Package 'fractalforest'

June 19, 2025

Title Simulate Fractal Trees and Forests

Version 1.0.0

Description Create and visualize fractal trees and fractal forests, based on the Lindenmayer system (L-system). For more details see Lindenmayer (1968a) <doi:10.1016/0022-5193(68)90079-9> and Lindenmayer (1968b) <doi:10.1016/0022-5193(68)90080-5>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports dplyr, purrr, stringi, stringr, magrittr, ggplot2, cowplot, sf, rlang, stats, utils

Suggests knitr, rmarkdown, spelling, forestmangr

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Sergio Costa [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-5432-317X>), Gilmara Machado [aut, cph] (ORCID: <https://orcid.org/0000-0003-1455-9086>)

Maintainer Sergio Costa <sergio.vscf@gmail.com>

Contents

---

build_forest_profile	*Construct a 2D Forest Profile Using L-System Trees from Inventory Data.*

Description

Construct a 2D Forest Profile Using L-System Trees from Inventory Data.

Usage

```
build_forest_profile(
  data,
  height,
  diameter,
  label,
  string = NULL,
  tree_model = "binary_tree",
  n_iter = NULL,
  angle = 15,
  crown_diameter = NULL,
  h_reduction = (1 + sqrt(5))/2 - 1,
  d_reduction = (1 + sqrt(5))/2 - 1,
  randomness = FALSE,
  angle_cv = 0.1,
  length_cv = 0.1,
  leaf_size = NULL,
  sample = TRUE,
  n_trees,
  dist = 3,
  dist_cv = 0
)
```

Arguments

data	A data frame containing forest inventory data (field-measured or simulated). Each row represents an individual tree.
height	Name of the column (character) containing tree heights, measured in meters.
diameter	Name of the column (character) containing tree diameters, measured in centimeters.
label	Name of the column (character) containing tree labels or unique IDs.

string	The column (character) or string containing turtle graphics instructions generated by the `iterate_lsystem` function. If NULL (default), the `tree_model` argument is used to create instructions.
tree_model	The predefined tree model to use when `string` is NULL. Accepts either an integer or a character string corresponding to a model name (e.g., 'binary_tree'). See `fractal_tree_model()` for details.
n_iter	An integer specifying the number of iterations to generate the `tree_model` when `string` is NULL. Defaults to model-specific settings.
angle	A numeric value specifying the branching angle (in degrees). Default is 15.
crown_diameter	Column name (unquoted) or numeric value for the crown diameter of the trees. Default is NULL.
h_reduction	A numeric value representing the reduction factor applied to branch lengths. Default is the golden ratio $((1 + \sqrt{5})/2 - 1)$.
d_reduction	A numeric value representing the reduction factor applied to branch diameters. Default is the golden ratio $((1 + \sqrt{5})/2 - 1)$.
randomness	A logical value. If set to TRUE, randomness is introduced to the branch angles. Default is FALSE.
angle_cv	A numeric value specifying the coefficient of variation for branch angles, applied when `randomness` is set to TRUE. Default is 0.1.
length_cv	A numeric value specifying the coefficient of variation for branch lengths, applied when `randomness` is set to TRUE. Default is 0.1.
leaf_size	An unquoted column name or a numeric value specifying the size of the leaves. Defaults to NULL, in which case the `d_reduction` value is used.
sample	A logical value. If TRUE, rows are sampled from data to construct the forest. The default is TRUE.
n_trees	An integer specifying the number of trees to sample when `sample` is set to TRUE..
dist	A numeric value representing the average linear distance between trees. The default is 3.
dist_cv	A numeric value representing the coefficient of variation for tree distances. The default is 0.3.

Details

- When `string` is specified, it must be a column in `data` containing turtle graphics strings for each tree.
- If `sample` is TRUE, `n_trees` must be provided.

Value

A data frame containing the 2D forest profile data, with individual tree id, component coordinates, component types, labels and diameters.

Examples

```
# Example data
data <- data.frame(
  height = c(10, 12, 14, 12, 18, 15, 13, 16, 10, 8, 10, 7),
  diameter = c(12, 15, 11, 17, 25, 18, 10, 21, 12, 10, 10, 11),
  label = c("Tree1", "Tree2", "Tree3","Tree4", "Tree5", "Tree6",
            "Tree7", "Tree8", "Tree9", "Tree10", "Tree11", "Tree12")
)

# Build forest profile
build_forest_profile(
  data = data,
  height = height,
  diameter = diameter,
  label = label,
  tree_model = "binary_tree",
  n_trees = 3
)
```

---

build_tree   *Build a 2D L-System Tree Using Turtle Graphics*

Description

This function generates a 2D representation of a tree using L-system string instructions and turtle graphics. The tree structure is determined by the L-system string, and various parameters control the shape, size, and randomness of the tree's growth.

Usage

```
build_tree(
  string = NULL,
  angle = 15,
  height = NULL,
  diameter = NULL,
  crown_diameter = NULL,
  h_reduction = (1 + sqrt(5))/2 - 1,
  d_reduction = (1 + sqrt(5))/2 - 1,
  randomness = FALSE,
  angle_cv = 0.1,
  length_cv = 0.1,
  leaf_size = NULL
)
```

Arguments

string   A character vector containing the turtle graphics instructions, typically generated by the `iterate_lsystem` function.

angle	A numeric value specifying the branching angle (in degrees). Default is 15.
height	A numeric value specifying the total height of the plant, in meters.
diameter	A numeric value specifying the base diameter of the plant, in centimeters.
crown_diameter	A numeric value specifying the diameter of the plant crown, in meters.
h_reduction	A numeric value representing the reduction factor applied to branch lengths. Default is the golden ratio ($(1 + \sqrt{5})/2 - 1$).
d_reduction	A numeric value representing the reduction factor applied to branch diameters. Default is the golden ratio ($(1 + \sqrt{5})/2 - 1$).
randomness	A logical value. If set to TRUE, randomness is introduced to the branch angles. Default is FALSE.
angle_cv	A numeric value specifying the coefficient of variation for branch angles, applied when randomness is set to TRUE. Default is 0.1.
length_cv	A numeric value specifying the coefficient of variation for branch lengths, applied when randomness is set to TRUE. Default is 0.1.
leaf_size	An unquoted column name or a numeric value specifying the size of the leaves. Defaults to NULL, in which case the d_reduction value is used.

Value

A data frame containing the 2D tree profile data (component coordinates and types).

---

fractal_tree_model   *Build a 2D L-System Tree from Predefined Tree Templates*

Description

This function generates 2D L-system trees using predefined templates. Each template is based on a specific tree model, and the function allows you to control the number of iterations to generate the desired tree structure.

Usage

```
fractal_tree_model(tree_model, n_iter)
```

Arguments

tree_model   An integer or a character string specifying the desired tree model. The following models are available:

- 1 or "binary_tree" (default n_iter = 6)
- 2 or "alternate_tree" (default n_iter = 5)
- 3 or "arrow_weed" (default n_iter = 5)
- 4 or "twiggy_weed" (default n_iter = 5)
- 5 or "stochastic_fuzzy_weed" (default n_iter = 4)

- 6 or "crooked_binary_tree" (default `n_iter` = 6)
- 7 or "crooked_alternate_tree" (default `n_iter` = 5)
- 8 or "crooked_arrow_weed" (default `n_iter` = 5)
- 9 or "crooked_twiggy_weed" (default `n_iter` = 5)
- 10 or "crooked_stochastic_fuzzy_weed" (default `n_iter` = 4)

n_iter   An integer specifying the number of iterations to build the tree. Each tree model
                has its own default value, but it can be overridden by specifying this parameter.

Details

This function uses a set of predefined L-system rules to generate the tree structure. The rules for
each tree model are stored in the `rules` data frame. The default number of iterations (`n_iter`) for
each tree model is specified in the list above. If `n_iter` is not provided, the default value for the
selected tree model will be used.

Value

A list representing the L-system tree structure after applying the specified number of iterations.

---

generate_point_cloud   *Generate a Point Cloud from 2D trees data frame*

Description

This function generates a point cloud based on the geometry of a tree or forest profile.

Usage

```
generate_point_cloud(tree_df, pt_distance = 0.3)
```

Arguments

tree_df   A data.frame resulting from the `build_tree` or `build_forest_profile` func-
                tions. It contains the tree geometry data.

pt_distance   A numeric value specifying the distance between points in the generated point
                cloud. The default is 0.3. Smaller values will result in a denser point cloud.
                Small values may significantly increase the execution time of the function due
                to the larger number of points being generated.

Details

This function processes the input `tree_df` by creating lines from the tree's geometry and then
buffers these lines to account for the tree's diameter. A grid is then created over the buffered
geometry, and points are generated where the grid intersects the geometry. The result is a point
cloud of 2D coordinates representing the tree's profile.

Value

A matrix of 2D coordinates representing the point cloud. Each row corresponds to a point in the cloud, with the columns representing the X and Y coordinates.

---

get_tree_matrix *Extract the Fractal Tree Matrix from a Tree Data Frame*

Description

Extract the Fractal Tree Matrix from a Tree Data Frame

Usage

```
get_tree_matrix(tree_df)
```

Arguments

tree_df   A data frame containing the tree data, typically resulting from the `build_tree` function. The data frame should include columns `from_x`, `from_y`, `to_x`, and `to_y`, which represent the coordinates of the tree segments.

Details

This function extracts the start and end coordinates of the tree segments and returns the result as a matrix.

Value

A matrix of unique coordinates (x, y) representing the positions of the tree's segments.

---

is_branch_symbol *Check if character is a branch symbol*

Description

Check if character is a branch symbol

Usage

```
is_branch_symbol(x)
```

Arguments

x   an R object to be tested.

Value

A logical vector the same length as x, with TRUE for elements that are *not* one of the branch symbols
(``"["``, ``"]"``, ``"+"``, ``"-"``, ``"("``, ``")"``, ``"<"``, ``">"``), and FALSE otherwise.

---

is_pop_symbol   *Check if character is a pop symbol*

Description

Check if character is a pop symbol

Usage

```
is_pop_symbol(x)
```

Arguments

x   an R object to be tested.

Value

A logical vector the same length as x, with TRUE for elements equal to the pop symbol ``"]"``, and
FALSE otherwise.

---

iterate_lsystem   *Rewrite an Axiom Using Production Rules to Generate a String for*
Turtle Graphics

Description

This function iterates over a set of production rules to transform an initial axiom (a string) into a
final string that can be used for turtle graphics. The rules are applied to the axiom over a specified
number of iterations, with each iteration updating the string according to the rules provided.

Usage

```
iterate_lsystem(init = NULL, rules = NULL, n_iter = 5, verbose = 0L)
```

Arguments

`init`	A character string representing the initial axiom (starting string) for the L-system.
`rules`	A data frame with two columns: `inp` (input variables) and `out` (corresponding replacement strings). The rules define how each symbol in the axiom should be replaced during the iterations.
`n_iter`	An integer specifying the number of iterations or cycles to apply the production rules. Default is 5.
`verbose`	An integer controlling the verbosity of the output during iterations. - `verbose = 1L`: prints basic information about each cycle. - `verbose > 1L`: provides detailed information about each step. - `verbose < 1L`: suppresses output.

Details

The function applies the provided production rules to the initial axiom for the specified number of iterations. In each cycle, the rules are applied simultaneously to the current string, and the resulting string is updated. The function also supports different verbosity levels for reporting the transformation progress.

Value

A character string that represents the final string after all iterations, ready for use in turtle graphics.

Examples

```
# Example of a simple L-system with a binary tree
init <- "0"
rules <- data.frame(
  inp = c("0", "1"),
  out = c("1[-(0)]+(0)", "1"),
  stringsAsFactors = FALSE
)
result <- iterate_lsystem(init, rules, n_iter = 3, verbose = 1L)
print(result)
```

---

plot_forest_profile	*Visualize the 2D profile of a fractal trees forest*

Description

This function generates a 2D visualization of the forest profile using data from a fractal trees. The forest is plotted with tree branches and leaves. The user can customize the appearance using different parameters.

Usage

```
plot_forest_profile(
  tree_df,
  d_col = NULL,
  branch_color = "black",
  leaf_color = "black",
  label = NULL,
  label_size = 4,
  simplify = FALSE,
  dTolerance = 0.15
)
```

Arguments

tree_df	A data frame resulting from the `build_forest_profile` function, containing the geometry and attributes of the trees.
d_col	Name of the column that contains the diameters along the branches. Usually 'diameter'. Default is NULL (diameters are not used).
branch_color	A character string specifying the color of the tree branches. Default is 'black'.
leaf_color	A character string specifying the color of the tree leaves. Default is 'black'.
label	A character string or column name specifying the labels for the trees (typically placed along the x-axis). Default is `NULL`.
label_size	A numeric value specifying the size of the labels on the plot. Default is 4.
simplify	A logical value. If `TRUE`, simplifies the geometries of the tree branches using a given tolerance. Default is `FALSE`.
dTolerance	A numeric value specifying the tolerance parameter used when simplifying geometries, applicable if `simplify = TRUE`. Default is `0.15`.

Value

A ggplot object representing the forest profile.

---

plot_tree	*Visualize a fractal tree*

Description

This function generates a visualization of a fractal tree based on the geometry and attributes from the `build_tree` function. The tree is plotted with branches and leaves. The user can customize the appearance using various parameters.

Usage

```
plot_tree(
  tree_df,
  d_col = NULL,
  branch_color = "black",
  leaf_color = "black",
  simplify = FALSE,
  dTolerance = 0.15
)
```

Arguments

tree_df	A data frame resulting from the `build_tree` function, containing the tree's geometry and attributes.
d_col	Name of the column that contains the diameters along the branches. Usually 'diameter'. Default is NULL (diameters are not used).
branch_color	A character string specifying the color of the tree branches. Default is 'black'.
leaf_color	A character string specifying the color of the tree leaves. Default is 'black'.
simplify	A logical value. If `TRUE`, simplifies the geometries of the tree branches using a given tolerance. Default is `FALSE`.
dTolerance	A numeric value specifying the tolerance parameter used when simplifying geometries, applicable if `simplify = TRUE`. Default is `0.15`.

Value

A ggplot object representing the tree profile.

Index