Supplement 11 - diveMove dive phases example

Chris Oosthuizen

2025-05-27

Introduction

A simple example showing quantile options to determine the bottom phase of a dive in diveMove.

```
#----------------------------------
# Setup
#----------------------------------

# Set system time zone
Sys.setenv(TZ = "GMT")

# Load packages
library(tidyverse)
library(diveMove)
library(here)
library(patchwork)

#==================================
# 1. Data
#==================================

file <- here("data_processed_1hz", "2023_01_14_KI18.rds")
dat <- readRDS(file)

  #---------------------
  # 1. import tag data
  #---------------------
  dat = readRDS(file)

  # Choose a depth threshold (in m) and time threshold (in sec) to define a dive
  # The time threshold can be used to filter dives after DiveMove calculated dives.
  depth.threshold = 3

  # remove dives that are not complete at the start and end of the video time series:
  # Find the index of the first and last values less than 0.5 depth
  dat = dat %>% arrange(date.time)
  index_start = min(which(dat$depth < depth.threshold))
  index_end = max(which(dat$depth < depth.threshold))

  # Slice the data frame from index_start to index_end
  dat = dplyr::slice(dat, index_start:index_end)
```

```r
# Calculate PCE in the data
dat$totalPCE = sum(dat$PCE_1hz)

#==================================
# 2. Divemove analysis
#==================================
dat = as.data.frame(dat)

# now start creating the TDR dive object
filename = "Divedata"

tdr <- createTDR(time = dat$date.time,
                 depth = dat$depth,
                 speed = FALSE,
                 dtime = 1,   #  sampling interval used in seconds
                 file = filename)

# Make sure 'depth' is positive and starts at zero!
# show(tdr)
```

calibrateDepth

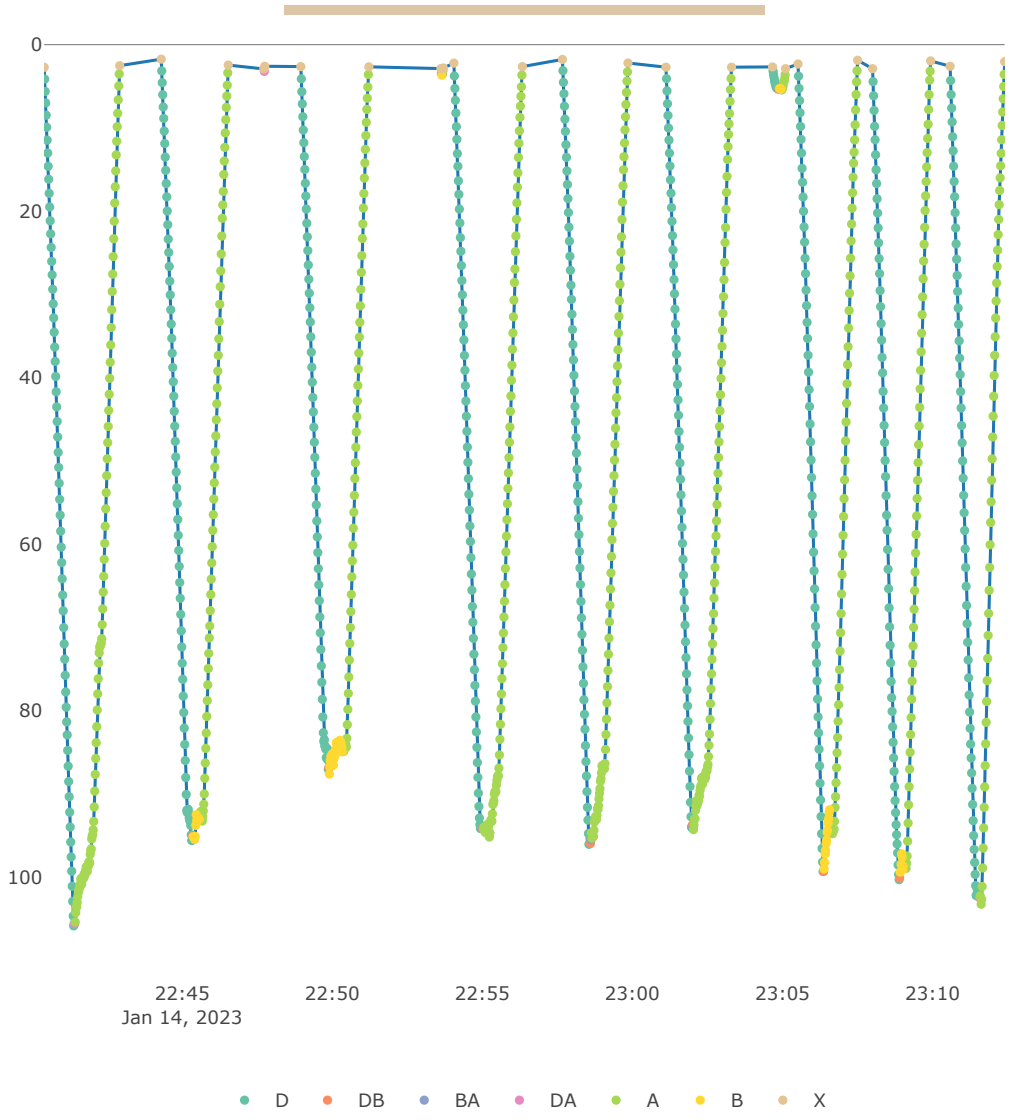
```r
# https://rdrr.io/cran/diveMove/man/calibrateDepth.html

# detect individual dives
tdr.calib_cran = calibrateDepth(tdr,
                                dive.thr = depth.threshold, # only select dives deeper than threshold
                                zoc.method='filter',
                                k=c(3, 5760),
                                probs=c(0.5, 0.02),
                                dive.model = "unimodal",
                                smooth.par=0.1,
                                knot.factor=20,
                                descent.crit.q=0.01,
                                ascent.crit.q=0,
                                na.rm=T)
```

```
## Record is truncated at the beginning and at the end
## 1 phases detected
```

```
## 193 dives detected
```

Plot some dives

```r
plotTDR(tdr.calib_cran , diveNo=123:134, what="phases", depth.lim=c(0, 120))
```



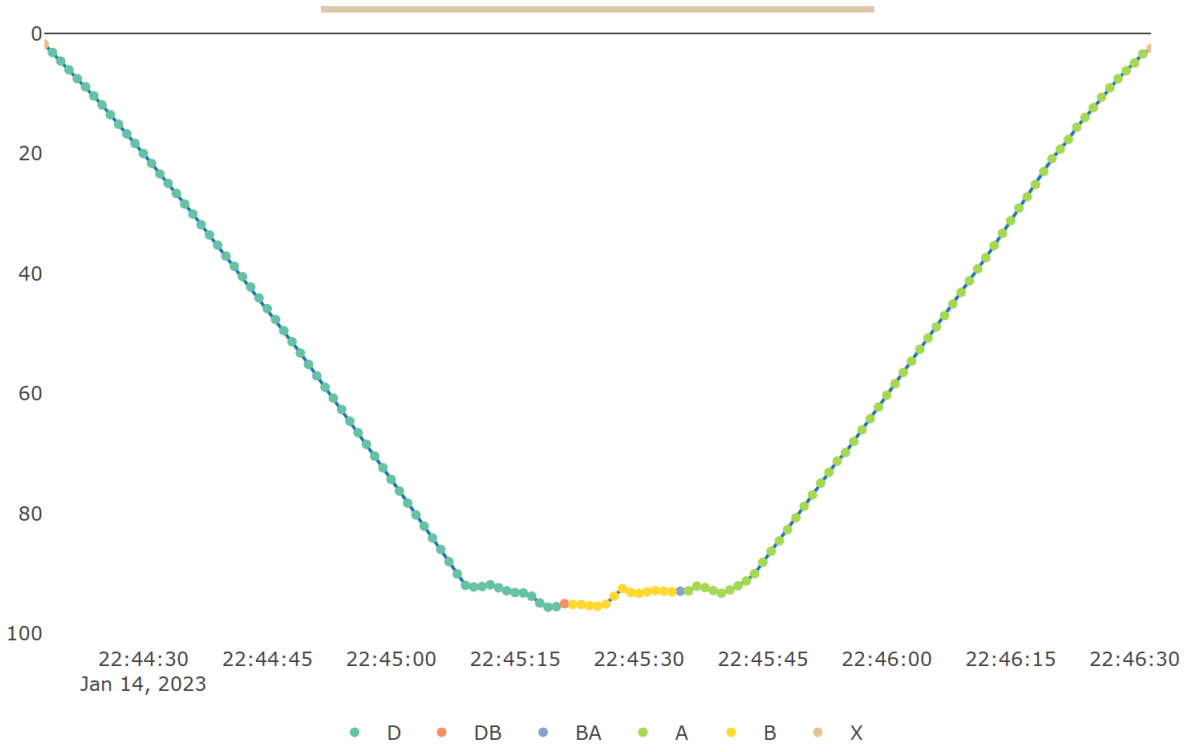

```
p1 = plotTDR(tdr.calib_cran , diveNo=124, what="phases", depth.lim=c(0, 120))
```

```
p2 = plotTDR(tdr.calib_cran , diveNo=126, what="phases", depth.lim=c(0, 120))

p3 = plotTDR(tdr.calib_cran , diveNo=128, what="phases", depth.lim=c(0, 120))

p4 = plotTDR(tdr.calib_cran , diveNo=129, what="phases", depth.lim=c(0, 120))

p5 = plotTDR(tdr.calib_cran , diveNo=130, what="phases", depth.lim=c(0, 120))

p6 = plotTDR(tdr.calib_cran , diveNo=132, what="phases", depth.lim=c(0, 120))
```

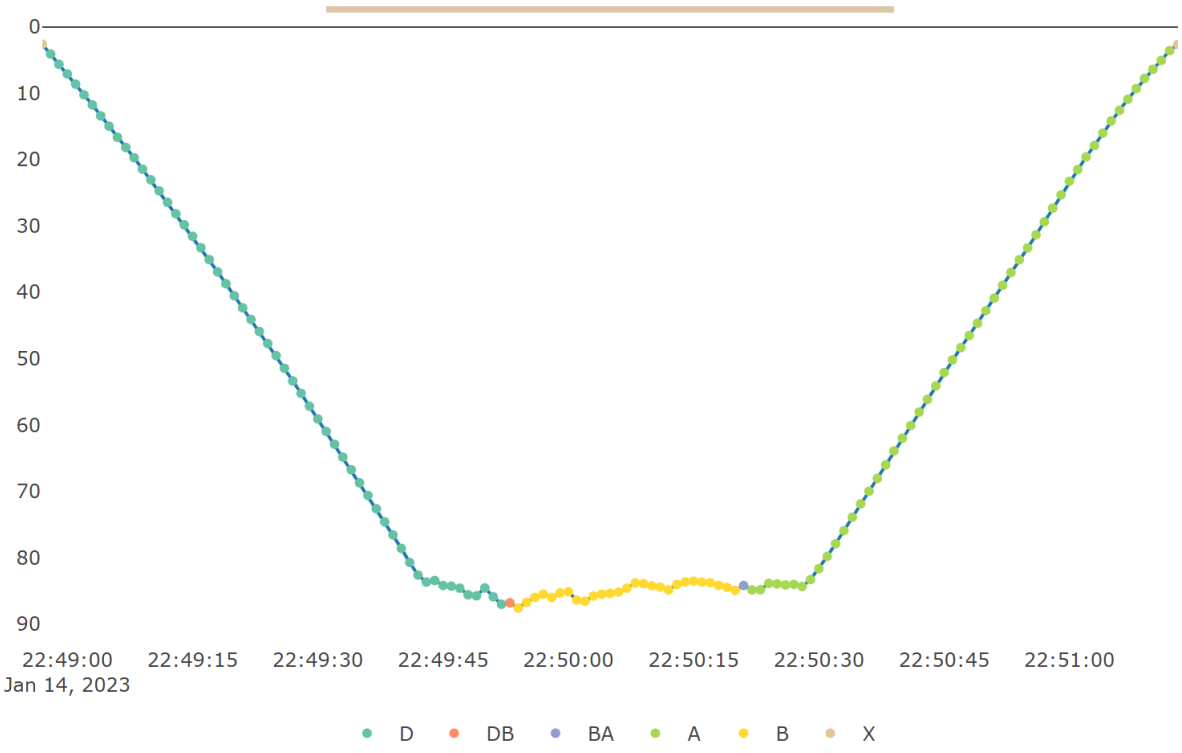
```
library(htmlwidgets)
library(webshot2)
```

```
## Warning: package 'webshot2' was built under R version 4.3.3
```

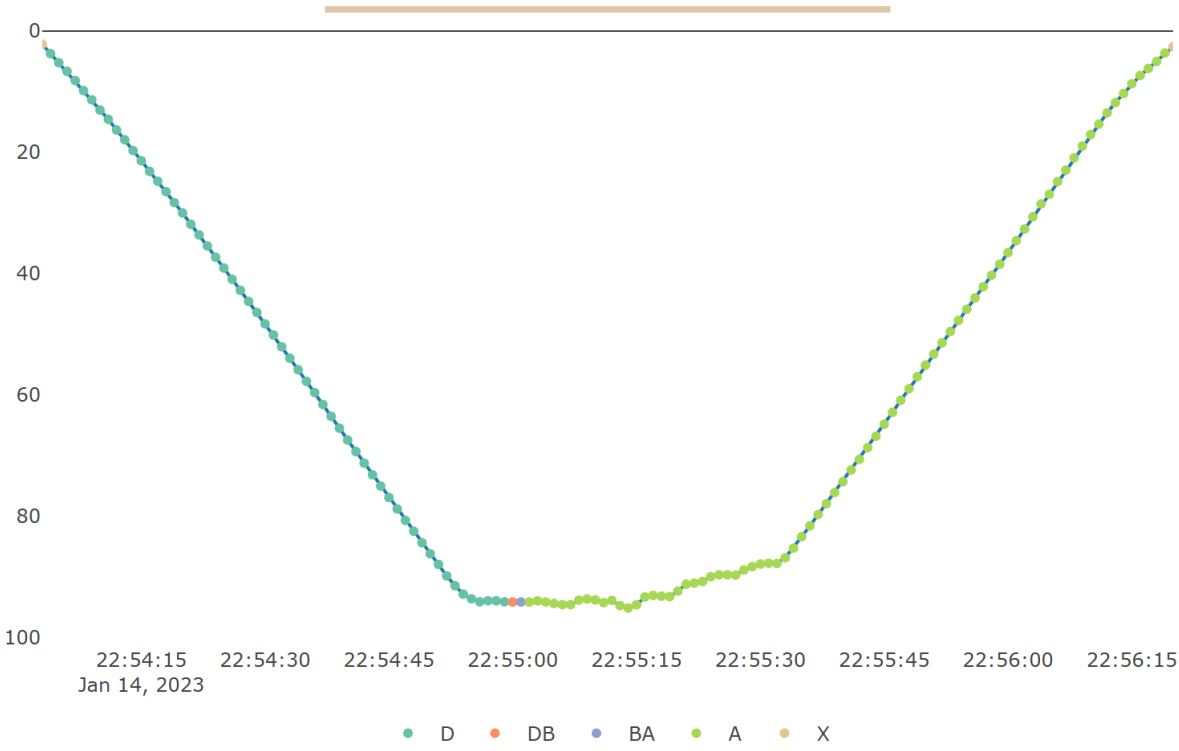
```
saveWidget(p1, "plot1.html", selfcontained = TRUE)
webshot("plot1.html", "plot1.png", vwidth = 800, vheight = 600, zoom = 2)
```



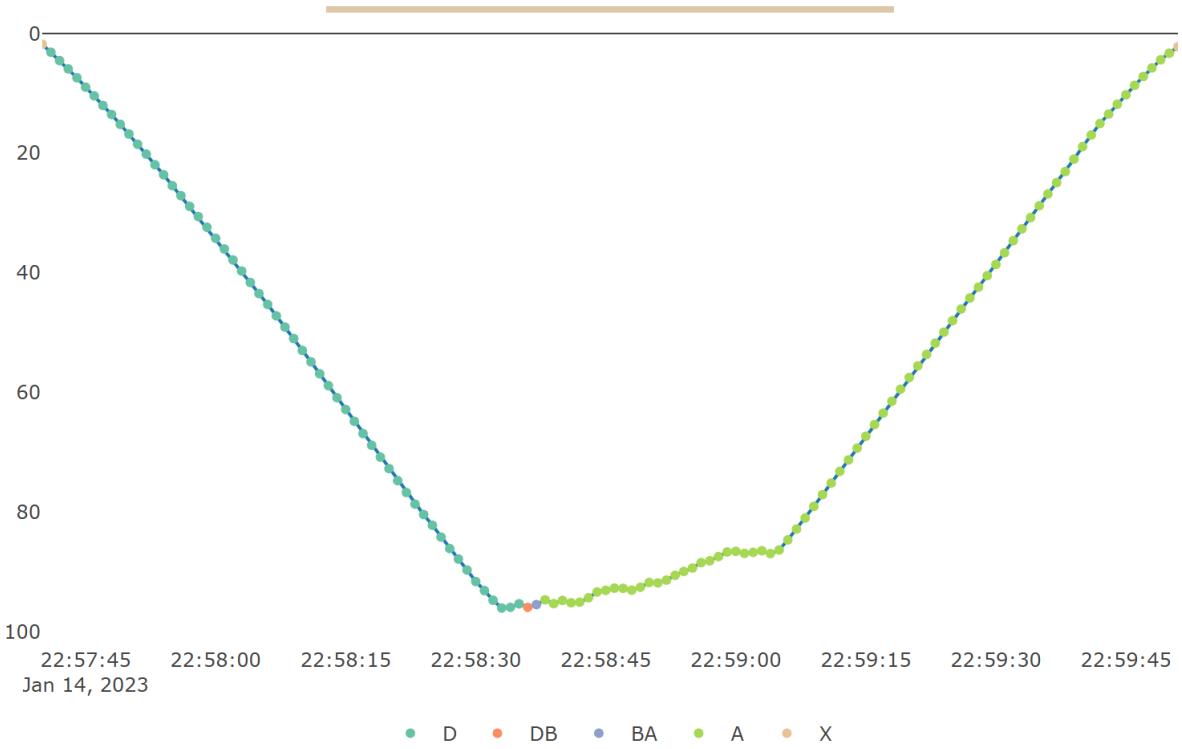

```
saveWidget(p2, "plot2.html", selfcontained = TRUE)
webshot("plot2.html", "plot2.png", vwidth = 800, vheight = 600, zoom = 2)
```



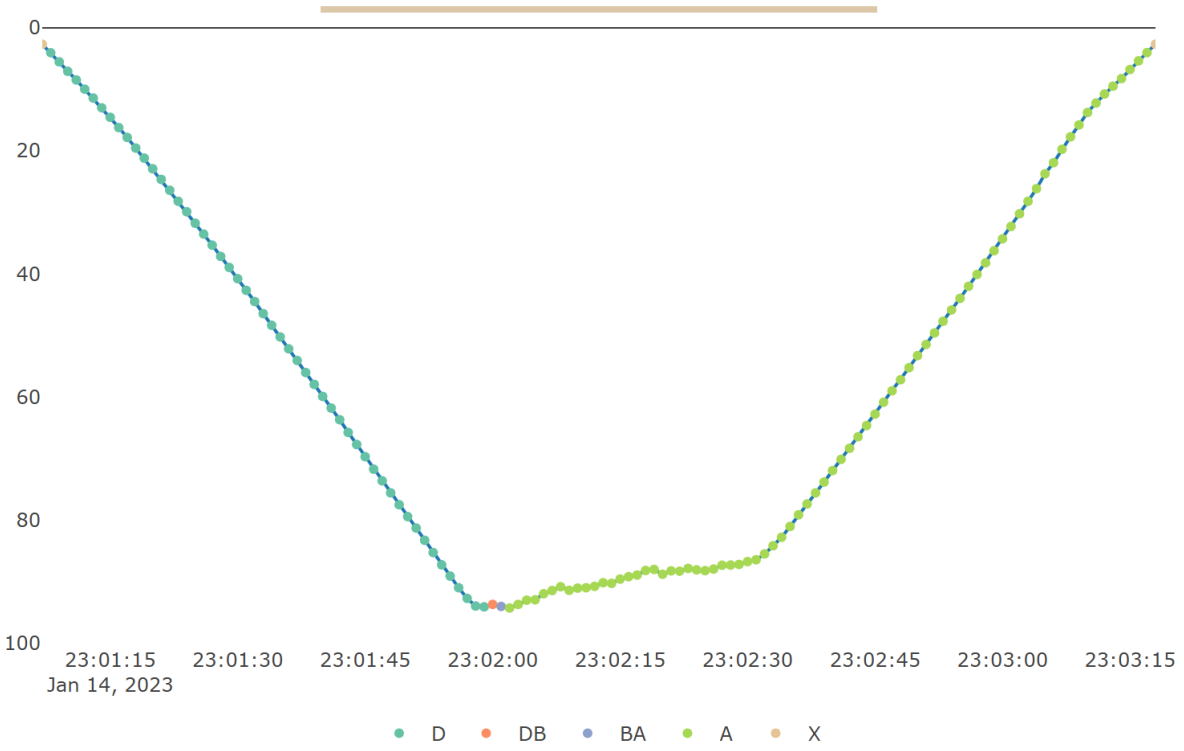

```
saveWidget(p3, "plot3.html", selfcontained = TRUE)
webshot("plot3.html", "plot3.png", vwidth = 800, vheight = 600, zoom = 2)
```



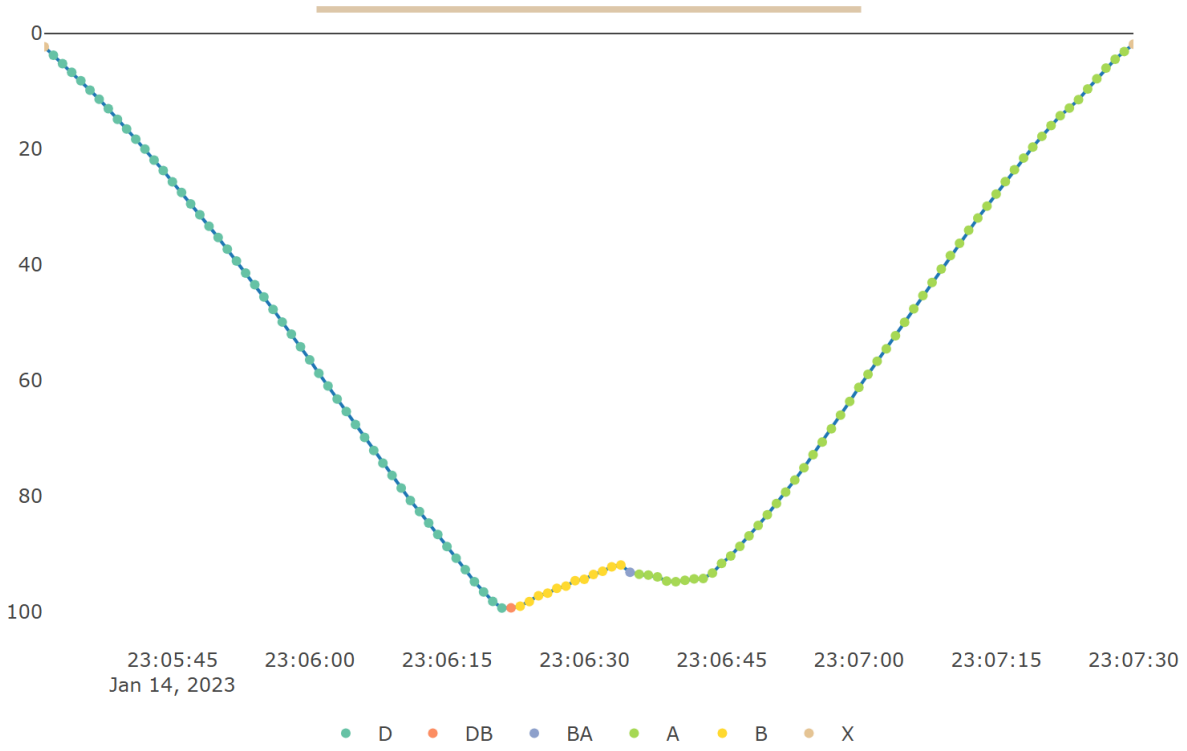

```
saveWidget(p4, "plot4.html", selfcontained = TRUE)
webshot("plot4.html", "plot4.png", vwidth = 800, vheight = 600, zoom = 2)
```




```
saveWidget(p5, "plot5.html", selfcontained = TRUE)
webshot("plot5.html", "plot5.png", vwidth = 800, vheight = 600, zoom = 2)
```




```
saveWidget(p6, "plot6.html", selfcontained = TRUE)
webshot("plot6.html", "plot6.png", vwidth = 800, vheight = 600, zoom = 2)
```

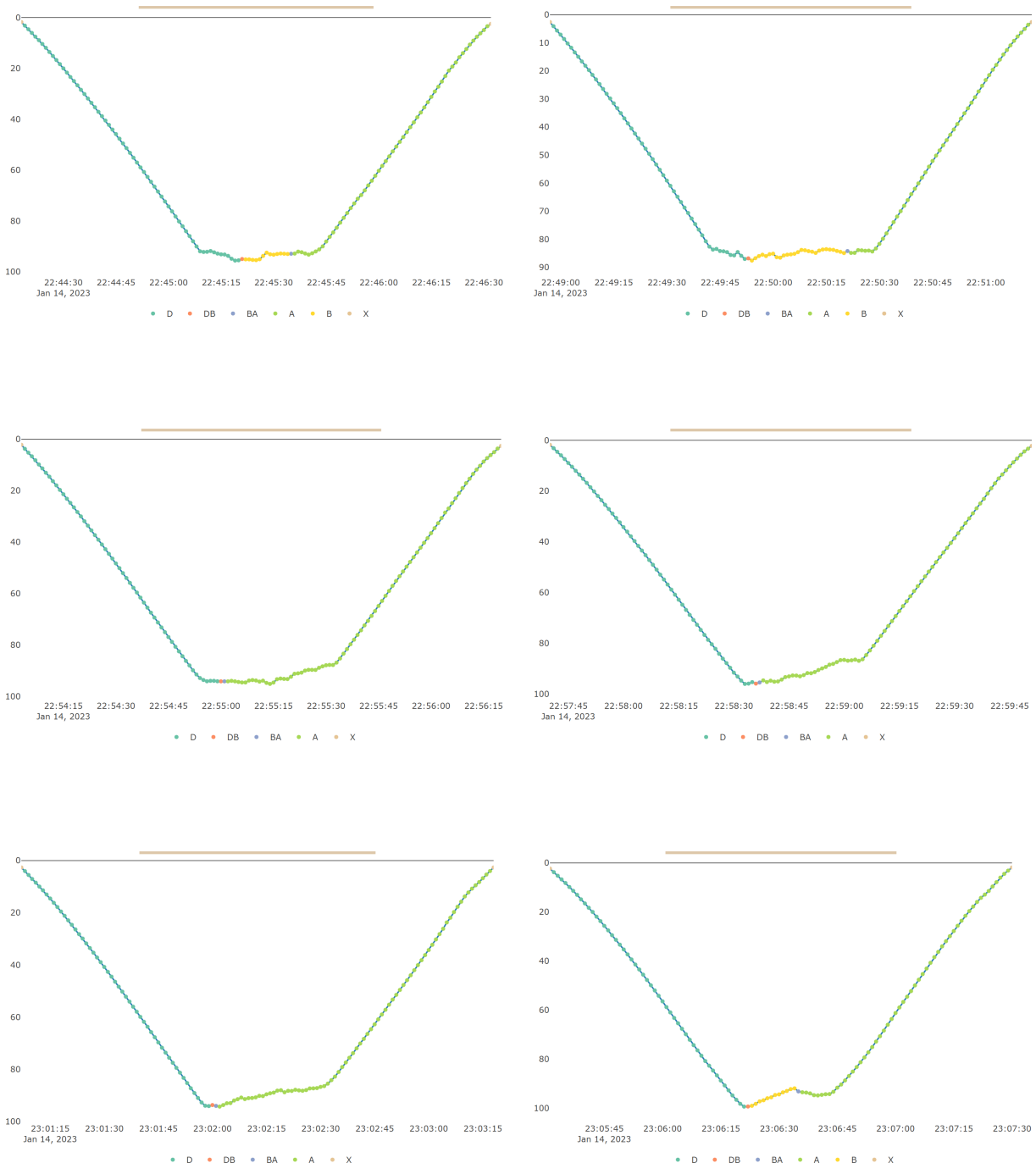



```
library(png)
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# Load and wrap PNGs as grobs
grobs <- lapply(paste0("plot", 1:6, ".png"), function(x) {
  rasterGrob(readPNG(x), interpolate = TRUE)
})

# Arrange with minimal spacing in a 2×3 grid
grid.arrange(
  grobs = grobs,
  nrow = 3, ncol = 2,
  top = NULL,
  bottom = NULL,
  left = NULL,
  right = NULL,
  padding = unit(0, "line") # Removes padding between plots
)
```



Change quantile thresholds:

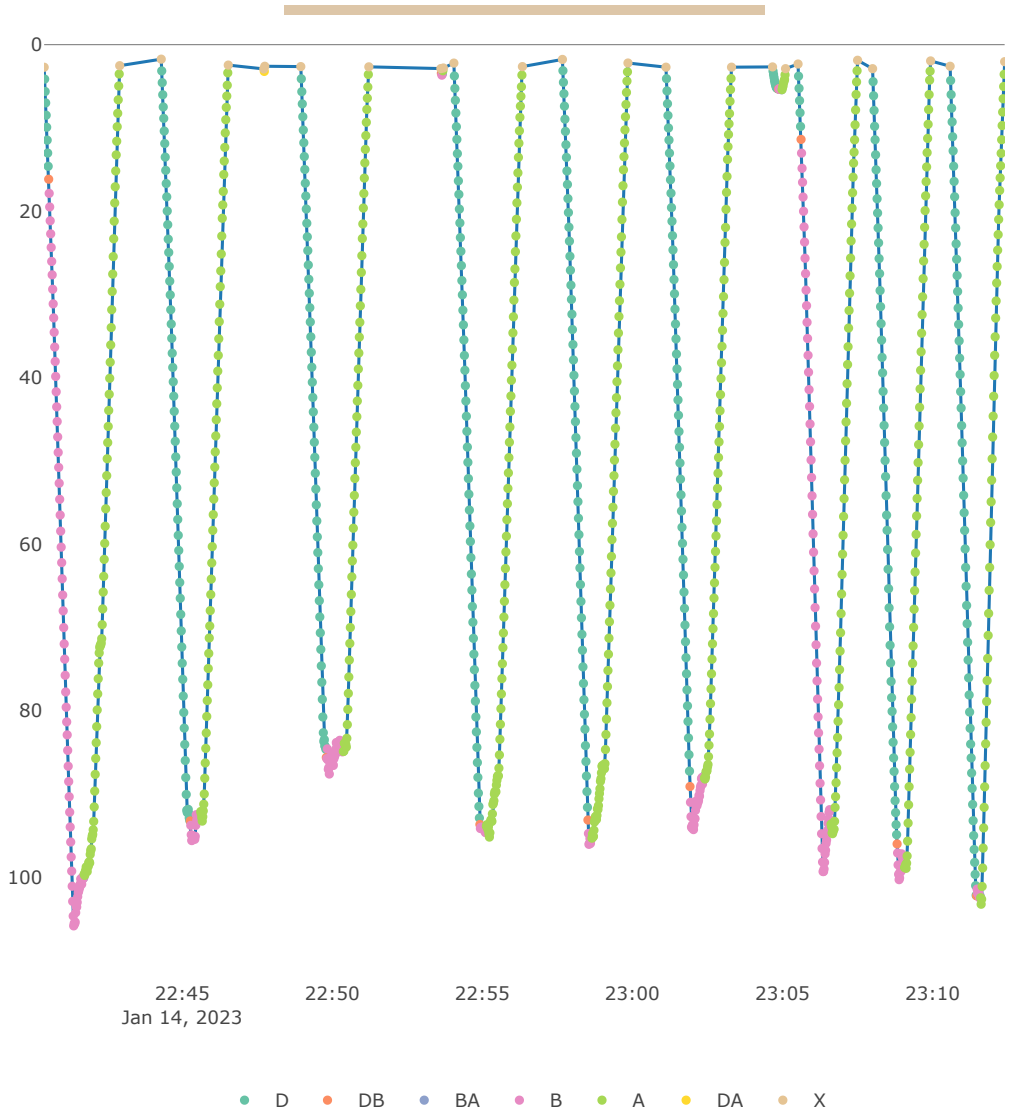
```
tdr.calib_quant01 = calibrateDepth(tdr,
                                   dive.thr = depth.threshold, # only select dives deeper than threshold
```

```
                            zoc.method='filter',
                            k=c(3, 5760),
                            probs=c(0.5, 0.02),
                            dive.model = "unimodal",
                            smooth.par=0.1,
                            knot.factor=20,
                            descent.crit.q=0.1,
                            ascent.crit.q=0.01,
                            na.rm=T)
```

```
## Record is truncated at the beginning and at the end
## 1 phases detected
```

```
## 193 dives detected
```

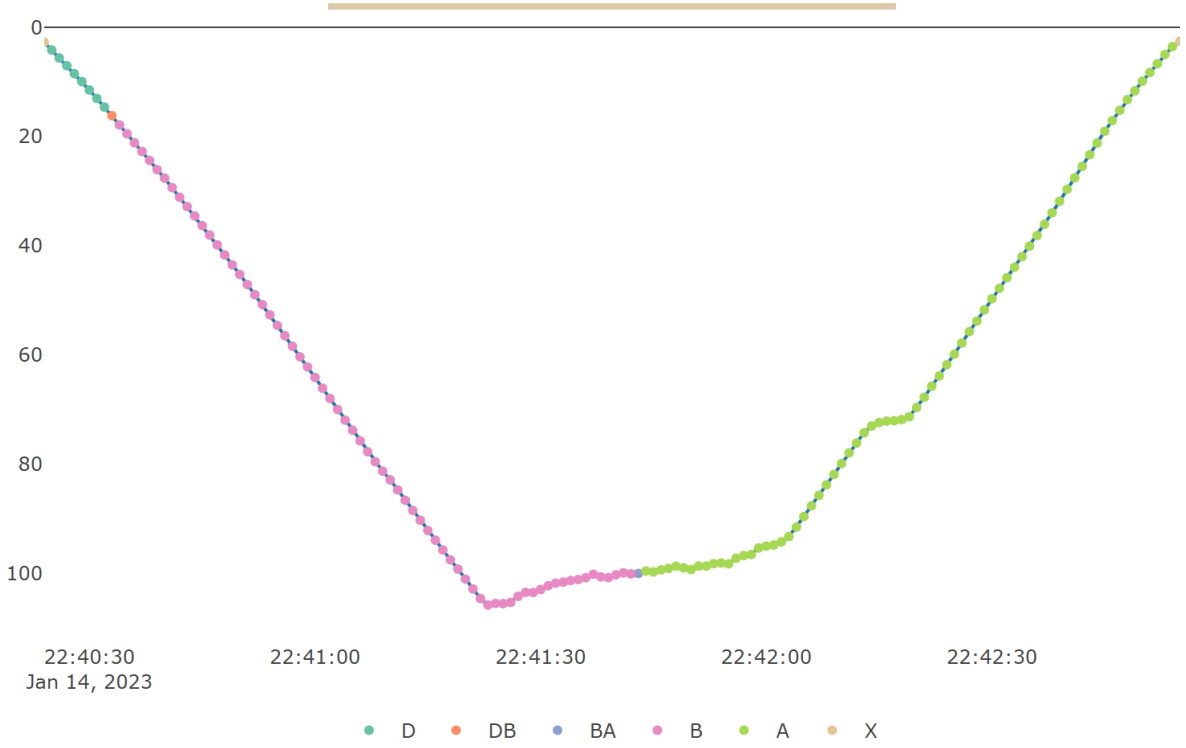
```
# Assess as few dive phases:

plotTDR(tdr.calib_quant01 , diveNo=123:134, what="phases", depth.lim=c(0, 120))
```



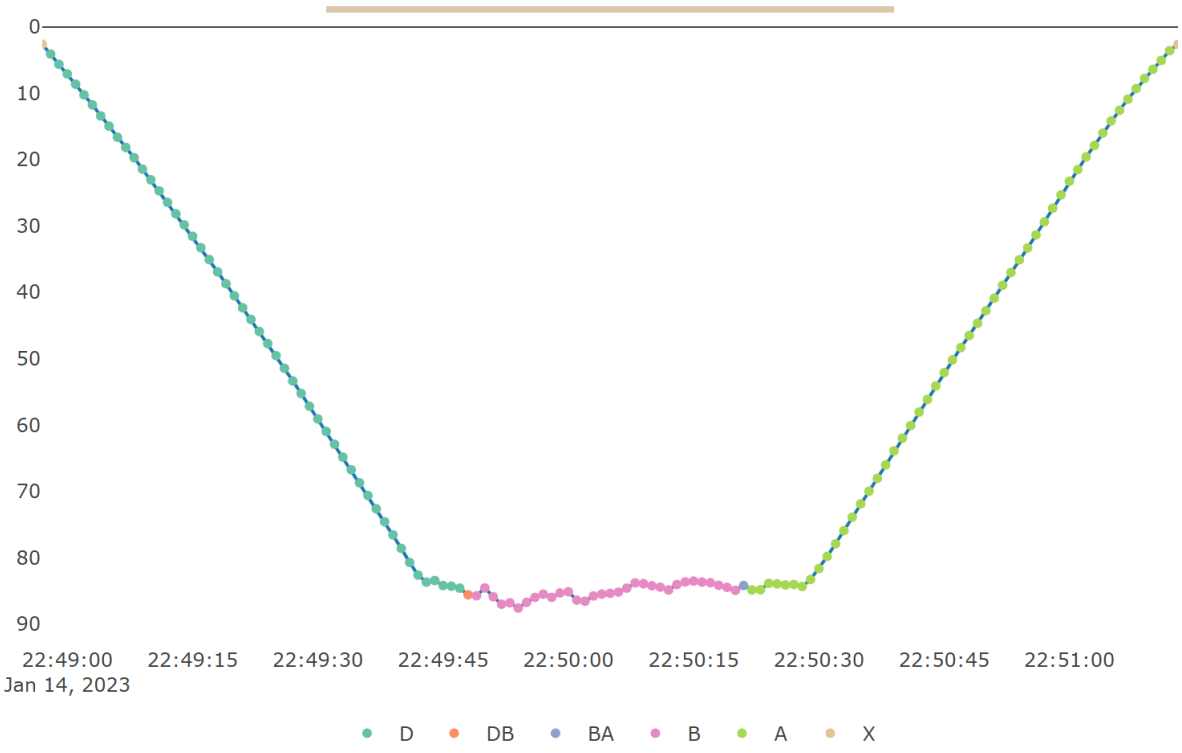

```
q1 = plotTDR(tdr.calib_quant01 , diveNo=123, what="phases", depth.lim=c(0, 120))
```

```
#plotTDR(tdr.calib_quant01 , diveNo=124, what="phases", depth.lim=c(0, 120))

q3 = plotTDR(tdr.calib_quant01 , diveNo=126, what="phases", depth.lim=c(0, 120))

q4 = plotTDR(tdr.calib_quant01 , diveNo=128, what="phases", depth.lim=c(0, 120))

# plotTDR(tdr.calib_quant01 , diveNo=129, what="phases", depth.lim=c(0, 120))
#
# plotTDR(tdr.calib_quant01 , diveNo=130, what="phases", depth.lim=c(0, 120))
#
q6 =  plotTDR(tdr.calib_quant01 , diveNo=132, what="phases", depth.lim=c(0, 120))
```

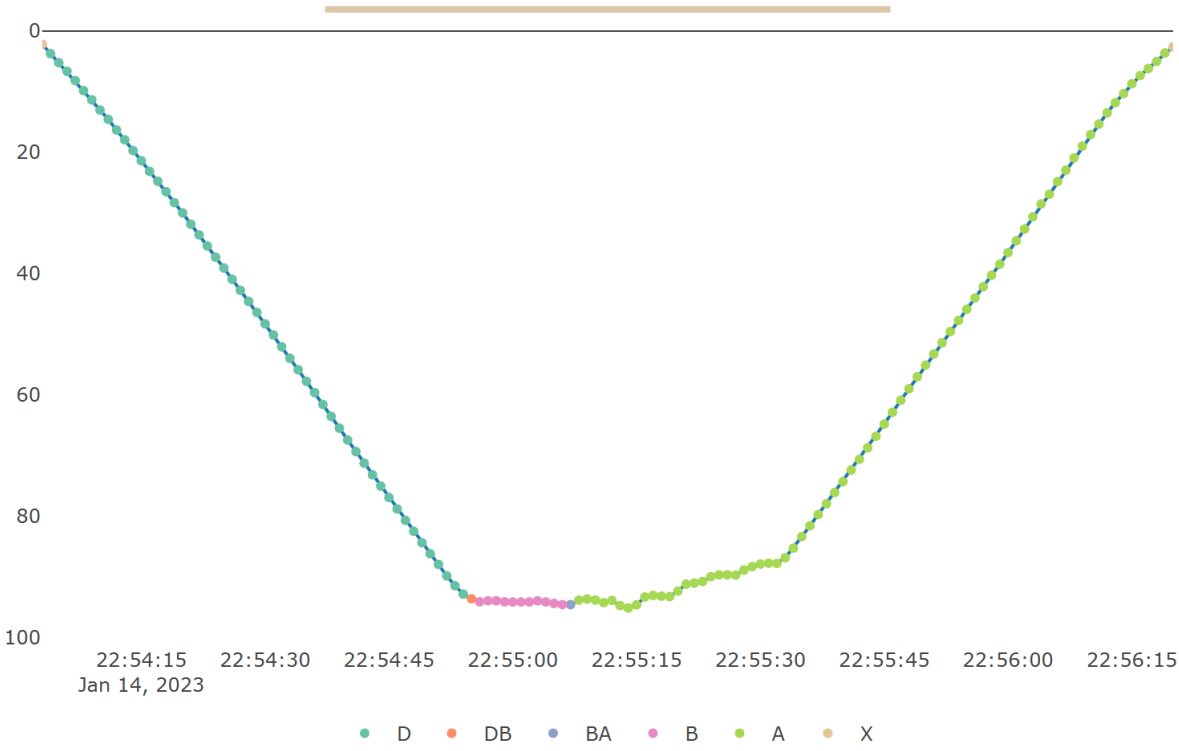
```
saveWidget(q1, "plotq1.html", selfcontained = TRUE)
webshot("plotq1.html", "plotq1.png", vwidth = 800, vheight = 600, zoom = 2)
```



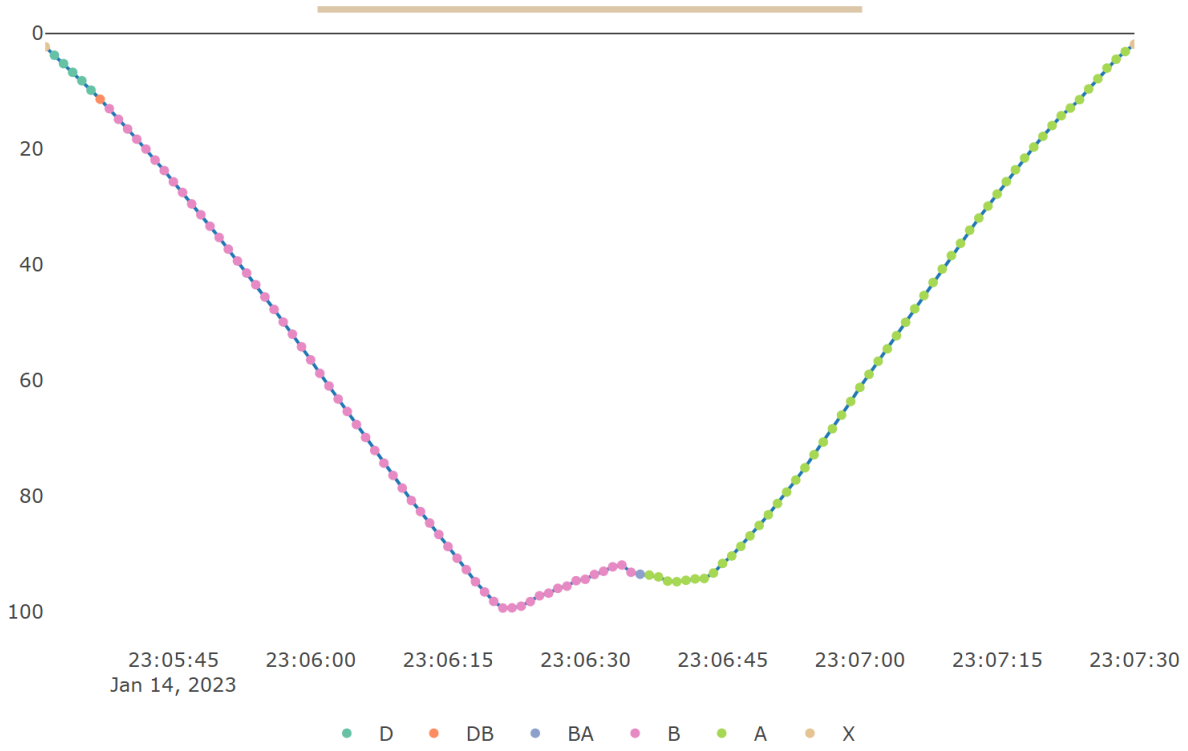

```
saveWidget(q3, "plotq3.html", selfcontained = TRUE)
webshot("plotq3.html", "plotq3.png", vwidth = 800, vheight = 600, zoom = 2)
```



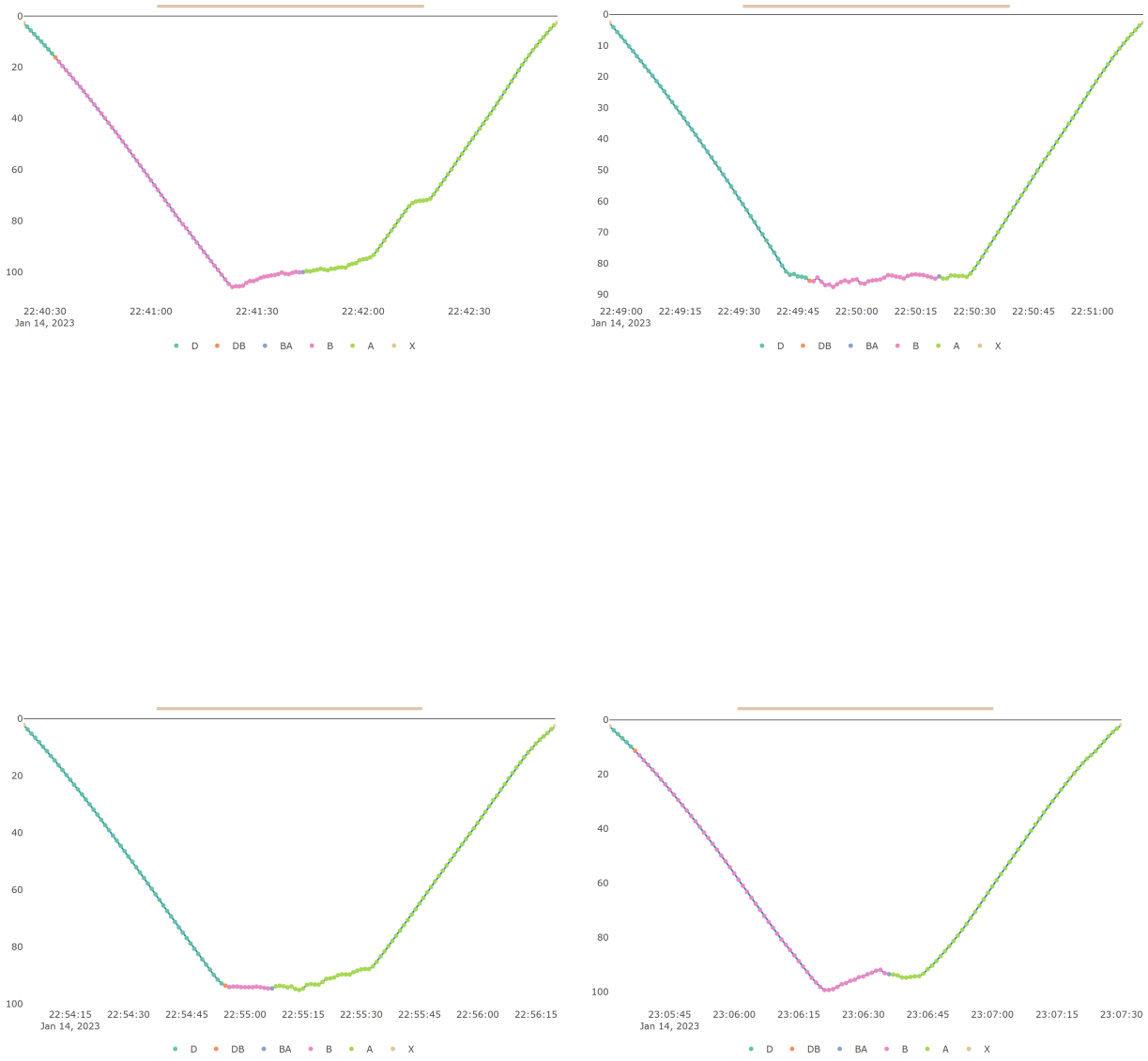

```
saveWidget(q4, "plotq4.html", selfcontained = TRUE)
webshot("plotq4.html", "plotq4.png", vwidth = 800, vheight = 600, zoom = 2)
```




```
saveWidget(q6, "plotq6.html", selfcontained = TRUE)
webshot("plotq6.html", "plotq6.png", vwidth = 800, vheight = 600, zoom = 2)
```




```r
# Load and wrap PNGs as grobs
grobs <- lapply(paste0("plotq", c(1,3,4,6), ".png"), function(x) {
  rasterGrob(readPNG(x), interpolate = TRUE)
})

# Arrange with minimal spacing in a 2×3 grid
grid.arrange(
  grobs = grobs,
  nrow = 2, ncol = 2,
  top = NULL,
  bottom = NULL,
  left = NULL,
  right = NULL,
  padding = unit(0, "line") # Removes padding between plots
)
```



Can't be too large - the 'descent phase' in some dives are classified as 'bottom phase'.

Increase the knot.factor

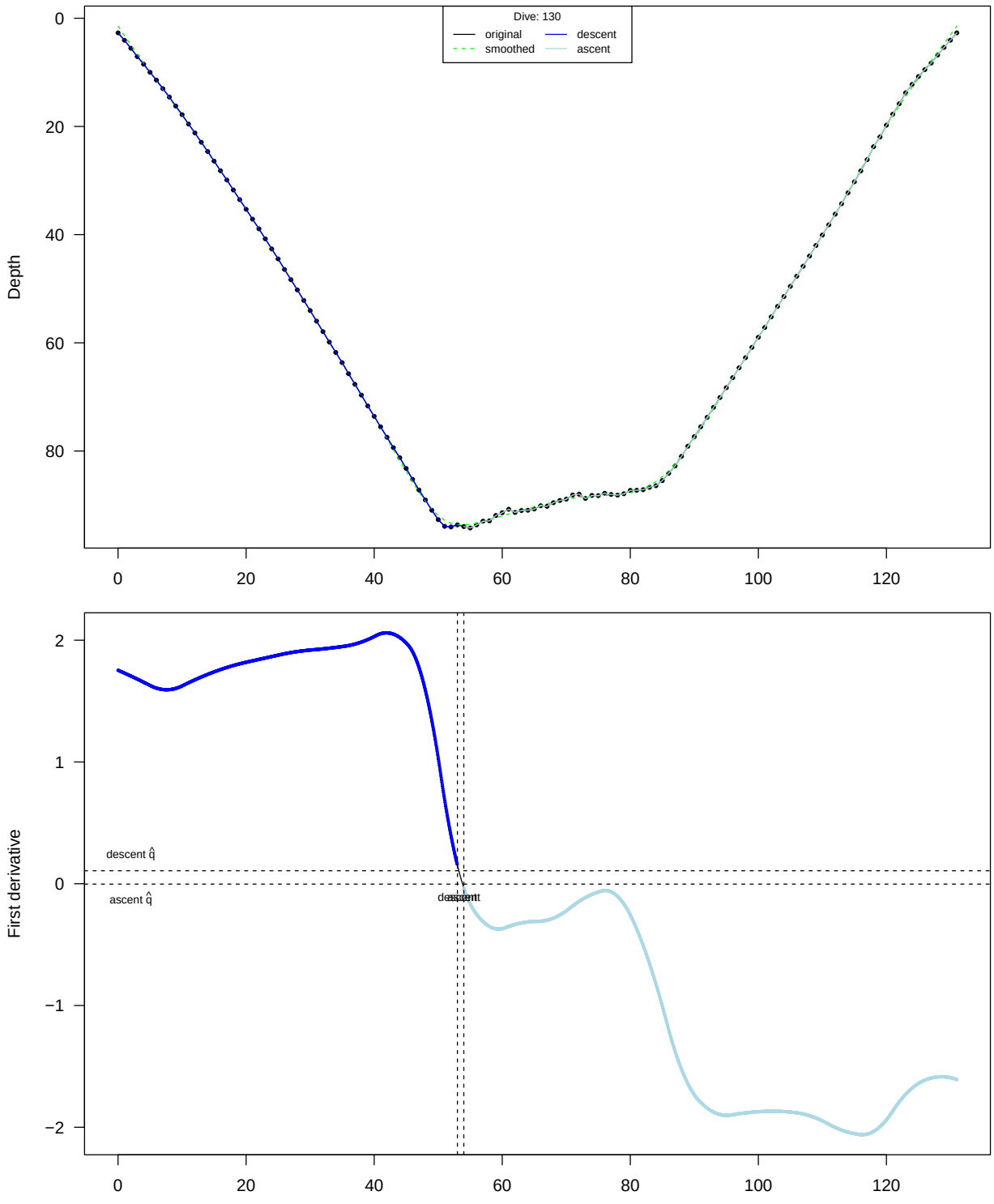
knot.factor=80 descent.crit.q=0.05 ascent.crit.q=0.03

```r
# Change quantile thresholds:

# detect individual dives
tdr.calib_knot80 = calibrateDepth(tdr,
                            dive.thr = depth.threshold, # only select dives deeper than threshold
                            zoc.method='filter',
                            k=c(3, 5760),
                            probs=c(0.5, 0.02),
                            dive.model = "unimodal",
                            smooth.par=0.1,
                            knot.factor=80,
                            descent.crit.q=0.05,
                            ascent.crit.q=0.03,
                            na.rm=T)
```

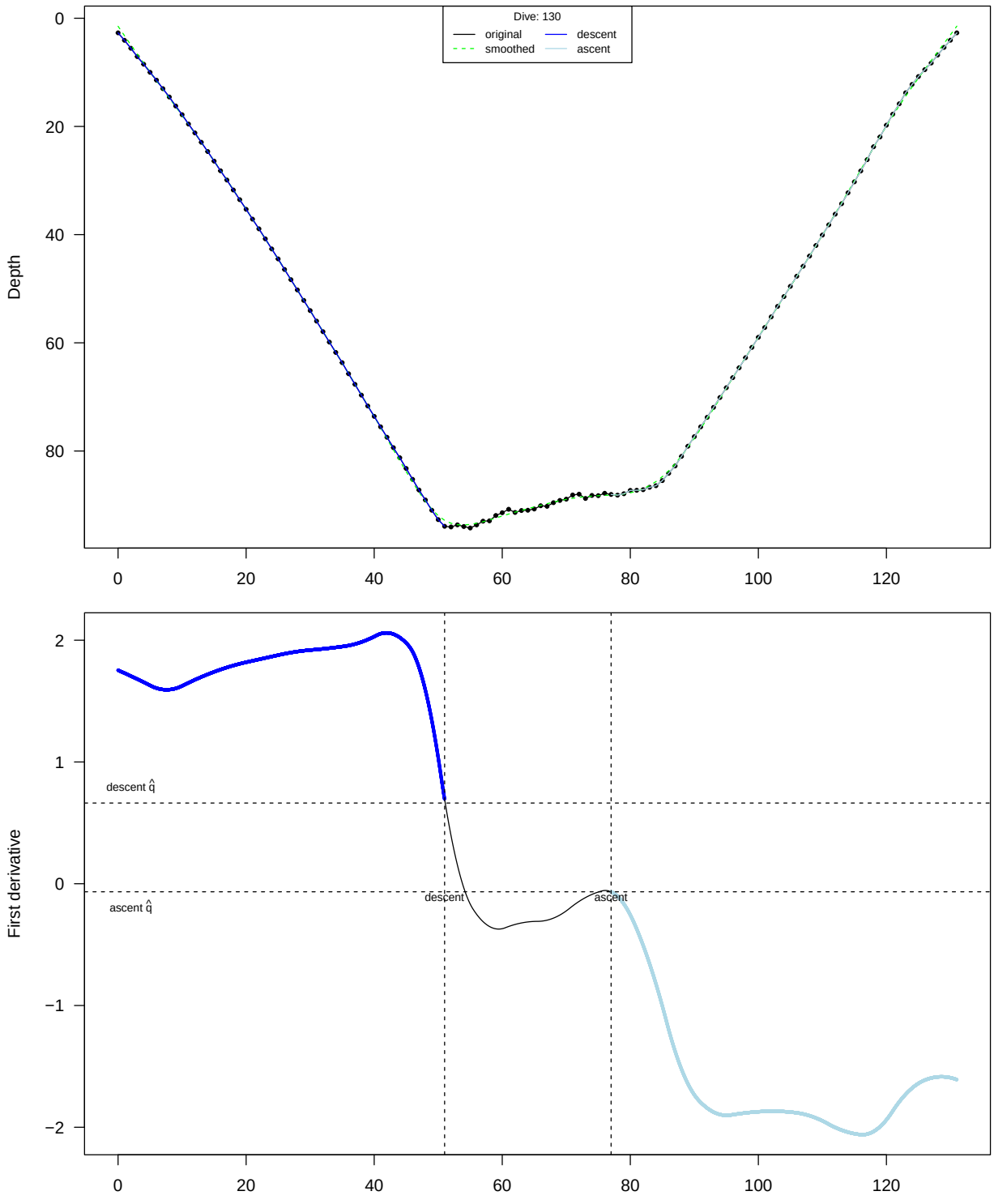
```
## Record is truncated at the beginning and at the end
## 1 phases detected
```

```
## 193 dives detected
```

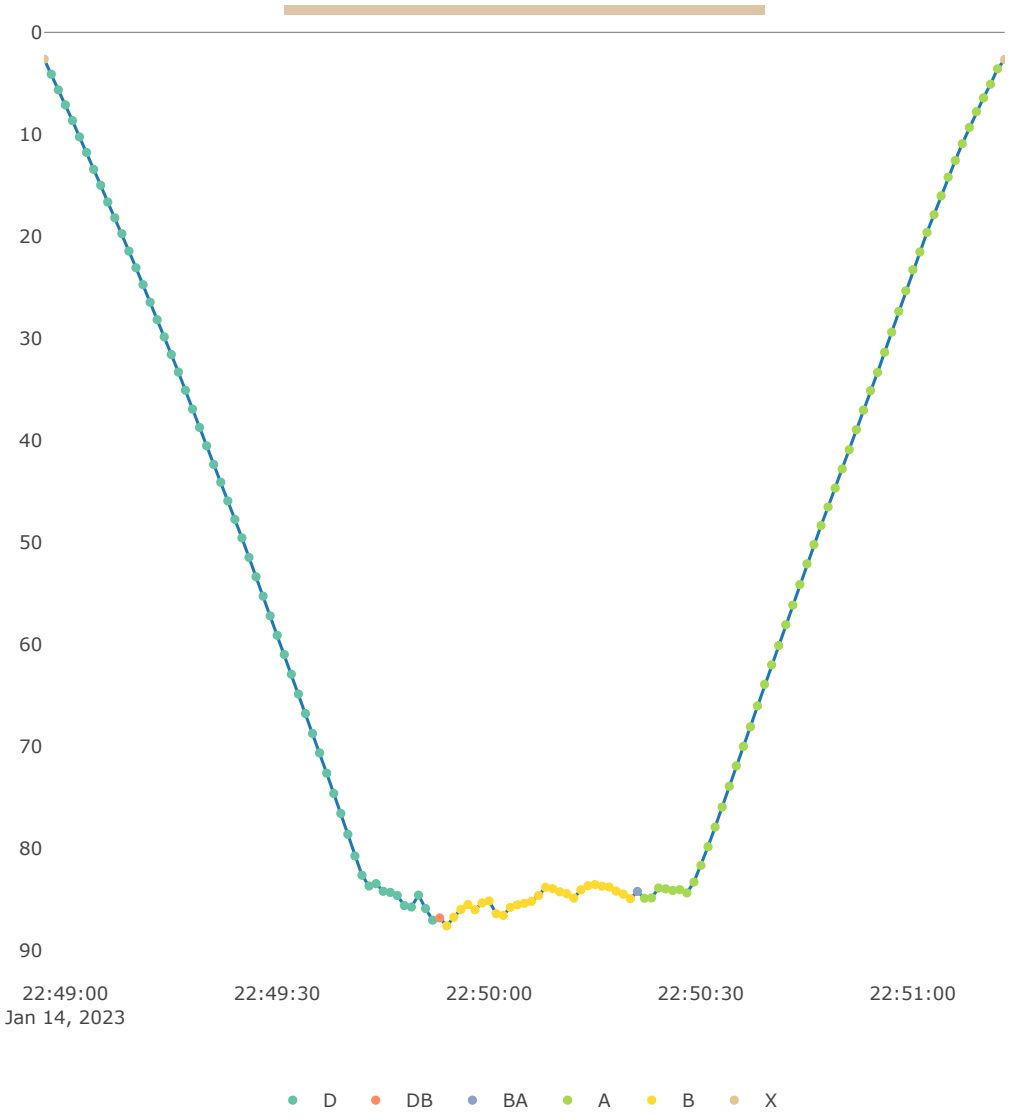
```r
# Compare to earlier analyses:
plotDiveModel(tdr.calib_cran, diveNo=130)
```



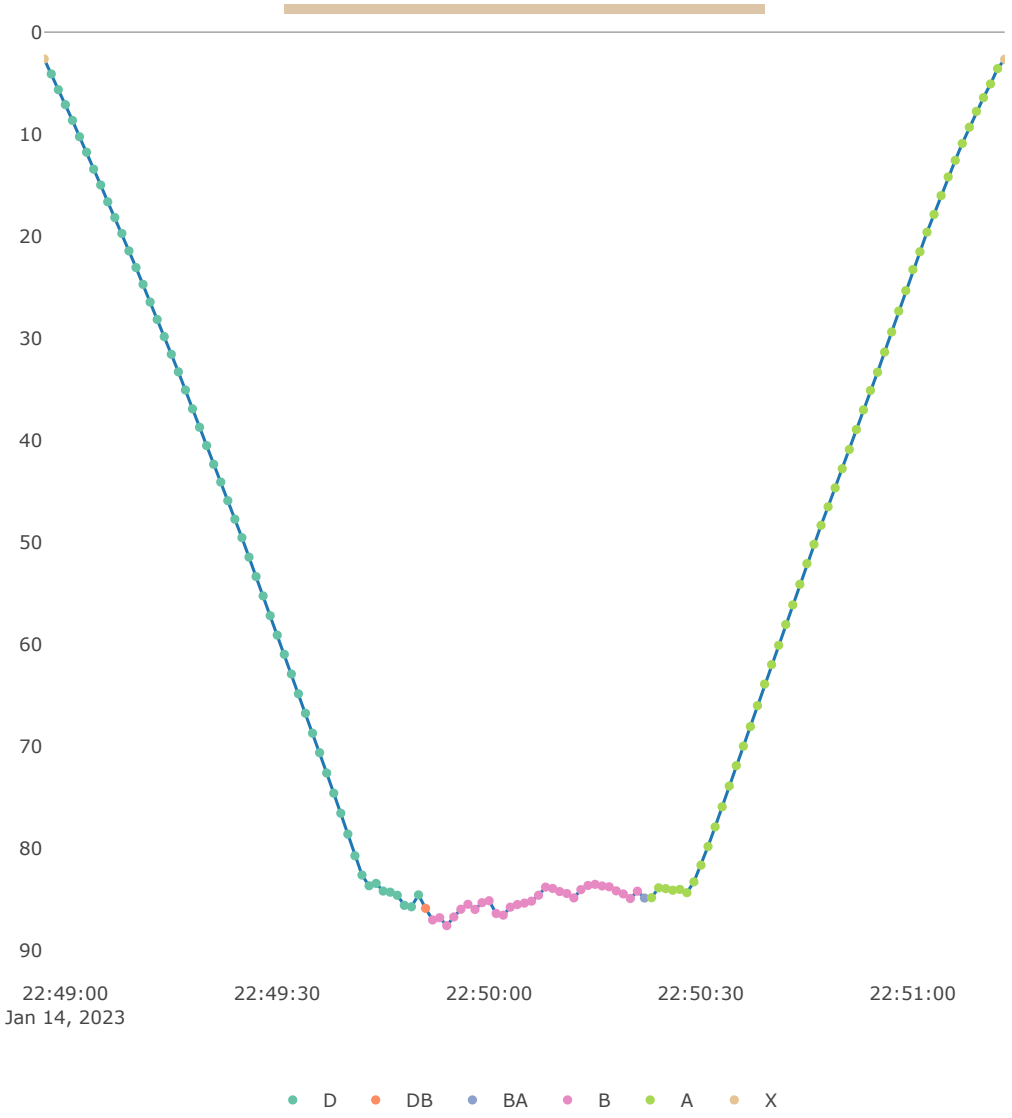

```
plotDiveModel(tdr.calib_knot80, diveNo=130)
```



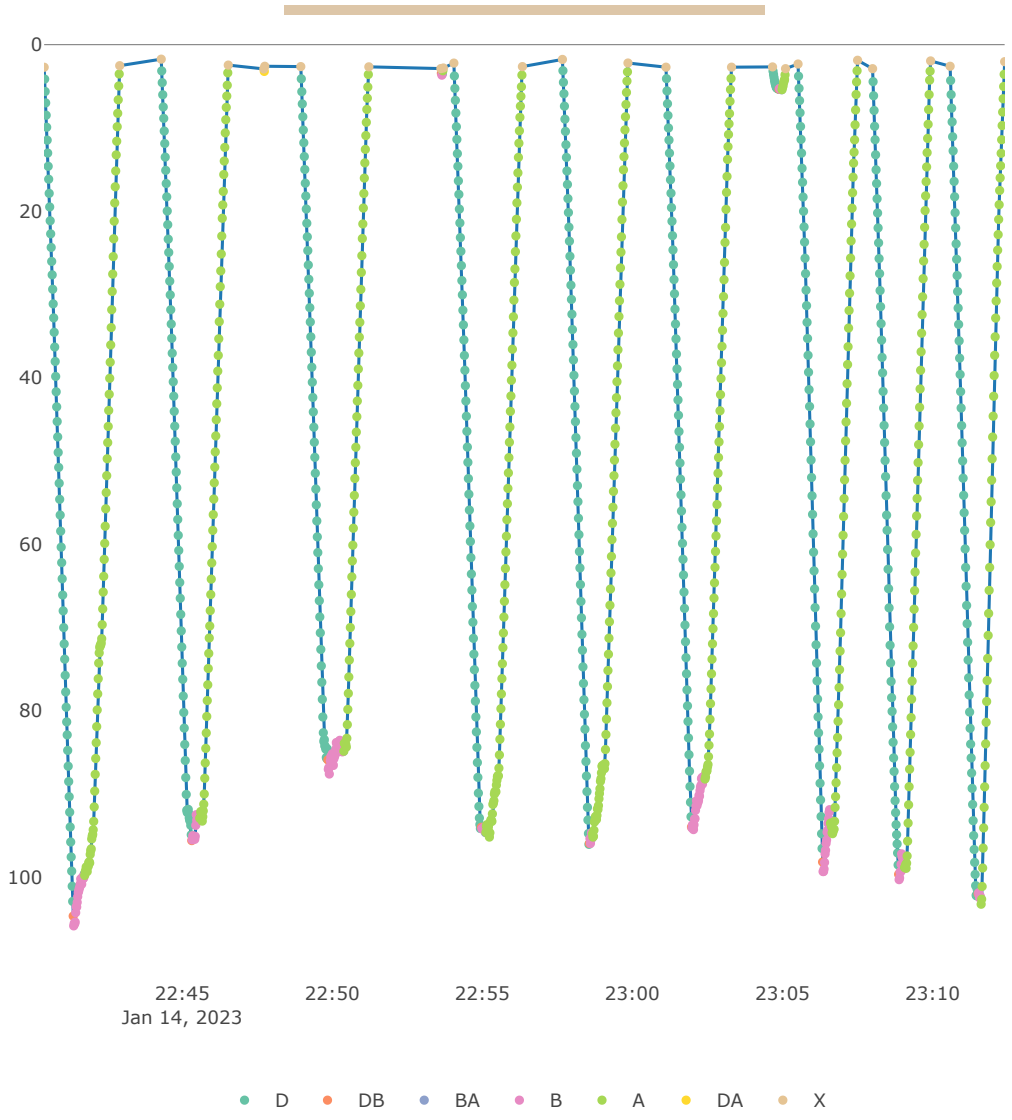

```
# Notice the increase in bottom phase boundaries
plotTDR(tdr.calib_cran , diveNo=126, what="phases", depth.lim=c(0, 120))
```


0
10
20
30
40
50
60
70
80
90
22:49:00
Jan 14, 2023
22:49:30
22:50:00
22:50:30
22:51:00
D
DB
BA
A
B
X

```
plotTDR(tdr.calib_knot80 , diveNo=126, what="phases", depth.lim=c(0, 120))
```



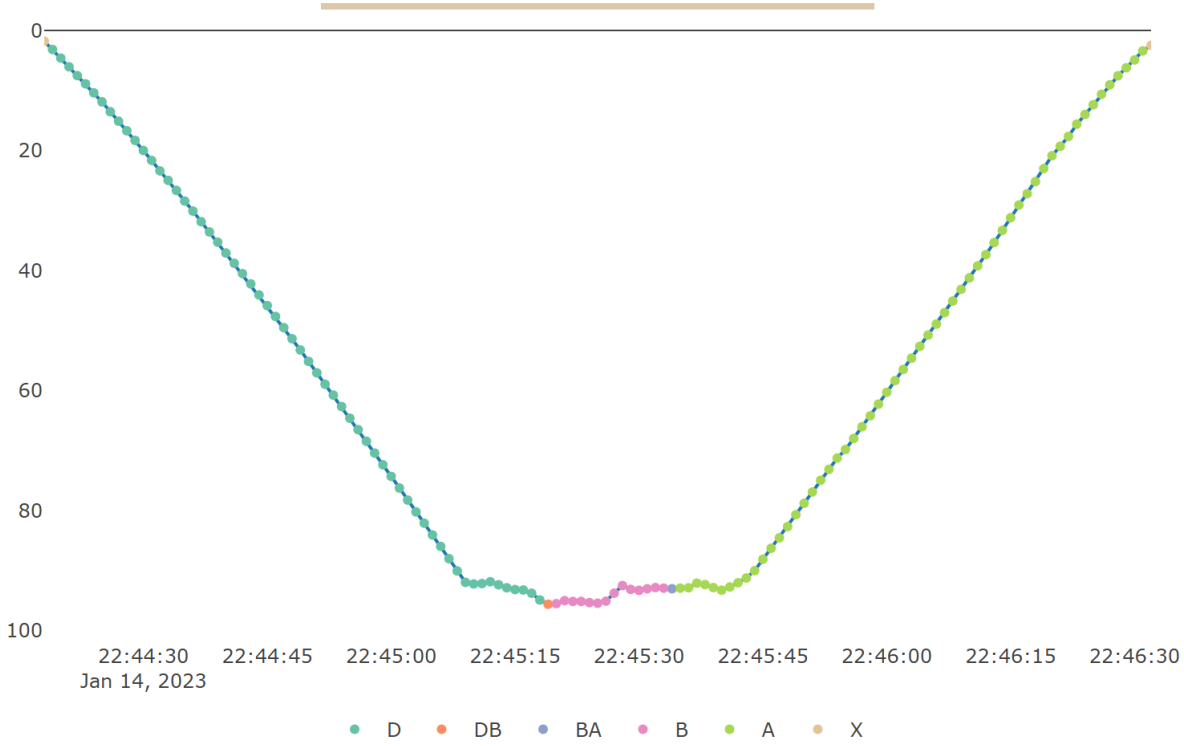
```r
# Assess a few dive phases:

plotTDR(tdr.calib_knot80 , diveNo=123:134, what="phases", depth.lim=c(0, 120))
```



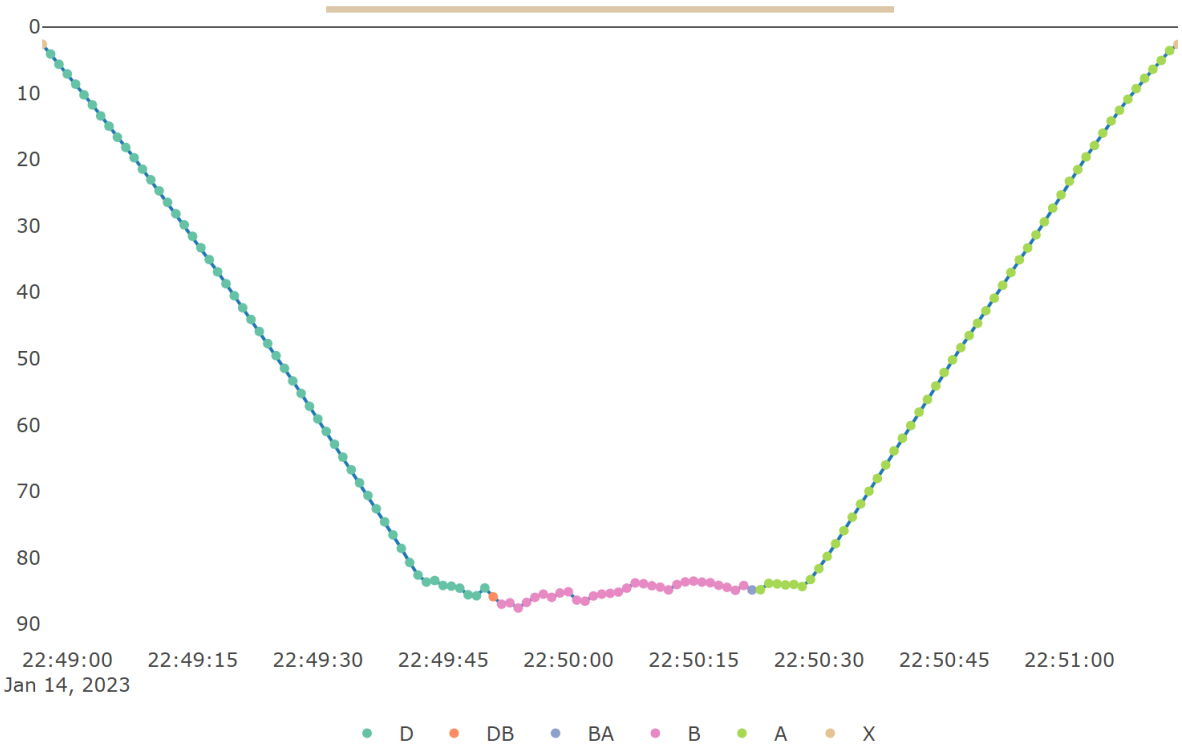

```
d1 = plotTDR(tdr.calib_knot80 , diveNo=124, what="phases", depth.lim=c(0, 120))
```

```
d2 = plotTDR(tdr.calib_knot80 , diveNo=126, what="phases", depth.lim=c(0, 120))

d3 = plotTDR(tdr.calib_knot80 , diveNo=128, what="phases", depth.lim=c(0, 120))

d4 = plotTDR(tdr.calib_knot80 , diveNo=129, what="phases", depth.lim=c(0, 120))

d5 = plotTDR(tdr.calib_knot80 , diveNo=130, what="phases", depth.lim=c(0, 120))

d6 = plotTDR(tdr.calib_knot80 , diveNo=132, what="phases", depth.lim=c(0, 120))
```

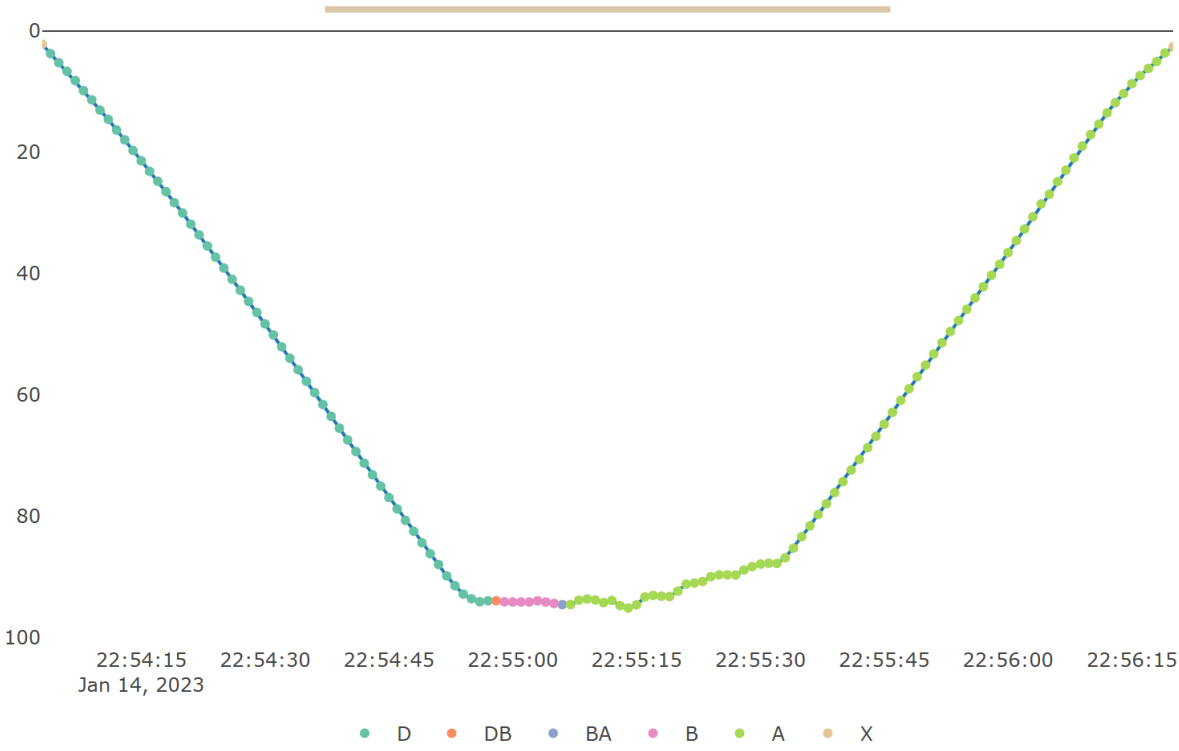
```
saveWidget(d1, "plotd1.html", selfcontained = TRUE)
webshot("plotd1.html", "plotd1.png", vwidth = 800, vheight = 600, zoom = 2)
```



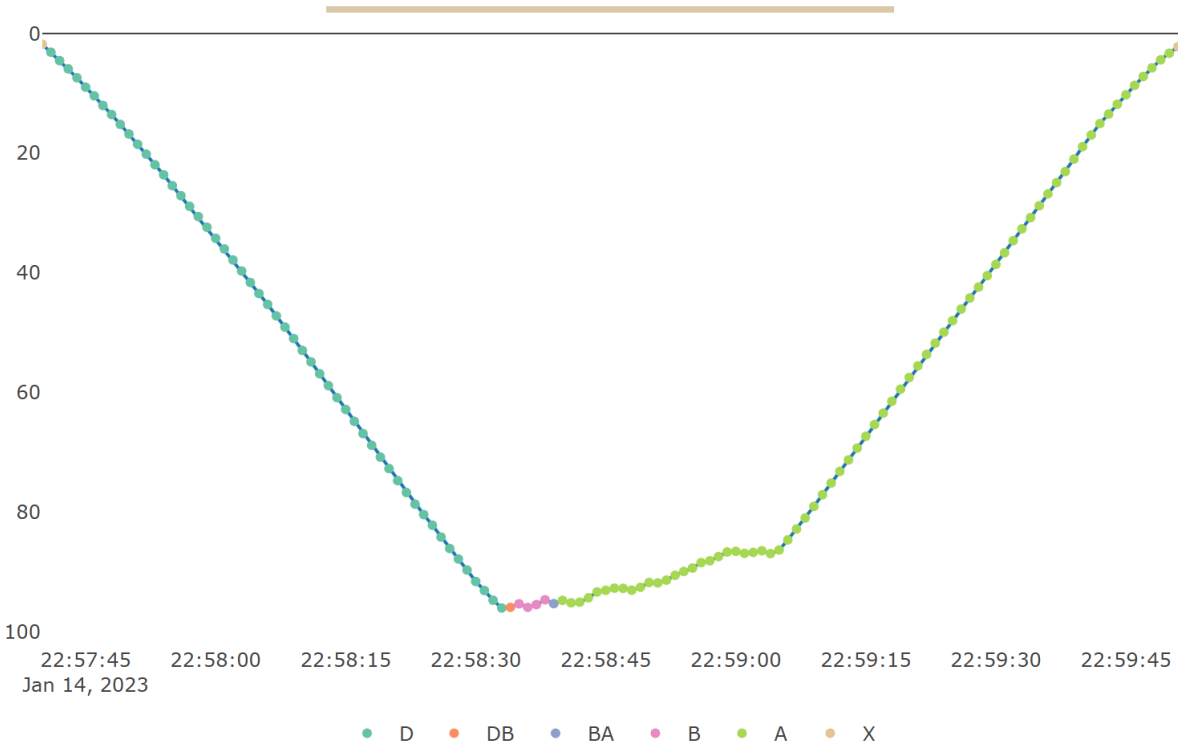

```
saveWidget(d2, "plotd2.html", selfcontained = TRUE)
webshot("plotd2.html", "plotd2.png", vwidth = 800, vheight = 600, zoom = 2)
```



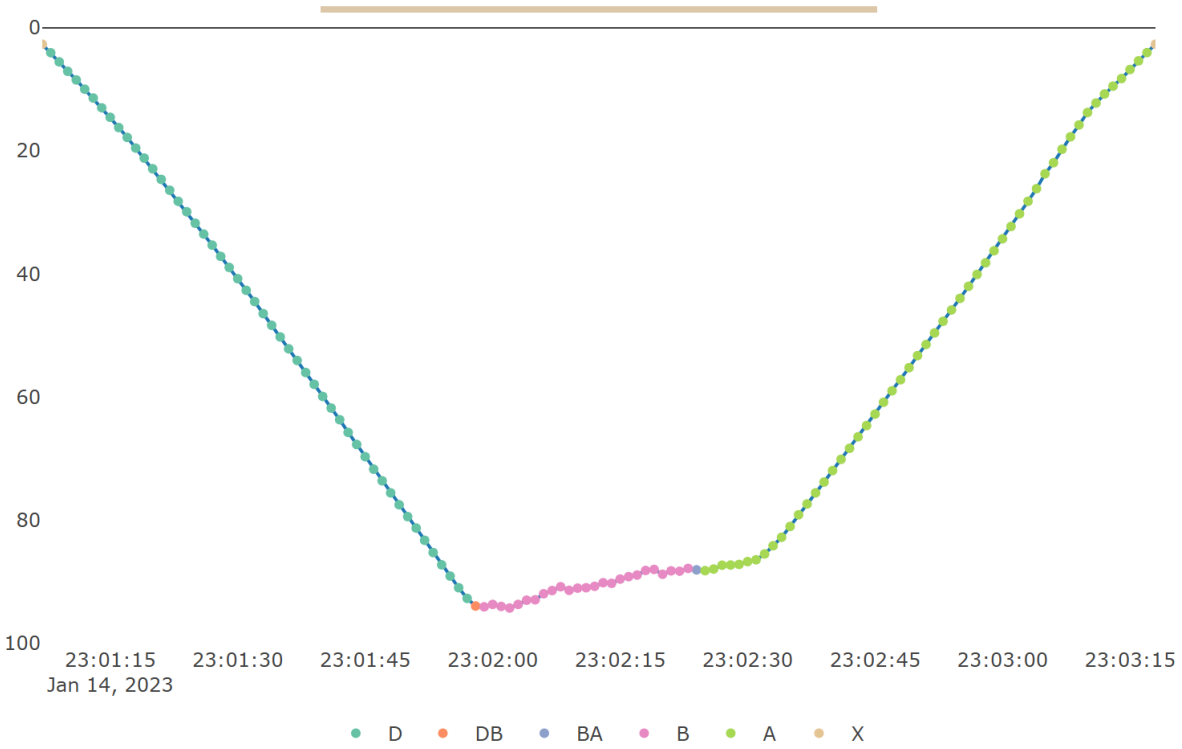

```
saveWidget(d3, "plotd3.html", selfcontained = TRUE)
webshot("plotd3.html", "plotd3.png", vwidth = 800, vheight = 600, zoom = 2)
```



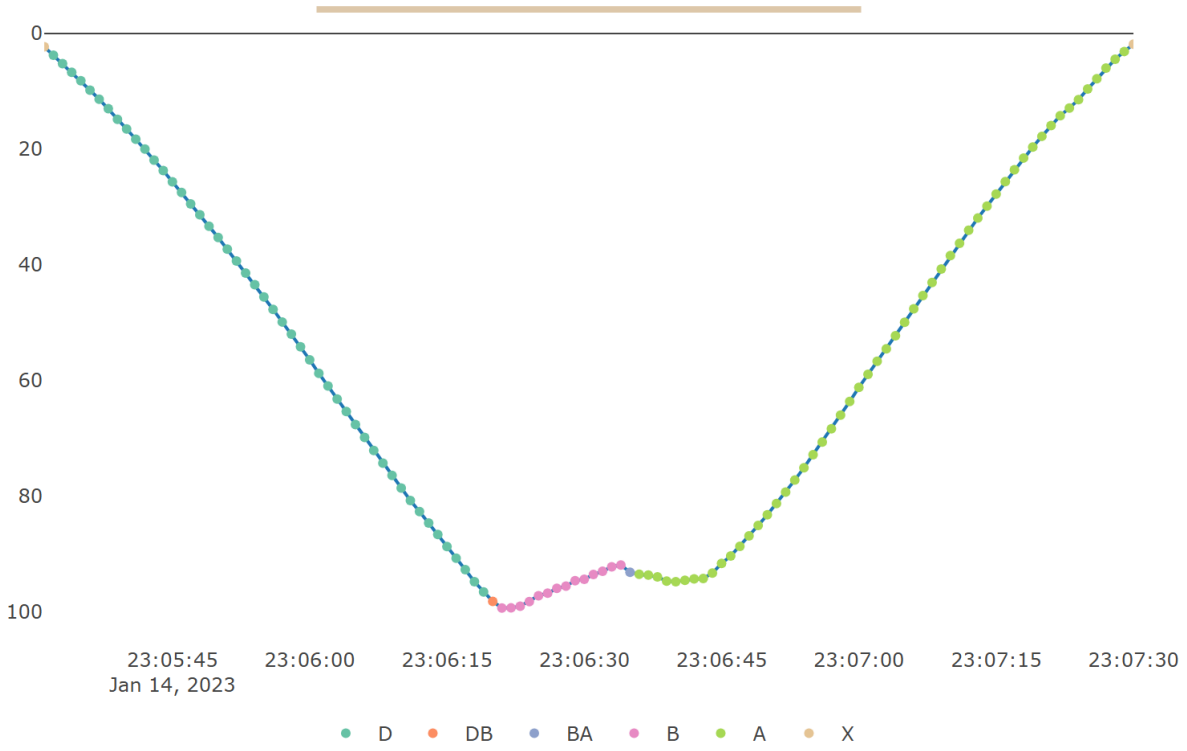

```
saveWidget(d4, "plotd4.html", selfcontained = TRUE)
webshot("plotd4.html", "plotd4.png", vwidth = 800, vheight = 600, zoom = 2)
```



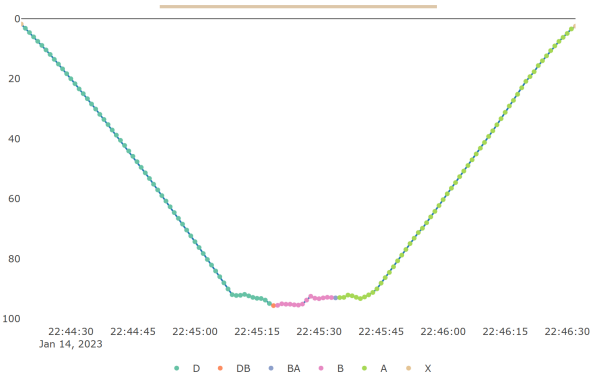

```
saveWidget(d5, "plotd5.html", selfcontained = TRUE)
webshot("plotd5.html", "plotd5.png", vwidth = 800, vheight = 600, zoom = 2)
```

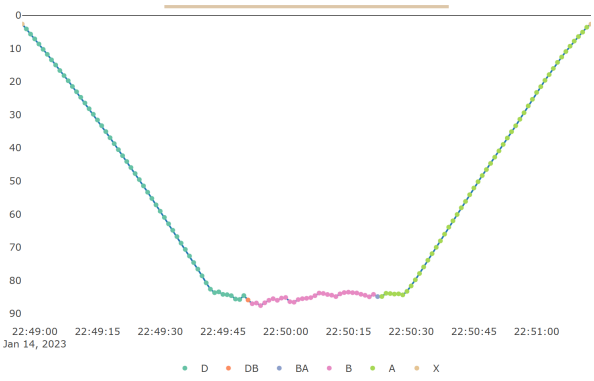


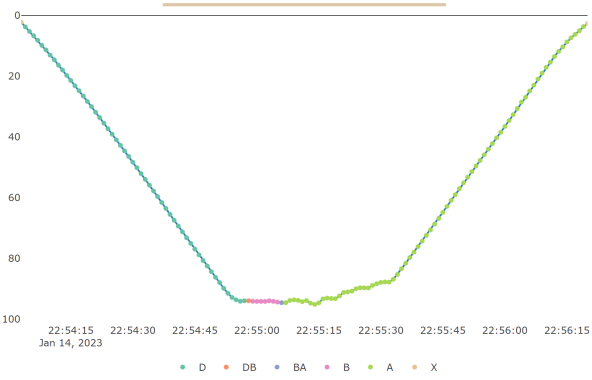

```
saveWidget(d6, "plotd6.html", selfcontained = TRUE)
webshot("plotd6.html", "plotd6.png", vwidth = 800, vheight = 600, zoom = 2)
```

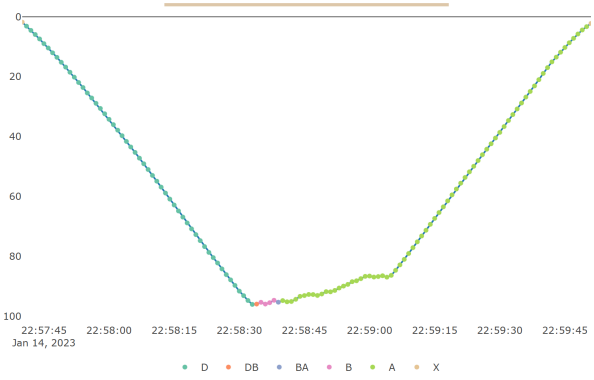


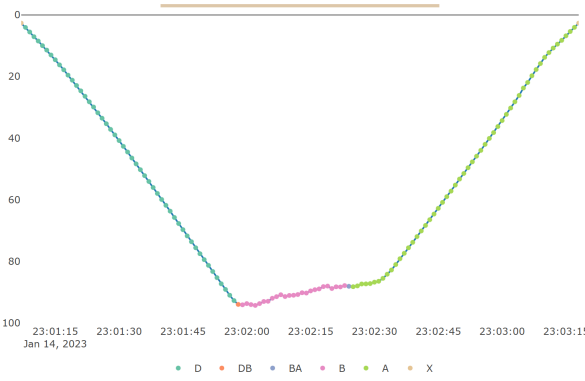

```r
# Load and wrap PNGs as grobs
grobs <- lapply(paste0("plotd", 1:6, ".png"), function(x) {
  rasterGrob(readPNG(x), interpolate = TRUE)
})

# Arrange with minimal spacing in a 2×3 grid
grid.arrange(
  grobs = grobs,
  nrow = 3, ncol = 2,
  top = NULL,
  bottom = NULL,
  left = NULL,
  right = NULL,
  padding = unit(0, "line") # Removes padding between plots
)
```

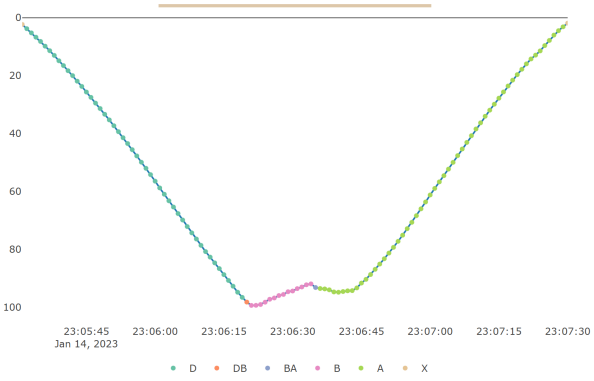

0
20
40
60
80
100
22:44:30
Jan 14, 2023
22:44:45
22:45:00
22:45:15
22:45:30
22:45:45
22:46:00
22:46:15
22:46:30
D
DB
BA
B
A
X


0
10
20
30
40
50
60
70
80
90
22:49:00
Jan 14, 2023
22:49:15
22:49:30
22:49:45
22:50:00
22:50:15
22:50:30
22:50:45
22:51:00
D
DB
BA
B
A
X


0
20
40
60
80
100
22:54:15
Jan 14, 2023
22:54:30
22:54:45
22:55:00
22:55:15
22:55:30
22:55:45
22:56:00
22:56:15
D
DB
BA
B
A
X


0
20
40
60
80
100
22:57:45
Jan 14, 2023
22:58:00
22:58:15
22:58:30
22:58:45
22:59:00
22:59:15
22:59:30
22:59:45
D
DB
BA
B
A
X


0
20
40
60
80
100
23:01:15
Jan 14, 2023
23:01:30
23:01:45
23:02:00
23:02:15
23:02:30
23:02:45
23:03:00
23:03:15
D
DB
BA
B
A
X


0
20
40
60
80
100
23:05:45
Jan 14, 2023
23:06:00
23:06:15
23:06:30
23:06:45
23:07:00
23:07:15
23:07:30
D
DB
BA
B
A
X