

Package: evacpath (via r-universe)

September 16, 2026

Title Least-Cost Pedestrian Evacuation Modeling

Version 0.2.0

Description Tools for road-constrained, least-cost pedestrian evacuation modeling. The package provides reusable functions for preparing hazard zones, generating road-based evacuation origin points, identifying escape/safety points, creating slope-based conductance surfaces, calculating least-cost distance to safety, and converting distance outputs into evacuation-time polygons. It is designed to support workflows like tsunami evacuation modeling while remaining adaptable to other regions and hazards. Tsunami-specific helpers support separate land-only hazard zones, water-combined escape zones, road-aware escape boundaries, and study-area inset cropping for quality assurance and quality control. Methods build on Cordero et al. (2025) <[doi:10.1007/s44367-025-00018-y](https://doi.org/10.1007/s44367-025-00018-y)>, Lewis (2021) <[doi:10.1007/s10816-021-09522-w](https://doi.org/10.1007/s10816-021-09522-w)>, and Joseph Lewis's 'leastcostpath' package (2023) <<https://CRAN.R-project.org/package=leastcostpath>>.

License MIT + file LICENSE

URL <https://el-cordero.github.io/evacpath/>,
<https://github.com/el-cordero/evacpath>

BugReports <https://github.com/el-cordero/evacpath/issues>

Language en-US

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1)

Imports terra, leastcostpath, sf, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0), devtools, fields,
pkgdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/website r-lib/pkgdown

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libglpk-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://el-cordero.r-universe.dev>

Date/Publication 2026-07-18 02:02:15 UTC

RemoteUrl <https://github.com/el-cordero/evacpath>

RemoteRef HEAD

RemoteSha 6330b4529c17fad7700065a6c991eb6cab98097a

Contents

evacpath-package	3
calc_evac_time	3
calc_min_distance_to_safety	4
calculate_lc_path	5
calculate_min_dist	6
clean_roads	6
compare_evac_scenarios	7
crop_roads_to_inner_extent	8
diagnose_evac_model	9
find_escape_points	10
has_errors	11
interpolate_distance_surface	12
make_conductance_surface	13
make_evac_grid	14
make_evac_polygons	15
make_output_clip_area	16
make_region_area	16
make_road_aware_escape_zone	17
make_road_mask	18
make_road_origins	19
make_roads_in_zone	20
map_evac_bottlenecks	21
prepare_evac_inputs	22
prepare_hazard_zone	23
prepare_tsunami_zones	24
read_spatial	25
run_evacpath	26
validate_evac_routes	29
write_evac_outputs	30

Index

32

evacpath-package	<i>evacpath: Least-cost pedestrian evacuation modeling in R</i>
------------------	---

Description

evacpath provides tools for modeling pedestrian evacuation distance and travel time from hazard zones to safety areas using road-constrained least-cost path analysis.

Details

The package is organized as a set of modular functions. Typical workflows use `prepare_tsunami_zones()` to create tsunami-specific hazard and escape zones, `crop_roads_to_inner_extent()` and `make_road_aware_escape_zones()` to prevent false escape points along artificial study-area boundaries while retaining bridge and walkway corridors, `find_escape_points()` to identify candidate exits, `make_conductance_surface()` to build a slope-based conductance surface, and `run_evacpath()` to run the full workflow.

Author(s)

Maintainer: Elvin Cordero <elvin.cordero@seamountgeo.com>

See Also

Useful links:

- <https://el-cordero.github.io/evacpath/>
- <https://github.com/el-cordero/evacpath>
- Report bugs at <https://github.com/el-cordero/evacpath/issues>

calc_evac_time	<i>Convert distance to evacuation time</i>
----------------	--

Description

Convert distance to evacuation time

Usage

```
calc_evac_time(distance_m, walking_speed_mps = 1.22, units = "minutes")
```

Arguments

distance_m	Distance in meters.
walking_speed_mps	Walking speed in meters per second. The paper-style default is 1.22 m/s, but this should be changed for local planning scenarios.
units	Output units: "minutes", "seconds", or "hours".

Value

Numeric vector of evacuation times.

Examples

```
calc_evac_time(c(0, 120, 240), walking_speed_mps = 1.2)
```

```
calc_min_distance_to_safety
```

Calculate minimum least-cost distance from origins to safety

Description

For each origin point, calculates least-cost paths to all candidate safety points and stores the shortest finite path distance. Destination points are appended to the output with distance equal to zero.

Usage

```
calc_min_distance_to_safety(
  cs,
  origins,
  destinations,
  include_destinations = TRUE,
  progress = FALSE,
  progress_every = 1L,
  check_locations = FALSE,
  return_routes = FALSE
)
```

Arguments

cs	A leastcostpath conductance surface.
origins	Origin points.
destinations	Escape/safety destination points.
include_destinations	Logical. Add destination points with distance = 0.
progress	Logical. Print simple progress messages.
progress_every	Integer. Print progress every n origins when progress = TRUE.
check_locations	Logical passed to leastcostpath::create_lcp(). The default is FALSE for speed after inputs have already been projected and cropped.
return_routes	Logical. If TRUE, return the selected shortest route for each reachable origin alongside the distance points. The default is FALSE to preserve the original return type and avoid storing route geometries when they are not needed.

Value

A point `SpatVector` with columns `distance` and `type`. When `return_routes = TRUE`, returns a list with `distance_points`, `routes`, and `unreachable_origins`.

Examples

```
dem <- terra::rast(nrows = 5, ncols = 5, xmin = 0, xmax = 5, ymin = 0, ymax = 5,
  vals = 1, crs = "EPSG:3857")
cs <- make_conductance_surface(dem)
origins <- terra::vect(data.frame(x = 0.5, y = 0.5), geom = c("x", "y"), crs = "EPSG:3857")
destinations <- terra::vect(data.frame(x = 4.5, y = 4.5), geom = c("x", "y"), crs = "EPSG:3857")
calc_min_distance_to_safety(cs, origins, destinations)
```

calculate_lc_path	<i>Calculate a least-cost path between one origin and one destination</i>
-------------------	---

Description

Compatibility wrapper around `leastcostpath::create_lcp()` that returns `NULL` when a path cannot be created.

Usage

```
calculate_lc_path(cs, origin, destination)
```

Arguments

<code>cs</code>	A <code>leastcostpath</code> conductance surface.
<code>origin</code>	Origin point.
<code>destination</code>	Destination point.

Value

A least-cost path object or `NULL`.

Examples

```
dem <- terra::rast(nrows = 5, ncols = 5, xmin = 0, xmax = 5, ymin = 0, ymax = 5,
  vals = 1, crs = "EPSG:3857")
cs <- make_conductance_surface(dem)
origin <- terra::vect(data.frame(x = 0.5, y = 0.5), geom = c("x", "y"), crs = "EPSG:3857")
destination <- terra::vect(data.frame(x = 4.5, y = 4.5), geom = c("x", "y"), crs = "EPSG:3857")
calculate_lc_path(cs, origin, destination)
```

calculate_min_dist	<i>Calculate the minimum path distance from a list of least-cost paths</i>
--------------------	--

Description

Calculate the minimum path distance from a list of least-cost paths

Usage

```
calculate_min_dist(lc_paths_list)
```

Arguments

`lc_paths_list` A list of least-cost path line vectors.

Value

Minimum non-zero, finite path distance.

Examples

```
line <- terra::vect(matrix(c(0, 0, 1, 1), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
calculate_min_dist(list(line))
```

clean_roads	<i>Clean a road/pathway network</i>
-------------	-------------------------------------

Description

Removes user-specified features from a road/pathway layer. This is useful for excluding piers, tunnels, private ways, or other features that should not be used in pedestrian evacuation modeling.

Usage

```
clean_roads(roads, exclude = NULL, target_crs = NULL)
```

Arguments

<code>roads</code>	A <code>SpatVector</code> or path to a vector layer.
<code>exclude</code>	Optional named list with field and values, for example <code>list(field = "man_made", values = "pier")</code> .
<code>target_crs</code>	Optional output CRS.

Value

A cleaned `SpatVector`.

Examples

```
roads <- terra::vect(
  list(
    matrix(c(0, 0, 0, 1), ncol = 2, byrow = TRUE),
    matrix(c(1, 0, 1, 1), ncol = 2, byrow = TRUE)
  ),
  type = "lines",
  crs = "EPSG:3857"
)
roads$kind <- c("road", "pier")
clean_roads(roads, exclude = list(field = "kind", values = "pier"))
```

compare_evac_scenarios

Compare evacuation scenarios

Description

Runs `run_evacpath()` repeatedly across named scenario overrides and returns scenario summaries with combined spatial outputs when those layers are available.

Usage

```
compare_evac_scenarios(
  scenarios,
  ...,
  scenario_id_col = "scenario",
  keep_routes = FALSE,
  quiet = FALSE
)
```

Arguments

<code>scenarios</code>	Named list. Each element is a named list of arguments that override shared arguments passed through ...
<code>...</code>	Shared arguments passed to <code>run_evacpath()</code> .
<code>scenario_id_col</code>	Name of the scenario identifier column added to combined spatial outputs.
<code>keep_routes</code>	Logical. Retain selected least-cost routes in each scenario result. Use TRUE when route-density analysis is needed.
<code>quiet</code>	Logical. If TRUE, suppress warnings when a scenario fails or summary metrics are unavailable.

Value

An `evac_scenario_comparison` list with results, summary, distance_points, time_polygons, and metadata.

Examples

```
dem <- terra::rast(nrows = 7, ncols = 7, xmin = 0, xmax = 7, ymin = 0, ymax = 7,
  vals = 1, crs = "EPSG:3857")
hazard <- terra::as.polygons(terra::crop(dem, terra::ext(1, 6, 1, 6)), dissolve = TRUE)
roads <- terra::vect(matrix(c(0, 3.5, 7, 3.5), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
comparison <- compare_evac_scenarios(
  scenarios = list(baseline = list(), slower = list(walking_speed_mps = 0.9)),
  hazard_zone = hazard, roads = roads, dem = dem, grid_resolution = 1,
  road_buffer_m = 0.2, escape_buffer_m = 0.3, final_road_buffer_m = 0.2,
  max_origins = 1, max_destinations = 2, seed = 1
)
comparison$summary
```

crop_roads_to_inner_extent

Crop roads to an inset extent before escape-point detection

Description

Crops a road/pathway layer to a slightly reduced bounding box around a zone. This is useful before escape-point detection because roads extending beyond the study-area coverage can intersect artificial raster or polygon extent edges and create false escape/safety points.

Usage

```
crop_roads_to_inner_extent(roads, zone, inset_x_m = 250, inset_y_m = 250)
```

Arguments

<code>roads</code>	Road/pathway network as a <code>SpatVector</code> or file path.
<code>zone</code>	Zone used to define the outer extent. Can be a <code>SpatVector</code> , <code>SpatRaster</code> , or file path.
<code>inset_x_m</code>	Numeric. Distance to inset the minimum and maximum x boundaries, in map units. Use meters when the data are projected.
<code>inset_y_m</code>	Numeric. Distance to inset the minimum and maximum y boundaries, in map units. Use meters when the data are projected.

Value

A cropped road/pathway `SpatVector`.

Examples

```
r <- terra::rast(nrows = 4, ncols = 4, xmin = 0, xmax = 4, ymin = 0, ymax = 4,
  vals = 1, crs = "EPSG:3857")
zone <- terra::as.polygons(r, dissolve = TRUE)
roads <- terra::vect(matrix(c(-1, 2, 5, 2), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
crop_roads_to_inner_extent(roads, zone, inset_x_m = 0.5, inset_y_m = 0)
```

diagnose_evac_model	<i>Diagnose evacuation-model inputs</i>
---------------------	---

Description

Runs spatial quality assurance and quality control checks before or after evacuation modeling. Unlike the modeling workflow, diagnostics collect issues into a report so several input problems can be reviewed together.

Usage

```
diagnose_evac_model(
  hazard_zone,
  roads,
  dem,
  target_crs = NULL,
  escape_zone = NULL,
  origins = NULL,
  destinations = NULL,
  conductance = NULL,
  check_reachability = TRUE,
  sample_size = 100,
  ...
)
```

Arguments

<code>hazard_zone</code>	Hazard-zone polygon, raster, or file path.
<code>roads</code>	Road/pathway line layer or file path.
<code>dem</code>	Digital elevation model raster or file path.
<code>target_crs</code>	Optional target coordinate reference system.
<code>escape_zone</code>	Optional escape-boundary zone.
<code>origins</code>	Optional modeled origin points.
<code>destinations</code>	Optional escape/safety destination points.
<code>conductance</code>	Optional leastcostpath conductance surface. When supplied with origins and destinations, routing checks are run.

check_reachability Logical. Attempt sampled routing checks when routing inputs are available.

sample_size Maximum number of origins used for sampled reachability checks.

... Reserved for future extensions.

Value

An evac_diagnostics list with issues, summary, diagnostic_layers, and metadata.

Examples

```
dem <- terra::rast(nrows = 5, ncols = 5, xmin = 0, xmax = 5, ymin = 0, ymax = 5,
  vals = 1, crs = "EPSG:3857")
hazard <- terra::as.polygons(dem, dissolve = TRUE)
roads <- terra::vect(matrix(c(0, 2.5, 5, 2.5), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
diagnostics <- diagnose_evac_model(hazard, roads, dem)
diagnostics
has_errors(diagnostics)
```

find_escape_points	<i>Identify escape/safety points where roads cross the hazard-zone boundary</i>
--------------------	---

Description

Intersects a road/pathway network with the boundary of the hazard zone and converts the intersection geometry to points. These points represent candidate exits from the hazard zone.

Usage

```
find_escape_points(
  hazard_zone,
  roads,
  study_area = NULL,
  region_buffer_m = 5000
)
```

Arguments

hazard_zone Hazard/evacuation zone.

roads Road/pathway network.

study_area Optional local study area used to crop candidate escape points to a broader region around the study area.

region_buffer_m Buffer distance passed to make_region_area() when study_area is supplied.

Value

A point SpatVector of candidate escape/safety points.

Examples

```
r <- terra::rast(nrows = 4, ncols = 4, xmin = 0, xmax = 4, ymin = 0, ymax = 4,
  vals = 1, crs = "EPSG:3857")
zone <- terra::as.polygons(r, dissolve = TRUE)
roads <- terra::vect(matrix(c(-1, 2, 5, 2), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
find_escape_points(zone, roads)
```

has_errors	<i>Test whether diagnostics contain errors</i>
------------	--

Description

Test whether diagnostics contain errors

Usage

```
has_errors(x, ...)
```

Arguments

x	An object returned by <code>diagnose_evac_model()</code> .
...	Additional arguments passed to methods.

Value

Logical value indicating whether diagnostics contain at least one error.

Examples

```
dem <- terra::rast(nrows = 2, ncols = 2, vals = 1, crs = "EPSG:3857")
hazard <- terra::as.polygons(dem, dissolve = TRUE)
roads <- terra::vect(matrix(c(0, 1, 2, 1), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
has_errors(diagnose_evac_model(hazard, roads, dem))
```

interpolate_distance_surface

Interpolate a distance surface from least-cost distance points

Description

Uses thin-plate spline interpolation to create a continuous distance surface. This preserves the original paper-script approach while keeping interpolation optional. For most package workflows, `make_evac_polygons()` is the simpler output.

Usage

```
interpolate_distance_surface(
  distance_points,
  region_area,
  study_area,
  resolution_coarse = 100,
  resolution_fine = 1,
  distance_col = "distance"
)
```

Arguments

<code>distance_points</code>	Point layer with a distance column.
<code>region_area</code>	Broader analysis region used for interpolation extent.
<code>study_area</code>	Final study area used to crop and mask the output.
<code>resolution_coarse</code>	Coarse interpolation resolution.
<code>resolution_fine</code>	Fine output resolution.
<code>distance_col</code>	Name of distance column in <code>distance_points</code> .

Value

A `SpatRaster` distance surface.

Examples

```
if (requireNamespace("fields", quietly = TRUE)) {
  pts <- terra::vect(
    data.frame(
      x = c(0, 1, 0, 1, 0.5, 1.5),
      y = c(0, 0, 1, 1, 0.5, 1.5),
      distance = c(0, 5, 10, 15, 7, 20)
    ),
    geom = c("x", "y"),
```

```

    crs = "EPSG:3857"
  )
  area <- terra::as.polygons(terra::rast(nrows = 2, ncols = 2, xmin = -1, xmax = 2,
    ymin = -1, ymax = 2, vals = 1, crs = "EPSG:3857"), dissolve = TRUE)
  interpolate_distance_surface(pts, area, area, resolution_coarse = 0.5, resolution_fine = 1)
}

```

make_conductance_surface

Create a slope-based conductance surface

Description

Masks a DEM to an optional road/pathway mask and creates a slope conductance surface using `leastcostpath`.

Usage

```

make_conductance_surface(
  dem,
  road_mask = NULL,
  resolution = NULL,
  method = "slope",
  lcp_cost_function = "tobler",
  lcp_neighbours = 16,
  lcp_crit_slope = 12,
  lcp_max_slope = NULL,
  ...
)

```

Arguments

<code>dem</code>	Elevation raster.
<code>road_mask</code>	Optional road/pathway mask.
<code>resolution</code>	Optional target DEM resolution before conductance creation.
<code>method</code>	Conductance method. Currently only "slope" is implemented.
<code>lcp_cost_function</code>	Character string or function passed to <code>leastcostpath::create_slope_cs()</code> .
<code>lcp_neighbours</code>	Neighbourhood passed to <code>leastcostpath::create_slope_cs()</code> . Use 4, 8, 16, 32, 48, or a custom matrix accepted by <code>leastcostpath</code> .
<code>lcp_crit_slope</code>	Numeric critical slope passed to <code>leastcostpath::create_slope_cs()</code> .
<code>lcp_max_slope</code>	Optional numeric maximum slope passed to <code>leastcostpath::create_slope_cs()</code> .
<code>...</code>	Additional named arguments passed to <code>leastcostpath::create_slope_cs()</code> , such as exaggeration.

Value

A leastcostpath conductance surface object.

Examples

```
dem <- terra::rast(nrows = 5, ncols = 5, xmin = 0, xmax = 5, ymin = 0, ymax = 5,
  vals = 1, crs = "EPSG:3857")
make_conductance_surface(dem, lcp_neighbours = 8)
```

make_evac_grid	Create an evacuation grid
----------------	---------------------------

Description

Creates polygon grid cells over a hazard/evacuation zone and masks the grid to the zone. The resulting cells can be intersected with buffered roads to create road-based origin points.

Usage

```
make_evac_grid(hazard_zone, resolution)
```

Arguments

- hazard_zone Hazard/evacuation zone as SpatVector, SpatRaster, or file path.
- resolution Grid cell resolution in map units. Use meters when data are in a projected CRS. Can be length 1 or 2.

Value

A polygon SpatVector grid clipped/masked to the hazard zone.

Examples

```
r <- terra::rast(nrows = 4, ncols = 4, xmin = 0, xmax = 4, ymin = 0, ymax = 4,
  vals = 1, crs = "EPSG:3857")
zone <- terra::as.polygons(r, dissolve = TRUE)
make_evac_grid(zone, resolution = 1)
```

make_evac_polygons	Create evacuation-distance and evacuation-time polygons
--------------------	---

Description

Creates Voronoi polygons from least-cost distance points, converts distance to travel time, and optionally clips the output to an inundated road/study mask.

Usage

```
make_evac_polygons(  
  distance_points,  
  clip_area = NULL,  
  walking_speed_mps = 1.22,  
  region_name = NULL,  
  distance_col = "DistToSafety",  
  time_col = "EvacTimeAvg"  
)
```

Arguments

distance_points	Point output from calc_min_distance_to_safety().
clip_area	Optional polygon used to crop the Voronoi output.
walking_speed_mps	Walking speed in meters per second.
region_name	Optional region/municipality name stored in the output.
distance_col	Name of the output distance column.
time_col	Name of the output time column.

Value

A polygon SpatVector.

Examples

```
pts <- terra::vect(  
  data.frame(x = c(0, 1, 0, 1), y = c(0, 0, 1, 1), distance = c(0, 5, 10, 15)),  
  geom = c("x", "y"),  
  crs = "EPSG:3857"  
)  
clip <- terra::as.polygons(terra::rast(nrows = 2, ncols = 2, xmin = -1, xmax = 2,  
  ymin = -1, ymax = 2, vals = 1, crs = "EPSG:3857"), dissolve = TRUE)  
make_evac_polygons(pts, clip_area = clip)
```

make_output_clip_area *Make an output clip area for evacuation polygons*

Description

Make an output clip area for evacuation polygons

Usage

```
make_output_clip_area(
  hazard_zone,
  roads_buffer,
  final_road_buffer_m = 3,
  clip_mode = c("hazard", "road_hazard", "hazard_plus_roads", "none")
)
```

Arguments

hazard_zone	Hazard zone polygon.
roads_buffer	Buffered roads.
final_road_buffer_m	Additional road buffer for output clipping.
clip_mode	One of "hazard", "road_hazard", "hazard_plus_roads", or "none".

Value

A SpatVector clip area or NULL.

make_region_area *Create a broader analysis region around a study area*

Description

Dissolves the full hazard zone, buffers a smaller study area, and crops the full zone to that buffer. This is useful when escape/safety points outside a municipality or local study area should still be considered.

Usage

```
make_region_area(hazard_zone, study_area, buffer_m = 5000)
```

Arguments

hazard_zone	Full hazard/evacuation zone.
study_area	Local study area.
buffer_m	Buffer distance in map units, typically meters.

Value

A polygon SpatVector for the broader analysis region.

Examples

```
r <- terra::rast(nrows = 4, ncols = 4, xmin = 0, xmax = 4, ymin = 0, ymax = 4,
  vals = 1, crs = "EPSG:3857")
zone <- terra::as.polygons(r, dissolve = TRUE)
study <- terra::as.polygons(terra::crop(r, terra::ext(1, 3, 1, 3)), dissolve = TRUE)
make_region_area(zone, study, buffer_m = 1)
```

make_road_aware_escape_zone

Add buffered road corridors to an escape-boundary zone

Description

Creates a road-aware escape zone by combining a base escape zone with buffered road/pathway corridors. This is useful for tsunami workflows in coastal cities where bridges, causeways, or other walkways over water can be lost when the inundation layer is split into land and water masks. The resulting object can be passed to `find_escape_points()` so escape/safety points are generated from a boundary that includes relevant road corridors as well as the tsunami zone.

Usage

```
make_road_aware_escape_zone(
  escape_zone,
  roads,
  road_buffer_m = 2,
  crop_buffer_m = 3,
  include_base_zone = TRUE
)
```

Arguments

escape_zone	Base escape-boundary zone, usually the land-inundation-plus- water zone from <code>prepare_tsunami_zones()</code> .
roads	Road/pathway network used to create the road-aware corridor.
road_buffer_m	First road buffer distance in map units.
crop_buffer_m	Optional second buffer applied before cropping/combining.
include_base_zone	Logical. If TRUE, combine the buffered roads with the original escape_zone. If FALSE, only the buffered road corridor is returned.

Value

A dissolved SpatVector escape-boundary zone.

Examples

```
r <- terra::rast(nrows = 4, ncols = 4, xmin = 0, xmax = 4, ymin = 0, ymax = 4,
  vals = 1, crs = "EPSG:3857")
zone <- terra::as.polygons(r, dissolve = TRUE)
roads <- terra::vect(matrix(c(0, 2, 4, 2), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
make_road_aware_escape_zone(zone, roads, road_buffer_m = 0.1)
```

make_road_mask	<i>Create a road-constrained analysis mask</i>
----------------	--

Description

Buffers the road network and candidate escape points, combines those buffered areas, and dissolves the result. The mask is used to constrain least-cost movement to the road/pathway network while allowing access around escape locations.

Usage

```
make_road_mask(
  roads,
  escape_points,
  road_buffer_m = 2,
  escape_buffer_m = 5,
  return_components = FALSE
)
```

Arguments

roads	Road/pathway network.
escape_points	Candidate escape/safety points.
road_buffer_m	Road buffer distance in map units, typically meters.
escape_buffer_m	Escape-point buffer distance in map units, typically meters.
return_components	Logical. If TRUE, return a list with the mask and component buffers.

Value

A dissolved road mask SpatVector, or a list when return_components = TRUE.

Examples

```
roads <- terra::vect(matrix(c(0, 1, 4, 1), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
escape <- terra::vect(data.frame(x = 4, y = 1), geom = c("x", "y"), crs = "EPSG:3857")
make_road_mask(roads, escape, road_buffer_m = 0.1, escape_buffer_m = 0.2)
```

make_road_origins	Create road-based origin points inside the evacuation zone
-------------------	--

Description

Intersects an evacuation grid with a buffered road network and converts the resulting road-crossing cell geometry to points.

Usage

```
make_road_origins(
  evac_grid,
  roads_buffer,
  hazard_zone = NULL,
  max_origins = NULL,
  seed = NULL
)
```

Arguments

evac_grid	Polygon grid from make_evac_grid().
roads_buffer	Buffered road/pathway network.
hazard_zone	Optional hazard-zone polygon used to crop candidate origins after intersecting the grid with the road buffer.
max_origins	Optional maximum number of origin points to retain. Useful for large regions or exploratory runs.
seed	Random seed used when max_origins is supplied.

Value

A point SpatVector of road-based evacuation origins.

Examples

```
r <- terra::rast(nrows = 4, ncols = 4, xmin = 0, xmax = 4, ymin = 0, ymax = 4,
  vals = 1, crs = "EPSG:3857")
grid <- make_evac_grid(terra::as.polygons(r, dissolve = TRUE), resolution = 1)
roads <- terra::vect(matrix(c(0, 2, 4, 2), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
roads_buffer <- terra::buffer(roads, 0.2)
make_road_origins(grid, roads_buffer, hazard_zone = terra::as.polygons(r, dissolve = TRUE),
  max_origins = 3, seed = 1)
```

make_roads_in_zone	<i>Create a buffered road area inside an inundation or analysis zone</i>
--------------------	--

Description

Mirrors the original script logic where roads were buffered, optionally buffered again for tolerance, cropped to the inundation zone, and then combined with the zone. This is useful for quality assurance and quality control and for reproducing the earlier road-plus-inundation analysis area.

Usage

```
make_roads_in_zone(
  roads,
  zone,
  road_buffer_m = 2,
  crop_buffer_m = 3,
  include_zone = TRUE
)
```

Arguments

roads	Road/pathway network.
zone	Polygon/raster zone used to crop buffered roads.
road_buffer_m	First road buffer distance.
crop_buffer_m	Optional second buffer applied before cropping.
include_zone	Logical. If TRUE, combine the cropped road buffer with zone and dissolve the result.

Value

A SpatVector.

Examples

```
r <- terra::rast(nrows = 4, ncols = 4, xmin = 0, xmax = 4, ymin = 0, ymax = 4,
  vals = 1, crs = "EPSG:3857")
zone <- terra::as.polygons(r, dissolve = TRUE)
roads <- terra::vect(matrix(c(0, 2, 4, 2), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
make_roads_in_zone(roads, zone, road_buffer_m = 0.1)
```

map_evac_bottlenecks *Map modeled evacuation bottlenecks*

Description

Creates a route-density raster and identifies high-use modeled evacuation corridors from supplied least-cost paths or an [run_evacpath\(\)](#) result created with `keep_routes = TRUE`.

Usage

```
map_evac_bottlenecks(
  routes = NULL,
  evac_result = NULL,
  template = NULL,
  quantile_threshold = 0.9,
  min_count = NULL,
  rescale = FALSE,
  return_polygons = TRUE,
  ...
)
```

Arguments

<code>routes</code>	An sf or SpatVector line layer of modeled routes.
<code>evac_result</code>	Optional output from run_evacpath() containing retained routes.
<code>template</code>	Optional SpatRaster defining the density grid. When omitted, the function attempts to infer a template from <code>evac_result</code> .
<code>quantile_threshold</code>	Numeric value between 0 and 1. Used to select high-density route cells when <code>min_count</code> is NULL.
<code>min_count</code>	Optional numeric minimum density count. When supplied, this takes precedence over <code>quantile_threshold</code> .
<code>rescale</code>	Logical passed to leastcostpath::create_lcp_density() .
<code>return_polygons</code>	Logical. Convert high-density raster cells to polygons.
<code>...</code>	Reserved for future extensions.

Value

An `evac_bottleneck` list with density rasters, optional polygons, threshold value, and summary.

Examples

```

template <- terra::rast(nrows = 5, ncols = 5, xmin = 0, xmax = 5, ymin = 0, ymax = 5,
  vals = 1, crs = "EPSG:3857")
routes <- sf::st_sf(
  route = c("a", "b"),
  geometry = sf::st_sfc(
    sf::st_linestring(matrix(c(0.5, 0.5, 4.5, 4.5), ncol = 2, byrow = TRUE)),
    sf::st_linestring(matrix(c(0.5, 4.5, 4.5, 0.5), ncol = 2, byrow = TRUE)),
    crs = 3857
  )
)
map_evac_bottlenecks(routes = routes, template = template)

```

```
prepare_evac_inputs
```

Read and project the core evacuation inputs

Description

Read and project the core evacuation inputs

Usage

```

prepare_evac_inputs(
  hazard_zone,
  roads,
  dem,
  target_crs = NULL,
  hazard_as_polygon = TRUE,
  dissolve_hazard = TRUE,
  road_exclude = NULL
)

```

Arguments

<code>hazard_zone</code>	Hazard/inundation zone as a SpatRaster, SpatVector, or file path.
<code>roads</code>	Road/pathway network as a SpatVector or file path.
<code>dem</code>	Elevation raster as a SpatRaster or file path.
<code>target_crs</code>	Optional projected CRS in meters.
<code>hazard_as_polygon</code>	Logical. Convert raster hazard zones to polygons.
<code>dissolve_hazard</code>	Logical. Dissolve hazard polygon pieces.
<code>road_exclude</code>	Optional road exclusion list passed to <code>clean_roads()</code> .

Value

A named list with `hazard_zone`, `roads`, and `dem`.

Examples

```
dem <- terra::rast(nrows = 5, ncols = 5, xmin = 0, xmax = 5, ymin = 0, ymax = 5,
  vals = 1, crs = "EPSG:3857")
hazard <- terra::as.polygons(dem, dissolve = TRUE)
roads <- terra::vect(matrix(c(0, 2.5, 5, 2.5), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
inputs <- prepare_evac_inputs(hazard, roads, dem)
names(inputs)
```

prepare_hazard_zone	<i>Prepare a hazard zone from an inundation raster</i>
---------------------	--

Description

Converts an inundation raster to a binary hazard-zone raster or polygon using a threshold. This is intentionally general so users can adapt it to tsunamis, flood, storm-surge, or other hazard layers.

Usage

```
prepare_hazard_zone(
  inundation,
  threshold = 0,
  land_mask = NULL,
  target_crs = NULL,
  as_polygon = TRUE,
  dissolve = TRUE
)
```

Arguments

<code>inundation</code>	A <code>SpatRaster</code> or path to a raster.
<code>threshold</code>	Numeric threshold. Cells greater than threshold are treated as inside the hazard zone.
<code>land_mask</code>	Optional <code>SpatRaster</code> or <code>SpatVector</code> used to mask the hazard zone to land.
<code>target_crs</code>	Optional output CRS. Use a projected CRS in meters for later distance calculations.
<code>as_polygon</code>	Logical. If TRUE, return a polygon hazard zone.
<code>dissolve</code>	Logical. If TRUE, dissolve polygon pieces.

Value

A binary `SpatRaster` or polygon `SpatVector`.

Examples

```
r <- terra::rast(nrows = 5, ncols = 5, xmin = 0, xmax = 5, ymin = 0, ymax = 5)
terra::values(r) <- c(rep(0, 12), rep(1, 13))
zone <- prepare_hazard_zone(r, threshold = 0, as_polygon = TRUE)
zone
```

prepare_tsunami_zones *Prepare separate tsunami zones for escape analysis and visualization*

Description

Tsunami evacuation workflows often need two different zone objects. The land-only inundation zone is the area where road origins and output time surfaces should be mapped. The escape-boundary zone should combine the land-only inundation zone with water so that the coastline is not treated as an artificial escape boundary. This prevents false escape/safety points along the water-land edge when roads touch or approach the shoreline.

Usage

```
prepare_tsunami_zones(
  inundation,
  dem,
  target_crs = NULL,
  inundation_threshold = 0,
  land_threshold = 0,
  water_threshold = 0,
  dem_sign_multiplier = 1,
  resample_method = "bilinear",
  as_polygon = TRUE,
  dissolve = TRUE
)
```

Arguments

inundation	Inundation-depth raster or path to a raster. Cells greater than <code>inundation_threshold</code> are treated as inundated.
dem	Elevation/topobathymetry raster or path to a raster.
target_crs	Optional projected CRS in meters for returned objects.
inundation_threshold	Numeric threshold used to define inundated cells.
land_threshold	Numeric DEM threshold used to define land. Default is 0, so land is <code>dem > 0</code> after applying <code>dem_sign_multiplier</code> .
water_threshold	Numeric DEM threshold used to define water. Default is 0, so water is <code>dem < 0</code> after applying <code>dem_sign_multiplier</code> .

dem_sign_multiplier	Multiplier applied to the DEM before land/water classification. Use -1 when the DEM sign convention is reversed.
resample_method	Method passed to <code>terra::resample()</code> when aligning the inundation raster to the DEM. Use "bilinear" for depth rasters and "near" for categorical rasters.
as_polygon	Logical. If TRUE, return polygon zones in addition to rasters.
dissolve	Logical. Dissolve polygon pieces.

Value

A named list with land-only `hazard_zone`, water-combined `escape_zone`, and supporting rasters. `hazard_zone` should usually be used for origin generation, mapping, and output clipping. `escape_zone` should usually be passed to `run_evacpath(escape_zone = ...)` or `find_escape_points()`.

Examples

```
dem <- terra::rast(nrows = 6, ncols = 6, xmin = 0, xmax = 6, ymin = 0, ymax = 6,
  crs = "EPSG:3857")
xy <- terra::crds(dem, df = TRUE)
terra::values(dem) <- -1.5 + 0.7 * xy$x + 0.2 * sin(xy$y)
inundation <- dem
terra::values(inundation) <- ifelse(terra::values(dem) > 0 & terra::values(dem) < 2.5, 1, 0)
zones <- prepare_tsunami_zones(inundation, dem, as_polygon = TRUE)
names(zones)
```

read_spatial	<i>Read a spatial input</i>
--------------	-----------------------------

Description

Accepts an existing terra object or a file path and returns a `SpatRaster` or `SpatVector`. Raster-like extensions are read with `terra::rast()` and vector-like extensions are read with `terra::vect()`.

Usage

```
read_spatial(x)
```

Arguments

`x` A `SpatRaster`, `SpatVector`, or file path.

Value

A `SpatRaster` or `SpatVector`.

Examples

```
r <- terra::rast(nrows = 2, ncols = 2, vals = 1)
read_spatial(r)
```

run_evacpath

*Run the full evacuation-path modeling workflow***Description**

This high-level wrapper runs the core evacpath pipeline: read/project inputs, create an evacuation grid, identify escape/safety points, build a road mask, create a slope-based conductance surface, calculate minimum least-cost distance to safety, and create evacuation-time polygons.

Usage

```
run_evacpath(
  hazard_zone,
  roads,
  dem,
  target_crs = NULL,
  region_name = NULL,
  escape_zone = NULL,
  roads_for_escape = NULL,
  escape_roads_inset_x_m = 0,
  escape_roads_inset_y_m = 0,
  road_aware_escape_zone = FALSE,
  escape_zone_road_buffer_m = NULL,
  escape_zone_crop_buffer_m = NULL,
  study_area = NULL,
  road_exclude = NULL,
  grid_resolution = NULL,
  grid_resolution_factor = 5,
  road_buffer_m = 2,
  escape_buffer_m = 5,
  final_road_buffer_m = 3,
  region_buffer_m = 5000,
  dem_resolution = NULL,
  max_origins = NULL,
  max_destinations = NULL,
  seed = 23401,
  walking_speed_mps = 1.22,
  lcp_cost_function = "tobler",
  lcp_neighbours = 16,
  lcp_crit_slope = 12,
  lcp_max_slope = NULL,
  lcp_args = list(),
  clip_mode = c("hazard", "road_hazard", "hazard_plus_roads", "none"),
  progress = FALSE,
  progress_every = 1L,
  lcp_check_locations = FALSE,
  keep_routes = FALSE
```

)

Arguments

hazard_zone	Hazard/inundation zone as SpatRaster, SpatVector, or path. This should usually be the land-only area where evacuation origins and final output surfaces are mapped.
roads	Road/pathway network as SpatVector or path.
dem	Elevation raster as SpatRaster or path.
target_crs	Optional projected CRS in meters, for example "EPSG:32748".
region_name	Optional region name stored in output polygons.
escape_zone	Optional boundary zone used only to identify escape/safety points. For tsunami workflows, pass the land-inundation-plus-water zone from prepare_tsunami_zones() here. If NULL, hazard_zone is used.
roads_for_escape	Optional road/pathway layer used only for escape-point detection. If NULL, roads is used. This is useful when the full road dataset extends beyond the reliable hazard-zone study extent.
escape_roads_inset_x_m	Optional x-direction inset applied to roads_for_escape before escape-point detection. This prevents roads from intersecting artificial study-area extent boundaries.
escape_roads_inset_y_m	Optional y-direction inset applied to roads_for_escape before escape-point detection.
road_aware_escape_zone	Logical. If TRUE, buffered roads_for_escape are combined with escape_zone before escape points are generated. This preserves bridge, causeway, and walkway corridors over water that can be lost when the tsunami layer is split into land and water masks.
escape_zone_road_buffer_m	Road buffer used when road_aware_escape_zone = TRUE.
escape_zone_crop_buffer_m	Additional buffer used when road_aware_escape_zone = TRUE.
study_area	Optional local study area for limiting escape-point search.
road_exclude	Optional list passed to clean_roads().
grid_resolution	Evacuation-grid resolution. If NULL, calculated from terra::res(dem) * grid_resolution_factor.
grid_resolution_factor	Multiplier applied to DEM resolution when grid_resolution = NULL.
road_buffer_m	Road buffer distance.
escape_buffer_m	Escape-point buffer distance.
final_road_buffer_m	Output clipping buffer around roads.

region_buffer_m	Buffer around study_area used when finding escape points.
dem_resolution	Optional DEM resolution used before conductance creation.
max_origins	Optional maximum number of road origin points.
max_destinations	Optional maximum number of escape/safety destination points. This is useful for quick tests in regions where roads intersect the hazard boundary many times.
seed	Random seed used when max_origins or max_destinations is supplied.
walking_speed_mps	Walking speed in meters per second. This controls conversion of modeled route distances into evacuation times. It does not change route geometry.
lcp_cost_function	Character string or function passed to <code>leastcostpath::create_slope_cs()</code> . This changes conductance and may change route geometry.
lcp_neighbours	Neighbourhood passed to <code>leastcostpath::create_slope_cs()</code> . Use 4, 8, 16, 32, 48, or a custom matrix accepted by leastcostpath.
lcp_crit_slope	Numeric critical slope passed to <code>leastcostpath::create_slope_cs()</code> .
lcp_max_slope	Optional numeric maximum traversable slope passed to <code>leastcostpath::create_slope_cs()</code> .
lcp_args	Named list of additional arguments passed to <code>leastcostpath::create_slope_cs()</code> , such as exaggeration. Explicit <code>lcp_*</code> arguments take precedence over duplicate entries in this list.
clip_mode	Output clipping mode. The default "hazard" creates a continuous time-grid polygon surface clipped to the land-only hazard zone. Use "road_hazard" for the older road-buffer-limited output, "hazard_plus_roads" to retain both, or "none" for unclipped Voronoi polygons.
progress	Logical. Print progress while running least-cost paths.
progress_every	Integer. Print progress every n origins when progress = TRUE.
lcp_check_locations	Logical passed to <code>leastcostpath::create_lcp()</code> . Default is FALSE for speed after projection/cropping checks.
keep_routes	Logical. If TRUE, retain the selected least-cost route for each reachable origin in <code>result\$routes</code> . The default is FALSE to avoid the additional memory cost.

Details

For tsunami applications, `hazard_zone` and `escape_zone` should often be different. Use a land-only `hazard_zone` for origins and output mapping, but use a water-combined `escape_zone` for escape-point detection so the coastline is not treated as an artificial safety boundary. The helper `prepare_tsunami_zones()` creates both objects.

Value

An `evacpath_result` list containing spatial outputs and parameters.

Examples

```
dem <- terra::rast(nrows = 7, ncols = 7, xmin = 0, xmax = 7, ymin = 0, ymax = 7,
  vals = 1, crs = "EPSG:3857")
hazard_raster <- terra::crop(dem, terra::ext(1, 6, 1, 6))
hazard <- terra::as.polygons(hazard_raster, dissolve = TRUE)
roads <- terra::vect(matrix(c(0, 3.5, 7, 3.5), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
result <- run_evacpath(
  hazard_zone = hazard,
  roads = roads,
  dem = dem,
  grid_resolution = 1,
  road_buffer_m = 0.2,
  escape_buffer_m = 0.3,
  final_road_buffer_m = 0.2,
  max_origins = 2,
  max_destinations = 2,
  seed = 1
)
result
```

validate_evac_routes	<i>Validate modeled evacuation routes</i>
----------------------	---

Description

Compares modeled routes with official, observed, collaborator-provided, or manually digitized reference routes. Buffer overlap and Hausdorff-distance comparisons are calculated with `sf`. Path deviation index comparisons use `leastcostpath::PDI_validation()` when available and compatible with the supplied routes.

Usage

```
validate_evac_routes(
  modeled_routes,
  reference_routes,
  method = c("pdi", "buffer_overlap", "hausdorff", "all"),
  buffer_m = 25,
  by = NULL,
  ...
)
```

Arguments

<code>modeled_routes</code>	An <code>sf</code> <code>LINESTRING</code> or <code>MULTILINESTRING</code> object.
<code>reference_routes</code>	An <code>sf</code> <code>LINESTRING</code> or <code>MULTILINESTRING</code> object.
<code>method</code>	Comparison method: "pdi", "buffer_overlap", "hausdorff", or "all".

buffer_m	Buffer distance in projected coordinate reference system units, typically meters.
by	Optional column name used to match modeled and reference routes. When NULL, each modeled route is matched to its nearest reference route.
...	Reserved for future extensions.

Value

An `evac_route_validation` list with route-level metrics, `matched_routes`, requested method, `buffer_m`, and summary.

Examples

```
modeled <- sf::st_sf(
  id = "route_a",
  geometry = sf::st_sfc(
    sf::st_linestring(matrix(c(0, 0, 100, 0), ncol = 2, byrow = TRUE)),
    crs = 3857
  )
)
reference <- sf::st_sf(
  id = "route_a",
  geometry = sf::st_sfc(
    sf::st_linestring(matrix(c(0, 5, 100, 5), ncol = 2, byrow = TRUE)),
    crs = 3857
  )
)
validate_evac_routes(modeled, reference, method = "buffer_overlap", buffer_m = 10)
```

write_evac_outputs	<i>Write evacpath outputs to disk</i>
--------------------	---------------------------------------

Description

Writes the main spatial outputs in an `evacpath_result` list to GeoPackage and GeoTIFF files. Non-spatial objects, including the conductance surface, are not written.

Usage

```
write_evac_outputs(
  result,
  output_dir,
  prefix = "evacpath",
  overwrite = TRUE,
  include_inputs = FALSE
)
```

Arguments

result	An object returned by run_evacpath().
output_dir	Output directory.
prefix	Filename prefix.
overwrite	Logical. Overwrite existing files.
include_inputs	Logical. Also write projected input layers.

Value

A named character vector of written file paths.

Examples

```
dem <- terra::rast(nrows = 7, ncols = 7, xmin = 0, xmax = 7, ymin = 0, ymax = 7,
  vals = 1, crs = "EPSG:3857")
hazard_raster <- terra::crop(dem, terra::ext(1, 6, 1, 6))
hazard <- terra::as.polygons(hazard_raster, dissolve = TRUE)
roads <- terra::vect(matrix(c(0, 3.5, 7, 3.5), ncol = 2, byrow = TRUE),
  type = "lines", crs = "EPSG:3857")
result <- run_evacpath(
  hazard_zone = hazard,
  roads = roads,
  dem = dem,
  grid_resolution = 1,
  road_buffer_m = 0.2,
  escape_buffer_m = 0.3,
  final_road_buffer_m = 0.2,
  max_origins = 2,
  max_destinations = 2,
  seed = 1
)
out_dir <- file.path(tempdir(), "evacpath-example")
paths <- write_evac_outputs(result, output_dir = out_dir, prefix = "example")
unlink(out_dir, recursive = TRUE)
```

Index

- * **package**
 - evacpath-package, [3](#)
- calc_evac_time, [3](#)
- calc_min_distance_to_safety, [4](#)
- calculate_lc_path, [5](#)
- calculate_min_dist, [6](#)
- clean_roads, [6](#)
- compare_evac_scenarios, [7](#)
- crop_roads_to_inner_extent, [8](#)
- diagnose_evac_model, [9](#)
- diagnose_evac_model(), [11](#)
- evacpath (evacpath-package), [3](#)
- evacpath-package, [3](#)
- find_escape_points, [10](#)
- has_errors, [11](#)
- interpolate_distance_surface, [12](#)
- leastcostpath::create_lcp(), [5](#)
- leastcostpath::create_lcp_density(),
[21](#)
- leastcostpath::create_slope_cs(), [13](#),
[28](#)
- leastcostpath::PDI_validation(), [29](#)
- make_conductance_surface, [13](#)
- make_evac_grid, [14](#)
- make_evac_polygons, [15](#)
- make_output_clip_area, [16](#)
- make_region_area, [16](#)
- make_road_aware_escape_zone, [17](#)
- make_road_mask, [18](#)
- make_road_origins, [19](#)
- make_roads_in_zone, [20](#)
- map_evac_bottlenecks, [21](#)
- prepare_evac_inputs, [22](#)
- prepare_hazard_zone, [23](#)
- prepare_tsunami_zones, [24](#)
- read_spatial, [25](#)
- run_evacpath, [26](#)
- run_evacpath(), [7](#), [21](#)
- terra::rast(), [25](#)
- terra::vect(), [25](#)
- validate_evac_routes, [29](#)
- write_evac_outputs, [30](#)