Package: SELECTRshed (via r-universe)

May 28, 2026

Type Package

Title Catchment and Subcatchment Delineation

Version 0.1.1

Description Use `terra` SpatRaster and SpatVector objects directly with WhiteBox Tools' stream and hydrology functions.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

SystemRequirements WhiteboxTools (https://github.com/jblindsay/whitebox-tools/releases/latest)

Additional_repositories https://txwri.r-universe.dev

Depends whitebox

Imports cli, fs, rlang, terra

Suggests SELECTRdata (>= 0.1.0), testthat (>= 3.0.0)

RoxygenNote 7.3.1

Config/testthat/edition 3

Config/pak/sysreqs cmake libgdal-dev gdal-bin libgeos-dev make libuv1-dev libproj-dev libsqlite3-dev

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

Date/Publication 2025-10-30 13:07:46 UTC

RemoteUrl https://github.com/TxWRI/SELECTRshed

RemoteRef HEAD

RemoteSha b4cdc8c83e3af67f6d685cb0448911828dd92e53

Contents

`create_breach_depression` *Breach depressions*

Description

Breaches all of the depressions in a DEM using Lindsay's (2016) algorithm. This should be preferred over depression filling in most cases.

Usage

```
create_breach_depression(
  dem,
  output = tempfile(fileext = ".tif"),
  whitebox_wd = NULL,
  max_depth = Inf,
  max_length = Inf,
  flat_increment = NULL,
  fill_pits = FALSE,
  ...
)
```

Arguments

`dem`	A character file path to a raster or a SpatRaster object.
`output`	A character file path specifying where the raster file should be stored. Defaults to a temporary file.
`whitebox_wd`	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
`max_depth`	numeric, Optional maximum breach depth (default is `Inf`).
`max_length`	numeric, Optional maximum breach channel length (in grid cells; default is `Inf`).
`flat_increment`	numeric, Optional elevation increment applied to flat areas. Default is `NULL`.

fill_pits	logical, Optional flag indicating whether to fill single-cell pits. Default is FALSE
...	optional arguments passed to whitebox::wbt()

Value

A SpatRaster object.

Examples

```
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
create_breach_depression(dem)
```

---

create_breach_depression_lc

Use the least cost pathway method to breach depressions

Description

Creates a DEM with breached depressions using the least cost pathway. This is a wrapper for whitebox::wbt_breach_depressions_least_cost() that accepts terra SpatRaster objects or a file path to common raster files.

Usage

```
create_breach_depression_lc(
  dem,
  output = tempfile(fileext = ".tif"),
  dist = 100,
  whitebox_wd = NULL,
  max_cost = Inf,
  min_dist = FALSE,
  flat_increment = NULL,
  fill = FALSE,
  ...
)
```

Arguments

dem	A character file path to a raster or a SpatRaster object.
output	A character file path specifying where the raster file should be stored. Defaults to a temporary file.
dist	Maximum search distance for breach paths in cells. Defaults to 100.

whitebox_wd	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
max_cost	numeric, Optional maximum breach cost (default is Inf).
min_dist	logical, Optional flag indicating whether to minimize breach distances. Default is FALSE.
flat_increment	Optional elevation increment applied to flat areas. Default is NULL.
fill	logical, Optional flag indicating whether to fill any remaining unbreached depressions. Default is FALSE
...	optional arguments passed to whitebox::wbt()

Value

A SpatRaster object.

Examples

```
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
create_breach_depression_lc(dem, dist = 10)
```

---

create_d8_fa *Create a D8 flow accumulation raster*

Description

Creates a flow accumulation grid from a D8 flow direction raster or imput DEM. This is a wrapper for whitebox::wbt_d8_flow_accumulation() that accepts terra SpatRaster objects or a file path to common raster files.

Usage

```
create_d8_fa(
  D8pointer,
  output = tempfile(fileext = ".tif"),
  whitebox_wd = NULL,
  out_type = "cells",
  log = FALSE,
  clip = FALSE,
  pntr = TRUE,
  esri_pntr = FALSE,
  ...
)
```

Arguments

D8pointer	A character file path to a raster or a SpatRaster object.
output	A character file path specifying where the raster file should be stored. Defaults to a temporary file.
whitebox_wd	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
out_type	A character value indicating the values calculated for each cell. One of `cells` (default), `catchment area`, or `specific contributing area`.
log	Optional flag to request the output be log-transformed.
clip	Optional flag to request clipping the display max by 1%.
pntr	Is the input raster a D8 flow pointer rather than a DEM?
esri_pntr	Input D8 pointer uses the ESRI style scheme.
...	optional arguments passed to `whitebox::wbt()`.

Value

A SpatRaster object.

Examples

```
# example code
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
D8pointer <- create_d8_pointer(dem)
create_d8_fa(D8pointer)
```

---

create_d8_pointer *Use D8 algorithm to create flow direction grid.*

Description

Creates a flow direction or pointer grid using the d8 algorithm. This is a wrapper for the whitebox tools `d8_pointer` that accepts terra SpatRaster objects or a file path to common raster files.

Usage

```
create_d8_pointer(
  dem,
  output = tempfile(fileext = ".tif"),
  whitebox_wd = NULL,
  esri_pntr = FALSE,
  ...
)
```

Arguments

dem	A character file path to a raster or a SpatRaster object.
output	A character file path specifying where the raster file should be stored. Defaults to a temporary file.
whitebox_wd	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
esri_pntr	logical, D8 pointer uses the ESRI style scheme.
...	optional arguments passed to whitebox::wbt()

Value

a SpatRaster object.

Examples

```
# example code
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
create_d8_pointer(dem)
```

---

create_fill_depressions

Use Whitebox Tools fill depression algorithm.

Description

Creates a DEM with all of the depressions filled and flat areas removed. This is a wrapper for whitebox::wbt_fill_depressions() that accepts terra SpatRaster objects or a file path to common raster files.

Usage

```
create_fill_depressions(
  dem,
  output = tempfile(fileext = ".tif"),
  whitebox_wd = NULL,
  fix_flats = TRUE,
  flat_increment = NULL,
  max_depth = NULL,
  ...
)
```

Arguments

dem	A character file path to a raster or a SpatRaster object.
output	A character file path specifying where the raster file should be stored. Defaults to a temporary file.
whitebox_wd	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
fix_flats	logical, indicating whether flat areas should have a small gradient applied. Defaults to TRUE.
flat_increment	Optional elevation increment applied to flat areas. Defaults to NULL.
max_depth	Optional maximum depression depth to fill. Defaults to NULL.
...	optional arguments passed to whitebox::wbt()

Value

A SpatRaster object.

Examples

```
# example code
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
create_fill_depressions(dem)
```

---

create_single_pit_fill

Fill single cell pits

Description

Wrapper for whitebox FillSingleCellPits. Raises pit cells to the elevation of their lowest neighbor.

Usage

```
create_single_pit_fill(
  dem,
  output = tempfile(fileext = ".tif"),
  whitebox_wd = NULL,
  ...
)
```

Arguments

dem	A character file path to a raster or a SpatRaster object.
output	A character file path specifying where the raster file should be stored. Defaults to a temporary file.
whitebox_wd	A valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
...	optional arguments passed to whitebox::wbt()

Value

A SpatRaster object

Examples

```
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
create_single_pit_fill(dem)
```

---

create_streams *Extract stream network*

Description

Creates a raster of likely stream cells from a flow-accumulation raster using the Whitebox Tools algorithm. This is a wrapper for whitebox::wbt_extract_streams() that accepts terra SpatRaster objects or a file path to common raster files.

Usage

```
create_streams(
  flow_accumulation,
  output = tempfile(fileext = ".tif"),
  whitebox_wd = NULL,
  threshold = 1000,
  zero_background = FALSE,
  ...
)
```

Arguments

flow_accumulation	SpatRaster object, Input raster D8 flow accumulation file.
output	A character file path specifying where the raster file should be stored. Defaults to a temporary file.
whitebox_wd	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
threshold	Numeric value indicating the threshold in flow accumulation values (number of cells) for channelization.
zero_background	logical, Flag indicating whether a background value of zero should be used.
...	optional arguments passed to `whitebox::wbt()`.

Value

A SpatRaster object.

Examples

```
# example code
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
D8pointer <- create_d8_pointer(dem)
D8fa <- create_d8_fa(D8pointer)
create_streams(D8fa)
```

---

create_streams_vector *Convert stream raster to vector*

Description

Creates a stream vector (terra SpatVector object) from a D8 flow direction raster and stream network raster. This is a wrapper for `whitebox::wbt_raster_streams_to_vector()` that accepts terra SpatRaster objects or a file path to common raster files.

Usage

```
create_streams_vector(
  streams,
  d8_pointer,
  output = tempfile(fileext = ".shp"),
  whitebox_wd = NULL,
```

```
  esri_pntr = FALSE,
  all_vertices = FALSE,
  ...
)
```

Arguments

streams	SpatRaster object, input raster streams.
d8_pointer	SpatRaster object, input raster D8 pointer.
output	A character file path specifying where the raster file should be stored. Defaults to a temporary file.
whitebox_wd	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
esri_pntr	logical, D8 pointer uses the ESRI style scheme.
all_vertices	logical, Do you want to preserve all vertices in output (i.e. no straight-line generalization).
...	optional arguments passed to `whitebox::wbt()`

Value

A terra SpatVector object

Examples

```
# example code
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
D8pointer <- create_d8_pointer(dem)
D8fa <- create_d8_fa(D8pointer)
streams <- create_streams(D8fa)
create_streams_vector(streams, D8pointer)
```

`create_subbasins` *Subbasins*

Description

Identifies the catchments, or sub-basin, draining to each link in a stream network.

Usage

```
create_subbasins(
  d8_pntr,
  streams,
  output = tempfile(fileext = ".tif"),
  esri_pntr = FALSE,
  whitebox_wd = NULL,
  type = "terra",
  ...
)
```

Arguments

d8_pntr	Input D8 pointer raster file.
streams	Input raster streams file.
output	A character file path specifying where the shapefile file should be stored. Defaults to a temporary file.
esri_pntr	D8 pointer uses the ESRI style scheme.
whitebox_wd	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
type	character, one of "terra" or "wbt". If type = "terra", returns an object of SpatVector. If type = "wbt", a wbt_result object is returned.
...	optional arguments passed to whitebox::wbt().

Value

SpatRaster

Examples

```
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
D8pointer <- create_d8_pointer(dem)
D8fa <- create_d8_fa(D8pointer)
streams <- create_streams(D8fa)
create_subbasins(D8pointer, streams)
```

`create_watershed` *Watershed*

Description

Identifies the watershed, or drainage basin, draining to a set of target cells.

Usage

```
create_watershed(
  d8_pntr,
  pour_pts,
  output = tempfile(fileext = ".tif"),
  esri_pntr = FALSE,
  whitebox_wd = NULL,
  type = "terra",
  ...
)
```

Arguments

`d8_pntr`	Input D8 pointer.
`pour_pts`	Input pour points (outlet).
`output`	A character file path specifying where the shapefile file should be stored. Defaults to a temporary file.
`esri_pntr`	D8 pointer uses the ESRI style scheme.
`whitebox_wd`	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
`type`	character, one of `"terra"` or `"wbt"`. If `type = "terra"`, returns an object of SpatVector. If `type = "wbt"`, a wbt_result object is returned.
`...`	optional arguments passed to `whitebox::wbt()`.

Value

A SpatRaster or wbt_result object.

Examples

```
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
gpkg <- system.file("extdata", "thompsoncreek.gpkg", package = "SELECTRdata")
pourpoint <- terra::vect(gpkg, layer = "pourpoint", crs = terra::crs(dem))
D8pointer <- create_d8_pointer(dem)
```

```
D8fa <- create_d8_fa(D8pointer)
streams_ras <- create_streams(D8fa)
## write pourpoints to temp folder
temp_pour_point_file <- tempfile("snapped", fileext = ".shp")
snapped_pour_point <- snap_pour_point(pour_pts = pourpoint,
                                      streams = streams_ras,
                                      output = temp_pour_point_file)
snapped_pour_point <- terra::vect(temp_pour_point_file)
pour_point_rast <-  terra::rasterize(snapped_pour_point, streams_ras)
create_watershed(d8_pntr = D8pointer, pour_pts = pour_point_rast)
## cleanup temp files
unlink(tempdir(), recursive = FALSE)
```

`snap_pour_point` *Jenson snap pour points*

Description

Moves outlet points used to specify points of interest in a watershedding operation to the nearest stream cell.

Usage

```
snap_pour_point(
  pour_pts,
  streams,
  output = tempfile(fileext = ".shp"),
  whitebox_wd = NULL,
  snap_dist = 90,
  type = "terra",
  ...
)
```

Arguments

`pour_pts`	Input vector pour points (outlet) file.
`streams`	Input raster streams file.
`output`	A character file path specifying where the shapefile file should be stored. Defaults to a temporary file.
`whitebox_wd`	valid working directory for whitebox to store temporary rasters. Defaults to NULL which stores in a temporary space that is deleted when the user session is over. You can use this to store rasters before they are processed into terra::rast objects. Otherwise it is suggested to leave this NULL and store the rasters using the output argument.
`snap_dist`	Maximum snap distance in map units.

type	character, one of "terra" or "wbt". If type = "terra", returns an object of SpatVector. If type = "wbt", a wbt_result object is returned.
...	optional arguments passed to whitebox::wbt().

Value

terra SpatVector object.

Examples

```
## example data
dem <- system.file("extdata", "thompsoncreek.tif", package = "SELECTRdata")
dem <- terra::rast(dem)
gpkg <- system.file("extdata", "thompsoncreek.gpkg", package = "SELECTRdata")
pourpoint <- terra::vect(gpkg, layer = "pourpoint", crs = terra::crs(dem))

## create flow direction and flow accumulation rasters
D8pointer <- create_d8_pointer(dem)
D8fa <- create_d8_fa(D8pointer)

## create streams network raster
streams_ras <- create_streams(D8fa)

## write pourpoints to temp folder
temp_pour_point_file <- tempfile("snapped", fileext = ".shp")

## snap pour points to stream raster network
snapped_pour_point <- snap_pour_point(pour_pts = pourpoint,
                                      streams = streams_ras,
                                      output = temp_pour_point_file)

## you probably need a raster of the pour point, you need to call
## terra::vect since it is a SpatVectorProxy
pour_point_rast <-  terra::rasterize(terra::vect(temp_pour_point_file), streams_ras)

## cleanup temp files
unlink(tempdir(), recursive = FALSE)
```

Index