GUI for the analysis of complexity and entropy in physiological signals

User Manual

David Mayor, Deepak Panday,
Harikala Kandel

2021

David Mayor is a visiting Fellow at the University of Hertfordshire (UK).
He may be reached at: davidmayor@welwynacupuncture.co.uk.
http://electroacupuncture.qeeg.co.uk/

For technical enquiries, please contact the software developer,
Deepak Panday: aranp2010@googlemail.com

CONTENTS

INTRODUCTION

CEPS was developed for the analysis of Complexity and Entropy in Physiological Signals, although in principle it could be applied to other continuous and discrete time series data and even to non-time series data. Some of the measures included would also be appropriate for very different data types, such as questionnaire responses. Data is accessed and analysis conducted via a graphical user interface (GUI), making it relatively user-friendly for those with basic computer skills but no programming experience. We have also tried as far as possible to write this Manual in non-technical language, so that CEPS might be more widely accessible to potentially interested but busy clinical researchers. However, MATLAB ® currently needs to be installed on your PC for the GUI to run.[1]

CEPS is based on an initial idea by David Mayor (DM), who, as a longterm acupuncture researcher with little computer science background, felt frustrated by the lack of suitable software to fulfil his research dreams. The GUI was created by Deepak Panday (DP) using Matlab (version R2019a), assisted by Harikala Kandel (HK). DM and HK wrote the manual, assisted by DP. HK and DP tested the software.

[1] For those at an academic institution, this should pose no problems. For others, a licence for student or personal use is relatively inexpensive.

We recognize that other open-access MATLAB-based GUI toolboxes exist for time series analysis. While CEPS does include some general (time series) data analysis tools, here we have focused on those complexity and entropy measures that appear most frequently in the literature, together with some recently introduced entropy measures which may have advantages over those that are now more established.

The list of functions we have initially provided may be useful for those engaged in heart rate variability (HRV), electroencephalography (EEG), respiration and other research using univariate physiological measures, and could be expanded in future versions as required.

The first functions included are for basic data analysis: descriptive statistics, some simple linear measures and tests for normality of distribution. Then follow some time- and frequency-domain measures, and a selection of methods for assessing stationarity and nonlinearity of data. Nine methods of estimating data complexity are included, and some 30 entropy measures. The GUI also includes a section for data pre-processing and ancillary methods to enable parameter estimation of embedding dimension m and time delay τ ('tau') where these are required. A 'test and plot' facility is provided to assist in parameter selection and multiscale analysis. A section on classification of results will eventually complete the toolbox. The measures available are listed in **Table 1** below. Those measures not yet fully implemented are shown in the Table on a grey background.

For further background information, see our article on CEPS, published in *Entropy*.

This is the first version of CEPS to be released, and we are aware that it is not perfect. Importantly, not all the codes used have been validated (see the article in *Entropy* for details). Any feedback is welcome, and can be provided using the feedback form accessible via the combo box (drop-down menu) under 'HELP' in the GUI or on the BitBucket site.

SYSTEM REQUIREMENTS

CEPS was developed initially using MATLAB2019a and then App Designer. A standalone version is planned, but currently MATLAB will need to be installed on your computer or network to use CEPS. A Windows or Linux 64-bit operating system is required, and for fast plotting and calculation, 8 GB of RAM and a screen resolution of 1400 × 900 pixels are recommended. As for other MATLAB-based GUIs developed using App Designer, for the current MATLAB-based installation, best results will be obtained with MATLAB R2018a or later. If you are running an earlier version of MATLAB, MATLAB should automatically compensate for this.[2]

INSTALLATION

This 64-bit Windows/Linux version of CEPS is publicly available for free, non-commercial use at https://bitbucket.org/deepak_panday/pipeline/src/pipeline_V2/. No subscription is required. The complete download will take 800 MB of disk space (this does not include the space required for MATLAB itself). This Manual and a Primer on complexity and entropy are included in the download as *.pdf files. MATLAB must be installed before use. To run CEPS, first close MATLAB, then go to the downloaded 'scripts' folder and click on the file CEPS.mlapp.[3] If you are running a version of MATLAB earlier than R2018a, you should click on the appropriate ProcessData_App file (i.e. CEPS_17a.mlapp for R2017a, and so forth).[2] MATLAB will then open, followed by CEPS. The CEPS Manual and Primer are accessible via the combo box under 'HELP' in the GUI, as are a feedback form and a link to the latest version of the software. Feedback may also be provided using the BitBucket 'Give Feedback' option. This is visible on the above website when you use the right bracket – ']' – key on your keyboard.

---

[2] See 'Compatibility Between Different Releases of App Designer'. *Mathworks Help Center*. (https://uk.mathworks.com/help/matlab/creating_guis/compatibility-between-different-releases-of-app-designer.html) [Retrieved December 27 2020].

[3] It is advisable not to rename this file, and not to attempt to open it via the 'Current Folder' pane in MATLAB or to open MATLAB first and *then* click on the file CEPS.mlapp in the scripts folder.

LOADING DATA

When CEPS first opens, ‘Help’, ‘Data Type’ and radio buttons for selecting whether your data is ‘Time Series’ or ‘Non Time Series’ are visible in the top left corner of the GUI. Below these are two text boxes in which to enter Epoch Length (if your data is subdivided into epochs, or you wish to analyse the data in epochs) and Sampling Rate. See [6] below for further explanation.

Below these two boxes is a drop-down list of six Application Modes. The six modes, which should be used in order, are: Load Data, Pre-Process Data, Test Parameters, Run Pipeline, Process Results and Classification (this latter is not yet implemented). When any of these are selected, three main panes are present on the right side of the GUI, one above the other (Load Data I, II and III, etc.). Between the upper and middle panes there is a row of combo boxes where you can select data File/s, Epoch/s within a file, Variables (‘columns’) within a file (if multivariate or containing multiple columns) or Scale (if you require a multiscale output). These boxes will not all be active in all Application Modes.

To proceed, the data you wish to analyse should be ready for processing and stored in a single location. Your data should be univariate but can be in single or multiple column format, with *.txt, *.csv, *.mat or *.xlsx file extensions. CEPS does not perform multivariate analysis.

Loading data requires the six following steps, shown in **Figure 1**.

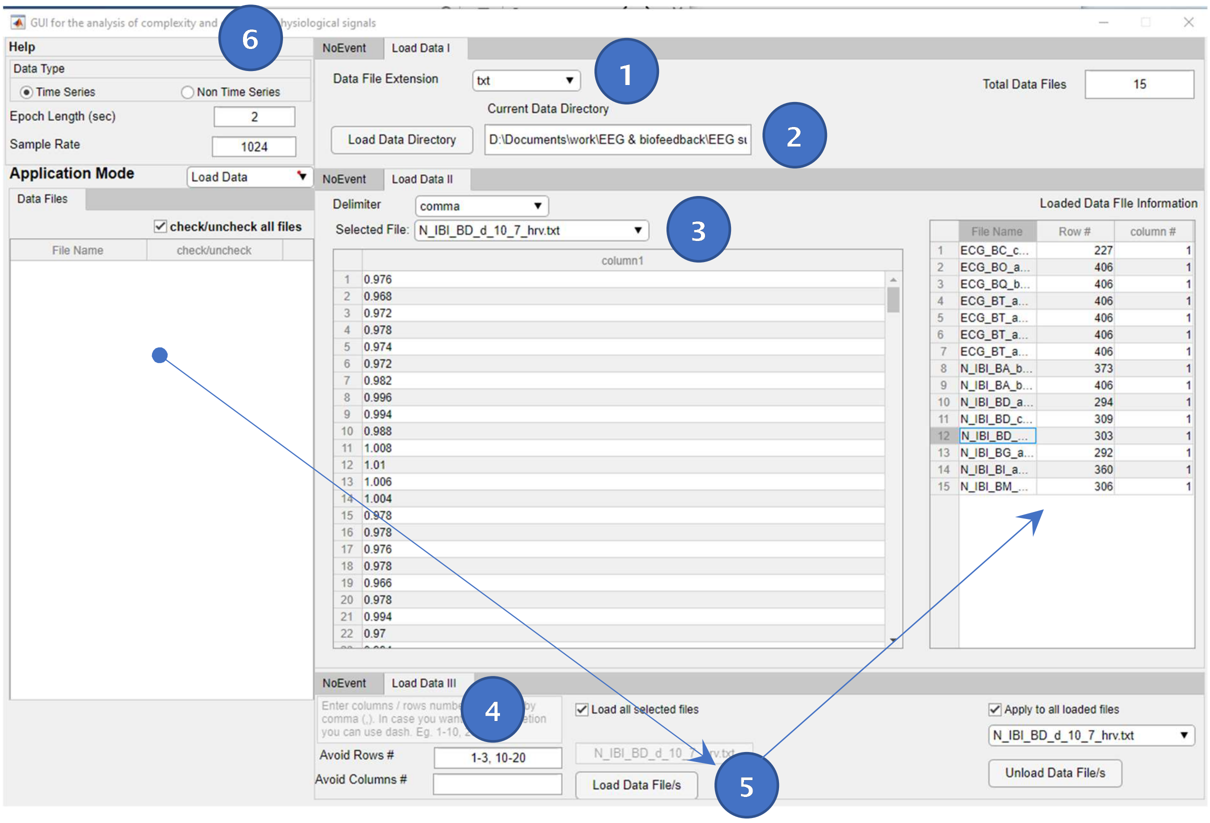

Figure 1. Loading Data.

[1] Starting in the upper pane, Load Data I, select the 'Data File Extension' of the file/s you wish to load. If you wish to load a file or files with multiple columns, select the correct column 'Delimiter' in the 'Load Data II' pane.

[2] Then click on 'Load Data Folder' and select the folder in which your data is stored. The pathway to this folder will appear in the 'Current Data Folder' box, the number of files in the folder with the selected Data File Extension will appear in the 'Total Data Files' box, and the file names themselves in the 'Data Files' list below the Application Mode combo box.

Notes

- If you reverse the order of [1] and [2], the file names will not appear correctly in the 'Data Files' list.

- If there are no files in the folder with the selected file extension, a warning message will appear:

This folder does not include data files like this.
Please select another folder or subfolder, or a different file extension.

- If a file has the correct extension but no data is found in the epoch/s selected, a warning message will appear:

Empty data for file number …

[Note: If you reverse the order of [1] and [2], the file names will not appear correctly in the 'Data Files' list.]

[3] If you wish to view the contents of one of the files in the 'Data Files' list, locate the name of the file in the 'Selected File' combo box in the central 'Load Data II' pane. The name of the file will appear in the 'Selected File' box, and its contents (whether single or multiple columns) will appear in the central 'Data Viewer' pane.

[4] It is possible to edit the matrix files being viewed to avoid certain rows (such as headers) or columns (such as time), so that only the data you require is available for further processing. Simply enter the column/row numbers you wish not to enter in the appropriate box/es in the 'Load Data III' pane, separated by commas or dashes: e.g. 1-3, 10-20. This will not affect the data in your original files.

If you wish to remove the same rows/columns from several files in the 'Data Files' list, make sure all their check boxes are ticked before proceeding.

[5] Next, you can load files with ticked check boxes. Simply tick 'All selected files' and click on 'Load Data File/s'. The names of your edited files will then move from the 'Data Files' list and appear in the right-hand column, 'Loaded Data File Information'.

If you wish to remove a single file name from the 'Loaded Data File Information' list, first click on the name in the list. This will bring up the same file name (greyed out) in the 'Loaded File' box below the list. You can then click on the 'Unload Data File/s' button to return the file to the 'Data Files' list on the left.

To remove all file names from the 'Loaded Data File Information', check the box 'Apply to all loaded files' before clicking on the 'Unload Data File/s' button.

[6] At the top of the GUI, below 'HELP', you can select whether the files you wish to process contain Time Series or Non Time Series Data. For the former, you will need to enter a Sample Rate (sampling frequency) in cycles per second (Hz), and if you wish to subdivide the data into non-overlapping epochs for analysis, an Epoch Length in seconds will also need to be specified. For Non Time Series Data, you will need to enter *N* Data Points per Epoch. The Total does not have to be an exact multiple of Epoch length. The default values provided may be changed to suit your needs.

Notes

- For RR interval (inter-beat) data derived from ECG recordings, there will be no Sample rate to enter.
- Segmenting your data into epochs may be appropriate if it is nonstationary, or if computing some measures for the whole dataset is too time-consuming. Obviously, Epoch length cannot exceed the length of your data sample.
- If the GUI is closed and re-opened, you will have to re-enter the required Epoch Length and Sample Rate (or non-time series equivalents).
- Be aware that your selection of Epoch Length and Sampling Rate (No. of Data Points per Epoch for non-time series) data will affect, for example, how your data is displayed and the settings you need to use to truncate it in the Pre-Processing Application Mode.

You can now move on to pre-processing your data.

Sample data provided

In addition to your own data, you can work with sample data packaged with CEPS in the 'SampleData' folder:

In subfolder 'RR_Data', 15 .txt files of unprocessed ('raw') 300-second HRV RRi data (recorded using the Mitsar 202-24 (DC) and WinEEG software (Mitsar, St Petersburg).

In subfolder 'EEG_Data', one file of raw 19-channel EEG in both *.csv and *.xlsx format ('BA_a_25_1_EEG'), 2000 data points long (sampling frequency 250 Hz). one column per channel. The same data, in 19 separate single-channel *.txt files.

Currently, white, blue, red, violet and pink noise, as well as sine and cosine waves, can be added to data in thePre-Processing 'Add Noise' Tab. The following synthetic data are also available in the com/ExternalPackages/DataShare folder (although not accessible via the GUI), for comparison or adding to signals to be processed:

- Signal with logarithmic chirp and AM modulation (amchirp.m)
- Chirp signal with constant amplitude from the MATLAB library (constantchirp.m)
- White, pink and brown noise, i.e. with power spectral density (PSD) constant, $\propto 1/f$ or $\propto 1/f^2$ (appendednoise.m)
- MIX signal with a periodic and a stochastic process (mixsignal.m)
- Amplitude-modulated quasi-periodic signal with additive white Gaussian noise (WGN) (quasipernoise.m).

PRE-PROCESSING DATA (optional)

If your data is ready to process, you do not need to use this Application Mode, but it can be very useful for exploring your data, removing outliers and truncating or deleting noisy or unwanted segments.

When the Pre-Process Data mode is selected, the list of Measures below the Application Mode selection box is inactivated and the central Panel on the right ('Data Pre-processing II') will show two empty plots in the 'View Data' Tab.

To plot your raw and pre-processed data, you will first need [1] to select the File you wish to view in the 'File/s' combo box and [2] then click on 'Plot Data' below the two plots. (You cannot view data from several files at once in this Tab.) In the View Data Tab, the raw data will appear in black in the upper plot, the pre-processed data in blue in the lower plot. Initially they will look the same (**Figure 2**).

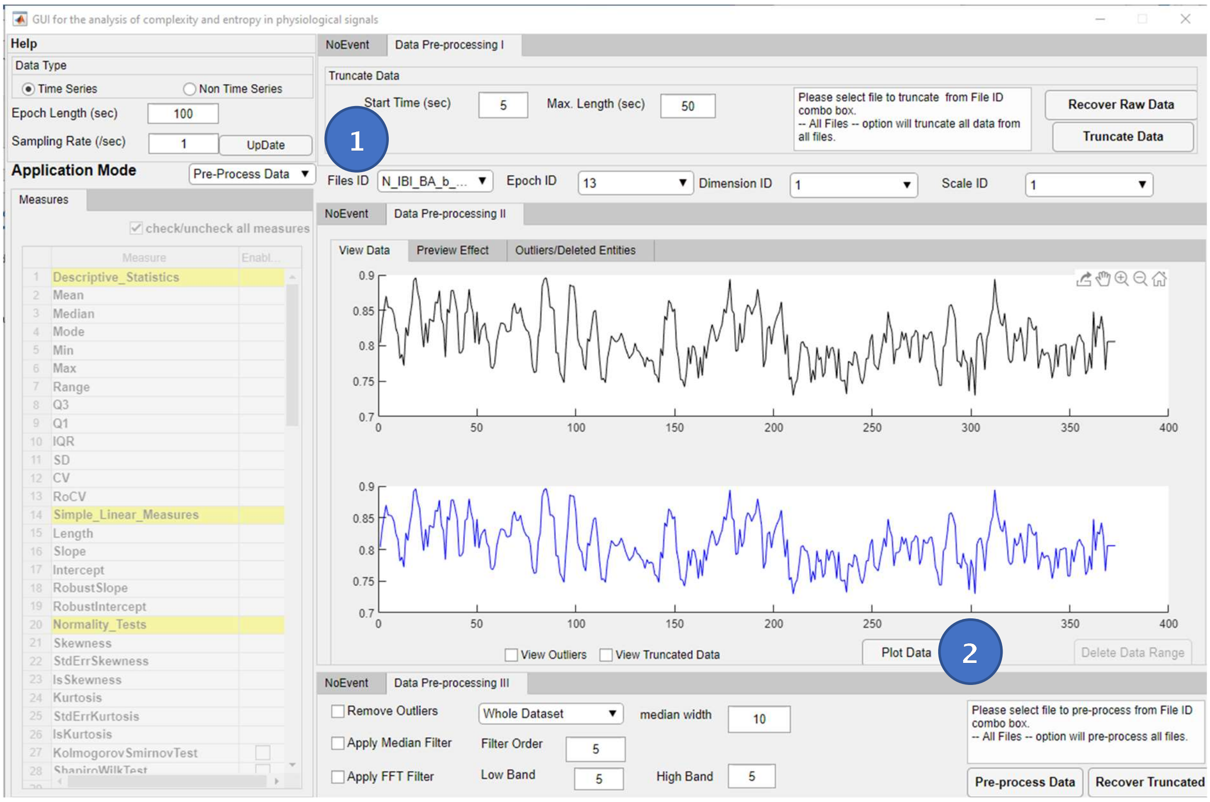

Figure 2. Pre-processing Data. The View Data Tab.

Truncating and Deleting Data

Above the plots (Data Preprocessing I), there are check boxes for truncating data. The box on the left is to truncate at the *start* of the data, *up to* the data point or time you enter; the box on the right is to truncate at the *end* of the data, *from* the data point or time you enter to the end of the data.

For example, if you wish to delete the first 5 seconds of data, and all data from the 100th data point, you would [1] enter 5 and 100 in the two boxes, respectively, [2] click on 'Truncate Data' and then [3] click on 'Plot Data' below the two plots. With 'View All Data' checked [4], you will then see the raw data in the upper plot, with the two ranges to be truncated highlighted in colour [5]. The lower plot will show the data remaining once the truncated segments have been removed [6] (**Figure 3.1**). If 'View All Data' is *not* checked, the remaining data will appear in both upper and lower plots. If you make a mistake, you can use the 'Recover Raw Data' button above 'Truncate Data' to clear the plots and start again.

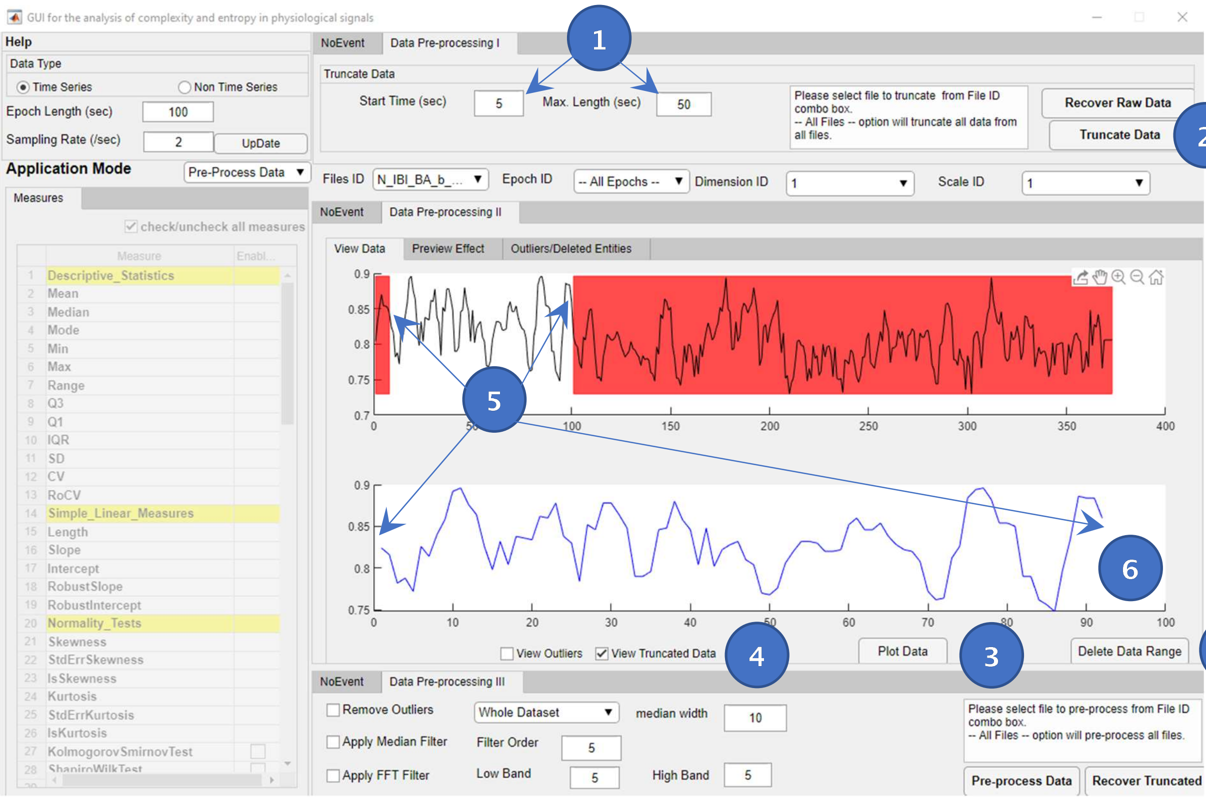

Figure 3.1. Pre-processing Data. Truncating Data using selection boxes.

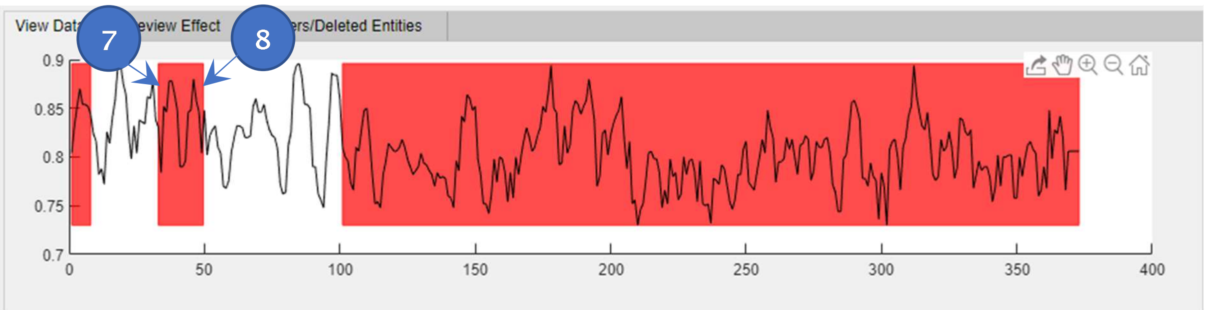

Figure 3.2. Pre-processing Data. Truncating Data using the mouse.

To remove/delete a single data point or range of data anywhere in the chosen file, you can also manually [7] click the left mouse button to start and [8] the right mouse button to end your selection further to the right, then [9] click 'Delete Data Range' to delete the selected segment. As when truncating data, results can be viewed by checking the 'View All Data' box. **Figure 3.2** shows data truncated using both the 'Truncate Data' selection boxes and the mouse method. (You cannot truncate several ranges at the same time using the mouse method.)

Truncating data may be useful if datasets to be compared are of different lengths, for instance, or if you wish to analyse a complete data sample but exclude some data at the start or end of your sample, or even delete a segment midway, perhaps because of the presence of artefacts or noise. For removing time series data at the start of your sample, enter the time *up to which* you wish to truncate data (in seconds). Similarly, to remove time series data at the end of your sample, enter the time *from which* you wish to truncate data, leaving the section of data you wish to analyse between them these two. For non-time series data, enter the corresponding Data Points.

You can truncate several files at once by selecting '—All Files—' from the File/s combo box.

Outliers and Filters

Below the plots (Data Preprocessing III), there are check boxes for removing outliers (defined in terms of Tukey's fences) or noise, or applying a Median or Fast Fourier Transform (FFT) filter to the data. Outliers may be removed for the Whole Dataset, or epoch by epoch (Per Epoch), but not both at once. How much height is removed from the outliers can be adjusted using the Median Width parameter check box (more may be removed if the median width is greater). Outliers can be viewed in the 'View Data' Tab by checking the 'View Outliers' box.

Data can also be smoothed by using the separate Median Filter setting: greater smoothing occurs with a higher order filter.

An FFT bandpass filter can also be applied to remove noise at low or high frequencies using the 'High Pass' and 'Low Pass' checkboxes, respectively.[4]

Note

[4] Low frequencies are removed and higher frequencies retained by the 'High Pass' filter, and vice versa for the 'Low Pass' filter.

Your data may include extreme values (outliers) that appear so different from other values in the data that they may be incorrect, the result, perhaps, of equipment malfunction or human error. On the other hand, although unusual or unexpected, they may in fact be 'real'. In the first case, you may wish to exclude outliers from your analysis, but in the second you may wish to include them. Or you may wish to compare results from your data with and without outliers.

In CEPS, outliers are defined in terms of 'Tukey's fences'. If Q1 and Q3 are the lower and upper quartiles of your data sample, respectively, then outliers are values outside the range [Q1 – 1.5 x (Q3 – Q1), Q3 + 1.5 x (Q3 – Q1)] (Tukey 1977).

You may wish to run some initial descriptive statistics or normality tests on your data to determine whether your data is normally distributed, before you decide whether or not to remove outliers and proceed to running your experiment.

Outliers may occur singly, with very steep onset and offset slopes, or clustered together, amplitude increasing and decreasing more slowly before and after. You therefore have the option to choose 'Median Width', i.e. for how many data points around each outlier peak you wish to replace the outliers with the median value of the data series over a number of data points centred about that value. In other words, a Median Width of 10 indicates that the median value is taken over ± 10 data points either side of the middle value, so over 20 points in total.

Like the 'Median Width' function, Median filtering replaces each value in a data series with the median of the data series over a number of data points centred about that value. This number is called the filter order, which may be odd or even. If some of the points to be used are outside the boundaries of the time series, their values are ignored when calculating the median value. Default order is 5. Median filtering may be useful for removing spikes or suppressing noise.

Once you have [1] selected the file you wish to pre-process and [2] chosen what Filter and Outlier parameters to apply, [3] click on 'Pre-process Data' in 'Data Pre-processing III' and then [4] on 'Plot Data' to view their effects (**Figure 3**).

Five Views for Pre-processing

Using the two different plots in the central 'View Data' section of the GUI (Data Preprocessing II) enables comparison between raw data and data with outliers removed, for example.

If you wish to change your mind after pre-processing data and decide instead not to apply any filters on your selected data, you can [5] click on the

'Recover Pre-processed' button to go back to the previous stage, undoing the effects of any filters you have applied.

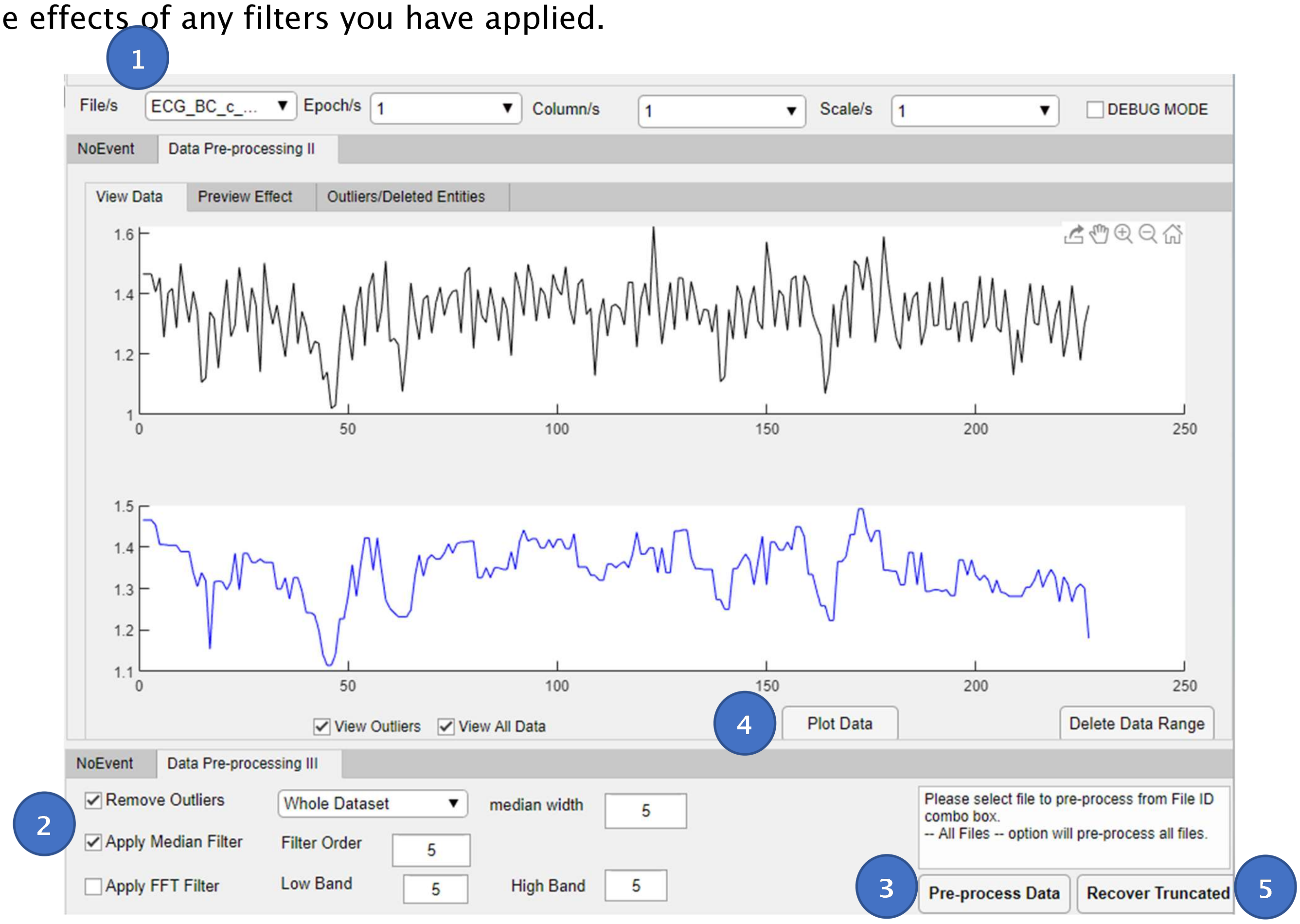

Figure 3. The 'View Data' Tab, showing the effect of removing outliers and applying a median filter.

Superimposed rather than separate plots can be displayed in the second, 'Preview Effect', Tab (**Figure 4**). As before, [1] select the file you wish to process, and, if required, [2] the data column within that file. To assist in parameter selection when pre-processing data, here you can preview the effect of selecting a range of different parameters for Outlier Median Width, Median Filter Order, and FFT High or Low Pass. You will [3] need to select *one* of these methods in the 'Select Filter Parameter Range' combo box, and [4] a corresponding range of values, together with [6] 'Per Epoch' (if required), before clicking on [7] the 'Preview Effect' button. Parameters for the other pre-processing methods will be those selected in the Data Pre-processing III pane [5].

You do not need to click on ‘Pre-process Data’ just to preview the effect of using different parameters. Plots can be cleared in both the ‘View Data’ and ‘Preview Effect’ Tabs by processing a new file. (‘Recover Raw Data’ will not work in the Preview Effect Tab.)

> **Note**
>
> If a plot remains blank, try clicking on the ‘home’ ‘Restore View’ icon in the upper right corner of the plot. If it doesn’t look right, try clicking on the ‘Recover Raw Data’ button and re-plot.

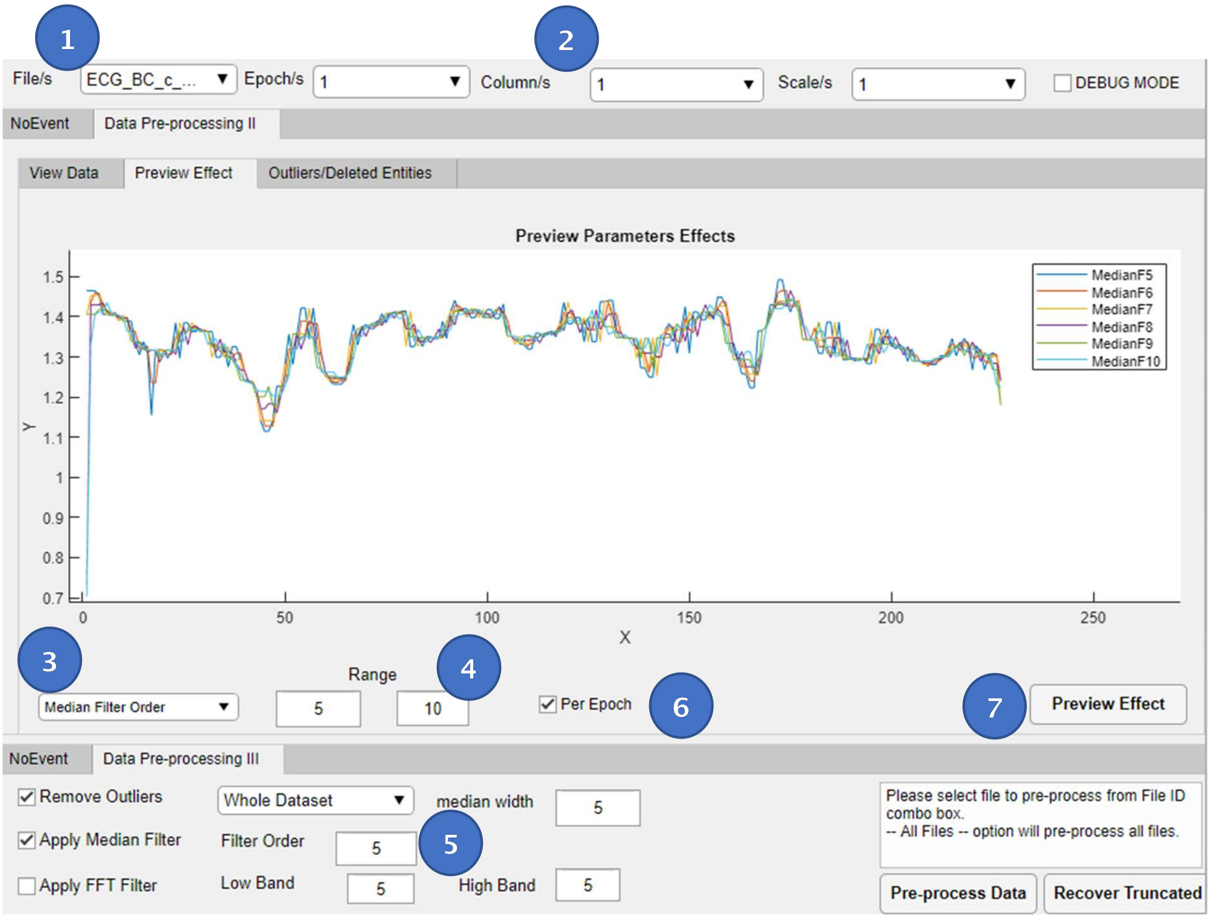

Figure 4. Pre-processing Data. The Preview Effect pane.

Outliers and Deleted Entities are shown in two lists in the third Tab. Results for all or selected files can be viewed (using the ‘File/s’ combo box [1]), and the ‘Column/s’ box [2], if required. Outliers, their original values and their values when replaced using the median width setting can be viewed on the left [3], and deleted or truncated segments on the right [4], showing where the removed segments start and end. To view the Outlier List, click on ‘Pre-process Data’ [5] and then ‘Refresh’ [6], just below the two lists. To view the ‘Deleted Entities’, you will need to have either truncated or deleted data in a previous step, or you can use the ‘Truncate Data’ button [7] now and then ‘Refresh’ once more.

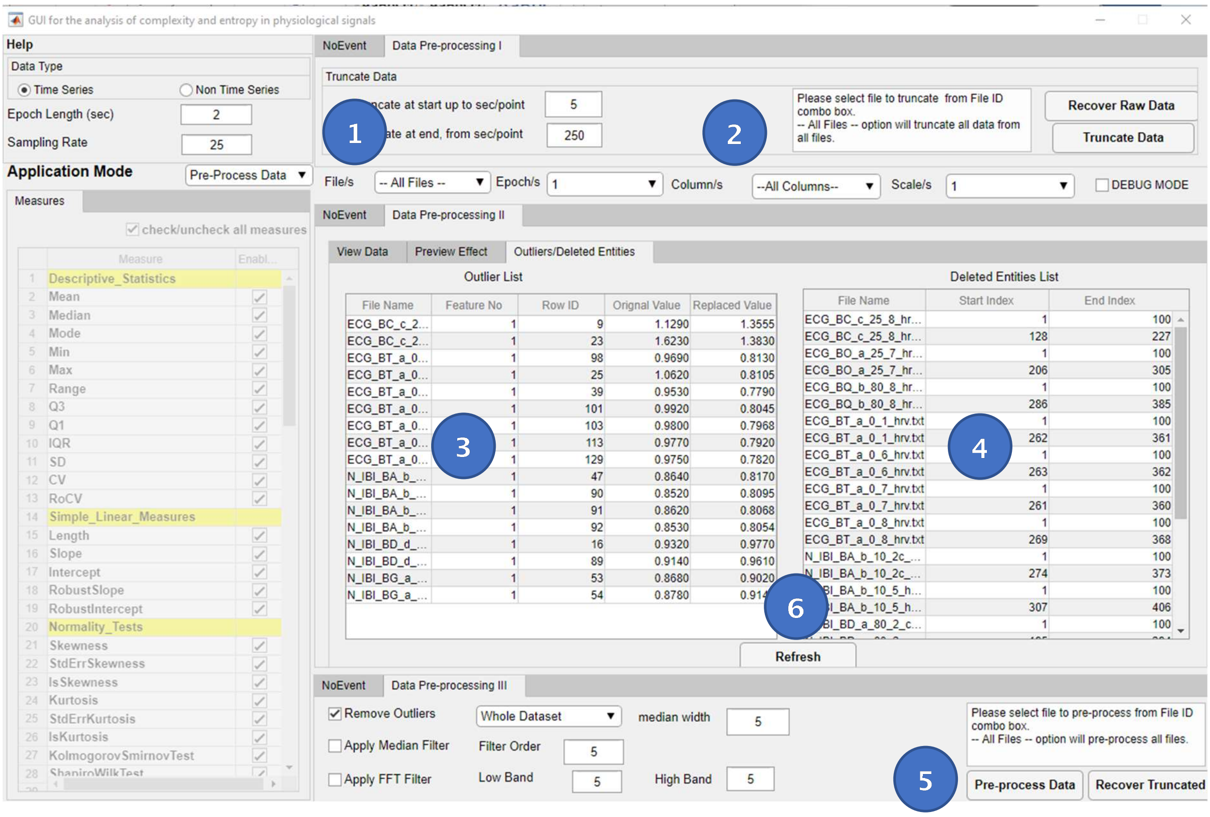

Figure 5. Pre-processing Data. The Outliers/Deleted Entities Lists.

> Note
>
> - If the GUI is closed and re-opened, you will lose any settings you have entered during Pre-Processing.
> - All filtered data can be recovered by clicking the ‘Recover Pre-Processed’ button. CEPS also provide the facility to recover raw data by clicking the ‘Recover Raw Data’ button to recover even deleted data and take you back to the original file.

In the fourth Tab (‘Compare Data Files’), shown in **Figure 6**, different files can be viewed simultaneously. For this, their names should be selected sequentially in the File/s combo box [1], clicking on ‘Plot Data’ [2] after each file is chosen. Plots for the selected files will appear *superimposed* in the upper plot (‘Raw Data Files’) and lower plot (‘Processed Data Files’). The two sets of plots will be identical until parameters are selected in the ‘Data Pre-processing III’ pane [3]. Then, when the ‘Pre-process Data’ button [4] is pressed, followed again by ‘Plot Data’ [2], it becomes possible to view the effects of pre-processing several files at once in the lower plot. It is also possible to set parameters first, and then click on ‘Pre-process Data’ [4], followed by ‘Plot Data’ [2], for each file in succession. If the upper and lower plots look identical even though data has been pre-processed, the ‘Recover Raw Data’ [5] button can be used, followed by ‘Pre-process Data’ [4] and ‘Plot Data’ [2] once more. ’Clear Plot’ [6] and ‘Recover Raw Data’ (for ‘All Files’ in the File/s combo box [1]) can be used to start again.

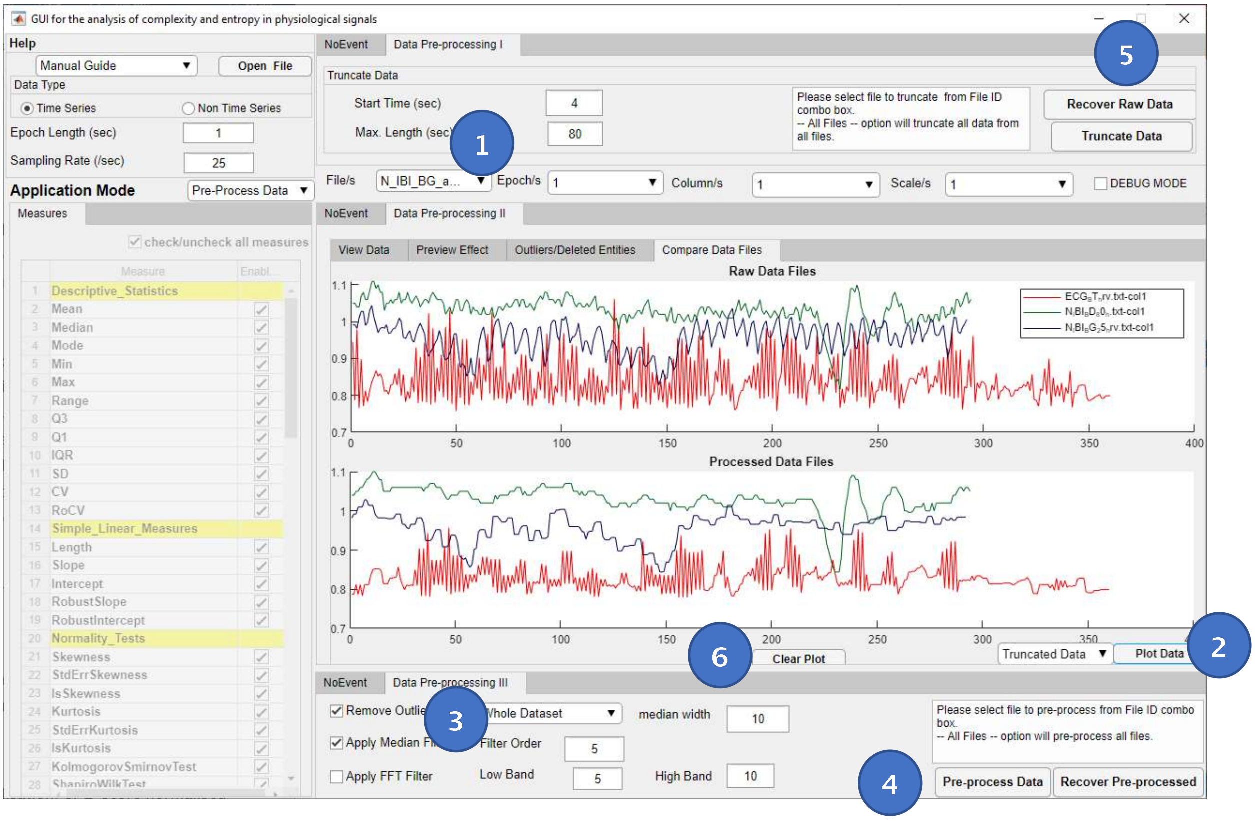

Figure 6. Pre-processing Data. The Compare Data Files Tab.

The first four Tabs in the ‘Data Pre-processing II’ panel can all be helpful when deciding how much data to process, and how much to remove as outliers. Of course, outlier removal, data smoothing and coarse-graining may well affect complexity and entropy results, so should be explored carefully.

In the fifth Tab, it is possible to add noise of different types to single- or multi-column data for experimental purposes. As shown in **Figure 7**: [1] Select a file; [2] If data is multi-column, choose the column to which you wish to add noise; [3] Select the type of noise to add; [4] Set the 'blending coefficient' to determine the relative amount of noise added; [5] Set Noise Wave Frequency and Amplitude (if 'Sine Wave' or 'Cosine Wave' have been selected in Step 3). [6] Click on 'Preview Effect' to view the result. If you wish to add noise to all columns of your data, check the 'Apply in all Columns' box [7]. Finally [8], Click on 'Save Noisy Data' to save your original data file, now with noise added, to the folder CPES/outputs/noisyData, where it will appear with the original file name followed by that of the noise type added.

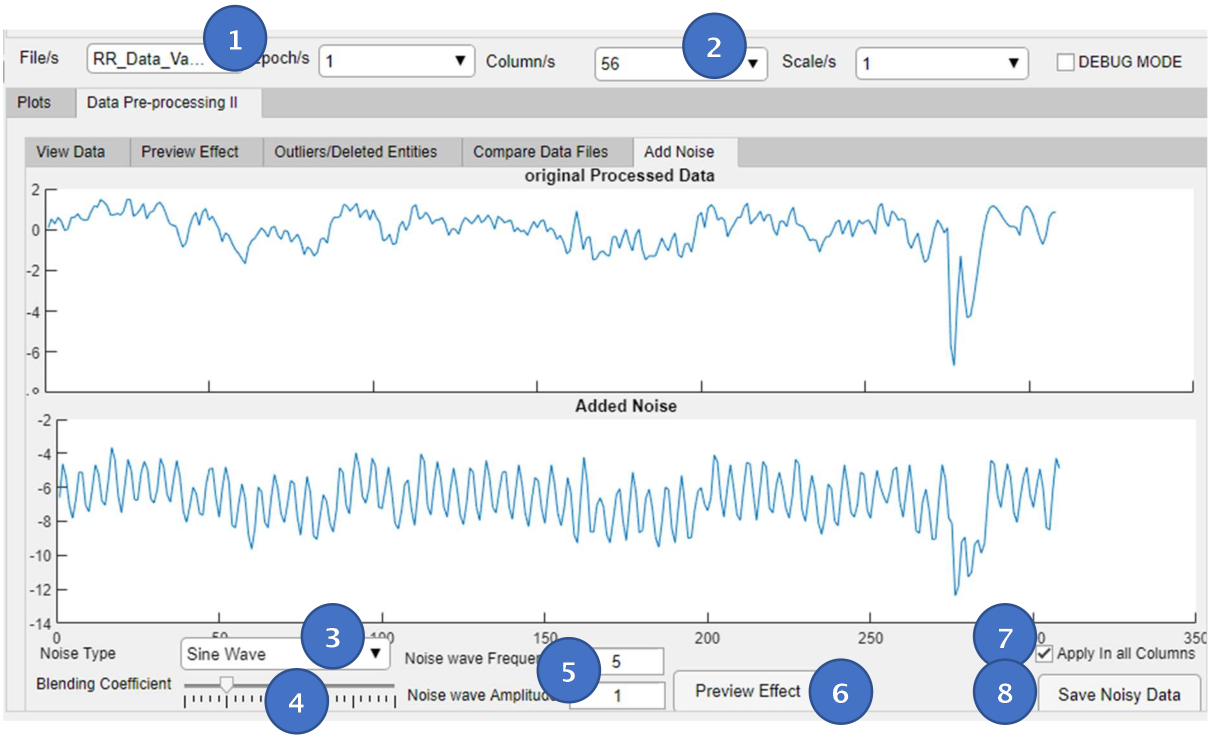

Figure 7. Pre-processing Data. The Add Noise Tab.

In a future version of CEPS, it will also be possible for data to be detrended, rescaled (using min-max normalisation) or Z-score normalised (standardised). Detrending may be required if there is an obvious slope or wobble to the data baseline. Options will be provided for linear or polynomial detrending, accordingly. Normalising data may be useful when comparing recordings made with different settings, and may also be a prerequisite for computing some entropy measures whose code does not already include normalisation.

In addition, because many entropy measures are appropriate for discrete rather than continuous data (see Table 9 in the paper on CEPS to be published in *Entropy*), a selection of coarse-graining methods will be provided. These are explained in more detail in the CEPS Primer.

TESTING PARAMETERS

A serious problem with many complexity and entropy measures is how to select the parameters used to estimate them. Using fixed parameters may not work well all the time [Shi 2017]. A 'test and plot' facility is therefore provided to assist in both parameter selection and multiscale analysis, for single or multiple files, or epochs within a file. The 'problem' becomes an opportunity for experimentation.

In the Test Parameters mode, only those measures in the list on the left of the GUI which require parameter settings can be selected using the check boxes provided, and only one at a time.

Selecting a measure in the list [1] will bring up two boxes for each parameter required in the central panel of the GUI (Run Parameter Test II). When the measure is selected, the name of its parent section in the Measures List replaces 'Run Parameter Test II' as the name of the central panel [2]. A range of values for each parameter can then be tested, again for only one parameter at a time (it would be possible, but confusing, to test the effects of changing two or more parameters at once, so this is not implemented in the current version of CEPS). The minimum value to be tested is entered in the left-hand box for the parameter, and the maximum in the right-hand box [3] **(Figure 8**). Care should be taken to enter values that will give meaningful results (guidelines are provided for each measure **in Table 1** below). If a minimum or maximum value is selected that is completely out of the acceptable range for the parameter being tested, a warning tone may sound when trying to plot results and no plot (or a blank plot) will appear.

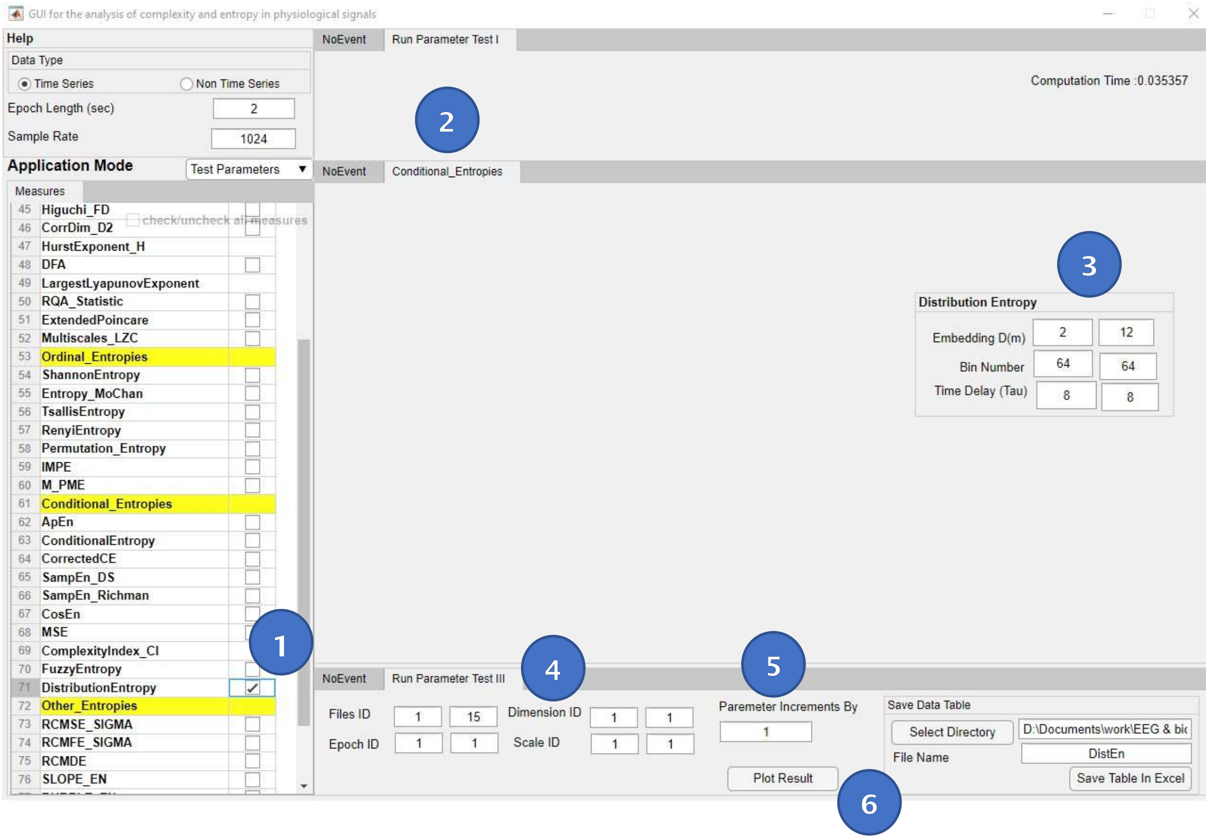

Figure 8. Setting a range of values for the 'Embedding Dimension' (DistEn) and saving the results.

When a particular measure is selected for parameter testing, using the checkboxes above the central panel you can choose whether to Enter Epoch Number/s, select All Epochs (individually), or Consider as Single Dataset (all epochs taken together). In the pane 'Run Parameter Test III' below the central panel, you can choose for how many File/s or Epoch/s you wish to show a Plot Result, with the minimum value entered in the left-hand box and the maximum in the right-hand box [4].

Parameter increments between the minimum and maximum values chosen can also be set [5]. Whether integers (1, 2, etc) or decimals (0.1, 0.2, etc.) are entered here will depend on the Parameter being tested. If only whole numbers are accepted but a decimal is used, the test may return zero or no values. Results can be plotted by clicking on 'Plot Result' [6] and, if required, displayed in a Data Table as well (**Figure 9**).

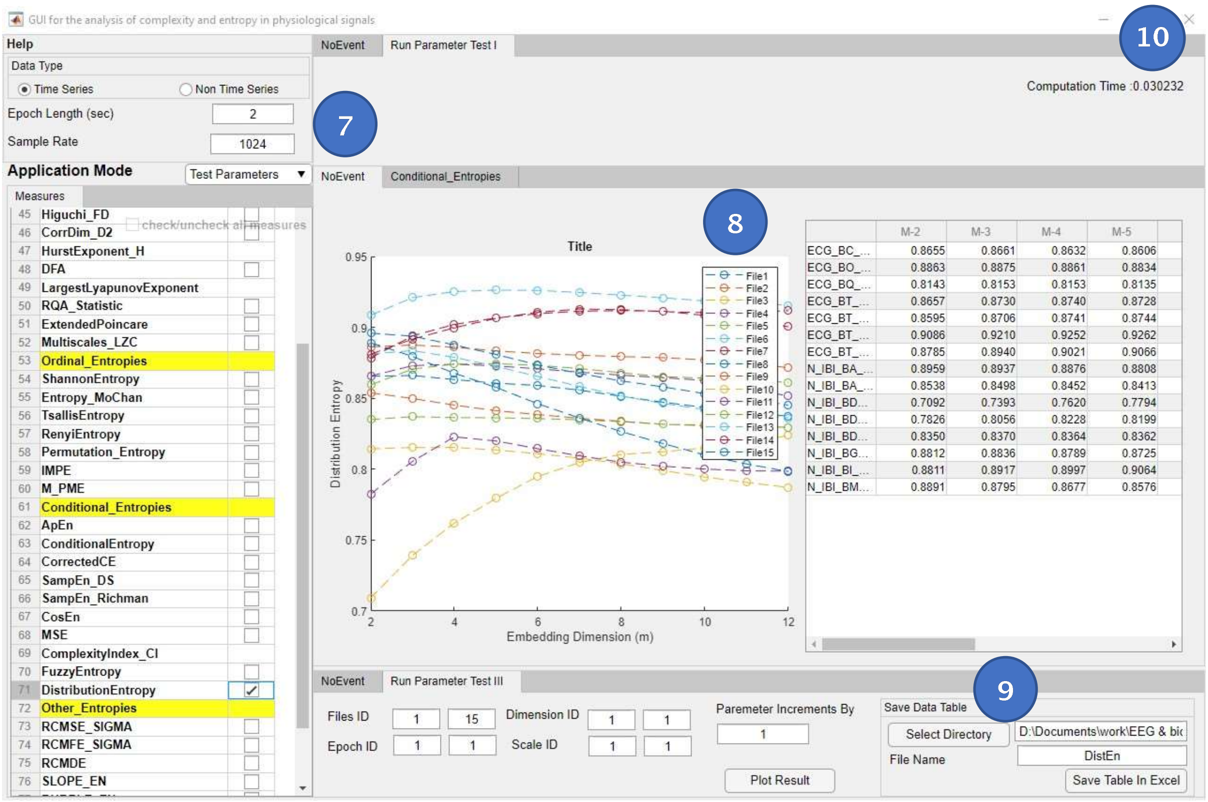

Figure 9. 'Test and Plot': Plot and Table results in Parameter Test Application Mode.

Both plot and table are shown in the central area of the GUI [7], but in a different Tab from the one in which parameters are entered.

Plots may be resized (using the 'home' 'Restore View' icon in the top right corner of the plot) or exported using the arrow icon [8], and the data in the table may be saved, as follows: In the 'Run Parameter Test III' pane, after selecting a Results Folder and File Name [9], results can be saved in Excel. It is thus possible to double check the effects of selecting different parameters on the behaviour of the measure investigated. Computation Time is shown in the lower pane, 'Run Parameter Test III' [10].

Parameter settings

Many of the measures implemented in CEPS require parameters to be set before they can be used. In **Table 1** we provide guidelines on default and other suggested settings. For more detailed information on the measures and their parameters, see the *Primer on Complexity and Entropy*, accessible via the drop-down menu under 'HELP' in the CEPS GUI. The various measures listed can be used on data in any of the accepted formats (txt, csv, mat or xlsx).

Table 1. Table of Measures.

Measures listed in CEPS, with default and suggested parameter ranges. Measures not yet fully implemented are shown on a light grey background. A dark grey background in the following columns indicates that no parameter setting is required.

Measure	Default Parameters	Parameter ranges	Notes
Descriptive Statistics			
Measures of central tendency			
Mean			
Median			
Mode			
Min			
Max			
Range			
UpperQuartile_Q3			
LowerQuartile_Q1			
Measures of dispersion			
InterquartileRange_IQR			
StandardDeviation_SD			
Coefficient_of_Variation_CV			
Robust_CV_RoCV			
Simple Linear Measures			
Length			
Slope			
Intercept			
RobustSlope			
RobustIntercept			
Normality Tests			

Skewness			
StdErrSkewness			
IsSkewness			
Kurtosis			
StdErrKurtosis			
IsKurtosis			
KolmogorovSmirnovTest	$p = 0.05$	$p \geq 0.05$	
ShapiroWilkTest	$p = 0.05$	$p \geq 0.05$	
Time Domain methods			
RMS			
RMS_SD			
Hjorth parameters			
HjorthActivity			s
HjorthMobility			s
HjorthComplexity			s
Frequency Domain methods			
FastLomb			
FFT			
Stationarity & Nonlinearity			
AutoCorrelation	N lags = 1 N SDs = 1	? ?	
AutoCovariance	$k = 1$	?	
ReverseA_Test1			
ReverseA_Test2			
MovingWindowTest	$m = 2$ $t1 = 0.25$ $t2 = 0.50$	? ? ?	
Nonlinearity_VM			sl
COMPLEXITY			
Dimensions & Exponents			
Higuchi_FD	$k_{max} = 6$	$k_{max} = 1$ to $N/2$	sl
AllanFactor_AF	Varies with N and 'period'	T = 'period'/2 to Length/10	l
CorrDim_D2	$m = 2$ $\tau = 1$	$m = 5$ to 20 $\tau = 1$ to 10	l
HurstExponent_H			
DFA	$n = 5$ $m = 2$		s

LargestLyapunovExponent_LLE	m = 10 τ = 10 MI = 50		l
RQA	m = 10 τ = 1 $r = \sqrt{10}$ L_{min} = 2		sl
ExtendedPoincaré_EPP	k = 1 (m)		sl
ComplexCorrelationMeasure_CCM	k = 1	k = 1 to 100	m
Multiscale_LZC	w = odd n ($\neq$1)		sl
SYMBOLIC DYNAMICS (SymDyn)			
SymDyn_Equal_Intervals	ζ = 6 k = 3		l
SymDyn_Mean_Based	α = 0.1 k = 3		l
SymDyn_Binary_Change	τ = 10 ms(h)		l
ENTROPIES			
Shannon & Generalised Entropies			
ShannonEntropy_SE	ζ = 6 k = 3(p)		sl p
RényiEntropy_RE	q = 0.25 or 4, $q \neq 1$		m
TsallisEntropy_TE	tbc	0.5 to 5.0	
Further developments from Shannon Entropy			
AverageEntropy_AE	max from data min from data s_1 = 55, 300 τ	τ = 1 to 10	cd sl
Entropy.of.Entropy_EoE	max from data min from data s_1 = 55, 300 τ	τ = 1 to 10	cd sl
ToneEntropy_T_E	tbc		d m
Entropy.of.Difference_EoD	m = 4 s = 1		
KullbachLeiblerDivergence_KLD	m = 4 s = 1		
Ordinal Entropies			
PermutationEntropy_PE	m = 5		sl

	$\tau = 1$		
AmplitudeAware_PE	$m = 6$ $\tau = 1$ $A = 0.5$	$m = 4$ to 9 $\tau = 1$ to 10 $A = 0.1$ to 0.5	m
ImPE	$d = 2$ $\tau = 1$		s
mPM_E	$m = 3$ $\tau = 1$	$m = 3$ to 6 $\tau \geq 1$	sl
Conditional Entropies			
ConditionalEntropy_CE	$L = 10$ $\zeta = 6$		ml
CorrectedConditionalEntropy_CCE	$L = 10$ $\zeta = 6$		ml
ApEn	$m = 2$ $r = 0.2$	$m = 2$ to 10 $r = 0.1$ to 0.25	l
SampEn	$m = 2$ $r = 0.2$ $\tau = 1$		m
CosEn_&_QSE	As for SampEn		s
fSampEn	$N = 200$ $\% = 95$ $m = 1$ $r = 0.3$		m
mSE	$m = 2$ $r = 0.2$ $\tau = 20$		m
ComplexityIndex_CI & mSlope	As for MSE		m
FuzzyEntropy_FE	$m = 2$ $r = 0.3$ $\tau = 1$ mf = Gaussian(s) mf = Exp(l)		s
Other Entropies			
RCmSE_σ	$m = 2$ $r = 0.15$ $\tau = 1$ $\tau_{max} = 15$	$\tau = 1$ to 4, $\tau = 5$ to 12	l
RCmFE_σ	$m = 2$ $r = 0.15$ $\tau = 1$	$\tau = 1$ to 50	sl

	τ_{max} = 15 FP = 2		
RCmDE	m = 3 c = 6 τ = 1 τ_{max} = 30	c = 2 to 9 τ_{max} = 4 to 30	sl
DistributionEntropy_DistEn	m = 3 M = 64 τ = 8		s
SlopeEntropy_SlopeEn	m = 6 τ = 1 γ = 1 δ = 0.001		sl
BubbleEntropy_BE	m = 10		sl
PhaseEntropy_PhEn	k = 16 τ = 1	k = 2 to 50	sl
Time-Frequency Domain methods			
SpectralEntropy_SpEn			s
DifferentialEntropy_DiffEn			l
ANCILLARY METHODS			
AutoMutualInformation_AMI	$B \approx \sqrt{(n/8)}$ τ_{max} = 20		
FalseNearestNeighbours_FNN	m_{max} = 10 τ = 1 rf = 2		
AveragedFalseNeighbours_AFN	τ = 1	τ = 1 to > 10	

c: suited to continuous data; **d**: suited to discrete data; **h**: in HRV studies; **l**: suited to long datasets (10,000 points or more); **m**: returns the conventional (scale = 1) measure; **n**: data requires normalisation to unit SD for CE and CCE – this is included in the code for CE and CCE; **p**: parameters are only required for Porta's algorithm; **s**: suited to short datasets (i.e. 100 data points or possibly less).

RUNNING THE PIPELINE

Both when testing parameters and running the pipeline, in the upper pane ('Pipeline Execution I') you can select whether to process either a complete data series or some or all of the epochs into which it may be subdivided. Epochs selected may be separated by commas (e.g. 1, 3, 6) and/or a dash (e.g. 1-10).

When running the pipeline, in contrast to when testing parameters, several measures can be selected in the Measures List at the same time, according to your requirements and research design, the main limitation being computer processing speed. However, now only one value can be assigned to each parameter for a particular measure, and this value will be used for all data files processed (hence the importance of testing parameters in order to make an informed decision about which values to use). As when testing parameters, options can be selected in the 'Pipeline Execution II' pane/s (**Figure 10**).

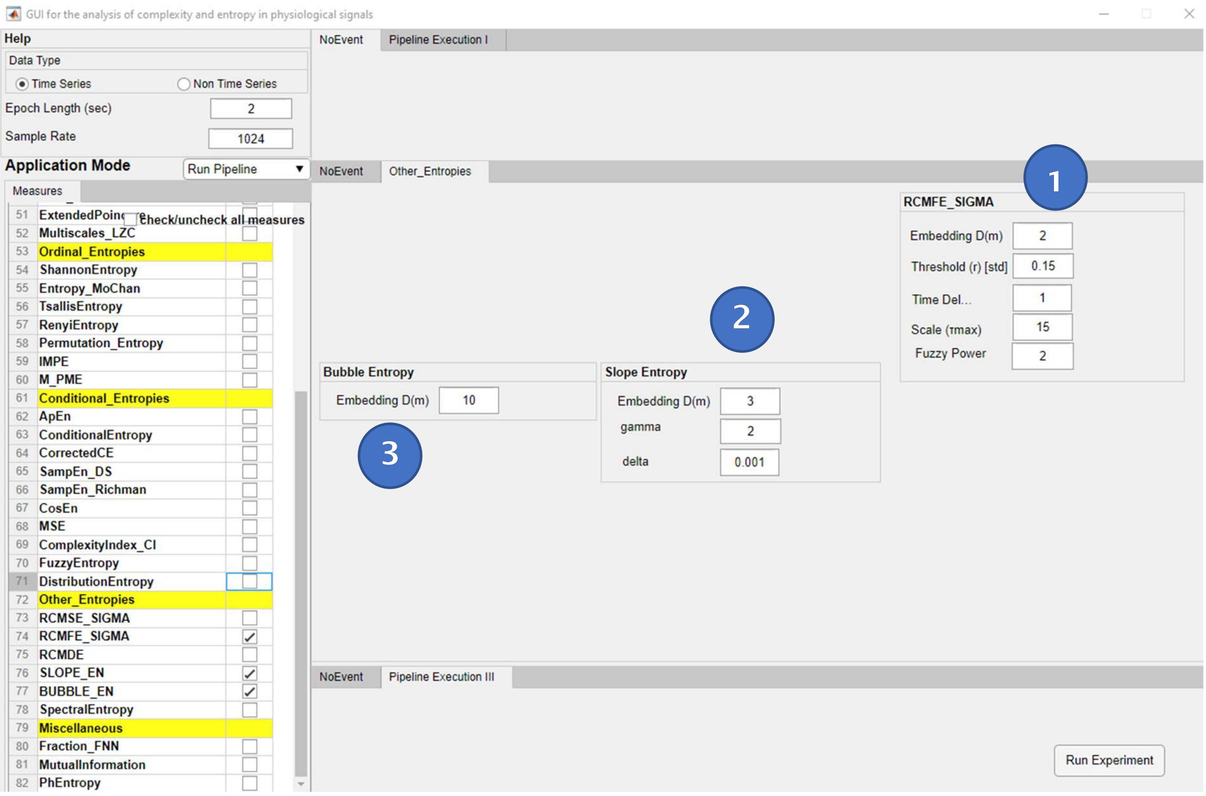

Figure 10. The GUI in Running Pipeline Application Mode, showing parameters selected for three measures.

A full list of the measures listed is provided in **Table 1** above. They are described in both broad-brush terms and more technical detail in the CEPS Primer accessible using via the drop-down menu under 'HELP' in the GUI.

Note

- Obviously, the number of Epochs analysed cannot exceed the length of your data sample.
- If the GUI is closed and re-opened, you may have to re-enter any parameters you have used previously and wish to re-use, if these differed from the default parameters.

PROCESSING RESULTS

In the ‘Process Results III’ pane, after selecting a Results Folder and an Experiment Name [1], results can be saved in Excel and/or matlab format [2]. Parameters used for the various measures will be saved in the same file, as will Computation time if this option is selected [3]. Results for each file processed can all be saved in the same Excel worksheet, or in separate sheets [4]. For batch processing, the option is given to aggregate results by mean and standard deviation or median and interquartile range [5] (**Figure 11**).

When saving the file, the Experiment Name selected should not include a slash (/) or back slash (\), question mark (?), asterisk (*), colon (:), ‘pipe’ (|), or the ‘greater than’ (>) or ‘less than’ (<) signs. All other characters are permitted.

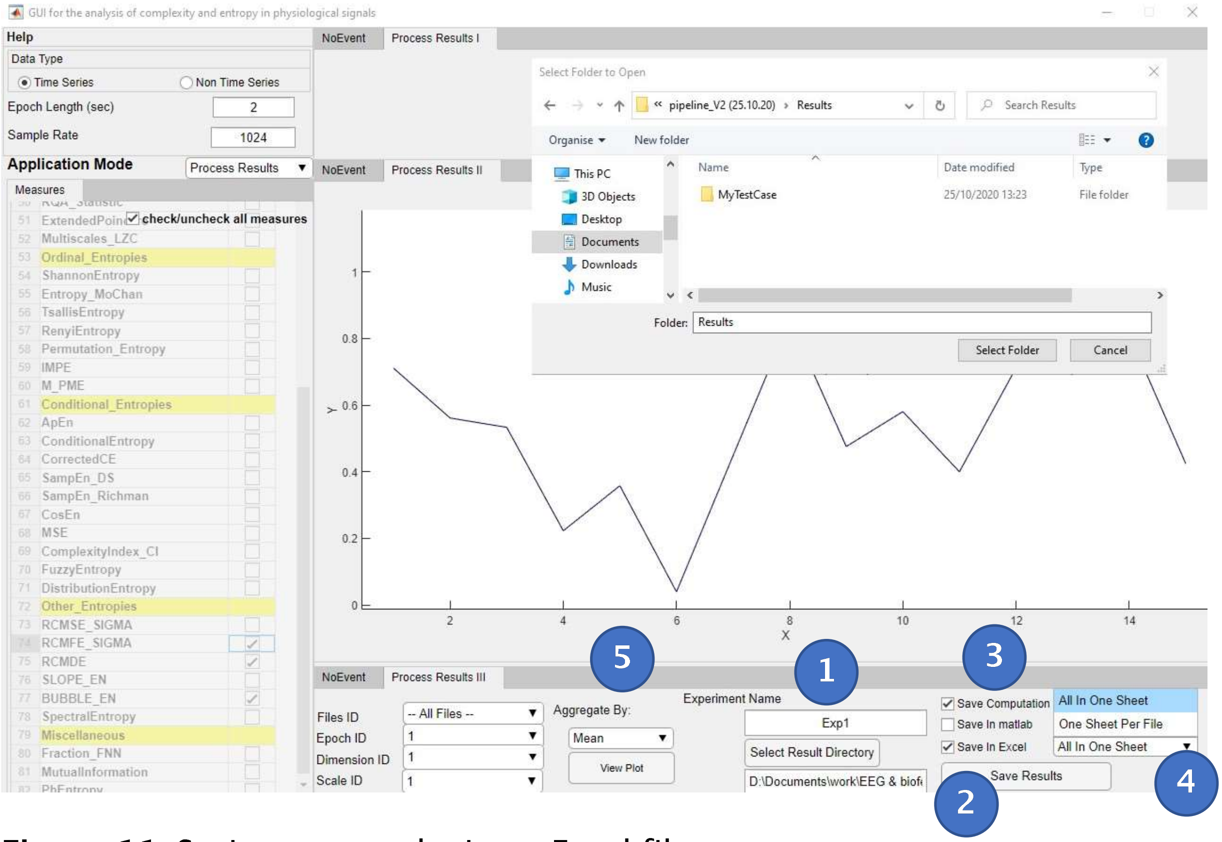

Figure 11. Saving out results in an Excel file.

CLASSIFYING RESULTS

This Mode, not yet implemented, will be useful when batch processing. For Simple classification and binary groupings of individuals (as 'well' or 'unwell', for example), it will be possible to assess and compare the different measures for their sensitivity, specificity (selectivity), precision and accuracy in classifying the individuals in each group. It will also be possible to compute the Matthews Correlation Coefficient (MCC) for unequal groupings and for more than two groups. Results for the different groupings will be displayed using 'View Plot', the plots exportable using the MATLAB export icon (curved arrow).

For ensemble classification, when more than two measures are considered at once, advanced and more computationally demanding methods are needed, such as *k*=nearest neighbours, linear discriminant analysis or AdaBoost. These are currently beyond the scope of CEPS, but are available in MATLAB's Statistics and Machine Learning Toolbox.

TROUBLESHOOTING

CEPS was created using MATLAB App Designer. To open CEPS, you will need to use MATLAB release R2018a or later. If you are running an earlier release, MATLAB should allow you to access a compatible version of CEPS.

Error messages may appear in the MATLAB Command Window or as pop-ups in the GUI itself. If a persistent problem occurs, you can try running the pipeline in DEBUG MODE (available at any time when using CEPS). If the problem persists, it may be reported on the BitBucket site, or directly to Deepak Panday.

When testing parameters or Running the pipeline, if no result appears for Computation Time, the algorithm has failed to perform.

If you hear a warning tone and little else happens when trying to start an operation, the simplest method to halt it is to close and re-open MATLAB. When you have done this, remember to check settings in the Data Selection, Parameters and Running the Experiment sections before trying to use the GUI again.

LICENCE & DISCLAIMER

If you use CEPS, please cite the following:

Mayor D, Panday D, Kandel HK, Steffert T, Banks D. (2021). CEPS: An Open Access MATLAB Graphical User Interface (GUI) for the Analysis of Complexity and Entropy in Physiological Signals. *Entropy* **23**(3), 321, 1–34; https://doi.org/10.3390/e23030321. Also available at: https://www.mdpi.com/1099-4300/23/3/321.

APPENDIX. Methods and Measures – about the MATLAB codes used

This Appendix provides a list of methods and measures used in CEPS to pre-process and process data, together with sources for the MATLAB codes implemented. So far as possible, we tried to use Matlab code already tested and published. If no source is given, code was written by Deepak Panday.

The Complexity, Entropy and other measures for which further information may be found in the CEPS Primer are asterisked in the list below. Details are provided only for measures not included there that may be unfamiliar to some CEPS users.

Data Pre-processing

Outlier Removal

FFT Filter

Truncating Data

Data Normalisation

Detrending

Coarse-graining*

1. Decimation
2. Equal intervals
3. Mean-based symbolisation
4. Binary Change Δ
5. Binary Change |Δ|
6. Coarse-graining

Data Testing and Processing

Descriptive statistics I. Measures of central tendency

Mean

This is a built-in standard MATLAB function for calculating the average of a single column ('vector') of data values.

Median

This is a built-in standard MATLAB function for calculating the median or middle value (second quartile, Q2) of a single column ('vector') of data values.[5]

For data that includes outliers or is not normally distributed when tested, it may be more appropriate to use the median as a measure of central tendency rather than the mean.

Mode

This is a built-in standard MATLAB function for finding the value that occurs most frequently in a single column ('vector') of data values.

Descriptive statistics II. Measures of extension

Min

This is a built-in standard MATLAB function for finding the minimum value that occurs in a given dataset.

Max

This is a built-in standard MATLAB function for finding the maximum value that occurs in a given dataset.

Range

[5] Please note that the MATLAB median and quartile functions may return slightly different values than the corresponding functions in EXCEL, for example. A useful explanation of such differences can be found at https://en.wikipedia.org/wiki/Quartile [Retrieved August 13, 2020].

The range of a list of numbers is the difference between the maximum and minimum values in the list. The GUI uses this built-in standard MATLAB function to determine the range of value in the dataset or epoch/s being investigated.

> Note on NaNs
>
> The Range function treats 'NaNs' as missing values, and ignores them. A NaN ('Not a Number') results from an operation such as dividing by zero or taking the logarithm of a negative number. In Excel, NaNs appear as #NUM! or #DIV/0!

UpperQuartile_Q3

The upper (or third) quartile of a set of numbers is the middle value between the median and the highest value in the data set. As 75% of the data lies below this point, it is also sometimes known as the '75th percentile'. It is calculated using the built-in standard MATLAB function for 'quantile'.

LowerQuartile_Q1

The lower (or first) quartile of a set of numbers is the middle value between the lowest and the median value in the data set. As 25% of the data lies below this point, it is also sometimes known as the '25th percentile'. It is calculated using the built-in standard MATLAB function for 'quantile'

Descriptive statistics II. Measures of extension

InterquartileRange_IQR

The interquartile range (IQR) is the difference between Q3 and Q1, and is another built-in MATLAB function. Like standard deviation (SD), it is a measure of dispersion or variability, but can be used for data which is not normally distributed.

StandardDeviation_SD

Standard deviation (SD) of a data sample, the square root of the data's variance, is another built-in MATLAB function. Like 'Range', it ignores NaNs.

Coefficient_of_Variation_CV

The coefficient of variation (CV) of a data sample is defined as the ratio of the SD of the sample over its mean. Unlike the SD, the CV is independent of the units of measurement that are used, so allows comparison of measurement data using quite different units, or different means. A proviso is that the CV is only appropriate for data measured on a ratio scale. The built-in MATLAB function for CV, like that for SD, ignores NaNs.

MATLAB: Bettinardi RG (2020). getCV(x). *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/62972-getcv-x) [Retrieved July 22, 2020].

Robust_CV_RoCV

'Robust' statistical methods are more resistant to the effects of outliers than the classical ones. An example would be using the median instead of the mean. The CV, like the mean, is sensitive to outliers, so for data with outliers, or that are not normally distributed, the robust CV (RoCV) can be used. There are at least two ways of estimating this, based on the IQR and the 'median absolute deviation' (MAD), respectively (Arachchige et al. 2019). The latter method is implemented in CEPS.

If m = median of a dataset xi, that contains n data points, then:

MAD = median of |xi – m| for i = 1, 2, …. n

and

RoCV = 1.4826 x MAD/m.

The multiplier 1.4826 adjusts the resulting RoCV so that it is the same as the CV for normally distributed data.

This function, like others in the GUI, could be applied not only to physiological series data, but to the epoched results of using some of the complexity and entropy measures implemented in the GUI, which will themselves be time series data.

Simple Linear Measures

Length

This is simply the number of data points in a particular dataset or epoch. It may be useful if you wish to check the lengths of multi-column data quickly when these are not all the same.

Slope

The purpose of the function is to estimate whether dataset or epoch values are increasing, decreasing or remaining constant overall (**Figure 12**).

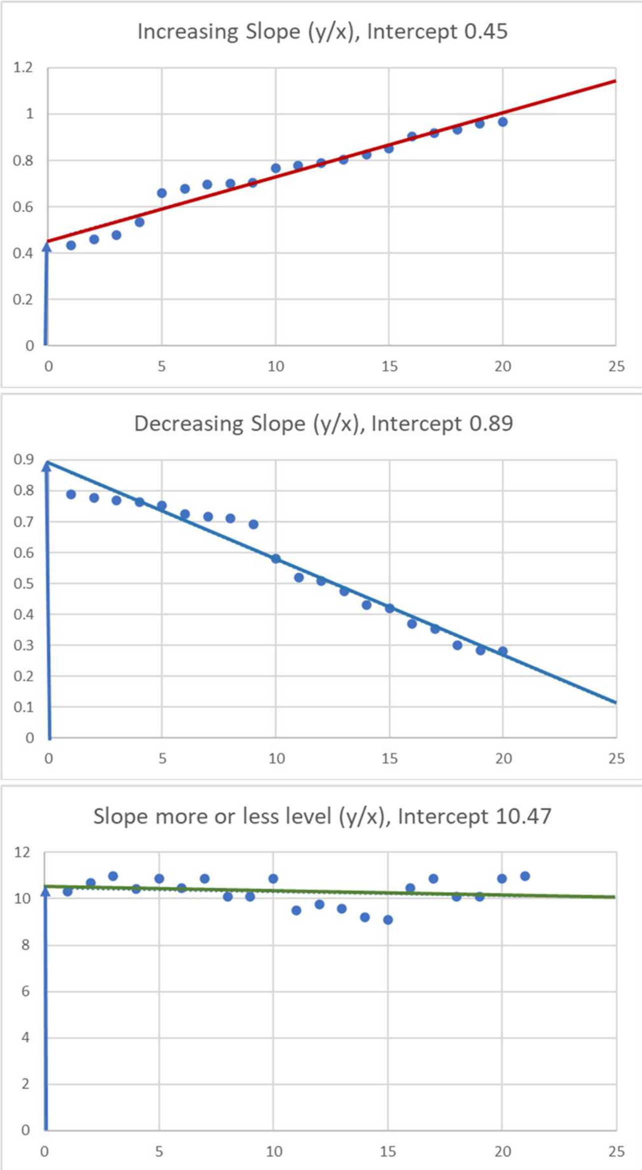

Figure 12. Slope examples, with intercepts.

The slope is calculated from the linear regression equation.

Related slope functions are RobustSlope and RobustIntercept.

Intercept

Like the Slope, the Intercept on the y-axis is calculated from the linear regression equation (see **Figure 12**).

RobustSlope and RobustIntercept

For many datasets, the underlying assumptions of the usual linear regression method are violated, and methods that are robust to nonlinearity, outliers and 'heteroscedasticity' (changes in variance) are preferable. Just as RoCV is more robust to outliers than CV, robust regression methods can be used to provide measures of Slope and Intercept that may be more widely useful with real-world physiological data than those of the usual linear regression method.

RobustSlope and RobustIntercept are implemented in the GUI using the built-in MATLAB function robustfit.

Normality Tests

'Parametric' statistical methods of analysis require data to be distributed in a predictable way that can be described using a fixed set of parameters. Normal (or Gaussian) distribution can be described, for example, in terms of the data's mean and standard deviation. For data that cannot be so described, or for very small samples, 'non-parametric' methods have to be used. In that case, the data's median and IQR may be more appropriate measures. It is therefore important to conduct some normality tests before deciding which further statistical measures and methods to use. The major tests included here are the Kolmogorov-Smirnov and Shapiro-Wilk tests (Yap & Sim 2011). The former is often considered more appropriate for larger datasets (a few hundred or more), the latter when sample size is 50 or less (but not less than 3). The Shapiro-Wilk test can also be used with larger datasets, but does not work well if there are many tied values in the data. Two simpler tests based on skewness and kurtosis are also included, although they are less rigorous. Graphical methods to test normality have not been implemented in the GUI, as they are tedious to use when testing many datasets at the same time.

Skewness (Sk)

If the probability distribution of a set of numbers is symmetrical, then its mean and median are the same. When the distribution is not symmetrical, this is not the case, and the data are said to be skewed. Skewness (Sk), or asymmetry of distribution, may be positive (right-skewed), negative (left-skewed) or zero. For normally distributed data, skewness will be close to zero.

Skewness can be measured in various ways. Here we use the built-in MATLAB function for Pearson's moment coefficient of skewness. This treats NaNs as missing values, and removes them.

StdErrSkewness (SE_Sk)

The standard error of skewness (StdErrSkewness) is a function only of sample size, regardless of the value of skewness itself. The standard error of Skewness is also a built-in MATLAB function.

In SPSS, the squared standard error of the skewness statistic is calculated as:

V_skew = 6*N*(N-1) / ((N-2)*(N+1)*(N+3)),

where N is the sample size (IBM Support 78312). This is the equation used in CEPS.

IsSkewness

IsSkewness is defined on the basis of the ratio of the absolute (signless) value of Skewness divided by its standard error. If |Sk/SE_Sk| < 1.96, skewness of a dataset is considered close enough to zero to suggest that the data are likely to be normally distributed (Abu-Bader 2010, pp. 63-64), although normal distribution is not guaranteed.

The value of IsSkewness will be 1 (TRUE) if |Sk/SE_Sk| < 1.96, and 0 (FALSE) if ≥ 1.96.

Kurtosis (K)

Kurtosis is a measure of the length of the ‘tails’ of a distribution peak. Kurtosis of a simple normal distribution is 3 (not, as for skewness, 0). Kurtosis > 3 corresponds to more outliers being further from the peak, rather than the sharpness of the peak itself, while if kurtosis is < 3, outliers will be fewer and less extreme. Here we use the built-in MATLAB function to quantify the kurtosis of a sample (dataset or epoch).

StdErrKurtosis (SE_K)

The standard error of kurtosis (SE_K) is a function only of sample size, regardless of the value of skewness itself. The standard error of K is also a built-in MATLAB function.

In SPSS, the squared standard error of the kurtosis statistic is calculated as:

V_kur = 4*(N^2-1)*V_skew / ((N-3)*(N+5)),

where N is the sample size or epoch length (IBM Support 78312). This is the equation used in CEPS.

IsKurtosis

IsKurtosis is defined on the basis of the ratio of the absolute (signless) value of Kurtosis divided by its standard error. If $|K/SE\_K| < 1.96$, the kurtosis of a dataset is considered close enough to 3 to suggest that the data are likely to be normally distributed (Abu-Bader 2010, pp. 63-64), although normal distribution is not guaranteed.

The value of IsKurtosis will be 1 (TRUE) if $|K/SE\_K| < 1.96$, and 0 (FALSE) if ≥ 1.96.

KolmogorovSmirnovTest

The single-sample KSTEST is a built-in MATLAB function (implemented without the Lilliefors correction). The null hypothesis tested is that the data comes from a standard normal distribution rather than a non-normal distribution. The result is 1 if the test rejects the null hypothesis at the 5% significance level, or 0 otherwise. In other words, if significance is < 0.05, then data is *not* normally distributed, and the result is 1.

Parameter settings

Default significance level (alpha) is 0.05. If you wish to be less rigorous in your acceptance of data as normally distributed, this value may be reduced. Use the KolmogorovSmirnovTest rather than the ShapiroWilkTest if sample size > 5000.

ShapiroWilkTest

In the implementation of the SWTEST function in CEPS, when the data series is 'leptokurtic' (i.e. with kurtosis > 3), SWTEST performs the Shapiro–Francia rather than the Shapiro–Wilk test. As for the Kolmogorov–Smirnov test, the result is 1 if the test rejects the null hypothesis at the 5% significance level, or 0 otherwise. In other words, if significance is < 0.05, then data is *not* normally distributed, and the result is 1.

MATLAB: BenSaïda A. (2020). Shapiro-Wilk and Shapiro-Francia normality tests. *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/13964-shapiro-wilk-and-shapiro-francia-normality-tests), MATLAB Central File Exchange) [Retrieved July 23, 2020].

Data requirements

Sample size can be between a minimum of 3 and a maximum of 5000, but results may not be accurate for *n* < 25.

RMS

The root-mean-square (RMS) amplitude of a set of time-series data points, used over successive epochs (or 'windows') of the data, can provide a smoothed estimate of the data's amplitude. RMS is a built-in MATLAB function.

> **Caution**
>
> Smoothing data may affect the values of some Complexity and Entropy measures.

RMSSD

The root-mean-square of successive differences (RMSSD) is used to analyse *intervals* between peaks (rather than their amplitude) or zero-line crossings in periodically or irregularly repeating data such as the heart beat or respiration (here, 'SD' does not refer to 'standard deviation' but to 'successive differences'). RMSSD is commonly used in heart rate variability (HRV) research, but can also be used in respiration analysis (Soni & Muniyandi 2019).

> **Data requirements**
>
> RMSSD is one of the most robust of HRV measures to noise and missing RR intervals (Baek & Shin 2017). However, RMSSD in HRV data is affected by sampling rate (Durosier *et al.* 2014), with some researchers recommending a sampling rate of 1000 Hz (Hejjel & Roth 2004). RMSSD in pulse rate variability (PRV) data is less affected by sampling rates provided these are > 25 Hz (Choi & Shin 2017).

The Hjorth parameters

The Swedish researcher Bo Hjorth developed these in the late 1960s for application in EEG analysis (Hjorth 1970). They have also been used in heart rate, EMG, tremor and respiration analysis (Spiridou *et al.* 2017; Mouzé-Amady & Horwat 1996; Yao *et al.* 2020; Singh *et al* 2018). However, application of these time domain measures to signals with strong frequency domain characteristics like the EEG has been questioned (Zhao *et al.* 2019).

MATLAB: Code for the Hjorth parameters was provided by Firgan Ferodov, although it is also available in other MATLAB toolkits.

4.4.3.1. HjorthActivity

Hjorth Activity is, very simply, the square of the standard deviation of amplitude, or variance, *var* (equivalent to mean power in the EEG) (Hjorth 1970).

Data requirements

Used mostly with EEG data, but also ECG and photoplethysmogram (PPG) data. May be used with continuous or discrete data (Deshmane 2009). Has been used with EEG sub-band data obtained with a tunable Q wavelet transform (Dash & Kolekar 2020). Feature classification accuracy may improve with bandpass filtering (Oh *et al.* 2014). The effects of noise and sampling rate, as well as the nonstationarity and nonlinearity of data, require further investigation.

Parameter setting

No parameter is required for this measure.

Reported values – some examples

EEG c. 50 (N = 20, Spehr *et al.* 1977)
EEG c. 1700 to 1900 (N = 22, Kamousi *et al.* 2019)

HjorthMobility

Hjorth Mobility is a measure of the standard deviation (SD) of the slope of the signal (*dy/dt*) divided by the SD of the signal amplitude. It is expressed as a ratio per time unit (1/*t*), and so may be thought of as the mean or central frequency of the signal (Hjorth 1970; Spehr *et al.* 1977):

$$\text{Mobility} = \sqrt{\frac{\text{var}(\frac{dy(t)}{dt})}{\text{var}(y(t))}}.$$

Data requirements

Used mostly with EEG data, but also ECG and photoplethysmogram (PPG) data. May be used with continuous or discrete data (Deshmane 2009). Has been used with EEG sub-band data obtained with a tunable Q wavelet transform (Dash & Kolekar 2020). Feature classification accuracy may improve with bandpass filtering (Oh *et al.* 2014). The effects of noise and sampling rate, as well as the nonstationarity and nonlinearity of data, require further investigation.

Parameter setting

No parameter is required for this measure.

Reported values – some examples

EEG c. 7.4 to 7.9 (*N* = 20, Spehr *et al.* 1977)
EEG c. 0.6 to 0.7 (*N* = 22, Kamousi *et al.* 2019)

HjorthComplexity

Hjorth Complexity, or 'Complexity of the first order' (Hjorth 1973), quantifies deviation of a signal's shape from a simple sine wave, with values further from unity (1) indicating greater complexity (Hjorth 1970). Mathematically, it is the ratio of the mobility of the slope of the signal (*dy/dt*), divided by the slope of the signal amplitude:

$$\text{Complexity} = \frac{\text{Mobility}(\frac{dy(t)}{dt})}{\text{Mobility}(y(t))}.$$

Hjorth Complexity is a linear measure, not to be confused with the nonlinear Complexity measures implemented in CEPS.

Data requirements

Used mostly with EEG data, but also ECG and photoplethysmogram (PPG) data. May be used with continuous or discrete data (Deshmane 2009). Has been used with EEG sub-band data obtained with a tunable Q wavelet transform (Dash & Kolekar 2020). Feature classification accuracy may improve with bandpass filtering (Oh *et al.* 2014). The effects of noise and sampling rate, as well as the nonstationarity and nonlinearity of data, require further investigation.

Parameter setting

No parameter is required for this measure.

Reported values – some examples

Foetal heart rate c. 1.4 (*N*=92, Spyridou *et al.* 2017)

EEG c. 7.4 to 7.9 (*N* = 20, Spehr *et al.* 1977)
EEG c. 0.6 to 0.7 (*N* = 22, Kamousi *et al.* 2019)

Frequency Domain Measures

According to Fourier's theorem, any 'reasonably well-behaved' steady-state or periodic mathematical function can be considered as the sum of a series of sine and cosine functions with different frequencies and phases (Morin 2009). Together, these components that make up the periodic function form its spectrum. Spectral analysis can be used to determine the components of the periodic function, with their frequency, phase and amplitude (or power). As an extension of this approach, even non-periodic signals can be broken down into components of different frequencies, some showing greater amplitude or power than others. Various methods can be used to do this and produce a 'periodogram' or estimate of the power spectral density of a data series. Welch's method is one, and the short-time Fourier transform (STFT) is another. There are inherent uncertainties with both methods: Welch's periodogram sacrifices frequency resolution in favour of noise reduction, while for the STFT either frequency or location in time can be estimated, but not both.

All the methods mentioned can be applied not only to time series data, but just as well to data consisting of measurements in physical space or in other, more abstract domains.

Here we have chosen to implement two methods, a version of the Lomb–Scargle periodogram and the standard fast Fourier transform (FFT).

Lomb–Scargle periodogram (FastLomb)

This approach is more able to cope with noisy sets of unequally spaced data than FFT.

MATLAB: Saragiotis C. (2020). Lomb normalized periodogram. *MATLAB Central File Exchange* (https://www.mathworks.com/matlabcentral/fileexchange/22215-lomb-normalized-periodogram) [Retrieved August 30, 2020].

This has not yet been completely implemented in CEPS.

Data requirements

For accurate results, data must exhibit approximate linearity and weak-sense stationarity (see below).

Resampling and replacement of outliers is not required for this method, which is also quite robust to noise (Moody 1993).

Parameter setting

A default value of 4 for the oversampling factor is selected in the software.

Reported values – some examples

[To be completed]

The use of plots

[To be completed]

Fast Fourier Transform (FFT)

The built-in MATLAB function fft is implemented in this GUI for the purpose of spectral analysis.

Implementation has not been finalised.

Data requirements

For accurate results, data must exhibit approximate linearity and weak-sense stationarity (see below). Noisy data may not give clear results, and downsampling is likely to attenuate high frequency results (Moody 1993).

In principle, the fft function computes the discrete Fourier transform for a data series with 2^m points, where m is an integer. If the number of data points n is between 2^{m-1} and 2^m, then fft ignores the remaining signal values past the nth entry and returns a truncated result.[6]

Parameter setting

No parameters are required for this method.

Reported values – some examples

[To be completed]

The use of plots

[To be completed]

Measures of stationarity and nonlinearity

Nonlinearity and stationarity are briefly explained in the CEPS Primer. Many Complexity and Entropy measures in CEPS require data to be at least weak-sense stationarity if results are to be useful, as indicated in **Table 1** above. In CEPS, data can be tested for stationarity using the Mean, SD, AutoCovariance and AutoCorrelation functions, or ‘Reverse Arrangement’ and ‘Moving Window’ tests (Cheynet 2020). However, the results of these tests are not necessarily easy to interpret.

Cheynet E. (2020). Stationarity test. GitHub. (https://github.com/ECheynet/stationaryTests/releases/tag/v1.2.1) [Retrieved July 27, 2020].

[6] Other options are possible when analysing a discrete time series whose length is not an integer power of 2 (Kirsch *et al.* 2012), but these are not implemented here.

We have not included more elaborate methods, such as the Dickey-Fuller test (Dickey et al. 1979). An overview of some of these tests in simple language may be found online (Palachy 2019). A more down-to-earth method, although not necessarily very precise, is simply to inspect the data visually for weak-sense stationarity (Faes et al. 2019b).

If your data are not stationary, you could also check for Slope or RobustSlope to see if detrending may be required – although not all measures of complexity and entropy require stationarity of data (see below). Transforming the data by 'differencing' the data series (computing the differences between consecutive data points), or using other methods to create a new data series, may help improve stationarity. Dividing the data into shorter epochs, for instance, may be helpful (e.g. Raghu et al. 2017).

AutoCorrelation (ACR)

Autocorrelation quantifies the degree of similarity between a given time series and a lagged version of itself over successive time intervals, using Pearson's *R*, so results may be between +1 and −1.

Stronger autocorrelation ('memory', or reduced variability over time) in some physiological measures 'can be a disadvantage for an adaptive system' (Bottaro et al. 2020), and may be associated with greater disease severity (Satti et al. 2019).

MATLAB code used here is by Calvin Price: Price C. (2020). Autocorrelation Function (ACF). *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/30540-autocorrelation-function-acf) [Retrieved December 29, 2020].

Data requirements

This function may be used with both continuous and discrete data. For accurate results, data should exhibit stationarity and approximate linearity

Parameter setting

The number of lags must be stipulated. [To be completed]

Reported values – some examples

[To be completed]

The use of plots

'Correlograms' of autocorrelation against lag can be viewed on many internet webpages.

AutoCovariance (ACV)

Autocovariance quantifies similarity between the variance of a time series and a lagged version of itself. It is closely related to autocorrelation (sometimes even considered identical), but is far less commonly used to assess nonstationarity.

MATLAB code used here is based on that in the Dalhousie University Statistics course lecture notes: Dowd M, Thompson K, Smith B. (2006). Section 3: Auto-covariance and Auto-correlation. *Statistics course (STAT 4390/5390) lecture notes*. Halifax, NS: Dalhousie University (https://www.mathstat.dal.ca/~stat5390/Section_3_ACF.pdf) [Retrieved July 27, 2020].

Data requirements

This function may be used with both continuous and discrete data (for a discussion, see Mobley 2020). It is robust to noise (Moore *et al.* 2020).

Parameter setting

Lag k must be < n, the length of the data series.

Reported values – some examples

[To be completed]

The use of plots

In a study on Correlation dimension in HRV, Carvajal *et al.* (2005) include a plot of autocovariance against lag τ, estimating τ from the first relative minimum of the autocovariance function, as in the FNN method.

'Covariograms' of autocovariance against lag or distance can be viewed on many internet webpages.

Reverse Arrangement Test 1 for Nonstationarity

This 'reverse arrangement' test for nonstationarity is based on a method described in a standard textbook (Bendat & Piersol 2010).

Reverse Arrangement Test 2 for Nonstationarity

This alternative reverse arrangement test for nonstationarity is based on a method described by Siegel (Siegel & Castellan 1988).

Moving Window Test for Nonstationarity

This parametric moving window test for nonstationarity is based on a method described by Priestley (Priestley 1982). The instantaneous mean (or standard deviation) of the data is compared to the one obtained without any detrending.

MATLAB implementations of these three tests are by Etienne Cheynet: Cheynet E. (2020). Stationarity test. *GitHub*. (https://github.com/ECheynet/stationaryTests/releases/tag/v1.2.1) [Retrieved July 27, 2020].

Data requirements

These three functions may be used with both continuous and discrete data. They are not robust to noise (Cappa et al. 2001), and may not be sensitive to small changes in stationarity. The moving window test has been used with nonlinear data (Jones 2012).

Parameter setting

Three parameters are required for the Moving Window Test:

Window length will need to be selected according to the nature of the data and the stringency of your requirements for stationarity.

Threshold 1 ($t1$) is the acceptable value for the relative error between the instantaneous mean and the static mean of the dataset. Any value above $t1$ will classify the time series as non-stationary.

Threshold 2 ($t2$) is the acceptable value for the relative error between the instantaneous standard deviation (SD) and the static SD of the dataset. Any value above $t2$ will classify the time series as non-stationary.

Reported values – some examples

0 (if data not stationary), 1 (if data stationary).

The use of plots

Plots are not used for these particular tests.

Nonlinearity (VM)*

This measure is described in the CEPS Primer.

MATLAB and Fortran codes made available by Pedro Bernaola-Galván for use in CEPS, with permission.

COMPLEXITY MEASURES

Higuchi's Fractal Dimension (HFD)*

MATLAB: Monge-Álvarez J. (2020a). Higuchi and Katz fractal dimension measures. *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/50290-higuchi-and-katz-fractal-dimension-measures) [Retrieved July 29, 2020].

Allan Factor (AF)*

C++ code provided for use in CEPS, with permission, by David Cornforth, but not yet implemented.

Correlation Dimension (D_2)*

MATLAB: Code implemented is from the MATLAB Predictive Maintenance Toolbox, following on from an earlier efficient algorithm by Theiler (Theiler 1987). In the near future, we plan to implement other code for D_2 by Faranda and Vaienti (2018).

Hurst Exponent (H)*

MATLAB: Davidson B. (2020). Hurst exponent. *MATLAB Central File Exchange*. (https://www.mathworks.com/matlabcentral/fileexchange/9842-hurst-exponent) [Retrieved August 6, 2020].

Detrended Fluctuation Analysis (DFA)*

If you wish to calculate two scaling exponents, short–term $\alpha1$ and long–term exponent $\alpha2$, as in HRV, insert the first and last data points for each range considered. If you wish to calculate a single scaling exponent α, insert the first datapoint for Range 1 and the last datapoint for Range 2.

MATLAB: Magris M. (2020). Detrended fluctuation analysis (DFA) *MATLAB Central File Exchange*. (https://www.mathworks.com/matlabcentral/fileexchange/67889-detrended-fluctuation-analysis-dfa) [Retrieved August 3, 2020]. Not yet implemented.

Multifractal Multiscale Detrended Fluctuation Analysis (mFmDFA)*

MATLAB: Code from Castiglioni P, Faini A, (2019). A fast DFA algorithm for multifractal multiscale analysis of physiological time series. *Frontiers in Physiology* 10, 115. To be implemented, in consultation with the authors.

Largest Lyapunov Exponent (LLE)*

MATLAB: Kizilkaya M. (2020a). Largest Lyapunov exponent with Rosenstein's algorithm. *MATLAB Central File Exchange*. (https://www.mathworks.com/matlabcentral/fileexchange/38424-largest-lyapunov-exponent-with-rosenstein-s-algorithm) [Retrieved August 3, 2020].

This code has been used before in a published study (Samson *et al.* 2017) and is based on a method by Rosenstein *et al.* (1993) that is suitable for shorter datasets.

Recurrence Quantification Analysis (RQA)*

MATLAB: Yang H. (2020). Tool box of recurrence plot and recurrence quantification analysis. *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/58246-tool-box-of-recurrence-plot-and-recurrence-quantification-analysis) [Retrieved August 4, 2020].

This toolbox has been used in a number of published studies (e.g. Nayak *et al.* 2018). It includes functions for the AMI method of estimating τ and the FNN method of determining m, with threshold ε chosen as 5% of the maximal distance in phase space (Schinkel *et al.* 2008; Chen & Yang 2012).

The Poincaré plot (PP) and Extended Poincaré plot (EPP)*

MATLAB: Code is provided in the supplementary material to Satti *et al.* (2019). It is used here with permission.

The Complex Correlation Measure (CCM)*

C++ code provided for use in CEPS, with permission, by David Cornforth, but not yet fully tested.

Lempel-Ziv Complexity (LZC)*

MATLAB: Thai Q. (2020). calc_lz_complexity. *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/38211-calc_lz_complexity) [Retrieved August 25, 2020].

Multiscale Lempel-Ziv Complexity (mLZC)*

MATLAB: Code provided for use in CEPS, with permission, by Antonio Ibáñez.

SYMBOLIC DYNAMICS (SymDyn)*

MATLAB code not yet implemented.

ENTROPIES

Shannon and Generalised Entropies

Shannon Entropy (SE)*

MATLAB: Monge-Álvarez J. (2020b). A set of Entropy measures for temporal series (1D signals). *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/50289-a-set-of-entropy-measures-for-temporal-series-1d-signals) [Retrieved August 8, 2020].

This code was used as it was designed to enable calculation of CCE and CE. Data is normalised before SE is computed.

MATLAB: An alternative code is implemented for comparison, from Chen M. (2020*). Pattern Recognition and Machine Learning Toolbox*. (https://github.com/PRML/PRMLT), GitHub. Retrieved November 14, 2020.

Rényi Entropy (RE)*

MATLAB: Guan WY. (2020). *Shannon and non-extensive entropy*. (https://www.mathworks.com/matlabcentral/fileexchange/18133-shannon-and-non-extensive-entropy), MATLAB Central File Exchange. Retrieved November 14, 2020.

Tsallis Entropy (TE)*

MATLAB: Guan WY. (2020). *Shannon and non-extensive entropy*. (https://www.mathworks.com/matlabcentral/fileexchange/18133-shannon-and-non-extensive-entropy), MATLAB Central File Exchange. Retrieved November 14, 2020.

Guan Wenye’s codes for RE and TE have been implemented in CEPS, but should be used with caution, as results are not consistent with those for SE when computed using the same toolbox. We plan to replace codes for these two entropies in future versions of CEPS.

Diffusion Entropy (DnEn)*

Code for DnEn is not yet implemented in CEPS.

Further developments from Shannon Entropy

Entropy of Entropy (EoE)*

MATLAB: Code provided for use in CEPS, with permission, by Chang Stephen Hsu.

Average Entropy (AE)*

MATLAB: Code provided for use in CEPS, with permission, by Chang Stephen Hsu.

Tone-Entropy (T-E)*

MATLAB: Code written by Deepak Panday, with amendments by Chandan Karmakar.

Entropy of Difference (EoD)*

Mathematica code provided for use in CEPS, with permission, by Pasquale Nardone, translated into MATLAB by Deepak Panday.

Kullbach-Leibler Divergence (KLD)*

Mathematica code provided for use in CEPS, with permission, by Pasquale Nardone, translated into MATLAB by Deepak Panday.

Ordinal Entropies

Multiscale Permutation Entropy (mPE)*

MATLAB: Azami H, Escudero J. (2015). Matlab codes for "Improved Multiscale Permutation Entropy for Biomedical Signal Analysis: Interpretation and Application to Electroencephalogram Recordings". *University of Edinburgh, School of Engineering, Institute for Digital Communications*. (https://doi.org/10.7488/ds/273) [Retrieved August 9, 2020].

Combined with: Ouyang GX. (2020). Multiscale Permutation Entropy (MPE). *MATLAB Central File Exchange* (https://www.mathworks.com/matlabcentral/fileexchange/37288-multiscale-permutation-entropy-mpe) [Retrieved December 22, 2020].

Amplitude-Aware Permutation Entropy (AAPE)*

MATLAB: Azami H, Escudero J. (2016). Matlab codes for "Amplitude-aware Permutation Entropy: Illustration in Spike Detection and Signal Segmentation", [software]. *University of Edinburgh. School of Engineering, Institute for Digital Communications.* (https://doi.org/10.7488/ds/1339) [Retrieved November 14, 2020].

Note that we have corrected this code in accordance with advice received from David Cuesta-Frau (26 February 2021), inserting some missing brackets.

Improved Multiscale Permutation Entropy (ImPE)*

MATLAB: Azami H, Escudero J. (2015). Matlab codes for "Improved Multiscale Permutation Entropy for Biomedical Signal Analysis: Interpretation and Application to Electroencephalogram Recordings". *University of Edinburgh, School of Engineering, Institute for Digital Communications*. (https://doi.org/10.7488/ds/273) [Retrieved August 9, 2020].

Multiscale Permutation Min-Entropy (mPM-E)*

MATLAB: Code provided for use in CEPS, with permission, by Luciano Zunino.

Conditional Entropies

Conditional Entropy (CE)*

MATLAB: Monge-Álvarez J. (2020b). A set of Entropy measures for temporal series (1D signals). *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/50289-a-set-of-entropy-measures-for-temporal-series-1d-signals) [Retrieved August 8, 2020].

Corrected Conditional Entropy (CCE)*

MATLAB: Monge-Álvarez J. (2020b). A set of Entropy measures for temporal series (1D signals). *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/50289-a-set-of-entropy-measures-for-temporal-series-1d-signals) [Retrieved August 8, 2020].

Approximate Entropy (ApEn)*

MATLAB: Monge-Álvarez J. (2020b). A set of Entropy measures for temporal series (1D signals). *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/50289-a-set-of-entropy-measures-for-temporal-series-1d-signals) [Retrieved August 8, 2020].

Sample Entropy (SampEn)*

Three codes have been implemented in CEPS, for comparative purposes:

MATLAB: Azami H, Escudero J. (2016). Matlab codes for "Refined Multiscale Fuzzy Entropy based on Standard Deviation for Biomedical Signal Analysis", [software]. *University of Edinburgh, School of Engineering, Institute for Digital Communications*. (https://doi.org/10.7488/ds/1477) [Retrieved November 14, 2020].

MATLAB: Monge-Álvarez J. (2020b). A set of Entropy measures for temporal series (1D signals). *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/50289-a-set-of-entropy-measures-for-temporal-series-1d-signals) [Retrieved August 8, 2020].

MATLAB: Martínez-Cagigal V. (2018). Sample Entropy. *MATLAB Central File Exchange.* (https://uk.mathworks.com/matlabcentral/fileexchange/69381-sample-entropy) [Retrieved November 14, 2020].

Coefficient of Sample Entropy (CosEn)* and Quadratic SampEn (QSE)*

These codes are not yet implemented.

Fixed Sample Entropy (fSampEn)*

MATLAB: Code provided for use in CEPS, with permission, by Luis Estrada Petrocelli.

Multiscale Entropy (mSE)*

MATLAB Codes for three different versions of MSE have been implemented in CEPS, for comparative purposes (mSEμ, MSEσ and MSEσ^2):

Azami H, Escudero J. (2016). Matlab codes for "Refined Multiscale Fuzzy Entropy based on Standard Deviation for Biomedical Signal Analysis", [software]. *University of Edinburgh, School of Engineering, Institute for Digital Communications*. (https://doi.org/10.7488/ds/1477) [Retrieved November 14, 2020].

Complexity Index (CI)* and Multiscale Slope (mSlope)*

These codes are not yet implemented.

Fuzzy Entropy (FE)*

MATLAB: Azami H. (2019). Natlab codes for "Fuzzy entropy metrics for the analysis of biomedical signals: assessment and comparison, *IEEE ACCESS*, 2019". *GitHub.* (https://github.com/HamedAzami/FuzzyEntropy_Matlab) [[Retrieved November 14, 2020].

Other Entropies

Refined Composite Multiscale Sample Entropy based on standard deviation (RCmSEσ)*

MATLAB: Azami H, Escudero J. (2016). Matlab codes for "Refined Multiscale Fuzzy Entropy based on Standard Deviation for Biomedical Signal Analysis", [software]. *University of Edinburgh, School of Engineering, Institute for Digital Communications*. (https://doi.org/10.7488/ds/1477) [Retrieved November 14, 2020].

Refined Composite Multiscale Fuzzy Entropy based on standard deviation (RCmFEσ)*

MATLAB: Azami H, Escudero J. (2016). Matlab codes for "Refined Multiscale Fuzzy Entropy based on Standard Deviation for Biomedical Signal Analysis", [software]. *University of Edinburgh, School of Engineering, Institute for Digital Communications*. (https://doi.org/10.7488/ds/1477) [Retrieved November 14, 2020].

Refined Composite Multiscale Dispersion Entropy (RCmDE)*

MATLAB: Azami H, Escudero J. (2017). Matlab codes for "Refined Composite Multiscale Dispersion Entropy and its Application to Biomedical Signals", [dataset]. *University of Edinburgh, School of Engineering, Institute for Digital Communications*. (https://doi.org/10.7488/ds/1982) [Retrieved November 14, 2020].

Distribution Entropy (DistEn)*

MATLAB: Code provided for use in CEPS, with permission, by Peng Li.

Slope Entropy (SlopeEn)*

MATLAB: Code provided for use in CEPS, with permission, by David Cuesta-Frau.

Bubble Entropy (BE)*

PYTHON code provided for use in CEPS, with permission, by George Manis.

Phase Entropy (PhEn)*

MATLAB: Code written by Deepak Panday on the basis of pseudo-code in Rohila & Sharma (2019), and then corrected by Ashish Rohila for use in CEPS, with permission.

Entropies for time-frequency domain analysis

Spectral Entropy (SpEn)*

The code is not yet fully implemented.

Differential Entropy (DiffEn)*

This code is not yet implemented.

Ancillary Measures

Auto-Mutual Information (AMI)*

MATLAB: Yang H. (2020). Tool box of recurrence plot and recurrence quantification analysis. *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/58246-tool-box-of-recurrence-plot-and-recurrence-quantification-analysis) [Retrieved August 4, 2020].

False Nearest Neighbours (FNN)*

MATLAB: Yang H. (2020). Tool box of recurrence plot and recurrence quantification analysis. *MATLAB Central File Exchange.* (https://www.mathworks.com/matlabcentral/fileexchange/58246-tool-box-of-recurrence-plot-and-recurrence-quantification-analysis) [Retrieved August 4, 2020].

Averaged False Neighbours (AFN)*

MATLAB: Kizilkaya M. (2020). Minimum embedding dimension. *MATLAB Central File Exchange*. (https://www.mathworks.com/matlabcentral/fileexchange/36935-minimum-embedding-dimension), [Retrieved August 7, 2020].

A NOTE ON ABBREVIATIONS

Lists of abbreviations may be found in the CEPS Primer and the paper on CEPS published in *Entropy*.

REFERENCES[7]

Abu-Bader SH. (2010). *Advanced and Multivariate Statistical Methods for Social Science Research, with a Complete SPSS Guide*. Chicago, IL: Lyceum.

Arachchige CNPG, Prendergast LA, Stuadte RG. (2019). Robust analogues to the coefficient of variation. *arxiv preprint arxiv:*1907.01110.

Baek HJ, Shin J. (2017). Effect of missing inter-beat interval data on heart rate variability analysis using wrist-worn wearables. *Journal of Medical Systems* **41**(10), 147.

Bendat JS, Allan G. Piersol AG. (2010). *Random Data: Analysis and Measurement Procedures* (1st edn.). New York: John Wiley & Sons, p. 99.

Cappa P, Silvestri S, Sciuto SA. (2001). On the robust utilization of non-parametric tests for evaluation of combined cyclical and monotonic drift. *Measurement Science and Technology* **12**(9), 67-72.

Choi A, Shin H. (2017). Photoplethysmography sampling frequency: pilot assessment of how low can we go to analyze pulse rate variability with reliability? *Physiological Measurement* **38**(3):586-600.

Dash DP, H Kolekar M. (2020). Hidden Markov model based epileptic seizure detection using tunable Q wavelet transform. *Journal of Biomedical Research* **34**(3), 170-179.

Deshmane AV. (2009). *False Arrhythmia Alarm Suppression Using ECG, ABP, and Photoplethysmogram*. Thesis in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical Engineering and Computer Science, Massachusetts Institute of Technology (lcp.mit.edu/pdf/DeshmaneThesis09.pdf) [Retrieved August 13, 2020].

Dickey DA, Fuller WA. (1979). Distribution of the estimators for autoregressive time series with a unit root*. Journal of the American Statistical Association* **74**(366), 427-431.

[7] Only References not included in the CEPS Primer are listed here.

Durosier LD, Green G, Batkin I *et al.* (2014). Sampling rate of heart rate variability impacts the ability to detect acidemia in ovine fetuses near-term. *Frontiers in Pediatrics* **2**, 38.

Faranda D, Vaienti S. (2018). Correlation dimension and phase space contraction via extreme value theory. *Chaos* **28**, 041103, 1–15.

Hejjel L, Roth E. (2004). What is the adequate sampling interval of the ECG signal for heart rate variability analysis in the time domain? *Physiological Measurement* **25**(6):1405–1411.

Hjorth B. (1970). EEG analysis based on time domain properties. *Electroencephalography and Clinical Neurophysiology* **29**(3), 306–310.

IBM Support. (2020). Standard errors of skewness and kurtosis are all the same for a set of variables. *IBM Support*, historical number 78312. (https://www.ibm.com/support/pages/node/421769) [Retrieved July 22, 2020].

Jones RN. (2012). Detecting and attributing nonlinear anthropogenic regional warming in southeastern Australia. *Journal of Geophysical Research* **117**, D04105, 1–13.

Kamousi B, Grant AM, Bachelder B *et al.* (2019). Comparing the quality of signals recorded with a rapid response EEG and conventional clinical EEG systems. *Clinical Neurophysiology Practice* **4**, 69–75.

Mobley C. (2020). Autocovariance Functions: Theory. In: *Ocean Optics Web Book* (https://oceanopticsbook.info/view/surfaces/level-2/autocovariance-functions-theory) [Retrieved August 14, 2020].

Moody GB. (1993). Spectral analysis of heart rate without resampling. *Proceedings of Computers in Cardiology Conference*, London, 715–718.

Morin D. (2009). Fourier analysis (Ch. 3). *Waves (Draft).* Cambridge, MA: Harvard University. (https://scholar.harvard.edu/files/david-morin/files/waves_fourier.pdf) [Retrieved July 24, 2020].

Mouzé-Amady M, Horwat F. (1996). Evaluation of Hjorth parameters in forearm surface EMG analysis during an occupational repetitive task. *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control* **101**(2), 181-183.

Oh S-H, Lee Y-R, Kim H-N. (2014). A novel EEG feature extraction method using Hjorth parameter. *International Journal of Electronics and Electrical Engineering* **2**(2), 106-110.

Palachy S. (2019). Detecting stationarity in time series data. *Towards Data Science*. (https://towardsdatascience.com/detecting-stationarity-in-time-series-data-d29e0a21e638) [Retrieved July 27, 2020].

Priestley MB. (1981). *Spectral Analysis and Time Series*. NY: Academic Press.

Raghu S, Sriraam N, Pradeep Kumar G. (2017). Classification of epileptic seizures using wavelet packet log energy and norm entropies with recurrent Elman neural network classifier. *Cognitive Neurodynamics* **11**(1), 51-66.

Samson N, Praud J-P, Quenet GB *et al.* (2017). New insights into sucking, swallowing and breathing central generators: A complexity analysis of rhythmic motor behaviors. *Neuroscience Letters* **638**, 90-95.

Siegel S, Castellan NJ. (1988). *Nonparametric Statistics for the Behavioral Sciences.* NY: McGraw-Hill.

Singh OP, Howe TA, Malarvili MB. (2018). Real-time human respiration carbon dioxide measurement device for cardiorespiratory assessment. *Journal of Breath Research* **12**(2), 026003.

Soni R, Muniyandi M. (2019). Breath rate variability: a novel measure to study the meditation effects. *International Journal of Yoga* **12**(1), 45-54.

Spehr W, Sartorius H, Berglund K *et al.* (1977). EEG and haemodialysis. A structural survey of EEG spectral analysis, Hjorth's EEG descriptors, blood variables and psychological data. *Electroencephalography and Clinical Neurophysiology* **43**(6), 787-797.

Spyridou K, Chouvarda I, Hadjileontiadis L, *et al*. (2017). The effect of cigarette smoking on fetal heart rate tracing during pregnancy. *Journal of Perinatal Medicine* **45**(4), 403–411.

Theiler J. (1987). Efficient algorithm for estimating the correlation dimension from a set of discrete points. *Physical Review A* **36**(9), 44–56.

Tukey JW (1977). *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

Yao L, Brown P, Shoaran M. (2020). Improved detection of Parkinsonian resting tremor with feature engineering and Kalman filtering. *Clinical Neurophysiology* **131**(1), 274–284.

Yap BW, Sim CH. (2011). Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation* **81**(12), 2141–2155.

Zhao DC, Wang Y, Wang QQ *et al.* (2019). Comparative analysis of different characteristics of automatic sleep stages. *Computer Methods and Programs in Biomedicine* **175**, 53–72.