Package 'PiC'

May 7, 2026

Type Package

Title Pointcloud Interactive Computation

Version 1.2.7

Description
Provides advanced algorithms for analyzing pointcloud data from terrestrial laser scanner in forestry applications. Key features include fast voxelization of large datasets; segmentation of point clouds into forest floor, understorey, canopy, and wood components. The package enables efficient processing of large-scale forest pointcloud data, offering insights into forest structure, connectivity, and fire risk assessment. Algorithms to analyze pointcloud data (.xyz input file). For more details, see Ferrara & Arrizza (2025) <https://hdl.handle.net/20.500.14243/533471>. For single tree segmentation details, see Ferrara et al. (2018) <doi:10.1016/j.agrformet.2018.04.008>.

License GPL (>= 3)

Depends R (>= 4.3)

Imports collapse, conicfit, data.table, dbscan, dplyr, foreach, magrittr, sf, stats, tictoc, utils

Suggests DT, fs, ggplot2, later, plotly, shiny, shinycssloaders, shinydashboard, shinydashboardPlus, shinyFeedback, shinyFiles, shinyjs, shinythemes, shinyWidgets, testthat (>= 3.0.0), tools, withr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

URL https://github.com/rupppy/PiC

BugReports https://github.com/rupppy/PiC/issues

NeedsCompilation no

Author Roberto Ferrara [aut, cre] (ORCID: <https://orcid.org/0009-0000-3627-6867>), Stefano Arrizza [ctb] (ORCID: <https://orcid.org/0009-0009-2290-3650>)

Maintainer Roberto Ferrara <roberto.ferrara@cnr.it>

Repository CRAN

Date/Publication 2025-11-07 15:30:08 UTC

Contents

`Calculate_trees_metrics` *Calculate tree and canopy metrics*

Description

Computes metrics for individual trees and forest canopy from segmented point clouds.

Usage

```
Calculate_trees_metrics(
  woodpoint,
  a,
  AGB_def,
  Forest_floor,
  plot,
  filename,
  output_path,
  canopy_voxel_size = 0.1,
  min_canopy_height = 1.5,
  coverage_method = "mean_normalized"
)
```

Arguments

`woodpoint`	Wood points (trunks and branches) with cluster attribute
`a`	Original point cloud
`AGB_def`	Non-wood (foliage) points
`Forest_floor`	Forest floor points
`plot`	Plot/output file prefix

filename	Original file prefix
output_path	Output directory
canopy_voxel_size	Voxel size for canopy analysis
min_canopy_height	Minimum height for canopy analysis
coverage_method	Coverage degree calculation method

Value

List containing tree metrics, canopy metrics, and file paths

Floseg *Forest floor segmentation*

Description

Segments the input .xyz pointcloud file into different forestry layers: forest floor and above ground biomass.

Usage

```
Floseg(a, filename="XXX", soil_dim = 0.3, th = 20, N=500, output_path = tempdir())
```

Arguments

a	- Input file (.xyz)
filename	- Output file prefix
soil_dim	- Voxel dimension (m) for forest floor segmentation - Default = 0.30
th	- Minimum number of point to generate a voxel. Default = 20
N	- Minimum number of voxel to generate a cluster. Default = 500
output_path	Directory where output files will be written. Default = tempdir()

Value

2 files (.txt) output. 1. Forest floor pointcolud; 2. AGB pointcloud

Forest_seg *Forest component segmentation*

Description

Segments an input .xyz point cloud file into different forestry layers (soil, wood, foliage), computes individual tree metrics, and provides summary statistics and canopy metrics.

Usage

```
Forest_seg(
  a,
  filename = "XXX",
  dimVox = 2,
  th = 2,
  eps = 2,
  mpts = 9,
  h_tree = 1,
  soil_dim = 0.1,
  N = 500,
  R = 30,
  Vox_print = FALSE,
  WoodVox_print = FALSE,
  output_path = tempdir(),
  analyze_canopy = TRUE,
  canopy_voxel_size = 0.1,
  min_canopy_height = 1.5,
  coverage_method = "mean_normalized"
)
```

Arguments

a	Input point cloud data frame (.xyz) or file path
filename	Output file prefix
dimVox	Voxel dimension (cm) for wood segmentation (default = 2)
th	Minimum number of points to generate a voxel (default = 2)
eps	Epsilon neighborhood radius for DBSCAN (default = 2)
mpts	Minimum points required in eps neighborhood for core points (default = 9)
h_tree	Minimum trunk length in meters (default = 1)
soil_dim	Voxel dimension (m) for forest floor segmentation (default = 0.1)
N	Minimum number of voxels in a wood cluster (default = 500)
R	Cluster shape parameter threshold (default = 30)
Vox_print	Logical; if TRUE, saves point cloud voxelization (default = FALSE)
WoodVox_print	Logical; if TRUE, saves wood voxelization (default = FALSE)

output_path Output directory (default = tempdir())

analyze_canopy Logical; if TRUE, performs canopy analysis (default = TRUE)

canopy_voxel_size
Voxel size for canopy analysis in meters (default = 0.1)

min_canopy_height
Minimum height threshold for canopy analysis (default = 1.5)

coverage_method
Method for calculating coverage degree (default = "mean_normalized")

Value

List containing file paths and metrics for trees and canopy.

---

run_PiC *Launch PiC Shiny App*

Description

Launch the Shiny app for interactive 3D point cloud processing.

Usage

```
run_PiC()
```

Details

This function launches an interactive web application for analyzing forest point cloud data. The app requires additional packages that are not installed by default. If these packages are missing, you will be prompted to install them.

Value

No return value, called for side effects (launches Shiny app)

Examples

```
## Not run:
# Launch the interactive app
run_PiC()

## End(Not run)
```

SegOne *Single Tree Wood-Leaf Segmentation and Comprehensive Metrics Calculation*

Description

Performs wood-leaf segmentation and calculates comprehensive structural metrics for individual trees from terrestrial laser scanning (TLS) point cloud data. This function implements a unified approach consistent with the Forest_seg pipeline, ensuring methodological coherence across the PiC package.

The analysis follows a four-stage processing pipeline:

1. Voxelization and wood component identification using DBSCAN clustering
2. Foliage separation through voxel-based subtraction
3. Tree structural metrics calculation (height, DBH, crown base)
4. Canopy volume quantification using density-weighted voxel analysis

Usage

```
SegOne(a, filename = "Elab_single_tree", dimVox = 2, th = 2,
       eps = 2, mpts = 6, N = 1000, R = 30, output_path = tempdir(),
       calculate_metrics = TRUE, voxel_size_canopy = 0.1,
       coverage_method = "linear")
```

Arguments

a	Input point cloud data. Can be either: (1) a data frame with x, y, z coordinates, or (2) a file path to a .txt or .xyz file containing point cloud data
filename	Character string specifying the output file prefix (default: "Elab_single_tree")
dimVox	Numeric value specifying voxel dimension in centimeters for wood segmentation. Typical range: 1-5 cm. Smaller values increase computational cost but improve spatial resolution (default: 2)
th	Integer specifying minimum number of points required to generate a voxel. Used to filter noise and low-density regions (default: 2)
eps	Numeric value specifying the epsilon neighborhood radius for DBSCAN clustering in voxel units. Determines spatial connectivity of wood clusters (default: 2)
mpts	Integer specifying minimum number of points required in epsilon neighborhood for a voxel to be considered a core point in DBSCAN algorithm (default: 6)
N	Integer specifying minimum number of voxels required to form a valid wood cluster. Filters small non-wood clusters (default: 1000)
R	Numeric threshold for cluster shape parameter (standard deviation proportion of variance from PCA). Used to identify cylindrical/linear wood structures. Higher values are more restrictive (default: 30)

`output_path` Character string specifying directory path for output files. If directory does not exist, it will be created (default: tempdir())

`calculate_metrics`
Logical flag indicating whether to calculate comprehensive tree metrics. If FALSE, only wood-leaf segmentation is performed (default: TRUE)

`voxel_size_canopy`
Numeric value specifying voxel size in meters for canopy volume calculation. Typical range: 0.05-0.2 m (default: 0.1)

`coverage_method`
Character string specifying method for calculating coverage degree in canopy volume analysis. Options: "linear", "mean_normalized", "exponential", "threshold", "mediterranean" (default: "linear")

Details

Processing Pipeline

Stage 1: Wood Segmentation

Wood components are identified through a multi-step process:

1. Point cloud voxelization at resolution specified by `dimVox`
2. DBSCAN clustering applied to voxel centroids using `eps` and `mpts`
3. Principal Component Analysis (PCA) filtering to identify cylindrical structures
4. Retention of clusters with shape parameter R exceeding threshold (cylindrical wood)

The PCA-based filtering exploits the geometric properties of tree stems and branches, which exhibit high first principal component values due to their elongated structure.

Stage 2: Foliage Separation

Foliage points are extracted using voxel-based subtraction:

1. Both wood and total point cloud are voxelized at 0.2 m resolution
2. Wood-occupied voxels are identified
3. Non-wood voxels are retained and mapped back to original points

This approach ensures complete spatial separation between wood and foliage components.

Stage 3: Structural Metrics Calculation

When `calculate_metrics = TRUE`, the following metrics are computed:

- **Tree Base Location (X, Y, Z_min)**: Coordinates of lowest wood point
- **Tree Height**: Vertical distance from base to highest point within 1 m buffer
- **DBH (Diameter at Breast Height)**: Calculated at 1.3 m using Pratt circle fitting algorithm with +/- 5 cm tolerance. Valid range: 5-300 cm. DBH value is always reported in CSV output. DBH_RMSE_cm column provides fit quality (max 5 cm for validation). DBH_valido flag indicates whether measurement meets quality standards.
- **Crown Base Height**: Detected using vertical density analysis to filter noise, followed by gap analysis. Uses 0.5 m bins with minimum 3 points per bin. Valid range: 0.5 m to 90

Stage 4: Canopy Volume Quantification

Canopy volume metrics are calculated using density-weighted voxel analysis:

1. Foliage points voxelized at resolution `voxel_size_canopy`
2. Point density calculated per voxel
3. Coverage degree computed using specified `coverage_method`
4. Two volume metrics calculated:
   - **Canopy Volume**: Total occupied voxel volume (m^3)
   - **Occupied Volume**: Density-weighted volume accounting for point distribution
5. Coverage area computed from ground projection of occupied voxels (mq)

Improvements in Version 2.0

Enhanced Crown Base Calculation:

- Vertical density analysis filters noise points (isolated points near trunk)
- Uses 0.5 m vertical bins with minimum 3 points per bin threshold
- Combines density filtering with gap analysis for robust detection
- Validates results with multiple criteria

Improved DBH Validation:

- Updated maximum diameter to 3.0 m (300 cm) for large/monumental trees
- RMSE quality check (max 5 cm) used to flag poor circle fits
- DBH value always reported (even if RMSE threshold exceeded)
- DBH_RMSE_cm column provides fit quality indicator
- DBH_valido flag indicates whether measurement meets all quality standards
- Enhanced diagnostic messages showing fit quality

Coverage Degree Methods

The `coverage_method` parameter determines how point density is translated to coverage degree:

- **linear**: Linear normalization by column maximum: $CD = N/N_{max}$
- **mean_normalized**: Normalization by mean density: $CD = N/\bar{N}$
- **exponential**: Exponential saturation: $CD = 1 - exp(-N/\bar{N})$
- **threshold**: Binary classification at 50th percentile
- **mediterranean**: Power-law scaling optimized for sparse Mediterranean canopies: $CD = (N/N_{max})^{0.7}$

For single trees, "linear" is recommended as it provides intuitive interpretation of point density relative to maximum observed density.

Quality Assurance

The function implements several validation checks:

- DBH validation: radius 2.5-150 cm, minimum 5 points, RMSE reported (< 5 cm for valid flag)
- Crown base validation: density filtering, minimum 0.5 m, maximum 90
- Point count validation: sufficient points in measurement zones
- Comprehensive error handling with diagnostic messages

Output Files

Three files are generated in `output_path`:

1. **Wood points**: `<filename>_Wood_eps<eps>_mpts<mpts>.txt`
   - Format: x, y, z, cls (cluster ID)
   - Contains all points classified as wood components
2. **Foliage points**: `<filename>_AGBnoWOOD_eps<eps>_mpts<mpts>.txt`
   - Format: x, y, z
   - Contains all non-wood vegetation points
3. **Metrics**: `<filename>_metrics.csv` (if calculate_metrics = TRUE)
   - Contains all calculated structural metrics
   - DBH_cm: Always populated when calculation succeeds
   - DBH_RMSE_cm: Fit quality indicator (lower is better, <5 cm is valid)
   - Semicolon-delimited format

Value

Invisibly returns a named list containing:

wood_file Character string with full path to wood component point cloud file

leaf_file Character string with full path to foliage point cloud file

metrics_file Character string with full path to metrics CSV file (NULL if calculate_metrics = FALSE)

metrics data.table containing calculated tree structural metrics (NULL if calculate_metrics = FALSE)

Parameter Selection Guidelines

Voxel Size (dimVox):

- Small trees or fine branches: 1-2 cm
- Medium trees: 2-3 cm
- Large trees: 3-5 cm

DBSCAN Parameters (eps, mpts):

- Densely scanned trees: eps = 1-2, mpts = 4-6
- Sparsely scanned trees: eps = 2-3, mpts = 3-4
- Complex branch structures: lower eps, higher mpts

Cluster Size (N):

- Small trees: N = 500-1000
- Medium trees: N = 1000-2000
- Large trees: N = 2000-5000

Note

Version 2.0 addresses two critical issues identified in field testing:

1. Crown base calculation now robust against noise points near trunk
2. DBH validation extended to 3 m diameter with quality checks

This implementation is fully consistent with the Forest_seg approach, ensuring methodological coherence across the PiC package.

References

Ferrara, R., Virdis, S.G.P., Ventura, A., Ghisu, T., Duce, P., & Pellizzaro, G. (2018). An automated approach for wood-leaf separation from terrestrial LIDAR point clouds using the density based clustering algorithm DBSCAN. Agricultural and Forest Meteorology, 262, 434-444. doi:10.1016/j.agrformet.2018.04.008

Pratt, V. (1987). Direct least-squares fitting of algebraic surfaces. ACM SIGGRAPH Computer Graphics, 21(4), 145-152.

Examples

```
## Not run:
# Basic usage with default parameters
result <- SegOne(
  a = "single_tree.xyz",
  filename = "tree_analysis",
  output_path = "~/results"
)

# View calculated metrics
print(result$metrics)

# Advanced usage for large tree
result <- SegOne(
  a = my_point_cloud,
  filename = "large_oak",
  dimVox = 3,           # Larger voxels for large tree
  eps = 2,              # Increased connectivity
  mpts = 6,             # More stringent clustering
  N = 2000,             # Larger minimum cluster size
  R = 35,               # Stricter cylindrical filter
  output_path = "~/tree_metrics",
  voxel_size_canopy = 0.15,
  coverage_method = "linear"
)
```

```
## End(Not run)
```

`Voxels` *Voxelize point cloud*

Description

Transform pointcloud in voxel

Usage

```
Voxels(a, filename = "XXX", dimVox = 2, th = 2, output_path = tempdir())
```

Arguments

`a`	- input file
`filename`	- file output prefix
`dimVox`	- voxel dimension in cm - Default = 2
`th`	Minimum number of point to generate a voxel (Default = 1) Is a parameter that should be used with caution; it generates a lightened cloud with fewer points. To be evaluated in relation with the dimVox parameter, for high point densities it is efficae to remove noise (outliers)
`output_path`	Directory in cui scrivere i file di output. Default = tempdir()

Value

Voxelized pointcloud

Index