Package 'ConconiAnaerobicThresholdTest'

January 20, 2025

Type Package

Title Conconi Estimate of Anaerobic Threshold from a TCX File

Version 1.0.0

Maintainer Levi Waldron <lwaldron.research@gmail.com>

Description Analyzes data from a Conconi et al. (1996) <doi:10.1055/s-2007-972887> treadmill fitness test where speed is augmented by a constant amount every set number of seconds to estimate the anaerobic (lactate) threshold speed and heart rate. It reads a TCX file, allows optional removal observations from before and after the actual test, fits a change-point linear model where the change-point is the estimate of the lactate threshold, and plots the data points and fit model. Details of administering the fitness test are provided in the package vignette. Functions work by default for Garmin Connect TCX exports but may require additional data preparation for heart rate, time, and speed data from other sources.

License GPL (>= 3)

Depends dplyr, ggplot2

Imports trackeR, SiZer, methods

Encoding UTF-8

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/waldronlab/ConconiAnaerobicThresholdTest

NeedsCompilation no

Author Levi Waldron [aut, cre]

Repository CRAN

Date/Publication 2024-01-22 17:22:49 UTC

Contents

`fitmodel` *Fit piecewise linear model*

Description

Fit piecewise linear model

Usage

```
fitmodel(dat, alldata = FALSE, textsize = 5, title = "")
```

Arguments

`dat`	data.frame output by the prepdata() function
`alldata`	If FALSE (default), only the final 5 heart rate measurements of each step are used to fit the changepoint model. If TRUE, all data are used.
`textsize`	size of the breakpoint speed & pace text printed on plot (default: 5)
`title`	title of plot (default: ")

Value

creates a plot showing the piecewise fit and breakpoint

Examples

```
# Note, files in this package are gzipped to save space. TCX files exported
# from Garmin Connect or others will not have the `.gz` extension and you
# should not use `gzfile()`.
fname = system.file(file = "extdata/2023-09-15.tcx.gz", package = "ConconiAnaerobicThresholdTest")
x1 <- prepdata(gzfile(fname), startminutes = 23.8, endminutes = 40.1,
         useDeviceSpeed = FALSE)
fitmodel(x1)
```

`prepdata` *Load, trim, fit, and display model*

Description

Load, trim, fit, and display model

Usage

```
prepdata(
  fname,
  startminutes = 0,
  endminutes = 1000,
  speedmin = 6,
  speedstep = 1,
  timestep = 1.5,
  useDeviceSpeed = FALSE
)
```

Arguments

fname	Path to the tcx file
startminutes	Time (default: 0 minutes) at the start of the first step
endminutes	Time (default: 1000, in minutes) at the end of the last step
speedmin	(default: 6 km/h) Speed of the first step (set on treadmill)
speedstep	(default: 1 km/h) Speed increment of each step
timestep	(default: 1.5 minutes) Length of time of each step in minutes
useDeviceSpeed	(default: FALSE) If TRUE, use the speed as returned by the device instead of the manually-set step speeds

Details

Actually you don't need to import a TCX file, what matters for the 'fitmodel()' function is that the data.frame has columns 'time', 'heart_rate', and optionally 'speed'.

If you import a TCX file that is not from Garmin, you may need to rename the column containing heart rate to 'heart_rate' and the column containing time to 'time'. The 'time' column should be in seconds or a format that can be coerced to seconds using 'as.numeric()', such as the POSIXct/POSIXlt formats that most services likely provide. If 'useDeviceSpeed' is FALSE, then the speed column should be 'speed'.

Value

a data.frame with early and late times potentially trimmed, and speed potentially over-ridden with manually set step values.

Examples

```
# Note, files in this package are gzipped to save space. TCX files exported
# from Garmin Connect or others will not have the `.gz` extension and you
# should not use `gzfile()`.
fname = system.file(file = "extdata/2023-09-15.tcx.gz",
                    package = "ConconiAnaerobicThresholdTest")
# These plots can help get the start and end time correct.
x0 <- prepdata(gzfile(fname), useDeviceSpeed = TRUE)
oldpar <- par(mfrow=c(2, 2))
```

```
plot(x0$minutes, x0$speed)
plot(x0$minutes, x0$cadence_running)
plot(x0$minutes, x0$heart_rate)
# Once you have start and end times correct, set useDeviceSpeed = FALSE
# if speeds were set manually on the treadmill.
x1 <- prepdata(gzfile(fname), startminutes = 23.8, endminutes = 40.1,
         useDeviceSpeed = FALSE)
par(mfrow=c(2, 2))
plot(x1$minutes, x1$speed)
plot(x1$minutes, x1$cadence_running)
plot(x1$minutes, x1$heart_rate)
par(oldpar)
```

Index