Package: fitplc (via r-universe)

October 30, 2026

Type Package

Title Fit Hydraulic Vulnerability Curves

Version 1.3

Maintainer Remko Duursma <remkoduursma@gmail.com>

Description Fits Weibull or sigmoidal models to percent loss conductivity (plc) curves as a function of plant water potential, computes confidence intervals of parameter estimates and predictions with bootstrap or parametric methods, and provides convenient plotting methods.

License GPL

URL https://www.bitbucket.org/remkoduursma/fitplc

BugReports https://www.bitbucket.org/remkoduursma/fitplc/issues

Depends nlme, car

Suggests testthat

LazyData yes

RoxygenNote 6.1.1

Config/pak/sysreqs cmake make libicu-dev

Repository https://remkoduursma.r-universe.dev

Date/Publication 2019-01-21 11:31:28 UTC

RemoteUrl https://bitbucket.org/remkoduursma/fitplc

RemoteRef HEAD

RemoteSha aabfe69fadb9a2236bc9400da299a7e55a948358

Contents

fitcond 2
fsigmoidal 6
fweibull 7
getPx 7
plot.plcfit 9
stemvul 10

Index 12

`fitcond` *Fit a PLC curve*

Description

Fit a curve to measurements of stem or leaf conductivity at various water potentials. If measurements are organized as 'percent loss conductivity' (PLC), use the `fitplc` function. If they are organized as the actual conductance or conductivity (as is common for leaf hydraulic conductance data, for example), use the `fitcond` function. You can choose to either fit the Weibull function (the default), or the sigmoidal-exponential model. See Details and Examples for more information on how to use these functions.

It is also possible to fit multiple curves at once, for example one for each species or site, with the `fitplcs` and `fitconds` functions. This is useful when you have data for multiple curves organized in one file.

Random effects may be incorporated via the `random` argument (see Examples), in which case `nlme` will be used (in case of the Weibull), or `lme` (in case of the sigmoidal model).

See `plot.plcfit` for documentation on plotting methods for the fitted objects, and the examples below.

Usage

```
fitcond(dfr, varnames = c(K = "K", WP = "MPa"), Kmax = NULL,
  WP_Kmax = NULL, rescale_Px = FALSE, ...)

fitplc(dfr, varnames = c(PLC = "PLC", WP = "MPa"), weights = NULL,
  random = NULL, model = c("Weibull", "sigmoidal", "loess",
  "nls_sigmoidal"), x = 50, coverage = 0.95, bootci = TRUE,
  nboot = 999, quiet = TRUE, startvalues = NULL,
  shift_zero_min = FALSE, loess_span = 0.7, msMaxIter = 1000, ...)

fitplcs(dfr, group, ...)

fitconds(dfr, group, ...)
```

Arguments

`dfr`	A dataframe that contains water potential and plc or conductivity/conductance data.
`varnames`	A vector specifying the names of the PLC and water potential data (see Examples).
`Kmax`	Maximum conduct(ance)(ivity), optional (and only when using `fitcond`). See Examples.
`WP_Kmax`	Water potential above which Kmax will be calculated from the data. Optional (and only when using `fitcond`). See Examples.

rescale_Px	Logical (default FALSE). If TRUE, rescales calculation of Px relative to the fitted value of conductance/PLC at the maximum (least negative) water potential in the dataset. Use this argument only when you know exactly what that means. Identical to rescale_Px argument in getPx.
...	Further parameters passed to fitplc.
weights	A variable used as weights that must be present in the dataframe (unquoted, see examples).
random	Variable that specifies random effects (unquoted; must be present in dfr).
model	Either 'Weibull', 'sigmoidal', 'loess' or 'nls_sigmoidal'. See Details.
x	If the P50 is to be returned, x = 50. Set this value if other points of the PLC curve should be estimated (although probably more robustly done via getPx).
coverage	The coverage of the confidence interval for the parameters (0.95 is the default).
bootci	If TRUE, also computes the bootstrap confidence interval.
nboot	The number of bootstrap replicates used for calculating confidence intervals.
quiet	Logical (default FALSE), if TRUE, don't print any messages.
startvalues	Obsolete - starting values for Weibull now estimated from sigmoidal model fit.
shift_zero_min	Logical (default FALSE). If TRUE, shifts the water potential data so that the highest (least negative) value measured is set to zero. This has consequences for estimation of Kmax, and is only used for fitcond.
loess_span	Only used when model="loess", the span parameter setting the desired degree of smoothness (see loess).
msMaxIter	Maximum iterations for nlminb. Only change when needed.
group	Character; variable in the dataframe that specifies groups. The curve will be fit for every group level.

Details

Parameters - Regardless of the model chosen, the fitplc function estimates PX (water potential at which X at PX, MPa per percent). **Models** - Four different models can be fit with the fitplc function. Two of these additionally have the option to account for a random effect (using the random argument): the Weibull and sigmoidal models.

Weibull The Weibull model is fit as reparameterized by Ogle et al. (2009), using non-linear regression (nls) or a non-linear mixed-effects model if a random effect is present (nlme).

sigmoidal The sigmoidal-exponential model follows the specification by Pammenter and van Willigen (1998) : PLC is log-transformed so a linear fit can be obtained with lm or lme in the presence of a random effect.

loess A non-parametric, local regression smoother (using loess), appropriate when parametric models fit poorly (such as for very linear responses).

nls_sigmoidal Equivalent to the sigmoidal model, except the fit is obtained via non-linear regression (not via linear regression following transformation), which in certain cases can give drastically better fits (as noted on small sample, low variance datasets).

Bootstrap - We recommend, where possible, to use the bootstrapped confidence intervals for inference (use at least ca 1000 resamples). The default is TRUE, and it can only be switched off for the Weibull model (in case speed is warranted). The bootstrap is not applied when a random effect is present.

Confidence intervals - For the Weibull model, the CI based on profiling ('Normal approximation') is always performed, and a non-parametric bootstrap when `bootci=TRUE`. Both are output in `coef`, and the bootstrap CI is used in plotting unless otherwise specified (see `plot.plcfit`). When a random effect is specified (for the Weibull model), the CI is calculated with `intervals.lme`. For the sigmoidal model, PX and SX are functions of parameters of a linearized fit, and we thus always use the bootstrap when no random effect is present (it cannot be switched off). When a random effect is included in the sigmoidal model, we use `deltaMethod` from the `car` package.

Weights - If a variable with the name Weights is present in the dataframe, this variable will be used as the `weights` argument to perform weighted (non-linear) regression. See Examples on how to use this option. **Note:** the use of weights has been tested very little in the context of fitting PLC curves.

Random effects - If the `random` argument specifies a factor variable present in the dataframe, random effects will be estimated both for SX and PX. This affects `coef` as well as the confidence intervals for the fixed effects. For both the Weibull model and the sigmoidal model, only the random intercept terms are estimated (i.e. `random=~1|group`).

Examples

```
# We use the built-in example dataset 'stemvul' in the examples below. See ?stemvul.
# Most examples will fit the Weibull model (the default); try running some of the examples
# with 'model="sigmoidal"' and compare the results.

# 1. Fit one species (or fit all, see next example)
dfr1 <- subset(stemvul, Species =="dpap")

# Fit Weibull model. Store results in object 'pfit'
# 'varnames' specifies the names of the 'PLC' variable in the dataframe,
# and water potential (WP).
# In this example, we use only 50 bootstrap replicates but recommend you set this
# to 1000 or so.
pfit <- fitplc(dfr1, varnames=c(PLC="PLC", WP="MPa"), nboot=50)

# Look at fit
pfit

# Make a standard plot. The default plot is 'relative conductivity',
# (which is 1.0 where PLC = 0). For plotting options, see ?plot.plcfit
plot(pfit)

# Or plot the percent embolism
plot(pfit, what="embol")

# Get the coefficients of the fit.
coef(pfit)

# Repeat for the sigmoidal model
# Note that varnames specification above is the same as the default, so it
```

```
# can be omitted.
pfit2 <- fitplc(dfr1, model="sigmoid")
plot(pfit2)
coef(pfit2)

# 2. Fit all species in the dataset.
# Here we also set the starting values (which is sometimes needed).
# In this example, we use only 50 bootstrap replicates but recommend you set this
# to 1000 or so.
allfit <- fitplcs(stemvul, "Species", varnames=c(PLC="PLC", WP="MPa"), nboot=50)

# 3. Plot the fits.
plot(allfit, onepanel=TRUE, plotci=FALSE, px_ci="none", pxlinecol="dimgrey")

# Coefficients show the estimates and 95% CI (given by 'lower' and 'upper')
# Based on the CI's, species differences can be decided.
coef(allfit)

# 3. Specify Weights. The default variable name is Weights, if present in the dataset
# it will be used for weighted non-linear regression
# In this example, we use only 50 bootstrap replicates but recommend you set this
# to 1000 or so.
dfr1$Weights <- abs(50-dfr1$PLC)^1.2
pfit <- fitplc(dfr1, varnames=c(PLC="PLC", WP="MPa"), weights=Weights, nboot=50)
coef(pfit)

# 4. Fit the Weibull curve directly to the raw conductance data.
# Use this option when you don't want to transform your data to PLC.
# You have two options: specify the 'maximum' conductance yourself (and provide Kmax),
# or set the threshold water potential (Kmax_WP), which is then used to calculate Kmax
# (from the average of the conductance values where WP > Kmax_WP).

# Option 1 : maximum conductivity (i.e. at full hydration) is known, and used as input.
kfit1 <- fitcond(dfr1, varnames=c(K="Cond", WP="MPa"), Kmax=7.2, nboot=50)

# Option 2 : calculate maximum cond. from data where water potential : -0.3 MPa.
# In this example, we use only 50 bootstrap replicates but recommend you set this
# to 1000 or so.
kfit2 <- fitcond(dfr1, varnames=c(K="Cond", WP="MPa"), WP_Kmax = -0.3, nboot=50)
# Use plot(kfit1) as for fitplc, as well as coef() etc.

# Fit multiple conductivity curves at once (bootstrap omitted for speed).
kfits3 <- fitconds(stemvul, "Species", varnames=list(K="Cond", WP="MPa"), WP_Kmax=-0.3, boot=FALSE)
plot(kfits3, onepanel=TRUE, ylim=c(0,12), px_ci="none")

# 5. Random effects.
# This example takes into account the fact that the individual data points for a species are not
# independent, but rather clustered by branch.
fitr <- fitplc(dfr1, random=Branch)

# Visualize the random effects.
plot(fitr, plotrandom=TRUE)
```

`fsigmoidal` *Sigmoidal vulnerability curve*

Description

A sigmoidal-exponential function, which describes the relative conductivity as a function of the plant water potential. The relative conductivity is scaled to be 1 when water potential is zero. This function was used by Pammenter and vander Willigen (1998), but note that this implementation gives the relative conductivity, not the PLC (but relK = 1 - PLC). The slope of relK versus P at the inflection point can be calculated from the shape parameter (a) as slope = -a/4.

Usage

```
fsigmoidal(P, PX, a, X = 50)
```

Arguments

`P`	Water potential (positive-valued MPa)
`PX`	Water potential at X loss of conductivity (positive valued).
`a`	Shape parameter, related to the slope at the inflection point (see Description).
`X`	If 50, PX is the P50.

References

Pammenter, N.W., Willigen, C.V. der, 1998. A mathematical and statistical analysis of the curves illustrating vulnerability of xylem to cavitation. Tree Physiol 18, 589-593. doi:10.1093/treephys/18.8-9.589

Examples

```
curve(fsigmoidal(x, PX=-2, a=5), from=0, to=-5)
curve(fsigmoidal(x, PX=-2, a=2), add=TRUE)

# Comparison to Weibull
curve(fweibull(x, PX=3, SX=40), from=0, to=6)
curve(fsigmoidal(x, PX=3, a=4*(40/100)), add=TRUE, col="red")
```

fweibull *Weibull vulnerability curve*

Description

The Weibull function, as re-parameterized by Ogle et al. (2009), which describes the relative conductivity as a function of the plant water potential. The relative conductivity (relK) is scaled to be 1 when water potential is zero. The slope of relK versus P at the inflection point can be calculated from the shape parameter (SX) as slope = -SX/100.

Usage

```
fweibull(P, SX, PX, X = 50)
```

Arguments

P	Water potential (positive-valued MPa)
SX	Shape parameter
PX	Water potential at X loss of conductivity.
X	If 50, PX is the P50.

References

Ogle, K., Barber, J.J., Willson, C., Thompson, B., 2009. Hierarchical statistical modeling of xylem vulnerability to cavitation. New Phytologist 182, 541-554.

Examples

```
curve(fweibull(x, SX=30, PX=2), from=0, to=5)
```

getPx *Extract Px from fitted objects*

Description

Extract esimates of Px from an object returned by `fitplc`, `fitplcs`, `fitcond` or `fitconds`. This function allows extraction of estimates of P88 or other values when the fit estimated P50 (or other).

With the Weibull model, it appears to be more robust to set `x=50` when fitting the curve, and extracting other points with `getPx`.

See examples for use of this function. Note that the confidence interval is based on the bootstrap resampling performed by `fitplc`. If the bootstrap was not performed durinf the fit (i.e. `boot=FALSE` in `fitplc` or elsewhere), it only returns the fitted values, and not the confidence intervals.

Usage

```
getPx(object, x = 50, coverage = 0.95, rescale_Px = FALSE, ...)

## Default S3 method:
getPx(object, x = 50, coverage = 0.95,
  rescale_Px = FALSE, ...)

## S3 method for class 'manyplcfit'
getPx(object, ...)
```

Arguments

object	Object returned by any of the fitting functions (e.g. fitplc)
x	The x in Px, that is, if P50 should be returned, x=50. Can be a vector, to return multiple points at once.
coverage	The desired coverage of the confidence interval (0.95 is the default).
rescale_Px	Logical (default FALSE). If TRUE, rescales calculation of Px for the sigmoidal model, by finding water potential relative to K at zero water potential (which for the sigmoidal model, is not equal to Kmax). If you fitted fitcond with rescale_Px = TRUE, make sure to set TRUE here as well to be consistent (it is not assumed from the fitted model, yet).
...	Further arguments passed to methods (none yet).

Details

Note that this function does not return a standard error, because the bootstrap confidence interval will be rarely symmetrical. If you like, you can calculate it as the mean of the half CI width (and note it as an 'approximate standard error'). A better approach is to only report the CI and not the SE.

Sometimes the upper CI cannot be calculated and will be reported as NA. This indicates that the upper confidence bound is outside the range of the data, and can therefore not be reliably reported. It is especially common when x is large, say for P88.

Methods (by class)

- default: Calculate Px for a single fitted curve.
- manyplcfit: Calculate Px for many fitted curves.

Examples

```
# A fit
somefit <- fitplc(stemvul, x=50, model="sigmoid")

# Extract P12, P88
# Note NA for upper CI for P88; this is quite common
# and should be interpreted as falling outside the range of the data.
getPx(somefit, x=c(12,88))
```

```
# Extract P88 from multiple fitted curves
fits <- fitplcs(stemvul, "Species", boot=FALSE)
getPx(fits, 88)
```

`plot.plcfit` *Plot a fitted vulnerability curve*

Description

Standard plots of fitted curves (objects returned by `fitplc`, `fitplcs`, `fitcond` or `fitconds`), with plenty of options for customization.

Usage

```
## S3 method for class 'plcfit'
plot(x, xlab = NULL, ylab = NULL, ylim = NULL,
  pch = 19, plotPx = TRUE, plotci = TRUE, plotdata = TRUE,
  plotfit = TRUE, add = FALSE, multiplier = NULL,
  px_ci = c("bootstrap", "parametric", "none"),
  px_ci_type = c("vertical", "horizontal"), px_ci_label = TRUE,
  plotrandom = FALSE, pointcol = "black", linecol = "black",
  linetype = 1, linelwd = 1, linecol2 = "blue", pxlinecol = "red",
  pxcex = 0.7, citype = c("polygon", "lines"), cicol = "#D3D3D3CC",
  what = c("relk", "PLC", "embol"), xaxis = c("positive", "negative"),
  ...)

## S3 method for class 'manyplcfit'
plot(x, what = c("relk", "embol", "PLC"),
  onepanel = FALSE, linecol = NULL, pointcol = NULL, pch = 19,
  legend = TRUE, legendwhere = "topright", ...)
```

Arguments

`x`	A fitted curve returned by `fitplc`
`xlab, ylab`	Optionally, X and Y axis labels (if not provided, a default is used).
`ylim`	Optionally, Y-axis limits.
`pch`	Optionally, the plotting symbol (default = 19, filled circles)
`plotPx`	Logical (default TRUE), whether to plot a vertical line for the P50.
`plotci`	Logical (default TRUE), whether to plot the confidence interval (if computed with bootci=TRUE).
`plotdata`	Logical (default TRUE), whether to add the data to the plot.
`plotfit`	Logical (default TRUE), whether to add the fitted curve to the plot.
`add`	Logical (default FALSE), whether to add the plot to a current device. This is useful to overlay two plots or curves, for example.

multiplier	Multiply the scaled data (for plotting).
px_ci	Option for the confidence interval around Px, either 'parametric' (confidence interval computed with confint), 'bootstrap' (computed with non-parametric bootstrap) or 'none' (no plotting of the confidence interval) (formerly argument was called selines)
px_ci_type	Either 'vertical' (default), or 'horizontal', to plot confidence limits for Px.
px_ci_label	Logical (default TRUE), whether to write a label next to the CI for Px.
plotrandom	Logical. If TRUE (the default is FALSE), plots the predictions for the random effects (only if random effects were included in the model fit).
pointcol	The color(s) of the data points.
linecol	The color(s) of the fitted curve (or color of the random effects curves if plotrandom=TRUE).
linetype	Line type for fitted curve (see options for lty in par).
linelwd	Width of the line (see options for lwd in par).
linecol2	The color of the fixed effects curve (if plotrandom=TRUE; otherwise ignored).
pxlinecol	The color of the lines indicating Px and its confidence interval
pxcex	Character size for the Px label above the Y-axis.
citype	Either 'polygon' (default), or 'lines', specifying formatting of the confidence interval in the plot.
cicol	The color of the confidence interval band (if plotted).
what	Either 'relk' or 'PLC' (or synonym 'embol'); it will plot either relative conductivity or percent loss conductivity (percent embolism).
xaxis	Either 'positive' (default), so that water potential is plotted as positive values, or 'negative', plotting negative-valued water potentials.
...	Further parameters passed to plot, or points (when add=TRUE)
onepanel	For plotting of many curve fits, plot all curves in one panel (TRUE) or in separate panels (FALSE)
legend	(for fitconds and fitplcs only) Logical (default TRUE), whether to include a simple legend when plotting multiple fits
legendwhere	(for fitconds and fitplcs only) As in legend, specification of where to place legend (e.g. 'bottomleft'; coordinates not accepted)

---

stemvul *An example vulnerability curve*

Description

Percent loss conductivity as a function of water potential for three species.

Format

Species One of dpap, egran, ssay

Branch Replicate branch, multiple branches were measured for each species

MPa Xylem water potential (MPa)

PLC Percent loss conductivity

Cond Raw, unscaled conductivity of branch segment (units)

Index

confint, *10*

deltaMethod, *4*

fitcond, 2, *7*, *9*
fitconds, *7*, *9*
fitconds *(*fitcond*)*, 2
fitplc, *7–9*
fitplc *(*fitcond*)*, 2
fitplcs, *7*, *9*
fitplcs *(*fitcond*)*, 2
fsigmoidal, 6
fweibull, 7

getPx, *3*, 7

intervals.lme, *4*

legend, *10*
lm, *3*
lme, *3*
loess, *3*

nlme, *3*
nlminb, *3*
nls, *3*

par, *10*
plot.manyplcfit *(*plot.plcfit*)*, 9
plot.plcfit, *2*, *4*, 9

stemvul, 10