

Package: PhysioECG (via r-universe)

October 3, 2026

Title ECG Analysis Functions for PhysioExperiment Objects

Version 0.4.3

Description Provides electrocardiography (ECG) analysis functions for PhysioExperiment objects. Includes R-peak detection (adaptive Pan-Tompkins), RR interval computation, HRV time-domain metrics (SDNN, RMSSD, pNN50), HRV frequency-domain analysis (VLF, LF, HF power), nonlinear HRV (Poincare, Sample Entropy, DFA), ECG morphology analysis (QRS boundaries, QT/QTc intervals), and signal quality assessment with ectopic beat correction.

Depends R (>= 4.2), PhysioExperiment (>= 2.0.0)

Imports SummarizedExperiment, S4Vectors, stats, graphics, ggplot2

Suggests PhysioCore, knitr, rmarkdown, signal, RHRV, testthat (>= 3.2.0)

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Collate 'ecg-data.R' 'ecg-peaks.R' 'ecg-morphology.R' 'ecg-rhythm.R'
'ecg-arrhythmia.R' 'ecg-hrv-time.R' 'ecg-detrend.R'
'ecg-hrv-freq.R' 'ecg-hrv-tv.R' 'ecg-hrv-nonlinear.R'
'ecg-hrv-geometric.R' 'ecg-hrr.R' 'ecg-edr.R' 'resp.R' 'ppg.R'
'ecg-hrt-prsa.R' 'ecg-pipeline.R' 'ecg-plot.R' 'ecg-quality.R'
'zzz.R'

RoxygenNote 7.3.3

URL <https://github.com/x-biosignal/PhysioECG>,
<https://x-biosignal.r-universe.dev/PhysioECG>,
<https://x-biosignal.github.io/PhysioECG/>

BugReports <https://github.com/x-biosignal/PhysioECG/issues>

Config/pak/sysreqs zlib1g-dev

Repository <https://x-biosignal.r-universe.dev>

Date/Publication 2026-10-03 13:11:44 UTC

RemoteUrl <https://github.com/x-biosignal/PhysioECG>

RemoteRef HEAD

RemoteSha 5164ded72b2faa71eda9a72446ade7334138f02d

Contents

breathingRate	3
dfaThresholdCrossings	4
ecgAnalyze	5
ecgBaselineCorrect	5
ecgBeatSQI	7
ecgBeatSQIgate	8
ecgDelineate	9
ecgDetectAF	10
ecgDetectPVC	12
ecgDetectRpeaks	13
ecgDFA	15
ecgDFArolling	16
ecgEctopicBurden	17
ecgEDR	17
ecgHRR	19
ecgHRT	20
ecgHRVasymmetry	21
ecgHRVautonomic	23
ecgHRVfreq	24
ecgHRVgeometric	26
ecgHRVnonlinear	27
ecgHRVpoincare	28
ecgHRVsegmented	29
ecgHRVtime	30
ecgHRVtimevarying	31
ecgIntervals	32
ecgProcess	34
ecgPRSA	35
ecgQualityCheck	36
ecgRRcorrect	37
ecgRRintervals	38
ecgRSA	39
ecgSampleEntropy	40
ecgSignalQuality	41
make_ecg	42
make_ecg_irregular	43
make_ecg_noisy	44
make_ecg_pqrst	45
make_ppg	46
perfusionIndex	47

plotEcgWave	48
plotHRVpsd	49
plotPoincare	50
plotTachogram	51
ppgDetectPulses	51
ppgQuality	53
ppgRespiration	53
pulseRateVariability	54
pulseWaveFeatures	55
respiratoryOnsets	56
ripCalibrate	57
smoothnessPriorsDetrend	59
tidalMetrics	60
ventilatoryThreshold	61
Index	62

breathingRate	<i>Estimate breathing rate</i>
---------------	--------------------------------

Description

Estimates rate from complete inter-trough breath periods or from the dominant respiratory-band spectral peak.

Usage

```
breathingRate(  
  x,  
  channel = NULL,  
  method = c("peaks", "spectral"),  
  band = c(0.1, 0.6),  
  min_period_sec = 1,  
  sr = NULL,  
  assay_name = NULL  
)
```

Arguments

- x, channel, band, min_period_sec, sr, assay_name

As in [respiratoryOnsets\(\)](#).
- method

"peaks" for breath intervals or "spectral" for the dominant respiratory frequency.

Value

A data frame with one row per selected channel and columns channel, method, rate_bpm, n-breaths, and sd_bpm.

Examples

```
sr <- 10
signal <- sin(2 * pi * 0.25 * (0:599) / sr)
breathingRate(signal, sr = sr)
```

dfaThresholdCrossings *Detect DFA-a1 threshold crossings*

Description

Finds where a rolling alpha1 trajectory crosses the two exercise-intensity markers of Gronwald & Rogers: AT1 (aerobic threshold, alpha1 = 0.75) and AT2 (anaerobic threshold, alpha1 = 0.5).

Usage

```
dfaThresholdCrossings(alpha1, at1 = 0.75, at2 = 0.5)
```

Arguments

alpha1	Numeric vector of rolling alpha1 values (e.g. the alpha1 column from ecgDFArolling).
at1, at2	Threshold values (defaults 0.75 and 0.5).

Value

A list with at1, at2 and integer vectors at1_crossings / at2_crossings giving the indices immediately before each crossing (either direction).

See Also

[ecgDFArolling](#)

Examples

```
a1 <- seq(1.2, 0.3, length.out = 20) # descending intensity ramp
dfaThresholdCrossings(a1)
```

Description

Aggregates the full HRV panel – time-domain, frequency-domain (AR spectrum with normalized units), geometric and nonlinear – together with a QTc morphology summary into a single per-channel data.frame, from an object produced by [ecgProcess](#) (after the neurokit2 `ecg_analyze()` pattern).

Usage

```
ecgAnalyze(processed)
```

Arguments

`processed` A `PhysioExperiment` returned by [ecgProcess](#).

Value

A data.frame with one row per channel and columns for the time-domain (`sdnn`, `rmssd`, `pnn50`, `mean_hr`), frequency-domain (`lf`, `hf`, `lf_hf_ratio`, `lf_nu`, `hf_nu`, `total_power`), geometric (`HRV_triangular_index`, `TINN`), nonlinear (`sd1`, `sd2`, `sample_entropy`, `alpha1`, `alpha2`) and morphology (`qt_ms`, `qtc_bazett`, `qrs_ms`) metrics. Metrics that cannot be computed (e.g. DFA on a short recording) are NA.

See Also

[ecgProcess](#), [ecgHRVtime](#), [ecgHRVfreq](#), [ecgHRVgeometric](#), [ecgHRVnonlinear](#), [ecgIntervals](#)

Examples

```
pe <- make_ecg(n_time = 40000, sr = 500)
ecgAnalyze(ecgProcess(pe))
```

Description

Removes low-frequency baseline drift from ECG data using either a high-pass moving-average subtraction or a running-median subtraction.

Usage

```
ecgBaselineCorrect(  
  x,  
  method = c("highpass", "median"),  
  cutoff = 0.5,  
  assay_name = NULL,  
  output_assay = "baseline_corrected"  
)
```

Arguments

x	A <code>PhysioExperiment</code> object with ECG data.
method	Correction method: "highpass" (default) subtracts a moving average; "median" subtracts a running median.
cutoff	Approximate cutoff frequency in Hz (default: 0.5). Controls the window size of the moving average or median filter.
assay_name	Name of the input assay. If NULL the default assay is used.
output_assay	Name of the assay in which to store the corrected signal (default: "baseline_corrected").

Value

A `PhysioExperiment` with the corrected signal stored in `output_assay`.

References

Clifford, G.D., Azuaje, F. & McSharry, P.E. (2006). *Advanced Methods and Tools for ECG Data Analysis*. Artech House.

See Also

[ecgSignalQuality](#) for signal quality assessment, [ecgDetectRpeaks](#) for R-peak detection, [ecgQualityCheck](#) for ectopic beat detection.

Examples

```
set.seed(1)  
pe <- make_ecg_noisy(n_time = 5000, sr = 500)  
pe <- ecgBaselineCorrect(pe, method = "highpass")  
SummarizedExperiment::assayNames(pe)
```

ecgBeatSQI

*Beat-level Signal Quality Index (bSQI)***Description**

Quantifies ECG quality per channel by the agreement between two independent QRS detectors. A clean signal makes both detectors agree on nearly every beat (bSQI near 1); noise makes them disagree (bSQI drops).

Usage

```
ecgBeatSQI(
  x,
  detector_a = "pan_tompkins",
  detector_b = "elgendi",
  match_window_ms = 150,
  threshold_factor = 0.6,
  refractory_ms = 200,
  beta = 0.08,
  assay_name = NULL
)
```

Arguments

x A `PhysioExperiment` object with ECG data.

detector_a, detector_b Detector methods to compare (see [ecgDetectRpeaks](#)); defaults "pan_tompkins" and "elgendi".

match_window_ms Two beats match when within this window (default 150).

threshold_factor, refractory_ms, beta Passed to the detectors.

assay_name Assay to use (default: the default assay).

Value

A `data.frame` with one row per channel: `n_a`, `n_b` (beat counts), `n_matched`, and `bSQI = matched / (n_a + n_b - matched)` (Jaccard agreement in $[0, 1]$; NA for a silent channel).

References

Li, Q., Mark, R.G. & Clifford, G.D. (2008). Robust heart rate estimation from multiple asynchronous noisy sources. *Physiological Measurement*, 29(1), 15-32.

See Also

[ecgDetectRpeaks](#), [ecgBeatSQIgate](#)

Examples

```
pe <- make_ecg(n_time = 5000, sr = 500, heart_rate = 72)
ecgBeatSQI(pe)
```

ecgBeatSQIgate	<i>Accept/reject ECG channels by bSQI</i>
----------------	---

Description

Thin wrapper over [ecgBeatSQI](#) adding a logical accept column for channels whose bSQI meets a threshold.

Usage

```
ecgBeatSQIgate(x, threshold = 0.8, ...)
```

Arguments

x	A <code>PhysioExperiment</code> object with ECG data.
threshold	Minimum bSQI to accept a channel (default 0.8).
...	Passed to ecgBeatSQI .

Value

The [ecgBeatSQI](#) data.frame with an added logical accept column.

See Also

[ecgBeatSQI](#)

Examples

```
pe <- make_ecg(n_time = 5000, sr = 500, heart_rate = 72)
ecgBeatSQIgate(pe, threshold = 0.8)
```

ecgDelineate	<i>Delineate ECG Waveform Morphology</i>
--------------	--

Description

For each detected R-peak, identifies QRS complex boundaries (onset and offset) and P-wave and T-wave peaks. QRS boundaries are detected using a gradient-based search from the R-peak, while P and T waves are found as local maxima in physiologically plausible time windows.

Usage

```
ecgDelineate(x, peaks, assay_name = NULL)
```

Arguments

x	A <code>PhysioExperiment</code> object with ECG data.
peaks	A <code>data.frame</code> of detected R-peaks as returned by ecgDetectRpeaks , with columns <code>channel</code> and <code>sample</code> .
assay_name	Name of the assay to use. If <code>NULL</code> , the default assay is used.

Value

A `data.frame` with one row per beat and the following columns:

channel Integer channel index (1-based).

beat Integer beat number within the channel (1-based).

r_peak Sample index of the R-peak.

qrs_onset Sample index of QRS complex onset.

qrs_offset Sample index of QRS complex offset (J-point).

qrs_duration_ms QRS complex duration in milliseconds.

p_peak Sample index of P-wave peak, or NA if not found in the search window (300–80 ms before R-peak).

t_peak Sample index of T-wave peak, or NA if not found in the search window (80–500 ms after R-peak).

t_end Sample index of T-wave end estimated by tangent-intercept method, or NA if T-wave not found.

Returns a zero-row `data.frame` with the same column structure if no beats are delineated.

References

Goldberger, A.L., et al. (2000). "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals." *Circulation*, 101(23), e215–e220. doi:[10.1161/01.CIR.101.23.e215](https://doi.org/10.1161/01.CIR.101.23.e215)

See Also

[ecgDetectRpeaks](#) for R-peak detection, [ecgIntervals](#) for computing clinical ECG intervals from delineation results, [ecgSignalQuality](#) for signal quality assessment.

Examples

```
set.seed(1)
pe <- make_ecg_pqrst(n_time = 5000, sr = 500, heart_rate = 72)$pe
peaks <- ecgDetectRpeaks(pe)
delin <- ecgDelineate(pe, peaks)
head(delin)
```

ecgDetectAF

Detect Atrial Fibrillation / Frequent Ectopy from RR Irregularity

Description

Classifies the cardiac rhythm of each channel as regular sinus, atrial fibrillation (AF), or frequent ectopy from the irregularity of the RR interval series. HRV metrics are physiologically undefined in AF, so this gate is used to invalidate them (see `rhythm_check` in [ecgHRVtime](#), [ecgHRVfreq](#) and [ecgHRVnonlinear](#)).

Usage

```
ecgDetectAF(
  rr,
  window_beats = NULL,
  ectopic_frac_max = 0.05,
  rmssd_ratio_af = 0.15,
  entropy_af = 0.7,
  n_bins = 16L,
  drr_range = c(-600, 600),
  threshold_ms = 300,
  min_beats = 20L
)
```

Arguments

<code>rr</code>	A data.frame with columns <code>channel</code> and <code>rr_ms</code> , as returned by ecgRRintervals .
<code>window_beats</code>	If non-NULL, an integer window length (in beats): instead of a single per-channel verdict, a continuous AF probability is returned for each non-overlapping window of <code>window_beats</code> beats (see <code>Value</code>).
<code>ectopic_frac_max</code>	Maximum ectopic fraction for a "sinus" verdict (default 0.05).
<code>rmssd_ratio_af</code>	Minimum RMSSD/mean-RR ratio consistent with AF (default 0.15).

entropy_af	Minimum normalized dRR Shannon entropy consistent with AF (default 0.7, range 0-1).
n_bins	Number of histogram bins for the dRR entropy (default 16).
drr_range	Numeric length-2 dRR window (ms) for the entropy histogram (default c(-600, 600)).
threshold_ms	Deviation threshold (ms) passed to ecgQualityCheck for the ectopic fraction (default 300).
min_beats	Minimum RR intervals required to classify; fewer yields NA (default 20).

Details

Three descriptors are combined: the ectopic-beat fraction from [ecgQualityCheck](#), the RMSSD/mean-RR normalized irregularity index, and the normalized Shannon entropy of the successive-difference (dRR) histogram. AF requires *both* large beat-to-beat variability (rmssd_ratio) and a broad, disordered dRR distribution (drr_entropy); a large but concentrated irregularity (e.g. an ectopic couplet) falls through to "frequent_ectopy".

Value

With window_beats = NULL (default), a data.frame with one row per channel and columns channel, rhythm ("sinus", "AF" or "frequent_ectopy"), ectopic_frac, rmssd_ratio, drr_entropy, af_prob (a continuous AF probability in 0-1) and n_beats. With window_beats set, a data.frame with one row per non-overlapping window and columns channel, window, beat_start, n_beats, af_prob and its components (rmssd_ratio, drr_entropy, sample_entropy, sd1_sd2).

References

Tateno, K. & Glass, L. (2001). Automatic detection of atrial fibrillation using the coefficient of variation and density histograms of RR and dRR intervals. *Medical & Biological Engineering & Computing*, 39(6), 664-671. Sarkar, S., Ritscher, D. & Mehra, R. (2008). A detector for a chronic implantable atrial tachyarrhythmia monitor. *IEEE Transactions on Biomedical Engineering*, 55(3), 1219-1224.

See Also

[ecgQualityCheck](#), [ecgHRVtime](#)

Examples

```
set.seed(1)
rr <- data.frame(channel = 1L, rr_ms = 800 + rnorm(100, sd = 20),
                 time_sec = cumsum(rep(0.8, 100)))
ecgDetectAF(rr)
```

ecgDetectPVC

Detect Premature Ventricular Contractions (PVCs)

Description

Flags ventricular ectopic beats from three classic features: an abnormal QRS morphology (low correlation with an average-beat template), prematurity of the preceding RR interval, and a full compensatory pause (the premature beat plus its following pause together span about two normal cycles). The compensatory pause is what distinguishes a PVC from a non-compensatory supraventricular ectopic beat or sinus arrhythmia.

Usage

```
ecgDetectPVC(
  x,
  peaks,
  rr,
  qrs_ms = 100,
  corr_threshold = 0.9,
  prematurity_max = 0.85,
  compensatory_min = 0.9,
  require_compensatory = FALSE,
  assay_name = NULL
)
```

Arguments

x	A <code>PhysioExperiment</code> object with ECG data.
peaks	R-peak table from ecgDetectRpeaks (columns <code>channel</code> , <code>sample</code>).
rr	RR-interval table from ecgRRintervals (columns <code>channel</code> , <code>rr_ms</code>).
qrs_ms	QRS window half-width in ms for template matching (default 100).
corr_threshold	Template-correlation below which a beat is morphologically abnormal (default 0.9).
prematurity_max	Prematurity ratio ($RR_{pre} / \text{local median } RR$) below which a beat is premature (default 0.85).
compensatory_min	Minimum $(RR_{pre} + RR_{post}) / (2 * \text{local median } RR)$ for a full compensatory pause (default 0.90).
require_compensatory	If TRUE, also require a compensatory pause for a PVC verdict (default FALSE).
assay_name	Input assay name (default: first assay).

Value

A data.frame with one row per beat: channel, beat, sample, rr_pre_ms, rr_post_ms, prematurity, compensatory, morph_corr and the logical is_pvc.

References

Petrenas, A., Marozas, V. & Sornmo, L. (2015). Low-complexity detection of atrial fibrillation in continuous long-term monitoring. *Computers in Biology and Medicine*, 65, 184-191.

See Also

[ecgDetectRpeaks](#), [ecgRRintervals](#), [ecgDetectAF](#)

Examples

```
pe <- make_ecg_pqrst(n_time = 5000, sr = 500)$pe
pk <- ecgDetectRpeaks(pe)
rr <- ecgRRintervals(pe, pk)
head(ecgDetectPVC(pe, pk, rr))
```

ecgDetectRpeaks

Detect R-Peaks in ECG Signal

Description

Identifies R-peaks in ECG data using one of four QRS detectors. All share a bandpass front end and automatic inverted-lead handling but differ in their feature transform and threshold rule: "pan_tompkins" (5-15 Hz, derivative-square-integrate, adaptive dual threshold), "hamilton" (8-16 Hz, rectified derivative, 80 ms integration, running mean of the last eight QRS/noise peaks), "elgendi" (8-20 Hz, squared, two moving averages with an event-block threshold), and "christov" (9-30 Hz, rectified derivative smoothed over 30 ms, adaptive steep-slope M-threshold with decay).

Usage

```
ecgDetectRpeaks(
  x,
  method = c("pan_tompkins", "hamilton", "elgendi", "christov"),
  threshold_factor = 0.6,
  refractory_ms = 200,
  assay_name = NULL,
  beta = 0.08
)
```

Arguments

<code>x</code>	A <code>PhysioExperiment</code> object with ECG data.
<code>method</code>	Detection method: one of "pan_tompkins" (default), "hamilton", "elgendi" or "christov".
<code>threshold_factor</code>	Retained for backward compatibility; the Pan-Tompkins path uses its built-in fractional threshold (default: 0.6).
<code>refractory_ms</code>	Refractory period in milliseconds. No two peaks can be closer than this (default: 200).
<code>assay_name</code>	Name of the assay to use. If NULL, the default assay is used.
<code>beta</code>	Offset factor for the Elgendi event threshold $MA_beat + beta * mean(squared\ signal)$ (default: 0.08); ignored by the other methods.

Value

A data.frame with one row per detected R-peak and the following columns:

channel Integer channel index (1-based).

sample Integer sample index of the R-peak within the assay matrix.

time_sec Time of the R-peak in seconds from signal onset.

amplitude Amplitude of the raw signal at the R-peak location (in original units, not inverted).

Returns a zero-row data.frame with the same column structure if no peaks are detected.

References

- Pan, J. & Tompkins, W.J. (1985). "A real-time QRS detection algorithm." *IEEE Transactions on Biomedical Engineering*, 32(3), 230–236. doi:[10.1109/TBME.1985.325532](https://doi.org/10.1109/TBME.1985.325532)
- Hamilton, P. (2002). "Open source ECG analysis." *Computers in Cardiology*, 29, 101–104.
- Elgendi, M. (2013). "Fast QRS detection with an optimized knowledge-based method." *PLoS ONE*, 8(9), e73557. doi:[10.1371/journal.pone.0073557](https://doi.org/10.1371/journal.pone.0073557)
- Christov, I. (2004). "Real time electrocardiogram QRS detection using combined adaptive threshold." *BioMedical Engineering OnLine*, 3, 28. doi:[10.1186/1475925X328](https://doi.org/10.1186/1475925X328)

See Also

[ecgRRintervals](#) for computing RR intervals from detected peaks, [ecgBeatsQI](#) for detector-agreement quality, [ecgDelineate](#) for full waveform morphology analysis, [ecgSignalQuality](#) for signal quality assessment.

Examples

```
set.seed(1)
pe <- make_ecg(n_time = 5000, sr = 500, heart_rate = 72)
peaks <- ecgDetectRpeaks(pe)
head(peaks)
```

Description

Performs detrended fluctuation analysis (DFA) on RR interval data to characterize fractal scaling properties. Computes α_1 (short-range correlations, 4–16 beats) and α_2 (long-range correlations, 16–64 beats).

Usage

```
ecgDFA(rr, short_range = c(4, 16), long_range = c(16, 64))
```

Arguments

rr	A data.frame with columns channel, rr_ms, and time_sec, as returned by ecgRRintervals .
short_range	Numeric vector of length 2 defining the scale range (in beats) for α_1 (default: c(4, 16)).
long_range	Numeric vector of length 2 defining the scale range (in beats) for α_2 (default: c(16, 64)).

Value

A data.frame with one row per channel and the following columns:

channel Integer channel index.

alpha1 Short-range scaling exponent. Values near 1.0 indicate fractal-like (healthy) correlations; values near 0.5 indicate uncorrelated (random) behavior; values near 1.5 suggest Brownian noise. NA if the series is too short.

alpha2 Long-range scaling exponent with the same interpretation as α_1 but over larger time scales. NA if the series is too short.

References

Peng, C.-K., et al. (1994). "Mosaic organization of DNA nucleotides." *Physical Review E*, 49(2), 1685–1689. doi:[10.1103/PhysRevE.49.1685](https://doi.org/10.1103/PhysRevE.49.1685)

See Also

[ecgHRVnonlinear](#) for the combined nonlinear analysis wrapper, [ecgSampleEntropy](#) for sample entropy, [ecgHRVpoincare](#) for Poincare plot descriptors.

Examples

```
set.seed(1)
n <- 300
time_sec <- cumsum(rep(0.85, n))
rr_ms <- 850 + 30 * sin(2 * pi * 0.1 * time_sec) + rnorm(n, sd = 15)
rr <- data.frame(channel = 1L, rr_ms = rr_ms, time_sec = time_sec)
ecgDFA(rr)
```

ecgDFArolling

Rolling (time-resolved) DFA alpha1

Description

Computes the short-range detrended-fluctuation scaling exponent (α_1) over a sliding window of beats, yielding a time-resolved α_1 trajectory. This is the basis of the DFA- α_1 exercise-intensity thresholds of Gronwald & Rogers (2020).

Usage

```
ecgDFArolling(rr, window_beats = 300, step_beats = 30, short_range = c(4, 16))
```

Arguments

<code>rr</code>	A data.frame with columns <code>channel</code> and <code>rr_ms</code> .
<code>window_beats</code>	Window length in beats (default 300).
<code>step_beats</code>	Step between successive windows in beats (default 30).
<code>short_range</code>	Scale range (in beats) for α_1 (default <code>c(4, 16)</code>).

Value

A data.frame with one row per window and columns `channel`, `beat_start`, `beat_end`, `beat_center` and α_1 .

References

Peng, C.K. et al. (1995). Quantification of scaling exponents. *Chaos*, 5(1), 82-87. Gronwald, T. & Rogers, B. (2020). Fractal correlation properties of heart rate variability as a biomarker. *Frontiers in Physiology*, 11, 550572.

See Also

[ecgDFA](#), [dfaThresholdCrossings](#)

Examples

```
set.seed(1)
rr <- data.frame(channel = 1L, rr_ms = 800 + cumsum(rnorm(1000, sd = 2)))
traj <- ecgDFArolling(rr, window_beats = 200, step_beats = 50)
```

ecgEctopicBurden	<i>Ectopic Beat Burden</i>
------------------	----------------------------

Description

Convenience summary of the ectopic-beat burden in an RR-interval data.frame that has been passed through [ecgQualityCheck](#).

Usage

```
ecgEctopicBurden(rr)
```

Arguments

rr	A data.frame with an is_ectopic logical column.
----	---

Value

A list with n_ectopic (count of ectopic beats), n_total (total beats) and pct_ectopic (percent ectopic).

See Also

[ecgQualityCheck](#), [ecgRRcorrect](#)

Examples

```
rr <- data.frame(channel = 1L, rr_ms = c(800, 810, 400, 805, 815),  
                 is_ectopic = c(FALSE, FALSE, TRUE, FALSE, FALSE))  
ecgEctopicBurden(rr)
```

ecgEDR	<i>ECG-Derived Respiration (EDR)</i>
--------	--------------------------------------

Description

Estimates a respiration surrogate from beat-to-beat modulation of the QRS complex (Moody et al. 1985). Respiration shifts the cardiac electrical axis, modulating R-wave amplitude, QRS area or upslope; the per-beat feature series is resampled to a uniform grid to form the EDR signal, and the respiratory rate is read from its spectral peak.

Usage

```
ecgEDR(
  x,
  peaks,
  method = c("amplitude", "area", "slope"),
  fs = 4,
  resp_band = c(0.1, 0.5),
  qrs_ms = 50,
  assay_name = NULL
)
```

Arguments

<code>x</code>	A <code>PhysioExperiment</code> object with ECG data.
<code>peaks</code>	R-peak table from ecgDetectRpeaks (columns <code>channel</code> , <code>sample</code> , <code>time_sec</code> ; <code>amplitude</code> is used by the "amplitude" method when present).
<code>method</code>	Beat feature: "amplitude" (R-wave amplitude, default), "area" (QRS area) or "slope" (maximum QRS upslope).
<code>fs</code>	Resampling rate (Hz) of the EDR signal (default 4).
<code>resp_band</code>	Respiratory frequency band (Hz) for the rate estimate (default <code>c(0.1, 0.5)</code> , i.e. 6-30 breaths/min).
<code>qrs_ms</code>	QRS half-window (ms) for the "area"/"slope" features (default 50).
<code>assay_name</code>	Input assay name (default: first assay).

Value

A list with:

method The feature used.

beats A `data.frame` (`channel`, `time_sec`, `feature`) of the per-beat feature.

edr A `data.frame` (`channel`, `time_sec`, `edr`) of the uniform-resampled respiration surrogate.

resp_rate A `data.frame` (`channel`, `resp_rate_hz`, `resp_rate_bpm`) of the estimated respiratory rate.

References

Moody, G.B. et al. (1985). Derivation of respiratory signals from multi-lead ECGs. *Computers in Cardiology*, 12, 113-116.

See Also

[ecgDetectRpeaks](#), [ecgRSA](#)

Examples

```
pe <- make_ecg(n_time = 10000, sr = 250)
pk <- ecgDetectRpeaks(pe)
edr <- ecgEDR(pe, pk, method = "area")
edr$resp_rate
```

ecgHRR

Heart-Rate Recovery (HRR)

Description

Computes post-exercise heart-rate recovery: the fall in heart rate from the exercise peak to fixed offsets into recovery (Cole et al. 1999; Imai et al. 1994), plus the recovery slope. Accepts either a heart-rate (bpm) or an RR (ms) series.

Usage

```
ecgHRR(
  rr_or_hr,
  time_sec = NULL,
  peak_time = NULL,
  recovery_offsets_sec = c(60, 120),
  input = c("auto", "hr", "rr"),
  abnormal_bpm = 12
)
```

Arguments

<code>rr_or_hr</code>	A numeric vector of heart rate (bpm) or RR intervals (ms), or a data.frame with a <code>time_sec</code> column and an <code>hr</code> or <code>rr_ms</code> column.
<code>time_sec</code>	Time of each sample in seconds (required for a vector input; taken from the <code>time_sec</code> column of a data.frame otherwise).
<code>peak_time</code>	Time (seconds) of the exercise peak / start of recovery. If NULL (default), auto-detected as the time of maximum heart rate.
<code>recovery_offsets_sec</code>	Offsets (seconds) after the peak at which to measure recovery (default <code>c(60, 120)</code> , i.e. HRR1 and HRR2).
<code>input</code>	Interpretation of a vector input: "auto" (default; decided by magnitude – values above 250 are treated as RR in ms), "hr" or "rr".
<code>abnormal_bpm</code>	HRR at 1 minute below which recovery is flagged abnormal (default 12 bpm, per Cole et al. 1999).

Value

A list with:

peak_time, peak_hr Recovery start time (s) and peak HR (bpm).

hrr A data.frame with columns `offset_sec`, `hr_at_offset` and `hrr_bpm` (peak HR minus HR at the offset).

hrr1, hrr2 Convenience scalars for the first two offsets.

slope_bpm_per_min Slope of a linear fit of HR vs time over the recovery window (negative during recovery).

hrr_1min HRR at exactly 60 s (used for the abnormal flag).

abnormal TRUE if `hrr_1min < abnormal_bpm`.

References

Cole, C.R. et al. (1999). Heart-rate recovery immediately after exercise as a predictor of mortality. *New England Journal of Medicine*, 341(18), 1351-1357. Imai, K. et al. (1994). Vagally mediated heart rate recovery after exercise is accelerated in athletes but blunted in patients with chronic heart failure. *Journal of the American College of Cardiology*, 24(6), 1529-1535.

See Also

[ecgRRintervals](#), [ecgHRVtime](#)

Examples

```
# HR decays linearly from 170 to 120 bpm over 120 s of recovery
t <- seq(0, 120, by = 1)
hr <- 170 - (170 - 120) * t / 120
res <- ecgHRR(hr, time_sec = t, peak_time = 0)
res$hrr
res$abnormal
```

ecgHRT

Heart-Rate Turbulence (HRT)

Description

Computes turbulence onset (TO) and turbulence slope (TS), the biphasic sinus-rhythm response to a ventricular premature beat (Schmidt et al. 1999). RR intervals are indexed by beat so that `rr[i]` is the interval between beat `i` and beat `i+1`; for a PVC at beat `p` the coupling interval is `rr[p-1]` and the compensatory pause `rr[p]`, which are excluded. The local tachograms of all usable PVCs are averaged before TO/TS.

Usage

```
ecgHRT(rr, pvc_index, n_post = 15L)
```

Arguments

<code>rr</code>	RR intervals (ms): a numeric vector or a data.frame with an <code>rr_ms</code> column.
<code>pvc_index</code>	Integer beat indices of the ventricular premature beats (e.g. the <code>beat</code> column of ecgDetectPVC where <code>is_pvc</code>).
<code>n_post</code>	Number of post-PVC sinus intervals used for TS (default 15).

Value

A list with `to` (turbulence onset, percent; negative is normal), `ts` (turbulence slope, ms per RR interval; positive is normal), `n_pvc` (PVCs averaged) and `tachogram` (the averaged post-PVC sinus RR series).

References

Schmidt, G. et al. (1999). Heart-rate turbulence after ventricular premature beats as a predictor of mortality after acute myocardial infarction. *Lancet*, 353(9162), 1390-1396.

See Also

[ecgDetectPVC](#), [ecgPRSA](#)

Examples

```
# canonical post-PVC pattern: early acceleration then deceleration
rr <- c(rep(800, 5), 500, 1100, 780, 785, 790 + 5 * (0:12))
ecgHRT(rr, pvc_index = 7)
```

ecgHRVasymmetry

Heart-Rate Asymmetry (Poincare Plot Asymmetry) Descriptors

Description

Computes heart-rate asymmetry (HRA) descriptors from RR interval data. The Poincare plot is asymmetric about the line of identity because heart-rate *decelerations* (prolongations of RR, points above the line, $RR_{n+1} > RR_n$) and *accelerations* (shortenings, points below the line) contribute unequally to variability – a signature of vagal activity and of the temporal irreversibility of heart-rate dynamics. Decelerations typically dominate short-term variability while accelerations dominate long-term variability (Piskorski & Guzik, 2011).

Usage

```
ecgHRVasymmetry(rr)
```

Arguments

<code>rr</code>	A data.frame with columns <code>channel</code> , <code>rr_ms</code> , and <code>time_sec</code> , as returned by ecgRRintervals .
-----------------	---

Value

A data.frame with one row per channel and the columns:

channel Channel identifier.

gi Guzik's Index ($\backslash$ points from the line of identity divided by that of all points).

si Slope Index ($\backslash$ points divided by that of all points).

ai Area Index ($\backslash$ points divided by that of all points).

pi Porta's Index ($\backslash$ the number of points off the line of identity).

c1d, c1a Contributions of decelerations / accelerations to short-term HRV ($c1d + c1a = 1$).

sd1d, sd1a Short-term variance components of decelerations / accelerations (ms).

c2d, c2a Contributions of decelerations / accelerations to long-term HRV ($c2d + c2a = 1$).

sd2d, sd2a Long-term variance components (ms).

cd, ca Total contributions of decelerations / accelerations to HRV.

sdnnd, sdnna Total variance components (ms).

A value indicating asymmetry is $c1d > c1a$ (decelerations dominate short-term variability). Columns are NA for channels with fewer than three RR intervals.

References

Guzik, P., et al. (2006). "Heart rate asymmetry by Poincare plots of RR intervals." *Biomedizinische Technik*, 51(4), 272–275.

Piskorski, J., & Guzik, P. (2011). "Asymmetric properties of long-term and total heart rate variability." *Medical & Biological Engineering & Computing*, 49(11), 1289–1297. doi:10.1007/s11517-0110834z

Porta, A., et al. (2008). "Temporal asymmetries of short-term heart period variability are linked to autonomic regulation." *American Journal of Physiology*, 295(2), R550–R557.

See Also

[ecgHRVpoincare](#) for the symmetric Poincare descriptors (SD1/SD2), [ecgDFA](#) for detrended fluctuation analysis.

Examples

```
set.seed(1)
n <- 300
time_sec <- cumsum(rep(0.85, n))
rr_ms <- 850 + 30 * sin(2 * pi * 0.1 * time_sec) + rnorm(n, sd = 15)
rr <- data.frame(channel = 1L, rr_ms = rr_ms, time_sec = time_sec)
ecgHRVasymmetry(rr)
```

ecgHRVautonomic

*Cardiac Autonomic Indices (Toichi CSI / CVI) from the Poincare Plot***Description**

Computes the cardiac sympathetic and vagal indices (Toichi et al., 1997) from RR interval data. The indices are read off the Poincare plot's $4 \times \text{SD}$ bounding box: the long axis $L = 4 \text{SD}_2$ and the short axis $T = 4 \text{SD}_1$.

`csi` Cardiac Sympathetic Index, L/T ($= \text{SD}_2/\text{SD}_1$); rises with sympathetic predominance.

`cvi` Cardiac Vagal Index, $\log_{10}(L \times T)$; rises with parasympathetic (vagal) activity.

`csi_modified` Modified CSI, L^2/T , a more sensitive sympathetic marker (Jeppesen et al., 2014) used in seizure detection.

Here SD_1 is the universal beat-to-beat term ($\text{SDSD}/\sqrt{2}$) and SD_2 is the geometric (paired-projection) long-term term, matching the convention in which the Toichi indices are defined (as in NeuroKit2's `hrv_nonlinear`); this SD_2 differs by a small $O(1/n)$ amount from the analytical closed-form SD_2 reported by [ecgHRVpoincare](#).

Usage

```
ecgHRVautonomic(rr)
```

Arguments

`rr` A `data.frame` with columns `channel`, `rr_ms`, and `time_sec`, as returned by [ecgRRintervals](#).

Value

A `data.frame` with one row per channel and the columns `channel`, `csi`, `cvi`, and `csi_modified`. Columns are NA for channels with fewer than three RR intervals.

References

Toichi, M., Sugiura, T., Murai, T., & Sengoku, A. (1997). "A new method of assessing cardiac autonomic function and its comparison with spectral analysis and coefficient of variation of R-R interval." *Journal of the Autonomic Nervous System*, 62(1-2), 79–84.

Jeppesen, J., et al. (2014). "Using Lorenz plot and Cardiac Sympathetic Index of heart rate variability for detecting seizures for patients with epilepsy." *IEEE EMBC*, 4563–4566.

See Also

[ecgHRVpoincare](#) for SD_1/SD_2 , [ecgHRVasymmetry](#) for heart-rate asymmetry.

Examples

```
set.seed(1)
n <- 300
time_sec <- cumsum(rep(0.85, n))
rr_ms <- 850 + 30 * sin(2 * pi * 0.1 * time_sec) + rnorm(n, sd = 15)
rr <- data.frame(channel = 1L, rr_ms = rr_ms, time_sec = time_sec)
ecgHRVautonomic(rr)
```

ecgHRVfreq

HRV Frequency-Domain Analysis

Description

Computes heart rate variability frequency-domain metrics from RR interval data using Welch's method or Lomb-Scargle periodogram.

Usage

```
ecgHRVfreq(
  rr,
  method = c("welch", "lomb", "ar"),
  vlf_band = c(0.003, 0.04),
  lf_band = c(0.04, 0.15),
  hf_band = c(0.15, 0.4),
  detrend = FALSE,
  detrend_lambda = 500,
  ar_order = NULL,
  rhythm_check = TRUE
)
```

Arguments

rr	A data.frame with columns channel, rr_ms, and time_sec, as returned by ecgRRintervals .
method	Spectral estimation method: "welch" (default) for uniformly resampled FFT-based PSD, "lomb" for Lomb-Scargle periodogram on unevenly sampled data, or "ar" for a Burg autoregressive PSD.
vlf_band	Numeric vector of length 2 defining VLF band in Hz (default: c(0.003, 0.04)).
lf_band	Numeric vector of length 2 defining LF band in Hz (default: c(0.04, 0.15)).
hf_band	Numeric vector of length 2 defining HF band in Hz (default: c(0.15, 0.4)).
detrend	Logical; if TRUE, apply smoothness-priors detrending (smoothnessPriorsDetrend) to the resampled tachogram before spectral estimation (Welch method). Default FALSE (mean removal).
detrend_lambda	Smoothing parameter passed to smoothnessPriorsDetrend when detrend = TRUE (default 500).

ar_order	Integer AR model order for method = "ar"; NULL (default) selects the order automatically by AIC.
rhythm_check	Logical; if TRUE (default), gate the result on ecgDetectAF : metrics for channels classified as AF or frequent ectopy are set to NA and rhythm/hrv_valid columns are appended (HRV is undefined in atrial fibrillation).

Value

A data.frame with one row per channel and the following columns:

channel Integer channel index.

vlf Absolute power in the very-low-frequency band (ms^2).

lf Absolute power in the low-frequency band (ms^2), associated with sympathetic and parasympathetic modulation.

hf Absolute power in the high-frequency band (ms^2), associated with parasympathetic (vagal) modulation.

lf_hf_ratio Ratio of LF to HF power, or NA if HF power is zero.

total_power Sum of VLF, LF, and HF power (ms^2).

lf_nu, hf_nu LF and HF power in normalized units, $100 * \text{LF} / (\text{LF} + \text{HF})$ and $100 * \text{HF} / (\text{LF} + \text{HF})$; these sum to 100.

lf_peak, hf_peak Frequency (Hz) of the spectral peak within the LF and HF bands, or NA if the band is empty.

References

Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996). "Heart rate variability: Standards of measurement, physiological interpretation and clinical use." *Circulation*, 93(5), 1043–1065.

See Also

[ecgRRintervals](#) for computing RR intervals, [ecgHRVtime](#) for time-domain HRV metrics, [ecgHRVnonlinear](#) for nonlinear HRV analysis, [ecgRRcorrect](#) for ectopic beat correction before analysis.

Examples

```
n <- 300
time_sec <- cumsum(rep(0.85, n))
rr_ms <- 850 + 30 * sin(2 * pi * 0.1 * time_sec)
rr <- data.frame(channel = rep(1L, n), rr_ms = rr_ms, time_sec = time_sec)
result <- ecgHRVfreq(rr, method = "welch")
```

ecgHRVgeometric

Geometric HRV indices (triangular index and TINN)

Description

Computes the geometric heart-rate-variability measures defined by the Task Force (1996) from a series of NN (normal-to-normal) intervals: the HRV triangular index and the triangular interpolation of the NN interval histogram (TINN).

Usage

```
ecgHRVgeometric(rr, bin_ms = 7.8125)
```

Arguments

<code>rr</code>	Numeric vector of NN/RR intervals in milliseconds, or a data frame with an <code>rr_ms</code> column (as produced by the RR-interval readers); a data frame is reduced to its <code>rr_ms</code> column.
<code>bin_ms</code>	Histogram bin width in milliseconds. Defaults to 7.8125 ms (1/128 s), the Task Force standard.

Value

A list with:

HRV_triangular_index Total number of NN intervals divided by the height (modal count) of the NN histogram.

TINN Baseline width, in milliseconds, of the triangle that best approximates the NN histogram in a least-squares sense.

References

Task Force of the ESC and NASPE (1996). Heart rate variability: standards of measurement. *Circulation*, 93(5), 1043-1065.

See Also

[ecgHRVsegmented](#), [ecgHRVtime](#)

Examples

```
set.seed(1)
rr <- 800 + rnorm(500, sd = 40) # NN intervals in ms
ecgHRVgeometric(rr)
```

ecgHRVnonlinear	<i>Nonlinear HRV Analysis (Convenience Wrapper)</i>
-----------------	---

Description

Computes all nonlinear HRV metrics by calling [ecgHRVpoincare](#), [ecgSampleEntropy](#), and [ecgDFA](#), and merging the results into a single data.frame.

Usage

```
ecgHRVnonlinear(
  rr,
  m = 2L,
  r_factor = 0.2,
  short_range = c(4, 16),
  long_range = c(16, 64),
  rhythm_check = TRUE
)
```

Arguments

rr	A data.frame with columns channel, rr_ms, and time_sec, as returned by ecgRRintervals .
m	Embedding dimension for sample entropy (default: 2).
r_factor	Tolerance factor for sample entropy (default: 0.2).
short_range	Scale range for DFA alpha1 (default: c(4, 16)).
long_range	Scale range for DFA alpha2 (default: c(16, 64)).
rhythm_check	Logical; if TRUE (default), gate the result on ecgDetectAF : metrics for channels classified as AF or frequent ectopy are set to NA and rhythm/hrv_valid columns are appended (HRV is undefined in atrial fibrillation).

Value

A data.frame with columns: channel, sd1, sd2, sd1_sd2_ratio, sample_entropy, m, r, alpha1, alpha2.

References

Shaffer, F. & Ginsberg, J.P. (2017). "An overview of heart rate variability metrics and norms." *Frontiers in Public Health*, 5, 258. doi:10.3389/fpubh.2017.00258

Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996). "Heart rate variability: Standards of measurement, physiological interpretation and clinical use." *Circulation*, 93(5), 1043–1065.

See Also

[ecgHRVpoincare](#) for Poincare plot descriptors, [ecgSampleEntropy](#) for sample entropy, [ecgDFA](#) for detrended fluctuation analysis, [ecgHRVtime](#) for time-domain HRV metrics, [ecgHRVfreq](#) for frequency-domain HRV analysis.

Examples

```
set.seed(1)
n <- 300
time_sec <- cumsum(rep(0.85, n))
rr_ms <- 850 + 30 * sin(2 * pi * 0.1 * time_sec) + rnorm(n, sd = 15)
rr <- data.frame(channel = 1L, rr_ms = rr_ms, time_sec = time_sec)
ecgHRVnonlinear(rr)
```

ecgHRVpoincare

Poincare Plot Descriptors for HRV Analysis

Description

Computes Poincare plot descriptors (SD1, SD2, SD1/SD2 ratio) from RR interval data. SD1 reflects short-term variability (perpendicular to the identity line), while SD2 reflects long-term variability (along the identity line).

Usage

```
ecgHRVpoincare(rr)
```

Arguments

rr A data.frame with columns channel, rr_ms, and time_sec, as returned by [ecgRRintervals](#).

Value

A data.frame with one row per channel and the following columns:

channel Integer channel index.

sd1 Standard deviation perpendicular to the identity line (ms), reflecting beat-to-beat (short-term) variability.

sd2 Standard deviation along the identity line (ms), reflecting long-term variability.

sd1_sd2_ratio Ratio of SD1 to SD2, or NA if SD2 is zero.

References

Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996). "Heart rate variability: Standards of measurement, physiological interpretation and clinical use." *Circulation*, 93(5), 1043–1065.

See Also

[ecgHRVnonlinear](#) for the combined nonlinear analysis wrapper, [ecgSampleEntropy](#) for sample entropy, [ecgDFA](#) for detrended fluctuation analysis.

Examples

```
set.seed(1)
n <- 300
time_sec <- cumsum(rep(0.85, n))
rr_ms <- 850 + 30 * sin(2 * pi * 0.1 * time_sec) + rnorm(n, sd = 15)
rr <- data.frame(channel = 1L, rr_ms = rr_ms, time_sec = time_sec)
ecgHRVpoincare(rr)
```

ecgHRVsegmented

*Segmented (long-term) HRV indices: SDANN and SDNN index***Description**

Splits an NN-interval series into consecutive fixed-duration segments (5 minutes by default) and returns the two Task Force (1996) long-term variability measures.

Usage

```
ecgHRVsegmented(rr, segment_sec = 300)
```

Arguments

rr Numeric vector of NN/RR intervals in milliseconds.

segment_sec Segment length in seconds (default 300 = 5 minutes).

Value

A list with:

SDANN Standard deviation of the per-segment mean NN intervals (ms) - long-term variability.

SDNN_index Mean of the per-segment SDNN values (ms) - average short-term variability.

n_segments Number of segments used.

References

Task Force of the ESC and NASPE (1996). *Circulation*, 93(5), 1043-1065.

See Also

[ecgHRVgeometric](#), [ecgHRVtime](#)

Examples

```
set.seed(1)
rr <- 800 + rnorm(3000, sd = 40)
ecgHRVsegmented(rr, segment_sec = 300)
```

ecgHRVtime	<i>HRV Time-Domain Metrics</i>
------------	--------------------------------

Description

Compute standard heart rate variability (HRV) time-domain metrics from RR interval data. Calculates SDNN, RMSSD, pNN50, mean RR interval, and mean heart rate for each channel.

Usage

```
ecgHRVtime(rr, rhythm_check = TRUE)
```

Arguments

rr	A data.frame with columns channel, rr_ms, and time_sec, as returned by ecgRRintervals .
rhythm_check	Logical; if TRUE (default), gate the result on ecgDetectAF : metrics for channels classified as AF or frequent ectopy are set to NA and rhythm/hrv_valid columns are appended (HRV is undefined in atrial fibrillation).

Value

A data.frame with one row per channel and the following columns:

- channel** Integer channel index.
- mean_rr** Mean RR interval in milliseconds.
- sdnn** Standard deviation of all RR intervals (ms), reflecting overall HRV.
- rmssd** Root mean square of successive RR interval differences (ms), reflecting short-term vagal modulation.
- pnn50** Percentage of successive RR intervals differing by more than 50 ms.
- mean_hr** Mean heart rate in beats per minute (60000 / mean_rr).

References

Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996). "Heart rate variability: Standards of measurement, physiological interpretation and clinical use." *Circulation*, 93(5), 1043–1065.

See Also

[ecgRRintervals](#) for computing RR intervals, [ecgHRVfreq](#) for frequency-domain HRV analysis, [ecgHRVnonlinear](#) for nonlinear HRV metrics, [ecgQualityCheck](#) for ectopic beat detection before analysis.

Examples

```
set.seed(1)
n <- 300
time_sec <- cumsum(rep(0.85, n))
rr_ms <- 850 + 30 * sin(2 * pi * 0.1 * time_sec) + rnorm(n, sd = 15)
rr <- data.frame(channel = 1L, rr_ms = rr_ms, time_sec = time_sec)
ecgHRVtime(rr)
```

ecgHRVtimevarying	<i>Time-Varying HRV (Sliding-Window Time and Frequency Metrics)</i>
-------------------	---

Description

Computes heart-rate-variability trajectories by sliding a time window over the RR series and evaluating time-domain (SDNN, RMSSD) and frequency-domain (LF, HF, LF/HF) metrics in each window, reusing [ecgHRVtime](#) and [ecgHRVfreq](#) (Task Force 1996; Mainardi 2009).

Usage

```
ecgHRVtimevarying(
  rr,
  window_sec = 300,
  step_sec = 30,
  freq_method = c("ar", "welch", "lomb"),
  detrend = FALSE,
  detrend_lambda = 500,
  min_beats = 20L,
  rhythm_check = FALSE
)
```

Arguments

<code>rr</code>	A data.frame with columns <code>channel</code> , <code>rr_ms</code> and <code>time_sec</code> (as returned by ecgRRintervals), assumed ordered in time per channel.
<code>window_sec</code>	Analysis window length in seconds (default 300).
<code>step_sec</code>	Step between successive windows in seconds (default 30).
<code>freq_method</code>	Spectral method for ecgHRVfreq : "ar" (default), "welch" or "lomb".
<code>detrend, detrend_lambda</code>	Passed to ecgHRVfreq for optional smoothness-priors detrending of the resampled tachogram.

min_beats	Minimum RR intervals in a window to compute metrics; windows with fewer beats yield NA metrics (default 20).
rhythm_check	Passed to the per-window HRV functions (default FALSE, so the trajectory is continuous and not gated by the AF detector).

Value

A data.frame with one row per window per channel and columns channel, window, time_start, time_center, n_beats, sdn, rmssd, lf, hf and lf_hf.

References

Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996). Heart rate variability: standards of measurement, physiological interpretation, and clinical use. *Circulation*, 93(5), 1043-1065. Mainardi, L.T. (2009). On the quantification of heart rate variability spectral parameters using time-frequency and time-varying methods. *Philosophical Transactions of the Royal Society A*, 367(1887), 255-275.

See Also

[ecgHRVtime](#), [ecgHRVfreq](#), [ecgRRintervals](#)

Examples

```
set.seed(1)
n <- 600
rr <- data.frame(channel = 1L,
                 rr_ms = 800 + 25 * sin(2 * pi * 0.1 * cumsum(rep(0.8, n))) +
                   rnorm(n, sd = 10))
rr$time_sec <- cumsum(rr$rr_ms) / 1000
traj <- ecgHRVtimevarying(rr, window_sec = 120, step_sec = 30)
head(traj)
```

ecgIntervals

Compute ECG Intervals from Delineation

Description

Calculates standard clinical ECG intervals from the waveform delineation produced by [ecgDelineate](#): PR interval, QT interval, QTc (Bazett correction), QRS duration, and RR interval.

Usage

```
ecgIntervals(delineation, sr)
```

Arguments

delineation A data.frame as returned by [ecgDelineate](#), with columns channel, beat, r_peak, qrs_onset, qrs_offset, p_peak, t_peak, t_end.

sr Sampling rate in Hz.

Value

A data.frame with one row per beat and the following columns:

channel Integer channel index (1-based).

beat Integer beat number within the channel.

pr_ms PR interval in milliseconds (P-wave peak to QRS onset), or NA if the P wave was not detected.

qt_ms QT interval in milliseconds (QRS onset to T-wave end), or NA if the T wave was not detected.

qtc_ms Corrected QT interval using Bazett's formula ($QT / \sqrt{RR_sec}$), or NA if QT or RR is unavailable. Kept as a backward-compatible alias of `qtc_bazett`.

qtc_bazett, qtc_fridericia, qtc_framingham, qtc_hodges QT corrected for heart rate by, respectively, Bazett ($QT / \sqrt{RR}$), Fridericia ($QT / RR^{(1/3)}$), Framingham ($QT + 154 * (1 - RR)$) and Hodges ($QT + 1.75 * (HR - 60)$); QT in ms, RR in seconds. All four equal `qt_ms` at $RR = 1$ s.

qrs_ms QRS complex duration in milliseconds.

rr_ms RR interval in milliseconds to the next beat, or NA for the last beat in each channel.

Returns a zero-row data.frame with the same column structure if no beats are present.

References

Goldberger, A.L., et al. (2000). "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals." *Circulation*, 101(23), e215–e220. doi:10.1161/01.CIR.101.23.e215

See Also

[ecgDelineate](#) for waveform delineation, [ecgDetectRpeaks](#) for R-peak detection, [ecgRRintervals](#) for RR interval computation.

Examples

```
set.seed(1)
pe <- make_ecg_pqrst(n_time = 5000, sr = 500, heart_rate = 72)$pe
peaks <- ecgDetectRpeaks(pe)
delin <- ecgDelineate(pe, peaks)
intervals <- ecgIntervals(delin, samplingRate(pe))
head(intervals)
```

Description

Runs the standard ECG preprocessing chain end to end – baseline correction, R-peak detection, RR extraction, ectopic-beat correction and (optionally) a bSQI signal-quality gate – and stores the peaks, RR series and provenance in the object's `metadata()` (after the `neurokit2 ecg_process()` pattern).

Usage

```
ecgProcess(
  x,
  detector = "pan_tompkins",
  correct = "cubic_spline",
  sqi_gate = TRUE,
  baseline = TRUE,
  assay_name = NULL
)
```

Arguments

<code>x</code>	A <code>PhysioExperiment</code> object with ECG data.
<code>detector</code>	R-peak detector passed to ecgDetectRpeaks (default "pan_tompkins").
<code>correct</code>	Ectopic-beat correction passed to ecgRRcorrect : "cubic_spline" (default), "interpolate" or "remove".
<code>sqi_gate</code>	If TRUE (default), also compute the bSQI quality gate (ecgBeatSQIgate).
<code>baseline</code>	If TRUE (default), baseline-correct with ecgBaselineCorrect and detect on the corrected assay.
<code>assay_name</code>	Input assay name (default: first assay).

Value

The (baseline-corrected) `PhysioExperiment` with an `ecg` entry in `metadata()` holding peaks, `rr` (with ectopic flags), `rr_corrected`, `bsqi` and the detect-assay name, plus a provenance record appended via [withProvenance](#).

See Also

[ecgAnalyze](#), [ecgDetectRpeaks](#), [ecgRRcorrect](#), [ecgBeatSQIgate](#)

Examples

```
pe <- make_ecg(n_time = 20000, sr = 500)
proc <- ecgProcess(pe)
names(S4Vectors::metadata(proc)$ecg)
```

ecgPRSA

*Phase-Rectified Signal Averaging (PRSA): DC and AC***Description**

Computes the deceleration capacity (DC) and acceleration capacity (AC) of the RR series by phase-rectified signal averaging (Bauer et al. 2006). Anchors are points where the RR level increases (DC) or decreases (AC); segments centered on the anchors are averaged into a PRSA curve, and the capacity is a Haar-wavelet coefficient of that curve at scale s .

Usage

```
ecgPRSA(rr, T = 1L, s = 2L, L = 15L)
```

Arguments

<code>rr</code>	RR intervals (ms): a numeric vector or a data.frame with an <code>rr_ms</code> column.
<code>T</code>	Averaging window (beats) for anchor selection (default 1: an anchor is a single beat larger/smaller than the previous).
<code>s</code>	Wavelet scale for the DC/AC coefficient (default 2).
<code>L</code>	Half-length of the averaged PRSA segments (default 15; must be $\geq s$).

Value

A list with `dc` (deceleration capacity, ms; positive with dominant decelerations), `ac` (acceleration capacity, ms; negative), the PRSA curves `prsa_dc/prsa_ac`, the anchor counts `n_dc/n_ac`, and the parameters `T`, `s`, `L`.

References

Bauer, A. et al. (2006). Deceleration capacity of heart rate as a predictor of mortality after myocardial infarction: cohort study. *Lancet*, 367(9523), 1674-1681.

See Also

[ecgHRT](#), [ecgRRintervals](#)

Examples

```
set.seed(1)
rr <- 800 + cumsum(rnorm(600, sd = 3))      # RR random walk
prsa <- ecgPRSA(rr)
c(DC = prsa$dc, AC = prsa$ac)
```

ecgQualityCheck	<i>Detect Ectopic Beats in RR Interval Data</i>
-----------------	---

Description

Identifies ectopic (abnormal) beats by comparing each RR interval to the local median computed over a sliding window of 5 beats. Beats with deviation exceeding the threshold are marked as ectopic.

Usage

```
ecgQualityCheck(rr, threshold_ms = 300)
```

Arguments

rr	A data.frame with columns channel, rr_ms, and time_sec, as returned by ecgRRintervals .
threshold_ms	Maximum allowed deviation from local median in milliseconds (default: 300).

Value

The input data.frame with an additional logical column is_ectopic.

References

Clifford, G.D., Azuaje, F. & McSharry, P.E. (2006). *Advanced Methods and Tools for ECG Data Analysis*. Artech House.

Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996). "Heart rate variability: Standards of measurement, physiological interpretation and clinical use." *Circulation*, 93(5), 1043–1065.

See Also

[ecgRRcorrect](#) for correcting detected ectopic beats, [ecgRRintervals](#) for computing RR intervals, [ecgHRVtime](#) for time-domain HRV analysis, [ecgSignalQuality](#) for signal quality assessment.

Examples

```
set.seed(1)
n <- 200
rr_ms <- 800 + rnorm(n, sd = 20)
rr_ms[100] <- 430 # a premature (ectopic) beat
rr <- data.frame(channel = 1L, rr_ms = rr_ms,
                 time_sec = cumsum(rr_ms) / 1000)
qc <- ecgQualityCheck(rr)
sum(qc$is_ectopic)
```

ecgRRcorrect

*Correct Ectopic Beats in RR Interval Data***Description**

Replaces or removes ectopic beats identified by [ecgQualityCheck](#). The "interpolate" method uses linear interpolation from surrounding non-ectopic intervals, "cubic_spline" uses natural cubic-spline interpolation (C1-smooth, following Kubios-style automatic correction), and "remove" simply drops ectopic rows.

Usage

```
ecgRRcorrect(rr, method = c("interpolate", "cubic_spline", "remove"))
```

Arguments

rr	A data.frame with an is_ectopic logical column, as returned by ecgQualityCheck .
method	Correction method: "interpolate" (default) replaces ectopic values with linear interpolation; "cubic_spline" uses natural cubic-spline interpolation (falls back to linear when fewer than four valid beats are available in a channel); "remove" drops ectopic rows.

Value

A data.frame with corrected RR intervals. The is_ectopic column is removed from the output. The result carries the ectopic burden as attributes n_ectopic (count) and pct_ectopic (percent of beats flagged ectopic).

References

Clifford, G.D., Azuaje, F. & McSharry, P.E. (2006). *Advanced Methods and Tools for ECG Data Analysis*. Artech House.

See Also

[ecgQualityCheck](#) for detecting ectopic beats, [ecgRRintervals](#) for computing RR intervals, [ecgHRVtime](#) for time-domain HRV analysis.

Examples

```
set.seed(1)
n <- 200
rr_ms <- 800 + rnorm(n, sd = 20)
rr_ms[100] <- 430 # a premature (ectopic) beat
rr <- data.frame(channel = 1L, rr_ms = rr_ms,
                 time_sec = cumsum(rr_ms) / 1000)
qc <- ecgQualityCheck(rr)
corrected <- ecgRRcorrect(qc, method = "interpolate")
attr(corrected, "pct_ectopic")
```

ecgRRintervals

*Compute RR Intervals from Detected R-Peaks***Description**

Calculates the time intervals between consecutive R-peaks for each channel. The resulting RR interval series is the standard input for all HRV analysis functions in this package.

Usage

```
ecgRRintervals(x, peaks)
```

Arguments

x A `PhysioExperiment` object.

peaks A `data.frame` of detected peaks as returned by `ecgDetectRpeaks`, with columns `channel`, `sample`, and `time_sec`.

Value

A `data.frame` with one row per consecutive beat pair and the following columns:

channel Integer channel index (1-based).

rr_ms RR interval in milliseconds (time between successive R-peaks).

time_sec Time of the first beat in each pair (seconds from signal onset).

Returns a zero-row `data.frame` with the same column structure if fewer than two peaks are available.

References

Pan, J. & Tompkins, W.J. (1985). "A real-time QRS detection algorithm." *IEEE Transactions on Biomedical Engineering*, 32(3), 230–236. doi:[10.1109/TBME.1985.325532](https://doi.org/10.1109/TBME.1985.325532)

See Also

`ecgDetectRpeaks` for R-peak detection, `ecgHRVtime` for time-domain HRV metrics, `ecgHRVfreq` for frequency-domain HRV analysis, `ecgQualityCheck` for ectopic beat detection.

Examples

```
set.seed(1)
pe <- make_ecg(n_time = 5000, sr = 500, heart_rate = 60)
peaks <- ecgDetectRpeaks(pe)
rr <- ecgRRintervals(pe, peaks)
head(rr)
```

ecgRSA	<i>Respiratory Sinus Arrhythmia (RSA)</i>
--------	---

Description

Quantifies respiratory sinus arrhythmia – the respiration-linked oscillation of the RR interval – by the peak-valley method (Grossman) or the Porges-Bohrer method (band-limited variance of the tachogram).

Usage

```
ecgRSA(
  rr,
  resp = NULL,
  method = c("peak_valley", "porges_bohrer"),
  fs = 4,
  resp_band = c(0.12, 0.4)
)
```

Arguments

<code>rr</code>	A data.frame with columns <code>rr_ms</code> and <code>time_sec</code> (and optionally <code>channel</code>), or a numeric vector of RR intervals (ms), in which case beat times are taken as their cumulative sum.
<code>resp</code>	Optional respiration surrogate (unused by the current methods, reserved for respiration-gated peak-valley).
<code>method</code>	"peak_valley" (mean band-filtered peak-to-valley RR amplitude, default) or "porges_bohrer" (natural log of the band-filtered RR variance).
<code>fs</code>	Resampling rate (Hz) of the tachogram (default 4).
<code>resp_band</code>	Respiratory band (Hz) for filtering (default <code>c(0.12, 0.4)</code>).

Value

A list with `method`, `rsa` (the RSA magnitude: mean peak-to-valley RR in ms, or $\ln(\text{variance})$ for Porges-Bohrer), and, for "peak_valley", `n_cycles` (number of respiratory half-cycles used).

References

Grossman, P., van Beek, J. & Wientjes, C. (1990). A comparison of three quantification methods for estimation of respiratory sinus arrhythmia. *Psychophysiology*, 27(6), 702-714. Lewis, G.F. et al. (2012). Statistical strategies to quantify RSA: are commonly used metrics equivalent? *Biological Psychology*, 89(2), 349-364.

See Also

[ecgEDR](#), [ecgRRintervals](#)

Examples

```
set.seed(1)
t <- cumsum(rep(0.8, 300))
rr <- data.frame(rr_ms = 800 + 40 * sin(2 * pi * 0.25 * t) + rnorm(300, 5),
                 time_sec = t)
ecgRSA(rr)$rsa
```

ecgSampleEntropy

Sample Entropy of RR Intervals

Description

Computes sample entropy (SampEn) from RR interval data. Sample entropy measures the regularity or predictability of a time series. Lower values indicate more regular (predictable) signals, while higher values indicate more complex (irregular) signals.

Usage

```
ecgSampleEntropy(rr, m = 2L, r_factor = 0.2)
```

Arguments

rr	A data.frame with columns channel, rr_ms, and time_sec, as returned by ecgRRintervals .
m	Embedding dimension (default: 2). Length of template patterns to compare.
r_factor	Tolerance factor (default: 0.2). The tolerance r is computed as $r_factor * sd(rr_ms)$.

Value

A data.frame with one row per channel and the following columns:

channel Integer channel index.

sample_entropy Sample entropy value (nats). Lower values indicate more regular signals; higher values indicate more complex signals. NA if the series is too short or constant.

m Embedding dimension used.

r Tolerance threshold (ms) computed as $r_factor * sd(rr_ms)$.

References

Richman, J.S. & Moorman, J.R. (2000). "Physiological time-series analysis using approximate entropy and sample entropy." *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6), H2039–H2049. doi:[10.1152/ajpheart.2000.278.6.H2039](https://doi.org/10.1152/ajpheart.2000.278.6.H2039)

See Also

[ecgHRVnonlinear](#) for the combined nonlinear analysis wrapper, [ecgHRVpoincare](#) for Poincare plot descriptors, [ecgDFA](#) for detrended fluctuation analysis.

Examples

```
set.seed(1)
n <- 300
time_sec <- cumsum(rep(0.85, n))
rr_ms <- 850 + 30 * sin(2 * pi * 0.1 * time_sec) + rnorm(n, sd = 15)
rr <- data.frame(channel = 1L, rr_ms = rr_ms, time_sec = time_sec)
ecgSampleEntropy(rr)
```

ecgSignalQuality	<i>Assess ECG Signal Quality Per Channel</i>
------------------	--

Description

Computes per-channel signal quality metrics for ECG data stored in a `PhysioExperiment` object. When detected R-peak locations are provided the signal-to-noise ratio is estimated from QRS vs. baseline power; otherwise a variance-based estimate is used.

Usage

```
ecgSignalQuality(x, peaks = NULL, assay_name = NULL)
```

Arguments

x	A <code>PhysioExperiment</code> object with ECG data.
peaks	Optional <code>data.frame</code> of detected R-peaks as returned by <code>ecgDetectRpeaks</code> , with at least columns <code>channel</code> and <code>sample</code> .
assay_name	Name of the assay to use. If <code>NULL</code> the default assay is used.

Value

A `data.frame` with one row per channel and columns:

- channel** Integer channel index.
- snr_db** Signal-to-noise ratio in decibels.
- baseline_wander** RMS amplitude of the low-frequency drift.
- saturation_ratio** Fraction of samples within 1\ min or max.
- quality_score** Composite quality score in the range [0, 1], where 1 indicates excellent quality.

References

Clifford, G.D., Azuaje, F. & McSharpy, P.E. (2006). *Advanced Methods and Tools for ECG Data Analysis*. Artech House.

Shaffer, F. & Ginsberg, J.P. (2017). "An overview of heart rate variability metrics and norms." *Frontiers in Public Health*, 5, 258. doi:10.3389/fpubh.2017.00258

See Also

[ecgDetectRpeaks](#) for R-peak detection, [ecgBaselineCorrect](#) for baseline wander correction, [ecgQualityCheck](#) for ectopic beat detection.

Examples

```
set.seed(1)
pe <- make_ecg_noisy(n_time = 5000, sr = 500)
ecgSignalQuality(pe)
```

make_ecg

*Create Simulated ECG PhysioExperiment***Description**

Generates a PhysioExperiment object containing synthetic ECG data with Gaussian-shaped R-peaks at regular intervals. The resulting object is suitable for testing R-peak detection, RR interval computation, and HRV analysis pipelines.

Usage

```
make_ecg(n_time = 5000, n_channels = 1, sr = 500, heart_rate = 72)
```

Arguments

n_time	Number of time points (default: 5000).
n_channels	Number of ECG channels (default: 1).
sr	Sampling rate in Hz (default: 500).
heart_rate	Heart rate in beats per minute (default: 72).

Value

A PhysioExperiment object with a single "raw" assay containing the simulated ECG signal.

References

Pan, J. & Tompkins, W.J. (1985). "A real-time QRS detection algorithm." *IEEE Transactions on Biomedical Engineering*, 32(3), 230–236. doi:[10.1109/TBME.1985.325532](#)

Clifford, G.D., Azuaje, F. & McSharry, P.E. (2006). *Advanced Methods and Tools for ECG Data Analysis*. Artech House.

See Also

[make_ecg_irregular](#) for ECG with ectopic beats, [make_ecg_pqrst](#) for ECG with full PQRST morphology, [make_ecg_noisy](#) for ECG with noise artifacts, [ecgDetectRpeaks](#) for R-peak detection.

Examples

```
pe <- make_ecg(n_time = 5000, sr = 500, heart_rate = 72)
pe
```

make_ecg_irregular	Create Simulated ECG with Irregular R-R Intervals
--------------------	---

Description

Generates a `PhysioExperiment` object containing synthetic ECG data with irregular beat timing. Every 5th beat is premature (60\ interval), followed by a compensatory pause (140\ Useful for testing ectopic beat detection and RR interval correction.

Usage

```
make_ecg_irregular(n_time = 5000, sr = 500, heart_rate = 72)
```

Arguments

n_time	Number of time points (default: 5000).
sr	Sampling rate in Hz (default: 500).
heart_rate	Base heart rate in beats per minute (default: 72).

Value

A `PhysioExperiment` object with a single "raw" assay containing the simulated ECG signal with irregular beats.

References

Clifford, G.D., Azuaje, F. & McSharry, P.E. (2006). *Advanced Methods and Tools for ECG Data Analysis*. Artech House.

Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996). "Heart rate variability: Standards of measurement, physiological interpretation and clinical use." *Circulation*, 93(5), 1043–1065.

See Also

[make_ecg](#) for regular ECG data, [ecgQualityCheck](#) for ectopic beat detection, [ecgRRcorrect](#) for ectopic beat correction, [ecgRRintervals](#) for RR interval computation.

Examples

```
pe <- make_ecg_irregular(n_time = 5000, sr = 500, heart_rate = 72)
pe
```

make_ecg_noisy

Create Simulated ECG with Noise Artifacts

Description

Generates a `PhysioExperiment` object containing synthetic ECG data contaminated with multiple noise sources: baseline wander (0.3 Hz sinusoidal drift), powerline interference (50 Hz), and broadband Gaussian noise. Useful for testing signal quality assessment, baseline correction, and filtering pipelines.

Usage

```
make_ecg_noisy(
  n_time = 5000,
  n_channels = 1,
  sr = 500,
  heart_rate = 72,
  baseline_amp = 0.3,
  powerline_amp = 0.1,
  noise_sd = 0.15
)
```

Arguments

<code>n_time</code>	Number of time points (default: 5000).
<code>n_channels</code>	Number of ECG channels (default: 1).
<code>sr</code>	Sampling rate in Hz (default: 500).
<code>heart_rate</code>	Heart rate in beats per minute (default: 72).
<code>baseline_amp</code>	Amplitude of baseline wander in arbitrary units (default: 0.3).
<code>powerline_amp</code>	Amplitude of 50 Hz powerline noise (default: 0.1).
<code>noise_sd</code>	Standard deviation of broadband Gaussian noise (default: 0.15).

Value

A `PhysioExperiment` object with a single "raw" assay containing the noisy ECG signal.

References

Clifford, G.D., Azuaje, F. & McSharry, P.E. (2006). *Advanced Methods and Tools for ECG Data Analysis*. Artech House.

Shaffer, F. & Ginsberg, J.P. (2017). "An overview of heart rate variability metrics and norms." *Frontiers in Public Health*, 5, 258. doi:10.3389/fpubh.2017.00258

See Also

[make_ecg](#) for clean ECG data, [ecgSignalQuality](#) for signal quality assessment, [ecgBaselineCorrect](#) for baseline wander removal, [ecgDetectRpeaks](#) for R-peak detection.

Examples

```
pe <- make_ecg_noisy(n_time = 5000, sr = 500)
pe
```

make_ecg_pqrst	<i>Create Simulated ECG with PQRST Morphology</i>
----------------	---

Description

Generates a `PhysioExperiment` object containing synthetic ECG data with physiologically realistic P, Q, R, S, and T wave components. Returns both the `PhysioExperiment` and a `data.frame` of known fiducial points for validation testing of waveform delineation algorithms.

Usage

```
make_ecg_pqrst(
  n_time = 10000,
  n_channels = 1,
  sr = 500,
  heart_rate = 72,
  noise_sd = 0.02
)
```

Arguments

<code>n_time</code>	Number of time points (default: 10000, i.e., 20 seconds at 500 Hz).
<code>n_channels</code>	Number of ECG channels (default: 1).
<code>sr</code>	Sampling rate in Hz (default: 500).
<code>heart_rate</code>	Heart rate in beats per minute (default: 72).
<code>noise_sd</code>	Standard deviation of baseline noise (default: 0.02).

Value

A list with two components:

pe `PhysioExperiment` object with the simulated ECG signal.

fiducials A `data.frame` with columns: `beat`, `r_peak`, `p_peak`, `q_point`, `s_point`, `t_peak`, `qrs_onset`, `qrs_offset` (all in sample indices).

References

Goldberger, A.L., et al. (2000). "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals." *Circulation*, 101(23), e215–e220. doi:10.1161/01.CIR.101.23.e215

Clifford, G.D., Azuaje, F. & McSharry, P.E. (2006). *Advanced Methods and Tools for ECG Data Analysis*. Artech House.

See Also

[make_ecg](#) for basic ECG data, [ecgDelineate](#) for PQRST waveform delineation, [ecgIntervals](#) for computing clinical ECG intervals, [ecgDetectRpeaks](#) for R-peak detection.

Examples

```
result <- make_ecg_pqrst(n_time = 10000, sr = 500, heart_rate = 72)
pe <- result$pe
fiducials <- result$fiducials
head(fiducials)
```

make_ppg

Simulate PPG with known fiducials

Description

Generates DC-offset PPG pulses from systolic and diastolic Gaussian waves plus a sharp foot dip, with optional respiratory amplitude modulation.

Usage

```
make_ppg(
  n_time = 15000,
  n_channels = 1,
  sr = 125,
  heart_rate = 72,
  dc = 2,
  sys_amp = 1,
  dia_amp = 0.4,
  crest_ms = 150,
  dia_offset_ms = 350,
  noise_sd = 0.02,
  foot_dip = 0.1,
  resp_freq = NULL,
  resp_depth = 0.3
)
```

Arguments

n_time	Number of samples.
n_channels	Number of PPG channels.
sr	Sampling rate in Hz.
heart_rate	Pulse rate in beats per minute.
dc	Non-pulsatile baseline level.
sys_amp, dia_amp	Systolic and diastolic wave amplitudes.
crest_ms	Foot-to-systolic interval in milliseconds.
dia_offset_ms	Foot-to-diastolic interval in milliseconds.
noise_sd	Gaussian noise standard deviation.
foot_dip	Depth of the local foot minimum.
resp_freq	Optional respiratory amplitude-modulation frequency in Hz.
resp_depth	Fractional respiratory modulation depth.

Value

A list with `pe`, a `PhysioExperiment`, and `truth`, a fiducial data frame (`beat`, `foot`, `systolic`, `diastolic`).

Examples

```
set.seed(1)
simulation <- make_ppg(n_time = 3000)
head(simulation$truth)
```

perfusionIndex	<i>Calculate the PPG perfusion index</i>
----------------	--

Description

Perfusion index is 100 times the pulsatile systolic-to-foot amplitude divided by the raw-signal DC level.

Usage

```
perfusionIndex(x, peaks = NULL, assay_name = NULL, agg = c("median", "mean"))
```

Arguments

x	A raw, DC-preserving PPG <code>PhysioExperiment</code> .
peaks	Optional systolic peak table from <code>ppgDetectPulses()</code> .
assay_name	Optional raw PPG assay name.
agg	Aggregation of per-pulse AC amplitudes.

Value

A per-channel data frame containing AC, DC, perfusion index, and pulse count.

References

Charlton PH, Kyriacou PA, Mant J, et al. (2022). Wearable photoplethysmography for cardiovascular monitoring. *Proceedings of the IEEE*, 110:355-381. doi:10.1109/JPROC.2022.3149785

Examples

```
simulation <- make_ppg(n_time = 5000)
perfusionIndex(simulation$pe)
```

plotEcgWave	<i>Plot Annotated ECG Waveform</i>
-------------	------------------------------------

Description

Plots the ECG waveform with P, QRS and T fiducials from [ecgDelineate](#). The detected R-peak sample indices are attached as the "peaks" attribute (a beat-editing hook); the delineation is attached as "delineation".

Usage

```
plotEcgWave(x, peaks, channel = 1, window_sec = 10, assay_name = NULL)
```

Arguments

x	A PhysioExperiment object with ECG data.
peaks	R-peak table from ecgDetectRpeaks .
channel	Channel to plot (default: 1).
window_sec	Seconds of signal to display from the start (default 10; NULL for the whole recording).
assay_name	Input assay name (default: first assay).

Value

A ggplot object with "peaks" and "delineation" attributes.

References

Pan, J. & Tompkins, W.J. (1985). A real-time QRS detection algorithm. *IEEE TBME*, 32(3), 230-236.

See Also

[ecgDelineate](#), [ecgDetectRpeaks](#)

Examples

```
pe <- make_ecg(5000, sr = 500)
pk <- ecgDetectRpeaks(pe)
plotEcgWave(pe, pk)
```

plotHRVpsd

*Plot HRV Power Spectral Density with Frequency Bands***Description**

Plots the RR-interval power spectrum with the VLF, LF and HF bands shaded and their integrated powers annotated. The PSD and band integration reuse the same internals as [ecgHRVfreq](#), so the shaded band areas equal the [ecgHRVfreq](#) band powers. The band powers are attached as the "band_power" attribute.

Usage

```
plotHRVpsd(
  rr,
  method = c("welch", "ar", "lomb"),
  detrend = FALSE,
  channel = NULL,
  vlf_band = c(0.003, 0.04),
  lf_band = c(0.04, 0.15),
  hf_band = c(0.15, 0.4)
)
```

Arguments

rr	An rr data.frame or an object from ecgProcess .
method	Spectral method ("welch" default, "ar" or "lomb"), matching ecgHRVfreq .
detrend	Smoothness-priors detrending, matching ecgHRVfreq .
channel	Channel to plot (default: first).
vlf_band, lf_band, hf_band	Band edges (Hz).

Value

A ggplot object with a "band_power" attribute.

References

Task Force (1996). Heart rate variability. *Circulation*, 93(5), 1043-1065.

See Also

[ecgHRVfreq](#)

Examples

```
pe <- make_ecg(40000, sr = 500)
rr <- ecgRRintervals(pe, ecgDetectRpeaks(pe))
plotHRVpsd(rr)
```

plotPoincare

Plot Poincare (Lorenz) Recurrence Map

Description

Scatters each RR interval against the next (RR_n vs RR_{n+1}) and overlays the Poincare SD1/SD2 dispersion ellipse from [ecgHRVpoincare](#). The SD1 and SD2 used for the ellipse are attached as the "sd1"/"sd2" attributes of the returned plot.

Usage

```
plotPoincare(rr)
```

Arguments

rr An rr data.frame or an object from [ecgProcess](#).

Value

A ggplot object with "sd1"/"sd2" attributes (per channel).

References

Brennan, M., Palaniswami, M. & Kamen, P. (2001). Do existing measures of Poincare plot geometry reflect nonlinear features of heart rate variability? *IEEE TBME*, 48(11), 1342-1347.

See Also

[ecgHRVpoincare](#), [ecgHRVnonlinear](#)

Examples

```
pe <- make_ecg(30000, sr = 500)
rr <- ecgRRintervals(pe, ecgDetectRpeaks(pe))
plotPoincare(rr)
```

plotTachogram	<i>Plot RR Tachogram</i>
---------------	--------------------------

Description

Plots the RR interval series against time, marking ectopic beats when an `is_ectopic` column is present.

Usage

```
plotTachogram(rr)
```

Arguments

`rr` An `rr` data.frame (columns `rr_ms`, `time_sec` and optionally `channel`, `is_ectopic`) or an object returned by [ecgProcess](#).

Value

A `ggplot` object.

References

Task Force (1996). Heart rate variability. *Circulation*, 93(5), 1043-1065.

See Also

[ecgRRintervals](#), [ecgQualityCheck](#)

Examples

```
pe <- make_ecg(20000, sr = 500)
rr <- ecgQualityCheck(ecgRRintervals(pe, ecgDetectRpeaks(pe)))
plotTachogram(rr)
```

ppgDetectPulses	<i>Detect systolic peaks in PPG</i>
-----------------	-------------------------------------

Description

Implements the Elgendi two event-related moving-average (TERMA) detector. The shipped windowed-sinc FIR is used in place of the original Butterworth filter. Its linear-phase group delay is explicitly undone by snapping each event-block candidate to the corresponding raw-signal maximum.

Usage

```
ppgDetectPulses(
  x,
  w1_ms = 111,
  w2_ms = 667,
  beta = 0.02,
  low = 0.5,
  high = 8,
  order = 65,
  refractory_ms = 300,
  assay_name = NULL
)
```

Arguments

x	A PPG PhysioExperiment.
w1_ms	Systolic event window in milliseconds.
w2_ms	Beat window in milliseconds.
beta	Moving-average threshold offset.
low, high	PPG band-pass cutoffs in Hz.
order	FIR order; even values are promoted to the next odd value.
refractory_ms	Minimum systolic-peak separation in milliseconds.
assay_name	Optional raw PPG assay name.

Details

This implementation uses the package's pure-R windowed-sinc FIR rather than the Butterworth filter in the original method. A strong late dirotic wave may require increasing `refractory_ms` so that it is not counted as a second systolic pulse.

Value

A data frame with `channel`, `sample`, `time_sec`, and `raw amplitude`.

References

Elgendi M, Norton I, Brearley M, Abbott D, Schuurmans D (2013). Systolic peak detection in acceleration photoplethysmograms measured from emergency responders in tropical conditions. *PLoS ONE*, 8:e76585. doi:[10.1371/journal.pone.0076585](https://doi.org/10.1371/journal.pone.0076585)

Examples

```
set.seed(1)
simulation <- make_ppg(n_time = 3000)
head(ppgDetectPulses(simulation$pe))
```

ppgQuality	<i>Assess PPG signal quality</i>
------------	----------------------------------

Description

Computes population skewness and raw kurtosis, either per channel or in non-overlapping windows.

Usage

```
ppgQuality(x, window_sec = NULL, skew_threshold = -0.3, assay_name = NULL)
```

Arguments

x	A PPG PhysioExperiment.
window_sec	Optional non-overlapping window duration in seconds.
skew_threshold	Minimum skewness SQI accepted as usable.
assay_name	Optional raw PPG assay name.

Value

A data frame with channel, segment, start time, skewness SQI, kurtosis SQI, and acceptance flag.

References

Elgendi M (2016). Optimal signal quality index for photoplethysmogram signals. *Bioengineering*, 3:21. [doi:10.3390/bioengineering3040021](https://doi.org/10.3390/bioengineering3040021)

Examples

```
simulation <- make_ppg(n_time = 2000)
ppgQuality(simulation$pe)
```

ppgRespiration	<i>Estimate respiration from PPG modulation</i>
----------------	---

Description

Extracts amplitude modulation, baseline wander, or frequency modulation from successive PPG pulses, resamples the feature to a uniform grid, and estimates its respiratory-band spectral peak.

Usage

```
ppgRespiration(
  x,
  peaks = NULL,
  method = c("am", "bw", "fm"),
  fs = 4,
  resp_band = c(0.1, 0.5),
  assay_name = NULL
)
```

Arguments

<code>x</code>	A raw PPG <code>PhysioExperiment</code> .
<code>peaks</code>	Optional systolic peak table from <code>ppgDetectPulses()</code> .
<code>method</code>	"am" for pulse amplitude, "bw" for raw systolic level, or "fm" for instantaneous pulse rate.
<code>fs</code>	Uniform feature resampling rate in Hz.
<code>resp_band</code>	Two-element respiratory band in Hz.
<code>assay_name</code>	Optional raw PPG assay name.

Value

A list containing method, per-pulse features, uniformly resampled EDR-like traces, and respiratory rates.

References

Charlton PH, Bonnici T, Mainardi L, et al. (2016). An assessment of algorithms to estimate respiratory rate from the electrocardiogram and photoplethysmogram. *Physiological Measurement*, 37:610-626. doi:[10.1088/09673334/37/4/610](https://doi.org/10.1088/09673334/37/4/610)

Examples

```
set.seed(1)
simulation <- make_ppg(n_time = 15000, resp_freq = 0.25)
ppgRespiration(simulation$pe)
```

`pulseRateVariability` *Calculate pulse-rate variability*

Description

Converts detected PPG pulses to pulse-to-pulse intervals and delegates time-, frequency-, and optional nonlinear-domain metrics to the existing HRV engine with ECG rhythm gating disabled.

Usage

```
pulseRateVariability(
  x,
  peaks = NULL,
  freq = TRUE,
  nonlinear = FALSE,
  assay_name = NULL,
  ...
)
```

Arguments

x	A PPG PhysioExperiment.
peaks	Optional peak table from <code>ppgDetectPulses()</code> .
freq	Logical; calculate frequency-domain PRV.
nonlinear	Logical; calculate nonlinear PRV.
assay_name	Optional raw PPG assay name.
...	Detection arguments passed to <code>ppgDetectPulses()</code> when peaks is NULL.

Value

A named list containing ppi, time, freq, and nonlinear.

References

Schafer A, Vagedes J (2013). How accurate is pulse rate variability as an estimate of heart rate variability? *International Journal of Cardiology*, 166:15-29. doi:[10.1016/j.ijcard.2012.03.119](https://doi.org/10.1016/j.ijcard.2012.03.119)

Examples

```
simulation <- make_ppg(n_time = 15000)
pulseRateVariability(simulation$pe, freq = FALSE)
```

pulseWaveFeatures	<i>Extract PPG pulse-wave features</i>
-------------------	--

Description

Measures foot-to-systolic rise time, pulse amplitude, dicrotic and diastolic fiducials, reflection and augmentation indices, pulse width, and optional ECG-to-PPG pulse transit time and pulse-wave velocity.

Usage

```
pulseWaveFeatures(
  x,
  peaks = NULL,
  ecg_peaks = NULL,
  distance_m = NULL,
  assay_name = NULL
)
```

Arguments

x	A raw PPG PhysioExperiment.
peaks	Optional systolic peak table from <code>ppgDetectPulses()</code> .
ecg_peaks	Optional ECG R-peak table containing sample and optionally channel.
distance_m	Optional ECG-to-PPG path length in metres for PWV.
assay_name	Optional raw PPG assay name.

Value

A data frame with one row per pulse and waveform/PTT/PWV features.

References

Elgendi M (2012). On the analysis of fingertip photoplethysmogram signals. *Current Cardiology Reviews*, 8:14-25. doi:[10.2174/157340312801215782](https://doi.org/10.2174/157340312801215782)

Examples

```
simulation <- make_ppg(n_time = 5000)
head(pulseWaveFeatures(simulation$pe))
```

respiratoryOnsets	<i>Detect respiratory cycle onsets</i>
-------------------	--

Description

Detects end-expiratory troughs in a zero-phase band-passed respiratory signal. Consecutive troughs delimit complete breaths, and the intervening maximum marks end inspiration.

Usage

```
respiratoryOnsets(
  x,
  channel = NULL,
  band = c(0.1, 0.6),
  min_period_sec = 1,
  sr = NULL,
  assay_name = NULL
)
```

Arguments

x	A <code>PhysioExperiment</code> , numeric respiration vector, or <code>rip_calibration</code> .
channel	Optional channel index or label.
band	Two-element respiratory band in Hz.
min_period_sec	Minimum breath period in seconds.
sr	Sampling rate for a numeric vector.
assay_name	Optional assay name for a <code>PhysioExperiment</code> .

Value

A data frame with one row per complete breath, containing sample and time coordinates for onset, peak, and offset plus inspiratory amplitude.

See Also

[breathingRate\(\)](#), [tidalMetrics\(\)](#), [ripCalibrate\(\)](#)

Examples

```
sr <- 10
signal <- sin(2 * pi * 0.25 * (0:599) / sr)
head(respiratoryOnsets(signal, sr = sr))
```

ripCalibrate

Calibrate two-belt respiratory inductance plethysmography

Description

Fits the Konno-Mead two-degree-of-freedom volume model from ribcage and abdominal belt signals. Least-squares calibration uses a reference volume trace; qualitative diagnostic calibration (QDC) equalizes the variability of the two belt contributions.

Usage

```
ripCalibrate(
  rc,
  ab,
  reference = NULL,
  sr = NULL,
  method = c("qdc", "lsq"),
  band = c(0.1, 0.6),
  window = NULL,
  fit_intercept = FALSE,
  known_volume = NULL
)
```

Arguments

<code>rc</code>	Ribcage signal as a numeric vector, or a <code>PhysioExperiment</code> .
<code>ab</code>	Abdominal numeric vector, or a channel index/label when <code>rc</code> is a <code>PhysioExperiment</code> .
<code>reference</code>	Optional reference volume signal, required for "lsq".
<code>sr</code>	Sampling rate in Hz for numeric signals. A <code>PhysioExperiment</code> supplies its own sampling rate.
<code>method</code>	Calibration method, "qdc" or "lsq".
<code>band</code>	Two-element respiratory band in Hz.
<code>window</code>	Optional inclusive calibration sample range <code>c(lo, hi)</code> .
<code>fit_intercept</code>	Logical; include an intercept in least squares.
<code>known_volume</code>	Optional known inspiratory volume used for absolute scaling.

Value

A `rip_calibration` object containing coefficients, calibration diagnostics, and the calibrated volume-proportional signal.

References

Konno K, Mead J (1967). Measurement of the separate volume changes of rib cage and abdomen during breathing. *Journal of Applied Physiology*, 22:407-422.

Sackner MA, Watson H, Belsito AS, et al. (1989). Calibration of respiratory inductive plethysmograph during natural breathing. *Journal of Applied Physiology*, 66:410-420. doi:[10.1152/jappl.1989.66.1.410](https://doi.org/10.1152/jappl.1989.66.1.410)

Examples

```
sr <- 10
time <- (0:599) / sr
rc <- 2 * sin(2 * pi * 0.25 * time)
ab <- sin(2 * pi * 0.25 * time)
ripCalibrate(rc, ab, sr = sr)
```

smoothnessPriorsDetrend

Smoothness-priors detrending

Description

Removes slow trends from a uniformly sampled series (e.g. an interpolated RR tachogram) using the smoothness-priors method of Tarvainen et al. (2002). The trend estimate is $(I + \lambda^2 D_2^\top D_2)^{-1} z$ where D_2 is the second-order difference operator, and the detrended (stationary) component is $z_{stat} = (I - (I + \lambda^2 D_2^\top D_2)^{-1}) z$. Larger lambda lowers the cutoff frequency, removing more low-frequency content.

Usage

```
smoothnessPriorsDetrend(series, lambda = 500, fs = 4)
```

Arguments

series	Numeric vector, uniformly sampled.
lambda	Smoothing parameter (regularisation). Higher values remove slower trends (lower cutoff). Default 500.
fs	Sampling rate of series in Hz, used only to report the approximate half-power cutoff frequency. Default 4.

Value

The detrended series (numeric vector, near-zero mean), carrying attributes lambda and cutoff_hz (the approximate half-power cutoff of the implied high-pass filter).

References

Tarvainen, M.P., Ranta-aho, P.O. & Karjalainen, P.A. (2002). An advanced detrending method with application to HRV analysis. *IEEE Transactions on Biomedical Engineering*, 49(2), 172-175.

See Also

[ecgHRVfreq](#)

Examples

```
t <- seq(0, 60, by = 0.25)
x <- 0.01 * t + sin(2 * pi * 0.25 * t) # slow trend + HF oscillation
xd <- smoothnessPriorsDetrend(x, lambda = 500, fs = 4)
abs(mean(xd)) < 1e-6
```

tidalMetrics	<i>Calculate tidal breathing metrics</i>
--------------	--

Description

Derives inspiratory and expiratory timing, duty cycle, tidal amplitude, mean inspiratory flow, breathing rate, and minute ventilation.

Usage

```
tidalMetrics(
  x,
  onsets = NULL,
  channel = NULL,
  calibration = NULL,
  band = c(0.1, 0.6),
  min_period_sec = 1,
  sr = NULL,
  assay_name = NULL
)
```

Arguments

x	A respiration signal source accepted by respiratoryOnsets() .
onsets	Optional precomputed table from respiratoryOnsets() .
channel, band, min_period_sec, sr, assay_name	As in respiratoryOnsets() .
calibration	Optional rip_calibration or positive numeric amplitude-to-liters scale.

Value

A named list containing per-breath breaths and per-channel summary data frames.

Examples

```
sr <- 10
signal <- sin(2 * pi * 0.25 * (0:599) / sr)
tidalMetrics(signal, sr = sr)
```

ventilatoryThreshold *Detect ventilatory thresholds from incremental exercise data*

Description

VT1 is estimated by V-slope breakpoint regression of VCO₂ on VO₂. VT2 is estimated above VT1 from the breakpoint of VE on VCO₂.

Usage

```
ventilatoryThreshold(  
  vo2,  
  vco2 = NULL,  
  ve = NULL,  
  time = NULL,  
  average_n = 1L,  
  min_segment = 3L  
)
```

Arguments

vo2	Numeric VO ₂ vector, or a data frame containing vo2, vco2, and ve.
vco2, ve	Numeric vectors matching vo2 when it is not a data frame.
time	Optional time vector. For data-frame input, an existing time column is used unless this argument is supplied.
average_n	Positive integer number of ordered samples per block mean.
min_segment	Minimum number of observations fitted on each side of a breakpoint.

Value

A ventilatory_threshold object with VT1, VT2, and the data actually fitted.

References

Beaver WL, Wasserman K, Whipp BJ (1986). A new method for detecting anaerobic threshold by gas exchange. *Journal of Applied Physiology*, 60:2020-2027. doi:10.1152/jappl.1986.60.6.2020

Examples

```
vo2 <- seq(1, 3, by = 0.05)  
vco2 <- ifelse(vo2 <= 2, 0.9 * vo2, 1.8 + 1.3 * (vo2 - 2))  
ventilatoryThreshold(vo2, vco2, ve = 25 * vco2)
```

Index

breathingRate, [3](#)
breathingRate(), [57](#)

dfaThresholdCrossings, [4, 16](#)

ecgAnalyze, [5, 34](#)
ecgBaselineCorrect, [5, 34, 42, 45](#)
ecgBeatSQI, [7, 8, 14](#)
ecgBeatSQIgate, [7, 8, 34](#)
ecgDelineate, [9, 14, 32, 33, 46, 48](#)
ecgDetectAF, [10, 13, 25, 27, 30](#)
ecgDetectPVC, [12, 21](#)
ecgDetectRpeaks, [6, 7, 9, 10, 12, 13, 13, 18, 33, 34, 38, 41, 42, 45, 46, 48](#)
ecgDFA, [15, 16, 22, 27–29, 40](#)
ecgDFArolling, [4, 16](#)
ecgEctopicBurden, [17](#)
ecgEDR, [17, 39](#)
ecgHRR, [19](#)
ecgHRT, [20, 35](#)
ecgHRVasymmetry, [21, 23](#)
ecgHRVautonomic, [23](#)
ecgHRVfreq, [5, 10, 24, 28, 31, 32, 38, 49, 59](#)
ecgHRVgeometric, [5, 26, 29](#)
ecgHRVnonlinear, [5, 10, 15, 25, 27, 29, 31, 40, 50](#)
ecgHRVpoincare, [15, 22, 23, 27, 28, 28, 40, 50](#)
ecgHRVsegmented, [26, 29](#)
ecgHRVtime, [5, 10, 11, 20, 25, 26, 28, 29, 30, 31, 32, 36–38](#)
ecgHRVtimevarying, [31](#)
ecgIntervals, [5, 10, 32, 46](#)
ecgProcess, [5, 34, 49–51](#)
ecgPRSA, [21, 35](#)
ecgQualityCheck, [6, 11, 17, 31, 36, 37, 38, 42, 43, 51](#)
ecgRRcorrect, [17, 25, 34, 36, 37, 43](#)
ecgRRintervals, [10, 12–15, 20, 21, 23–25, 27, 28, 30–33, 35–37, 38, 39, 40, 43, 51](#)

ecgRSA, [18, 39](#)
ecgSampleEntropy, [15, 27–29, 40](#)
ecgSignalQuality, [6, 10, 14, 36, 41, 45](#)

make_ecg, [42, 43, 45, 46](#)
make_ecg_irregular, [42, 43](#)
make_ecg_noisy, [42, 44](#)
make_ecg_pqrst, [42, 45](#)
make_ppg, [46](#)

perfusionIndex, [47](#)
plotEcgWave, [48](#)
plotHRVpsd, [49](#)
plotPoincare, [50](#)
plotTachogram, [51](#)
ppgDetectPulses, [51](#)
ppgDetectPulses(), [47, 54–56](#)
ppgQuality, [53](#)
ppgRespiration, [53](#)
pulseRateVariability, [54](#)
pulseWaveFeatures, [55](#)

respiratoryOnsets, [56](#)
respiratoryOnsets(), [3, 60](#)
ripCalibrate, [57](#)
ripCalibrate(), [57](#)

smoothnessPriorsDetrend, [24, 59](#)

tidalMetrics, [60](#)
tidalMetrics(), [57](#)

ventilatoryThreshold, [61](#)

withProvenance, [34](#)