Unidimensional and Multidimensional Methods for Recurrence Quantification Analysis with crqa

by Moreno I. Coco, Dan Mønster, Giuseppe Leonardi, Rick Dale and Sebastian Wallot

Abstract Recurrence quantification analysis is a widely used method for characterizing patterns in time series. This article presents a comprehensive survey for conducting a wide range of recurrence-based analyses to quantify the dynamical structure of single and multivariate time series and capture coupling properties underlying leader-follower relationships. The basics of recurrence quantification analysis (RQA) and all its variants are formally introduced step-by-step from the simplest auto-recurrence to the most advanced multivariate case. Importantly, we show how such RQA methods can be deployed under a single computational framework in R using a substantially renewed version of our **crqa** 2.0 package. This package includes implementations of several recent advances in recurrence-based analysis, among them applications to multivariate data and improved entropy calculations for categorical data. We show concrete applications of our package to example data, together with a detailed description of its functions and some guidelines on their usage.

1   Introduction

In the current article, we present the updated 2.0 version of the R package **crqa** to perform many variants of recurrence-based analyses (Coco and Dale, 2014), including some very recent developments for the treatment of multivariate and categorical data. Recurrence-based techniques allow the quantification of temporal structure and generalized autocorrelation properties of individual time series (Webber and Zbilut, 1994; Zbilut and Webber, 1992), the quantification of bivariate relationships, and coupling between two time series (Zbilut et al., 1998; Marwan and Kurths, 2002), as well as the quantification of multidimensional dynamics of multivariate time series (Wallot et al., 2016b; Wallot and Mønster, 2018). Recurrence-based techniques originate from the description and analysis of dynamical systems (Marwan et al., 2007; Marwan and Kurths, 2002) and have been widely applied to data from physics (Alex et al., 2015; Ambrożkiewicz et al., 2019; Donner and Thiel, 2007; Hilarov, 2017; Zolotova and Ponyavin, 2006), physiology (Marwan et al., 2002; Langbein et al., 2004; Thomasson et al., 2001; Mestivier et al., 2001; Mønster et al., 2016; Timothy et al., 2017), and psychology (Abney et al., 2014; Coco et al., 2018, 2016; Shockley et al., 2003; Wallot et al., 2019; Wijnants et al., 2012; Pagnotta et al., 2020), to name a few fields.

The real success of recurrence-based analyses has revolved around their power of capturing the dynamics of complex and non-stationary time series data and of time series exhibiting qualitatively different patterns along with their temporal evolution (Marwan et al., 2007). This is because recurrence-based analyses are model-free techniques that make few assumptions and hence are well suited for the analysis of complex systems. Moreover, recurrence-based analyses are versatile and can be applied to interval-scale data as well as nominal data, continuously sampled data, and inter-event data alike (Dale et al., 2011; Zbilut et al., 1998).

The new version of the **crqa** package features the integration of major developments in recurrence analysis, such as its extension to multidimensional data, as well as a key simplification of its design and a marked improvement of the underlying computational procedures. It includes useful new functions, including a tool for mining the parameter settings for continuous data and piece-wise computation of recurrence plots to mitigate the computational cost of long time series.

The remainder of this article is divided into two broad sections. In the first section, we provide a concise introduction to the core concepts of recurrence analysis from the simplest case of auto-recurrence of a unidimensional time series to the most complex case of multidimensional cross-recurrence, which is now integrated into the new version of the **crqa** package. In the second section, we showcase example applications of the different analysis methods using empirical data, hence providing a hands-on tutorial for how to use the different functions of the package.

2   Methodological background

In section 2.2.1, we briefly introduce the framework of recurrence quantification analysis (RQA hereafter) with the simplest case of a unidimensional time series. Then, in section 2.2.2, we discuss

how RQA can be applied to two different unidimensional times series. Finally, in sections 2.2.3 and 2.2.4 we explain how RQA methods can be extended to multidimensional data.

Recurrence quantification analysis (RQA)

The concept of recurrence is at the heart of all recurrence-based analyses (Marwan and Kurths, 2002; Trulla et al., 1996), which mostly apply to time series or sequenced data (but see Wallot and Leonardi (2018b)). As we will see, recurrences are often defined in terms of phase space coordinates, not directly in terms of the values of a single time series, but the concept is easily demonstrated using this case—a single time series or sequence—as a starting point. Loosely speaking, a recurrence is the repetition of a value in a sequence of data points. More precisely, recurrence in a time series x, with n data points $x_1, x_2, \ldots, x_n$ is defined as:

$$R_{ij} = \begin{cases} 1 & : x_i = x_j \\ 0 & : x_i \neq x_j \end{cases} \quad i, j = 1, 2, \ldots n \tag{1}$$

The above equation defines the elements R_{ij} of the recurrence matrix **R** in terms of identical repetitions. Such a definition is useful for a nominal sequence, where there are categorical elements that are either identical or not, and so no meaningful 'distance' norm among categories can be defined (see section 2.3.3 for an application to text data). In order to define recurrences for continuous data, we need to establish a threshold parameter (or radius parameter) ε, which provides the width of a tolerance band in the chosen distance norm within which similar but not identical values in a time series are counted as recurrent:

$$R_{ij} = \begin{cases} 1 & : |x_i - x_j| \leq \varepsilon \\ 0 & : |x_i - x_j| > \varepsilon \end{cases} \quad i, j = 1, 2, \ldots n \tag{2}$$

Setting a threshold is necessary in most cases for empirical data because such data feature intrinsic fluctuations as well as measurement error (Marwan et al., 2007). Using recurrences as defined above, we can convert any unidimensional time series x into a recurrence plot (RP), which is a two-dimensional portrait of its dynamics expressed through its recurrence characteristics.

Figure 1 shows some examples of time series of various complexity (i.e., a sinusoidal, a chaotic attractor, and white noise) and their associated RPs, given some value for the threshold parameter ε.

If a time series x constitutes the one-dimensional measurement of an underlying multidimensional system, and the dynamics of the underlying dimensions are co-dependent, then these underlying dimensions can be recovered via the method of time-delayed embedding from the unidimensional time series (Packard et al., 1980; Takens, 1981). In these cases, the time series x is delayed (or lagged) by a certain number of data points, τ, and the number of times such delays are applied to x depends on an embedding dimension parameter m. The time-shifted copies of x, x_τ, $x_{2\tau}, \ldots, x_{(m-1)\tau}$, can now be integrated into a single m-dimensional phase space, which shows the recovered multidimensional dynamics behind the measured unidimensional time series (Figure 2).

If the time series data comes from a multidimensional system, embedding the data into a higher dimensional phase space before the computation of an RP will improve the quantification of the dynamics of the systems from which that time series was recorded (Marwan et al., 2007). Now, with embedded data, recurrences are defined not in terms of the individual values of the original, unidimensional time series x but in terms of coordinates in m-dimensional phase space. It is possible to construct a total of $N = n - (m - 1)\tau$ points in phase space, with the following coordinates:

$$\begin{aligned}
\mathbf{X}_1 &= (x_1, x_{1+\tau}, x_{1+2\tau}, \ldots, x_{1+(m-1)\tau}) \\
\mathbf{X}_2 &= (x_2, x_{2+\tau}, x_{2+2\tau}, \ldots, x_{2+(m-1)\tau}) \\
&\vdots \\
\mathbf{X}_k &= (x_k, x_{k+\tau}, x_{k+2\tau}, \ldots, x_{k+(m-1)\tau}) \\
&\vdots \\
\mathbf{X}_N &= (x_N, x_{N+\tau}, x_{N+2\tau}, \ldots, x_n)
\end{aligned} \tag{3}$$
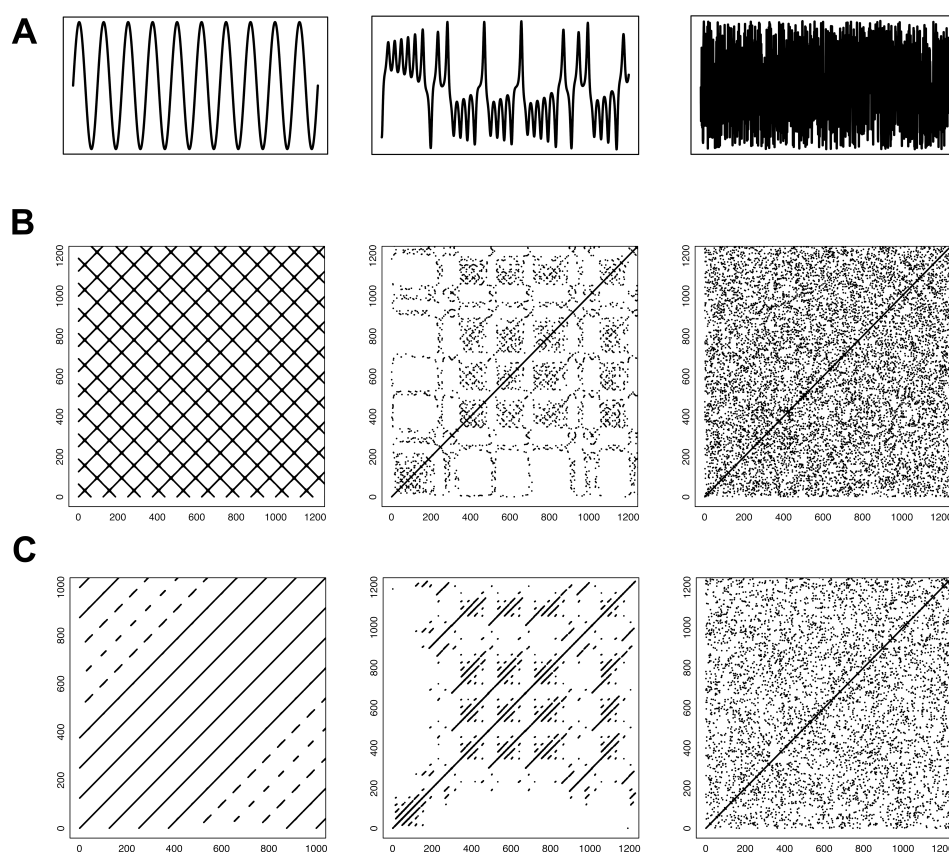
Figure 1: The three-time series in the first row (A) — a periodical sine wave, one of the dimensions of the chaotic Lorenz attractor and a white noise signal — were subjected to recurrence analysis without dimensional embedding. That is, the recurrence plots depict recurrences based on the values of the one-dimensional time series (B). The third row (C) shows their associated recurrence plots with dimensional embedding (the sine wave was embedded in 2 dimensions, the Lorenz attractor and the white noise signal in 3 dimensions).
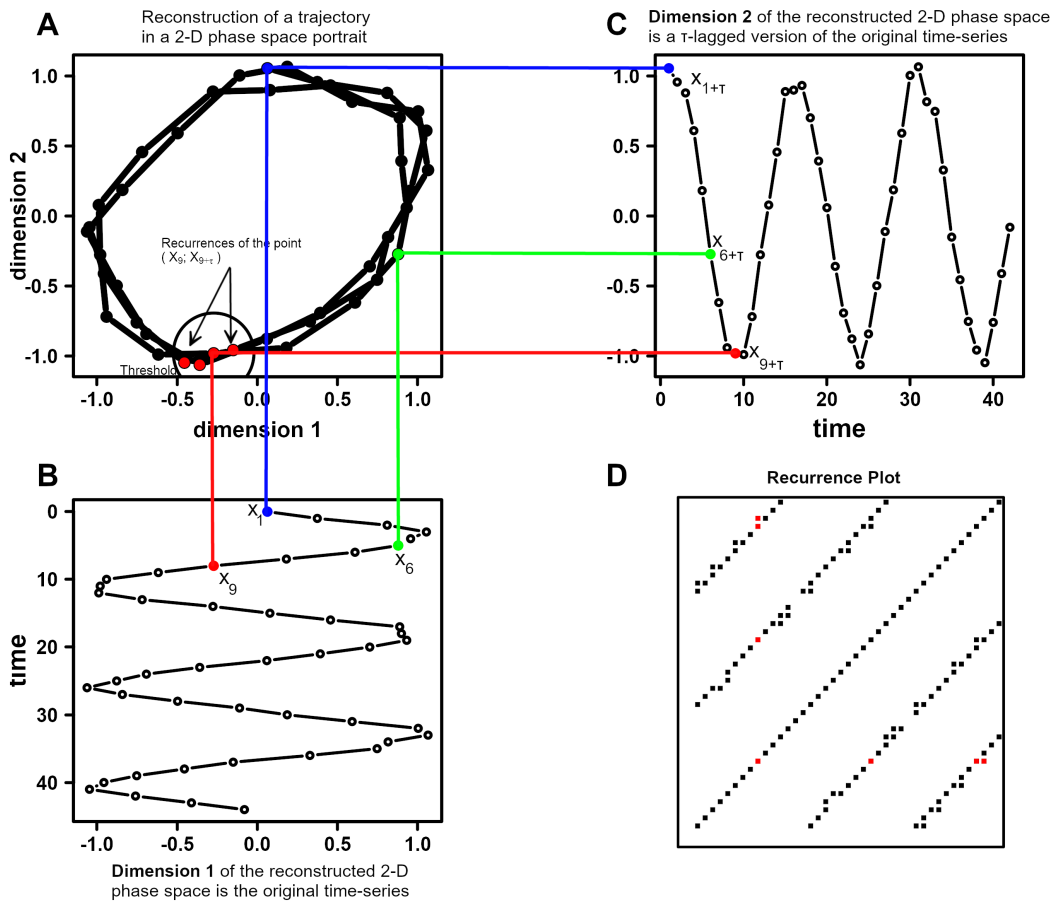
Figure 2: Graphical exemplification of the embedding method. A unidimensional time series (B)—here represented with time running along the y-axis—is embedded in two-dimensional phase space (A) by choosing as the second dimension (C) of every point in panel B the τ-lagged value (here $\tau = 3$) of the very same original unidimensional time series (B). After selecting an appropriate value of the threshold (or radius) parameter (here $\varepsilon = 0.25$), recurrence analysis can be applied to the trajectory in the two-dimensional phase space (A), and a recurrence plot is generated (D). In the recurrence plot (D), the recurrence points around the coordinates of $(X_9; X_{9+\tau})$ are highlighted in red color.

In terms of points in the m-dimensional phase space, the elements of the recurrence matrix are then given by

$$R_{ij} = \begin{cases} 1 & : ||\mathbf{X}_i - \mathbf{X}_j|| \leq \varepsilon \\ 0 & : ||\mathbf{X}_i - \mathbf{X}_j|| > \varepsilon \end{cases} \qquad i, j = 1, 2, \ldots N \tag{4}$$

where $|| \cdot ||$ is a distance norm in the m-dimensional phase space and ε is the threshold used to determine whether the points are close enough in phase space to be considered recurring or not. Examples of embedding and the resulting RPs are shown in Figure 1, lower panel, and in Figure 2, panel D. Note, however, that in the process of phase space reconstruction, the number of coordinates available in phase space is less than the number of data points in the original time series, the difference being equal to $n - N = \tau(m - 1)$. Moreover, the resulting phase space portraits do not exactly reflect the true underlying multidimensional dynamics but are isomorphic to them (Garland et al., 2016). For continuous time series (i.e., not categorical), the delay τ, the embedding dimension m and the radius ε are usually unknown and have to be estimated from the data. τ can be obtained by examining the average mutual information function (AMI) of x (Fraser and Swinney, 1986), and the false-nearest-neighbor function (FNN) of x, given some value for τ, can, in turn, be used to determine m (Kennel et al., 1992). The radius ε is then chosen to achieve the desired proportion of recurrence points that is expected given the type of data at hand (Webber Jr and Zbilut (2005); see Wallot and Mønster (2018) for practical information on parameter estimation). The **crqa** package implements functions to carry out parameter estimation semi-automatically.

RPs are a very useful visualization to qualitatively explore the dynamics of a time series. However, their main advantage is that they provide the basis to quantify the dynamics of a time series based on the patterns of recurrence points found in the plot (Zbilut and Webber, 1992). There are various measures that can be computed from RPs. Here, we briefly describe a selection of measures, including the most common ones, that are implemented in the new version of the **crqa** package.

The most basic measure is the recurrence rate (RR) or percentage recurrence, which is defined as the sum of all recurrence points in an RP divided by the area of that RP. RR provides a measure for how many individual values of a time series—or its phase space coordinates—recur over time:

$$\text{RR} = \frac{1}{N^2} \sum_{i,j=1}^{N} R_{ij} \tag{5}$$

All other measures characterize dynamics by exploiting the patterns of recurrences along with the vertical and diagonal structures of the RP (see Table 1 for a concise summary of the measures). In particular, measures based on diagonal-line structures reflect repetitions of the trajectories of the time series, whereas measures based on vertical-line structures focus on the states during which a time series slows down its dynamics. The entropy of the time series can also be computed on the basis of diagonal and vertical line structures of the RP. In version 2.0 of the **crqa** package, we include a novel entropy measure, which is based on the distribution of the areas of the rectangular structures in an RP generated from categorical time series. This measure provides a more accurate estimation of entropy over the classic diagonal-line entropy for categorical time series that predominantly evolve in terms of changes of states (Leonardi, 2018).

Cross-recurrence quantification analysis (CRQA)

Until now, we focused on quantifying the dynamics of a system by way of recurrences of a single time series. However, the concept of recurrence can be extended to that of cross-recurrence, which extends the univariate recurrence analysis to a bivariate analysis technique that allows quantification of the temporal coupling properties or similarity of two time series (Zbilut et al., 1998; Marwan et al., 2007). In other words, cross-recurrence is the recurrence of a value in a time series x, with data points $x_1, x_2, \ldots, x_n$ with the values of a time series y, with data points $y_1, y_2, \ldots, y_n$. Formally:

$$\text{CR}_{ij} = \begin{cases} 1 & : |x_i - y_j| \leq \varepsilon \\ 0 & : |x_i - y_j| > \varepsilon \end{cases} \qquad i, j = 1, 2, \ldots n \tag{6}$$

As for Equation 2, a threshold parameter ε is applied to identify similar, but not necessarily identical, values that are recurrent across the two time series. As in the univariate case, this parameter can be set to values closer to 0, forcing cross-recurrences to be identical values, such as between two nominal (categorical) sequences. Also, in the case of cross-recurrence analysis, the two time series x and y can be embedded before computing the cross-recurrence plot (CRP):

Measure	Abbreviation	Definition
Recurrence Rate	RR	$\dfrac{1}{N^2} \sum\limits_{i,j=1}^{N} R_{ij}$
Determinism	DET	$\sum\limits_{l=l_{\min}}^{N} lP(l) \Big/ \sum\limits_{l=1}^{N} lP(l)$
Average Diagonal Line Length	L	$\sum\limits_{l=l_{\min}}^{N} lP(l) \Big/ \sum\limits_{l=l_{\min}}^{N} P(l)$
Maximum Diagonal Line Length	maxL	$\max(\{l_i\}_{i=1}^{N_l}), \quad N_l = \sum\limits_{l \geq l_{\min}} P(l)$
Diagonal Line Entropy	ENTR	$-\sum\limits_{l=l_{\min}}^{N} p(l) \log p(l)$
Laminarity	LAM	$\sum\limits_{v=v_{\min}}^{N} vP(v) \Big/ \sum\limits_{v=1}^{N} vP(v)$
Trapping Time	TT	$\sum\limits_{v=v_{\min}}^{N} vP(v) \Big/ \sum\limits_{v=v_{\min}}^{N} P(v)$
Categorical Area-based Entropy	catH	$-\sum\limits_{a>1}^{N_a} p(a) \log p(a)$

Table 1: Summary and definition of RQA measures. Here, l is some diagonal line length on the recurrence plot, i.e., the number of diagonally adjacent recurrence points; $P(l)$ is the histogram or frequency distribution of such diagonal line lengths; $p(l)$ is the probability of some diagonal line length; v is some vertical line length on the recurrence plot, i.e., the number of vertical adjacent recurrence points; $P(v)$ is the histogram or frequency distribution of such diagonal line lengths; $l_{\min}$ and $v_{\min}$ are the minimum diagonal and vertical line lengths included in the measures (≥ 2); a is the value of the area of a rectangular recurrence block generated in a categorical recurrence analysis; $p(a)$ is the probability of the recurrence blocks of area a.

$$\mathrm{CR}_{ij} = \left\{ \begin{array}{ll} 1 & : ||\mathbf{X}_i - \mathbf{Y}_j|| \leq \varepsilon \\ 0 & : ||\mathbf{X}_i - \mathbf{Y}_j|| > \varepsilon \end{array} \right. \qquad i,j = 1,2,\ldots N \tag{7}$$

Commonly, it is expected that x and y have the same number of data points, and that the delay and embedding dimension parameters, τ and m, have to be the same too (see Wallot and Mønster (2018); Wallot and Leonardi (2018a) for practical aspects of parameter estimation)[1]. The recurrence measures obtained from cross-recurrence quantification analysis are calculated in the same way as for the univariate recurrence quantification analysis (see Table 1). However, the values for cross-recurrence now reflect the coupled dynamics of the two time series (Shockley et al., 2002) rather than the dynamics of an individual time series in univariate RQA. There is a key difference between RPs and CRPs. RPs always have recurrence points all along the main diagonal line of the plot (so-called line of identity, LOI), because a time series is by definition recurrent with itself at lag 0, which is what recurrences along the main diagonal reflect (i.e., $x_i = x_j$ when $i = j$). This is not necessarily the case for CRPs as two time series are not necessarily synchronized (x_i need not be the same as y_j when $i = j$). If the time series are not synchronized, cross-recurrences around the main diagonal are absent or sparse. The presence of a full LOI in a CRP implies that the dynamics of the two time series are identical or that they have a strictly linear relationship to each other.

Multidimensional recurrence quantification analysis (MdRQA)

Multidimensional recurrence quantification analysis (MdRQA) is one of the recent extensions of RQA (Wallot et al., 2016b) that is now also available in the **crqa** package. MdRQA allows the analysis of multidimensional time series $\mathbf{z}$ with samples $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n$, where each point (sample) in $\mathbf{z}$ has d dimensions:

---

[1]It is possible for x and y to be different lengths, and so to produce a rectangular CRP, but this is very rare in practice and it introduces some complications in computing synchrony measures

$$\mathbf{z}_k = (z_{k,1}, z_{k,2}, \ldots, z_{k,d}) \tag{8}$$

Here, recurrences are defined on the d-dimensional coordinate space made up of the points of $\mathbf{z}$:

$$R_{ij} = \begin{cases} 1 & : ||\mathbf{z}_i - \mathbf{z}_j|| \leq \varepsilon \\ 0 & : ||\mathbf{z}_i - \mathbf{z}_j|| > \varepsilon \end{cases} \quad i, j = 1, 2, \ldots n \tag{9}$$

Like their unidimensional counterparts, multidimensional time series can be embedded into a higher dimensional space. The logic of estimating the delay and embedding dimension parameters, τ and m, are the same as with univariate RQA (Wallot, 2017; Wallot and Leonardi, 2018a). However, suppose one has multivariate time series and wants to estimate embedding parameters for those time series. In that case, one should use the multivariate embedding functions which are also provided with the new version of the **crqa** package, because they provide superior estimates of embedding parameters for multidimensional time series (Wallot and Mønster, 2018).

Depending on the underlying data, MdRQA can, in principle, be used for two different purposes. On the one hand, MdRQA can be used to quantify a multidimensional construct, such as physiological arousal, by simultaneously looking at its different measurable dimensions (e.g., heart rate, breathing, and body temperature). This would be the multivariate version of how univariate RQA quantifies the dynamics of a single unidimensional time series (e.g., breathing alone). On the other hand, MdRQA can be used to examine the shared dynamics of multiple individual time series, such as, for example, three electrodermal signals measured from three members of a team performing a collaborative task. Here, MdRQA variables would be interpreted as capturing higher-order inter-correlative properties between the three signals at the level of the group (Wallot et al., 2016b).

In either case, one has to make a decision about whether to normalize the different dimensions of the time series or not. If each dimension of the multidimensional time series is normalized, for example, z-scored, it would effectively give each time series equal weight for the definition of recurrence. In particular, if one does not know how the different time series interact, or if they are measured on different scales without regard to their potential effects on one another, then normalization is recommended. Otherwise, the risk is to assign a greater weight to the time series bearing greater variance. If one is certain that the values of each dimension of the multidimensional time series are already properly scaled with regard to each other, such as when simultaneously analyzing the three dimensions of the Lorenz system (Lorenz, 1963), then one needs not—and perhaps should not—normalize the dimensions.

Multidimensional cross-recurrence quantification analysis (MdCRQA)

Multidimensional cross-recurrence quantification analysis (MdCRQA) extends MdRQA in the same way that CRQA extends RQA. Effectively, MdCRQA allows for the computation of cross-recurrences between two multidimensional time series $\mathbf{x}$ and $\mathbf{y}$ (Wallot, 2019), where cross-recurrences are defined between the two d-dimensional coordinate spaces between the points of $\mathbf{x}$ and $\mathbf{y}$:

$$\mathbf{x}_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,d}) \tag{10}$$
$$\mathbf{y}_j = (z_{j,1}, z_{j,2}, \ldots, z_{j,d}) \tag{11}$$

$$\mathrm{CR}_{ij} = \begin{cases} 1 & : ||\mathbf{x}_i - \mathbf{y}_j|| \leq \varepsilon \\ 0 & : ||\mathbf{x}_i - \mathbf{y}_j|| > \varepsilon \end{cases} \quad i, j = 1, 2, \ldots n \tag{12}$$

It is important that the different dimensions of the two multivariate time series enter the analysis in the same order. For the example of physiological arousal, if the heart rate is the first dimension in $\mathbf{x}$, breathing is the second dimension in $\mathbf{x}$, and body temperature is the third dimension in $\mathbf{x}$, then heart rate also needs to be the first dimension in $\mathbf{y}$, breathing needs to be the second dimension in $\mathbf{y}$, and accordingly body temperature needs to be the third dimension in $\mathbf{y}$. Otherwise, the resulting MdCRQA measure will not be interpretable.

Unidimensional and multidimensional cross-recurrence analysis can also be performed in a time-dependent manner. This so-called windowed cross-recurrence analysis is conceptually very similar to windowed cross-correlation analysis as in Boker et al. (2002), and it can be used to track how cross-recurrence changes over the time course. To that end, one simply partitions the time series of interest into a number of overlapping or non-overlapping sub-series and calculates CRPs for each of the sub-series. For each CRP, i.e., each sub-series of the original time series, the recurrence measures are calculated, which allows tracking changes in cross-recurrence over time.
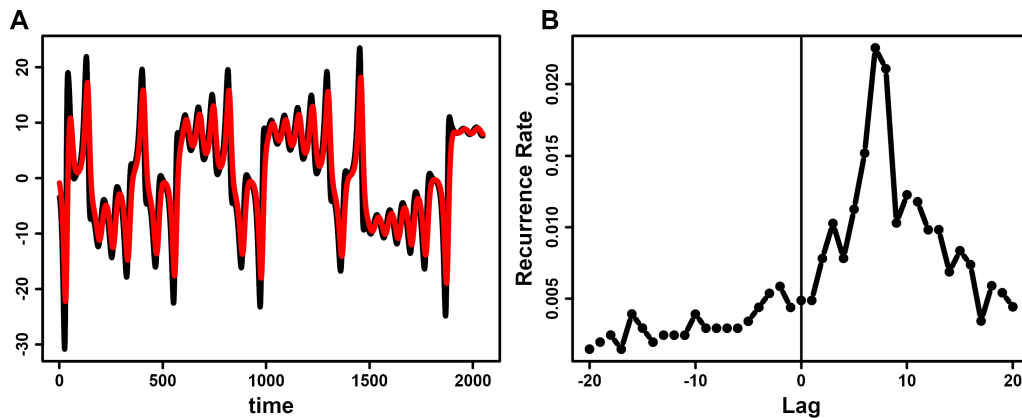
Figure 3: Illustration of DCRP. Time series of two of the three dimensions from the Lorenz system, one in red, the other one in black (A). After computation of their CRP, one can define their diagonal cross-recurrence profile (DCRP) in order to quantify their time-lagged coupling properties (B). As can be seen, the two time series are most strongly coupled at around the lags of 5 to 9, where a peak in the DCRP can be observed.

The diagonal cross-recurrence profile (DCRP)

Finally, from cross-recurrence plots of either two unidimensional time series (i.e., CRQA) or two multidimensional time series (i.e., MdCRQA), it is possible to extract the diagonal cross-recurrence profiles (DCRPs) and use them to capture leader-follower-relationships (Dale et al., 2011; Marwan et al., 2007). To that end, one has to specify a window size w for the number of lags on the recurrence plot that one wants to investigate. For example, a window-size of 10 data-points, i.e., $w = 10$ would span recurrences of $\pm$ 10 diagonals from the LOI. Specifically, this procedure determines the cross-recurrence rates of each diagonal, CR_w, by summing up all cross-recurrence points that fall along such diagonal and divide them by their length:

$$\text{CR}_w = \frac{1}{N-w} \sum_{j-i=w}^{N-w} R_{i,j} \tag{13}$$

The DCRP permits quantification of the recurrence rate over different relative lags between two time series. If the peak of cross-recurrences falls along the central diagonal, the line of synchronization (LOS), then this suggests strong coupling at lag 0 between the two time series. If the peak of cross-recurrences falls instead on one of the diagonals off the LOS, it indicates that the dynamics of one time series follow the dynamics of the other time series by some lag equal to that diagonal position (Figure 3).

Note, however, interpreting the lags in terms of the sampling rate of the underlying measured time series only applies to CRPs based on unembedded (often categorical) time series. If the time series are embedded, this means that the observed lag is based on coordinates that are made up of several data points from the respective original time series, and hence introduces a degree of uncertainty with regard to the precise time interval of the lag when one tries to map particular recurrence points back to data points of the original time series. In addition, the leader-follower interpretation of the lags cannot be granted the status of a causal interpretation. For example, a parent can deliberately 'lag' their behavior behind that of their child. In such a case, it would be odd to say definitively that the child's behavior is causing the parent's behavior. For this reason, the DCRP has to be interpreted with caution and is best treated as a general description of relative temporal relationships.

3   Package design

The **crqa** package is available from the Comprehensive R Archive Network (CRAN) at https://cran.r-project.org/web/packages/crqa/index.html. The software is written in R with the exception of the spdiags function, which contains a section written in Fortran. The main function is crqa, which takes its name from the package. This function performs all types of recurrence quantification analyses discussed in the previous section, and it returns the actual recurrence plot, along with the measures listed in Table 1 extracted from it. Here is how to call this primary function, with arguments and defaults:

Argument	Description	Usage Notes
ts1 and ts2	The input time series, either unidimensional or multidimensional time series	If auto-recurrence is used (see method = rqa), then t1 and t2 should be the same time series.
delay	A constant indicating the number of time points used to lag the time series	It corresponds to the τ of the equations.
embed	A constant indicating the number of embedding dimensions applied to the time series	It corresponds to the m of the equations.
rescale	Whether the distance matrix on which recurrence is evaluated should be rescaled and how	If rescale = 0, keep the distance matrix as is; if rescale = 1, rescale the distance matrix to its mean; if rescale = 2, rescale to distance matrix to its maximum value.
radius	A constant used to decide whether the distance between two points is small enough to be considered recurrent	For categorical time series, the radius needs to be set at values smaller than 1. For continuous time series, the value of the radius needs to tailored to the type of data and its range.
normalize	Normalize the time series	if normalize = 0, keep the time series at their original scale; if normalize = 1, normalize the time series to unit interval; if normalize = 2, z-score the time series.
mindiagline	The minimum number of contiguous points along the diagonals to consider the system into a recurrent state	The default value is usually 2, as it takes a minimum of two points to define any line.
minvertline	The minimum number of contiguous points along the vertical lines to consider the system into a recurrent state	The default value is usually 2, as it takes a minimum of two points to define any line.
tw	The Theiler window parameter	It defines the number of diagonals off the line of identity that are excluded from recurrence quantification.
whiteline	A logical flag to calculate (TRUE) or not (FALSE) empty vertical lines.	The default is FALSE, as the calculation of such lines adds on the time of computation.
recpt	A logical flag indicating whether measures of cross-recurrence are calculated directly from a recurrent plot (TRUE) or not (FALSE)	It is mostly useful if the user wants to compute joint-recurrence analysis. The RP or CRP is supplied in place of ts1, and ts2 needs to be assigned as NA.
side	A string indicating the side of the recurrence plot on which recurrence measures should be calculated	For side = upper, recurrence measures are calculated on the upper triangle of the RP, for side = lower on the lower triangle of the RP, for size = both on the full RP. Note, the line of identity is automatically excluded for upper and lower setting.
method	A string to indicate the type of recurrence analysis to perform	For method = rqa, Auto-recurrence is calculated, i.e., a unidimensional series; for method = crqa, cross-recurrence is calculated; for method = mdcrqa, multidimensional recurrence is calculated. Note, the default value is crqa.
metric	A string to indicate the type of distance metric used	To see the list of all other possible metrics, see the help for the rdist function. Note, the default is euclidean.
datatype	A string (continuous or categorical) to indicate the nature of the data type	If the time series contain categorical information, it will automatically be recoded into a continuous integer-based time series with a warning sent to the user.

Table 2: Overview of the arguments that can be used in crqa to set up the recurrence quantification analysis.

```
crqa(ts1, ts2, delay, embed, rescale = 0, radius, normalize = 0, mindiagline = 2,
    minvertline = 2, tw = 0, whiteline = FALSE, recpt = NULL, side = "upper",
    method = "crqa", metric = "euclidean", datatype = "continuous")
```

Several arguments in this function allow the user to refine aspects of the computation. For example, the user can modify the metric to obtain the distance matrix when estimating recurrence or the settings of thresholds to accept contiguous points as recurring (see Table 2 for the list of arguments of crqa with a brief explanation of each). drpfromts can be used to obtain the diagonal cross-recurrence profile, and this function is built using the crqa at its core. In fact, most arguments stay exactly the same, and we will illustrate the additional arguments that are specific to this function when describing it. Likewise, the functions wincrqa and windowdrp are built on the main function crqa and are used to compute windowed cross-recurrence. The package also contains functions, such as optimizeParam, to automatically estimate the settings for the three main parameters of RQA analyses, i.e., radius ε, delay τ, and embedding dimension m, for continuous measures. As recurrence quantification analysis is heavy on memory requirements, the package features a function (piecewiseRQA) which can be used to break down the analysis of long time series into smaller and more manageable chunks that are computationally more tractable. Finally, the package provides the user with functions to simulate data from classic examples of dynamical systems, such as the Lorenz (lorenzattractor) or categorical series with different distributions (simts). In what follows, we briefly describe the data available with the package. These data are used to illustrate the different functionalities of the crqa package.

Related Packages

A search of CRAN shows very few other packages offering alternative methods to compute recurrence quantification analysis. In particular, **tseriesChaos**, **nonlinearTseries**, and **RHRV**. The function recurr in **tseriesChaos** computes a recurrence plot for a (section of) a single unidimensional time series, but it does not compute any derived measures from the plot characterizing the dynamics of the system, nor does it handle cross recurrence plots or any of the many extensions to simple recurrence plots provided by our **crqa** package. Going a little further, the function rqa in **nonlinearTseries** returns key measures from the recurrence plots (e.g., recurrence rate) and quick visualization of the recurrence plot (available also with the function RecurrencePlot). Finally, the function RecurrencePlot in **RHRV** is simply a wrapper built on the function in **nonlinearTseries** with the same name but specifically tailored to electrocardiogram data. The functions available in **nonlinearTseries** provide very basic functionality in terms of the range of metrics available and optimization routines. They do not allow the user to examine diagonal structures for leader-follower analyses, nor explore the evolution of recurrence rate using windowed methods. All such features are integrated into **crqa**, which, to the best of our knowledge, is the most comprehensive statistical package to perform recurrence quantification analysis in R.

Data

Different types of categorical and continuous time series, both unidimensional and multidimensional, are available with the package. The command load(crqa) will load the data into the R workspace. In particular, we include a nursery rhyme "The wheels on the bus" by Verna Hills to illustrate the most basic recurrence quantification analysis. This text is a vector of 120 strings (i.e., the words of the song), and as it is extremely simple and highly repetitive, it makes it a very good example to illustrate the core concept of recurrence. Then, we move on to cross-recurrence quantification analysis and include in the data object of the package eye-tracking data from the study by Richardson and Dale (2005). In this study, a narrator describes the characters of a TV series (Friends) to a listener, who will have to later answer some comprehension questions about them, while their eye-movement is co-registered. Here, we use a single trial of this study, which is stored as a data frame of 2,000 observations of six possible screen locations that are looked at by the narrator and the listener. These are numerically coded from 1 to 6, representing a 2x3 visual grid[2]. This data will also be used to illustrate diagonal and windowed cross-recurrence. Finally, we illustrate multidimensional cross-recurrence analysis using the hand movement data from the study by Wallot et al. (2016a). In this study, dyads were instructed to cooperate in a complex LEGO joint construction task under different conditions, while their hand movements and heart rates were co-registered. Again, we select only a single trial of hand-movement from the turn-taking condition. The data frame comprises 5,799 observations from two participants (P1 and P2) for the dominant (_d) and non-dominant (_n) hand.

Using the crqa package

RQA

As explained in section 2.2.1, RQA entails computing the auto-recurrence of a unidimensional time series. In the context of the nursery rhyme, we expect clear phases of recurrence to emerge because the words greatly repeat. In order to run this analysis, we use the main function crqa and specify in the argument method that we are running an RQA analysis (method = "rqa"). As the data that we use is categorical, we need to specify in the argument datatype that the nature of the data is categorical (datatype = "categorical"). This will automatically recode the categorical states of the series (i.e., the words) into unique numerical integers so that recurrence can be computed using a radius that has to be smaller than 1 (e.g., radius = 0.01) so that we can capture the recurrence of identical words. For categorical RQA, the delay and embedding dimension have to be set to 1 (delay = 1; embed = 1[3].). We also need to set the Theiler window parameter to 1 (tw = 1) so that we can exclude the LOI from all recurrence measures. Finally, the same unidimensional time series has to be input both as ts1 and ts2 to obtain its auto-recurrence.

---

[2]There are two more states, 10 and 11, to indicate when the listener or the narrator blinked or looked outside of the screen or to identify possible blinks. They are coded with a different number so that these two states will not recur when the radius is set near 0.

[3]If embedding dimension is set to higher values, this becomes equivalent to doing recurrence on n-grams, where $m = n$. In fact, this interpretation of categorical recurrence creates bridges to traditional natural language processing, summarized in Dale et al. (2018)
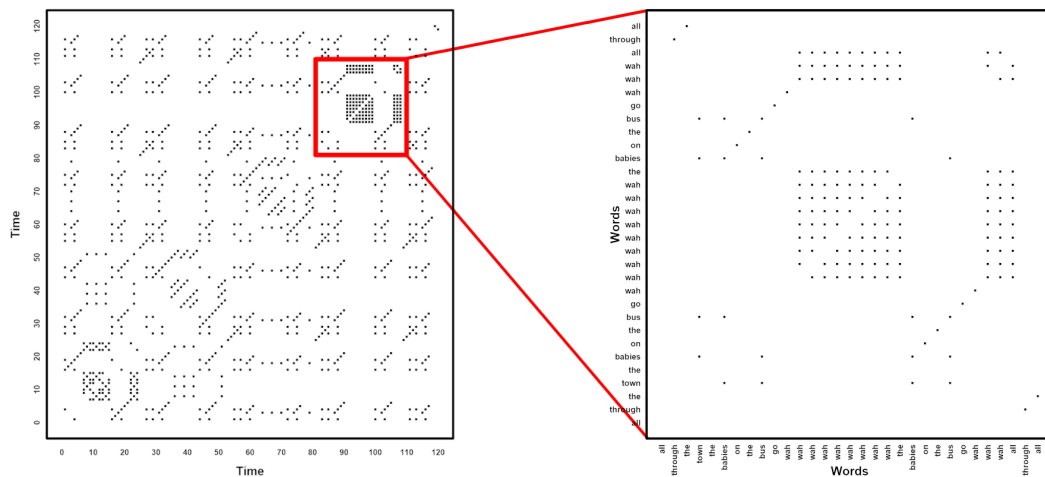
Figure 4: Recurrence quantification of the nursery rhyme 'wheels on the bus'. On the left panel, we show the full recurrence and the red box indicates the region of RP that is zoomed in on the right panel.

```
res <- crqa(text, text, delay = 1, embed = 1, rescale, radius = 0.01, normalize,
            mindiagline, minvertline, tw = 1, whiteline, recpt, side, method = "rqa",
            metric = "euclidean", datatype = "categorical")
```

We can use the plotting function `plotRP` to visualize the resulting recurrence plot. This function provides some basic arguments to change the size of the points in the plot (`pcex`), the color (`cols`), or their type (`pch`), which are taken verbatim from the generic `plot` function.

```
RP <- res$RP
parC <- list(unit = 10, labelx = "Time", labely = "Time", cols = "black", pcex = .5,
            pch = 15, las = 0, labax = seq(0, nrow(RP), 10),
            labay = seq(0, nrow(RP), 10))
plotRP(RP, parC)
```

In order to get a closer understanding of recurrences, we zoom in into a segment of the text, re-run the `crqa()` function, and visualize it (Figure 4). We add the labels of the axes (x,y), print the words vertically using the `las` argument, and decrease the temporal `unit` argument to print each individual word on the axes.

```
text_zoom <- text[81:110]
ans_zoom <- crqa(text_zoom, text_zoom, delay, embed, rescale, radius, normalize,
            mindiagline, minvertline, tw, whiteline, recpt, side, method, metric,
            datatype)
RP <- ans_zoom$RP
parC$labay <- parC$labax <- text_zoom
parC$las <- 2
parC$unit <- 1
parC$labelx <- parC$labely <- "Words"
plotRP(RP, parC)
```

As it can be clearly seen in Figure 4, the bulk of recurrence in that portion is driven by the repeated use of the single word *wah*. When looking at a few measures associated with the RP, we observe an overall determinism of 85.4%, which implies that the system is fairly repetitive, an average diagonal line length of 3.88, which implies that on average, there are sequences of four words that repeat, and a maximum diagonal length of 9, which means that the longest sequence repeating is made of 9 words. Some measures such as determinism or the average diagonal line length depending on the setting of the argument `mindiagline` (equivalent to l_{min} in Table 1). The default value of this argument is 2, as two contiguous points form a line, but it can depend on the type of data (e.g., words vs. eye-movement) or the sample rate at which it is acquired (55 Hz vs. 1,000 Hz). For example, if we have acquired data at 1,000 Hz, we would practically have one data point every 2 ms. This means that if we use a default `mindiagline` of 2, we would be considering as lines any states that contiguously repeat over a 4 ms window. This value would certainly be unrealistic for some type of responses that unfolds over a longer period of time (e.g., an eye-movement fixation lasts for an average of 200 ms).
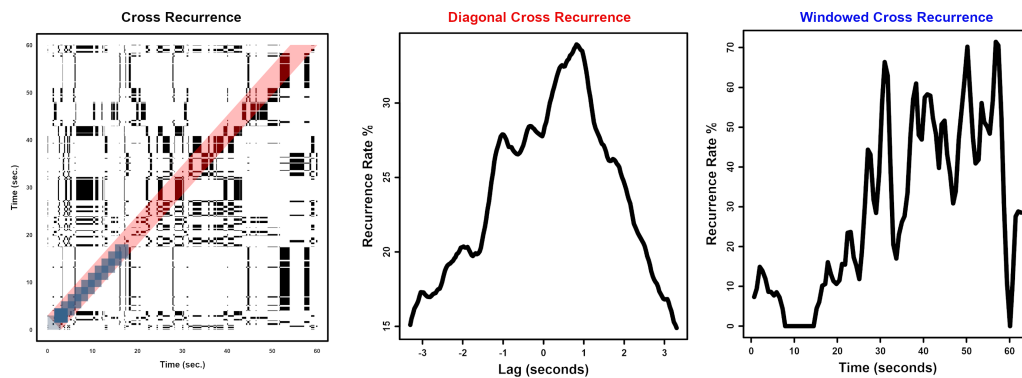
Figure 5: Cross-recurrence using a single trial of eye-movement data measured on a listener and a narrator. On the left panel, we visualize the full cross-recurrence plot. The transparent red band represents the lags around the line of coincidence that have been used to calculate the diagonal cross-recurrence plot (center panel), whereas the transparent blue squares within it are the overlapping windows used to compute the windowed cross-recurrence profile (right panel).

CRQA: `crqa`

As already explained in section 2.2.2, cross-recurrence is an extension of auto-recurrence to two different unidimensional time series. In order to run a cross-recurrence analysis with the **crqa** package, we simply need to change the `method` argument to `method = "crqa"`. Here, we illustrate its use through two time series of eye-movement data explained in section 2.3.2. Also, we input two different time series of eye-movement data, `narrator` and `listener`, rather than just one. An optional argument that is available in the `crqa` function is the `side` (upper, lower, or both) of the recurrence plot on which recurrence measures are computed. This may be useful, for example, for researchers interested in leader-follower dynamics (see Figure 5, left panel, for the visualization of the cross-recurrence plot).

Diagonal-CRQA: `drpfromts`

In section 2.2.2, we explained what diagonal cross-recurrence is and how it can be used. In the `crqa` package, this measure is computed by the function `drpfromts`, which utilizes the same arguments of the main `crqa` function plus an additional argument, `windowsize`, to define the number of lags (or diagonals) around the line of synchronization (LOS) of the CRP over which recurrence rate is computed. In the example visualized in Figure 5, center panel, we have chosen a window of 100 lags, which spans about ± 3 seconds around the LOS. We can clearly see that the peak recurrence is shifted by ≈ 1 second from the LOS, i.e., lag 0. This reflects the time taken by the listener to look at the same panel the narrator was looking at—namely, about 1 second for a listener to "catch up" to the speaker.

```
res <- drpfromts(narrator, listener, windowsize = 100, radius = 0.001,
                 delay = 1, embed = 1, rescale = 0, normalize = 0, mindiagline = 2,
                 minvertline = 2, tw = 0, whiteline = F, recpt = F, side = "both",
                 method = "crqa", metric = "euclidean", datatype = "continuous")
```

Windowed-CRQA: `windowdrp`

Windowed cross-recurrence captures the evolution of recurrence rate over time. In the context of the eye-movement data, this measure reflects how consistently listener and narrator are looking at the same scene location at any given point in time. More importantly, the windowed methodology reveals how recurrence changes over time (refer to section 2.2.5 for more details). We use the function `windowdrp` to compute this measure, which again shares the same arguments of `crqa`, plus three more that are specific to it. In particular, we have to set: (a) the size of the window that slides over the time-course, e.g., `windowsize = 100`, (b) the step that we want this window to move, e.g., `windowstep = 20`, and (c) the number of lags[4] within the window of interest over which recurrence rate is computed, e.g., `lagwidth = 50`. In Figure 5, right panel, we observe that the recurrence rate grows over time, which means that the narrator and the listener tend to look more and more at the same panels as the trial progresses.

---

[4]Note, that the number of lags cannot be greater than the size of the window.

```
res <- windowdrp(narrator, listener, windowstep = 20, windowsize = 100, lagwidth = 50,
                 radius = 0.001, delay = 1, embed = 1, rescale = 0, normalize = 0,
                 mindiagline = 2, minvertline = 2, tw = 0, whiteline = F, side = "both",
                 method = "crqa", metric = "euclidean", datatype = "continuous")
```

MdCRQA

Finally, we compute multidimensional cross-recurrence by setting the `method` argument, i.e., `method = "mdcrqa"`. Note, in order to compute multidimensional recurrence, the user needs to provide the same data frame as input to both `ts1` and `ts2`. This method is also available for `drpfromts` and `windowdrp`. Applied to the hand movement data, we first restructure the data of the two participants into independent sets. We re-use the same parameter settings of Wallot et al. (2016a) to compute MdCRQA and leave all other arguments with their default values.

```
P1 <- cbind(handset$P1_TT_d, handset$P1_TT_n)
P2 <- cbind(handset$P2_TT_d, handset$P2_TT_n)
res <- crqa(P1, P2, delay = 5, embed = 2, rescale = 0, radius = 0.1,
            normalize = 0, mindiagline = 10, minvertline = 10, tw = 0,
            whiteline, recpt, side, method = "mdcrqa", metric, datatype)
```

Breaking down the computation: `piecewiseRQA`

Often, researchers are interested in time series which contain several thousands of observations. Sometimes the dimensionality of these time series can be reduced without losing too much information, such as by down-sampling. This strategy may not always be possible or serve the researcher's purpose. Recurrence quantification analysis can require more RAM than is available in standard laptops or personal workstations, making it nearly impossible to run. In the new version of the **crqa** package, we provide the user with the `piecewiseRQA` function, which can be used to compute all different variants of recurrence quantification analysis described above on long time series. Conceptually, this function divides the time series into blocks, obtains a recurrence plot for each individual block, and then fills the original recurrence plot with all such sub-blocks, before computing the measures.[5]

For example, if we are handling a time series of 10,000 observations, we could divide it into 10 blocks of 1,000 observations each. `piecewiseRQA` has exactly the same arguments that we have already encountered in the main `crqa` function but has an additional two that are used to control the size of the block, e.g., `blockSize = 100`, and the argument `typeRQA`, which can take two options, either `full` or `diagonal`. If the value for `typeRQA` is `diagonal`, only the diagonal cross-recurrence will be computed; if `full`, the recurrence measures will be obtained out of the full plot[6]. In Figure 6, we visualize the computational speed (left panel) and memory demand (right panel) on simulated time series of sinusoids of increasing size and compare what happens if we run the `piecewiseRQA`, also with blocks of increasing size, as compared to running the main `crqa` function. We can clearly see that for time series of increasing length, memory demands are kept lower by the `piecewiseRQA` function as compared to the `crqa`. However, we can also see that there is a wide variance for blocks of different sizes. Therefore, it may be wise to explore the block sizes to find the one that can optimize the computational performance over a single trial before running the piecewise recurrence analysis on an entire dataset.

Estimating starting parameters: `optimizeParam`

The last function we showcase is `optimizeParam`, which helps the user exploring the space of values for the parameters of delay, embedding dimension and radius to compute recurrence quantification on continuous-valued time series. In particular, `optimizeParam` first estimates the average mutual information (AMI) of either the unidimensional or multidimensional time series and chooses the value that minimizes it. Then, it takes such a delay value and evaluates the embedding dimensions maximizing the false-nearest neighbors (FNN)[7]. As a last step, `optimizeParam` applies the values of delay and embedding dimension obtained to find a radius which returns a recurrence rate within a minimum and maximum value established by the user. We apply `optimizeParam()` to simulated

---

[5]This function is similar to the `crp_big` function in the long-standing CRP-toolbox for MATLAB by Marwan and colleagues: http://tocsy.pik-potsdam.de.

[6]Currently, the windowed cross-recurrence is not implemented to work with the `piecewiseRQA`.

[7]Users can also access the functions to compute delay (`MdDelay`) and embedding dimensions (`MdFnn`) independently of `optimizeParam`.
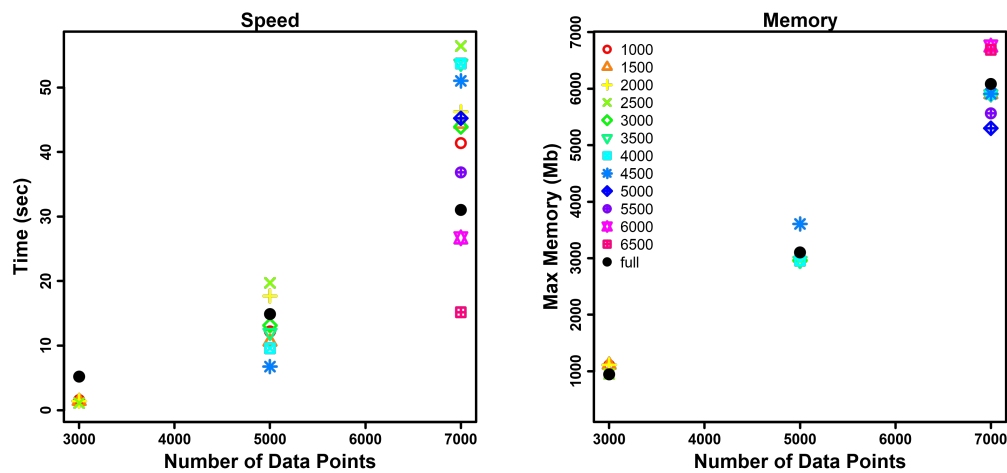
Figure 6: Evaluating the speed (time in seconds, left panel) and memory (peak RAM in MB, right panel) performance of crqa() and piecewiseRQA() for simulated data of increasing size (from 3000 to 7000 data points). We compare the use of blocks of different sizes (from 1000 to 6500 in increments of 500, coded using color and point type) with the case of running crqa() on the entire sequence of data points.

sinusoids to show the unidimensional case, and to the hand movement data of Wallot et al. (2016a) to show the multidimensional case.

In order to set up the estimation of the delay and embedding dimensions for unidimensional time series, the user has to decide how to choose the value of average mutual information (i.e., typeami = mindip, the lag at which minimal information is observed, or typeami = maxlag, the maximum lag at which minimal information is observed) and the relative percentage of information gained in FNN, relative to the first embedding dimension, when higher embeddings are considered (i.e., fnnpercent). Then, as crqa is integrated into the optimizeParam to estimate the radius, most of the arguments are the same (e.g., mindiagline or tw), except the number of values of that are considered (i.e., radiusspan = 100).

```
ts1 <- seq(0.1, 200, .1)
ts1 <- sin(ts1) + linspace(0, 1,length(ts1))
ts2 <- ts1
par <- list(method = "rqa", metric = "euclidean", maxlag =  20, radiusspan = 100,
            normalize = 0, rescale = 4,  mindiagline = 10, minvertline = 10, tw = 0,
            whiteline = FALSE, recpt = FALSE, side = "both", datatype = "continuous",
            fnnpercent  = 10,  typeami = "mindip")
results <- optimizeParam(ts1, ts2, par, min.rec = 2, max.rec = 5)
print(unlist(results))


radius emddim  delay
0.17   2       18
```

For multidimensional series, the user needs to specify the right RQA method (i.e., method = "mdcrqa"). Then, for the estimation of the delay via AMI: (1) nbins, which is the number of breaks used to define the bins within which the two-dimensional histogram (or frequency distribution) of the original and delayed time series are computed, and (2) the criterion to select the delay (firstBelow to use the lowest delay at which the AMI function drops below the value set by the threshold argument, and localMin to use the position of the first local AMI minimum). The estimation of the embedding dimensions instead needs the following arguments: (1) maxEmb, which is the maximum number of embedding dimensions considered, (2) noSamples, which is the number of randomly drawn coordinates from phase space used to estimate the percentage of false-nearest neighbors, (3) Rtol, which is the first distance criterion for separating false neighbors, and (4) Atol, which is the second distance criterion for separating false neighbors. The radius is estimated as before.

```
par$method <- "mdcrqa"
par$nbins <- 50
par$criterion <- "firstBelow"
par$threshold <- 1.6
par$maxEmb <- 20
```

```
par$numSamples <- 500
par$Rtol <- 10
par$Atol <- 2
results <- optimizeParam(P1, P2, par, min.rec = 2, max.rec = 5)
print(unlist(results))

radius  emddim  delay
0.032   11      2
```

4 Conclusion

This paper describes recurrence quantification analysis, a statistical method to characterize the nonlinear dynamics of a system. It has received a growing interest from researchers across very different fields from physiology to psychology because of its flexibility, ease of application, and explanatory power. In particular, we explain recurrence analysis from the simplest case of auto-recurrence of a unidimensional time series to the most complex case of multidimensional cross-recurrence. More importantly, we presented a significantly updated version of the **crqa** to perform all different variants of recurrence analysis described in the theoretical section of this manuscript. We showcased the different functions available in **crqa** with real data of categorical and continuous nature, and illustrated how starting parameters for continuous data could be obtained (i.e., radius, embedding dimension, and delay) as well as handling long time series in a memory-efficient way using additional functions available in **crqa**.

It is useful to end with some observations regarding the broader relevance of the package presented here. The RQA methodology and the updated crqa tap into a number of evolving problems in data analysis across various disciplines. For example, there is a drive to improve measures and models of multimodal data (Abawajy, 2015; Dale, 2015; Lahat et al., 2015), including multi-person measures (Cooke et al., 2013; López Pérez et al., 2017; Schilbach et al., 2013; von Zimmermann and Richardson, 2016; Wallot et al., 2016a). RQA and its multidimensional counterpart implemented in our package constitute an extraordinarily expansive analysis tool for exploring varied kinds of complex and multidimensional data. In addition, a demand for more dynamic quantitative analyses has now also penetrated into the social sciences (Chemero, 2011; Friedenberg, 2009; Spivey, 2008; Ward, 2002; Pagnotta et al., 2020). crqa is designed to be a comprehensive analysis package for studying the dynamics of diverse systems, especially systems that exhibit high degrees of interdependence, and that show signatures in their dynamics that are critical for understanding them.

Acknowledgments

Bibliography

J. Abawajy. Comprehensive analysis of big data variety landscape. *International journal of parallel, emergent and distributed systems*, 30(1):5–14, 2015. [p126]

D. H. Abney, A. S. Warlaumont, A. Haussman, J. M. Ross, and S. Wallot. Using nonlinear methods to quantify changes in infant limb movements and vocalizations. *Frontiers in psychology*, 5:771, 2014. [p112]

P. Alex, S. Arumugam, K. Jayaprakash, and K. Suraj. Order–chaos–order–chaos transition and evolution of multiple anodic double layers in glow discharge plasma. *Results in Physics*, 5:235–240, 2015. [p112]

B. Ambrożkiewicz, Y. Guo, G. Litak, and P. Wolszczak. Dynamical response of a planetary gear system with faults using recurrence statistics. In *Topics in Nonlinear Mechanics and Physics*, pages 177–185. Springer, 2019. [p112]

S. M. Boker, J. L. Rotondo, M. Xu, and K. King. Windowed cross-correlation and peak picking for the analysis of variability in the association between behavioral time series. *Psychological methods*, 7(3): 338, 2002. [p118]

A. Chemero. *Radical embodied cognitive science*. MIT press, 2011. [p126]

M. I. Coco and R. Dale. Cross-recurrence quantification analysis of categorical and continuous time series: an r package. *Frontiers in psychology*, 5:510, 2014. URL https://doi.org/10.3389/fpsyg.2014.00510. [p112]

M. I. Coco, L. Badino, P. Cipresso, A. Chirico, E. Ferrari, G. Riva, A. Gaggioli, and A. D'Ausilio. Multilevel behavioral synchronization in a joint tower-building task. *IEEE Transactions on Cognitive and Developmental Systems*, 9(3):223–233, 2016. [p112]

M. I. Coco, R. Dale, and F. Keller. Performance in a collaborative search task: The role of feedback and alignment. *Topics in cognitive science*, 10(1):55–79, 2018. [p112]

N. J. Cooke, J. C. Gorman, C. W. Myers, and J. L. Duran. Interactive team cognition. *Cognitive science*, 37(2):255–285, 2013. [p126]

R. Dale. An integrative research strategy for exploring synergies in natural language performance. *Ecological Psychology*, 27(3):190–201, 2015. [p126]

R. Dale, A. S. Warlaumont, and D. C. Richardson. Nominal cross recurrence as a generalized lag sequential analysis for behavioral streams. *International Journal of Bifurcation and Chaos*, 21(04): 1153–1161, 2011. URL https://doi.org/10.1142/S0218127411028970. [p112, 119]

R. Dale, N. D. Duran, and M. Coco. Dynamic natural language processing with recurrence quantification analysis. *arXiv preprint arXiv:1803.07136*, 2018. [p121]

R. Donner and M. Thiel. Scale-resolved phase coherence analysis of hemispheric sunspot activity: a new look at the north-south asymmetry. *Astronomy & Astrophysics*, 475(3):L33–L36, 2007. [p112]

A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A*, 33:1134–1140, Feb 1986. URL https://doi.org/10.1103/PhysRevA.33.1134. [p116]

J. Friedenberg. *Dynamical psychology: Complexity, self-organization and mind*. ISCE Publishing, 2009. [p126]

J. Garland, E. Bradley, and J. D. Meiss. Exploring the topology of dynamical reconstructions. *Physica D: Nonlinear Phenomena*, 334:49 – 59, 2016. URL https://doi.org/10.1016/j.physd.2016.03.006. Topology in Dynamics, Differential Equations, and Data. [p116]

V. Hilarov. Detection of the fracture zone by the method of recurrence plot. *Physics of the Solid State*, 59 (12):2401–2406, 2017. [p112]

M. B. Kennel, R. Brown, and H. D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, 45:3403–3411, Mar 1992. URL https://doi.org/10.1103/PhysRevA.45.3403. [p116]

D. Lahat, T. Adali, and C. Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015. [p126]

J. Langbein, G. Nürnberg, and G. Manteuffel. Visual discrimination learning in dwarf goats and associated changes in heart rate and heart rate variability. *Physiology & behavior*, 82(4):601–609, 2004. [p112]

G. Leonardi. A method for the computation of entropy in the recurrence quantification analysis of categorical time series. *Physica A: Statistical Mechanics and its Applications*, 512:824 – 836, 2018. ISSN 0378-4371. URL https://doi.org/10.1016/j.physa.2018.08.058. [p116]

D. López Pérez, G. Leonardi, A. Niedźwiecka, A. Radkowska, J. Rączaszek-Leonardi, and P. Tomalski. Combining recurrence analysis and automatic movement extraction from video recordings to study behavioral coupling in face-to-face parent-child interactions. *Frontiers in psychology*, 8:2228, 2017. [p126]

E. N. Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963. [p118]

N. Marwan and J. Kurths. Nonlinear analysis of bivariate data with cross recurrence plots. *Physics Letters A*, 302(5-6):299–307, 2002. [p112, 113]

N. Marwan, N. Wessel, U. Meyerfeldt, A. Schirdewan, and J. Kurths. Recurrence-plot-based Measures of Complexity and their Application to Heart-rate-variability Data. *Physical Review E*, 66(2):026702, aug 2002. doi: 10.1103/PhysRevE.66.026702. [p112]

N. Marwan, M. C. Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5):237 – 329, 2007. ISSN 0370-1573. URL https://doi.org/10.1016/j.physrep.2006.11.001. [p112, 113, 116, 119]

D. Mestivier, H. Dabiré, and N. P. Chau. Effects of autonomic blockers on linear and nonlinear indexes of blood pressure and heart rate in shr. *American Journal of Physiology-Heart and Circulatory Physiology*, 281(3):H1113–H1121, 2001. [p112]

D. Mønster, D. D. Håkonsson, J. K. Eskildsen, and S. Wallot. Physiological evidence of interpersonal dynamics in a cooperative production task. *Physiology & Behavior*, 156:24 – 34, 2016. URL https://doi.org/10.1016/j.physbeh.2016.01.004. [p112]

N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Physical review letters*, 45(9):712, 1980. URL https://doi.org/10.1103/PhysRevLett.45.712. [p113]

M. Pagnotta, K. N. Laland, and M. I. Coco. Attentional coordination in demonstrator-observer dyads facilitates learning and predicts performance in a novel manual task. *Cognition*, 201:104314, 2020. [p112, 126]

D. C. Richardson and R. Dale. Looking to understand: The coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension. *Cognitive science*, 29(6):1045–1060, 2005. [p121]

L. Schilbach, B. Timmermans, V. Reddy, A. Costall, G. Bente, T. Schlicht, and K. Vogeley. Toward a second-person neuroscience 1. *Behavioral and brain sciences*, 36(4):393–414, 2013. [p126]

K. Shockley, M. Butwill, J. P. Zbilut, and C. L. J. Webber. Cross recurrence quantification of coupled oscillators. *Physics Letters A*, 305:59–69, 2002. [p117]

K. Shockley, M.-V. Santana, and C. A. Fowler. Mutual interpersonal postural constraints are involved in cooperative conversation. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):326, 2003. [p112]

M. Spivey. *The continuity of mind*. Oxford University Press, 2008. [p126]

F. Takens. Detecting strange attractors in turbulence. In D. Rand and L.-S. Young, editors, *Dynamical Systems and Turbulence, Warwick 1980*, pages 366–381, Berlin, Heidelberg, 1981. Springer Berlin Heidelberg. URL https://doi.org/10.1007/BFb0091924. [p113]

N. Thomasson, T. J. Hoeppner, C. L. Webber Jr, and J. P. Zbilut. Recurrence quantification in epileptic eegs. *Physics Letters A*, 279(1-2):94–101, 2001. [p112]

L. T. Timothy, B. M. Krishna, and U. Nair. Classification of mild cognitive impairment eeg using combined recurrence and cross recurrence quantification analysis. *International Journal of Psychophysiology*, 120:86–95, 2017. [p112]

L. Trulla, A. Giuliani, J. Zbilut, and C. Webber Jr. Recurrence quantification analysis of the logistic equation with transients. *Physics Letters A*, 223(4):255–260, 1996. [p113]

J. von Zimmermann and D. C. Richardson. Verbal synchrony and action dynamics in large groups. *Frontiers in Psychology*, 7:2034, 2016. [p126]

S. Wallot. Recurrence quantification analysis of processes and products of discourse: A tutorial in r. *Discourse Processes*, 54(5-6):382–405, 2017. URL https://doi.org/10.1080/0163853X.2017.1297921. [p118]

S. Wallot. Multidimensional cross-recurrence quantification analysis (mdcrqa) – a method for quantifying correlation between multivariate time-series. *Multivariate Behavioral Research*, 54(2):173–191, 2019. URL https://doi.org/10.1080/00273171.2018.1512846. [p118]

S. Wallot and G. Leonardi. Analyzing multivariate dynamics using cross-recurrence quantification analysis (crqa), diagonal-cross-recurrence profiles (dcrp), and multidimensional recurrence quantification analysis (mdrqa) – a tutorial in r. *Frontiers in Psychology*, 9:2232, 2018a. ISSN 1664-1078. URL https://doi.org/10.3389/fpsyg.2018.02232. [p117, 118]

S. Wallot and G. Leonardi. Deriving inferential statistics from recurrence plots: A recurrence-based test of differences between sample distributions and its comparison to the two-sample kolmogorov-smirnov test. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(8):085712, 2018b. URL https://doi.org/10.1063/1.5024915. [p113]

S. Wallot and D. Mønster. Calculation of average mutual information (ami) and false-nearest neighbors (fnn) for the estimation of embedding parameters of multidimensional time series in matlab. *Frontiers in psychology*, 9, 2018. URL https://doi.org/10.3389/fpsyg.2018.01679. [p112, 116, 117, 118]

S. Wallot, P. Mitkidis, J. J. McGraw, and A. Roepstorff. Beyond synchrony: joint action in a complex production task reveals beneficial effects of decreased interpersonal synchrony. *PloS one*, 11(12): e0168306, 2016a. [p121, 124, 125, 126]

S. Wallot, A. Roepstorff, and D. Mønster. Multidimensional recurrence quantification analysis (mdrqa) for the analysis of multidimensional time-series: A software implementation in matlab and its application to group-level data in joint action. *Frontiers in psychology*, 7:1835, 2016b. URL https://doi.org/10.3389/fpsyg.2016.01835. [p112, 117, 118]

S. Wallot, J. T. Lee, and D. G. Kelty-Stephen. Switching between reading tasks leads to phase-transitions in reading times in l1 and l2 readers. *PloS one*, 14(2):e0211502, 2019. [p112]

L. M. Ward. *Dynamical cognitive science*. MIT press, 2002. [p126]

C. L. Webber and J. P. Zbilut. Dynamical assessment of physiological systems and states using recurrence plot strategies. *Journal of Applied Physiology*, 76(2):965–973, 1994. URL https://doi.org/10.1152/jappl.1994.76.2.965. [p112]

C. Webber Jr and J. Zbilut. Recurrence quantification analysis of nonlinear dynamical systems. national science foundation, washington dc, usa. chapter 2, methods for the behavioral sciences, eds. m. riley and g. van orden, 2005. URL https://nsf.gov/pubs/2005/nsf05057/nmbs/nmbs.jsp. [p116]

M. Wijnants, F. Hasselman, R. Cox, A. Bosman, and G. Van Orden. An interaction-dominant perspective on reading fluency and dyslexia. *Annals of dyslexia*, 62(2):100–119, 2012. [p112]

J. P. Zbilut and C. L. Webber. Embeddings and delays as derived from quantification of recurrence plots. *Physics Letters A*, 171(3):199 – 203, 1992. URL https://doi.org/10.1016/0375-9601(92)90426-M. [p112, 116]

J. P. Zbilut, A. Giuliani, and C. L. Webber. Detecting deterministic signals in exceptionally noisy environments using cross-recurrence quantification. *Physics Letters A*, 246(1):122 – 128, 1998. ISSN 0375-9601. URL https://doi.org/10.1016/S0375-9601(98)00457-5/. [p112, 116]

N. Zolotova and D. Ponyavin. Phase asynchrony of the north-south sunspot activity. *Astronomy & Astrophysics*, 449(1):L1–L4, 2006. [p112]

Moreno I. Coco
School of Psychology
University of East London
E154LZ, London, UK
E-mail: M.Coco@uel.ac.uk

Dan Mønster
School of Business and Social Sciences
Aarhus University
DK-8210, Aarhus V, Denmark
ORCiD: 0000-0001-9639-8823
E-mail: danm@econ.au.dk

Giuseppe Leonardi
Institute of Psychology
University of Economics and Human Sciences in Warsaw
01-043, Warsaw, Poland
E-mail: g.leonardi@vizja.pl

Rick Dale
Department of Communication
University of California, Los Angeles
CA 90005, Los Angeles, USA
E-mail: rdale@ucla.edu

Sebastian Wallot
Department of Language and Literature
Max Planck Institute for Empirical Aesthetics
GR-60322, Frankfurt a.M., Germany
E-mail: sebastian.wallot@ae.mpg.de