

A Software Toolkit for Nonlinear Heart Rate Variability Analysis

Constantino A García¹, Abraham Otero², Jesús Presedo¹, Xosé Vila³, Paulo Félix¹

¹Centro Singular de Investigación en Tecnoloxías da Información (CITIUS), University of Santiago de Compostela, Santiago de Compostela, Spain

²Department of Information Systems Engineering, University San Pablo CEU, Madrid, Spain

³Department of Computer Science, University of Vigo, Ourense, Spain

Abstract

This paper presents an extension to the RHRV package that enables nonlinear Heart Rate Variability (HRV) analysis. RHRV is an open source package for the R environment for statistical computing which already has support for a wide variety of time and frequency HRV analysis methods. Now RHRV also provides a phase space reconstruction algorithm that uses the delays method and that deals with the problem of discretization in the RR series. The package provides automatic estimation of both the embedding dimension and the time lag needed for the phase space reconstruction. Functionality to perform nonlinearity tests and nonlinear noise reduction is also provided. A wide variety of nonlinear statistics can be calculated, including largest Lyapunov exponent, correlation dimension, SD1/SD2 of Poincaré plots, detrended fluctuation analysis, approximate entropy and recurrence quantification analysis.

1. Introduction

In the last decades, nonlinear analysis techniques have been recognized as useful tools for analyzing signals with biological origin. Nonlinear mechanisms are thought to be involved in the genesis of most biological signals and thus, nonlinear analysis is a proper framework to study the dynamics of such complex systems. One of the most complex systems in humans is the Heart Rate (HR) regulation system. The rhythm of the heart is modulated by the sinoatrial node, which is innervated by both the sympathetic and parasympathetic branches of the autonomic nervous system. At the same time, the autonomic nervous system is largely influenced by humoral effects and haemodynamic variables, among others. There are multiple feedback loops among the mechanisms controlling heart rate. It is unlikely that all of these feedback loops result in a linear superposition. Hence the interest in using nonlinear techniques for analyzing the HR time series.

Nonlinear analysis of the HR signals has already been used to detect ventricular fibrillation, predicting heart failure situations and assessing loss in HR complexity in patients with congestive heart failure and myocardial infarction [1]. Nonlinear Heart Rate Variability analysis is a rapidly growing field. The query «nonlinear + “Heart Rate Variability”» in Google Scholar returns about 20,400 results; of these 16,000 were published after 2000; 13,600 after 2005 and 7,100 after 2010. However, support for these techniques in HRV toolkits is weak. Often, they only permit the calculation of the more basic nonlinear statistics, and completely lack functionality to perform nonlinearity tests (which should be performed before applying nonlinear analysis techniques). They also usually lack support for nonlinear noise reduction, or for dealing with the problems that arise when searching for nearest neighbours in a phase space calculated from a discretized temporal series, such as the RR intervals. Thus, researchers who use these techniques have to either implement the algorithms themselves (which is prone to errors and makes it harder to reproduce their results) or use general nonlinear analysis packages not geared towards RR analysis. Both solutions are suboptimal.

This paper presents an extension to the RHRV (R-Heart Rate Variability) open source package that enables nonlinear HRV analysis. The paper is organized as follows. Section 2 provides a quick overview on the RHRV package. Section 3 describes the nonlinear functionality that has been included in the package. In Section 4, an illustrative example of a nonlinear HRV analysis is presented. Finally, some discussion and conclusions are provided.

2. The RHRV toolkit

RHRV is an open source package for the R environment that provides a complete set of tools for performing HRV analysis. RHRV has been made available as a third party contribution to the R-CRAN repository [2]. In this paper, we shall only review the most salient aspects of the RHRV package. Further information about the package may be

found in [3], [4].

The package uses a data structure called *HRVData* to store all the information related to the HRV analysis. The *HRVData* structure is created with the *CreateHRVData* function. RHRV provides several utility functions for loading heartbeat positions or HR data into this structure. Supported formats include ASCII (*LoadBeatAscii* and *LoadBeatRR* functions), EDF (*LoadBeatEDFPlus*), Polar (*LoadBeatPolar*), Suunto (*LoadBeatSuunto*) and WFDB data files (*LoadBeatWFDB*). Once the heartbeats positions are known, the *BuildNIHR* utility can be used for computing both RR and instantaneous HR time series. Outliers present in the RR (HR) time series can be eliminated using the *FilterNIHR* function. This core utility implements an adaptive filter that shall discard those RR intervals (HR points) with unacceptable physiological values. Once the RR time series has been built, RHRV provides a wide variety of analysis techniques for studying it. Different time-domain statistics may be computed with the *CreateTimeAnalysis* function. The *CalculatePowerBand* function implements the most important frequency-domain analysis techniques, and it can use both Fourier and wavelet based analysis [4].

3. Nonlinear analysis with RHRV

All the nonlinear algorithms in the RHRV package take the *HRVData* as input data. Results from the nonlinear analysis are stored in an R list called *NonLinearAnalysis*. This list is part of the *HRVData* structure. The *NonLinearAnalysis* structure can be created using the *CreateNonLinearAnalysis* function. The nonlinear analysis structures can be referred to by an index. This field allows the *HRVData* structure to store different nonlinear analysis (usually several nonlinear analysis performed over the same signal but using different parameters).

3.1. Nonlinear testing

As other scientific methods, nonlinear analysis techniques are only applicable if some requirements are met. Thus, in order to avoid unreliable results, the nonlinearity of the HRV time series should be established before applying nonlinear techniques. The RHRV package provides some of the most widely used tests for nonlinearity as well as functionality for performing surrogate data testing [5]. The *NonlinearityTests* permits running a wide variety of tests that may be useful for determining if the HR series being analyzed is linear or not. This function includes two tests for neglected nonlinearity that are based on neural networks, the Keenan test for nonlinearity, the McLeod and Li test for nonlinearity and the Tsay's test.

Surrogate data testing tests the null hypothesis that the data was generated from a stationary linear stochastic pro-

cess with Gaussian inputs. The generated surrogate data must have the same values for linear statistics as the HR time series and it must also fulfill the null hypothesis being tested. Since the free parameters of a model consistent with the null hypothesis are the mean and the autocorrelation function, Fourier based surrogates may be used [6]. The *SurrogateTest* function generates the surrogate data and allows the user to specify the statistic S to be computed on each of the surrogates, such as the Takens-Theiler estimator or the time-asymmetry statistic. For a significance level of α , $2/\alpha - 1$ surrogates are generated and the null hypothesis is rejected when the statistic calculated for the HR series is smaller or larger than those calculated for the surrogate set, i.e. there is a significant difference.

3.2. Phase space reconstruction

We have implemented a phase space reconstruction algorithm that uses the delays method of Takens' theorem [7]. Takens' theorem states that phase space reconstruction from a single time series $s(n)$, can be achieved using the vectors:

$$\{\mathbf{s}_i = [s(i), s(i + \tau), \dots, s(i + (m - 1) \cdot \tau)]\}, \quad (1)$$

where $i = 1, 2, \dots, N - (m - 1) \cdot \tau$; N is the length of the time series; τ is the time delay; and m is the embedding dimension. The package provides automatic estimation of both the embedding dimension m and the time lag τ needed for the phase space reconstruction. The estimation of the embedding dimension is based on the algorithm described in [8]. The criteria for estimating a proper time lag τ is based on the autocorrelation function. If the time lag used for building the Takens' vectors is too small, the coordinates will have a high temporal correlation and the embedding will tend to cluster around the diagonal in the phase space. On the other hand, if the time lag chosen is too large, the coordinates may be almost uncorrelated and the resulting embedding will be very complicated.

Since the RR intervals are derived from an ECG, the RR time series may suffer from discretization problems. If the ECG was discretized very coarsely, many RR intervals shall have the same value. Thus, there will be several identical points in phase space, which may bias the nonlinear algorithms. RHRV provides functionality that addresses those situations in which the ECG was discretized very coarsely. In order to solve the discretization problem, uniform white noise of a magnitude equal to the resolution of the RR intervals is added. If the ECG from which the RR intervals were derived was sampled at f_s Hz, uniform $[-0.5, 0.5]/f_s$ white noise is added. Then, a nonlinear noise reduction algorithm is applied [5]. Note that this algorithm is only necessary if the ECG from which the RR series has been built was discretized very coarsely, i.e. the

plot of the RR time series shows many RR intervals with the same value.

3.3. Nonlinear statistics

A wide variety of nonlinear statistics can be calculated, including largest Lyapunov exponent (*CalculateMaxLyapunov*, *EstimateMaxLyapunov* and *PlotMaxLyapunov*), generalized correlation dimension (*CalculateCorrDim*, *EstimateCorrDim* and *PlotCorrDim*), SD1/SD2 of Poincaré plots (*PoincarePlot*), detrended fluctuation analysis (*CalculateDFA*, *EstimateDFA* and *PlotDFA*), sample entropy (*CalculateSampleEntropy*, *EstimateSampleEntropy* and *PlotSampleEntropy*) and recurrence quantification analysis (*RQA*). Table 1 summarizes the most important nonlinear statistics that have been included in RHRV.

Some statistics share a common name convention for their functions: the *CalculateX*, the *EstimateX* and the *PlotX* functions, being *X* the name of the statistic. This name convention reflects the common method used for computing them: first, the *CalculateX* function performs some heavy computations that are required for obtaining the value of the statistic. The statistic is then obtained as the slope of a regression involving these computations. The regression is performed with the *EstimateX* function. The *PlotX* function draws the variables involved in the regression, which may be useful to check if the statistics follow the expected behaviour. As an example, the divergence of infinitesimally close trajectories must be computed in order to obtain an estimate of the maximal Lyapunov exponent. The rate of separation for close trajectories at different times $\delta x(t)$ is obtained using the *CalculateMaxLyapunov* function. In chaotic systems, close trajectories are expected to diverge at a rate given by $|\delta x(t)| \propto e^{\lambda t} |\delta x_0|$, being δx_0 the initial divergence and λ the Lyapunov exponent. Thus, the Lyapunov exponent may be computed using the regression function implemented in *EstimateMaxLyapunov* applied over the divergence computations obtained with *ComputeMaxLyapunov*. The *PlotMaxLyapunov* function may be useful for checking if the divergence is exponential or not and for selecting a proper regression range.

4. A practical example

Listing 1 shows the RHRV code required to perform a basic nonlinear HRV analysis. Lines 1-5 load the heartbeats occurrence times, use them for building the RR time series and filter it. Line 6 creates a *NonLinearAnalysis* structure that shall store results of the nonlinear analysis performed in the following lines. Line 8 performs the set of nonlinear tests presented in section 3.1. Line 9 uses the *timeReversibility* function to perform surrogate testing.

Listing 1. Basic nonlinear analysis with RHRV

```

1 library(RHRV)
2 hrvData = CreateHRVData(Verbose = TRUE)
3 hrvData = LoadBeatAscii(hrvData, "BeatsTimes.txt")
4 hrvData = BuildNIHR(hrvData)
5 hrvData = FilterNIHR(hrvData)
6 hrvData = CreateNonLinearAnalysis(hrvData)
7 # Testing
8 hrvData = NonlinearityTests(hrvData)
9 hrvData = SurrogateTest(hrvData, significance = 0.05,
10   useFunction = timeReversibility)
11 # Computing statistics
12 hrvData = CalculateMaxLyapunov(hrvData, radius = 2.6)
13 PlotMaxLyapunov(hrvData)
14 hrvData = EstimateMaxLyapunov(hrvData,
15   regressionRange = c(1,15), doPlot = T)

```

The *timeReversibility* implements the third-order statistic $\psi(\tau) = \frac{1}{N-\tau} \sum_{i=\tau+1}^N (s_n - s_{n-\tau})^3$. This statistic measures the asymmetry of a series under time reversal. Since linear stochastic series are symmetric under time reversal, this statistic may be used for testing the assertion that the data was generated from a stationary linear stochastic process (the null hypothesis). In this example, the results for both *NonlinearityTests* and *SurrogateTest* support that the time series shows some degree of nonlinearity (p-value < 0.01 in all the tests).

Line 12 computes the divergence function of the HR series at different times $\delta x(t)$. By using the *radius* parameter it is possible to specify what is a close trajectory. In the example, close trajectories are looked for in a neighbourhood of *radius* = 2.6. Since the embedding dimension and time lag parameters for phase space reconstruction have not been specified, the *CalculateMaxLyapunov* automatically computes a proper choice for both parameters. Line 13 plots a $\delta x(t)$ Vs. *t* graphics. This is useful in order to visually assess where the scaling region in which the regression should be applied is; i.e., to find a strong linear region before the divergence function saturates. Line 14 performs the regression in order to obtain the maximal Lyapunov exponent estimate. The regression range can be specified with the *regressionRange* parameter. The *doPlot* parameter shows the plot of $\delta x(t)$ Vs. *t* with the regression superimposed to it (see Figure 1).

A copy of the R script and the recording used in the example can be obtained by contacting the authors.

5. Discussion and conclusions

In this paper, a new extension to the RHRV package for the R environment that enables nonlinear HRV analysis has been presented. RHRV implements a phase space reconstruction algorithm that uses the delays method, providing automatic estimation of both the embedding dimen-

Statistic	Interpretation
Maximum Lyapunov Exponent	Quantifies the rate of divergence of close trajectories
Generalized Correlation Dimension	Quantifies the dimensionality of the reconstructed phase space
Sample Entropy	Measures the complexity of the time series being studied
Recurrence Quantification Analysis (RQA)	Quantifies the number and duration of recurrences of a time series in its phase space
Detrended Fluctuation Analysis (DFA)	Quantifies the presence of fractal correlation properties in non-stationary data
Poincaré Plot	Characterizes the system dynamics by using a two dimensional embedding

Table 1. Summary of the most broadly used nonlinear statistics.

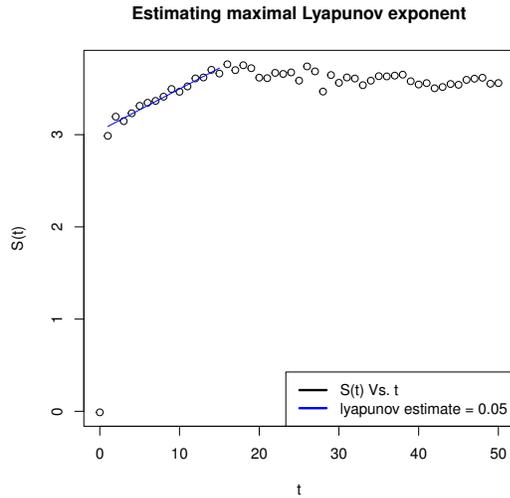


Figure 1. Maximal Lyapunov exponent estimate. Here, $\delta x(t) = S(t)$. Note the linear region of the divergence function before it saturates.

sion and the time lag. Functionality to perform nonlinearity tests and computing a wide variety of nonlinear statistics has also been included. The RHRV software, as well as a complete tutorial about its use, can be freely downloaded from [2].

One of the main advantages of the RHRV package is that it is simple to use. Free parameters present in the nonlinear algorithms have been chosen in such a way that they match those commonly used in the HRV literature and/or estimates for them are calculated automatically if the user does not provide any value. Therefore, users without much experience on nonlinear analysis can use the package. It must also be noted that, since all of these parameters may be specified by the user, RHRV does not constrain experienced researchers in any way.

An open source package such as RHRV can provide invaluable support when performing nonlinear HRV studies. By using RHRV, the users would not have to implement the most broadly used analysis algorithms themselves nor use general nonlinear analysis packages not focused on HRV analysis. Furthermore, opensource tools such as RHRV may help nonlinear HRV results to be more reproducible.

Since nonlinear algorithms usually require resource-

intensive calculations, most part of the nonlinear HRV algorithms have been implemented in the C language. However, some functions are still in beta phase and thus, they are still implemented in R, which makes them slow. Future versions of the package will solve this issue. Other future work line is the inclusion of more advanced nonlinear functionality in the package, such as multifractal analysis or Coarse Graining Spectral Analysis (CGSA).

Acknowledgments

This work was supported by the European Regional Development Fund (ERDF/FEDER) under the project CN2012/151 of the Galician Ministry of Education and the Spanish MICINN (TIN2009-14372-C03-03).

References

- [1] Voss A, Schulz S, Schroeder R, Baumert M, Caminal P. Methods derived from nonlinear dynamics for analysing heart rate variability. *Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences* 2009;367(1887):277–296.
- [2] The RHRV project homepage, <http://rhrv.r-forge.r-project.org/>, website last accessed September 2013.
- [3] Rodríguez-Liñares L, Méndez A, Lado M, Olivieri D, Vila X, Gómez-Conde I. An open source tool for heart rate variability spectral analysis. *Computer methods and programs in biomedicine* 2011;103(1):39–50.
- [4] García CA, Otero A, Vila X, Márquez DG. A new algorithm for wavelet-based heart rate variability analysis. *Biomedical Signal Processing and Control* 2013;8(6):542–550.
- [5] Kantz H, Schreiber T. *Nonlinear time series analysis*, volume 7. Cambridge university press, 2004.
- [6] Schreiber T, Schmitz A. Surrogate time series. *Physica D Nonlinear Phenomena* 2000;142(3):346–382.
- [7] Takens F. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence*, Warwick 1980. Springer, 1981; 366–381.
- [8] Cao L. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D Nonlinear Phenomena* 1997;110(1):43–50.

Address for correspondence:

Constantino A. García

Centro Singular de Investigación en Tecnoloxías da Información (CITIUS)

University of Santiago de Compostela, Spain

constantinoantonio.garcia@usc.es