

# Package ‘RHRT’

June 29, 2021

**Type** Package

**Title** Heart Rate Turbulence Analysis

**Version** 1.0.1

**Date** 2021-06-17

**BugReports** <https://github.com/VBlesius/RHRT/issues>

**Description** Methods to scan RR interval data for Premature Ventricular Complexes (PVCs) and parameterise and plot the resulting Heart Rate Turbulence (HRT). The methodology of HRT analysis is based on the original publication by Schmidt et al. <[doi:10.1016/S0140-6736\(98\)08428-1](https://doi.org/10.1016/S0140-6736(98)08428-1)> and extended with suggestions from <[doi:10.1088/1361-6579/ab98b3](https://doi.org/10.1088/1361-6579/ab98b3)>.

**Depends** R (>= 3.5.0)

**License** GPL-2

**LazyData** TRUE

**Imports** methods, graphics, utils

**RoxygenNote** 7.1.1

**Suggests** knitr, rmarkdown, testthat, matrixStats

**VignetteBuilder** knitr

**Collate** 'HRT.R' 'avHRT.R' 'HRTLlist.R' 'Testdata-description.R'  
'constants.R' 'conversion\_to\_HRT.R' 'roll.R'

**NeedsCompilation** no

**Author** Valeria Blesius [aut, cre] (<<https://orcid.org/0000-0002-2391-242X>>)

**Maintainer** Valeria Blesius <[valeria.blesius@mni.thm.de](mailto:valeria.blesius@mni.thm.de)>

**Repository** CRAN

**Date/Publication** 2021-06-29 06:00:02 UTC

## R topics documented:

avHRT . . . . .	2
calcAvHRT . . . . .	4
calcHRTParams . . . . .	5

calcTO . . . . .	6
calcTS . . . . .	7
checkAnnotations . . . . .	7
checkForHRT . . . . .	8
checkInput . . . . .	8
checkValidity . . . . .	9
cleanInput . . . . .	9
getHRTParams . . . . .	10
getHRTs . . . . .	11
getPositions . . . . .	11
getResults . . . . .	12
getRRs . . . . .	13
HRT . . . . .	14
HRTList . . . . .	15
plot,HRT-method . . . . .	16
plot,HRTList-method . . . . .	18
roll . . . . .	19
testdataLong . . . . .	20
testdataLong_Ann . . . . .	20
vectorToHRT . . . . .	20

## Index 22

---

avHRT	<i>S4 class to represent an avHRT object</i>
-------	--

---

### Description

This class extends the HRT class. An avHRT is the average of an HRTList and saves the way in which it was calculated.

### Usage

```
## S4 method for signature 'avHRT'
initialize(
  .Object,
  av = mean,
  orTO = "avAfter",
  orTS = "avBefore",
  pTO = NA_real_,
  pTS = NA_real_,
  pTT = NA_real_,
  pnTS = NA_real_,
  nRMSSD = NA_real_,
  couplRR = NA_real_,
  compRR = NA_real_,
  preRRs = NA_real_,
  postRRs = NA_real_
)
```

**Arguments**

.Object	The name of the class
av	(Function) Type of averaging, either mean or median
orTO	(Character) Order in which TO was calculated, either "avAfter" (assessment of parameter and averaging) or "avBefore" (averaging of the VPCSs and assessment of parameter)
orTS	(Character) Order in which TS was calculated, either "avAfter" (assessment of parameter and averaging) or "avBefore" (averaging of the VPCSs and assessment of parameter)
pTO	(Numeric) p-value of t-test checking the validity of TO
pTS	(Numeric) p-value of t-test checking the validity of TS
pTT	(Numeric) p-value of t-test checking the validity of TT
pnTS	(Numeric) p-value of t-test checking the validity of normalised TS
nRMSSD	(Numeric) RMSSD normalised to HR
coupIRR	(Numeric) Coupling interval
compRR	(Numeric) Compensatory interval
preRRs	(Numeric vector) Preceding intervals
postRRs	(Numeric vector) Following intervals

**Value**

(avHRT) A new avHRT object

**Slots**

av	(Function) Type of averaging, either mean or median
orTO	(Character) Order in which TO was calculated, either "avAfter" (assessment of parameter and averaging) or "avBefore" (averaging of the VPCSs and assessment of parameter)
orTS	(Character) Order in which TS was calculated, either "avAfter" (assessment of parameter and averaging) or "avBefore" (averaging of the VPCSs and assessment of parameter)
pTO	(Numeric) p-value of t-test checking the validity of TO
pTS	(Numeric) p-value of t-test checking the validity of TS
pTT	(Numeric) p-value of t-test checking the validity of TT
pnTS	(Numeric) p-value of t-test checking the validity of normalised TS

---

 calcAvHRT

*Calculate an avHRT object*


---

### Description

For each index the average of the intervals across all HRTs in the HRTList is calculated and the averaged HRT returned. The type of averaging, the order of HRT parameter assessment and interval lengths for normalising TS can be passed.

### Usage

```
calcAvHRT(
  HRTListObj,
  av = mean,
  orTO = "avAfter",
  orTS = "avBefore",
  IL = HRTListObj@IL,
  normIL = c_normIL,
  normHallstrom = TRUE,
  coTO = COTO,
  coTS = COTS,
  coTT = COTT
)

## S4 method for signature 'HRTList'
calcAvHRT(
  HRTListObj,
  av = mean,
  orTO = "avAfter",
  orTS = "avBefore",
  IL = HRTListObj@IL,
  normIL = c_normIL,
  normHallstrom = TRUE,
  coTO = COTO,
  coTS = COTS,
  coTT = COTT
)
```

### Arguments

HRTListObj	HRTList object
av	(Function) Type of averaging the VPCSs, either mean or median
orTO	(Character) Order in which TO was calculated, either "avAfter" (assessment of parameter and averaging) or "avBefore" (averaging of the VPCSs and assessment of parameter)

orTS	(Character) Order in which TS was calculated, either "avAfter" (assessment of parameter and averaging) or "avBefore" (averaging of the VPCSs and assessment of parameter)
IL	(Numeric) The overall arithmetic mean of the interval length of the measurement to normalise TS
normIL	(Numeric) The interval length to which TS should be normalised
normHallstrom	(Boolean) Should the normalisation of Hallstrom be used?
coTO	(Numeric) Cut-off value for TO
coTS	(Numeric) Cut-off value for TS and nTS
coTT	(Numeric) Cut-off value for TT

### Details

To eliminate other RR variability TS is commonly assessed after averaging the VPCSs. TO is commonly first calculated from the single VPCS and then averaged. (See 'Heart Rate Turbulence: Standards of Measurement, Physiological Interpretation, and Clinical Use, Axel Bauer et al., Journal of the American College of Cardiology, Volume 52, Issue 17, Pages 1353-1365')

### Value

(avHRT) The avHRT object of the given HRTLlist

### Examples

```
# You need an HRTLlist
hrtl <- vectorToHRT(testdataLong, testdataLong_Ann)

# Recalculate the avHRT with different normalisation
calcAvHRT(hrtl, normIL = 1000, normHallstrom = FALSE)

# Recalculate the avHRT based on a different calculation order
calcAvHRT(hrtl, orTO = "avBefore", orTS = "avAfter")

# Set custom parameter cut-offs for the reliability check
## You should keep in mind to give the same cut-offs when calling getResults()
calcAvHRT(hrtl, coTO = 0.022, coTS = 1.42, coTT = 12)
```

---

calcHRTParams

*Calculate HRT parameters*

---

### Description

Calculates all HRT parameters needed for an HRT object and saves them in the corresponding slots.

**Usage**

```
calcHRTParams(HRTObj, IL = c_normIL, normIL = c_normIL)

## S4 method for signature 'HRT'
calcHRTParams(HRTObj, IL = c_normIL, normIL = c_normIL)
```

**Arguments**

HRTObj	(HRT) The HRT object of which the parameters should be calculated
IL	(Numeric) The overall arithmetic mean of the interval length of the measurement to normalise TS
normIL	(Numeric) The interval length to which TS should be normalised

**Details**

This method is a wrapper for the methods calcTO and calcTS.

**Value**

(HRT) An HRT object with (re)calculated HRT parameters

---

calcTO	<i>Calculate TO parameters</i>
--------	--------------------------------

---

**Description**

Calculates the TO parameters and saves it in the corresponding slot

**Usage**

```
calcTO(HRTObj)

## S4 method for signature 'HRT'
calcTO(HRTObj)
```

**Arguments**

HRTObj	(HRT) The HRT object, for which TO should be calculated
--------	---

**Value**

(HRT) An HRT object with (re)calculated TO

---

calcTS	<i>Calculate TS parameters</i>
--------	--------------------------------

---

**Description**

Calculates all TS parameters (TS itself, its index TT (turbulence timing) and the intercept for the plot) and saves them in the corresponding slots. Can also calculate normalised TS and intercept.

**Usage**

```
calcTS(HRTObj, normalising = FALSE, IL = c_normIL, normIL = c_normIL)

## S4 method for signature 'HRT'
calcTS(HRTObj, normalising = FALSE, IL = c_normIL, normIL = c_normIL)
```

**Arguments**

HRTObj	(HRT) The HRT object, for which TS should be calculated
normalising	(Boolean) Should the normalised TS be calculated?
IL	(Numeric) The overall arithmetic mean of the interval length of the measurement to normalise TS
normIL	(Numeric) The interval length to which TS should be normalised

**Value**

(HRT) An HRT object with (re)calculated TS+intercept or nTS+nintercept

---

checkAnnotations	<i>Checks annotations for compatibility</i>
------------------	---

---

**Description**

Checks annotations for compatibility

**Usage**

```
checkAnnotations(annotations, input, PVCAnn, label)
```

**Arguments**

annotations	(Character vector) Annotations matching input
input	(Numeric vector) RR intervals in ms
PVCAnn	(Character) Character that marks a VPC in the annotations
label	(Character) Name of the data given and formatted for output

**Value**

No return value, possibly throws errors/warnings

---

checkForHRT	<i>Checks RR-intervals for HRT criteria and returns an HRT object</i>
-------------	---

---

**Description**

Checks RR-intervals for HRT criteria and returns an HRT object

**Usage**

```
checkForHRT(intervals, numPreRRs = c_numPreRRs, numPostRRs = c_numPostRRs)
```

**Arguments**

intervals	(Numeric vector) RR intervals in ms
numPreRRs	(Numeric) Number of RRs before the coupling interval that are used for filtering
numPostRRs	(Numeric) Number of RRs after the compensatory interval that are used for filtering

**Value**

(HRT) A single HRT object

---

checkInput	<i>Checks data input for compatibility</i>
------------	--

---

**Description**

Checks data input for compatibility

**Usage**

```
checkInput(input, numSnippet, label)
```

**Arguments**

input	(Numeric vector) RR intervals in ms
numSnippet	(Numeric) number of RRs in the the HRT snippet
label	(Character) Name of the data given and formatted for output

**Value**

No return value, possibly throws errors/warnings

---

checkValidity	<i>Checks whether slots are set</i>
---------------	-------------------------------------

---

**Description**

Checks whether slots are set  
 Check for HRTLlist class

**Usage**

```
checkValidity(x, ...)
```

```
## S4 method for signature 'HRT'
```

```
checkValidity(x)
```

```
## S4 method for signature 'HRTLlist'
```

```
checkValidity(x, av = FALSE, pos = FALSE)
```

**Arguments**

x	HRTLlist
...	Other parameters
av	(Boolean) Should avHRT be checked?
pos	(Boolean) Should pos be checked?

**Value**

No return value, possibly throws errors/warnings  
 No return value, possibly throws errors

---

cleanInput	<i>Cleans data input for further checks or calculation</i>
------------	--

---

**Description**

Cleans data input for further checks or calculation

**Usage**

```
cleanInput(input)
```

**Arguments**

input	(Numeric vector) RR intervals in ms
-------	-------------------------------------

**Value**

(numeric vector) Input vector without possible bias

---

getHRTParams

*Extracts all values of a special slot out of a HRTList*

---

**Description**

Extracts all values of the given slot in each HRT of the HRTList and returns them in a list

**Usage**

```
getHRTParams(HRTListObj, sl)

## S4 method for signature 'HRTList'
getHRTParams(HRTListObj, sl)
```

**Arguments**

HRTListObj	HRTList object
sl	(Character) Value of a slot saved by an HRT object

**Value**

(numeric vector or list) Vector or list of the numerics stored in the given slot

**Examples**

```
# You need an HRTList
hrtl <- vectorToHRT(testdataLong, testdataLong_Ann)

# Get all TOs of the HRTs in your HRTList
getHRTParams(hrtl, "TO")

# You can access all slots in the HRTs
getHRTParams(hrtl, "intercept")

# If you access slots that include more than one numeric, the function returns a list
getHRTParams(hrtl, "preRRs")
```

---

getHRTs	<i>Finds HRTs</i>
---------	-------------------

---

**Description**

Scans for HRTs in the given vector and returns an HRTLlist object.

**Usage**

```
getHRTs(
  intervals,
  annotations = NULL,
  PVCAnn = "V",
  numPreRRs = c_numPreRRs,
  numPostRRs = c_numPostRRs,
  numSnippet
)
```

**Arguments**

intervals	(Numeric vector) RR intervals in ms
annotations	(Character vector) Annotations matching input
PVCAnn	(Character) Character that marks a VPC in the annotations
numPreRRs	(Numeric) Number of RRs before the coupling interval that are used for filtering
numPostRRs	(Numeric) Number of RRs after the compensatory interval that are used for filtering
numSnippet	(Numeric) Number of RRs in the HRT snippet

**Value**

(HRTLlist) HRTLlist with only pos and HRTs set

---

getPosition	<i>Get positions of PVCs</i>
-------------	------------------------------

---

**Description**

Returns the positions of all ventricular premature complexes (VPCs) and accordingly the coupling intervals that were found in the given vector when the HRTLlist was created.

**Usage**

```
getPosition(HRTLlistObj)

## S4 method for signature 'HRTLlist'
getPosition(HRTLlistObj)
```

**Arguments**

HRTListObj (HRTList object)

**Value**

No return value, possibly throws errors/warnings

---

getResults	<i>Get averaged HRT parameters</i>
------------	------------------------------------

---

**Description**

Returns the HRT parameters of the HRTList. Turbulence onset is calculated for each HRT object and then averaged, turbulence slope is calculated via averaging the intervals of all HRT objects to one HRT object and then estimating the maximal slope.

**Usage**

```
getResults(
  HRTListObj,
  type = "class",
  TT = FALSE,
  nTS = FALSE,
  safe = TRUE,
  pmax = 0.05,
  num = FALSE,
  coTO = COTO,
  coTS = COTS,
  coTT = COTT
)

## S4 method for signature 'HRTList'
getResults(
  HRTListObj,
  type = "class",
  TT = FALSE,
  nTS = FALSE,
  safe = TRUE,
  pmax = 0.05,
  num = FALSE,
  coTO = COTO,
  coTS = COTS,
  coTT = COTT
)
```

**Arguments**

HRTListObj	HRTList object
type	(String) Determining the amount of output: 'class' gives the HRT class, 'parameter' the parameter values and 'full' additionally the p-values describing parameter reliability
TT	(Boolean) Should TT be given?
nTS	(Boolean) Should the normalised TS (nTS) be given or used for the determination of the HRT class?
safe	(Boolean) Should all values be given regardless of reliability checks? Note, that 'safe' is ignored when the type is 'full'.
pmax	(Numeric) The significance level
num	(Boolean) Should the results be numeric? This forces the results to stay numeric, but sets not reliable values as NA, if 'safe' is TRUE. Forced numeric values cannot be combined with type 'class'.
coTO	(Numeric) Cut-off value for TO
coTS	(Numeric) Cut-off value for TS and nTS
coTT	(Numeric) Cut-off value for TT

**Value**

(Named vector, character or numeric) Either HRT classes, HRT parameter values and/or p-values

**Examples**

```
# You need an HRTList
hrtl <- vectorToHRT(testdataLong, testdataLong_Ann)

# Get the HRT classes of your HRTList
getResults(hrtl)
getResults(hrtl, TT = TRUE)

# Get the HRT parameter values of your HRTList
getResults(hrtl, type = "parameter", TT = TRUE)
```

---

getRRs

*Returns the VPCS intervals in right order*


---

**Description**

Returns the VPCS intervals in right order

**Usage**

```
getRRs(HRTObj)

## S4 method for signature 'HRT'
getRRs(HRTObj)
```

**Arguments**

HRTObj            HRT

**Value**

(numeric vector) All VPCS intervals

---

HRT	<i>S4 class to represent an HRT object</i>
-----	--

---

**Description**

This class specifies an object to save the lengths of intervals surrounding a premature ventricular beat. It saves the HRT parameters turbulence onset (TO), slope (TS) and timing (TT) after calculation as well as the coefficients of an ab-line used for the plot. TS is saved after common calculation and after normalising.

**Usage**

```
## S4 method for signature 'HRT'
initialize(
  .Object,
  couplRR = NA_real_,
  compRR = NA_real_,
  preRRs = NA_real_,
  postRRs = NA_real_
)
```

**Arguments**

.Object	(Character) The name of the class
couplRR	(Numeric) Coupling interval
compRR	(Numeric) Compensatory interval
preRRs	(Numeric vector) Preceding intervals
postRRs	(Numeric vector) Following intervals

**Value**

(HRT) A new HRT object

**Slots**

coup1RR (Numeric) Coupling interval  
 compRR (Numeric) Compensatory interval  
 preRRs (Numeric vector) Preceding intervals  
 postRRs (Numeric vector) Following intervals  
 T0 (Numeric) Turbulence onset  
 TS (Numeric) Turbulence slope  
 TT (Numeric) Turbulence timing  
 intercept (Numeric) Intercept of regression line of TS  
 nTS (Numeric) Normalised Turbulence slope  
 nintercept (Numeric) Intercept of regression line of nTS

---

 HRTLlist

*S4 class to represent a list of HRT objects*


---

**Description**

This class specifies an object to save all HRT objects of a given vector. It also saves an averaged HRT for calculation of the averaged HRT parameters and plotting of all HRTs in a single plot.

**Usage**

```

## S4 method for signature 'HRTLlist'
initialize(
  .Object,
  name = NA_character_,
  IL = NA_real_,
  pos = NA_real_,
  HRTs = list(),
  avHRT = new("avHRT"),
  RMSSD = NA_real_
)

```

**Arguments**

.Object	(Character) The name of the class
name	(Character) Name of the vector if given
IL	(Numeric) Arithmetic mean of the overall interval length of the vector
pos	(Numeric vector) Positions of premature ventricular complexes in given input
HRTs	(List) All HRT objects
avHRT	(avHRT object) The average of all HRTs
RMSSD	(Numeric) Square root of the mean of the squared successive differences between adjacent intervals of the whole measurement

**Value**

(HRTLlist) A new HRTLlist object

**Slots**

name (Character) Name of the vector if given

IL (Numeric) Arithmetic mean of the overall interval length of the vector

pos (Numeric vector) Positions of premature ventricular complexes in given input

HRTs (List) All HRT objects

avHRT (avHRT object) The average of all HRTs

RMSD (Numeric) Square root of the mean of the squared successive differences between adjacent intervals of the whole measurement

**Note**

After using `vectorToHRT` all slots in the resulting HRTLlist object are set. Please do not set them manually since many functions of the HRTLlist class rely on valid values assigned to the needed slots.

---

`plot,HRT-method`

*Plot an HRT object*

---

**Description**

Plots RR-intervals saved in the HRT object and marks turbulence onset and turbulence slope.

**Usage**

```
## S4 method for signature 'HRT'
plot(
  x,
  cropped = TRUE,
  TT = FALSE,
  pch = 20,
  xlab = "# of RR interval",
  ylab = "length of RR interval (ms)",
  paramsLegend = TRUE,
  colTO = "#ec2023",
  colTS = "#006AFF",
  colTT = "#6800DE",
  add = FALSE,
  ...
)
```

**Arguments**

x	(HRT) A HRT object
cropped	(Boolean) Should the plot be cut to focus on the HRT parameters? To show all points use FALSE.
TT	(Boolean) Should Turbulence timing be marked?
pch	(Numeric) Plotting character, for other options see graphics::var
xlab	(Character) Label for the x axis
ylab	(Character) Label for the y axis
paramsLegend	(Boolean) Should the parameter values of the HRT be plotted?
colTO	(Character) Colour used to highlight TO
colTS	(Character) Colour used to highlight TS
colTT	(Character) Colour used to highlight TT
add	(Boolean) Should the given HRT be added to a plot?
...	Other arguments in tag = value form. See graphics::par for more information.

**Value**

No return value

**Note**

Please note that some graphics parameters (par) cannot be modified, since they are needed to be set inside the function.

**Examples**

```
# You need an HRT object
hrt <- vectorToHRT(testdataLong, testdataLong_Ann)@HRTs[[1]]

# Plot your HRT and zoom out
plot(hrt, cropped = FALSE)

# Include TT and customise it
plot(hrt, TT = TRUE, colTT = "green", pch = 7)

# Use standard graphics parameters
## Note: Some parameters are used inside the function and cannot be set
plot(hrt, TT = TRUE, main = "Example plot", bty = "n", cex.lab = 1.2)
```

---

plot,HRTList-method *Plot an HRTList object*

---

### Description

Plots RR-intervals saved in the HRT objects, especially the avHRT object, and marks the HRT parameters.

### Usage

```
## S4 method for signature 'HRTList'
plot(
  x,
  cropped = TRUE,
  TT = FALSE,
  pch = 20,
  xlab = "# of RR interval",
  ylab = "length of RR interval (ms)",
  paramsLegend = TRUE,
  colTO = "#ec2023",
  colTS = "#006AFF",
  colTT = "#6800DE",
  ...
)
```

### Arguments

x	HRTList
cropped	(Boolean) Should the plot be cut to focus on the HRT parameters? To show all points use FALSE.
TT	(Boolean) Should Turbulence timing be marked?
pch	(Numeric) Plotting character, for other options see graphics::var
xlab	(Character) Label for the x axis
ylab	(Character) Label for the y axis
paramsLegend	(Boolean) Should the parameter values of the HRT be plotted?
colTO	(Character) Colour used to highlight TO
colTS	(Character) Colour used to highlight TS
colTT	(Character) Colour used to highlight TT
...	Other arguments in tag = value form

### Value

No return value

**Note**

Please note that some graphics parameters (par) cannot be modified, since they are needed to be set inside the function.

**Examples**

```
# You need an HRTList
hrtl <- vectorToHRT(testdataLong, testdataLong_Ann)

# Plot your HRTList and zoom out
plot(hrtl, cropped = FALSE)

# Include TT and customise it
plot(hrtl, TT = TRUE, colTT = "green", pch = 7)

# Use standard graphics parameters
## Note: Some parameters are used inside the function and cannot be set
plot(hrtl, TT = TRUE, main = "Example plot", bty = "n", cex.lab = 1.2)
```

---

roll	<i>Apply method on sliding window</i>
------	---------------------------------------

---

**Description**

Applies a given function on a vector by rolling over it with a sliding window mechanism.

**Usage**

```
roll(intervals, width, fun, ...)
```

**Arguments**

intervals	vector
width	window size
fun	function to be applied
...	additional arguments for FUN

**Details**

This method was inspired by the function "wapply" by A. N. Spiess, University Hospital Hamburg-Eppendorf (<https://rmazing.wordpress.com/2013/04/23/wapply-a-faster-but-less-functional-rollapply-for-vector-sets/>), but adjusted for this package to speed it up.

**Value**

(list) List with return values of fun for each window

---

testdataLong	<i>Long term data</i>
--------------	-----------------------

---

**Description**

Artificial dummy interval data: This dataset represents a long-term measurement and includes 15 VPCSs that fit the HRT filter rules.

**Usage**

testdataLong

**Format**

A numeric vector.

---

testdataLong_Ann	<i>Long term data annotations</i>
------------------	-----------------------------------

---

**Description**

Artificial dummy interval data: This dataset contains the annotations matching testdataLong.

**Usage**

testdataLong\_Ann

**Format**

A vector of characters.

---

vectorToHRT	<i>Convert a vector to HRTList</i>
-------------	------------------------------------

---

**Description**

Scans for heart rate turbulence in a vector of RR-intervals and returns an HRTList object including all found HRT objects. The HRT criteria used were published by Schmidt et al. (more information can be found in the vignette.)

**Usage**

```
vectorToHRT(
  input,
  annotations = NULL,
  PVCAnn = "V",
  normIL = c_normIL,
  normHallstrom = TRUE,
  numPreRRs = c_numPreRRs,
  numPostRRs = c_numPostRRs,
  inputName = as.character(NA),
  minHRT = 5,
  cleaning = TRUE
)
```

**Arguments**

input	(Numeric vector) RR intervals in ms
annotations	(Character vector) Annotations matching input
PVCAnn	(Character) Character that marks a VPC in the annotations
normIL	(Numeric) The interval length to which TS should be normalised
normHallstrom	(Boolean) Should the normalisation of Hallstrom be used?
numPreRRs	(Numeric) Number of RRs before the coupling interval that are used for filtering
numPostRRs	(Numeric) Number of RRs after the compensatory interval that are used for filtering
inputName	(String) Name of the data
minHRT	(Numeric) Minimal number of HRTs that are needed to create an HRTList object
cleaning	(Boolean) Should the input be roughly cleaned from artefacts before calculating IL and RMSSD?

**Value**

(HRTList) An HRTList object

**Examples**

```
# You can use annotations to give the VPC indices
# Without annotation data RHRT will find VPCs based on common filtering criteria
vectorToHRT(testdataLong, annotations = testdataLong_Ann, PVCAnn = "V")

# Find HRTs with a broader range of sinus beats before and after the VPCs
vectorToHRT(testdataLong, inputName = "Dummy Measurement", numPreRRs = 10, numPostRRs = 20)

# Adjust the normalisation parameters
vectorToHRT(testdataLong, testdataLong_Ann, normHallstrom = FALSE, normIL = 900)
```

# Index

## \* datasets

testdataLong, [20](#)  
testdataLong\_Ann, [20](#)

avHRT, [2](#)

calcAvHRT, [4](#)  
calcAvHRT, HRTList-method (calcAvHRT), [4](#)  
calcHRTParams, [5](#)  
calcHRTParams, HRT-method  
(calcHRTParams), [5](#)  
calcT0, [6](#)  
calcT0, HRT-method (calcT0), [6](#)  
calcTS, [7](#)  
calcTS, HRT-method (calcTS), [7](#)  
checkAnnotations, [7](#)  
checkForHRT, [8](#)  
checkInput, [8](#)  
checkValidity, [9](#)  
checkValidity, HRT-method  
(checkValidity), [9](#)  
checkValidity, HRTList-method  
(checkValidity), [9](#)  
cleanInput, [9](#)

getHRTParams, [10](#)  
getHRTParams, HRTList-method  
(getHRTParams), [10](#)  
getHRTs, [11](#)  
getPosition, [11](#)  
getPosition, HRTList-method  
(getPosition), [11](#)  
getResults, [12](#)  
getResults, HRTList-method (getResults),  
[12](#)  
getRRs, [13](#)  
getRRs, HRT-method (getRRs), [13](#)

HRT, [14](#)  
HRTList, [15](#)

initialize, avHRT-method (avHRT), [2](#)  
initialize, HRT-method (HRT), [14](#)  
initialize, HRTList-method (HRTList), [15](#)

plot, HRT-method, [16](#)  
plot, HRTList-method, [18](#)

roll, [19](#)

testdataLong, [20](#)  
testdataLong\_Ann, [20](#)

vectorToHRT, [20](#)