Package 'pvcurveanalysis'

October 14, 2022

Type Package

Title Analysis of Pressure Volume Curves

Version 1.0.0

Depends ggplot2, R (>= 3.5.0)

Suggests knitr, rmarkdown

VignetteBuilder knitr

Description Enables the manufacturing, analysis and display of pressure volume curves. From the progression of the curves, turgor loss point, osmotic potential and apoplastic fraction can be derived. Methods adapted from Bartlett, Scoffoni and Sack (2012) <doi:10.1111/j.1461-0248.2012.01751.x>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Anna Raesch [aut, cre, cph]

Maintainer Anna Raesch <annaraesch@gmail.com>

Repository CRAN

Date/Publication 2021-10-13 10:00:02 UTC

R topics documented:

`ApplyCombMod` *Apply a combined exponential and linear model*

Description

a non linear model combining an exponential and a linear fit is applied to the data using the Gauss-Newton algorithm of nls. starting values are calculated based on the data. Weights are applied to the model based on the estimated insecurity of the data quality.

Usage

```
ApplyCombMod(data, y = "y", x = "x")
```

Arguments

`data`	data frame containg x and y data to which the model is ought to be applied to
`y`	name of column in data containing y data
`x`	name of column in data containing x data

Value

model parameters

`ExtractFitParam` *Extracts the fitting parameters from results list*

Description

Extracts the coefficients and confidence intervals from the fitting results of the functions analysing the pressure volume curve (TurgorLossPoint, OsmoticPot and ModElasticity)

Usage

```
ExtractFitParam(result_list)
```

Arguments

`result_list`	output list from the functions TurgorLossPoint, OsmoticPot or ModElasticity

Value

data frame containing the coefficients and the 0.95 confidence interval of the coefficients from the fit

---

`ExtractParam` *Extracts parameters from result list*

Description

Extracts the curve parameters from the result lists of the functions analysing the pressure volume curve (TurgorLossPoint, OsmoticPot and ModElasticity

Usage

```
ExtractParam(result_list)
```

Arguments

`result_list`	output list from the functions TurgorLossPoint, OsmoticPot or ModElasticity

Value

data frame containing the results from the curve analysis only, depending on the function used, relative water deficit at turgor loss point (rwd.tlp), water potential at turgor loss point (water.pot.tlp), apoplastic fraction (apo.fract), osmotic potential at full saturation (osmotic.pot.full.sat), modulus of elasticity (modulus.elasticity)

Examples

```
# use pressure volume data provided by package
pv_data <- pressure_volume_data

# do pressure volume curve analysis
pv_data <- RelativeWaterDeficit(pv_data)
results <- OsmoticPot(pv_data, graph = FALSE)

# extract curve values
ExtractParam(results)
```

`FMSaturated` *Saturated fresh mass estimation*

Description

Calculates saturated fresh mass by fitting fresh mass values above the turgor loss point linearly to water potential values.

Usage

```
FMSaturated(
  data,
  sample = "sample",
  water.potential = "water.potential",
  fresh.mass = "fresh.mass",
  dry.mass = "dry.mass"
)
```

Arguments

`data`	data frame, at least with a column containing numeric water potential (MPa), fresh.mass (g) and dry.mass (g) values, ordered by sample by descending water potential. A column containing the sample IDs is optionally required if several samples were measured.
`sample`	optional name of the column in data containing the sample IDs, default: "sample"
`water.potential`	optional name of the column in data containing the water potential values (MPa), default = "water.potential"
`fresh.mass`	optional name of the column in data containing the fresh mass values (g), default: "fresh mass"
`dry.mass`	optional name of the column containing the dry mass values (g), default: "dry.mass"

Details

Above the turgor loss point, a linear relationship between water content and water potential exists. Based on this premise, saturated water content is found where water potential is zero. First, turgor loss point is calculated based on the relative leaf water loss (fresh mass minus dry mass relativized by the maximum leaf water content value). Then, data above the turgor loss point is extracted and fresh mass is fitted linearly to water potential. The point where water potential of the linear regression line is zero is the saturated water content.

Before using this function, check the data for an initial plateau. Data points in the initial part of the water potential versus RWD plot with a stronger then expected decline need to be omitted.

Value

the original data frame (data) extended by a numeric column containing the saturated fresh mass values ("fresh.mass.saturated")

Examples

```
# get example data
df <- pressure_volume_data
# extend the data frame by saturated fresh mass
df <- FMSaturated(df)
```

`MergeDf` *Merges data to data frames*

Description

merges data frames containing all neccessary informations for plotting with PlotOutput()

Usage

```
MergeDf(
  x,
  y,
  y2 = FALSE,
  y3 = FALSE,
  legend,
  legend.y2 = FALSE,
  legend.y3 = FALSE
)
```

Arguments

`x`	vector containing the x values
`y`	vector containing the y values
`y2`	optional vector containing the values for the second y coordinates
`y3`	optional vector containing the values for the third y coordinates
`legend`	name of the y values in the legend
`legend.y2`	optional name of the second y values in the legend
`legend.y3`	optional name of the third y values in the legend

Value

data frame with columns containing all above information in equalized length as requested by gglot

ModElasticity *Modulus of elasticity*

Description

Determines pressure potential and the modulus of elasticity

Usage

```
ModElasticity(
  data,
  sample = "sample",
  water.potential = "water.potential",
  RWD = "RWD",
  graph = TRUE,
  show.legend = TRUE
)
```

Arguments

data	data frame containing columns of equal lengths giving the numerical coordinates of the curve: water potential (MPa) and RWD (%), ordered by sample by descending water potential. A column containing the sample IDs is optionally required if several samples were measured
sample	optional column name in data containing the sample ID, default: "sample"
water.potential	optional column name in data containing the water potential values of the leaf (MPa), default: "water.potential"
RWD	optional column name in data containing the relative water deficit values (%), default: "RWD"
graph	set FALSE if no plots are to be returned
show.legend	set FALSE if no legend is to be shown in the plots

Details

Relative water deficit at turgor loss point is determined via the function TurgorLossPoint() and osmotic potential is calculated via the function OsmoticPot().

Pressure potential is derived by subtracting osmotic potential from water potential. The part of the pressure potential prior the turgor loss point is then fitted linearly and after transformation of RWD (slope of the fitted line.

Before using this function, check the data for an initial plateau. Data points in the initial part of the water potential versus RWD plot with a stronger then expected decline need to be omitted.

Value

List splitted by sample consisting of

`modulus.elasticity`
: modulus of elasticity (MPa)

`formula`
: formula of the transformed linear osmotic potential fit (1/-MPa) and the pressure potential (MPa) fit

`coef`
: coefficients of the osmotic (1/-MPa) and pressure potential (MPa) fit

`conf_int`
: upper (97.5 %) and lower (2.5 %) border of 95 % confidence interval of model parameters

If graph = TRUE, the original data is displayed with the x- and y-axis intercepts of the turgor loss point, the osmotic potential fit and the linear regression line of the pressure potential.

Examples

```
#get example data, calculate Relative Water Deficit
data <- RelativeWaterDeficit(pressure_volume_data)[pressure_volume_data$sample == 10, ]

# determine modulus of elasticity and the fitting parameters. Do not plot results.
m_elasticity <- ModElasticity(data, graph = FALSE)
```

OrderCheck *Order Check*

Description

Checks for the correct ordering of the data: increasing for date.and.time and time.since start, decreasing for fresh.mass and water.potential. Done separatly for each sample. An individualized warning is printed if not ordered correctly.

Usage

```
OrderCheck(data, sample = FALSE, fresh.mass = FALSE, water.potential = FALSE)
```

Arguments

`data`
: data frame containing the data to be checked

`sample`
: name of the column containing the sample IDs, if present in data

`fresh.mass`
: name of the column containing the numeric fresh mass values, if present in data

`water.potential`
: name of the column containing the numeric water potential values, if present in data

Value

no return value

OsmoticPot *Pressure Volume Curve Analysis*

Description

Determines the coordinates of the turgor loss point, osmotic potential at full hydration and apoplastic fraction

Usage

```
OsmoticPot(
  data,
  sample = "sample",
  water.potential = "water.potential",
  RWD = "RWD",
  graph = TRUE,
  show.legend = TRUE
)
```

Arguments

data	data frame containing columns of equal lengths giving the numerical coordinates of the curve: water potential (MPa) and RWD (%), ordered by sample by descending water potential. A column containing the sample IDs is optionally required if several samples were measured.
sample	optional column name in data containing the sample ID, default: "sample"
water.potential	optional column name in data containing the numeric water potential values (MPa), default: "water.potential"
RWD	optional column name in data containing the relative water deficit values (%), default: "RWD"
graph	set FALSE if no plots are to be returned
show.legend	set FALSE if no legend is to be shown in the plots

Details

RWD at turgor loss point is derived by the function TurgorLossPoint().

The pressure-volume curve data is converted to -1/MPa. The osmotic potential is then derived by fitting a linear regression line with the Gauss-Newton algorithm of nls() to the water potential data following the turgor loss point. The y- and x-axis intercept of the regression line gives the osmotic potential at full hydration (op.full.sat) and the RWD at zero 1/-Psi, respectively. RWD at zero1/-Psi is then transferred to RWC at zero 1/-Psi to derive apoplastic fraction (apo.fract). The turgor loss point equals the value of the osmotic potential fit at the relative water deficit at turgor loss point.

Before using this function, check the data for an initial plateau. Data points in the initial part of the water potential versus RWD plot with a stronger then expected decline need to be omitted.

Value

List splitted by sample consisting of

`turgor.loss.point`
: x and y coordinates of the turgor loss point (RWD (%) and water.potential (MPa), respectively)

`osmotic.potential`
: x and y intercepts of the osmotic potential fit (apoplasic fraction (apo.fract) (%) and op.full.sat (MPa), respectively)

`formula`
: formula of the linear osmotic potential fit

`coef`
: coefficients of the linear model

`conf_int`
: upper (97.5 %) and lower (2.5 %) border of 95 % confidence interval of model parameters

If graph = TRUE, the plotted tranformed data is displayed with the x- and y-axis intercepts of the turgor loss point and the linear regression line of the osmotic potential showing the point of y-intercept (op.full.sat) and x-intercept (apo.fract).
Before using this function, check the raw data for an initial plateau. If the exponential decline does not onset directly, fitting might not succeed.

Examples

```
# get example data, calculate Relative Water Deficit
data <- RelativeWaterDeficit(pressure_volume_data)[pressure_volume_data$sample == 10, ]

# calculate pressure volume curve characteristics and plot graphs
pv_analysis <- OsmoticPot(data)
```

PlotOutput *Plot Output*

Description

plots the data as specified

Usage

```
PlotOutput(
  sub.sample,
  x,
  y,
  y2 = FALSE,
  y3 = FALSE,
  legend.y,
  legend.y2 = FALSE,
  legend.y3 = FALSE,
```

```
  x.axis,
  y.axis,
  x.intercept = FALSE,
  y.intercept = FALSE,
  legend.x.intercept = FALSE,
  line.x,
  line.y,
  line.y2 = FALSE,
  line.y3 = FALSE,
  legend.line.y,
  legend.line.y2 = FALSE,
  legend.line.y3 = FALSE,
  show.legend = show.legend
)
```

Arguments

sub.sample	sample ID
x	vector containing the x coordinates of the data to be plotted as points
y	vector containing the y coodinates of the data to be plotted as points
y2	optional vector containing the second y coordinates of the data to be plotted as points
y3	optional vector containing the third y coordinates of the data to be plotted as points
legend.y	string, name of data points to be printed in the legend
legend.y2	string, optional name of second set of data points to be printed in the legend
legend.y3	string, optional name of third set of data points to be printed in the legend
x.axis	string, label of x axis
y.axis	sring, label of y axis
x.intercept	vector containg the x coordinate of the intercept
y.intercept, legend.x.intercept	optional vector containg the y coordinate of the intercept string, name of x.intercept to be printed in the legend
line.x	vector containing the x coordinate for the lines
line.y	vector containing the y coordinates for the line
line.y2	vector containing the y coordinates for the second line
line.y3	vector containing the y coordinates for the second line
legend.line.y	string, name of line to be printed in the legend
legend.line.y2	string, name of second line to be printed in the legend
legend.line.y3	string, name of third line to be printed in the legend
show.legend	boolean, specifies whether a legend is to be printed

Value

graphic

pressure_volume_data *Pressure volume curve data*

Description

A dataset containing water potential and fresh mass measurements of repeatedly measured drying kohlrabi leaves subjected to different soil moisture conditions during their growth (n = 6) and their saturated fresh mass and dry mass.

Usage

```
pressure_volume_data
```

Format

A data frame with 160 rows and 8 variables

Details

- date: Date of measurement
- treatment: Soil moisture conditions during the last 6 days of Kohlrabi growth (10-30
- sample: Sample ID (1 - 12)
- fresh.mass.harvest: Fresh mass measured at harvest (12 h prior measurement of fresh.mass.saturated) (2.9813 - 7.1557)
- fresh.mass.saturated: Saturated fresh mass of the leaf in gramms (4.1276 - 7.0867)
- fresh.mass: Fresh mass of the leaf in gramms (2.7215 - 6.8246)
- dry.mass: Dry mass of the leaf in gramms (0.2937 - 0.7267)
- water.potential: Water potential of the leaf in MPa (-1.62 - -0.24)

RelativeWaterContent *Relative Water Content (RWC)*

Description

Calculates relative water content (RWC, %)

Usage

```
RelativeWaterContent(
  data,
  fresh.mass = "fresh.mass",
  dry.mass = "dry.mass",
  fresh.mass.saturated = "fresh.mass.saturated"
)
```

Arguments

`data`	data frame with columns of equal length containing at least columns with the fresh mass (g), the dry mass (g) and the saturated fresh mass (g)
`fresh.mass`	optional name of the column in data containing the numeric fresh mass values (g); default: "fresh.mass"
`dry.mass`	optional name of the column in data containing the numeric dry mass values (g); default: "dry.mass"
`fresh.mass.saturated`	optional name of the column in data containing the numeric saturated fresh mass values (g); default: "fresh.mass.saturated"

Details

Relative water content (%) is calculated as:

$$RWC = 100 * ((FM - DM)(FMs - DM)^{-}1)$$

whereas FM = fresh mass, DM = dry mass and FMs = fresh mass at water saturation.

Value

the original data frame extended by a numeric column with the relative water content (RWC) (%).

Examples

```
# get example data
df <- pressure_volume_data

# extend df by RWC
df_with_RWC <- RelativeWaterContent(df)
```

`RelativeWaterDeficit` *Relative Water Deficit (RWD)*

Description

Calculates relative water deficit (%)

Usage

```
RelativeWaterDeficit(
  data,
  fresh.mass = "fresh.mass",
  dry.mass = "dry.mass",
  fresh.mass.saturated = "fresh.mass.saturated"
)
```

Arguments

`data`	data frame with columns of equal length containing at least columns with the fresh mass (g), the dry mass (g) and the saturated fresh mass (g)
`fresh.mass`	optional name of the column in data containing the numeric fresh mass values (g); default: fresh.mass
`dry.mass`	optional name of the column in data containing the numeric dry mass values (g); default: dry.mass
`fresh.mass.saturated`	optional name of the column in data containing the numeric saturated fresh mass values (g); default: fresh.mass.saturated

Details

Relative water deficit (%) is calculated as:

$$RWD = 100 - 100 * ((FM - DM)(FMs - DM)^{-}1)$$

whereas FM = fresh mass, DM = dry mass and FMs = fresh mass at water saturation.

Value

the original data frame extended by a numeric column with the relative water deficit (RWD) (%).

Examples

```
# get example data
df <- pressure_volume_data

# extend df by RWD
df_with_RWD <- RelativeWaterDeficit(df)
```

TurgorLossPoint *Turgor Loss Point*

Description

Determines the x coordinate (RWD) of the turgor loss point in a set of experimentally obtained pressure volume curves.

Usage

```
TurgorLossPoint(
  data,
  sample = "sample",
  water.potential = "water.potential",
  RWD = "RWD",
  graph = TRUE,
  show.legend = TRUE
)
```

Arguments

data	data frame containing columns of equal lengths giving at least the numerical coordinates of the curve: water potential (MPa) and RWD (%), ordered by sample by descending water potential. A column containing the sample IDs is optionally required if several samples were measured.
sample	optional name of the column in data containing the sample ID, default: "sample"
water.potential	optional name of the column in data containing the numeric water potential values (MPa), default: "water.potential"
RWD	optional name of the column in data containing numeric relative water deficit values (%), default: "RWD"
graph	set FALSE if no plots are to be returned
show.legend	set FALSE if no legend is to be shown in the plots

Details

Before using this function, check the data for an initial plateau. Data points in the initial part of the water potential versus RWD plot with a stronger then expected decline need to be omitted.

The data is fitted using the Gauss-Newton algorithm of nls() to a combined exponential and linear model. The exponential and linear parts are extracted and RWD at turgor loss point is localized at their point of minimum distance.

Value

List splitted by sample consisting of

turgor.loss.point	coordinates of the turgor loss point (RWD)
formula	formula of the exponential and linear part of the combined fits
coef	coefficients of combined model
conf_int	upper (97.5 %) and lower (2.5 %) border of 95 % confidence interval of model parameters

If graph = TRUE, the plotted original data is displayed with the exponential and linear fit of the combined model as well as the x-coordinate (RWD) of the turgor loss point.

Examples

```
# get sample data
data <- RelativeWaterDeficit(pressure_volume_data)[pressure_volume_data$sample == 10, ]

# identify turgor loss point in curve
turgor_loss_point <- TurgorLossPoint(data)
```

`ValidityCheck` *Ensures the validity of the input data*

Description

Ensures the validity of the input data

Usage

```
ValidityCheck(
  data,
  sample = FALSE,
  dry.mass = FALSE,
  fresh.mass.saturated = FALSE,
  fresh.mass = FALSE,
  water.potential = FALSE,
  RWD = FALSE
)
```

Arguments

`data`	data frame containing the data to be checked
`sample`	name of column containing the sample ID (default: sample)
`dry.mass`	name of column containing the dry mass (g) (default: dry mass)
`fresh.mass.saturated`	name of column containing the saturated fresh mass (g) (default: fresh.mass.saturated)
`fresh.mass`	name of column containing the fresh mass (g) (default: fresh.mass)
`water.potential`	name of column containing the water potential (MPa) (default: water.potential)
`RWD`	name of column containing the relative water deficit (default: RWD)

Value

no return value

`ValidityCheckDetail` *Checks if column exists in data, is numeric and has the same lenghts as the others existence*

Description

Checks if column exists in data, is numeric and has the same lenghts as the others existence

Usage

```
ValidityCheckDetail(data_in, value)
```

Arguments

data_in	data frame to be checked
value	column in data

Value

no return value

Index