Package: sealeveltools (via r-universe)

July 10, 2026

Type Package

Title Sea Level Adjustment and Coastal Terrain Modeling

Version 0.3.0

Date 2026-07-03

Imports grDevices, terra

Maintainer Jovan Kovačević <jkovacevic@grf.bg.ac.rs>

Description Simulates sea-level rise and fall from raster-based elevation models through functions for relative vertical datum transformation of Digital Terrain Models (DTMs) and integration with bathymetric data to produce continuous terrestrial–marine Digital Elevation Models (DEMs). Supports coastal exposure modeling, paleogeographic reconstruction, submerged landscape analysis, and climate change impact assessment. Optional gap-filling and threshold-based terrain filtering facilitate reconstruction of incomplete elevation surfaces and scenario-based landscape simulation. Applications span coastal engineering, geomorphology, archaeology, environmental modeling, and geospatial analysis.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Jovan Kovačević [aut, cre] (ORCID: <https://orcid.org/0000-0001-9980-5797>), Christopher Nuttall [aut] (ORCID: <https://orcid.org/0000-0003-2679-9677>)

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

Date/Publication 2026-07-10 20:50:12 UTC

RemoteUrl https://github.com/cran/sealeveltools

RemoteRef HEAD

RemoteSha 91eab21d20475f0d8cfa530b22d35d361183b533

Contents

`adjustSeaLevel` *Adjust terrain elevations relative to sea-level change*

Description

'adjustSeaLevel()' recalculates terrain elevations relative to a modified sea-level datum. The function simulates sea-level rise or fall by shifting elevation values accordingly.

If bathymetric data is provided, submerged terrain is merged with the DTM to produce a continuous DEM. This is particularly useful for: coastal engineering studies, paleogeographic reconstructions, archaeological landscape exposure modelling and climate change impact assessments.

Optional gap-filling enables reconstruction of missing terrain areas through iterative focal interpolation.

Usage

```
adjustSeaLevel(
  dtm,
  seaLevelChange,
  bathymetry = NULL,
  studyArea = NULL,
  fillGaps = FALSE,
  removeUnder = NULL,
  plot = FALSE,
  filename = "",
  overwrite = FALSE,
  wopt = list()
)
```

Arguments

`dtm`	'SpatRaster': Digital Terrain Model.
`seaLevelChange`	'numeric': Sea level change value. Positive = sea level rise. Negative = sea level drop.
`bathymetry`	'SpatRaster' (optional): Bathymetric raster to merge with DTM.
`studyArea`	'SpatVector' (optional): Study area polygon to which to crop results.
`fillGaps`	'logical' (default 'FALSE'): Whether to iteratively fill NA gaps.
`removeUnder`	'numeric' (optional): If specified, removes all elevations under defined threshold.
`plot`	'logical' (default 'FALSE'): Whether to visualize results

filename	'character' (optional): Output file name, skipped if left empty
overwrite	'logical' (default 'FALSE'): If 'TRUE', 'filename' is overwritten
wopt	'list()': list with named options for writing files as in 'writeRaster'

Value

'SpatRaster': DEM adjusted for sea level change.

Examples

```
library(terra)

### Create dummy DTM and Bathymetric data ###
nr <- 300
nc <- 300
lon <- seq(-10, 10, length.out = nc)
lat <- seq(-10, 10, length.out = nr)

data_mat <- matrix(NA, nrow = nr, ncol = nc)
for (i in 1:nr) {
  for (j in 1:nc) {
    terrain <- 500 * exp(-(lon[j]^2 + lat[i]^2) / 10)
   data_mat[i, j] <- -1000 + 1000 * exp(-(lon[j]^2 + lat[i]^2) / 50) + terrain + rnorm(1, 0, 5)
  }
}

### Create SpatRast data ###
rast_demo <- rast(ncols = 300, nrows = 300, xmin = 0, xmax = 300, ymin = 0, ymax = 300)
values(rast_demo) <- data_mat

dtm_demo <- rast_demo
dtm_demo[dtm_demo[] <= 0] <- NA
# plot(dtm_demo)

bathy_demo <- rast_demo
bathy_demo[bathy_demo[] > 0] <- NA
# plot(bathy_demo)

### Rise sea level ###
dtm_rise <- adjustSeaLevel(dtm = dtm_demo, seaLevelChange = 100, plot = TRUE)

### Lower sea level with bathymetric data ###
dtm_lower <- adjustSeaLevel(
  dtm = dtm_demo, seaLevelChange = -100,
  bathy = bathy_demo, plot = TRUE
)

### Lower sea level with bathymetric data and remove areas under -500 ###
dtm_lower <- adjustSeaLevel(
  dtm = dtm_demo, seaLevelChange = -100,
  bathy = bathy_demo, removeUnder = -500, plot = TRUE
)
```

```
### Lower sea level with bathymetric data and gap-filling ###
dtm_demo[dtm_demo[] <= 3] <- NA # introduce missing data
dtm_lower_gaps <- adjustSeaLevel(
  dtm = dtm_demo, seaLevelChange = -50,
  bathy = bathy_demo, plot = TRUE
)
dtm_lower_filled <- adjustSeaLevel(
  dtm = dtm_demo, seaLevelChange = -50,
  bathy = bathy_demo, fillGaps = TRUE, plot = TRUE
)
```

Index