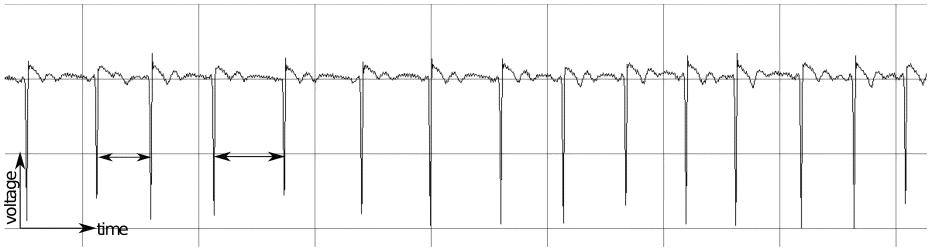
ESTIMATING HEART RATE

GÜNTHER SAWITZKI



Contents

Corrections and comments are welcome.

Date: June 4, 2014.

These notes are internal RFC for the R project RHRV, available from R-Forge. Material from RHRV may be used without explicit quotation.

Typeset, with minor revisions: June 4, 2014 from cvs *Revision* : 135

`gs@statlab.uni-heidelberg.de` .

1. Purpose

Heart rate variation is a non-invasive indicator of the physical state. However, heart rate variation is a derived variable, based on heart rate, and heart rate by itself is not an observable variable, but a derived construct.

There are several sources that can be used to derive the heart rate. The quality of the heart rate data, and hence the possibilities to assess the heart rate, depends on the quality of the original data and the preprocessing steps.

The classical means is to register some form of pulse. In all of these, classical mehtods the basic information is some local blood pressure, and the pulse is derived from it. Typically this information is collected as an average over a small number of beats ($\approx$ 15 beats) or a small time interval (e.g. 15 s).

More precise information can be derived from ECG information. The ECG records the potential difference between two or more electrodes at a chosen sampling rate. Typical amplitudes range from -0.5mV to 2 mv. As of 2014, the standard seems to be recording at 1024 Hz with 3 to 12 electrodes for clinical measurements. Technical facilities allow ambulatory sampling for continuous time, typically using two electrodes.

In clinical environment, the ECG recording is usually commented, and the commenting annotations are an additional source of information (Figure 1).

Figure 1. Annotated ECG

Heart rate, and heart rate variation are only means to judge the physical state, and oxygen supply ported by the blood may be one of the better indicators. Advanced devices can provide non-invasive measure of the oxygen concentration. This combines information circulation and respiration effects.

At present, we concentrate on ECG based data. These are common in clinical environment, and in ambulatory devices as the Polar series of monitors.

For evaluation, we concentrate on the RHRV package for R (version 4.0) Mendez *et al.* [2014].

Name	Variable	Unit, Remarks
Time	beat time	[s]
niHR	(single beat) heart rate	[beats/min] (rounded?)
RR	beat duration	[ms] (rounded?)

TABLE 1. Raw data inventory

2. RHRV Data Import

The data source for all ECG based data is an ECG recording. From this ECG recording, a beats data set is generated. Usually, pattern recognition is applied to detect QRS signals, and the time of the R component is reported as the beat time. This step may already include additional filtering.

RHRV can import various data formats. Section 5.2: "Reading several file formats" of the RHRV tutorial gives some of the common facilities. The LoadBeat function is a common interface for loading heart beat data. In particular, `LoadBeatAscii` loads an ASCII file with the time of beats, one beat per line. The time scale can be specified by the `scale` parameter of `LoadBeatAscii`.

The internal data is added to an extensible R list, a `HRVData` structure in terms of the RHRV tutorial (García *et al.* [2014]). At this point, the data are a vector of beat times [s], stored as a data frame (one variable) in component `$Beat`.

Input

```
hrv.data  = CreateHRVData()
hrv.data = SetVerbose(hrv.data, TRUE )
hrv.data = LoadBeatAscii(hrv.data, "example.beats",
       RecordPath = "../beatsFolder")
```

Output

```
** Loading beats positions for record: example.beats **
   Path: ../beatsFolder
   Scale: 1
   Date: 01/01/1900
   Time: 00:00:00
   Number of beats: 17360
```

The data structure is augmented by derived information in a second step (see Section 4.1.2 "Calculating HR and filtering" of García *et al.* [2014]).

Input

```
hrv.data = BuildNIHR(hrv.data)
```

Output

```
** Calculating non-interpolated heart rate **
   Number of beats: 17360
```

At this point, the data are as given in Table 1 and contained it the `Beat` component of the `HRVData` list. The data structure maintained by `RHRV` may contain additional information.

There may be beats missing, due to the previous processing steps, and there may be gremlins that are generated by false events from the signal detection. An additional step may remove some of the gremlins. `FilterNIHR` uses adaptive thresholds for rejecting those beats different from the given threshold more than a certain value. The rule for beat acceptation or rejection is to compare

Name	Variable	Unit, Remarks
Time	beat time. Note: some beats have been filtered.	[s]
niHR	(single beat) heart rate	[beats/min] (??? rounded?)
RR	beat duration	[ms] (??? rounded?)

TABLE 2. Data inventory for filtered data

with previous, following and with the updated mean. It also applies a comparison with acceptable physiological values (default values 25 and 200 bpm). Details can be controlled by parameters for *FilterNIHR*. The data structure is similar to Table 1 on the previous page, but the semantics has changed (Table 2).

Input
```
hrvfilt.data = FilterNIHR(hrv.data)
```

Output
```
** Filtering non-interpolated Heart Rate **
   Number of original beats: 17360
   Number of accepted beats: 17259
```

As a convenience, an interpolated version of the data can be provided to allow frequency domain analysis. But note: we are not dealing with stationary processes.

Input
```
# Note that it is not necessary to specify freqhr since it matches with
# the default value: 4 Hz
hrvipl.data = InterpolateNIHR (hrvfilt.data, freqhr = 4)
```

Output
```
** Interpolating instantaneous heart rate **
   Frequency: 4Hz
   Number of beats: 17259
   Number of points: 29594
```

The data may be imbedded in a *RHRVData* structure as outlined in Figure 2 on the next page.

3. FUNCTIONS

ToDo: handle Beat not avail in plot

3.1. **Plotsignal.** For signal representation, we use a common layout.

Input
```
plotsignal <- function (signal) {
#! alpha level should depend on expected number of overlaps
par(mfrow=c(1,2))
plot(signal, col=rgb(0,0,1,0.2), pch=20, xlab="beat" )

plot(signal, type="l",
        main=deparse(substitute(signal)), xlab="beat", col=rgb(0,0,0,0.1))
points(signal, col=rgb(0,0,1,0.05), pch=20 )
}
```

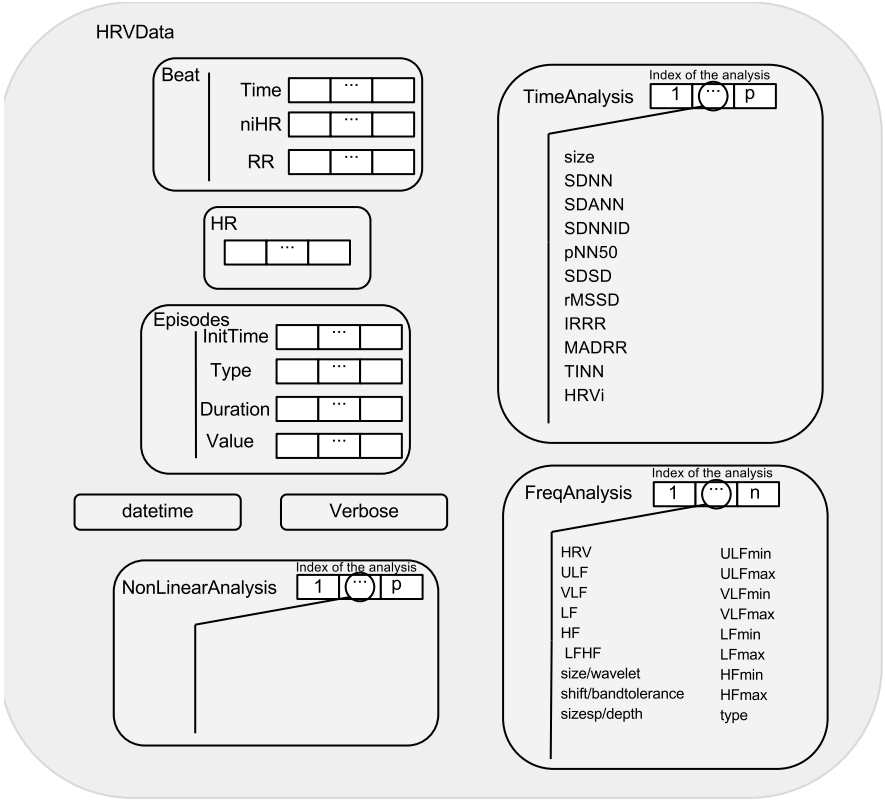


FIGURE 2. RHRV data structure: overview

3.2. **BuildNIDHR.** Since we are not interested in heart rate (or pulse), but in heart rate variation, a proposal is to use scaled differences.

Input

```
# source('/data/pulse/rhrv/pkg/R/BuildNIHR2.R', chdir = TRUE)
BuildNIDHR <-
function(HRVData, verbose=NULL) {
#------------------------------------------------------
# Obtains instantaneous heart rate variation from beats positions
# D for difference
#------------------------------------------------------
        if (!is.null(verbose)) {
                cat("  --- Warning: deprecated argument, using SetVerbose() instead ---\n    --- See help
                SetVerbose(HRVData,verbose)
        }

        if (HRVData$Verbose) {
                cat("** Calculating non-interpolated heart rate differences **\n")
        }

        if (is.null(HRVData$Beat$Time)) {
                cat("   --- ERROR: Beats positions not present... Impossible to calculate Heart Rate!! --
                return(HRVData)
        }

        NBeats=length(HRVData$Beat$Time)
        if (HRVData$Verbose) {
```

```
                cat("   Number of beats:",NBeats,"\n");
        }

  # addition gs
   #using NA, not constant extrapolation as else in RHRV
   #drr=c(NA,NA,1000.0*        diff(HRVData$Beat$Time, lag=1 , differences=2))
   HRVData$Beat$dRR=c(NA, NA,
           1000.0*diff(HRVData$Beat$Time, lag=1, differences=2))

   HRVData$Beat$avRR=(c(NA,HRVData$Beat$RR[-1])+HRVData$Beat$RR)/2

   HRVData$Beat$HRRV <- HRVData$Beat$dRR/HRVData$Beat$avRR

        return(HRVData)
}
```

4. First Inspection: example.beats

4.1. Hart Rate: example.beats.

Input
```
plotsignal(hrv.data$Beat$RR)
```

Input
```
plotsignal(hrvfilt.data$Beat$RR)
```

See Figure 3 for the unfiltered and Figure 7 on page 9 for the filtered version.

ToDo: W
outliers at
mately 2*RI
this be an
of prepr
filtering o
many impul

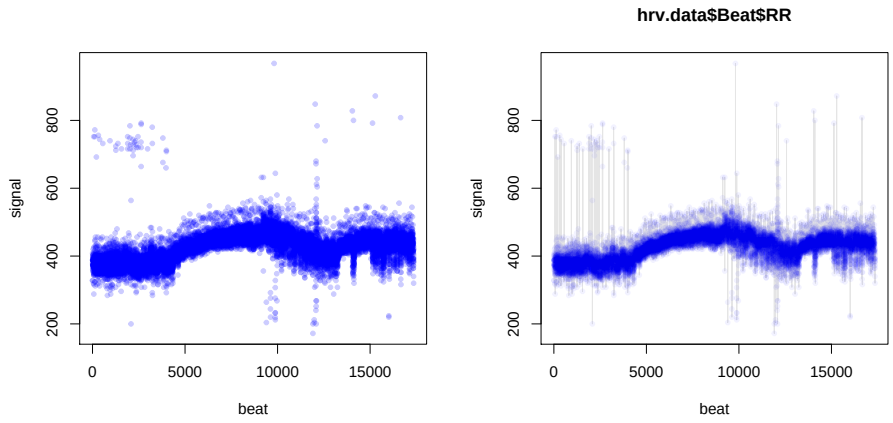


Figure 3. RHRV tutorial example.beats. Signal and linear interpolation.

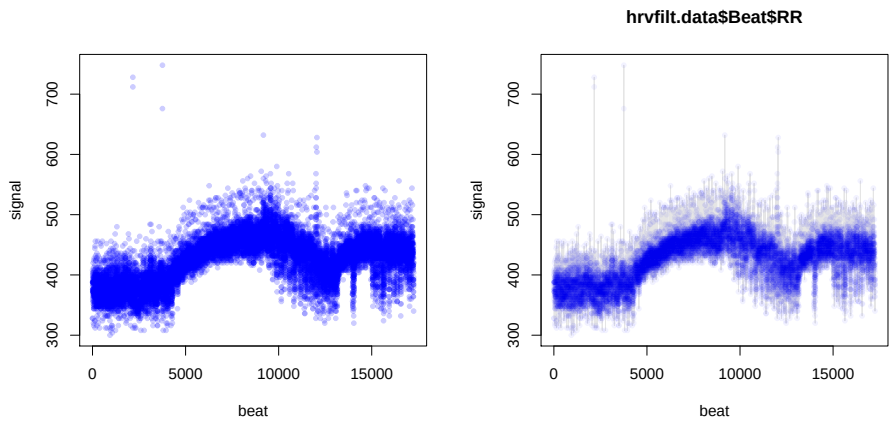


Figure 4. RHRV tutorial example.beats filtered. Signal and linear interpolation.

ToDo: Con
ing differenc
differences f

4.2. Hart Rate Variation. example beats.

Input
```
hrv.data <- BuildNIDHR(hrv.data)
```

Output
```
** Calculating non-interpolated heart rate differences **
   Number of beats: 17360
```

Input
```
HRRV <- hrv.data$Beat$HRRV
```

Input
```
plotsignal(HRRV)
```

See Figure 5,

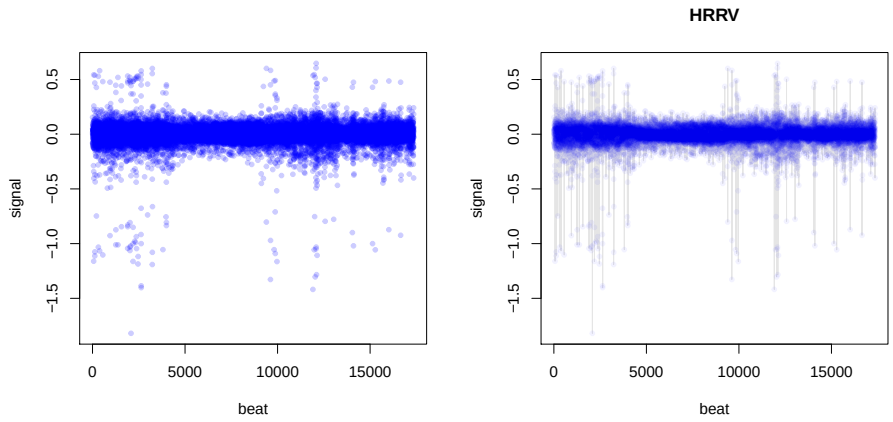


FIGURE 5. RHRV tutorial example.beats. HRRV Signal and linear interpolation.

5. First Inspection: example2.beats

Input
```
hrv2.data  = CreateHRVData()
hrv2.data = SetVerbose(hrv2.data, TRUE )
hrv2.data = LoadBeatAscii(hrv2.data, "example2.beats",
       RecordPath = "../beatsFolder")
```

Output
```
** Loading beats positions for record: example2.beats **
   Path: ../beatsFolder
   Scale: 1
   Removed 2437 duplicated beats
   Date: 01/01/1900
   Time: 00:00:00
   Number of beats: 2437
```

Input
```
hrv2.data = BuildNIHR(hrv2.data)
```

Output
```
** Calculating non-interpolated heart rate **
   Number of beats: 2437
```

Input
```
hrv2filt.data = FilterNIHR(hrv2.data)
```

Output
```
** Filtering non-interpolated Heart Rate **
   Number of original beats: 2437
   Number of accepted beats: 2434
```

5.1. **Hart Rate example 2.**

Input
```
plotsignal(hrv2.data$Beat$RR)
```

Input
```
plotsignal(hrv2filt.data$Beat$RR)
```

See Figure 6 for the unfiltered and Figure **??** on page ?? for the filtered version.

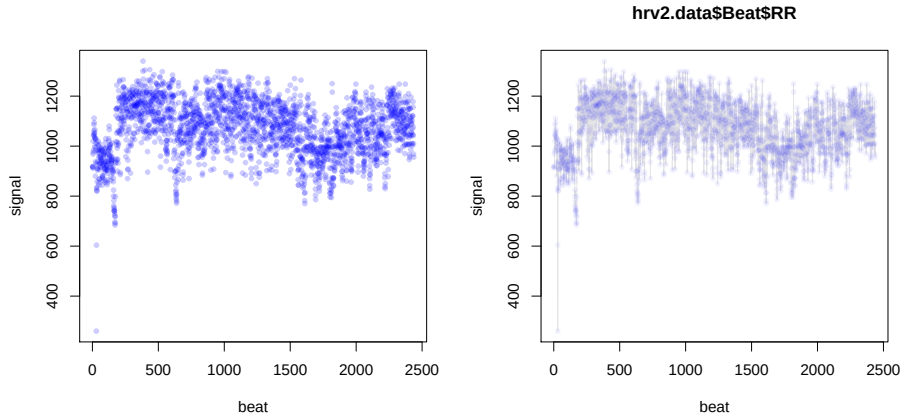


FIGURE 6. RHRV tutorial example2.beats. Signal and linear interpolation.

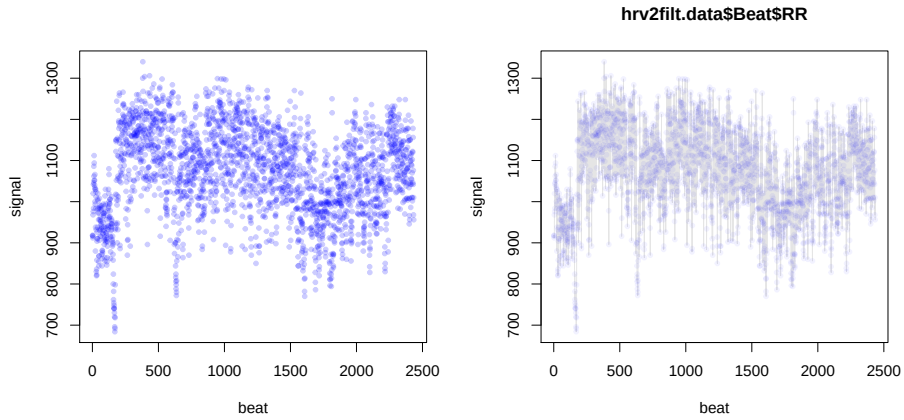


FIGURE 7. RHRV tutorial example2.beats filtered. Signal and linear interpolation.

5.2. **Hart Rate Variation: example2.beats.**

Input
```
hrv2.data <- BuildNIDHR(hrv2.data)
```

Output
```
** Calculating non-interpolated heart rate differences **
   Number of beats: 2437
```

Input
```
HRRV2 <- hrv2.data$Beat$HRRV
```

Input
```
plotsignal(HRRV2)
```

See Figure 8,

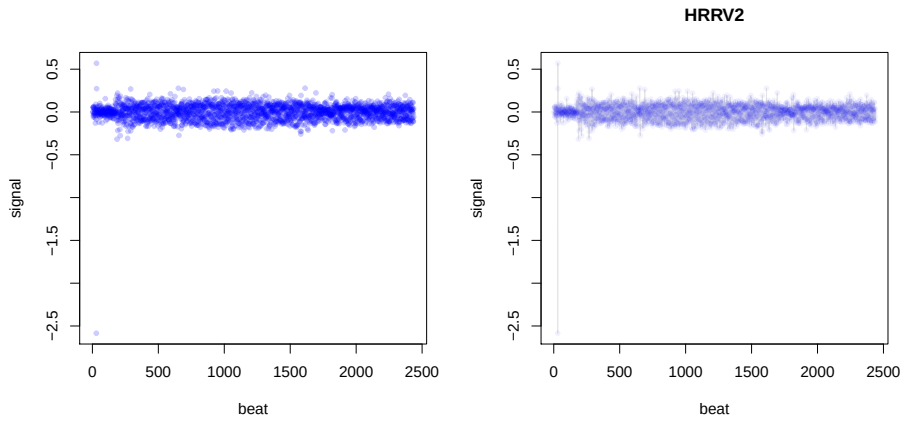


FIGURE 8. RHRV tutorial example2.beats. HRRV2 Signal and linear interpolation.

6. EXERCISES

Exercise 1. *Augment the data structure to reflect filtering.*

Exercise 2. *For analysis of heart rate variability, we are interested in effect in the range of less than 10s.*
Can you modify the displays to focus on this range?

REFERENCES

García CA, Otero A, Xosé Vila Arturo Méndez LRLn, Lado MJ (2014). "Getting started with RHRV." *Technical report*, R forge. URL `http://rhrv.r-forge.r-project.org/`.

Mendez A, Rodriguez-Linares L, Otero A, Garcia C, Vila X, Lado M (2014). *RHRV: Heart rate variability analysis of ECG data*. R package version 4.0, URL `http://CRAN.R-project.org/package=RHRV`.

Index

R session info:

Total Sweave time used: 14.817 sec. at Wed Jun 4 11:01:49 2014.

- R version 3.1.0 (2014-04-10), `x86_64-apple-darwin13.1.0`
- Locale: `en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8`
- Base packages: base, datasets, graphics, grDevices, methods, stats, tcltk, utils
- Other packages: leaps 2.9, locfit 1.5-9.1, Matrix 1.1-3, mgcv 1.7-29, nlme 3.1-117, nonlinearTseries 0.2.1, rgl 0.93.999, RHRV 4.0, tkrplot 0.0-23, TSA 1.01, tseries 0.10-32, waveslim 1.7.3
- Loaded via a namespace (and not attached): grid 3.1.0, lattice 0.20-29, quadprog 1.5-5, tools 3.1.0, zoo 1.7-11

LaTeX information:

textwidth: 6.00612in linewidth:6.00612in

textheight: 9.21922in

Bibliography style: jss

CVS/Svn repository information:

`$Source: /u/math/j40/cvsroot/lectures/src/dataanalysis/Rnw/recurrence.Rnw,v $`

`$HeadURL: svnssh://gsawitzki@scm.r-forge.r-project.org/svnroot/rhrv/gs/recurrence.Rnw +`

`$Revision: 135 $`

`$Date: 2014-02-20 12:16:47 0100`$(Thu, 20Feb2014)+$

`$name: $`

`$Author: gsawitzki $`

E-mail address: `gs@statlab.uni-heidelberg.de`

Günther Sawitzki
StatLab Heidelberg
Im Neuenheimer Feld 294
D 69120 Heidelberg