

# Package ‘LadderFuelsR’

November 2, 2024

**Type** Package

**Title** Automated Tool for Vertical Fuel Continuity Analysis using  
Airborne Laser Scanning Data

**Version** 0.0.7

**Description** Set of tools for analyzing vertical fuel continuity at the tree level using Airborne Laser Scanning data. The workflow consisted of: 1) calculating the vertical height profiles of each segmented tree; 2) identifying gaps and fuel layers; 3) estimating the distance between fuel layers; and 4) retrieving the fuel layers base height and depth. Additionally, other functions recalculate previous metrics after considering distances greater than certain threshold. Moreover, the package calculates: i) the percentage of Leaf Area Density comprised in each fuel layer, ii) remove fuel layers with Leaf Area Density (LAD) percentage less than 10, and iii) recalculate the distances among the remainder ones. On the other hand, it identifies the crown base height (CBH) based on different criteria: the fuel layer with the highest LAD percentage and the fuel layers located at the largest and at the last-distance. When there is only one fuel layer, it also identifies the CBH performing a segmented linear regression (breaking points) on the cumulative sum of LAD as a function of height. Finally, a collection of plotting functions is developed to represent: i) the initial gaps and fuel layers; ii) the fuels base height, depths and gaps with distances greater than certain threshold and, iii) the CBH based on different criteria. The methods implemented in this package are original and have not been published elsewhere.

**URL** <https://github.com/olgaviedma/LadderFuelsR>

**License** GPL-3

**Encoding** UTF-8

**Imports** dplyr, gdata, ggplot2, magrittr, segmented, stringr, tidyr,  
tibble, tidyselect

**RoxygenNote** 7.3.2

**Collate** 'calculate\_gaps\_perc.R' 'gap\_fbh.R' 'distances\_calculation.R'  
'depths\_calculation.R' 'corrected\_base\_heights.R'  
'corrected\_depth.R' 'corrected\_distances.R'  
'maxlad\_metrics\_25perc.R' 'cbh\_metrics.R'  
'cummlad\_breaks\_metrics.R' 'plot\_cbh\_bpoint.R'  
'plot\_gaps\_fbhs.R' 'plot\_cbh\_maxdist.R' 'plot\_cbh\_lastdist.R'  
'plot\_cbh\_maxlad.R' 'plot\_maxlad\_metrics.R' 'renamed0\_df.R'  
'renamed\_df.R' 'foofy.R'

**BugReports** <https://github.com/olgaviedma/LadderFuelsR/issues>

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Olga Viedma [aut, cph, cre],  
 Carlos Alberto Silva [aut, cph],  
 Jose Manuel Moreno [aut, cph],  
 Andrew T. Hudak [aut, cph]

**Maintainer** Olga Viedma <olga.viedma@uclm.es>

**Repository** CRAN

**Date/Publication** 2024-11-02 09:20:02 UTC

## Contents

calculate_gaps_perc . . . . .	2
get_cbh_metrics . . . . .	3
get_cum_break . . . . .	6
get_depths . . . . .	8
get_distance . . . . .	10
get_effective_gap . . . . .	11
get_gaps_fbhs . . . . .	13
get_layers_lad . . . . .	15
get_plots_cbh_bp . . . . .	18
get_plots_cbh_LAD . . . . .	19
get_plots_cbh_lastdist . . . . .	20
get_plots_cbh_maxdist . . . . .	21
get_plots_effective . . . . .	22
get_plots_gap_fbh . . . . .	23
get_real_depths . . . . .	25
get_real_fbh . . . . .	27
get_renamed0_df . . . . .	28
get_renamed_df . . . . .	29
<b>Index</b>	<b>30</b>

---

calculate\_gaps\_perc    *Compute the percentile value of each height*

---

## Description

This function calculates the percentile value of each height

## Usage

```
calculate_gaps_perc (LAD_profiles, min_height=1.5)
```

**Arguments**

LAD\_profiles      original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the *leafR* package. An object of the class text)

min\_height        Numeric value for the actual minimum base height (in meters).

**Value**

A data frame giving the percentile value of each height.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**Examples**

```
library(magrittr)
library(dplyr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
LAD_profiles$treeID <- factor(LAD_profiles$treeID)

trees_name1 <- as.character(LAD_profiles$treeID)
trees_name2 <- factor(unique(trees_name1))

percentile_list1<-list()

for (i in levels(trees_name2)) {
  tree1 <- LAD_profiles |> dplyr::filter(treeID == i)
  percentiles <- calculate_gaps_perc (tree1,min_height=1.5)
  percentile_list1[[i]] <- percentiles
}
gaps_perc <- dplyr::bind_rows(percentile_list1)
gaps_perc$treeID <- factor(gaps_perc$treeID)
```

---

get\_cbh\_metrics      *Methods to estimated the Crown Base Height of a tree: maximum LAD percentage, maximum distance and the last distance*

---

**Description**

This function determines the CBH of a segmented tree using three criteria: maximum LAD percentage, maximum distance and the last distance.

**Usage**

```
get_cbh_metrics(effective_LAD, min_height= 1.5, hdepth1_height = 2.5, verbose=TRUE)
```

**Arguments**

<code>effective_LAD</code>	Tree metrics with gaps (distances), fuel base heights, and depths of fuel layers with LAD percentage greater than a threshold (10 (output of <code>[get_layers_lad()]</code> function)). An object of the class <code>text</code> .
<code>min_height</code>	Numeric value for the actual minimum base height (in meters).
<code>hdepth1_height</code>	Numeric value for the depth height of the first fuel layer. If the first fuel layer has the maximum LAD and its depth is greater than the indicated value, then this fuel layer is considered as the CBH of the tree. On the contrary, if its depth is $\leq$ the value, the CBH with maximum LAD will be the second fuel layer, although it has not the maximum LAD.
<code>verbose</code>	Logical, indicating whether to display informational messages (default is <code>TRUE</code> ).

**Details**

List of tree metrics:

- `treeID`: tree ID with strings and numeric values
- `treeID1`: tree ID with only numeric values
- `dptf`: Depth of fuel layers (m) after considering distances greater than the actual height bin step
- `effdist`: Effective distance between consecutive fuel layers (m) after considering distances greater than any number of steps
- `Hcbh`: Base height of each fuel separated by a distance greater than the certain number of steps
- `Hdptf`: Height of the depth of fuel layers (m) after considering distances greater than the actual step
- `Hdist`: Height of the distance ( $>$  any number of steps) between consecutive fuel layers (m)
- `Hcbh_Hdptf` - Percentage of LAD values comprised in each effective fuel layer
- `maxlad_Hcbh` - Height of the CBH of the segmented tree based on the maximum LAD percentage
- `maxlad1_Hcbh` - Height of the CBH from the second fuel layer when the maximum LAD occurred in the first fuel layer but its depth  $\leq$  "hdepth1\_height"
- `max_Hcbh` - Height of the CBH of the segmented tree based on the maximum distance found in its profile
- `last_Hcbh` - Height of the CBH of the segmented tree based on the last distance found in its profile
- `maxlad_` - Values of distance and fuel depth and their corresponding heights at the maximum LAD percentage
- `maxlad1_` - Values of distance and fuel depth and their corresponding heights for the second fuel layer when the maximum LAD occurred in the first fuel layer but its depth  $\leq$  "hdepth1\_height"
- `max_` - Values of distance and fuel depth and their corresponding heights at the maximum distance

- last\_ - Values of distance and fuel depth and their corresponding heights at the last distance
- nlayers - Number of effective fuel layers
- max\_height - Maximum height of the tree profile

### Value

A data frame giving the Crown Base Height (CBH) of a tree using three criteria: maximum LAD percentage, maximum distance and the last distance. For the case of maximum LAD CBH, the output gives the actual CBH with maximum LAD and also, the CBH from the second fuel layer when the first fuel layer has the maximum LAD but its depth is lesser than the value indicated in the parameter "hdepth1\_height".

### Author(s)

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

### See Also

[get\\_layers\\_lad](#)

### Examples

```
library(magrittr)
library(stringr)
library(dplyr)

# Before running this example, make sure to run get_real_depths().
if (interactive()) {
  effective_LAD <- get_layers_lad()
  LadderFuelsR::effective_LAD$treeID <- factor(LadderFuelsR::effective_LAD$treeID)

  trees_name1 <- as.character(effective_LAD$treeID)
  trees_name2 <- factor(unique(trees_name1))

  cbh_dist_list <- list()

  for (i in levels(trees_name2)) {
    tree1 <- effective_LAD |> dplyr::filter(treeID == i)
    cbh_dist_metrics <- get_cbh_metrics(tree1, min_height= 1.5, hdepth1_height = 2.5, verbose=TRUE)
    cbh_dist_list[[i]] <- cbh_dist_metrics
  }

  # Combine the individual data frames
  cbh_metrics <- dplyr::bind_rows(cbh_dist_list)

  # Get original column names
  original_column_names <- colnames(cbh_metrics)

  # Specify prefixes
  desired_order <- c("treeID", "Hcbh", "dptf", "effdist", "dist", "Hdist", "Hdptf", "max_", "last_",
    "maxlad_", "maxlad1_", "nlayers")
```

```

# Identify unique prefixes
prefixes <- unique(sub("^[a-zA-Z]+).*", "\\1", original_column_names))
# Initialize vector to store new order
new_order <- c()

# Loop over desired order of prefixes
for (prefix in desired_order) {
  # Find column names matching the current prefix
  matching_columns <- grep(paste0("^", prefix), original_column_names, value = TRUE)
  # Append to the new order
  new_order <- c(new_order, matching_columns)
}
# Reorder values
cbh_metrics <- cbh_metrics[, new_order]
}

```

---

get\_cum\_break

*CBH estimation using the breaking point method and the LAD percentage below and above the CBH*


---

## Description

This function calculates the crown base height (CBH) of the vertical tree profile (VTP) using a segmented regression model fitted to the cumulative LAD values as a function of height. The function also calculates the percentage of LAD values below and above the identified CBH or breaking point.

## Usage

```
get_cum_break(LAD_profiles, cbh_metrics, threshold=75, min_height= 1.5, verbose=TRUE)
```

## Arguments

LAD_profiles	Original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the <i>leafR</i> package). An object of the class data frame.
cbh_metrics	CBH metrics based on three criteria: maximum LAD percentage, maximum distance and last distance (output of [get_cbh_metrics()] function). An object of the class data frame.
threshold	Numeric value of the LAD percentage below or above the breaking point to set the CBH (default 75).
min_height	Numeric value for the actual minimum base height (in meters).
verbose	Logical, indicating whether to display informational messages (default is TRUE).

## Details

# List of tree metrics:

- treeID: tree ID with strings and numeric values
- treeID1: tree ID with only numeric values
- Hcbh\_brpt: Height of the CBH based on the breaking point method (m)
- below\_hcbhbp: Percentage of LAD values below the CBH or breaking point
- above\_hcbhbp: Percentage of LAD values above the CBH or breaking point
- bp\_hcbh: Height of the CBH based on the breaking point method or on the maximum LAD criterium if there is not breaking point (m)
- bp\_Hdptf: Height of the canopy layer depth using the breaking point method or the maximum LAD criterium (m)
- bp\_dptf: Depth of the CBH using the breaking point method or the maximum LAD criterium (m)
- bp\_Hdist: Height of the distance between the CBH and the ground using the breaking point method or the maximum LAD criterium (m)
- bp\_effdist: Distance between the CBH and the ground using the breaking point method or the maximum LAD criterium (m)
- bp\_lad: Percentage of LAD comprised by the canopy layer
- cumlad: Cumulative LAD values at the CBH or breaking point
- nlayers - Number of effective fuel layers
- max\_height - Maximum height of the tree profile

## Value

A data frame identifying the CBH of the vertical tree profile (VTP) based on the breaking point method and the percentage of LAD values below and above the identified CBH or breaking point.

## Author(s)

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

## See Also

[get\\_cbh\\_metrics](#)

## Examples

```
library(magrittr)
library(segmented)
library(gdata)
library(dplyr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
```

```

LAD_profiles$treeID <- factor(LAD_profiles$treeID)

# Before running this example, make sure to run get_cbh_metrics().
if (interactive()) {
  cbh_metrics <- get_cbh_dist()
  LadderFuelsR::cbh_metrics$treeID <- factor(LadderFuelsR::cbh_metrics$treeID)

  trees_name1 <- as.character(cbh_metrics$treeID)
  trees_name2 <- factor(unique(trees_name1))

  cum_LAD_metrics_list <- list()

  for (i in levels(trees_name2)) {
    # Filter data for each tree
    tree1 <- LAD_profiles |> dplyr::filter(treeID == i)
    tree2 <- cbh_metrics |> dplyr::filter(treeID == i)

    # Get cumulative LAD metrics for each tree
    cum_LAD_metrics <- get_cum_break(tree1, tree2, threshold=75, min_height= 1.5, verbose=TRUE)
    cum_LAD_metrics_list[[i]] <- cum_LAD_metrics
  }

  # Combine the individual data frames
  cumulative_LAD <- dplyr::bind_rows(cum_LAD_metrics_list)
}

```

---

get\_depths

*Fuels depth in meters*


---

## Description

This function calculates fuels depth as the difference between gaps interleaved between fuel layers minus one step if the fuel depths are greater than one step.

## Usage

```
get_depths (LAD_profiles, distance_metrics, step= 1, min_height= 1.5, verbose=TRUE)
```

## Arguments

LAD_profiles	original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the <i>leafR</i> package. An object of the class text.
distance_metrics	tree metrics with gaps (distances) and fuel base heights (output of [get_distance()] function). An object of the class text.
step	Numeric value for the actual height bin step (in meters).
min_height	Numeric value for the actual minimum base height (in meters).
verbose	Logical, indicating whether to display informational messages (default is TRUE).



**Details**

# List of tree metrics:

- treeID: tree ID with strings and numeric values
- treeID1: tree ID with only numeric values
- cbh - Height of the fuel layers base height (m)
- gap - Height of gaps between consecutive fuel layers (m)
- dist: Distance between consecutive fuel layers (m)
- Hdist - Height of the distance between consecutive fuel layers (m)
- depth - Depth of fuel layers (m)
- Hdepth - Height of the depth of fuel layers (m)
- max\_height - Maximum height of the tree profile

**Value**

A data frame giving fuel layers depth and the height of the depths in meters.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**

[get\\_distance](#)

**Examples**

```
library(magrittr)
library(dplyr)

# Before running this example, make sure to run get_distance().
if (interactive()) {
  distance_metrics <- get_distance()
  LadderFuelsR::LAD_profiles$treeID <- factor(LadderFuelsR::LAD_profiles$treeID)
  LadderFuelsR::distance_metrics$treeID <- factor(LadderFuelsR::distance_metrics$treeID)

  metrics_depth_list <- list()

  for (i in levels(LAD_profiles$treeID)){

    tree1 <- LAD_profiles |> dplyr::filter(treeID == i)
    tree2 <- distance_metrics |> dplyr::filter(treeID == i)

    # Get depths for each tree
    metrics_depth <- get_depths(tree1, tree2, step= 1, min_height= 1.5, verbose=TRUE)
    metrics_depth_list[[i]] <- metrics_depth
  }
}
```

```
# Combine the individual data frames
depth_metrics <- dplyr::bind_rows(metrics_depth_list)
}
```

---

get\_distance                      *Distances between fuel layers*

---

### Description

This function calculates distances (and their heights) between fuel layers as the difference between consecutive gaps and fuel bases (the gap height always must be lower than the fuel base height).

### Usage

```
get_distance (gap_cbh_metrics, gaps_perc, step=1, min_height=1.5, verbose=TRUE)
```

### Arguments

gap_cbh_metrics	data frame with gaps (distances) and fuel base heights (output of [get_gaps_fbhs()] function). An object of the class text.
gaps_perc	data frame with Leaf Area Density (LAD) percentiles for each height values (output of [calculate_gaps_perc()] function). An object of the class text.
step	Numeric value for the actual height bin step (in meters).
min_height	Numeric value for the actual minimum base height (in meters).
verbose	Logical, indicating whether to display informational messages (default is TRUE).

### Details

# List of tree metrics:

- treeID: tree ID with strings and numeric values
- treeID1: tree ID with only numeric values
- cbh - Height of the fuel layers base height (m)
- gap - Height of gaps between consecutive fuel layers (m)
- dist: Distance between consecutive fuel layers (m)
- Hdist - Height of the distance between consecutive fuel layers (m)
- max\_height - Maximum height of the tree profile

### Value

A data frame giving distances (and their heights) between fuel layers in meters.

### Author(s)

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**[get\\_gaps\\_fbhs](#)[calculate\\_gaps\\_perc](#)**Examples**

```

library(magrittr)
library(gdata)
library(dplyr)

# Before running this example, make sure to run get_gaps_fbhs().
if (interactive()) {
  gap_cbh_metrics <- get_gaps_fbhs()
  LadderFuelsR::gap_cbh_metrics$treeID <- factor(LadderFuelsR::gap_cbh_metrics$treeID)

# Before running this example, make sure to run calculate_gaps_perc().
LadderFuelsR::gaps_perc$treeID <- factor(LadderFuelsR::gaps_perc$treeID)

trees_name1 <- as.character(gaps_perc$treeID)
trees_name2 <- factor(unique(trees_name1))

metrics_distance_list <- list()

for (i in levels(trees_name2)) {

# Filter data for each tree
tree1 <- gap_cbh_metrics |> dplyr::filter(treeID == i)
tree2 <- gaps_perc |> dplyr::filter(treeID == i)
# Get distance metrics for each tree
metrics_distance <- get_distance(tree1, tree2, step=1, min_height=1.5)
metrics_distance_list[[i]] <- metrics_distance
}
# Combine the individual data frames
distance_metrics <- dplyr::bind_rows(metrics_distance_list)
}

```

get\_effective\_gap

*Effective Distances between fuel layers***Description**

This function recalculates the distance between fuel layers after considering distances greater than any number of height bin steps.

**Usage**

```
get_effective_gap(effective_depth, number_steps = 1, min_height= 1.5, verbose=TRUE)
```

**Arguments**

<code>effective_depth</code>	Tree metrics with the recalculated depth values after considering distances greater than the actual height bin step (output of <code>[get_real_depths()]</code> function). An object of the class data frame.
<code>number_steps</code>	Numeric value for the number of height bin steps that can be jumped to reshape fuels layers.
<code>min_height</code>	Numeric value for the actual minimum base height (in meters).
<code>verbose</code>	Logical, indicating whether to display informational messages (default is TRUE).

**Details**

List of tree metrics:

- `treeID`: tree ID with strings and numeric values
- `treeID1`: tree ID with only numeric values
- `dist`: Distance between consecutive fuel layers (m)
- `dptf`: Depth of fuel layers (m) after considering distances greater than the actual height bin step
- `effdist`: Effective distance between consecutive fuel layers (m) after considering distances greater than any number of steps
- `Hcbh`: Base height of each fuel separated by a distance greater than the certain number of steps
- `Hdist`: Height of the distance (> any number of steps) between consecutive fuel layers (m)
- `Hdptf`: Height of the depth of fuel layers (m) after considering distances greater than the actual step
- `max_height`: Maximum height of the tree

**Value**

A data frame giving the effective distances (> any number of steps) between consecutive fuel layers.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**

[get\\_real\\_depths](#)

**Examples**

```

library(magrittr)
library(stringr)
library(dplyr)

# Before running this example, make sure to run get_real_depths().
if (interactive()) {
  effective_depth <- get_real_depths()
  LadderFuelsR::effective_depth$treeID <- factor(LadderFuelsR::effective_depth$treeID)

  trees_name1 <- as.character(effective_depth$treeID)
  trees_name2 <- factor(unique(trees_name1))

  corr_distance_metrics_list <- list()

  for (i in levels(trees_name2)) {
    tree1 <- effective_depth |> dplyr::filter(treeID == i)
    corr_distance_metrics <- get_effective_gap(tree1, number_steps = 1, min_height = 1.5, verbose = TRUE)
    corr_distance_metrics_list[[i]] <- corr_distance_metrics
  }

  # Combine the individual data frames
  effective_distances <- dplyr::bind_rows(corr_distance_metrics_list)

  # Get original column names
  original_column_names <- colnames(effective_distances)

  # Specify prefixes
  desired_order <- c("treeID", "Hcbh", "dptf", "effdist", "dist", "Hdist", "Hdptf", "max_")

  # Identify unique prefixes
  prefixes <- unique(sub("^[a-zA-Z]+.*", "\\1", original_column_names))
  # Initialize vector to store new order
  new_order <- c()

  # Loop over desired order of prefixes
  for (prefix in desired_order) {
    # Find column names matching the current prefix
    matching_columns <- grep(paste0("^", prefix), original_column_names, value = TRUE)
    # Append to the new order
    new_order <- c(new_order, matching_columns)
  }
  effective_distances <- effective_distances[, new_order]
}

```

**Description**

This function calculates gaps and fuel layers base height (FBH) as the difference in percentiles between consecutive LAD values along the vertical tree profile (VTP). Negative differences are linked to gaps and positive differences to fuel base height. It has been updated for reordering input columns.

**Usage**

```
get_gaps_fbhs (LAD_profiles, step=1, min_height=1.5,
perc_gap= 25, perc_base= 25, verbose=TRUE)
```

**Arguments**

LAD_profiles	original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the <i>leafR</i> package. An object of the class text.
step	Numeric value for the actual height bin step (in meters).
min_height	Numeric value for the actual minimum base height (in meters).
perc_gap	Numeric value of the percentile threshold used to identify gaps (default percentile 25th).
perc_base	Numeric value of the percentile threshold used to identify fuels layers base height (default percentile 25th).
verbose	Logical, indicating whether to display informational messages (default is TRUE).

**Details**

# List of tree metrics:

- treeID: tree ID with strings and numeric values
- treeID1: tree ID with only numeric values
- cbh - Height of the fuel layer base height (m)
- gap - Height of gap between fuel layers (m)
- gap\_lad: LAD value in the gap height
- gap\_perc - Percentage of LAD in the gap height
- cbh\_lad - LAD value in the fuel base height
- cbh\_perc - Percentage of LAD in the fuel base height
- max\_height - Maximum height of the tree profile

**Value**

A data frame giving the height of gaps and fuel layers bases in meters.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**Examples**

```

library(magrittr)
library(dplyr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
LAD_profiles$treeID <- factor(LAD_profiles$treeID)

trees_name1 <- as.character(LAD_profiles$treeID)
trees_name2 <- factor(unique(trees_name1))

metrics_precentile_list1<-list()

for (i in levels(trees_name2)) {
tree1 <- LAD_profiles |> dplyr::filter(treeID == i)
metrics_precentil <- get_gaps_fbhs(tree1, step=1,
min_height=1.5,
perc_gap= 25,perc_base= 25,
verbose=TRUE)
metrics_precentile_list1[[i]] <- metrics_precentil
}

metrics_all_percentil <- dplyr::bind_rows(metrics_precentile_list1)
metrics_all_percentil$treeID <- factor(metrics_all_percentil$treeID)

# Remove the row with all NA values from the original data frame
# First remove "treeID" and "treeID1" columns
no_treeID <- metrics_all_percentil[, -which(names(metrics_all_percentil) == c("treeID", "treeID1"))]

# Check if any row has all NA values
NA_or_zero <- apply(no_treeID, 1, function(row) all(is.na(row) | row == 0))

# Get the row index with all NA values
row_index <- which(NA_or_zero)

# Remove the row with all NA values from the original data frame
if (length(row_index) > 0) {
gap_cbh_metrics <- metrics_all_percentil[-row_index, ]
} else {
gap_cbh_metrics <- metrics_all_percentil
}

```

**Description**

This function calculates the percentage of Leaf Area Density (LAD) within each fuel layer (first output) and removes those fuel layers with LAD percentage less than a specified threshold (default 10 the depth of the remaining ones (second output).

**Usage**

```
get_layers_lad(LAD_profiles, effective_distances,
              threshold=10, step = 1, min_height= 1.5, verbose=TRUE)
```

**Arguments**

LAD_profiles	Original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the <i>leafR</i> package). An object of the class text.
effective_distances	Tree metrics of fuel layers giving the effective distances (> any number of steps) between consecutive fuel layers (output of [get_effective_gap()] function). An object of the class text.
threshold	Numeric value for the minimum required LAD percentage in a fuel layer. The default threshold is 10.
step	Numeric value for the actual height bin step (in meters).
min_height	Numeric value for the actual minimum base height (in meters).
verbose	Logical, indicating whether to display informational messages (default is TRUE).

**Details**

- treeID: tree ID with strings and numeric values
- treeID1: tree ID with only numeric values
- dptf: Depth of fuel layers (m) after considering distances greater than the actual height bin step
- effdist: Effective distance between consecutive fuel layers (m) after considering distances greater than any number of steps
- Hcbh: Base height of each fuel separated by a distance greater than the certain number of steps
- Hdptf: Height of the depth of fuel layers (m) after considering distances greater than the actual step
- Hdist: Height of the distance (> any number of steps) between consecutive fuel layers (m)
- Hcbh\_Hdptf - Percentage of LAD values comprised in each effective fuel layer
- max\_height - Maximum height of the tree profile
- nlayers - Number of effective fuel layers

**Value**

A data frame identifying the fuel layers with their corresponding LAD percentage.



**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**

[get\\_renamed\\_df](#)

[get\\_effective\\_gap](#)

**Examples**

```
library(magrittr)
library(gdata)
library(dplyr)
library(stringr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
LAD_profiles$treeID <- factor(LAD_profiles$treeID)

# Before running this example, make sure to run get_effective_gap().
if (interactive()) {
effective_distances <- get_effective_gap()
LadderFuelsR::effective_distances$treeID <- factor(LadderFuelsR::effective_distances$treeID)

trees_name1 <- as.character(effective_distances$treeID)
trees_name2 <- factor(unique(trees_name1))

LAD_metrics1 <- list()
LAD_metrics2 <- list()

for (i in levels(trees_name2)) {
# Filter data for each tree
tree1 <- LAD_profiles |> dplyr::filter(treeID == i)
tree2 <- effective_distances |> dplyr::filter(treeID == i)

# Get LAD metrics for each tree
LAD_metrics <- get_layers_lad(tree1, tree2,
threshold=10,
step = 1,min_height= 1.5,
verbose=TRUE)

LAD_metrics1[[i]] <- LAD_metrics$df1
LAD_metrics2[[i]] <- LAD_metrics$df2
}

all_LAD <- dplyr::bind_rows(LAD_metrics1)
effective_LAD <- dplyr::bind_rows(LAD_metrics2)
}
```

---

get_plots_cbh_bp	<i>Plot of the Crown Base Height (CBH) based on the breaking point method</i>
------------------	---

---

### Description

This function plots the crown base height (CBH) based on breaking point over the cumulative LAD values and gives the LAD percentage of the canopy layer

### Usage

```
get_plots_cbh_bp(LAD_profiles, cumulative_LAD, min_height = 1.5)
```

### Arguments

LAD_profiles	original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function from leafR package). An object of the class text.
cummulative_LAD	tree metrics derived from using breaking points on cumulative LAD (output of [get_cum_break()] function). An object of the class text.
min_height	Numeric value for the actual minimum base height (in meters).

### Value

A plot of the Crown Base Height (CBH) based on the breaking point method and Leaf Area Density (LAD) percentage of the canopy layer.

### Author(s)

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

### See Also

[get\\_cum\\_break](#)

### Examples

```
library(ggplot2)
library(dplyr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
LAD_profiles$treeID <- factor(LAD_profiles$treeID)

# Before running this example, make sure to run get_cum_break().
if (interactive()) {
cummulative_LAD <- get_cum_break()
```

```
LadderFuelsR::cummulative_LAD$treeID <- factor(LadderFuelsR::cummulative_LAD$treeID)

# Generate cumulative LAD plots
plots_cbh_bp <- get_plots_cbh_bp(LAD_profiles, cummulative_LAD, min_height = 1.5)
}
```

---

get_plots_cbh_LAD	<i>Plots the Crown Base Height (CBH) based on the maximum LAD percentage criterium</i>
-------------------	--

---

### Description

This function plots the CBH of a segmented tree based on the fuel layer with the maximum LAD percentage.

### Usage

```
get_plots_cbh_LAD(LAD_profiles, cbh_metrics, min_height=1.5)
```

### Arguments

LAD_profiles	Original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the <i>leafR</i> package). An object of the class <code>text</code> .
cbh_metrics	CBH metrics based on three criteria: maximum LAD percentage, maximum distance and last distance. (output of [get_cbh_metrics()] function). An object of the class <code>text</code> .
min_height	Numeric value for the actual minimum base height (in meters).

### Value

A plot drawing the Crown Base Height (CBH) of the fuel layer with the maximum Leaf Area Density (LAD) percentage.

### Author(s)

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

### See Also

[get\\_layers\\_lad](#)

**Examples**

```

library(ggplot2)
library(dplyr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
LAD_profiles$treeID <- factor(LAD_profiles$treeID)

# Before running this example, make sure to run get_cbh_metrics().
if (interactive()) {
cbh_metrics <- get_cbh_metrics()
LadderFuelsR::cbh_metrics$treeID <- factor(LadderFuelsR::cbh_metrics$treeID)

trees_name1 <- as.character(cbh_metrics$treeID)
trees_name2 <- factor(unique(trees_name1))

# Generate plots for fuels LAD metrics
plots_trees_LAD <- get_plots_cbh_LAD(LAD_profiles, cbh_metrics, min_height=1.5)
}

```

---

```
get_plots_cbh_lastdist
```

*Plots the Crown Base Height (CBH) based on the last distance criterion*

---

**Description**

This function plots the CBH of a segmented tree based on the fuel layer located at the last distance.

**Usage**

```
get_plots_cbh_lastdist(LAD_profiles, cbh_metrics, min_height=1.5)
```

**Arguments**

LAD_profiles	Original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the <i>leafR</i> package). An object of the class <code>text</code> .
cbh_metrics	CBH metrics based on three criteria: maximum LAD percentage, maximum distance and last distance. (output of [get_cbh_metrics()] function). An object of the class <code>text</code> .
min_height	Numeric value for the actual minimum base height (in meters).

**Value**

A plot drawing the Crown Base Height (CBH) of the fuel layer located at the last distance.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**

[get\\_cbh\\_metrics](#)

**Examples**

```
library(ggplot2)
library(dplyr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
LAD_profiles$treeID <- factor(LAD_profiles$treeID)

# Before running this example, make sure to run get_cbh_metrics().
if (interactive()) {
cbh_metrics <- get_cbh_metrics()
LadderFuelsR::cbh_metrics$treeID <- factor(LadderFuelsR::cbh_metrics$treeID)

trees_name1 <- as.character(cbh_metrics$treeID)
trees_name2 <- factor(unique(trees_name1))

# Generate plots for CBH based on the fuel layer at the last distance
plots_cbh_lastdist <- get_plots_cbh_lastdist(LAD_profiles, cbh_metrics, min_height=1.5)
}
```

---

get\_plots\_cbh\_maxdist *Plots the Crown Base Height (CBH) based on the maximum distance criterium*

---

**Description**

This function plots the CBH of a segmented tree based on the fuel layer located at the maximum distance.

**Usage**

```
get_plots_cbh_maxdist(LAD_profiles, cbh_metrics, min_height=1.5)
```

**Arguments**

LAD_profiles	Original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the <i>leafR</i> package). An object of the class <code>text</code> .
cbh_metrics	CBH metrics based on three criteria: maximum LAD percentage, maximum distance and last distance. (output of [get_cbh_metrics()] function). An object of the class <code>text</code> .
min_height	Numeric value for the actual minimum base height (in meters).

**Value**

A plot drawing the Crown Base Height (CBH) of the fuel layer located at the maximum distance.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**

[get\\_cbh\\_metrics](#)

**Examples**

```
library(ggplot2)
library(dplyr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
LAD_profiles$treeID <- factor(LAD_profiles$treeID)

# Before running this example, make sure to run get_cbh_metrics().
if (interactive()) {
  cbh_metrics <- get_cbh_metrics()
  LadderFuelsR::cbh_metrics$treeID <- factor(LadderFuelsR::cbh_metrics$treeID)

  trees_name1 <- as.character(cbh_metrics$treeID)
  trees_name2 <- factor(unique(trees_name1))

  # Generate plots for fuels LAD metrics
  plots_cbh_maxdist <- get_plots_cbh_maxdist(LAD_profiles, cbh_metrics, min_height=1.5)
}
```

---

get\_plots\_effective     *Plots of fuel layers with LAD percentage > 10*

---

**Description**

This function plots fuel layers with LAD percentage greater than 10.

**Usage**

```
get_plots_effective(LAD_profiles, effective_LAD, min_height = 1.5)
```

**Arguments**

LAD_profiles	Original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the <i>leafR</i> package). An object of the class text.
effective_LAD	Tree metrics with gaps (distances), fuel base heights, and depths of fuel layers with LAD percentage greater than 10 (output of [get_layers_lad()] function). An object of the class text.
min_height	Numeric value for the actual minimum base height (in meters).

**Value**

A plot drawing fuel layers with LAD percentage greater than 10.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**

[get\\_layers\\_lad](#)

**Examples**

```
library(ggplot2)
library(dplyr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
LAD_profiles$treeID <- factor(LAD_profiles$treeID)

# Before running this example, make sure to run get_layers_lad().
if (interactive()) {
effective_LAD <- get_layers_lad()
LadderFuelsR::effective_LAD$treeID <- factor(LadderFuelsR::effective_LAD$treeID)

trees_name1 <- as.character(effective_LAD$treeID)
trees_name2 <- factor(unique(trees_name1))

# Generate plots for fuels LAD metrics
plots_trees_LAD <- get_plots_cbh_LAD(LAD_profiles, effective_LAD, min_height = 1.5)
}
```

---

get\_plots\_gap\_fbh      *Plots of tree profiles with gaps and fuel layers base height (fbh)*

---

**Description**

This function plots gaps and fuel layers base height (fbh) in the vertical tree profile (VTP).

**Usage**

```
get_plots_gap_fbh (LAD_profiles, gap_cbh_metrics, min_height=1.5)
```

**Arguments**

**LAD\_profiles** original tree Leaf Area Density (LAD) profile (output of [lad.profile()] function in the *leafR* package. An object of the class text

**gap\_cbh\_metrics** data frame with gaps (distances) and fuel base heights (output of [get\_gaps\_fbhs()] function). An object of the class text.

**min\_height** Numeric value for the actual minimum base height (in meters).

**Value**

A plot drawing by lines the height of gaps and fuel layers bases in tiff format.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**

[get\\_gaps\\_fbhs](#)

**Examples**

```
library(ggplot2)
library(dplyr)

# LAD profiles derived from normalized ALS data after applying [lad.profile()] function
LAD_profiles <- read.table(system.file("extdata", "LAD_profiles.txt", package = "LadderFuelsR"),
header = TRUE)
LAD_profiles$treeID <- factor(LAD_profiles$treeID)

# Before running this example, make sure to run get_gaps_fbhs().
if (interactive()) {
gap_cbh_metrics <- get_gaps_fbhs()
LadderFuelsR::gap_cbh_metrics$treeID <- factor(LadderFuelsR::gap_cbh_metrics$treeID)

# Generate plots for gaps and fbhs
plots_gaps_fbhs <- get_plots_gap_fbh(LAD_profiles, gap_cbh_metrics, min_height=1.5)
}
```



---

get_real_depths	<i>Effective fuel layers depth</i>
-----------------	------------------------------------

---

**Description**

This function recalculates fuel layers depth after considering distances greater than the actual height bin step.

**Usage**

```
get_real_depths (effective_fbh, step=1, min_height=1.5, verbose=TRUE)
```

**Arguments**

effective_fbh	tree metrics with the recalculated base height of fuel layers after considering distances greater than any number of height bin steps (output of [get_real_fbh()] function).An object of the class text.
step	Numeric value for the actual height bin step (in meters).
min_height	Numeric value for the actual minimum base height (in meters).
verbose	Logical, indicating whether to display informational messages (default is TRUE).

**Details**

# List of tree metrics:

- treeID: tree ID with strings and numeric values
- treeID1: tree ID with only numeric values
- dist: Distance between consecutive fuel layers (m)
- Hdist - Height of the distance between consecutive fuel layers (m)
- Hcbh - Base height of each fuel separated by a distance greater than the certain number of steps
- dptf - Depth of fuel layers (m) after considering distances greater than the actual height bin step
- Hdptf - Height of the depth of fuel layers (m) after considering distances greater than the actual height bin step
- max\_height - Maximum height of the tree profile

**Value**

A data frame giving new fuel layers depth after considering distances greater than the actual height bin step.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**

[get\\_renamed0\\_df](#)  
[get\\_real\\_fbh](#)

**Examples**

```
library(magrittr)
library(tidyr)
library(dplyr)

# Before running this example, make sure to run get_real_fbh().
if (interactive()) {
  effective_fbh <- get_real_fbh()
  LadderFuelsR::effective_fbh$treeID <- factor(LadderFuelsR::effective_fbh$treeID)

  trees_name1 <- as.character(effective_fbh$treeID)
  trees_name2 <- factor(unique(trees_name1))

  depth_metrics_corr_list <- list()

  for (i in levels(trees_name2)){
    # Filter data for each tree
    tree3 <- effective_fbh |> dplyr::filter(treeID == i)
    # Get real depths for each tree
    depth_metrics_corr <- get_real_depths(tree3, step=1, min_height=1.5, verbose=TRUE)
    depth_metrics_corr_list[[i]] <- depth_metrics_corr
  }

  # Combine depth values for all trees
  effective_depth <- dplyr::bind_rows(depth_metrics_corr_list)

  # Reorder columns
  original_column_names <- colnames(effective_depth)

  # Specify prefixes
  desired_order <- c("treeID", "Hcbh", "dptf", "dist", "Hdist", "Hdptf", "max_height")

  # Identify unique prefixes
  prefixes <- unique(sub("^[a-zA-Z]+).*", "\\1", original_column_names))
  # Initialize vector to store new order
  new_order <- c()

  # Loop over desired order of prefixes
  for (prefix in desired_order) {
    # Find column names matching the current prefix
    matching_columns <- grep(paste0("^", prefix), original_column_names, value = TRUE)
    # Append to the new order
    new_order <- c(new_order, matching_columns)
  }
  effective_depth <- effective_depth[, new_order]
}
```

---

get_real_fbh	<i>Fuels base height recalculation after after considering distances greater than any number of height bin steps</i>
--------------	--

---

### Description

This function reshapes fuel layers after removing distances equal to any number of height bin steps, keeping the first "base height" from those consecutive ones separated by such distance.

### Usage

```
get_real_fbh(depth_metrics, step= 1, number_steps = 1, min_height=1.5, verbose=TRUE)
```

### Arguments

depth_metrics	Tree metrics with distances, fuel base heights, and depths (output of [get_depths()] function). An object of the class text.
step	Numeric value for the actual height bin step (in meters).
number_steps	Numeric value for the number of height bin steps that can be jumped to reshape fuels layers.
min_height	Numeric value for the actual minimum base height (in meters).
verbose	Logical, indicating whether to display informational messages (default is TRUE).

### Details

# List of tree metrics:

- treeID: tree ID with strings and numeric values
- treeID1: tree ID with only numeric values
- dist: Distance between consecutive fuel layers (m)
- Hdist - Height of the distance between consecutive fuel layers (m)
- Hcbh - Base height of each fuel separated by a distance greater than the certain number of steps
- depth - Depth of fuel layers (m)
- Hdepth - Height of the depth of fuel layers (m)
- max\_height - Maximum height of the tree profile

### Value

A data frame giving the first "base height" from those consecutive ones separated by the number of height bin steps indicated in the function. The function returns new fuel layers separated by distances greater than the indicated number of steps.

**Author(s)**

Olga Viedma, Carlos Silva, JM Moreno and A.T. Hudak

**See Also**

[get\\_depths](#)

**Examples**

```
library(magrittr)
library(dplyr)
#Before running this example, make sure to run get_depths()
if (interactive()) {
  depth_metrics <- get_depths()
  LadderFuelsR::depth_metrics$treeID <- factor(LadderFuelsR::depth_metrics$treeID)

  trees_name1 <- as.character(depth_metrics$treeID)
  trees_name2 <- factor(unique(trees_name1))

  fbh_corr_list <- list()

  for (i in levels(trees_name2)){
    # Filter data for each tree
    tree3 <- depth_metrics |> dplyr::filter(treeID == i)
    # Get real fbh for each tree
    fbh_corr <- get_real_fbh(tree3, step= 1, number_steps = 1, min_height=1.5, verbose=TRUE)
    # Store fbh values in a list
    fbh_corr_list[[i]] <- fbh_corr
  }

  # Combine fbh values for all trees
  effective_fbh <- dplyr::bind_rows(fbh_corr_list)
  effective_fbh$treeID <- factor(effective_fbh$treeID)
}
```

---

get\_renamed0\_df

*Rename and reorder columns (I)*

---

**Description**

This function reorders columns and appends numeric suffixes. Don't run it. It is an internal function.

**Usage**

```
get_renamed0_df (df)
```

**Arguments**

df                    internal data frame derived from [get\_real\_depths()] function

**Value**

No return value. The function is called for side effects.

**Examples**

```
library(dplyr)
# get_renamed0_df function reorders columns and appends numeric suffixes
```

---

get_renamed_df	<i>Rename and reorder columns (II)</i>
----------------	--

---

**Description**

This function deals with concatenated column names, reorders columns and appends numeric suffixes. Don't run it. It is an internal function.

**Usage**

```
get_renamed_df (df)
```

**Arguments**

df                    internal data frame derived from [get\_layers\_lad()] function

**Value**

No return value. The function is called for side effects.

**Examples**

```
library(dplyr)
# get_renamed_df concatenates column names, reorders columns and appends numeric suffixes
```

# Index

`calculate_gaps_perc`, [2](#), [11](#)

`get_cbh_metrics`, [3](#), [7](#), [21](#), [22](#)

`get_cum_break`, [6](#), [18](#)

`get_depths`, [8](#), [28](#)

`get_distance`, [9](#), [10](#)

`get_effective_gap`, [11](#), [17](#)

`get_gaps_fbhs`, [11](#), [13](#), [24](#)

`get_layers_lad`, [5](#), [15](#), [19](#), [23](#)

`get_plots_cbh_bp`, [18](#)

`get_plots_cbh_LAD`, [19](#)

`get_plots_cbh_lastdist`, [20](#)

`get_plots_cbh_maxdist`, [21](#)

`get_plots_effective`, [22](#)

`get_plots_gap_fbh`, [23](#)

`get_real_depths`, [12](#), [25](#)

`get_real_fbh`, [26](#), [27](#)

`get_renamed0_df`, [26](#), [28](#)

`get_renamed_df`, [17](#), [29](#)