patterns. Based on FlowMapR by R. A. (Bob) MacMillan, LandMapper Environmental Solutions.

Usage

```
flow_mapper(
  file,
  nrow,
  ncol,
  grid = NULL,
  missing_value = -9999,
  max_area = 10,
  max_depth = 0.5,
  out_folder = NULL,
  out_format = "rds",
  clean = FALSE,
  clim = NULL,
  rlim = NULL,
  resume = NULL,
  log = TRUE,
  report = TRUE,
  verbose = FALSE,
  quiet = FALSE,
  debug = FALSE
)
```

Arguments

`file` Character. Elevation file (see `load_file`) for supported file types.

`nrow` Numeric. Number of rows in dem file (required for dbf files with a single column, but can be automatically assessed from files with x and y coordinates.

`ncol` Numeric. Number of columns in dem file (required for dbf files with a single column, but can be automatically assessed from files with x and y coordinates.

`grid` Numeric. Grid size in m of the input DEM file

`missing_value` Numeric/Character. Symbols which define missing data

`max_area` Numeric. Largest area of pits to be removed during initial pit removal

`max_depth` Numeric. Largest depth of pits to be removed during initial pit removal

`out_folder` Character. Folder in which to store output files. Defaults to folder in the same location and with the same name as the dem file

`out_format` Character. What format should the data be output as? "rds" for R data format (default), "csv" for Comma-separated values. This format is used for all subsequent functions (i.e. `form_mapper()`, `facet_mapper()` and `wepp_mapper()`.

`clean` Logical. Remove all backup files and output files from previous runs in this folder?

`clim` Numeric vector. Column limits if specifying a subset of the dem

`rlim` Numeric vector. Row limits if specifying a subset of the dem

`resume` Character. From which stage should the run be resumed? (see

`log` Logical. Create log file recording progress?

`report` Logical. Create html report of results?

`verbose` Logical. Output extra progress messages.

`quiet` Logical. Suppress all messages.

`debug` Logical. If TRUE, output files contain intermediate columns useful for debugging (e.g., 'buffer', 'seqno_buffer', etc.) Default FALSE.

Details

For information regarding loading other file types see `load_file`.

For resuming a run, `resume` must be one of the following:

1. `directions` (Calculating Directions)
2. `watersheds` (Calculating Watersheds)
3. `local` (Initial Pit Removal)
4. `pond` (Calculating Pond Shed Statistics - Second Pit Removal)
5. `fill` (Calculating Fill Shed Statistics - Third Pit Removal)

6. `slope` (Slope Gradient and Curvature values)
7. `idirections` (Calculating Directions on Inverted DEM)
8. `iwatersheds` (Calculating Inverted Watersheds)
9. `inverted` (Inverted Pit Removal)
10. `report` (Create the final report)

Examples

```
# Basic Run
flow_mapper(file = system.file("extdata", "testELEV.dbf", package = "LITAP"),
           out_folder = "./testELEV/", nrow = 90, ncol = 90, grid = 1)

# Specify parameters for initial pit removal
flow_mapper(file = system.file("extdata", "testELEV.dbf", package = "LITAP"),
            out_folder = "./testELEV/", nrow = 90, ncol = 90, grid = 1,
            max_area = 5, max_depth = 2)

# Clean up (remove created folder and output)
unlink("./testELEV/", recursive = TRUE)
```

`flow_plot` *Create plots of water flow*

Description

Plots water flow and watersheds. See the flow_plot article/vignette for examples.

Usage

```
flow_plot(
  db,
  type = "relief",
  dir = FALSE,
  seqno = FALSE,
  highlight = FALSE,
  shed = FALSE,
  shed_type = "local",
  pits = FALSE,
  upslope_threshold = NULL,
  cells = NULL,
  clim = NULL,
  rlim = NULL,
  stats = NULL,
  missing = NA
)
```

Arguments

`db` Data frame. Cell by cell data on the elevation of the watershed. Output by LITAP's `flow_mapper()` function.

`type` Character. Either relief or elevation. Defaults to relief.

`dir` Logical. Include flow directions?

`seqno` Logical. Include cell numbering?

`highlight` Logical. Highlight selected cells?

`shed` Logical. Show watersheds?

`shed_type` Character. Which type of watershed, must be included as a column in the data frame. Can be one of 'initial', 'local', 'fill', 'inv_initial', or 'inv_local'/'inverted'.

`pits` Logical. Show watershed pits (lowest point)

`upslope_threshold`
Numeric. If dir = TRUE, only show flow directions for cells with >= this many cells which drain to it.

`cells` Vector. Which cells to show

`clim` Numeric vector. Column limits in format c(0, 100)

`rlim` Numeric vector. Row limits in format c(0, 100)

`stats` Data frame. Data frame of watershed stats to highlight pour points.

`missing` Character. What is the value of missing data? Defaults to NA

`form_mapper` *Map form and relief of the landscape*

Description

This function takes backup data frame output from `flow_mapper()` and calculates form, wetness indices, reflief and stream/crest lengths (among other metrics). Based on FormMapR by R. A. (Bob) MacMillan, LandMapper Environmental Solutions.

Usage

```
form_mapper(
  folder,
  str_val = 10000,
  ridge_val = 10000,
  resume = NULL,
  log = TRUE,
  clean = FALSE,
  verbose = FALSE,
  quiet = FALSE,
  debug = FALSE
)
```

Arguments

`folder` Character. Location of `flow_mapper()` output

`str_val` Numeric. Definition of a stream (number of upslope cells)

`ridge_val` Numeric. Definition of a ridge (number of downslope cells)

`resume` Character. From which stage should the run be resumed? (see

`log` Logical. Create log file recording progress?

`clean` Logical. Remove all output files from previous runs in this folder?

`verbose` Logical. Output extra progress messages.

`quiet` Logical. Suppress all messages.

`debug` Logical. If TRUE, output files contain intermediate columns useful for debugging (e.g., 'buffer', 'seqno_buffer', etc.) Default FALSE.

Details

For resuming a run, `resume` must be one of the following:

1. `weti` (Calculating Wetness Indices)
2. `relief` (Calculating Relief Derivitives)
3. `length` (Calculating Slope Length)

Note that some variables have a version 1 and a version 2 (i.e. `qweti1` and `qweti2`). These reflect variables calculated (1) area based on number of cells vs. (2) area based on actual grid cell area values.

Examples

```
# First need to run flow_mapper()
flow_mapper(file = system.file("extdata", "testELEV.dbf", package = "LITAP"),
          out_folder = "./testELEV/", nrow = 90, ncol = 90, grid = 5)

# Now can run form_mapper()
form_mapper(folder = "./testELEV/")

# Clean up (remove all output)
unlink("./testELEV/", recursive = TRUE)
```

`load_file` *Load and prep elevation data*

Description

This function is used by the `mapper` functions to load input files and prepare them for analysis. It can also be used to load input files independently for plotting with `flow_plot` and/or trouble-shooting.

Usage

```
load_file(
  file,
  nrow = NULL,
  ncol = NULL,
  missing_value = -9999,
  rlim = NULL,
  clim = NULL,
  grid = NULL,
  edge = TRUE,
  verbose = TRUE
)
```

Arguments

`file`	Character. The location of the file containing elevation data. See details for accepted file types
`nrow`	Numeric. Number of rows in dem file (required for dbf files with a single column, but can be automatically assessed from files with x and y coordinates.
`ncol`	Numeric. Number of columns in dem file (required for dbf files with a single column, but can be automatically assessed from files with x and y coordinates.
`missing_value`	Vector. The number or character string specifying missing data.
`rlim`	Vector. Two numbers specifying the start and end of a subset of rows to extract
`clim`	Vector. Two numbers specifying the start and end of a subset of columns to extract
`grid`	Numeric. Grid size in m of the input DEM file
`edge`	Logical. Whether to add an edge (buffer) around the data.
`verbose`	Logical. Output extra progress messages.

Details

All x/y data must be in a format such that greater values indicate East or North respectively.

This function uses file extensions to guess the file type to be loaded.

dBase files: These files are loaded via the `read.dbf` function from the foreign package. Columns must be named and must have a valid name (case doesn't not matter). X/Y coordinates are optional and must be named as "x", "lon", "long", "longitude", or "y", "lat", "latitude". Elevation columns can be "elev", "elevation", or "z". If no "x" and "y" columns are suppled, `nrow` and `ncol` arguments must be supplied. Column names matter, column order does not. Extra columns, if present, are ignored.

- dBase files (.dbf)

Grid file types: These file types are loaded via the `raster` function.

- Surfer grid files (.grd)
- Esri grid files (binary .adf or ascii .asc)

- Geo Tiff (.tif)
- Floating point raster files (.flt) (**Note** the companion header, .hdr, file must also be present)

Text/Spreadsheet file types: Data in these files are all assumed to be arranged in three columns reflecting x, y, and z dimensions (z = elevation). Column **order** is important. Column names don't matter, but they should be present.

- Text files (.txt, .dat, .csv) are loaded via R base `link[utils]{read.table}` function.
- Excel files (.xls, .xlsx) are loaded via the `read_excel` function.

Value

Returns a data frame containing elevation data in a format suitable for analysis

merge_all *Combine flow and form output dems*

Description

`flow_mapper()` and `form_mapper()` each provide output information per cell of a dem file. This function takes the fill dem from `flow_mapper()` as well as the length and weti dem files from `form_mapper()` and merges them together into a complete dem file with all information. This file is saved to the project folder.

Usage

```
merge_all(folder, out_format = NULL)
```

Arguments

`folder`	Character. Folder with previous LITAP runs (i.e. where output of `flow_mapper()` etc. are)
`out_format`	Character. Output format (rds or csv) that merged file should be saved as (if different from the rest; by default uses the format of the other LITAP output files)

slope_gc *Calculate slope gradient and curvature, and hills*

Description

Computes row (east/west) and column (north/south) slope gradients and curvatures. Also calculates hill slopes as points when slope gradients switch directions

Usage

```
slope_gc(db, grid = 1)
```

Arguments

`db`	Dataframe dem
`grid`	Numeric. Grid size for the original dem

Details

Assume the following cells, and the calculations on elevation for focal point 5:

c7 c8 c9

c4 c5 c6

c1 c2 c3

Slope gradients and curvature

- Row slope gradient towards east (sgre): (c4 - c6) / (2 * grid)
  - Positive slope is downslope (and east-facing)
  - Negative slope is upslope (and west-facing)
- Row slope gradient (sgr): abs(sgre)
- Column slope gradient towards north (sgcn): (c2 - c8) / (2 * grid)
  - Positive slope is downslope (and north-facing)
  - Negative slope is upslope (and south-facing)
- Column slope gradient (sgc): abs(sgcn)
- Row slope curvature (scr): (2 * c5 - c4 - c6) / (grid^2)
- Column slope curvature (scc): (2 * c5 - c2 - c8) / (grid^2)

Where missing neighbours, assume same elevation as central point (i.e. assume an extension of the field).

Zero values replaced with arbitrarily small value, 0.00001

For gradients, zero values replaced with arbitrarily small value (0.00001) but with the sign of the previous point (i.e. to left (n4, west) if row; to bottom (n2, south) if column). If that previous point is missing, look to next point (i.e. n6 or n8). If cannot resolve by working through points, assign to positive.

Hill slopes

- hill_r_n: Number of hillslopes in that row. It goes from 1 to whatever it ends. Hillslope number changes when it goes from downslope to upslope or from upslope to downslope (SGRE changes sign)
- hill_r_dir: Direction the slope is facing, east facing as 2 and west facing as 4, all cells with the same hillslope no. have same hillslope direction
- hill_r_cell: Order number of the cell in the hillslope, starts from 1 to how many cells there are in a given hillslope
- hill_c_n: Same as for row, changes when SGCN changes sign
- hill_c_dir: Same as for row, north facing as 1 and south facing as 3
- hill_c_cell: Same as for row.

Directions:

N (1) -> E (2) -> S (3) -> W (4)

Examples

```
d <- slope_gc(test_dem)

library(ggplot2)
flow_plot(d, type = "elevation") +
  geom_point(aes(colour = factor(hill_r_n)))
```

---

test_dem *Test DEM*

Description

A small data frame for testing and running examples from helper functions. This is how DEMs look once imported by LITAP.

Usage

```
test_dem
```

Format

A data frame with 35154 rows and 6 variables:

seqno Cell number

elev Cell elevation

x Longitudinal value

y Latitudinal value

row Cell row

col Cell column

missing Missing?

buffer Buffer cell? Buffers are padding added to edges of a dem

wepp_mapper *Calculate spatial entities required for WEPP*

Description

This function takes backup data frame output from flow_mapper() and form_mapper() calculates hillslope, channel segments and impoundment spatial entities required for input in to WEPP (Water Erosion and Prediction Project) (among other metrics). Based on FacetMapR by R. A. (Bob) MacMillan, LandMapper Environmental Solutions.

Usage

```
wepp_mapper(
  folder,
  chan_length = 200,
  upslope_threshold = 300,
  clean = FALSE,
  resume = NULL,
  log = TRUE,
  verbose = FALSE,
  quiet = FALSE,
  debug = FALSE
)
```

Arguments

folder	Character. Location of flow_mapper() output
chan_length	Numeric. Channel length maximum length. Used to split channels into segments
upslope_threshold	Numeric. Threshold of upslope cells to define channel cells. #' Cells with an upslope value larger than this are considered channel cells
clean	Logical. Remove all output files from previous runs in this folder?
resume	Character. From which stage should the run be resumed? (see
log	Logical. Create log file recording progress?
verbose	Logical. Output extra progress messages.
quiet	Logical. Suppress all messages.
debug	Logical. If TRUE, output files contain intermediate columns useful for debugging (e.g., 'buffer', 'seqno_buffer', etc.) Default FALSE.

Examples

```
## Not run:
# First need to run flow_mapper()
flow_mapper(file = system.file("extdata", "testELEV.dbf", package = "LITAP"),
           out_folder = "./testELEV/", nrow = 90, ncol = 90, grid = 5)

# Now can run wepp_mapper()
wepp_mapper(folder = "./testELEV/")

## End(Not run)
```

Index