ZooScatR - Calise et al simulations

Calise et al shapes in ZooscatR

ZooScatR can read shape files, which can be used for simulation studies.

The shape used in this tutorial should be part of the ZooScatR package when installed from Github.

Just load ZooScatR:

```
library(ZooScatR)
```

```
## Warning: replacing previous import 'shiny::runExample' by 'shinyjs::runExample'
## when loading 'ZooScatR'
```

```
library(ggplot2)
```

A few shape files are included in ZooScatR by default.

Those shape files are included in the *extdata/profiles* folder of the package. The folder can be found in the installation path of the package. If unsure where this is you can use *system.file(package='ZooScatR')*. On your system the package is located at C:/Users/sven/Documents/R/win-library/4.0/ZooScatR.

In this folder you should ifnd at least 4 shape files:

- *copepod0.dat* - very simple copepod approximation
- *copepod1.dat* - slightly more realistic copepod
- *euphaus0.dat* - very simple sketch of a euphausiid
- *euphaus1.dat* - a bit more realistic euphausiid shape
- *skaret.dat* - a copy of the krill representation found in (Calise and Skaret 2011)

We can check which files are available:
copepod0.dat, copepod1.dat, euphaus0.dat, euphaus1.dat, skaret.dat

If this returns an empty list please update your version of ZooScatR or download the shape files manually from the external data folder on the ZooScatR GitHub page.

Now we can load the configuration file and the shape file we want.
We will start with the standard configuration file and the Skared krill approximation shape.

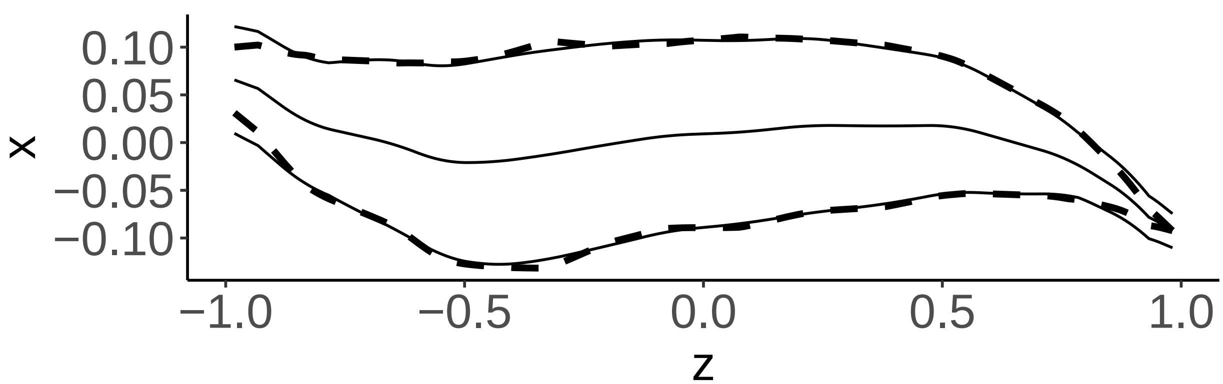
```
#Load the standard confiruation
fname <- paste0(system.file(package="ZooScatR"),"/extdata/configs/config_0.dat") #Loacation of the para
para = read_para(fname) #Read parameters file

#get shape file for skaret krill approximation
profname <- paste0(system.file(package="ZooScatR"),"/extdata/profiles/skaret.dat") #krill example
para$shape$prof_name = profname
```

```r
#set some other settings
para$orient$ang0 = 0 #set start orientation to 0
para$orient$angm = 0 #set mean orientation to 0
para$simu$var0 = 18 #simulate from 10
para$simu$var1 = 300 #...to 300 kHz
para$simu$ni = 200 #resudce the number of elements and frequencies to improve speed
para$simu$n = 283 #use 283 frequency steps

#Create list with soundspeed info
misc <- list(cw=1500)

res <- bscat(para=para, misc=misc, simOut = TRUE) #Target strength vs Frequency
```

Now we can have a look at the results:

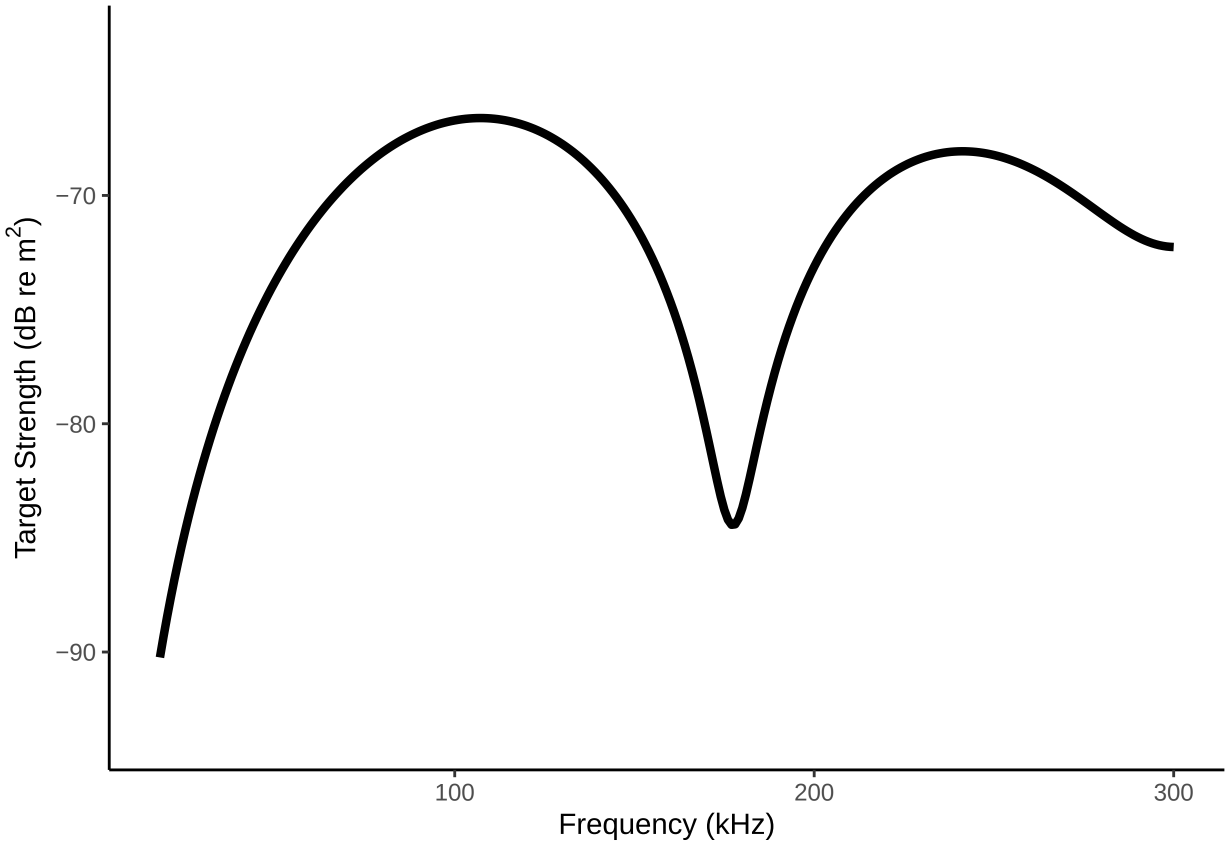
The shape of the simulated target:

```r
print(res$shplot)
```



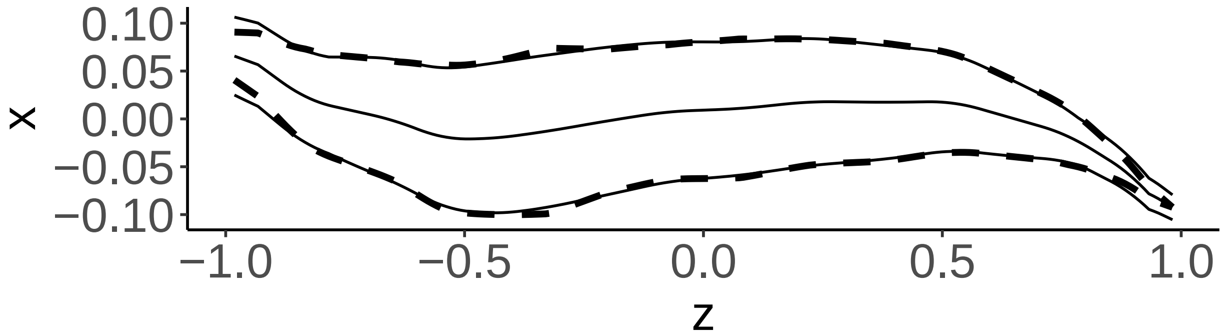
The resulting Target strength vs Frequency:

```r
res$rplot
```

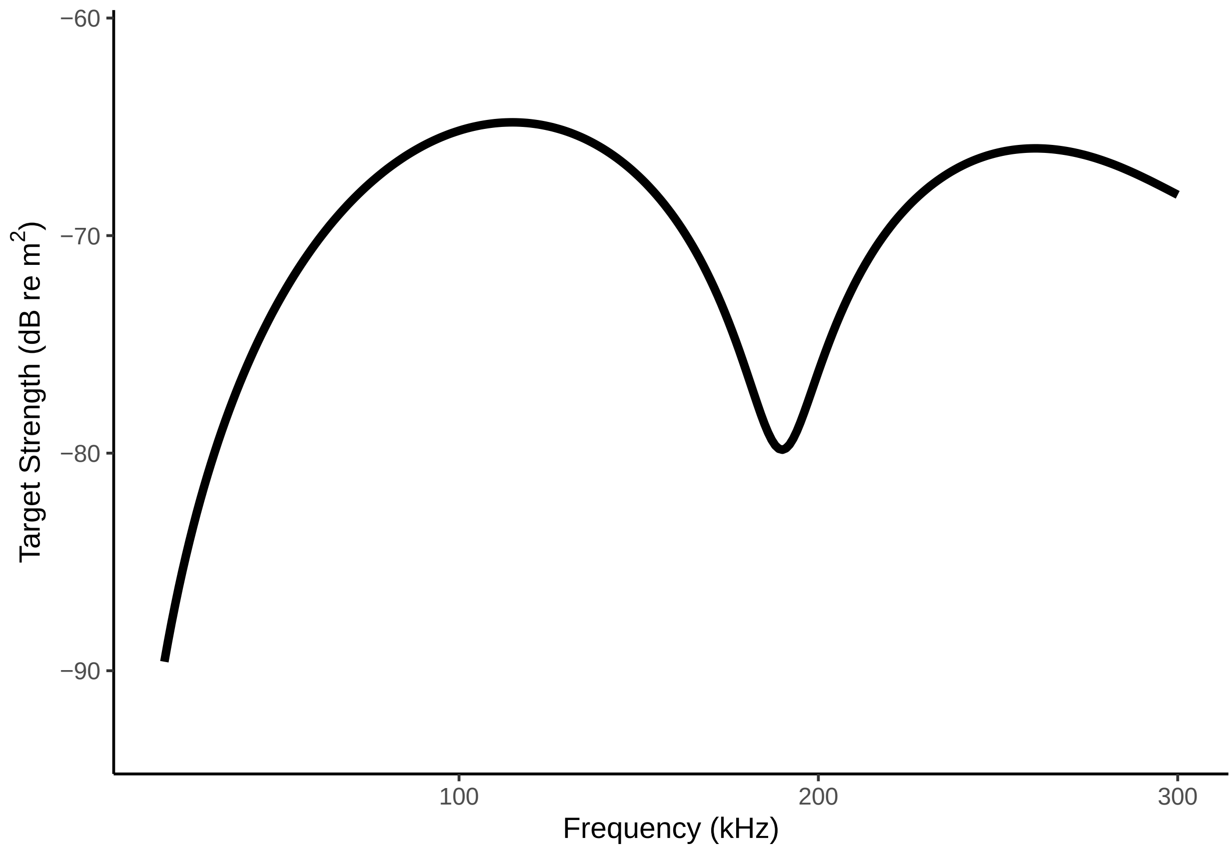


To modify the fat content as described in Calise et al. (2011), we have to change the L/a parameter, which defines the relationship between the central radius and the length of the target. This parameter can be used to modify the width of the target, as achieved with the fat content in Calise et al. (2011)
The *shplot* can be used to visually inspect the resulting shape.
When looking at the shape plots, please note the scale of the x and y axis, which is defaulted to an aspect ratio of 2. v

```
para$shape$L = 38 #set any length, default is 30 mm
para$shape$L_a = 22 #set the desired L/a value, defaults to 16

res1 <- bscat(para=para, misc=misc, simOut = TRUE) #Target strength vs Frequency
res1$shplot
```

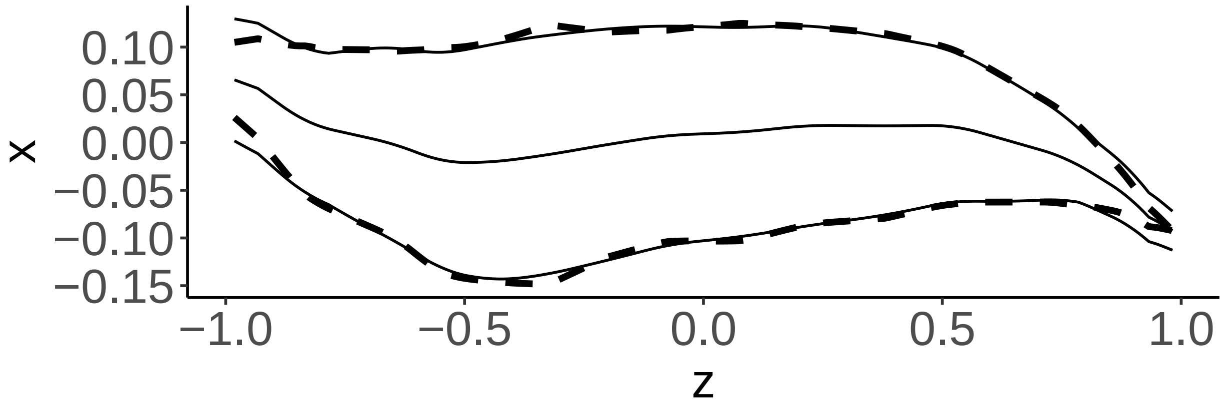



```
res1$rplot
```

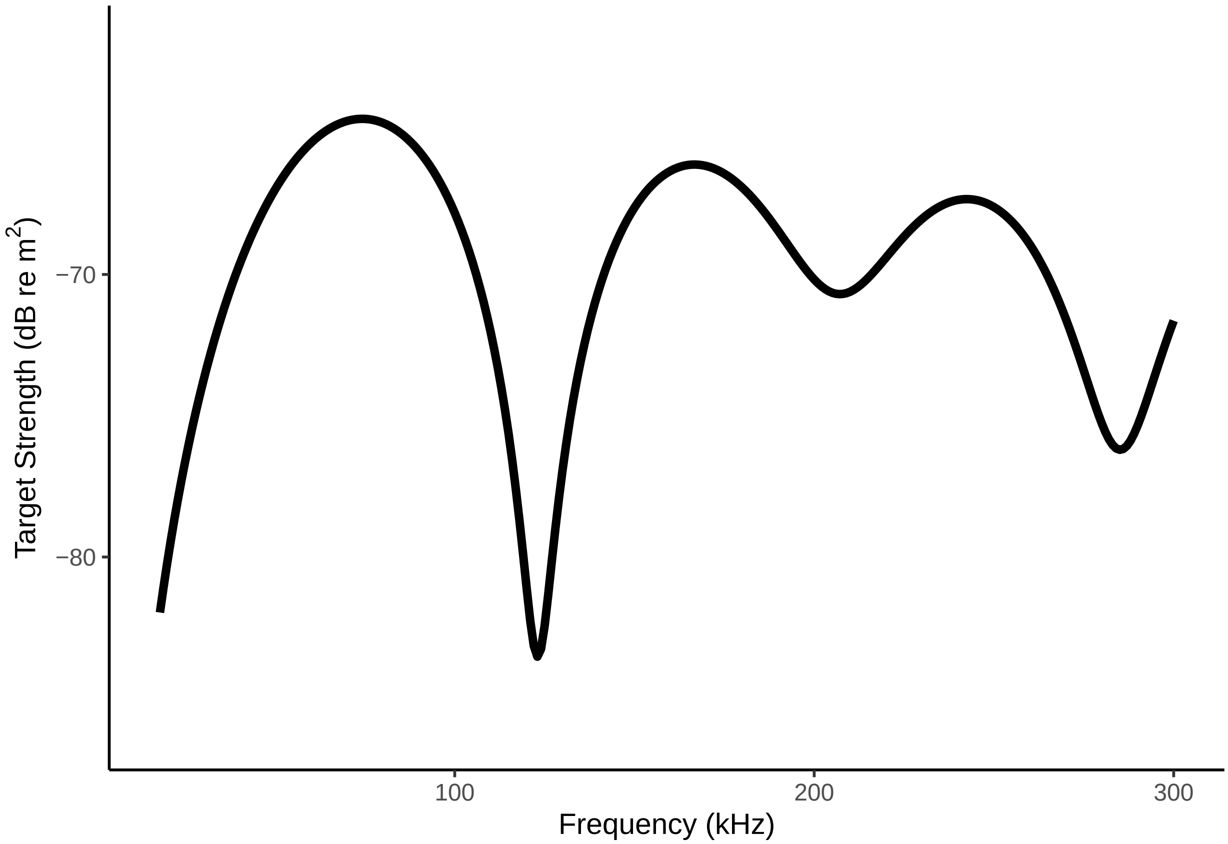


Have alook at another setting:

```
para$shape$L = 38 #set any length, default is 30 mm
para$shape$L_a = 14 #letś create a very fat krill

res2 <- bscat(para=para, misc=misc, simOut = TRUE) #Target strength vs Frequency
res2$shplot
```




```
res2$rplot
```

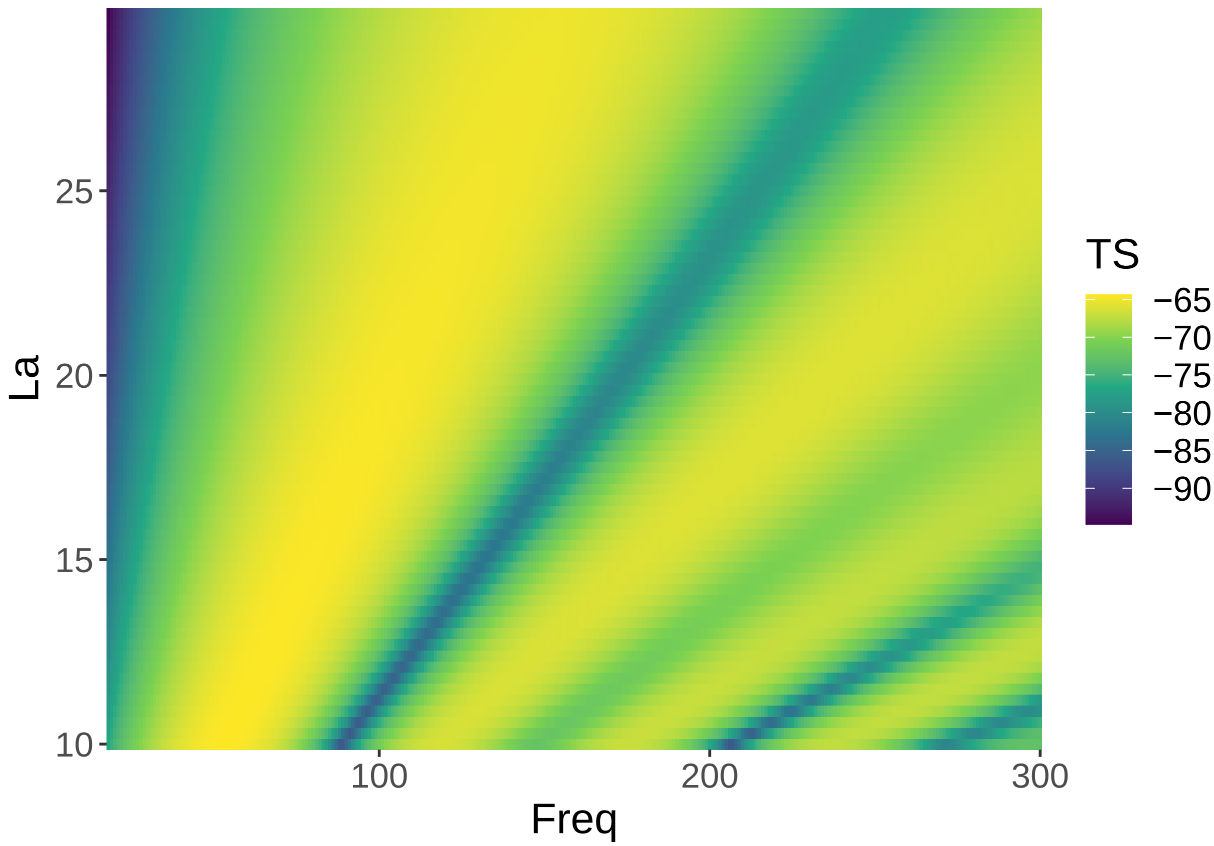


To study the effects of L/a on TS vs Frequency we can run a simulation:

```
la_sim <- function(la0,la1, interval=1,theta=0,L=38){
  runSim <- function(la, theta,L){
    para$shape$L = L #set any length, default is 30 mm
    para$shape$L_a = la #letś create a very fat krill
    para$orient$angm = theta

    res <- bscat(para=para, misc=misc, simOut = TRUE) #Target strength vs Frequency
    out = data.frame('Freq'=res$var,'TS' = res$y)
    out$La = la
    return(out)
  }
  lasim <- do.call(rbind, apply(as.matrix(seq(la0,la1,by=interval)),1,function(x) runSim(x,theta,L)))
  p = ggplot(data=lasim, aes(x=Freq,y=La,fill=TS))+
    geom_raster()+scale_fill_viridis_c()+
    scale_x_continuous(expand=c(0,0))+
    scale_y_continuous(expand=c(0,0))+
    theme(text=element_text(size=16))
  return(list(sim=lasim,plot=p))
  }
```

Let's for example run a simulation for L/a between 10 and 30:

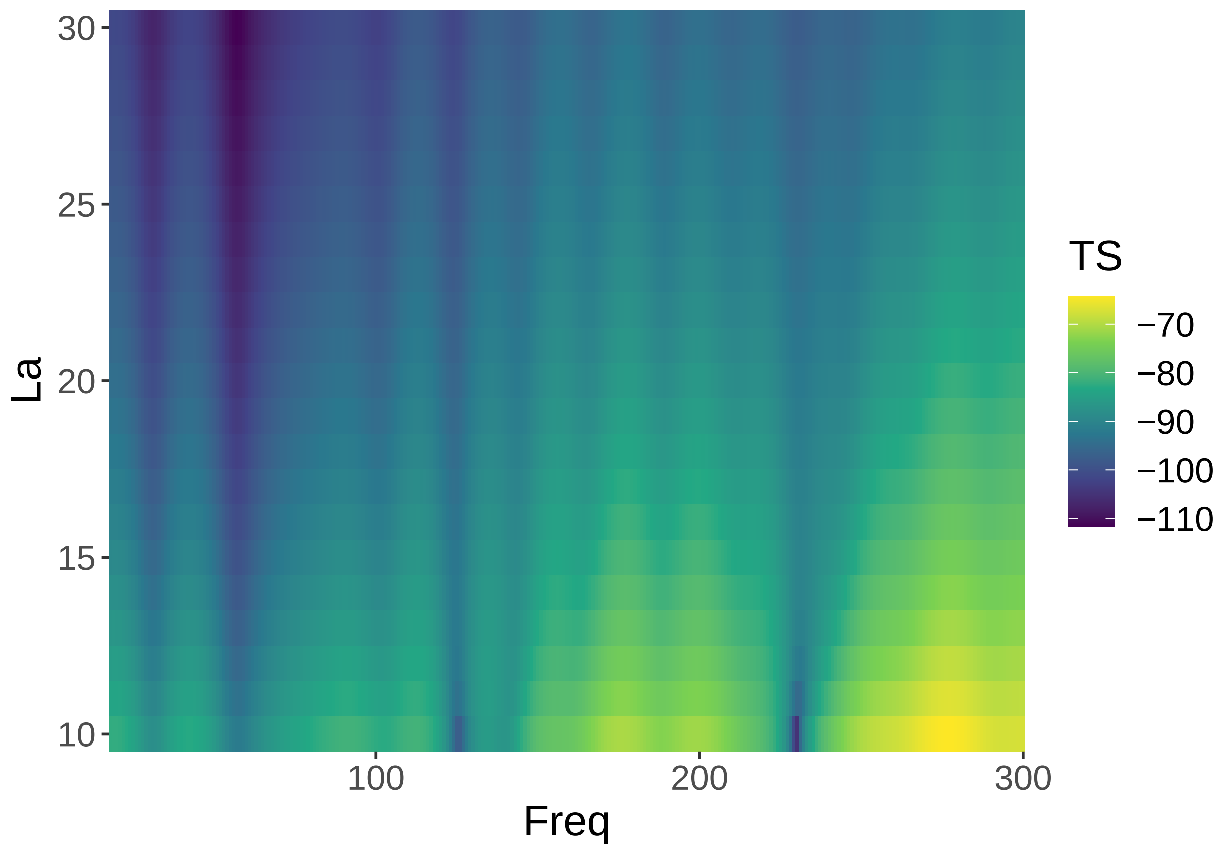
```
lasim1 = la_sim(10,30,0.3)
lasim1$plot
```



This simulation illustrates how the location of the nulls shifts with the width of the target. Keeping the length constant and varying L/a only changes the width of the target.

To prove this point, we can rotate the target by 90 degrees. In this case, the width will have a mus lower effect and the null locations remain largely the same, while the TS values will increase:

```
lasim2 = la_sim(10,30, theta=90)
lasim2$plot
```



References

Calise, L., and G. Skaret. 2011. "Sensitivity Investigation of the SDWBA Antarctic Krill Target Strength Model to Fatness, Material Contrasts and Orientation." *Ccamlr Science* 18: 97–122.