TapeR: Flexible stem taper modelling

Edgar Kublin[*], Johannes Breidenbach[†]

January 15, 2025

1 Introduction

The `TapeR` package implements methods described in (Kublin et al., 2013).

If R is running, the `TapeR` package can be installed by typing

```
> install.packages("TapeR")
```

into the console[1].

The command

```
> library(TapeR)
```

loads the package into the current workspace. We can get an overview of the packages' contents by typing

```
> ?TapeR
```

2 Using the TapeR functions

2.1 Fitting a taper model

A taper model can be fit to measured pairs of diameter and corresponding heights. Typically, those data are recorded for regular stem sections but the data can also be availabe at irregular distances. Important is for the method implemented in this packages is that the stem height is available too. We will load some example data and fit a taper model. Note that in real applications, the dataset must be considerably bigger in order to provide reliable taper models that are valid over larger spatial extents. For example, the data set `SK.par.lme` provides the taper model parameters based on 338 Norway spruce trees in Germany.

```
> #load example data
> data(DxHx.df)
> #prepare the data (could be defined in the function directly)
> Id = DxHx.df[,"Id"]
> x = DxHx.df[,"Hx"]/DxHx.df[,"Ht"]#calculate relative heights
> y = DxHx.df[,"Dx"]
> #plot the example data
> plot(x,y,pch=as.character(Id),xlab="Relative height (m)", ylab="Diameter (cm)")
> #define the relative knot positions and order of splines
>
> knt_x = c(0.0, 0.1, 0.75, 1.0) # B-Spline knots: fix effects
> ord_x = 4 # ord = order (4 = cubic)
```

[*]Forest Research Institute of Baden-Württemberg, Freiburg, Germany

[†]Norwegian Institute of Bioeconomy Research, 1431 Ås, Norway, job@nibio.no

[1]The character ">" is not part of the command. A working Internet connection is required.

```
> knt_z = c(0.0, 0.1, 1.0); ord_z = 4 # B-Spline knots: rnd effects
> #fit the model
> taper.model <- TapeR_FIT_LME.f(Id, x, y, knt_x, ord_x, knt_z, ord_z,
+ IdKOVb = "pdSymm")
> #save model parameters for documentation or dissimination
> #parameters can be load()-ed and used to predict the taper
> #or volume using one or several measured dbh
> spruce.taper.pars <- taper.model$par.lme
> #save(spruce.taper.pars, file="spruce.taper.pars.rdata")##uncomment to save
```
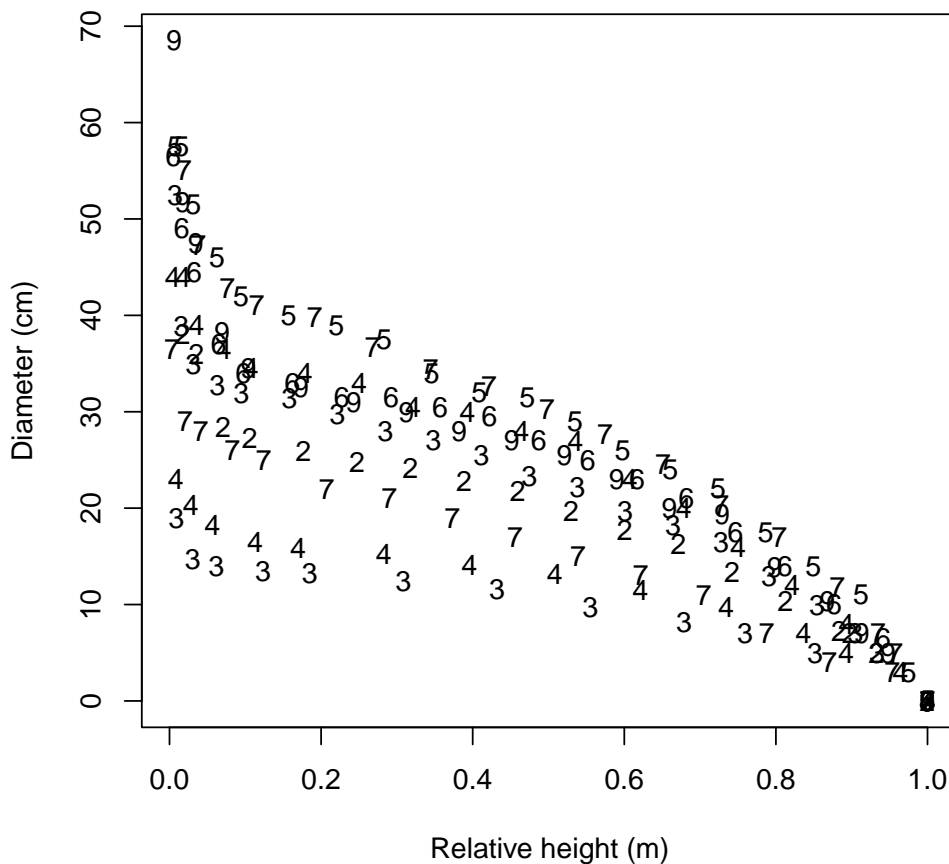

3   Example

This is from the R-help of the exported TapeR-functions: Let's take a look on the taper curve of the first tree in the example data set if it is only calibrated using the diameter measurement in 2m height.

```
> #example data
> data(DxHx.df)
> #taper curve parameters based on all measured trees
> data(SK.par.lme)
> #select data of first tree
> Idi <- (DxHx.df[,"Id"] == unique(DxHx.df$Id)[1])
> tree1 <- DxHx.df[Idi,]
> ## Predict the taper curve based on the diameter measurement in 2 m
> ## height and known height
```
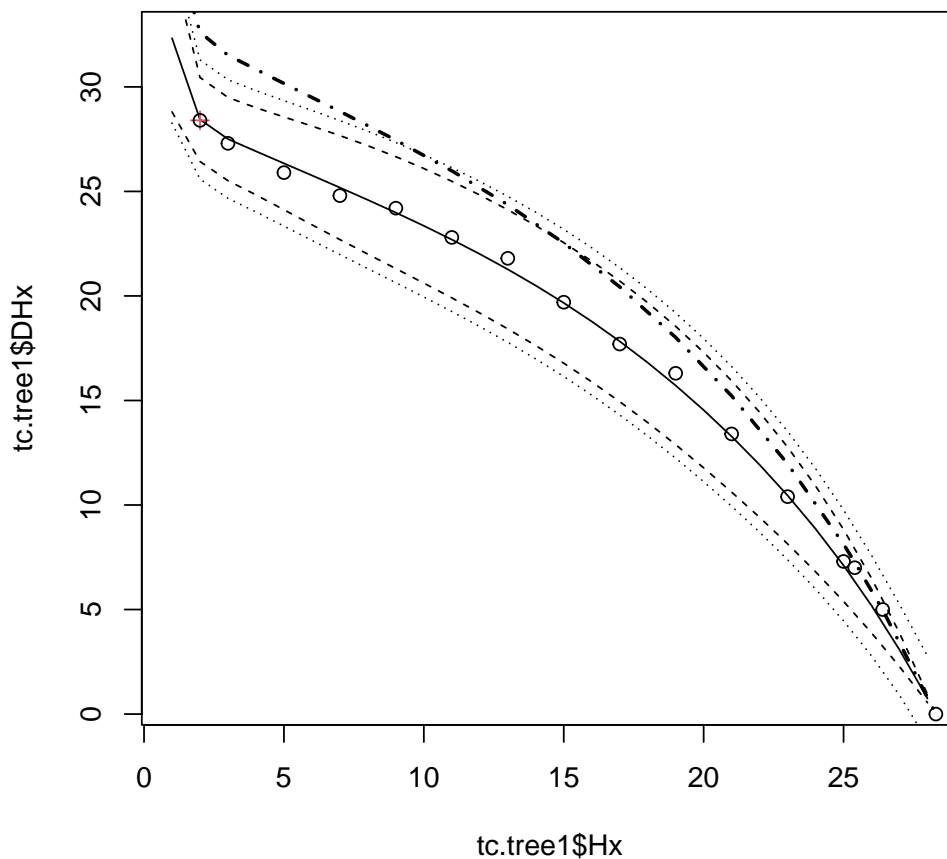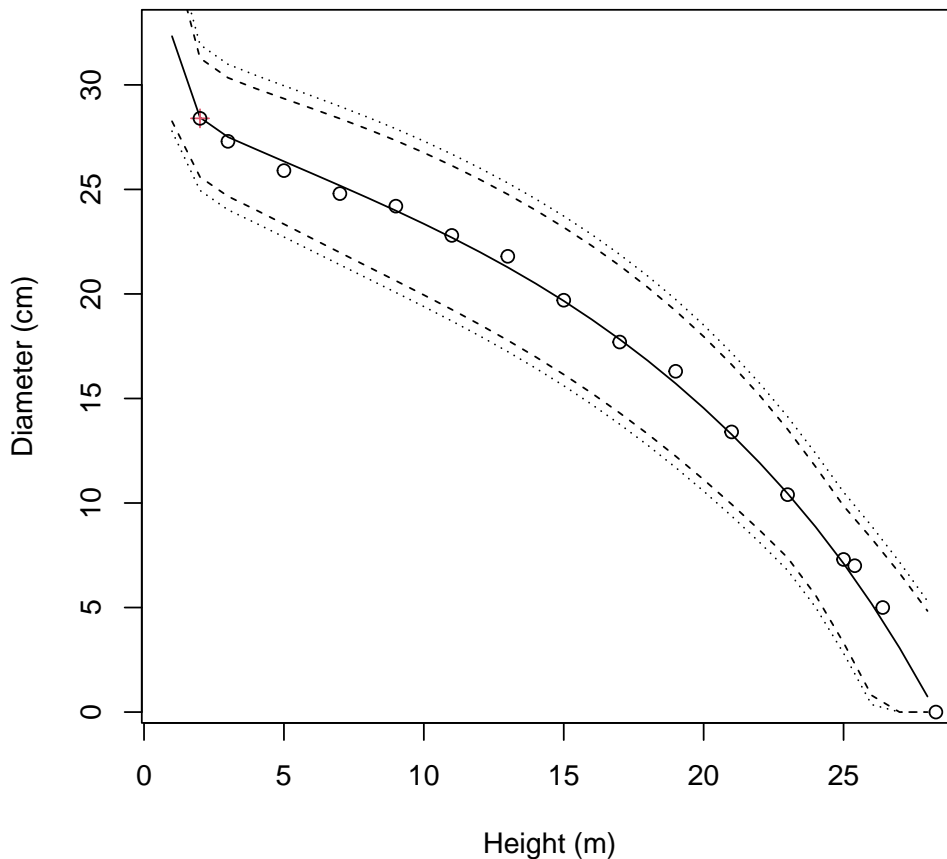
```
> tc.tree1 <- E_DHx_HmDm_HT.f(Hx=1:tree1$Ht[1], Hm=tree1$Hx[3],
+ Dm=tree1$Dx[3], mHt = tree1$Ht[1], sHt = 0, par.lme = SK.par.lme)
> #plot the predicted taper curve
> plot(tc.tree1$Hx, tc.tree1$DHx, type="l")
> #lower CI
> lines(tc.tree1$Hx, tc.tree1$CI_Mean[,1], lty=2)
> #upper CI
> lines(tc.tree1$Hx, tc.tree1$CI_Mean[,3], lty=2)
> #lower prediction interval
> lines(tc.tree1$Hx, tc.tree1$CI_Pred[,1], lty=3)
> #upper prediction interval
> lines(tc.tree1$Hx, tc.tree1$CI_Pred[,3], lty=3)
> #add measured diameter
> points(tree1$Hx[3], tree1$Dx[3], pch=3, col=2)
> #add the observations
> points(tree1$Hx, tree1$Dx)
> ## Add the population average taper curve (without calibration) to the
> ## plot (not of high practical interest but good to know how to get it).
> Ht  = tree1$Ht[1]
> Hx  = tree1$Hx
> #get fixed-effects design matrix for the Splines
> X = TapeR:::XZ_BSPLINE.f(x=Hx/Ht, knt = SK.par.lme$knt_x, ord = SK.par.lme$ord_x)
> #Calculate population average taper curve
> DHx_PA = X %*% SK.par.lme$b_fix
> #add to plot
> lines(tree1$Hx, DHx_PA, lwd=2, lty=4)
```

Let's see how intervals change, if some uncertainty in height is assumed.

```
> ## Let's see how CI's change if there's some uncertainty in the height
> ## measurement
> tc.tree1 <- E_DHx_HmDm_HT.f(Hx=1:tree1$Ht[1], Hm=tree1$Hx[3],
+ Dm=tree1$Dx[3], mHt = tree1$Ht[1], sHt = 1, par.lme = SK.par.lme)
> #plot the predicted taper curve
> plot(tc.tree1$Hx, tc.tree1$DHx, type="l", xlab="Height (m)",
+ ylab="Diameter (cm)")
> #lower CI
> lines(tc.tree1$Hx, tc.tree1$CI_Mean[,1], lty=2)
> #upper CI
> lines(tc.tree1$Hx, tc.tree1$CI_Mean[,3], lty=2)
> #lower prediction interval
> lines(tc.tree1$Hx, tc.tree1$CI_Pred[,1], lty=3)
> #upper prediction interval
> lines(tc.tree1$Hx, tc.tree1$CI_Pred[,3], lty=3)
> #add measured diameter
> points(tree1$Hx[3], tree1$Dx[3], pch=3, col=2)
> #add the observations
> points(tree1$Hx, tree1$Dx)
```

Additionally, exact confidence intervals can be calculated:

```
> plot(tc.tree1$Hx, tc.tree1$DHx, type="l", xlab="Height (m)",
+ ylab="Diameter (cm)")
> ## Calculate "exact" CIs. Careful: This takes a while!
> #library(pracma)# for numerical integration with gaussLegendre()
> tc.tree1.exact <- E_DHx_HmDm_HT_CIdHt.f(Hx=1:tree1$Ht[1],
+ Hm=tree1$Hx[3], Dm=tree1$Dx[3], mHt = tree1$Ht[1], sHt = 1, par.lme =
+ SK.par.lme)
> #add exact confidence intervals to approximate intervals above - fits
> #quite well
> lines(tc.tree1.exact[,1], tc.tree1.exact[,2], lty=2,col=2)
> lines(tc.tree1.exact[,1], tc.tree1.exact[,4], lty=2,col=2)
> ## Calculate the height given a certain diameter threshold, say 8.5 cm
> ht.tree1.d8.5 <- E_HDx_HmDm_HT.f (Dx=8.5, Hm=tree1$Hx[3],
+ Dm=tree1$Dx[3], mHt = tree1$Ht[1], sHt = 1, par.lme = SK.par.lme)
> #add to above created plot
> abline(v=ht.tree1.d8.5)
> ## Calculate the timber volume
> #for the whole stem
> E_VOL_AB_HmDm_HT.f(Hm=tree1$Hx[3], Dm=tree1$Dx[3], mHt = tree1$Ht[1]
+ , sHt = 1, par.lme = SK.par.lme)

$E_VOL
[1] 0.9589025
```

```
$VAR_VOL
[1] 0.0104883

$Hm
[1] 2

$Dm
[1] 28.4

$Ht
[1] 28.3

$Da
          [,1]
[1,] 42.70324

$Db
     [,1]
[1,]    0

$Ha
[1] 0

$Hb
[1] 28.3

$Rfn
$Rfn$fn
[1] "sig2"

> #Calculate the timber volume for a selected section given a height
> #threshold (0.3 - 5 m)
> E_VOL_AB_HmDm_HT.f(Hm=tree1$Hx[3], Dm=tree1$Dx[3], mHt = tree1$Ht[1]
+ , sHt = 1, par.lme = SK.par.lme, A=0.3, B=5, iDH = "H")

$E_VOL
[1] 0.3151162

$VAR_VOL
[1] 0.0006750674

$Hm
[1] 2

$Dm
[1] 28.4

$Ht
[1] 28.3

$Da
          [,1]
[1,] 38.6992
```

```
$Db
          [,1]
[1,] 26.34191


$Ha
[1] 0.3


$Hb
[1] 5


$Rfn
$Rfn$fn
[1] "sig2"


> #Calculate the timber volume for a selected section given a diameter
> #threshold (30 cm - 15 cm) (negative value if A<B)
> E_VOL_AB_HmDm_HT.f(Hm=tree1$Hx[3], Dm=tree1$Dx[3], mHt = tree1$Ht[1]
+ , sHt = 1, par.lme = SK.par.lme, A=30, B=15, iDH = "D")


$E_VOL
[1] 0.7536634


$VAR_VOL
[1] 0.006391593


$Hm
[1] 2


$Dm
[1] 28.4


$Ht
[1] 28.3


$Da
          [,1]
[1,] 29.99978


$Db
          [,1]
[1,] 14.99993


$Ha
[1] 1.446954


$Hb
[1] 19.61795


$Rfn
$Rfn$fn
[1] "sig2"


> #The variance estimate resulting from the tree height uncertainty using
> #a Normal approximation takes much longer...
```

```
> ptm <- proc.time()
> E_VOL_AB_HmDm_HT.f(Hm=tree1$Hx[3], Dm=tree1$Dx[3], mHt = tree1$Ht[1]
+ , sHt = 1, par.lme = SK.par.lme, IA=FALSE)

$E_VOL
[1] 0.9589025

$VAR_VOL
[1] 0.0104883

$Hm
[1] 2

$Dm
[1] 28.4

$Ht
[1] 28.3

$Da
         [,1]
[1,] 42.70324

$Db
     [,1]
[1,]    0

$Ha
[1] 0

$Hb
[1] 28.3

$Rfn
$Rfn$fn
[1] "sig2"

> proc.time() - ptm

      User      System verstrichen
      0.75        0.00        0.75

> #... than the calculation using a 2-point distribution...
> ptm <- proc.time()
> E_VOL_AB_HmDm_HT.f(Hm=tree1$Hx[3], Dm=tree1$Dx[3], mHt = tree1$Ht[1]
+ , sHt = 1, par.lme = SK.par.lme, IA=TRUE)

$E_VOL
[1] 0.9589024

$VAR_VOL
[1] 0.01048643

$Hm
[1] 2
```

```
$Dm
[1] 28.4

$Ht
[1] 28.3

$Da
          [,1]
[1,] 42.70324

$Db
     [,1]
[1,]    0

$Ha
[1] 0

$Hb
[1] 28.3

$Rfn
$Rfn$fn
[1] "sig2"

> proc.time() - ptm

      User      System verstrichen
      0.04        0.00        0.03

> #...fastest if no height variance is assumed
> ptm <- proc.time()
> E_VOL_AB_HmDm_HT.f(Hm=tree1$Hx[3], Dm=tree1$Dx[3], mHt = tree1$Ht[1]
+ , sHt = 0, par.lme = SK.par.lme, IA=FALSE)

$E_VOL
[1] 0.959331

$VAR_VOL
[1] 0.009858254

$Hm
[1] 2

$Dm
[1] 28.4

$Ht
[1] 28.3

$Da
          [,1]
[1,] 42.70324

$Db
```

```
         [,1]
[1,]    0

$Ha
[1] 0

$Hb
[1] 28.3

$Rfn
$Rfn$fn
[1] "sig2"


> proc.time() - ptm
       User      System verstrichen
       0.00        0.02        0.02

> #Also the number of supportive points for the numerical integration
> #influences the calculation time
> ptm <- proc.time()
> E_VOL_AB_HmDm_HT.f(Hm=tree1$Hx[3], Dm=tree1$Dx[3], mHt = tree1$Ht[1]
+ , sHt = 0, par.lme = SK.par.lme, IA=FALSE, nGL=10)

$E_VOL
[1] 0.959331

$VAR_VOL
[1] 0.009858254

$Hm
[1] 2

$Dm
[1] 28.4

$Ht
[1] 28.3

$Da
         [,1]
[1,] 42.70324

$Db
      [,1]
[1,]    0

$Ha
[1] 0

$Hb
[1] 28.3

$Rfn
```
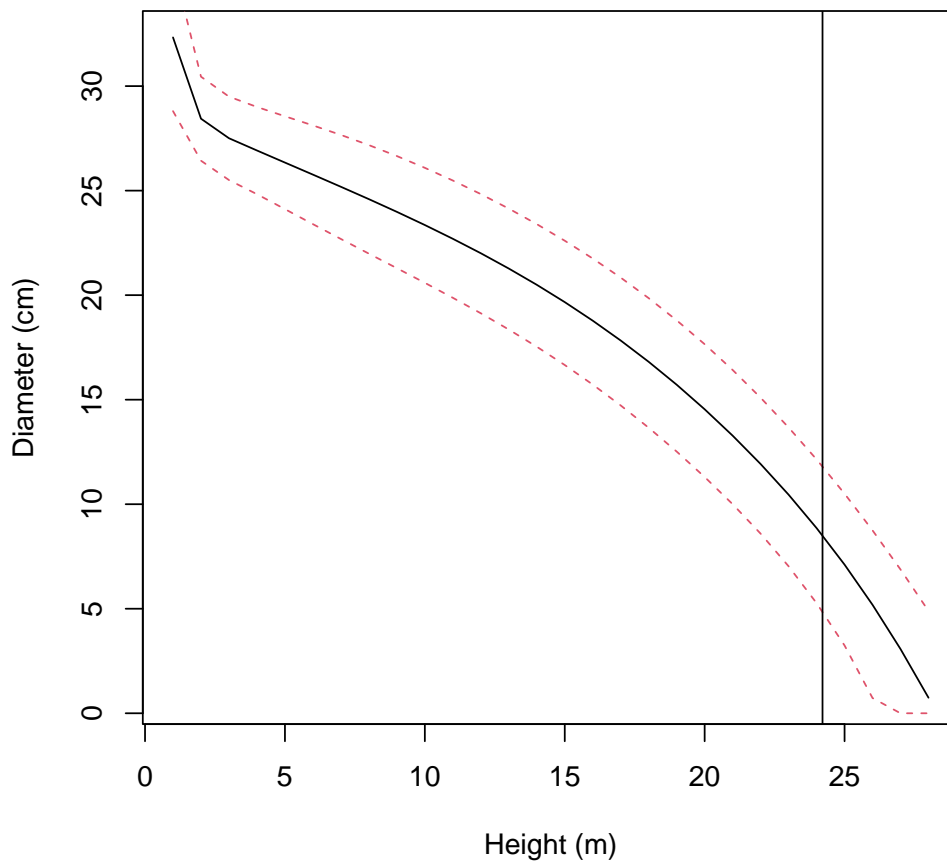
```
$Rfn$fn
[1] "sig2"

> proc.time() - ptm

      User      System verstrichen
      0.02        0.00        0.01

>
```


References

Kublin, E., Breidenbach, J., Kändler, G. (2013): *A flexible stem taper and volume prediction method based on mixed-effects B-spline regression*. Eur J For Res, 132:983-997.