

Package ‘pdi’

May 9, 2026

Type Package

Title Phenotypic Index Measures for Oak Decline Severity

Version 0.4.2

Description

Oak declines are complex disease syndromes and consist of many visual indicators that include aspects of tree size, crown condition and trunk condition. This can cause difficulty in the manual classification of symptomatic and non-symptomatic trees from what is in reality a broad spectrum of oak tree health condition. Two phenotypic oak decline indexes have been developed to quantitatively describe and differentiate oak decline syndromes in *Quercus robur*. This package provides a toolkit to generate these decline indexes from phenotypic descriptors using the machine learning algorithm random forest. The methodology for generating these indexes is outlined in Finch et al. (2021) <[doi:10.1016/j.foreco.2021.118948](https://doi.org/10.1016/j.foreco.2021.118948)>.

URL <https://jasenfinch.github.io/pdi>

BugReports <https://github.com/jasenfinch/pdi/issues>

License GPL-3

Encoding UTF-8

Imports dplyr, magrittr, purrr, randomForest, readxl, stringr, tibble, tidy, tidyselect

LazyData true

RoxygenNote 7.1.1

Suggests testthat, covr, knitr, rmarkdown, ggplot2

VignetteBuilder knitr

NeedsCompilation no

Author Jasen Finch [aut, cre] (ORCID: <<https://orcid.org/0000-0002-6070-7476>>)

Maintainer Jasen Finch <jsf9@aber.ac.uk>

Repository CRAN

Date/Publication 2021-02-09 17:40:02 UTC

Contents

agrilusExitHoleDensity	2
bleedPrevalence	3
calcDIs	3
crownCondition	4
crownProductionEfficiency	5
crownSurfaceArea	5
crownVolume	6
descriptorContributions	7
liveCrownRatio	8
makeAnalysisTable	8
mds	9
minMaxScale	10
phenotypingTemplate	11
preparePhenotypeData	11
readPhenotypeSheet	12
rf	12
siteAdjustment	13
siteAdjustmentFactors	14
Index	16

agrilusExitHoleDensity
Agrilus exit hole density (m⁻²)

Description

Calculate Agrilus biguttatus exit hole density.

Usage

```
agrilusExitHoleDensity(n, d, s = 2)
```

Arguments

n	number of Agrilus exit holes
d	diameter at breast height (m)
s	height to which stem surveyed from the tree base (m)

Examples

```
agrilusExitHoleDensity(2,1.02,1.3)
```

bleedPrevalence	<i>Estimated bleed prevalence (%)</i>
-----------------	---------------------------------------

Description

Calculate estimated bleed prevalence.

Usage

```
bleedPrevalence(a, A, b, B, d, s = 3)
```

Arguments

a	average active bleed size (mm)
A	number of active bleeds
b	average black stain size (mm)
B	number of black stains
d	diameter at breast height (m)
s	height to which stem surveyed from the tree base (m)

Examples

```
bleedPrevalence(30,10,40,5,1,1.3)
```

calcDIs	<i>Calculate Decline Indexes</i>
---------	----------------------------------

Description

Calculate Phenotypic Decline Index (PDI) and Decline Acuteness Index (DAI).

Usage

```
calcDIs(rfModels, PDI = TRUE, DAI = TRUE, invertPDI = TRUE, invertDAI = TRUE)
```

Arguments

rfModels	list containing random forest models as returned by <code>rf()</code>
PDI	TRUE/FALSE, calculate PDI?
DAI	TRUE/FALSE, calculate DAI?
invertPDI	invert the PDI scale? TRUE/FALSE. Ignored if argument PDI is FALSE
invertDAI	invert the DAI scale? TRUE/FALSE. Ignored if argument DAI is FALSE

Examples

```

library(dplyr)

## Retrieve file paths for example data
files <- list.files(system.file('phenotypeDataCollectionSheets',
  package = 'pdi'),full.names = TRUE)

## Prepare data
d <- map(files,readPhenotypeSheet) %>%
  map(preparePhenotypeData) %>%
  bind_rows() %>%
  siteAdjustment() %>%
  mutate(`Live crown ratio (%)` = liveCrownRatio(`Total height (m)`,
    `Lower crown height (m)`),
    `Crown condition (%)` = crownCondition(`Missing crown (%)`,
      `Crown transparency (%)`),
    `Crown volume (m^3)` = crownVolume(`Crown radius (m)`,
      `Total height (m)`,
      `Lower crown height (m)`,
      `Crown condition (%)`),
    `Bleed prevalence (%)` = bleedPrevalence(`Active bleed length (mm)`,
      `Active bleeds`,
      `Black staining length (mm)`,
      `Black staining`,
      `Diameter at breast height (m)`),
    `Agrilus exit hole density (m^-2)` = agrilusExitHoleDensity(`Agrilus exit holes`,
      `Diameter at breast height (m)`))

)

t <- makeAnalysisTable(d)

## Generate random forest models
m <- rf(t,cls = NULL,nreps = 10)

## Calculate decline indexes
DIs <- calcDIs(m,DAI = FALSE,invertPDI = FALSE) %>%
  bind_cols(d %>%
    select(Location,ID,Status))

```

crownCondition

Crown condition

Description

Calculate crown condition (%).

Usage

```
crownCondition(m, t)
```

Arguments

m missing crown (%)
t crown transparency (%)

Examples

`crownCondition(50,60)`

`crownProductionEfficiency`
Crown production efficiency

Description

Calculate the crown production efficiency

Usage

`crownProductionEfficiency(crown_surface_area, crown_volume)`

Arguments

`crown_surface_area`
 crown surface area (m²)
`crown_volume` crown volume (m³)

Examples

`crownProductionEfficiency(34,35)`

`crownSurfaceArea` *Crown surface area (m²)*

Description

Calculate the crown surface area

Usage

`crownSurfaceArea(r, h, l, c)`

Arguments

r	crown radius (m)
h	total height (m)
l	lower crown height (m)
c	crown condition (%)

Examples

```
crownSurfaceArea(3,15,10,50)
```

crownVolume	<i>Estimated crown volume (m³)</i>
-------------	---

Description

Calculate estimated crown volume.

Usage

```
crownVolume(r, h, l, c)
```

Arguments

r	crown radius (m)
h	total height (m)
l	lower crown height (m)
c	crown condition (%)

Examples

```
crownVolume(3,15,10,50)
```

```
descriptorContributions
      Descriptor contributions
```

Description

Calculate average descriptor contributions to random forest models.

Usage

```
descriptorContributions(rfModels)
```

Arguments

rfModels list containing random forest models as returned by rf()

Details

See see ?randomForest::importance for details on random forest importance metrics.

Examples

```
library(dplyr)

## Retrieve file paths for example data
files <- list.files(system.file('phenotypeDataCollectionSheets',
  package = 'pdi'),full.names = TRUE)

## Prepare data
d <- map(files,readPhenotypeSheet) %>%
  map(preparePhenotypeData) %>%
  bind_rows() %>%
  siteAdjustment() %>%
  mutate(`Live crown ratio (%)` = liveCrownRatio(`Total height (m)`,
    `Lower crown height (m)`),
    `Crown condition (%)` = crownCondition(`Missing crown (%)`,
      `Crown transparency (%)`),
    `Crown volume (m^3)` = crownVolume(`Crown radius (m)`,
      `Total height (m)`,
      `Lower crown height (m)`,
      `Crown condition (%)`),
    `Bleed prevalence (%)` = bleedPrevalence(`Active bleed length (mm)`,
      `Active bleeds`,
      `Black staining length (mm)`,
      `Black staining`,
      `Diameter at breast height (m)`),
    `Agrilus exit hole density (m^-2)` = agrilusExitHoleDensity(`Agrilus exit holes`,
      `Diameter at breast height (m)`))
)
```

```
t <- makeAnalysisTable(d)

## Generate random forest models
m <- rf(t,cls = NULL,nreps = 10)

descriptor_contributions <- m %>%
  descriptorContributions()
```

liveCrownRatio	<i>Live crown ratio</i>
----------------	-------------------------

Description

Calculate the live crown ratio

Usage

```
liveCrownRatio(h, l)
```

Arguments

h	total height (m)
l	lower crown height (m)

Examples

```
liveCrownRatio(15,10)
```

makeAnalysisTable	<i>Make analysis table</i>
-------------------	----------------------------

Description

prepare data table ready for random forest analysis

Usage

```
makeAnalysisTable(phenoData)
```

Arguments

phenoData	tibble containing phenotype data
-----------	----------------------------------

Examples

```

library(dplyr)

## Retrieve file paths for example data
files <- list.files(system.file('phenotypeDataCollectionSheets',
  package = 'pdi'),full.names = TRUE)

## Prepare data
d <- map(files,readPhenotypeSheet) %>%
  map(preparePhenotypeData) %>%
  bind_rows() %>%
  siteAdjustment() %>%
  mutate(`Live crown ratio (%)` = liveCrownRatio(`Total height (m)`,
    `Lower crown height (m)`),
    `Crown condition (%)` = crownCondition(`Missing crown (%)`,
      `Crown transparency (%)`),
    `Crown volume (m^3)` = crownVolume(`Crown radius (m)`,
      `Total height (m)`,
      `Lower crown height (m)`,
      `Crown condition (%)`),
    `Bleed prevalence (%)` = bleedPrevalence(`Active bleed length (mm)`,
      `Active bleeds`,
      `Black staining length (mm)`,
      `Black staining`,
      `Diameter at breast height (m)`),
    `Agrilus exit hole density (m^-2)` = agrilusExitHoleDensity(`Agrilus exit holes`,
      `Diameter at breast height (m)`))
)

t <- makeAnalysisTable(d)

```

mds

*Multidimensional scaling***Description**

perform multidimensional scaling of random forest proximities

Usage

```
mds(rfModels, dimensions = 2)
```

Arguments

rfModels list containing random forest models as returned by rf()
 dimensions number of dimensions to scale to

Examples

```

library(dplyr)

## Retrieve file paths for example data
files <- list.files(system.file('phenotypeDataCollectionSheets',
  package = 'pdi'),full.names = TRUE)

## Prepare data
d <- map(files,readPhenotypeSheet) %>%
  map(preparePhenotypeData) %>%
  bind_rows() %>%
  siteAdjustment() %>%
  mutate(`Live crown ratio (%)` = liveCrownRatio(`Total height (m)`,
    `Lower crown height (m)`),
    `Crown condition (%)` = crownCondition(`Missing crown (%)`,
      `Crown transparency (%)`),
    `Crown volume (m^3)` = crownVolume(`Crown radius (m)`,
      `Total height (m)`,
      `Lower crown height (m)`,
      `Crown condition (%)`),
    `Bleed prevalence (%)` = bleedPrevalence(`Active bleed length (mm)`,
      `Active bleeds`,
      `Black staining length (mm)`,
      `Black staining`,
      `Diameter at breast height (m)`),
    `Agrilus exit hole density (m^-2)` = agrilusExitHoleDensity(`Agrilus exit holes`,
      `Diameter at breast height (m)`
  )

t <- makeAnalysisTable(d)

## Generate random forest models
m <- rf(t,cls = NULL,nreps = 10)

mds_data <- mds(m,2)

```

minMaxScale

Min-max scaling

Description

Variable min-max scaling.

Usage

```
minMaxScale(vec)
```

Arguments

vec vector of numbers to scale

Examples

```
set.seed(1234)

d <- runif(20,1,10)

minMaxScale(d)
```

phenotypingTemplate *Phenotyping template*

Description

Export a copy of the oak phenotyping data collection spreadsheet.

Usage

```
phenotypingTemplate(path = ".")
```

Arguments

path directory path for export output

Examples

```
## Not run:
phenotypingTemplate()

## End(Not run)
```

preparePhenotypeData *Prepare phenotype data*

Description

Process parsed phenotype data sheets into a tibble suitable for random forest analysis.

Usage

```
preparePhenotypeData(phenotypeData)
```

Arguments

phenotypeData parsed phenotype data collection sheet returned from readPhenotypeSheet

Examples

```
library(dplyr)

## Retrieve file paths for example data
files <- list.files(system.file('phenotypeDataCollectionSheets',
  package = 'pdi'),full.names = TRUE)

## Prepare data
d <- map(files,readPhenotypeSheet) %>%
  map(preparePhenotypeData)
```

readPhenotypeSheet *Read phenotyping sheet*

Description

Parse .xlsx phenotype data collection sheets.

Usage

```
readPhenotypeSheet(file)
```

Arguments

file file path to excel file to parse

Examples

```
library(dplyr)

## Retrieve file paths for example data
files <- list.files(system.file('phenotypeDataCollectionSheets',
  package = 'pdi'),full.names = TRUE)

## Prepare data
d <- readPhenotypeSheet(files[1])
```

rf *Random forest analysis*

Description

Perform random forest repetitions.

Usage

```
rf(analysisTable, cls, params = list(), nreps = 100, seed = 1234)
```

Arguments

analysisTable	tibble of phenotype data suitable for random forest analysis as returned by preparePhenotypeData
cls	analysisTable column to use as response vector. NULL for unsupervised analyses.
params	additional arguments to pass to randomForest::randomForest
nreps	number of repetitions
seed	random number seed

Examples

```

library(dplyr)

## Retrieve file paths for example data
files <- list.files(system.file('phenotypeDataCollectionSheets',
  package = 'pdi'),full.names = TRUE)

## Prepare data
d <- map(files,readPhenotypeSheet) %>%
  map(preparePhenotypeData) %>%
  bind_rows() %>%
  siteAdjustment() %>%
  mutate(`Live crown ratio (%)` = liveCrownRatio(`Total height (m)`,
    `Lower crown height (m)`),
    `Crown condition (%)` = crownCondition(`Missing crown (%)`,
      `Crown transparency (%)`),
    `Crown volume (m^3)` = crownVolume(`Crown radius (m)`,
      `Total height (m)`,
      `Lower crown height (m)`,
      `Crown condition (%)`),
    `Bleed prevalence (%)` = bleedPrevalence(`Active bleed length (mm)`,
      `Active bleeds`,
      `Black staining length (mm)`,
      `Black staining`,
      `Diameter at breast height (m)`),
    `Agrilus exit hole density (m^-2)` = agrilusExitHoleDensity(`Agrilus exit holes`,
      `Diameter at breast height (m)`)
  )

t <- makeAnalysisTable(d)

## Generate random forest models
m <- rf(t,cls = NULL,nreps = 10)

```

siteAdjustment

Site adjustment

Description

Perform a site adjustment of selected descriptors.

Usage

```
siteAdjustment(
  phenoData,
  descriptors = c("Diameter at breast height (m)", "Lower crown height (m)",
    "Timber height (m)", "Total height (m)", "Crown radius (m)")
)
```

Arguments

phenoData phenoData tibble containing phenotype data
 descriptors columns of phenoData on which to perform site correction

Examples

```
library(dplyr)

## Retrieve file paths for example data
files <- list.files(system.file('phenotypeDataCollectionSheets',
  package = 'pdi'),full.names = TRUE)

## Prepare data
d <- map(files,readPhenotypeSheet) %>%
  map(preparePhenotypeData) %>%
  bind_rows() %>%
  siteAdjustment()
```

siteAdjustmentFactors *Site adjustment factors*

Description

Return site adjustment factors of selected phenotypic descriptors.

Usage

```
siteAdjustmentFactors(
  phenoData,
  descriptors = c("Diameter at breast height (m)", "Lower crown height (m)",
    "Timber height (m)", "Total height (m)", "Crown radius (m)")
)
```

Arguments

phenoData phenoData tibble containing phenotype data
 descriptors columns of phenoData on which calculate site correction factors

Examples

```
library(dplyr)

## Retrieve file paths for example data
files <- list.files(system.file('phenotypeDataCollectionSheets',
  package = 'pdi'),full.names = TRUE)

## Prepare data
d <- map(files,readPhenotypeSheet) %>%
  map(preparePhenotypeData) %>%
  bind_rows() %>%
  siteAdjustment()

sa_factors <- siteAdjustmentFactors(d)
```

Index

agrilusExitHoleDensity, [2](#)

bleedPrevalence, [3](#)

calcDIs, [3](#)

crownCondition, [4](#)

crownProductionEfficiency, [5](#)

crownSurfaceArea, [5](#)

crownVolume, [6](#)

descriptorContributions, [7](#)

liveCrownRatio, [8](#)

makeAnalysisTable, [8](#)

mds, [9](#)

minMaxScale, [10](#)

phenotypingTemplate, [11](#)

preparePhenotypeData, [11](#)

readPhenotypeSheet, [12](#)

rf, [12](#)

siteAdjustment, [13](#)

siteAdjustmentFactors, [14](#)