fractalRegression: An R package for multiscale regression and fractal analyses

Aaron D. Likens[1] & Travis J. Wiltshire[2]

[1] Department of Biomechanics, University of Nebraska at Omaha

[2] Department of Cognitive Science & Artificial Intelligence, Tilburg University

Author Note

Add complete departmental affiliations for each author here. Each new line herein must be indented, like this line. Enter author note here.

The authors made the following contributions. Aaron D. Likens: Conceptualization, Software, Writing - Original Draft Preparation, Writing - Review & Editing; Travis J. Wiltshire: Writing - Original Draft Preparation, Writing - Review & Editing, Software.

Correspondence concerning this article should be addressed to Aaron D. Likens, Postal address. E-mail: alikens@unomaha.edu

Abstract

Time series data from scientific fields as diverse as astrophysics, economics, human movement science, and neuroscience all exhibit fractal properties. That is, these time series often exhibit self-similarity and long-range correlations. This `fractalRegression` package implements a number of univariate and bivariate time series tools appropriate for analyzing noisy data exhibiting these properties. These methods, especially the bivariate tools (Kristoufek, 2015a; Likens, Amazeen, West, & Gibbons, 2019a) have yet to be implemented in an open source and complete package for the R Statistical Software environment. As both practitioners and developers of these methods, we expect these tools will be of interest to a wide audience of scientists who use R, especially those from fields such as the human movement, cognitive, and other behavioral sciences. The algorithms have been developed in C++ using the popular Rcpp (Eddelbuettel & Francois, 2011) and RcppArmadillo (Eddelbuettel & Sanderson, 2014) packages. The result is a collection of efficient functions that perform well even on long time series (e.g., $\geq$ 10,000 data points). In this work, we motivate introduce the package, each of the functions, and give examples of their use as well as issues to consider to correctly use these methods.

Keywords: long range correlation, fractal, multiscale, dynamics

Word count: X

fractalRegression: An R package for multiscale regression and fractal analyses

Introduction

Over time, many signals from living and complex systems exhibit systematic regularities and dependencies across spatial and temporal scales (**kello2010?**). These regularities often follow a power-law (i.e., self-similarity across scales) that are estimated using fractal analyses. Fractal analysis, in its many forms, has become an important framework in virtually every area of science, often serving as an indicator of system health (Goldberger et al., 2002), adaptability (Bak, Tang, & Wiesenfeld, 1987), control (Likens, Fine, Amazeen, & Amazeen, 2015), cognitive function (Euler, Wiltshire, Niermeyer, & Butner, 2016), and multi-scale interactions (Kelty-Stephen, 2017).

In particular, various methods related to Detrended Fluctuation Analysis (DFA) (Peng et al., 1994) have rose to prominence due to their relative ease of understanding and broad applicability to stationary and non-stationary time series, alike. More specifically, in areas of the social and cognitive sciences, DFA, or variants of DFA, have been used to study, for example, reaction times (Van Orden, Holden, & Turvey, 2003), eye gaze (Stephen, Boncoddo, Magnuson, & Dixon, 2009), gait (**delignières2009?**), limb movements (Delignières, Torre, & Lemoine, 2008), heart rate (Goldberger et al., 2002), and neurophysiological oscillations Euler et al. (2016). And, beyond an individual level, the methods have been used to study human-machine system interaction (Likens et al., 2015), tool use (**favela2021?**), and interpersonal coordination in a variety of modalities Delignières, Almurad, Roume, & Marmelat (2016).

Thus, there is a broad scientific appeal for these fractal-based analyses. While, the basic DFA algorithm has been implemented in numerous packages and software programs, more advanced methods such as Multifractal Detrended Fluctuation Analysis (MFDFA) (Kantelhardt et al., 2002), Detrended Cross Correlation (DCCA) (Podobnik & Stanley, 2008; Zebende, 2011), and, in particular, fractal regression techniques such as Multiscale

Regression Analysis (MRA) (Kristoufek, 2015a; Likens et al., 2019a) have not yet been implemented in a comprehensive software package. Thus, there is a clear need for a package that incorporates this functionality in order to advance theoretical research focused on understanding the time varying properties of natural phenomena and applied research that uses those insights in important areas such as healthcare (Cavanaugh, Kelty-Stephen, & Stergiou, 2017) and education (Snow, Likens, Allen, & McNamara, 2016). In this work, we provide an overview of our `fractalRegression` package, provide simulated and empirical examples of it's functions, and provide practical advice on the successful application of these methods.

Package Overview

Our `fractalRegression` package for R (Team, 2018) is built on a C++ architecture and includes a variety of uni- and bivariate fractal methods as well as functions for simulating data with known fractional properties (e.g., scaling, dependence, etc.), and surrogate testing. Some foundational efforts in fractal analyses, which partially overlap with the functionality of this package, have been implemented elsewhere. For example, a number of univariate fractal and multifractal analyses have been implemented in the 'fracLab' library for MATLAB (Legrand & Véhel, 2003) and other toolboxes that are mainly targeted at multifractal analysis (E. A. F. Ihlen, 2012; Espen A. F. Ihlen & Vereijken, 2010). In terms of open access packages, there are other packages that implement some, but not all of the same functions such as the `fathon` package (Bianchi, 2020) that has been implemented in Python as well as the R packages: `fractal` [defunct], `nonlinearTseries` (Garcia, 2020), and `MFDFA` (Laib, Golay, Telesca, & Kanevski, 2018). However, none of the above packages incorporate univariate monofractal and multifractal DFA with bivariate DCCA and MRA nor do they run on a C++ architecture. Our `fractalRegression` package is this unique in this combination of analyses and efficiency (particularly for long time series). For instance, we are not aware of any other packages that feature MRA and Multiscale Lagged Regression (MLRA). In addition, we expect that featuring simulation methods as well as surrogate testing strongly bolsters the accessibility of these methods for the social and cognitive science community in particular, but also science, more generally.

Methodological Details and Examples

In order to demonstrate the methods within the 'fractalRegression' package, we group this into univariate (DFA, MFDFA) and bivariate methods (DCCA, MRA, MRLA). For each method, we 1) highlight the key question(s) that can be answered with that method, 2) briefly describe the algorithm with sources for additional details, 3) describe some key consideration for appropriately applying the algorithm, and demonstrate the use of the functions on a 4) simulated and 5) empirical application of the function. An overview of the functions included in the package, the general objective of that function, and the output are shown below in Table 1. The additional details are included in the sections corresponding to those methods, in the package documentation, and in the original sources for the methods.

Table 1.

Overview of package functions, objectives, and output

Univariate Methods

Detrended Fluctuation Analysis. The key question that can be answered by Detrended Fluctuation Analysis (DFA) (Peng et al., 1994) is: *what is the magnitude and direction of long range correlation in a single time series?* While DFA has been described extensively elsewhere (Kantelhardt, Koscielny-Bunde, Rego, Havlin, & Bunde, 2001) and visualized nicely (Kelty-Stephen et al., 2016), we provide a brief summary here. DFA entails splitting a time series into several small bins (e.g., 16). In each bin, the least squares regression is fit and subtracted within each window. Residuals are squared and averaged within each window. Then, the square root is taken of the average squared residual across all windows of a given size. This process repeats for larger window sizes, growing by, say a power of 2, up to N/4, where N is the length of the series. In a final step, the logarithm of those scaled root mean squared residuals (i.e., fluctuations) is regressed on the logarithm of window sizes. The slope of this line is termed α and it provides a measure of the long range correlation. α is commonly used an as estimator of the Hurst exponent (H), where $\alpha < 1 = H$, and for $\alpha > 1$, $H = 1 - \alpha$. Conventional interpretation of α is: $\alpha < 0.5$ is anti-correlated, $\alpha\ = 0.5$ is uncorrelated, white noise, $\alpha > 0.5$ is temporally correlated, $\alpha\ = 1$ is long-range correlated, 1/f-noise, pink noise, $\alpha > 1$ is non-stationary and unbounded, and $\alpha\ = 1.5$ is fractional brownian motion.

DFA Examples. To demonstrate the use of `dfa()` we simulate three time series using the `fgn_sim()` function. This is a simple function based on the former `fARMA` R package. It only requires the number of observations `n`, and the Hurst exponent `H`. In particular, we simulate white noise, pink noise, and anti-correlated fractional Gaussian noise using the code below.

```
white.noise <- rnorm(5000)

pink.noise <- fgn_sim(n = 5000, H = 0.9)

anti.corr.noise <- fgn_sim(5000, H = 0.25)

scales <- logscale(scale_min = 16, scale_max = 1024, scale_ratio = 2)
```

Then, we can run DFA on these using the example code below. Note that this example uses a linear detrending with minimum scale of 16, a maximum scale that is at most 1/4 the time series length, and logarithmically spaced scale factor (`scale_ratio`) of 2.

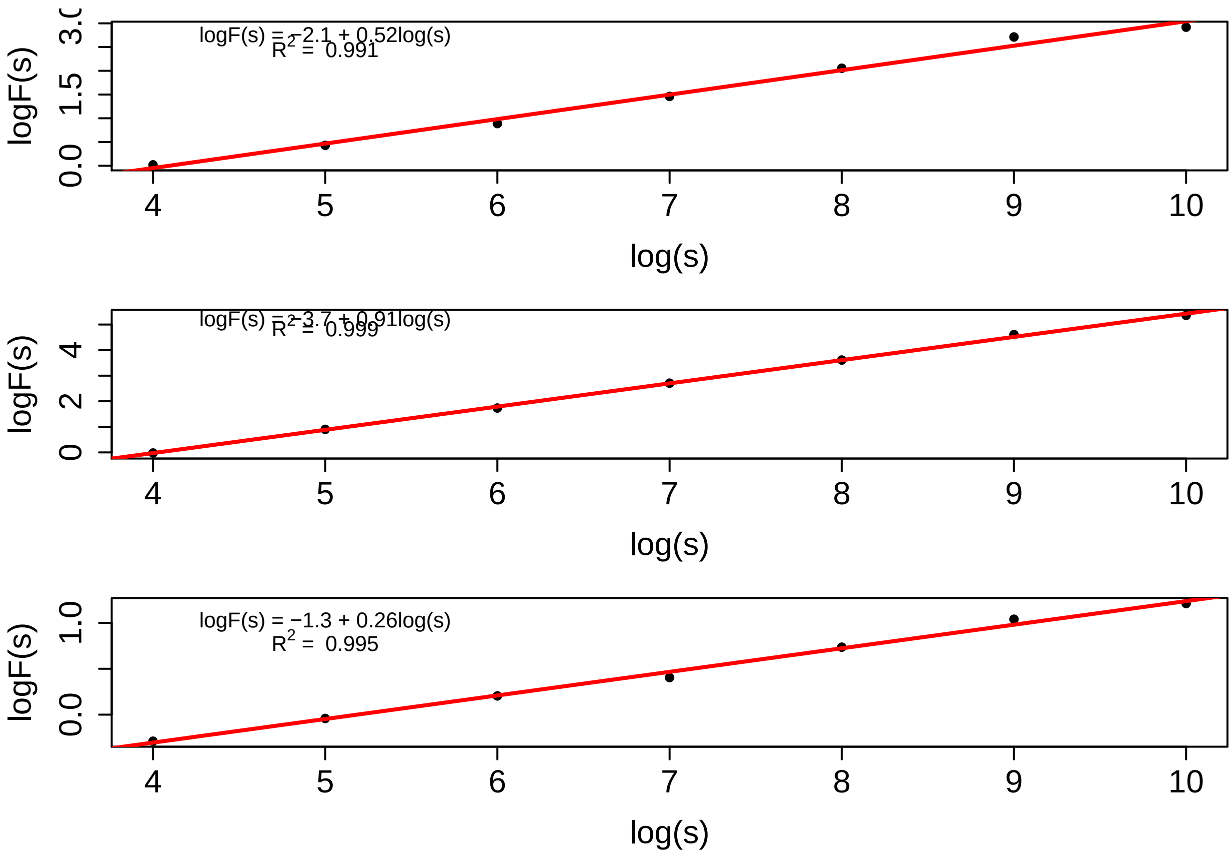
```
dfa.white <- dfa(x = white.noise, order = 1, verbose = 1, scales=scales, scale_ratio = 2

dfa.pink <- dfa(x = pink.noise, order = 1, verbose = 1,
scales=scales, scale_ratio = 2)

dfa.anti.corr <- dfa(x = anti.corr.noise, order = 1, verbose = 1, scales=scales, scale_r
```

In terms of output from the above examples, for white noise, we observed that $\alpha = 0.52$, for pink noise we observed that $\alpha = 0.91$, and since we simulated anti-correlated noise with H $= 0.25$, we observed a close estimate of the $\alpha = 0.26$. In terms of the objects saved from the `dfa()` function, one commonly inspects the `log_scales-log_rms` plots. Given the estimates above, we see in Figure 1 that the slopes for white noise, pink noise, and anti-correlated noise conform to our expectations. These slop estimates are provided n the equation listed above each respective line, and are generated using the `dfa.plot()` function.

Figure 1

Log scale-Log fluctuation plots for white noise (top), pink noise (middle), and anti-correlated noise (bottom)

```
par(mfrow=c(3,1))
dfa.plot(dfa.white)
dfa.plot(dfa.pink)
dfa.plot(dfa.anti.corr)
```

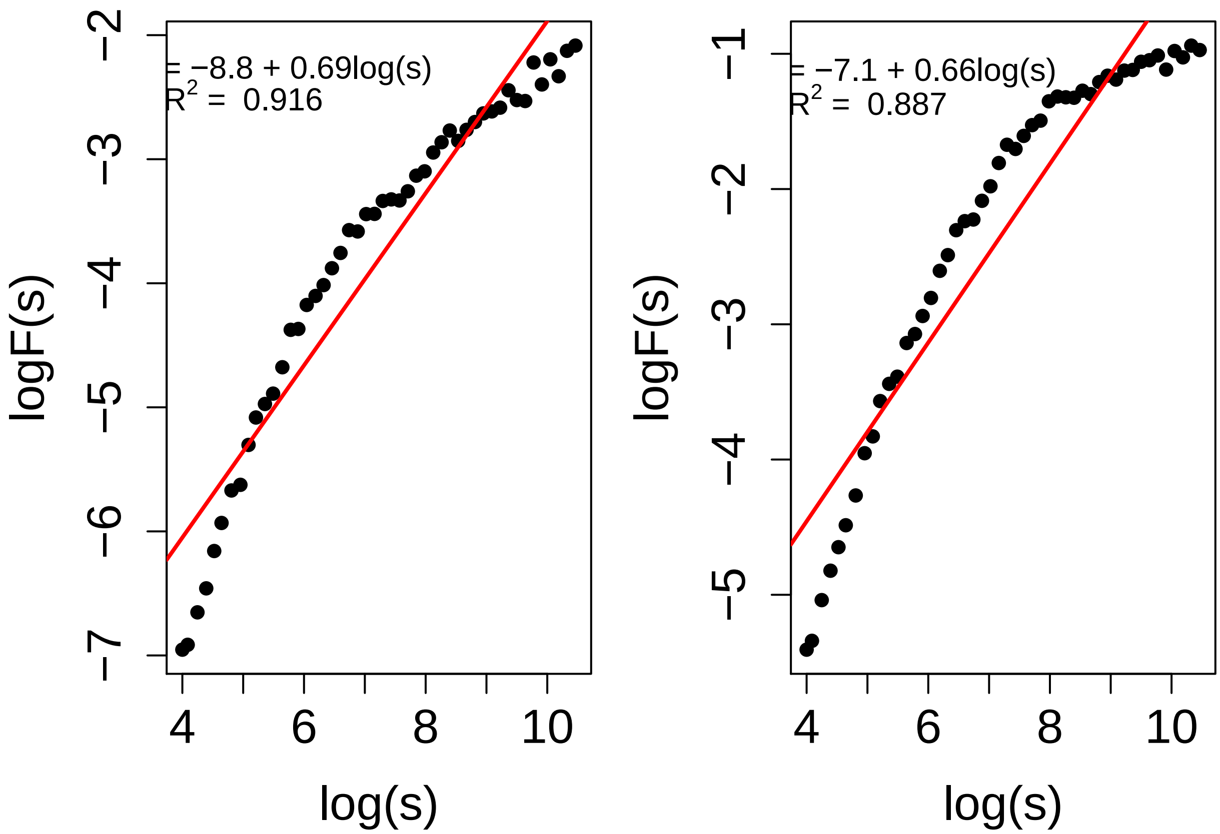


For an empirical example, we apply the `dfa()` function to the Human Balance Dataset (Santos & Duarte, 2016). For our purpose, this publicly available dataset includes signals from a force platform measured as the center of pressure in the x and y dimension for 87 young adults (we exclude the older adults for our analyses). Trials lasted 60s. See original paper for additional details on data processing (Santos & Duarte, 2016). For the empirical examples, we use two different time series featuring a participant standing on a firm (rigid) surface with eyes open and a foam (unstable) surface with eyes open. Note, however, that we analyze the dataset more fully in INSERT SECTION NAME HERE. We chose this dataset because postural sway data are known to exhibit interesting fractal dynamics (CITES) and we can systematically evaluate the data for all of the univariate and bivariate analyses detailed in this work.

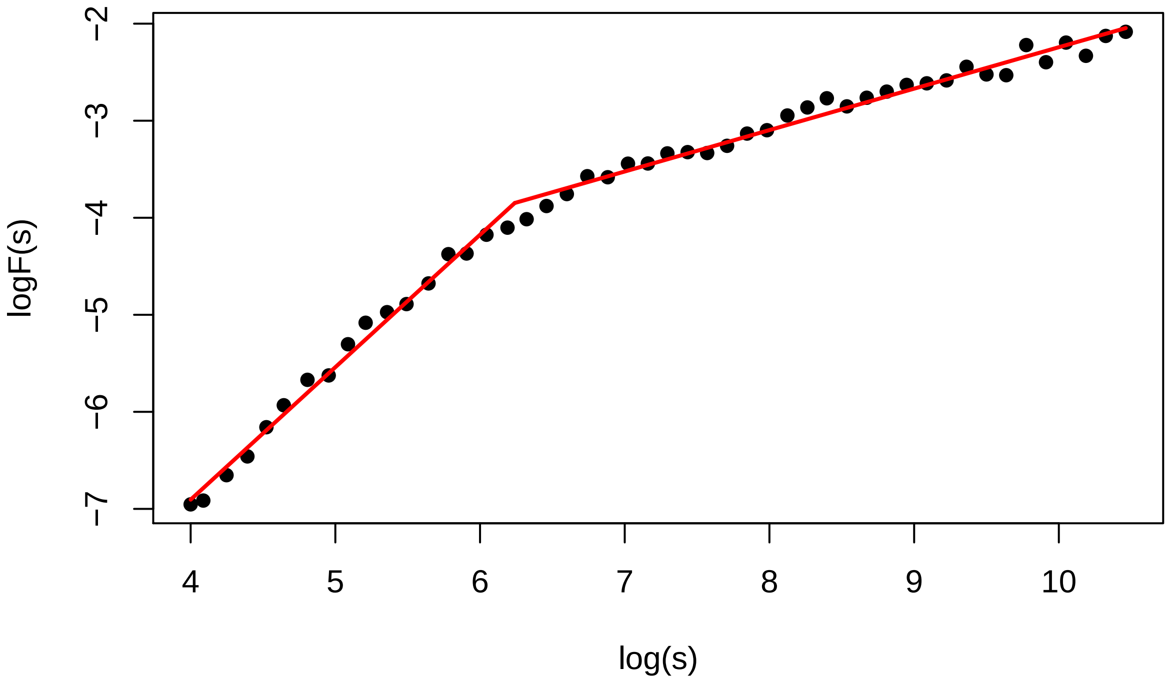
For the center of pressure (COP) data, we take the first order differences of each series. For the univariate analyses, we focus on analyses of the COP data in the x dimension. Then, we define the appropriate scales for the analyses.

Figure 2

Log scale-Log fluctuation plots for empirical difference COPx time series for rigid (left) and unstable (right) surfaces.



Importantly, regarding the question one can ask using DFA, we observe from Figure ##, that long range correlations are positive and approximately 0.69 - 0.66. However, from visual inspection of these plots we observe that two slopes might fit better than one for these time series; a phenomenon known as crossovers points (Ge & Leung, 2013). One common approach when such crossover points exist is to recognize that the signal might be best characterized by two scaling regions up to a particular time scale. We provide an example of how to check for where the break point is below using piece-wise regression.




```
## $log_scales
##            Est.  St.Err. t value CI(95%).l CI(95%).u
## slope1 1.36470 0.033545  40.683   1.29710   1.43230
## slope2 0.42691 0.013668  31.234   0.39936   0.45445
```

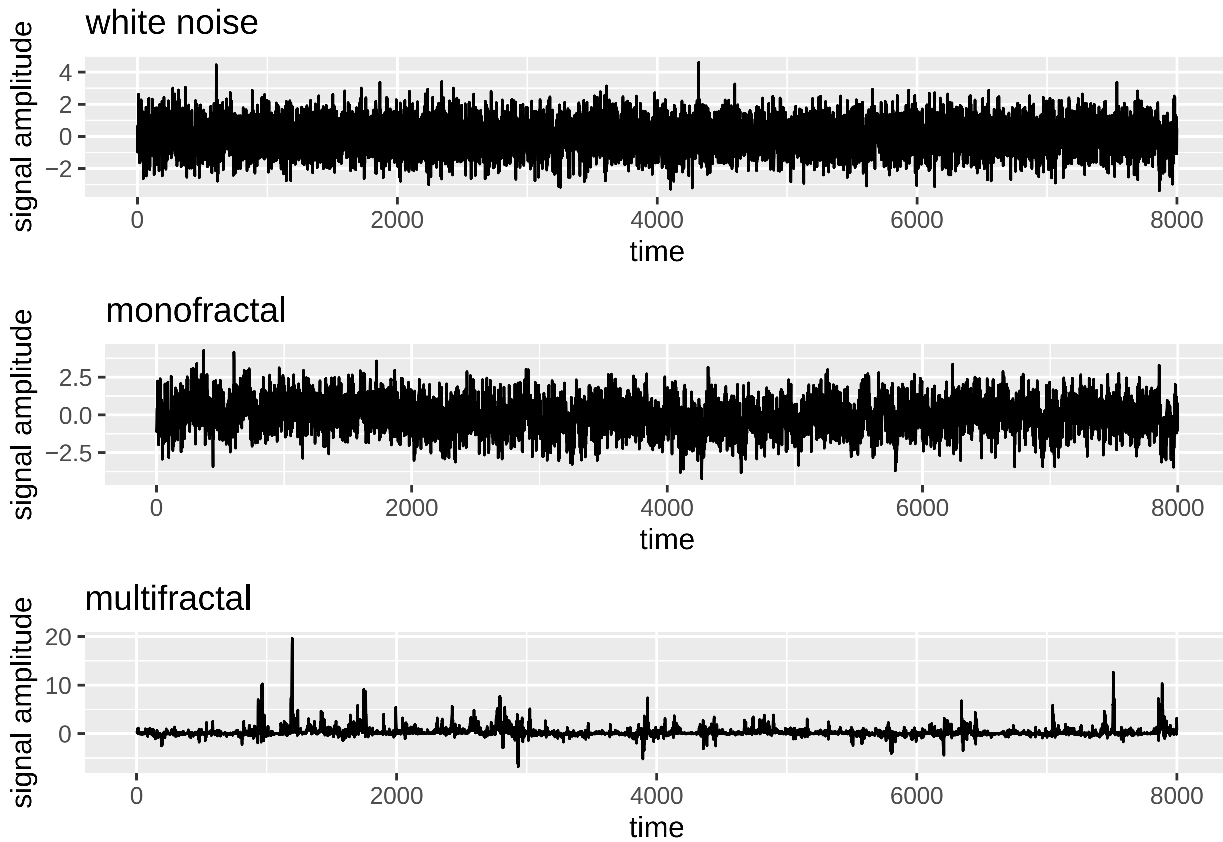
In the example above, we observe that the is a crossover point at INSERT VALUE. And, that there are two distinct scaling relationships: INSERT ALPHAS. The postural control literature has shown evidence of this type of dynamic before with short time scales displaying more DESCRIBE and longer time scale exhibiting DESCRIBE (CITE).

Multifractal Detrended Fluctuation Analysis. Multifractal Detrended Fluctuation Analysis (MFDFA; Kantelhardt et al. (2002)) is an extension of DFA by generalizing the fluctuation function to a range of exponents of the qth order. The key question that can be answered by MFDFA is: *how does the magnitude and direction of long* *range correlation change over time within a single time series?* Like DFA, MFDFA entails splitting a time series into several small bins (e.g., 16). In each bin, the least squares regression is fit and subtracted within each window. However, the residuals are raised to a range of exponents q and averaged within each window. So when $q = 2$, DFA is equal to MFDFA. When $q > 2$, larger residual are emphasized and when $q < 2$, smaller residuals are emphasized. The rest of the DFA algorithm is performed for each window and windows size for all values of q. We refer the reader to the work of Kelty-Stephens and colleagues Kelty-Stephen et al. (2016) Figure 3 for a visualization of the algorithm and to Kantelhardt and colleagues Kantelhardt et al. (2002) for additional mathematical description.

MFDFA Examples. To demonstrate the use of `mfdfa()`, we work with data included in our package (`fractaldata`), that was originally provided by E. A. F. Ihlen (2012) . It includes a white noise time series, monofractal time series, and a multifractal time series.

Figure 3

Time series from Ihlen (2012) corresponding to white noise, monofractal, and multifractal series.



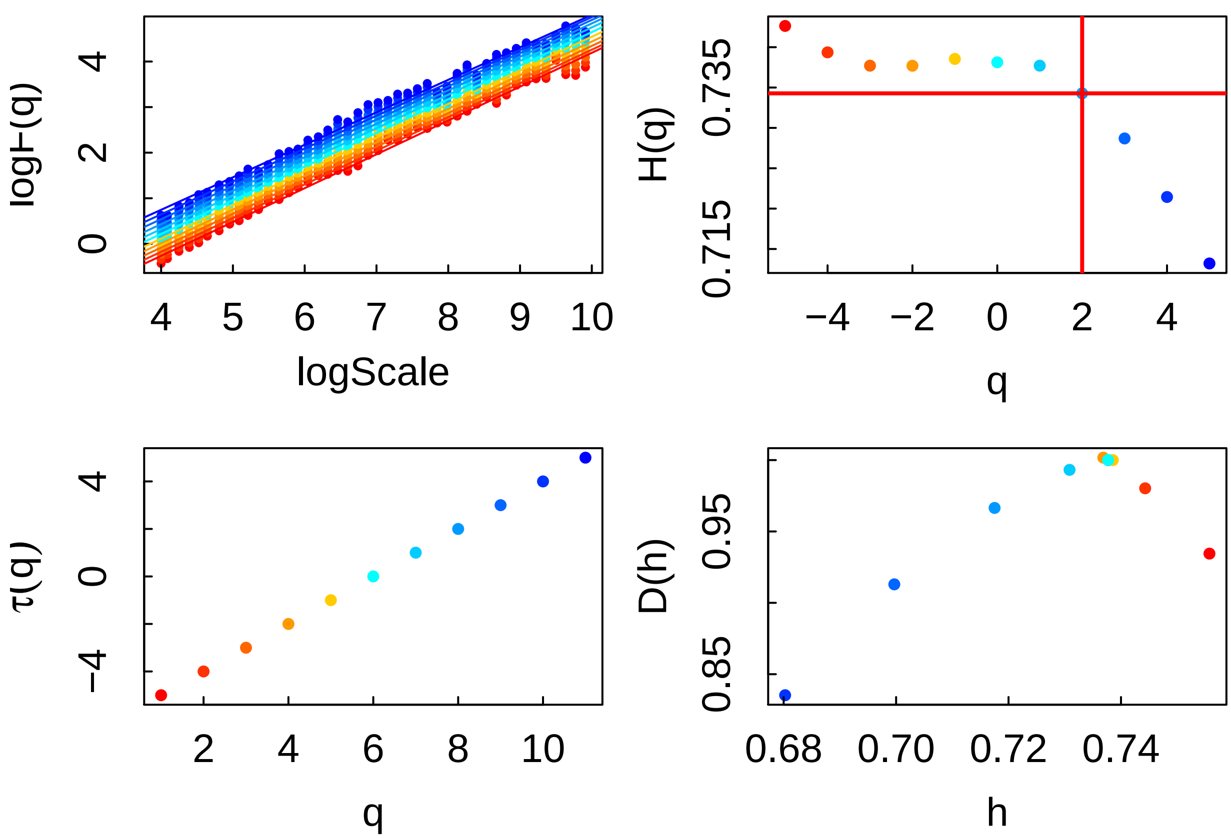
Performing MF-DFA is relatively straight forward with the `mfdfa()` function. As shown in the example below, one needs to enter the time series `x` to perform the analysis on, the range of `q` order exponents to use, the `order` of polynomial detrending, and the `scales` for the analysis. In this case, we define our `scales` by choosing logarithmically spaced scales and we select values of q from -5 to 5.

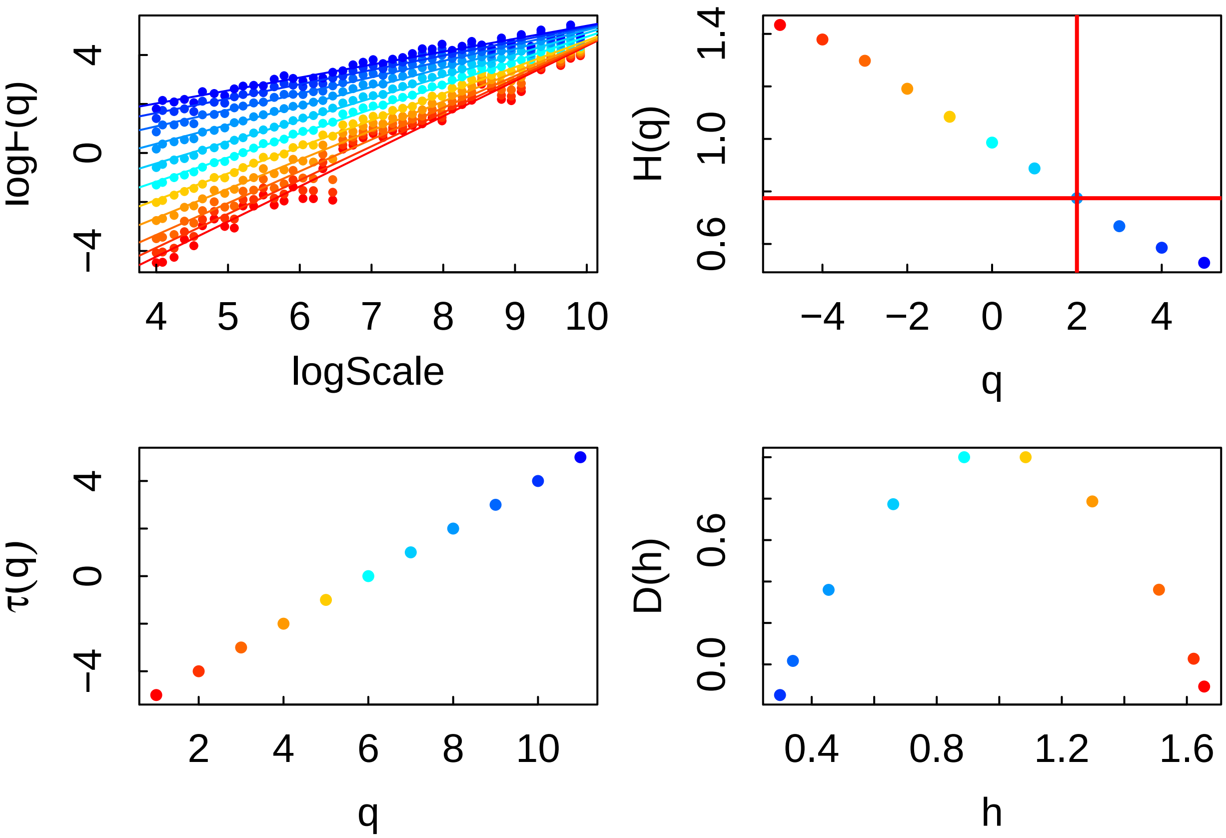
```
scales <- logscale(scale_min = 16,scale_max = 1024,scale_ratio = 1.1)

white.mf.dfa.out <- mfdfa(x = fractaldata$whitenoise, q = c(-5:5), order = 1, scales=sca

mono.mf.dfa.out <- mfdfa(x = fractaldata$monofractal, q = c(-5:5), order = 1,  scales=sc

multi.mf.dfa.out <- mfdfa(x = fractaldata$multifractal, q = c(-5:5), order = 1,  scales=
```

A common way to understand if there is evidence of multifractality is to examine a plot showing the slopes of the `log_fq` at the `log_scale` values. If all the plots have the same slope, that provides evidence of monofractality. If there are distinct slopes, then there is evidence of multifractality. It's also important to check here that the slopes of `log_scale` and `log_fq` are largely linear, thus implying that they are scale invariant. If not, then it could be the case that a higher order polynomial detrending is appropriate (see Kantelhardt et al., 2001). Figure 4 shows what we would expect for a monofractal and multifractal signal. In other words, the monofractal signal shows a consistent slope, whereas the multifractal signal shows variability in the slopes.

Figure 4

mfdfa.plots for mono-(top) and multifractal series (bottom). The four panels correspond to DETAILS HERE.





A common metric for comparing the multifractal spectrum is to calculate the width (W) as $h_{max} - h_{min}$. Let's do this to compare the monofractal and multifractal time series. We observe in this case that for the monofractal signal $W_{mono} = 0.08$ and $W_{multi} = 1.36$. If plot the multifractal spectra D(h) against h, we clearly observe the difference in the widths of the multifractal spectra for the mono- and multifractal signals, as shown in Figure 4 above.

For our empirical analysis, we again turn to the postural data. We set out parameters appropriate for the data and run `mfdfa()`.

DFA and MFDFA Considerations. We recommend a few points of consideration here in using this function. One is to be sure to evaluate whether there are cross-over points in the log scale-log fluctuation plots (Peng et al., 1994; Perakakis, Taylor, Martinez-Nieto, Revithi, & Vila, 2009). Cross-over points (or a visible change in the slope as a function of scale) indicate that a mono-fractal characterization does not sufficiently characterize the data. If cross-over points are evident, we recommend proceeding to estimate the two scaling regions with a piece-wise regression.

While it is common to use only linear detrending with DFA, it is important to inspect the trends in the data to determine if it would be more appropriate to use a higher order polynomial for detrending, and/or compare the DFA output for different polynomial orders (Kantelhardt et al., 2001).

General recommendations for choosing the min and max scale are minimum scale of 10 and a maximum scale of N/4, where N is the total number of observations in the signal. See Eke et al. (2002) (Eke, Herman, Kocsis, & Kozak, 2002) and Gulich and Zunino (2014) (Gulich & Zunino, 2014) for additional considerations.

Bivariate Methods

Detrended Cross-Correlation Analysis. Detrended Cross-Correlation Analysis (DCCA; Podobnik and Stanley (2008), Zebende (2011)) is a bivariate extension of the DFA algorithm generalizing it to a correlational case between two time series that may be non-stationary. The key questions that can be asked with it are: a) *How does correlation between two time series change as a function of scale?* and b) *What is/are the dominant (time) scale(s) of coordination? (those that are beyond a threshold, or statistically significant given a criteria, or of a certain magnitude?* For DCCA, the DFA algorithm gets applied to both time series providing the scale-wise estimates for both. DESCRIBE DCCA ALGORITHM HERE. Whereas in DFA, the key metric is α, in DCCA, one estimates the scale-specific, detrended cross-correlation coefficient ρ for the pair of time series.

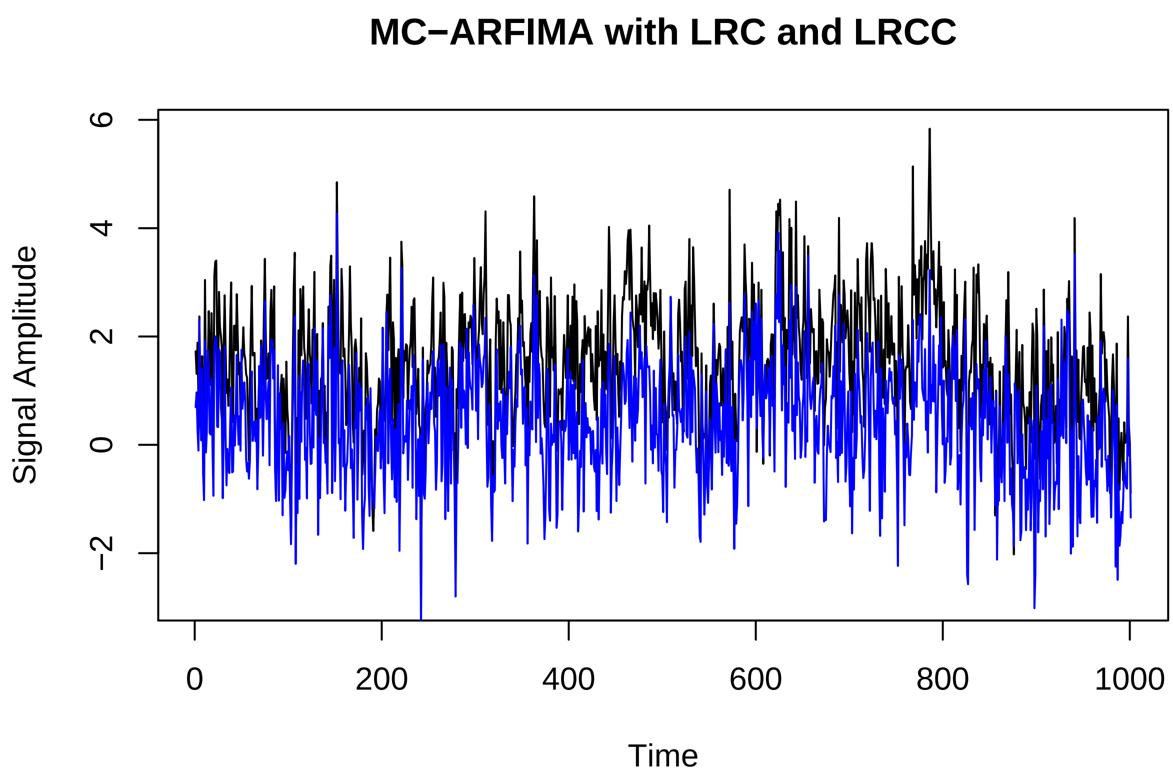
DCCA Examples. To demonstrate the use of `dcca()`, we used the `mc_arfima()` function from our package to simulate two time series with known properties. Specifically, we use the multicorrelated ARFIMA examples from Kristoufec's work (Kristoufek, 2013).

In this case, we use the parameters from Kristoufec (2013) for Model 1 (p. 6,487), that generates two time series of length 10,000 that exhibit long range correlations (LRC) as well as long range cross-correlations (LRCC). The code for simulating these two time series is shown below. Additionally, Figure #, shown below, visualizes a subset of these time series.

```
## Warning: package 'fracdiff' was built under R version 4.0.5
```

Figure #

Subset of two time series exhibiting long range correlation and long range cross-correlation



To perform the `dcca()` on these time series, we could use the code below, where we first define the `scales` using using the `logscale()` to define a set of logarithmically spaced scales to use for the analysis.

From here, we can visualize the output of the analysis as shown below in Figure #. We observe that, as expected, the correlation between the MC-ARFIMA processes are consistently high (all ρ's $> .8$) and continue to be high at increasing time scales. We also add standard errors plotted around each point, but note these are based on one simulation only and more robust estimates of standard error could be calculated using many time series.

Figure #

DCCA output for long range correlation and long range cross-correlation

As a point of comparison, we can generate a time series in contrast with this that exhibits processes with LRC and short-range cross-correlation (SRCC) using the code below. In contrast to the previous DCCA analysis, Figure # shows a signal that begins with a high cross-correlation (ρ's $> .6$) , but that begins to deviate and trend substantially lower at increasing scale sizes approaching $\rho = 0$.

Figure #

DCCA output for long range correlation and short range cross-correlation

Turning next to the empirical balance data, we apply DCCA to the COPx and COPy data.

INSERT HERE

Multi-scale Regression Analysis. Multi-scale regression analysis (MRA) is an adaptation of DCCA that brings the analyses into a predictive, regression framework Kristoufek (2015b) . The key questions that can be answered by it are: a) *How does the influence of one time series on another time series change as a function of scale?* and b) *What is/are the dominant (time) scale(s) of influence of one time series on another time series?* The algorithm is largely the same as DCCA, with a key difference being that instead of estimating scale-wise symmetric correlation coefficients, leveraging methods of

Ordinary Least Squares (OLS) regression, asymmetric β coefficients are estimated (see Likens, Amazeen, West, and Gibbons (2019b); Kristoufek (2015b)).

MRA Examples. Considering the LRC and LRCC simulations used for DCCA, we can examine whether the scale-wise fluctuations of one variable can predict the scale-wise fluctuations of the other using `mra()`. As with a traditional regression approach, we will use one of our variables as our predictor (x_t) and the other as our outcome (y_t). In the example below, we again first define our logarithmically spaced scales. We then apply the `mra()` function to the two simulated time series. In this case, it's important to specify which is variable is `x` (the predictor) and which is `y` (the outcome). For ease of plotting, we convert this to a dataframe in the process.

We can then visualize these results as shown below in Figure #. Generally, we observe that the β coefficients are relatively stable at increasing time scales with a general, perhaps quadratically increasing trend. Here it is also important to investigate the change in R^2 as well as the t-values. Below we see that the R^2 is quite high at most of the time scales with $R^2_{min} = 0.67$ and $R^2_{max} = 1.85$ and all of the t-values greater than the conventional cut-off at 1.96. So between these two component ARFIMA processes, the output of MRA shows that much of the scale specific variance in y_t is explained and predicted by x_t.

Figure #

MRA output for long range correlation and long range cross-correlation

Balance DATA ANALYSIS HERE

Multi-scale Lagged Regression Analysis. Multi-scale lagged regression analysis is an extension of MRA that allows for examining the influence as a function of scale, but also of time lag. In particular, the key questions that can be asked with MLRA are: a) *How does the influence of one time series on another time series change as a function of scale at different time lags?* and b) *Does the dominant time scale of influence change over successive time lags?* DESCRIBE MLRA ALGORITHM HERE.

MLRA Examples.

- MLRA
  - Key Question
  - Simulated data: Equation from Aaron from grant on MLRA
  - Empirical data: Balance Data

Surrogate Methods

In all of the above methods, one gets either a single estimate of a parameter (e.g., α) or a range of estimates (e.g., ρ, β). While those estimates are meaningful in and of themselves, it is common practice to perform some form of null hypothesis test regarding the estimate. These are generally referred to as surrogate methods Kantz and Schreiber (2003) . We present several options here that could be ranked in terms of increasing levels of rigor: randomized surrogates, iterative amplitude adjusted Fourier transformed (IAAFT) surrogates, and model based surrogates.

Randomized Surrogates. Randomized surrogates generally involve randomly shuffling the order of values of a time series. The idea is generally that the temporal structure is destroyed, yet the other features of the time series still exist (Kantelhardt et al., 2002). Note that additional options exist along these lines (see for example (Dumas, Nadel, Soussignan, Martinerie, & Garnero, 2010)). The key comparison here would be to compare the estimates extracted from a given analysis (e.g., DFA) on the observed sample of data with the estimates derived from and equally sized sample of the surrogate series (see Kantz and Schreiber (2003) Moulder, Boker, Ramseyer, and Tschacher (2018) (**wiltshire2019?**) for examples).

Randomizing the pink noise time series, which originally exhibited long range correlation ($\alpha = 0.91$), and performing DFA on it, now provides an estimate of $\alpha = 0.48$, which is consistent with a random or white noise process. These values are clearly different, however, performing inferential statistics on a sample of observed estimates compared to surrogate estimates would provide compelling evidence that the temporal dynamics suggested by the observed estimates are different than those derived from a random process.

Iterative Amplitude-Adjusted Fourier Transform Surrogates (IAAFT). The IAAFT algorithm was originally developed as a way to be able to evaluate, whether there is nonlinearity (Schreiber & Schmitz, 1996). More recently, it was proposed as a technique that could be use to see whether multifractal indices suggest interaction across scales (Espen A. F. Ihlen & Vereijken, 2010). Like with randomized shuffling, estimates derived from IAAFT surrogates should be also be different from the estimates derived from the empirical time series. Although in this case, the comparison is typically made between the multifractal spectra of the observed time series, and the spectra from a set of IAAFT surrogate series.

In the code below, we provide an example for generating IAAFT surrogates using the `iaafft()` function in the package. One enters the `signal`, which is the observed time series, and `N`, the number of surrogates to generate. There are a number of options here but a common number of surrogates is 19 (Kantz & Schreiber, 2003). Common practice is that surrogates are generated from many observed time series, but here we illustrate using only a single time series: the multifractal signal used previously in the MFDFA example. Then we use the same parameters for the `mfdfa()` function, but apply it to all of the IAAFFT surrogates.

Assuming we were using IAAFFT to compare the multifractal width (W) between the observed signal and the surrogate signals, recall that the observed widith was $W_{multi} = 1.36$. Now, we can calculate the average multifractal width across all of the generated surrogates and we observe that $W_{surr} = 0.64$, which is narrower than the spectrum from the multifractal signal. In practice, there are many surrogate options (Moulder et al., 2018), but, again, inferential statistics are commonly performed to compare observed estimates to the surrogate estimates to bolster evidence of the inferred dynamics.

Model-based Surrogates.

- Model based surrogate (Simulated exponents) - See Likens 2019 paper with model of postural sway/control, taking an educated guess about the data generating process underlying the time series. Estimates should not be different. See Roume et al 2018 windowed detrended CCA
- Can we incorporate lags into MC-ARFIMA?

General Discussion

- General value of methods and the types of questions (mention the types of data used in empirical examples.

- Practical consideration of univariate methods

  - Length of time series

- Practical consideration of bivariate methods

  - Length of time series

  - Likens et al. 2019 found positive bias of linear and quadratic trends on MRA beta estimates at larger scales that could be mitigated with larger detrending order. This involves checking the time series with time as a predictor and polynomials.

- Unique contribution of the methods

Appendix 1: Fundamental Equations

Here we will insert the fundamental equations for showcasing the algorithms. WE NEED Lagged functions and MFDFA.

DFA

$F_X = \sqrt{\frac{\sum_{j-1}^{T-s+1} f_X^2(s,j)}{T-s}}$

where

$f_X^2(s,j) = \frac{\sum_{k=j}^{j+s-1}(X_k - \widehat{X}_{k,j})}{s-1}$

DCCA

$F_Y = \sqrt{\frac{\sum_{j-1}^{T-s+1} f_Y^2(s,j)}{T-s}}$

where

$f_Y^2(s,j) = \frac{\sum_{k=j}^{j+s-1}(Y_k - \widehat{Y}_{k,j})}{s-1}$

and the scale-wise covariance is estimated as:

$f_{XY}^2(s,j) = \frac{\sum_{k=j}^{j+s-1}(X_k - \widehat{X}_{k,j})(Y_k - \widehat{Y}_{k,j})}{s-1}$

which forms the basis for the scale-wise correlation coefficient estimated as:

$\rho(s) = \frac{F_{XY}^2(s)}{F_X(s)F_Y(s)}$

and for the multi-scale regression coefficients, we replace the denominator in the $\rho(s)$ equation with scale-wise variance of the predictor to estimate the scale-wise regression coefficient from regression Y_t on X_t as:

$\widehat{\beta}(s) = \frac{F_{XY}^2(s)}{F_X^2(s)}$

and where the variance of $\widehat{\beta}(s)$ is:

$\sigma^2_{\widehat{\beta}(s)} = \frac{1}{T-2} \times \frac{F_u^2(s)}{F_Y^2(s)}$

and the scale-wise residual variance, $\widehat{F}_u^2(s)$ is estimated by applying the DFA algorithm to all scale-wise residuals, $\widehat{u}_t(s)$ as:

$$\widehat{u}_t(s) = y_t - x_t\widehat{\beta}(s) - \overline{y_t - x_t\widehat{\beta}(s)}$$

Acknowledgments

Author AL receives support from a National Institutes of Health Center grant (P20GM109090).

References

Bak, P., Tang, C., & Wiesenfeld, K. (1987). Self-organized criticality: An explanation of the 1/f noise. *Physical Review Letters*, *59*(4), 381–384. https://doi.org/10.1103/PhysRevLett.59.381

Bianchi, S. (2020). Fathon: A python package for a fast computation of detrendend fluctuation analysis and related algorithms. *Journal of Open Source Software*, *5*(45), 1828.

Cavanaugh, J. T., Kelty-Stephen, D. G., & Stergiou, N. (2017). *Multifractality, Interactivity, and the Adaptive Capacity of the Human Movement System: A Perspective for Advancing the Conceptual Basis of Neurologic Physical Therapy.* Retrieved from https://www.ingentaconnect.com/content/wk/npt/2017/00000041/00000004/art00007

Davis, T. J., Brooks, T. R., & Dixon, J. A. (2016). Multi-scale interactions in interpersonal coordination. *Journal of Sport and Health Science*, *5*(1), 25–34. https://doi.org/10.1016/j.jshs.2016.01.015

Delignières, D., Almurad, Z. M. H., Roume, C., & Marmelat, V. (2016). Multifractal signatures of complexity matching. *Experimental Brain Research*, *234*(10), 2773–2785. https://doi.org/10.1007/s00221-016-4679-4

Delignières, D., Torre, K., & Lemoine, L. (2008). Fractal models for event-based and dynamical timers. *Acta Psychologica*, *127*(2), 382–397. https://doi.org/10.1016/j.actpsy.2007.07.007

Dumas, G., Nadel, J., Soussignan, R., Martinerie, J., & Garnero, L. (2010). Inter-Brain Synchronization during Social Interaction. *PLOS ONE*, *5*(8), e12166. https://doi.org/10.1371/journal.pone.0012166

Eddelbuettel, D., & Francois, R. (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, *40*(1), 1–18. https://doi.org/10.18637/jss.v040.i08

Eddelbuettel, D., & Sanderson, C. (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Computational Statistics & Data Analysis*, *71*, 1054–1063. https://doi.org/10.1016/j.csda.2013.02.005

Eke, A., Herman, P., Kocsis, L., & Kozak, L. R. (2002). Fractal characterization of complexity in temporal physiological signals. *Physiological Measurement*, *23*(1), R1–R38. https://doi.org/10.1088/0967-3334/23/1/201

Euler, M. J., Wiltshire, T. J., Niermeyer, M. A., & Butner, J. E. (2016). Working memory performance inversely predicts spontaneous delta and theta-band scaling relations. *Brain Research*, *1637*, 22–33. https://doi.org/10.1016/j.brainres.2016.02.008

Garcia, C. A. (2020). *nonlinearTseries: Nonlinear Time Series Analysis*. Retrieved from https://CRAN.R-project.org/package=nonlinearTseries

Ge, E., & Leung, Y. (2013). Detection of crossover time scales in multifractal detrended fluctuation analysis. *Journal of Geographical Systems*, *15*(2), 115–147. https://doi.org/10.1007/s10109-012-0169-9

Goldberger, A. L., Amaral, L. A. N., Hausdorff, J. M., Ivanov, P. C., Peng, C.-K., & Stanley, H. E. (2002). Fractal dynamics in physiology: Alterations with disease and aging. *Proceedings of the National Academy of Sciences*, *99*(suppl 1), 2466–2472. https://doi.org/10.1073/pnas.012579499

Gulich, D., & Zunino, L. (2014). A criterion for the determination of optimal scaling ranges in DFA and MF-DFA. *Physica A: Statistical Mechanics and Its Applications*, *397*, 17–30. https://doi.org/10.1016/j.physa.2013.11.029

Hardstone, R., Poil, S.-S., Schiavone, G., Jansen, R., Nikulin, V. V., Mansvelder, H. D., & Linkenkaer-Hansen, K. (2012). Detrended Fluctuation Analysis: A Scale-Free View on Neuronal Oscillations. *Frontiers in Physiology*, *3*. https://doi.org/10.3389/fphys.2012.00450

Hausdorff, J. M., Purdon, P. L., Peng, C. K., Ladin, Z., Wei, J. Y., & Goldberger,

A. L. (1996). Fractal dynamics of human gait: Stability of long-range correlations in stride interval fluctuations. *Journal of Applied Physiology*, *80*(5), 1448–1457. https://doi.org/10.1152/jappl.1996.80.5.1448

Ihlen, E. A. F. (2012). Introduction to Multifractal Detrended Fluctuation Analysis in Matlab. *Frontiers in Physiology*, *3*. https://doi.org/10.3389/fphys.2012.00141

Ihlen, Espen A. F., & Vereijken, B. (2010). Interaction-dominant dynamics in human cognition: Beyond 1/f? fluctuation. *Journal of Experimental Psychology: General*, *139*(3), 436–463. https://doi.org/10.1037/a0019098

Kantelhardt, J. W., Koscielny-Bunde, E., Rego, H. H. A., Havlin, S., & Bunde, A. (2001). Detecting long-range correlations with detrended fluctuation analysis. *Physica A: Statistical Mechanics and Its Applications*, *295*(3), 441–454. https://doi.org/10.1016/S0378-4371(01)00144-3

Kantelhardt, J. W., Zschiegner, S. A., Koscielny-Bunde, E., Havlin, S., Bunde, A., & Stanley, H. E. (2002). Multifractal detrended fluctuation analysis of nonstationary time series. *Physica A: Statistical Mechanics and Its Applications*, *316*(1), 87–114. https://doi.org/10.1016/S0378-4371(02)01383-3

Kantz, H., & Schreiber, T. (2003). *Nonlinear Time Series Analysis*. Retrieved from /core/books/nonlinear-time-series-analysis/519783E4E8A2C3DCD4641E42765309C7

Kelty-Stephen, D. G. (2017). Threading a multifractal social psychology through within-organism coordination to within-group interactions: A tale of coordination in three acts. *Chaos, Solitons & Fractals*, *104*, 363–370. https://doi.org/10.1016/j.chaos.2017.08.037

Kelty-Stephen, D. G., Stirling, L. A., & Lipsitz, L. A. (2016). Multifractal temporal correlations in circle-tracing behaviors are associated with the executive function of rule-switching assessed by the Trail Making Test. *Psychological Assessment*, *28*(2), 171–180. https://doi.org/10.1037/pas0000177

Kristoufek, L. (2013). Mixed-correlated ARFIMA processes for power-law cross-correlations. *Physica A: Statistical Mechanics and Its Applications*, *392*(24), 6484–6493. https://doi.org/10.1016/j.physa.2013.08.041

Kristoufek, L. (2015a). Detrended fluctuation analysis as a regression framework: Estimating dependence at different scales. *Physical Review E*, *91*(2), 022802. https://doi.org/10.1103/PhysRevE.91.022802

Kristoufek, L. (2015b). Detrended fluctuation analysis as a regression framework: Estimating dependence at different scales. *Physical Review E*, *91*(2), 022802. https://doi.org/10.1103/PhysRevE.91.022802

Laib, M., Golay, J., Telesca, L., & Kanevski, M. (2018). Multifractal analysis of the time series of daily means of wind speed in complex regions. *Chaos, Solitons & Fractals*, *109*, 118–127. https://doi.org/10.1016/j.chaos.2018.02.024

Legrand, P., & Véhel, J. L. (2003). Signal and image processing with FracLab. *Thinking in Patterns: Fractals and Related Phenomena in Nature*, 321322.

Likens, A. D., Amazeen, P. G., West, S. G., & Gibbons, C. T. (2019a). Statistical properties of Multiscale Regression Analysis: Simulation and application to human postural control. *Physica A: Statistical Mechanics and Its Applications*, *532*, 121580. https://doi.org/10.1016/j.physa.2019.121580

Likens, A. D., Amazeen, P. G., West, S. G., & Gibbons, C. T. (2019b). Statistical properties of Multiscale Regression Analysis: Simulation and application to human postural control. *Physica A: Statistical Mechanics and Its Applications*, *532*, 121580. https://doi.org/10.1016/j.physa.2019.121580

Likens, A. D., Fine, J. M., Amazeen, E. L., & Amazeen, P. G. (2015). Experimental control of scaling behavior: What is not fractal? *Experimental Brain Research*, *233*(10), 2813–2821. https://doi.org/10.1007/s00221-015-4351-4

Moulder, R. G., Boker, S. M., Ramseyer, F., & Tschacher, W. (2018). Determining synchrony between behavioral time series: An application of surrogate data generation for establishing falsifiable null-hypotheses. *Psychological Methods*, *23*(4), 757–773. https://doi.org/10.1037/met0000172

Peng, C.-K., Buldyrev, S. V., Havlin, S., Simons, M., Stanley, H. E., & Goldberger, A. L. (1994). Mosaic organization of DNA nucleotides. *Physical Review E*, *49*(2), 1685–1689. https://doi.org/10.1103/PhysRevE.49.1685

Perakakis, P., Taylor, M., Martinez-Nieto, E., Revithi, I., & Vila, J. (2009). Breathing frequency bias in fractal analysis of heart rate variability. *Biological Psychology*, *82*(1), 82–88. https://doi.org/10.1016/j.biopsycho.2009.06.004

Podobnik, B., & Stanley, H. E. (2008). Detrended Cross-Correlation Analysis: A New Method for Analyzing Two Nonstationary Time Series. *Physical Review Letters*, *100*(8), 084102. https://doi.org/10.1103/PhysRevLett.100.084102

Santos, D. A., & Duarte, M. (2016). A public data set of human balance evaluations. *PeerJ*, *4*, e2648. https://doi.org/10.7717/peerj.2648

Schaworonkow, N., & Voytek, B. (2021). Longitudinal changes in aperiodic and periodic activity in electrophysiological recordings in the first seven months of life. *Developmental Cognitive Neuroscience*, *47*, 100895. https://doi.org/10.1016/j.dcn.2020.100895

Schreiber, T., & Schmitz, A. (1996). Improved surrogate data for nonlinearity tests. *Physical Review Letters*, *77*(4), 635–638. https://doi.org/10.1103/PhysRevLett.77.635

Snow, E. L., Likens, A. D., Allen, L. K., & McNamara, D. S. (2016). Taking Control: Stealth Assessment of Deterministic Behaviors Within a Game-Based System. *International Journal of Artificial Intelligence in Education*, *26*(4), 1011–1032. https://doi.org/10.1007/s40593-015-0085-5

Stephen, D. G., Boncoddo, R. A., Magnuson, J. S., & Dixon, J. A. (2009). The dynamics of insight: Mathematical discovery as a phase transition. *Memory & Cognition*, *37*(8), 1132–1149. https://doi.org/10.3758/MC.37.8.1132

Team, R. C. (2018). *R: A language and Environment for Statistical Computing.* Vienna, Austria.

Van Orden, G. C., Holden, J. G., & Turvey, M. T. (2003). Self-organization of cognitive performance. *Journal of Experimental Psychology: General*, *132*(3), 331–350. https://doi.org/10.1037/0096-3445.132.3.331

Zebende, G. F. (2011). DCCA cross-correlation coefficient: Quantifying level of cross-correlation. *Physica A: Statistical Mechanics and Its Applications*, *390*(4), 614–618. https://doi.org/10.1016/j.physa.2010.10.022

Function	Objective	Output
`dfa()`	Estimate long-range correlation in a time series	Object containing the overall α estimate and, if desired the `logScales` and `logRMS`
`mfdfa()`	Estimate the magnitude and range of long-range correlations in a time series	Object containing the $\log_2$ scales used for the analysis, the $\log_2$ fluctuation function for each scale and q, the various q-order exponents, Hq, Tau, h, and Dh
`dcca()`	Estimates of scale-specific correlation between two time-series	Object containing the scales used for the analysis and the ρ '`rho'` values for each scale
`mra()`	Estimates the scale specific regression coefficients for a predictor time series on and outcome time series	Object containing the scales and scale specific β estimates, R^2, and t statistics
`mlra()`	Estimates the scale specific regression coefficients for a predictor time series on and outcome time series at pre-specified lags	Object with lag-specific β coefficients
`fgn_sim()`	Simulate univariate fractional Gaussian noise	Returns a vector of length `n` according to the specified `H` Hurst exponent

Function	Objective	Output
`mBm_mGn()`	Simulate univariate multi-fractional Brownian motion and Gaussian noise	Returns two vectors of length `N` according to the specified H_t series
`mc_ARFIMA()`	Simulate various types of bivariate correlated noise processes.	Returns two vectors of length `N` according to the specified noise `process` and parameters
`iaaft()`	Generate surrogate series using the iterative amplitude adjusted Fourier transform	Returns a vector of same length as input time series