Package ‘ForestTools’

April 28, 2024

Type   Package
Title  Tools for Analyzing Remote Sensing Forest Data
Version 1.0.2
Date   2024-04-27
Description Tools for analyzing remote sensing forest data, including functions for detecting tree-
tops from canopy models, outlining tree crowns, and calculating textural metrics.
Depends R (>= 4.2)
License GPL-3
Encoding UTF-8
LazyData true
Imports terra, sf, Matrix, imager, GLCMTextures
Suggests testthat (>= 3.0.0), knitr, rmarkdown
RoxygenNote 7.3.1
URL https://github.com/andrew-plowright/ForestTools
BugReports https://github.com/andrew-plowright/ForestTools/issues
Config/testthat/edition 3
NeedsCompilation no
Author Andrew Plowright [aut, cre],
   Jean-Romain Roussel [ctb] (Contributed to segment-based GLCM
   segmentation)
Maintainer Andrew Plowright <andrew.plowright@alumni.ubc.ca>
Repository CRAN
Date/Publication 2024-04-28 04:20:03 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>glcm</td>
<td>2</td>
</tr>
<tr>
<td>kootenayBlocks</td>
<td>3</td>
</tr>
<tr>
<td>kootenayCHM</td>
<td>4</td>
</tr>
</tbody>
</table>
**glcm**

**Description**

Generate textural metrics using Grey-Level Co-Occurrence Matrices (GLCM). Can be applied to an entire or image or, if a coterminous raster of segments is provided, GLCM can be calculated for each segment.

**Usage**

```r
glcm(image, segs = NULL, n_grey = 32, angle = c(0, 1), discretize_range = NULL)
```

**Arguments**

- `image`: SpatRaster. A single-band raster layer from which texture is measured
- `segs`: SpatRaster. A segmented raster. Cell values should be equal to segment numbers. If `segs` are not provided, GLCM will be calculated for the entire image.
- `n_grey`: integer. Number of grey levels into which the image will be discretized
- `angle`: integer. Angle at which GLCM will be calculated. Ex.: ‘c(0,1)’
- `discretize_range`: numeric. Vector of two values indicating the minimum and maximum input values for discretizing the image. This can be useful when processing tiles of a larger image, for which you may want to impose a consistent value range.

**Value**

data.frame

**References**

kootenayBlocks

See Also

mcws

Examples

```r
## Not run:
library(terra)
library(ForestTools)

chm <- rast(kootenayCHM)
image <- rast(kootenayOrtho)[[1]]

# Generate raster segments
segs <- mcws(kootenayTrees, chm, minHeight = 0.2, format = "raster")

# Get textural metrics for ortho's red band
tex <- glcm(image, segs)

## End(Not run)
```

kootenayBlocks  

**Kootenay forest - Cut blocks**

Description

Boundaries of cut blocks within a 1.5 hectare section of forest in the Kootenay mountains, in British Columbia, Canada. Each block contains trees of different levels of maturity. Overlaps with kootenayTrees, kootenayCrowns, kootenayOrtho and kootenayCHM.

Usage

kootenayBlocks

Format

Simple polygon feature collection with the following attributes:

- **BlockID**  numerical identifier for each block
- **Shape_Leng**  length of polygon on meters
- **Shape_Area**  area of polygon in square meters

See Also

kootenayTrees kootenayCHM kootenayCrowns kootenayOrtho
**kootenayCHM**  
*Kootenay forest - Canopy height model*

**Description**

A canopy height model of a 1.5 hectare section of forest in the Kootenay mountains, in British Columbia, Canada.

**Usage**

kootenayCHM

**Format**

PackedSpatRaster object  
Cell values are equal to canopy height above ground (in meters)

**Source**

Data acquired from a photogrammetric drone survey performed by Spire Aerobotics on June 16th, 2016.

**See Also**

kootenayTrees kootenayBlocks kootenayCrowns kootenayOrtho

---

**kootenayCrowns**  
*Kootenay forest - Tree crowns*

**Description**

Outlines of tree crowns corresponding to the kootenayTrees treetops. Generated using mcws.

**Usage**

kootenayCrowns

**Format**

Simple polygon feature collection with the following attributes:

- **height** height of the tree’s apex, in meters above ground. Inherited from kootenayTrees.
- **winRadius** radius of the moving window at the treetop’s location. Inherited from kootenayTrees.
- **crownArea** area of crown outline in square meters

**See Also**

kootenayTrees kootenayCHM kootenayBlocks kootenayOrtho
**kootenayOrtho**

**Kootenay forest - Orthomosaic**

**Description**

An orthomosaic of a 1.5 hectare section of forest in the Kootenay mountains, in British Columbia, Canada.

**Usage**

kootenayOrtho

**Format**

PackedSpatRaster object

Cell values are equal to canopy height above ground (in meters)

**Source**

Data acquired from a photogrammetric drone survey performed by Spire Aerobotics on June 16th, 2016.

**See Also**

kootenayTrees kootenayBlocks kootenayCrowns kootenayCHM

---

**kootenayTrees**

**Kootenay forest - Dominant trees over 2 m**

**Description**

Dominant trees from a 1.5 hectare section of forest in the Kootenay mountains, in British Columbia, Canada. Trees were detected by applying the \texttt{vwf} function to the \texttt{kootenayCHM} raster dataset. Only trees over 2 m above ground were detected.

**Usage**

kootenayTrees

**Format**

Simple point feature collection with the following attributes:

- **height**: height of the tree’s apex, in meters above ground
- **winRadius**: radius of the moving window (see \texttt{vwf}) at the treetop’s location
See Also

kootenayCHM kootenayBlocks kootenayCrowns kootenayOrtho

---

**mcws**

**Marker-Controlled Watershed Segmentation**

**Description**

This function implements the `watershed` function to segment (i.e.: outline) crowns from a CHM (canopy height model). Segmentation is guided by the point locations of treetops, typically detected using the `vwf` function. See Meyer & Beucher (1990) for details on watershed segmentation.

**Usage**

```r
mcws(
  treetops,
  CHM,
  minHeight = 0,
  format = "raster",
  OSGeoPath = NULL,
  IDfield = "treeID"
)
```

**Arguments**

- `treetops` sf. The point locations of treetops in sf format.
- `CHM` SpatRaster. Canopy height model in SpatRaster format. This should be the same CHM that was used to the detect the treetops.
- `minHeight` numeric. The minimum height value for a CHM pixel to be considered as part of a crown segment. All CHM pixels beneath this value will be masked out. Note that this value should be lower than the minimum height of treetops.
- `format` string. Format of the function’s output. Can be set to either ‘raster’ or ‘polygons’.
- `OSGeoPath` character. Obsolete. Will be removed next version
- `IDfield` character. Name of the field for storing the unique tree identifier

**Details**

Crown segments are returned as either a SpatRaster or a sf (Simple Feature) class object, as defined using the `format` argument. For many analytic purposes, it is preferable to have crown outlines as polygons. However, polygonal crown maps take up significantly more disk space, and take longer to process. It is advisable to run this function using a raster output first to review results and adjust parameters.

NOTE: when setting `format` to ‘polygons’, orphaned segments (i.e.: outlines without an associated treetop) will be removed. This will NOT occur using ‘raster’ format. This issue will be resolved eventually but requires the watershed function to be rewritten.
Value

Depending on the setting for `format`, this function will return a map of outlined crowns as either a SpatRaster class object, in which distinct crowns are given a unique cell value, or a sf class object, in which each crown is represented by a polygon.

References


See Also

vwf

Examples

```r
## Not run:
library(terra)
library(ForestTools)

chm <- rast(kootenayCHM)
# Use variable window filter to detect treetops
ttops <- vwf(chm, winFun = function(x){x * 0.06 + 0.5}, minHeight = 2)

# Segment tree crowns
segs <- mcws(ttops, chm, minHeight = 1)
## End(Not run)
```

qualselBlocks  

**Description**

Boundaries of cut blocks within a 125 hectare section of forest in the Quesnel Timber Supply Area, in British Columbia, Canada. Each block contains trees of different levels of maturity. Overlaps with `quesnelTrees` and `quesnelCHM`.

**Usage**

qualselBlocks
**Format**

Simple polygon feature collection with the following attributes:

- **BlockID** numerical identifier for each block
- **Shape_Leng** length of polygon on meters
- **Shape_Area** area of polygon in square meters

**See Also**

quesnelTrees quesnelCHM

---

**Description**

A canopy height model of a 125 hectare section of forest in the Quesnel Timber Supply Area, in British Columbia, Canada.

**Usage**

quesnelCHM

**Format**

PackedSpatRaster object

Cell values are equal to canopy height above ground (in meters)

**Source**

Data acquired from a photogrammetric drone survey performed by Spire Aerobotics on September 15th, 2016.

**See Also**

quesnelTrees quesnelBlocks
**Description**

Dominant trees from a 125 hectare section of forest in the Quesnel Timber Supply Area, in British Columbia, Canada. Trees were detected by applying the `vwf` function to the `quesnelCHM` raster dataset. Only trees over 2 m above ground were detected.

**Usage**

`quesnelTrees`

**Format**

Simple point feature collection with the following attributes:

- **height**  height of the tree’s apex, in meters above ground
- **winRadius**  radius of the moving window (see `vwf`) at the treetop’s location

**See Also**

`quesnelCHM` `quesnelBlocks`

---

**vwf**  
*Variable Window Filter*

**Description**

Implements the variable window filter algorithm (Popescu & Wynne, 2004) for detecting treetops from a canopy height model.

**Usage**

```r
vwf(
    CHM,
    winFun,
    minHeight = NULL,
    warnings = TRUE,
    minWinNeib = "queen",
    IDfield = "treeID"
)
```
Arguments

**CHM**
SpatRaster. Canopy height model in SpatRaster format.

**winFun**
function. The function that determines the size of the window at any given location on the canopy. It should take the value of a given CHM pixel as its only argument, and return the desired *radius* of the circular search window when centered on that pixel. Size of the window is in map units.

**minHeight**
numeric. The minimum height value for a CHM pixel to be considered as a potential treetop. All CHM pixels beneath this value will be masked out.

**warnings**
logical. If set to FALSE, this function will not emit warnings related to inputs.

**minWinNeib**
character. Define whether the smallest possible search window (3x3) should use a queen or a rook neighborhood.

**IDfield**
character. Name of field for unique tree identifier

Details

This function uses the resolution of the raster to figure out how many cells the window needs to cover. This means that the raster value (representing height above ground) and the map unit (represented by the raster’s resolution), need to be in the _same unit_. This can cause issues if the raster is in lat/lon, whereby its resolution is in decimal degrees.

Value

Simple feature collection of POINT type. The point locations of detected treetops. The object contains two fields in its data table: `height` is the height of the tree, as extracted from the CHM, and `winRadius` is the radius of the search window when the treetop was detected. Note that `winRadius` does not necessarily correspond to the radius of the tree’s crown.

References


See Also

mcws

Examples

```
## Not run:
library(terra)
library(ForestTools)

chm <- rast(kootenayCHM)

# Set function for determining variable window radius
winFunction <- function(x){x * 0.06 + 0.5}

# Set minimum tree height (treetops below this height will not be detected)
```

minHgt <- 2

# Detect treetops in demo canopy height model
ttops <- vwf(chm, winFunction, minHgt)

## End(Not run)
Index

* datasets
  kootenayBlocks, 3
  kootenayCHM, 4
  kootenayCrowns, 4
  kootenayOrtho, 5
  kootenayTrees, 5
  quesnelBlocks, 7
  quesnelCHM, 8
  quesnelTrees, 9

  glcm, 2
  kootenayBlocks, 3, 4–6
  kootenayCHM, 3, 4, 4, 5, 6
  kootenayCrowns, 3, 4, 4, 5, 6
  kootenayOrtho, 3, 4, 5, 6
  kootenayTrees, 3–5, 5

  mcws, 3, 4, 6, 10
  quesnelBlocks, 7, 8, 9
  quesnelCHM, 7, 8, 8, 9
  quesnelTrees, 7, 8, 9

  vwf, 5–7, 9, 9

  watershed, 6