BIOLOGY Methods & Protocols

Biology Methods and Protocols, 2019, 1–8

doi: 10.1093/biomethods/bpz016
Methods Manuscript

METHODS MANUSCRIPT

A comprehensive comparison and overview of R packages for calculating sample entropy

Chang Chen,[1,‡] Shixue Sun,[1,‡] Zhixin Cao,[2,3,4] Yan Shi,[5] Baoqing Sun[6] and Xiaohua Douglas Zhang [1,7,*]

[1]Faculty of Health Sciences, University of Macau, Taipa, Macau, China, [2]Department of Respiratory and Critical Care Medicine, Beijing Chao-Yang Hospital, Capital Medical University, Beijing, China, [3]Beijing Institute of Respiratory Medicine, Beijing Chao-Yang Hospital, Capital Medical University, Beijing, China, [4]Beijing Engineering Research Center of Respiratory and Critical Care Medicine, Beijing, China, [5]School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, [6]State Key Laboratory of Respiratory Disease, The 1st Affiliated Hospital of Guangzhou Medical University, Guangzhou, China and [7]Department of Biostatistics, Yale University, New Haven, CT06511, USA

*Correspondence address. Faculty of Health Sciences, University of Macau, Taipa, Macau. Tel: +853-88224813; E-mail: douglaszhang@um.edu.mo
‡These authors contributed equally

Abstract

Sample entropy is a powerful tool for analyzing the complexity and irregularity of physiology signals which may be associated with human health. Nevertheless, the sophistication of its calculation hinders its universal application. As of today, the R language provides multiple open-source packages for calculating sample entropy. All of which, however, are designed for different scenarios. Therefore, when searching for a proper package, the investigators would be confused on the parameter setting and selection of algorithms. To ease their selection, we have explored the functions of five existing R packages for calculating sample entropy and have compared their computing capability in several dimensions. We used four published datasets on respiratory and heart rate to study their input parameters, types of entropy, and program running time. In summary, *NonlinearTseries* and *CGManalyzer* can provide the analysis of sample entropy with different embedding dimensions and similarity thresholds. *CGManalyzer* is a good choice for calculating multiscale sample entropy of physiological signal because it not only shows sample entropy of all scales simultaneously but also provides various visualization plots. *MSMVSampEn* is the only package that can calculate multivariate multiscale entropies. In terms of computing time, *NonlinearTseries*, *CGManalyzer*, and *MSMVSampEn* run significantly faster than the other two packages. Moreover, we identify the issues in *MVMSampEn* package. This article provides guidelines for researchers to find a suitable R package for their analysis and applications using sample entropy.

Keywords: R package; sample entropy; time series; comparison; nonlinear dynamics

Received: 23 July 2019; **Revised:** 6 November 2019; **Editorial decision:** 8 November 2019; **Accepted:** 17 November 2019

Introduction

In 1948, Shannon founded the discipline of information theory by extending the concept of entropy, originated from statistical physics, into the process of channel communication [1]. Since then, many researchers have improved Shannon's definition of entropy in different directions. Kolmogorov proposed Kolmogorov entropy (KS entropy) [2], which calculates the change rate of Shannon entropy in a system. To simplify the calculation of KS entropy, which requires profound mathematical reasoning, Pincus [3] proposed approximate entropy as an alternative for short and noisy physiological time series. Proposed by Richman and Moorman in 2000 [4], sample entropy removes self-matching of the time series in the approximate entropy to provide more accurate entropy values. Consider a distance less than a preset threshold as a match between two segments in a dataset of continuously measured data points. Regularity may be represented by the probability that, in a dataset of continuously measured dataset, two segments with one or more data points are still matched given the two non-added segments are matched. Sample entropy is defined as the negative logarithm of this probability, thus measuring irregularity. In 2000, Costa's team proposed multiscale sample entropy (MSE) [5] to calculate sample entropy in different scales, which may show a robust pattern of entropy values as an index of complexity. Based on MSE, multivariate MSE (MMSE) considers cross-correlation of variables in the time series obtained simultaneously from multisensor magnetoencephalogram (MEG) [6].

As a distinguishing feature of human's health and disease status, sample entropy has been studied in various physiological signals, such as RR interval (a time interval where two consecutive R waves crest in the electrocardiogram)[7, 8], blood glucose [9], respiratory flow [10, 11], and MEG [6]. Studies have confirmed that different status, such as gender, age [5], exercise habits, and disease [7, 9], can lead to the differences of sample entropy. For example, the sample entropy of RR interval in healthy people is higher than that in the patients with heart disease [7], and the sample entropy of blood glucose in healthy people is higher than in the patients with type I or type II diabetes [9]. The decrease in sample entropy of certain physiological dynamic index may reflect the debility of the corresponding organ or system of the body; thus, these various concepts of entropies have been broadly employed in classification of disease severity and prediction of disease progression.

The lack of a powerful tool to calculate entropies, however, is still an obstacle faced by investigators in the field of life sciences who mostly do not have a solid mathematical background. Currently, there are very few pieces of software that can calculate entropy. MATLAB has only a paid wavelet package which can indirectly calculate Shannon entropy, log entropy, and cross entropy; these entropies are not suitable for researches of physiological dynamics due to the complexity of calculation and restriction of the algorithms. MATLAB has no packages for other entropies; therefore, investigators have to write algorithms by themselves. Although there exist several published MATLAB programs for entropy calculation on the Internet, it is difficult for beginners to customize the codes to meet their requirements. There are also some C programs on the Internet that perform sample entropy computation. However, the requirement of advanced knowledge and experience in the C language obstructs the investigators without the background of computing science. Furthermore, according to our research, there are no programs or packages for entropy calculation in SAS, SPSS, Stata, and other statistical software. Besides, Kubios [12] has designed a software, Kubios HRV, to calculate the entropy of heart rate variability in several specific file formats. However, the free version of this software can only calculate sample entropy and approximate entropy for RR interval in certain formats; multiscale entropy is available in premium version only.

R is an open-source statistical computing language with a large repository of extensible packages. For physiological signals, there are R packages designed for approximate entropy, sample entropy, multiscale entropy, and even multivariate multiscale entropy. However, no investigation was performed hitherto demonstrating their differences and weighing up their pros and cons. Therefore, in order to facilitate the application in physiological signals, this article systematically compares the performance of these R packages.

Material and methods

R packages for sample entropy

We searched for R packages containing functions of sample entropy calculation. Until now, there were totally five R packages containing functions for calculating sample entropy: *mousetrap* [13], *pracma*, *nonlinearTseries*, *MSMVSampEn* [14], and *CGManalyzer* [15].

Based on *nonlinearTserie*, another package, *RHRV*, is developed to measure heart rate variability specifically in electrocardiogram graphs [16]. Since both packages have the same principle, we will discuss them together. Reading high-dimensional and multiscale datasets, *MSMVSampEn* calculates multivariate multiscale entropies for time series with multiple variables. *CGManalyzer* focuses on glucose data and provides a multiscale entropy algorithm.

Definition of sample entropy and its extensions

Sample entropy

The description of sample entropy and related terms is based on the following notation. Let m be an embedding dimension which shows the length of two segments in a sequence to be compared, k be time lag showing the effect of long-range autocorrelation to a sequence, r be similarity threshold showing the tolerance for accepting similar patterns between two segments. Sample entropy is calculated in the following steps below:

1. for a time series of length N: $\{X_i\} = \{x_1, \ldots, x_i, \ldots, x_N\}$;
2. define m-dimensional delay vectors $u_m(i) = \{x_i, x_{i+k}, x_{i+2k} \ldots, x_{i+(m-1)k}\}$, $1 \le i \le N-(m-1)k$, $1 \le k \le N-1$;
3. calculate the Euclidean distance (d) between $u_m(i)$ and $u_m(j)$: $(d\left[u_m(i), u_m(j)\right])$. For the defined parameter r, $n_i^m(r)$ represents the number of $d\left[u_m(i), u_m(j)\right]$ which is no larger than r ($d\left[u_m(i), u_m(j)\right] \le r$). $C_i^m(r)$ is defined as the ratio of $n_i^m(r)$ to the whole distances: $C_i^m(r) = n_i^m(r)/(N-m+1)$;
4. average $C^m(r)$, we get $C^m(r) = 1/(N-m+1)\sum_{i=1}^{(N-m+1)} C_i^m(r)$, which denotes the probability that the distance between any two vectors is no greater than r;
5. increase the dimension from m to $m+1$, then repeat Steps 2–4 to get $C^{m+1}(r)$; and
6. sample entropy is calculated as:

$$\text{Sample entropy}(m, r) = -\ln\left[\frac{C^{m+1}(r)}{C^m(r)}\right].$$

MSE

To quantify the system dynamics of a time series on different scales, MSE computes sample entropy of coarse-grained time series.

For a monovarietal discrete time series $\{X_i\} = \{x_1, \ldots, x_i, \ldots, x_N\}$, the coarse-grained time series $y^{(\tau)}$ is defined as:

$$y_j^{(\tau)} = \frac{1}{\tau} \sum_{i=(j-1)\tau+1}^{j\tau} x_i, \quad 1 \le j \le N/\tau,$$

where N/τ is the length of the coarse-grained time series $\{y^{(\tau)}\}$. When the scale is 1 ($y^{(1)} = \{X_i\}$), the coarse-grained time series equals the original series.

Then the sample entropy of time series $y^{(\tau)}$, which is the MSE of X_i, is calculated. τ as time scale shows the degree of granularity for each sequence.

MMSE

MSE is a univariate method which has been used to measure complexity of single channel physiological signals. Recent developments in sensor technology have achieved routine recording of multivariate time series from physical systems simultaneously, such as electroencephalogram recorder. In contrast to MSE which only considers each data channel separately, multivariate sample entropy considers possible associations among multivariate time series and obtains a comprehensive entropy value [6]. Its calculation requires the following two steps.

Step 1: As in MSE calculation, for a q-dimensional time series, calculate the coarse-grained q-dimensional time series $y_q^{(\tau)}$ where τ as time scale shows the degree of granularity of each sequence.

Step 2: Use the q-dimensional coarse-grained time series to calculate multivariate sample entropy, the result is the MMSE of q-dimensional time series in scale τ.

For each coarse-grained time series, the algorithm of calculating multivariate sample entropy is as follows.

1. For a q-dimensional time series of length N: $\{X_i\} = \{\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N\}$, where
$$\mathbf{x}_i = \{x_i(1), \ldots, x_i(j), \ldots, x_i(q)\}$$
2. Define $n = \max(\mathbf{M}) \times \max(\mathbf{K})$. $\mathbf{M}$ is the embedding vector and $\mathbf{K}$ is the time lag vector: $\mathbf{M} = \{m_1, \ldots, m_j, \ldots, m_q\}$, $\mathbf{K} = \{k_1, \ldots, k_j, \ldots, k_q\}$
3. Construct $(N - n)$ composite delay vectors $X_q(i) \in \mathbb{R}^q$.
4. Calculate the maximum distance (d) between $X_q(i)$ and $X_q(j)$, $(d[X_q(i), X_q(j)] = \max_{l=1,\ldots,q}\{|x(1 + k_l * i + l - 1) - x(1 + k_l * j + l - 1)|\}$. For the defined parameter r, let $n_i(r)$ represent the quantity of maximum d no greater than r ($d[X_q(i), X_q(j)] \le r$). The ratio of number $n_i(r)$ to the whole number is calculated as $B_i^M(r) = n_i(r)/(N - n + 1)$.
5. Average $B^M(r)$, we get $B^M(r) = 1/(N - n + 1) \sum_{i=1}^{(N-n+1)} B_i^M(r)$, which denotes the probability that the distance between any two vectors is no greater than r.
6. Extend the dimension of multivariate delay vector from $\mathbf{M}$ dimension to ($\mathbf{M}$+1) dimension and calculate the $B_i^{M+1}(r)$.

The multivariate sample entropy is calculated as:

$$\text{MSample Entropy}(M, K, N, r) = -\ln[B^{M+1}(r)/B^M(r)]$$

Physiological datasets

To test these programs, we used four online datasets, the features of which are shown in Table 1.

Datasets 1 and 2 are air flow records obtained from a published clinical trial [17]. Dataset 1 was collected from one patient by overnight respiratory monitoring devices at the frequencies of 1 Hz and Dataset 2 was collected from another patient at 5 Hz. Datasets 3 and 4 are downloaded from Physionet (https://physionet.org/). Dataset 3, downloaded from the Fantasia Database (https://physionet.org/physiobank/database/fantasia/) [18, 19], contains RR interval recordings from 10 young people and 10 elderly people. Dataset 4 is the first record of the QT Database (https://physionet.org/physiobank/database/qtdb/) [19, 20]. All four datasets and example R codes have been uploaded to the website https://quantitativelab.fhs.um.edu.mo/analytic-tool/. Readers can refer them to complete the sample entropy calculation quickly.

Table 1: General information of testing datasets

Dataset no	Data type	Data point number of Datasets	Data content
Dataset 1	Air flow	196 533	24-hours recording of one subject
Dataset 2	Air flow	44 419	24-hours recording of one subject
Dataset 3	RR interval	Average 6000	20 subjects
Dataset 4	RR interval	1134	One subject

Results

Our introduction and comparison are based on the latest version of the five R packages. The package details are in Table 2.

Introduction of the programs

In *mousetrap*, the computing program for sample entropy is *Mt_sample_entropy*. With the purpose of analyzing the data of computer mouse-tracking experiments, this package processes not only physical signals but also trajectory data. Therefore, researchers need to adjust their data format to comply with the requirement. In this program, adjustable parameters include embedding dimension and similarity threshold.

Pracma has two functions for computing entropy: *sample_entropy* for calculating sample entropy and *approx_entropy* for approximate entropy. This package is designed for numerical analysis and calculation of linear equations. Users can adjust embedding dimension, similarity threshold, and time lag.

In *nonlinearTseries*, a package focuses on nonlinear analysis of time series, the function *sampleEntropy* calculates sample entropy. Before calculating entropy with this package, correlation dimension should be precomputed, then sample entropy can be generated by dividing correlation sums in different embedding dimensions. By setting a series of embedding dimensions and similarity thresholds, *sampleEntropy* can simultaneously display entropy values under different conditions. By modifying the parameters associated with the amount of computation, users can shorten the operating time.

MSMVSampEn is the computing program designed to compute MMSE. This package is built on GitHub, so *Devtools* is required before its installation. Under particular parameter settings, it can calculate multivariate sample entropy, MSE, or multivariate MSE. The drawbacks, however, are that it can only output one type of entropy in each run and that users need to manually iterate each scale when calculating multiscale entropy. Adjustable parameters are embedding dimension, similarity threshold, time lag, and time scale. Both embedding dimension and time lag are expressed as vector. In the calculation of MMSE, all variables of one time series are generally

Table 2: Comparison of five R programs

Package	*mousetrap*	*pracma*	*nonlinearTseries*	*MVMSampEn*	*CGManalyzer*
Download web address	https://cran.r-project.org/web/packages/mousetrap/index.html	https://cran.r-project.org/web/packages/pracma/index.html	https://cran.r-project.org/web/packages/nonlinearTseries/index.html	https://github.com/areshenk/MSMVSampEn	https://cran.r-project.org/web/packages/CGManalyzer/index.html
Latest version	Version 3.1.3,	Version 2.2.5, 9 April 2019	Version 0.2.6, 21 February 2019	No exact version	Version 1.2
Updated time	4 October 2019			17 July 2017	23 October 2019
Core function	*mt_sample_entropy*	*sample_entropy*	*sampleEntropy*	*MSMVSampEn*	*MSEbyC.fn*
Type of entropy	Sample entropy	Sample entropy, Approximate entropy	Sample entropy	MMSE	MSE
Types of input data	Mouse movement trajectory	One-dimensional time series	One-dimensional time series	High-dimensional time series	One-dimensional time series
Embedding dimension	Single, modifiable	Single, modifiable	Multiple, modifiable	Single, modifiable	Multiple, modifiable
Time lag	Unchangeable (Time lag = 1)	Modifiable	Modifiable	Modifiable	Unchangeable (Time lag =1)
Similarity threshold	Single, require multiplying standard deviation	Single, require multiplying standard deviation	Multiple, require multiplying standard deviation	Single, require multiplying standard deviation	Multiple
Estimated value	No	No	Yes	No	No
Multiscale value	No	No	No	Yes	Yes
Output value	Single value	Single value	Multiple value	Single value	Multiscale value
Figures for displaying data/results	No	No	Yes	No	Yes
Required package/work	No	No	Require correlation dimension	Require *Devtools* package	No

set with the same embedding dimension and time lag. When calculating the entropy of a single variable, the vector dimension can be reduced to one dimension. When the time scale is 1, the result is sample entropy. MMSE was originally applied to MEG data. Since we did not find datasets in this type, we only analyzed univariate time series in this article.

MSEbyC.fn, the function for calculating MSE in *CGManalyzer*, is originally designed by one of the authors of the current paper, Xiaohua Douglas Zhang, to calculate MSE of glucose data generated by several continuously glucose monitoring (CGM) devices. *MSEbyC.fn* calculates sample entropy using C language, which significantly shortens the operating time, especially for huge dataset. Required parameters are embedding dimension, similarity threshold, time lag and time scale. One advantage of this package is that it can calculate and display multiple sample entropies on different scales (the default scale is from 1 to 10) at a time. Another advantage is its ability of handling missing values and checking/controlling data quality, which generates more reliable results of sample entropy.

Summary of feature

Table 2 shows the information of five packages and the key features of these programs for sample entropy calculation. For input data, *mousetrap* and *CGManalyzer* require a specific data format, while the other three programs accept one-dimensional time series data. Besides, *MVMSampEn* can also process high-dimensional time series. The adjustable parameters differ among these programs. For embedding dimension, *nonlinearTseries* and *CGManalyzer* can handle multiple embedding dimensions simultaneously, while the other three can only calculate entropy on single embedding dimension. The embedding dimension of all packages default to 2, under which the sample entropy has been proven to be able to retain enough information from time series and possess effective statistical properties [3]. For the time lag, it is fixed to 1 in *mousetrap* and *CGManalyzer*, and is modifiable in the other three packages, in which the default time lag is set to 1. For similarity threshold, *nonlinearTseries* and *CGManalyzer* can calculate the entropy under multiple similarity thresholds, while the other three can only calculate the entropy of a single similarity threshold. Extra supportive tools need to be pre-installed for *MVMSampEn* package: it calls functions in *Devtools* before installing. Besides, *nonlinearTseries* requires manual calculation of correlation dimension before computing sample entropy. For the outputs, *mousetrap*, *pracma*, and *MVMSampEn* give only one single entropy value, while *NonlinearTseries* produces entropy values on multiple embedding dimensions and multiple similarity thresholds, and *CGManalyzer* yields a series of multiscale entropy values. Moreover, *NonlinearTseries* and *CGManalyzer* also embed functions to draw graphs for further analysis.

Application in physiological datasets

Table 3 shows the calculated entropy values and operating time of these five programs on the four datasets. The related result of dataset 3 is obtained using the data in the first subject of the study. Pincus [3] and Richman [4] have proven that when the parameter embedding dimension $m = 2$, similarity threshold r is between $0.1 \times$ SD (SD represents the standard deviation of each sequence) and $0.25 \times$ SD and time lag $k = 1$, the sample entropy can retain enough information from time series and have effective statistical properties. Hence the parameters in our calculation were set as: embedding dimension $m = 2$, similarity threshold $r = 0.2 \times$ SD, and time lag $k = 1$. All computations were conducted in the same HP desktop computer: Elite Desk 800 G2 TWR, with Intel(R) Core (TM) i7-6700 CPU @ 3.40 GHz, 16.0 GB

Table 3: Sample entropy value and operating time for four datasets

Dataset	Dataset 1 (191 415 points)		Dataset 2 (44 419 points)		Dataset 3 series 1 (6823 points)		Dataset 4 (1134 points)	
Package	Value	Time	Value	Time	Value	Time	Value	Time
Mousetrap with *diff*	0.2187467	6 h 5 m 58 s	1.083679	19 m 40 s	0.8499059	31 s	1.315703	1 s
Mousetrap without *diff*	0.1780288	6 h 19 m 16 s	1.286609	18 m 45 s	0.9012826	29 s	1.530295	1 s
pracma	0.1780246	11 h 20 m 7 s	1.286609	34 m 58 s	0.9012826	49 s	1.530295	2 s
nonlinearTseries	0.1780273	1 m 14 s	1.287526	1 s	0.9039996	1 s	1.535612	1 s
MSMVSampEn	NaN	NA	1.28662	2 m 11 s	0.9013839	3 s	1.530749	1 s
CGManalyzer	0.178	2 m 2 s	1.287	6 s	0.901	1 s	1.530	1 s

Table 4: The percentage error in the sample entropy values compared with the *nonlinearTseries* package and other packages.

R package name compared with *nonlinerTseries*	40 000	20 000	5000	2000	500
Mousetrap without *diff*	0.02% ± 0.01%	0.03% ± 0.02%	0.19% ± 0.12%	0.42% ± 0.30%	2.33% ± 1.09%
pracma	0.02% ± 0.01%	0.05% ± 0.02%	0.19% ± 0.12%	0.42% ± 0.30%	2.33% ± 1.10%
nonlinearTseries					
MSMVSampEn	0.04% ± 0.03%	0.10% ± 0.06%	0.22% ± 0.20%	0.46% ± 0.20%	2.28% ± 2.40%
CGManalyzer	0.05% ± 0.01%	0.04% ± 0.03%	0.21% ± 0.18%	0.49% ± 0.32%	2.24% ± 1.15%

The percentage error is expressed as deviations from the values returned by the other packages and computed for datasets of different size. Values are given as mean ± standard deviation.

installed memory, and 64-bit Operating System. The development environment was R-studio Open Source Edition, based on R version 3.3.4.

For each physiological dataset, *mousetrap* returned a result that deviated considerably from the other four programs (shown as "*mousetrap* with *diff*" in Table 3). By reviewing its documentation, we found that this deviation is caused by an automatic computation of the first-order differences before calculating sample entropy when the parameter "use_diff" is set to "TRUE" (the default value). This setting is primarily desired when analyzing computer mouse trajectories. *Mt_sample_entropy* yielded consistent values (shown as '*mousetrap* without *diff*' in Table 3) after setting the parameter "use_diff" to "FALSE" (using the recently updated version of the package). When the data volume is small, the values from *nonlinearTseries* have a slight increase (about 3%) from the mean, and the deviation decreases as data volume increases. According to our four datasets, the difference was negligible when computing 40 000+ data points. We believed this is because the program approximates to sample entropy by calculating the ratio of correlation sum, which generates noticeably different results from that of other algorithms when data volume is less than 1000. In order to verify the robustness of this conclusion, we designed the following experiment: we created 5 subsets with sequence lengths of 40 000, 20 000, 5000, 2000, and 500, respectively, each subset containing 10 sequences intercepted from Dataset 1 randomly. We then calculated the mean and standard deviation of percentage errors between the sample entropy values of *NonlinearTseries* and that of other packages. The result is shown in Table 4. We found that for small datasets, the percentage error in the sample entropy values computed by the *nonlinearTseries* package is large, about 2–3% deviations from the values computed by the other packages. This percentage error decreases as data length increases. This observation is consistent with what we described in the previous sentence.

MSMVSampEn reported an error when computing Dataset 1: *MSMVSampEn* returned an error message "In log(p1): NaNs produced" and the program terminated. Repeating tests with datasets of different volumes showed that the package failed to work when handling more than 120 000 data points. The developer of *MSMVSampEn* attributes this to the limit of R's ability to process huge number of similarities.

Besides direct computation of the entropy values, *CGManalyzer* provides a function handling missing values to get more robust results. After the data are adjusted, it plots both the corrected time series and the original ones, allowing researchers to inspect their distributions and to ensure data quality by potentially excluding outliers. Normally, the researcher drops the missing values and connects the non-missing values together to form a new sequence for analysis because sample entropy calculation programs generally take only non-missing data as input. In *CGManalyzer* package, four methods for missing value processing are provided, one for directly removing missing values, one for linear interpolation, one for local polynomial regression fitting, and the last one estimates missing value by other periodic isomorphic data based on data period characteristics. This function simplifies the preprocessing of sequence and effectively increases the efficiency of analysis. For more details, users can read the "help" file of the *CGManalyzer* package. In our guideline, we used the method of linear interpolation to deal with the missing value of first 500 points in Dataset 2 for demonstration. It should be noted that the linear interpolation method does not work effectively in some situations such as in Dataset 2. People who are interested in handling missing values in the calculation of sample entropy may refer to our recent publication in Dong *et al.* [21]. Figure 1 shows the original time series (top) and corrected sequence (bottom). The red horizontal line is the mean of the sequence. These two figures clearly show the proportion, location of missing values, and sequence after the processing of missing values.

The execution time of the five programs when calculating a small dataset, such as Dataset 4, is similarly short. When calculating 10 000+ data points (such as Datasets 1 and 2), *pracma* and *mousetrap* take much longer time, from quarters to hours, than the other three packages which take <2 min. This is because these three programs pass the most time-consuming

tasks to embedded C or C++ that has a much higher computing speed than R. In order to investigate the stability of the results for the operating times, we used the five different length subsets built with Dataset 1 from the previous experiment and calculated the mean and standard deviation of operating time separately. The result is shown in Table 5. We found that for different sequences of the same length, the operation duration for each program to calculate sample entropy was close, which means the operating time is stable.

Only two packages, *nonlinearTseries* and *CGManalyzer*, can generate figures displaying calculated entropy values. *nonlinearTseries* plots a figure for each time series, while *CGManalyzer* plots entropy results of a set of time series into one figure. *NonlinearTseries* also provides a curve of similarity threshold and sample entropy values at different embedding dimensions (Fig. 2a) and a linear fit graph for identification of optimal entropy value (Fig. 2b). Figure 2 (a) shows the trend of sample entropy with similarity threshold. The investigator can obtain the appropriate sample entropy value for analysis based on selecting the appropriate similarity threshold (or similarity threshold interval) from the trend graph. Figure 2 (b) shows the sample entropy value after a linear fit for a similarity threshold interval. Furthermore, both graphs can show the effect of different embedding dimensions on the sample entropy value. Users can choose the appropriate embedding dimension according to their needs. Otherwise, researchers can directly select the embedding dimension = 2 according to the reference [3]. *CGManalyzer* draws two more graphs. One is a trend curve of sample entropy changes and time scale with error bars for different groups (Fig. 3a), and the other is an antenna plot which plots strictly standard mean difference (SSMD) against the mean difference and its confidence interval between a pair of different groups (Fig. 3b) [14]. The SSMD shows the degree of differentiation of sample entropy in the two groups [22–24]. Figure 3 shows that the MSE of heart rate variability for young people is higher than that of the elderly. When scale = 3, the difference between the two is about 0.2 and the confidence interval of mean difference is below 0, so the mean difference is significant. This example verifies that (multiscale)

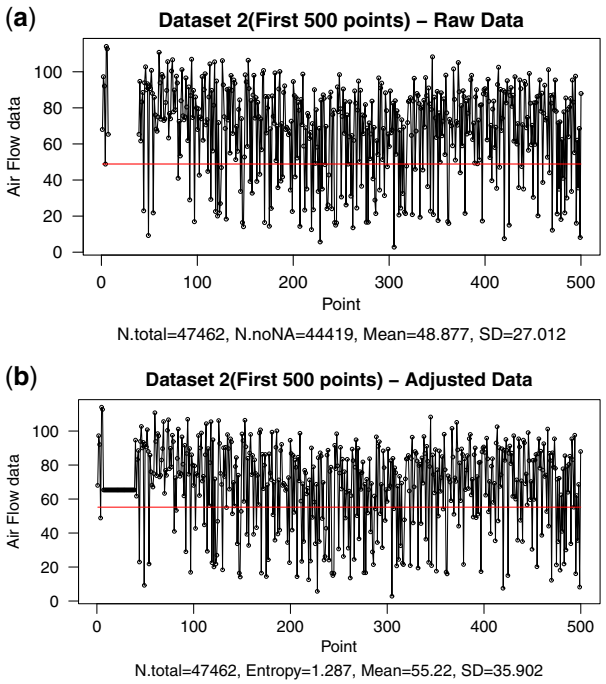

Figure 1: Adjustment of missing value using *CGManalyzer*. (**a**) First 500 data points from original Dataset 2. (**b**) Adjusted data after processing missing value by *CGManalyzer*. The red line indicates the mean value of the data.

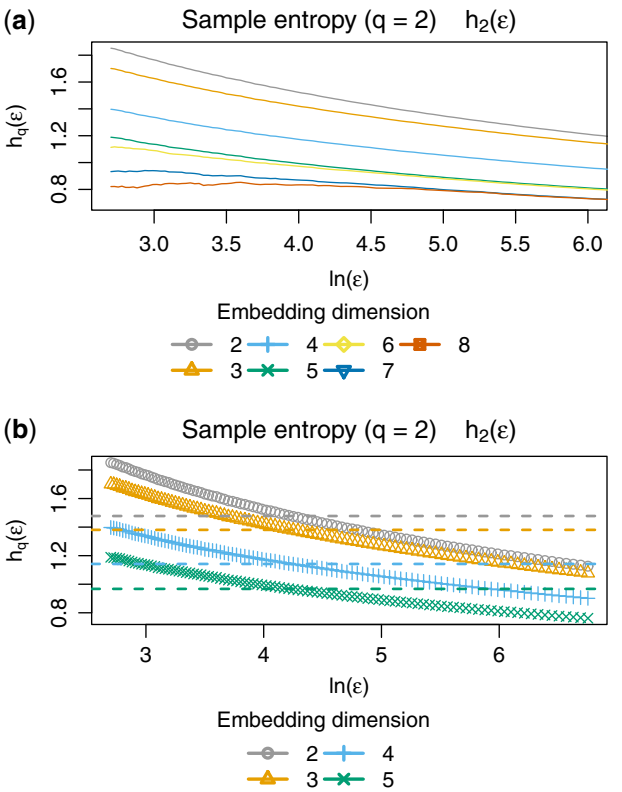

Figure 2: Graphs from *NonlinearTseries*. (**a**) Curve of sample entropy values changing with similarity threshold under different embedding dimensions for Dataset 2 (44419 points). (**b**) Linear fitting results for the curve of each embedding dimension. Users can select a specific similarity threshold interval for a linear fit and then calculate the mean of these linear fitted values as an approximation of the entropy values.

Table 5: The operating time (unit: second) of five packages for different lengths of data

R package name	40 000	20 000	5000	2000	500
Mousetrap without *diff*	1205.6 ± 428.73	278.4 ± 10.85	25.0 ± 3.37	19.6 ± 1.29	3.1 ± 0.32
Pracma	2034.0 ± 507.65	440.2 ± 5.16	27.1 ± 0.31	4.4 ± 0.52	0.3 ± 0.48
nonlinearTseries	7.2 ± 0.42	1.5 ± 0.52	0.1 ± 0.31	0.0 ± 0.0	0.0 ± 0.0
MSMVSampEn	104.8 ± 1.98	26.4 ± 0.70	1.7 ± 0.48	0.5 ± 0.53	0.0 ± 0.0
CGManalyzer	7.0 ± 1.05	2.0 ± 0	0.3 ± 0.48	0.3 ± 0.48	0.2 ± 0.42

Values are given as means ± standard deviation.

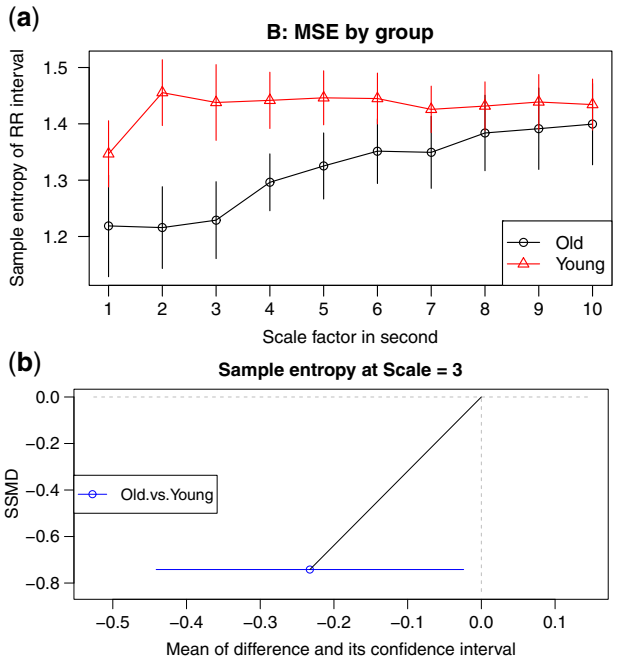

Figure 3: Display of calculated sample entropy using *CGManalyzer*. (**a**): Plot showing the mean and standard deviation of sample entropy, respectively, in two groups in Dataset 3 of RR interval data by different groups. (**b**): An antenna plot showing strictly SSMD against the mean difference and its confidence interval between old and young people.

sample entropy is significantly different between young people and elderly.

Discussion

Our comparison of the five R packages of sample entropy explores parameter configurations and output displays, covering almost all considerations of calculating sample entropy. Based on the needs, these programs can be adjusted to calculate various forms of sample entropy including univariate sample entropy, MSE, MMSE, etc., with several visualization plots. The major pros and cons for these five packages are summarized in the following paragraphs.

Compared with other packages, *mousetrap* [13] and *pracma* are easier to operate but with longer calculating time. While *pracma* calculates sample entropy using the untransformed values of the time series, *mousetrap* can compute sample entropy either using the untransformed values or the first-order differential (depending on the setting of the "use_diff" parameter). In addition, *pracma* can calculate the entropy of different time lags. C language is embedded into three packages, *NonlinearTseries*, *MVMSampEn* [14], and *CGManalyzer* [15], which calculate sample entropy significantly faster than *mousetrap* and *pracma*. *NonlinearTseries* and *MVMSampEn* can quickly calculate sample entropy and MSE for one parameter setting (embedding dimensions, similarity thresholds, or different time scales) each time. The *CGManalyzer* is developed for blood glucose data collected using continuous monitoring devices [9]; it can also process other kinds of signals when they are converted into the required format. The package documentation includes an introduction on necessary steps before running, such as format preparation of data and creation of working folders. *CGManalyzer* also provides data preprocessing functions for the adjustment of missing values and quality control. To visualize the final results, *CGManalyzer* generates a line graph with error bars, which is suitable for the analysis among groups with different features, and *nonlinearTseries* draws curves of sample entropy values with similarity thresholds to facilitate the estimation of exact sample entropy.

One disadvantage of *MVMSampEn* and *CGManalyzer* in their previous versions is that they cannot calculate time series with more than 120 000 data points; however, this limitation can be overcome by a minor revision in their corresponding C codes. We have not only discovered this issue but also find the way to solve this issue in *CGManalyzer*. After our finding of this limitation of *CGManalyzer*, the authors of *CGManalyzer* have now implemented this modification in the updated version in CRAN and https://quantitativelab.fhs.um.edu.mo/analytic-tool/. Readers can use the latest version to do the analysis. When *MVMSampEn* is applied to time series data with more than 120 000 points, researchers should make this modification in *MVMSampEn* accordingly.

In conclusion, *CGManalyzer* is more suitable for calculating MSE of physiological signals, especially blood glucose. It produces entropy values in different embedding dimensions and similarity thresholds at a time, and plots entropy values into line graphs with error bars for different groups of data. *MSMVSampEn* is the most suitable and the only one package for MMSE calculation. It can also compute univariate sample entropies and multiscale entropy after simple modification on the parameters. One disadvantage of *MVMSampEn* in their current versions is that they cannot calculate time series with more than 120 000 data points. Similar to *CGManalyzer*, *NonlinearTseries* also calculates entropy values in different embedding dimensions and similarity thresholds at a time. Its results, however, deviate slightly when the data length is short. The above three packages run fast when calculating entropy and the results can be displayed using figures according to research needs. Designed to analyze computer mouse-tracking data, *mousetrap* by default focuses on sample entropy for the first-order difference of the time series. In the recently updated version of the package, this behavior can also be changed, which is especially important when analyzing general physiological signals. To change this behavior, users can set the "use_diff" parameter to "FALSE". Users need to pay attention to the parameter selection. *Pracma* has simple commands to calculate both sample entropy and approximate entropy. These two packages take very long time for large dataset, so we do not recommend them for physiological signals with more than 10 000 data points. On the other hand, the original codes of these two programs are very easy for beginners to learn the calculation of entropy.

Our research in this article will provide a guideline for researchers in different areas with modest mathematics background on the choice of the most appropriate R program to study sample entropy, MSE, and MMSE in various signal types, as well as a suggestion on new R program development.

Acknowledgement

The authors would like to thank Lei Kuan Cheok for his diligent proofreading of this article.

Funding

This work was funded by University of Macau (grant numbers: FHS-CRDA-029-002-2017, EF005/FHS-ZXH/2018/GSTIC and MYRG2018-00071-FHS) and by The Science and

Technology Development Fund, Macau SAR (File no. 0004/2019/AFJ).

Conflict of interest statement. The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Shannon CE, Weaver W. The mathematical theory of information. *Mathematical Gazette* 1949;**97**:170–80.
2. Kolmogorov AN. A new metric invariant of transient dynamical systems and automorphisms in Lebesgue spaces. *Dokl Akad Nauk SSSR* 1958;**951**:861–4.
3. Pincus SM. Approximate entropy as a measure of system complexity. *Proc Natl Acad Sci USA* 1991;**88**:2297–301.
4. Richman JS, Moorman JR. Physiological time-series analysis using approximate entropy and sample entropy. *Am J Physiol Heart Circ Physiol* 2000;**278**:H2039–2049.
5. Goldberger AL, Amaral LAN, Hausdorff JM *et al*. Fractal dynamics in physiology: alterations with disease and aging. *Proc Natl Acad Sci USA* 2002;**99**:2466–72.
6. Ahmed MU, Mandic DP. Multivariate multiscale entropy analysis. *IEEE Signal Process Lett* 2012;**19**:91–4.
7. Chen C, Jin Y, Lo IL *et al*. Complexity change in cardiovascular disease. *Int J Biol Sci* 2017;**13**:1320–8.
8. Steinisch M, Torke PR, Haueisen J *et al*. Early detection of coronary artery disease in patients studied with magnetocardiography: an automatic classification system based on signal entropy. *Comput Biol Med* 2013;**43**:144.
9. Zhang XD, Pechter D, Yang L *et al*. Decreased complexity of glucose dynamics preceding the onset of diabetes in mice and rats. *PLos One* 2017;**12**:e0182810.
10. Jin Y, Chen C, Cao Z *et al*. Entropy change of biological dynamics in COPD. *Int J Chron Obstr Pulm Dis* 2017;**12**:2997.
11. Sun S, Jin Y, Chen C *et al*. Entropy change of biological dynamics in asthmatic patients and its diagnostic value in individualized treatment: a systematic review. *Entropy* 2018;**20**:402.
12. Tarvainen MP, Niskanen JP, Lipponen JA *et al*. Kubios HRV–heart rate variability analysis software. *Comput Methods Programs Biomed* 2014;**113**:210.
13. Kieslich PJ, Henninger F, Wulff DU *et al*. Mouse-tracking: a practical guide to implementation and analysis. In: Schulte-Mecklenbeck M, Kühberger A, and Johnson JG (eds), *A handbook of process tracing methods*, New York: Routledge, 2019, 111–30.
14. MSMVSampEn: Multiscale multivariate sample entropy in, R. http://areshenk-research-notes.com/sampen_in_r/ (8 November 2019, date last accessed).
15. Zhang XD, Zhang Z, Wang D. CGManalyzer: an R package for analyzing continuous glucose monitoring studies. *Bioinformatics* 2018;**34**:1609–11.
16. Rodríguez-Liñares L, Vila X, Méndez AJ et al. R-HRV: an R-based software package for heart rate variability analysis of ECG recordings. in Proceedings of the 3rd Iberian Conference on Information Systems and Technologies, Ourense, Spain. 2008. 565–74. New York, USA: IEEE.
17. Cao Z, Luo Z, Hou A *et al*. Volume-targeted versus pressure-limited noninvasive ventilation in subjects with acute hypercapnic respiratory failure: a multicenter randomized controlled trial. *Respiratory Care* 2016;**61**:1440.
18. Iyengar N, Peng C-K, Morin R *et al*. Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics. *Am J Physiol* 1996;**271**:1078–84.
19. Goldberger AL, Amaral LAN, Glass L *et al*. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation* 2003;**101**:e215–e220.
20. Laguna P, Mark RG, Goldberger AL *et al*. A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG. *Comput Cardiol* 1997;**24**:673–6.
21. Dong X, Chen C, Geng Q *et al*. An improved method of handling missing values in the analysis of sample entropy for continuous monitoring of physiological signals. *Entropy* 2019;**21**:274.
22. Zhang XD. A pair of new statistical parameters for quality control in RNA interference high throughput screening assays. *Genomics* 2007;**39**:552–61.
23. Zhang XD. *Optimal High-Throughput Screening: Practical Experimental Design and Data Analysis for Genome-Scale RNAi Research*. Cambridge, UK: Cambridge University Press, 2011.
24. Zhang XD, Ferrer M, Espeseth AS *et al*. The use of strictly standardized mean difference for hit selection in primary RNA interference high throughput screening experiments. *J Biomolecular Screening* 2007;**12**:497–509.