Singular Spectrum Analysis: Illustrated by Application to S Persei and RZ Cassiopeiae

Geoff B. Chaplin
Hokkaido, Kamikawa-gun, Biei-cho, Aza-omura Okubo-kyosei, 071-0216, Japan;geoff@geoffgallery.net

Received June 27, 2018; revised September 24, October 10, 2018; accepted October 11, 2018

Abstract We describe two methods of singular spectrum analysis, a data driven technique, providing and using code to analyze example data series, and introducing the public domain R package "Rssa." The analysis provides potential information about the underlying behavior of the series, stripping out noise, and is a pre-requisite for some further work such as non-linear time series analysis. Examples are taken from a long time series of S Per magnitude observations, and secular period changes in, and high frequency magnitude variations of, RZ Cas.

1. Introduction

Singular Spectrum Analysis ("SSA") has gained popularity since the mid-1980s as a data driven rather than model driven method for the analysis of time series in a wide range of disciplines, from meteorology, to medical sciences, engineering, finance, and physics. Papers on SSA commenced with Broomhead and King (1986a, 1986b), although some ideas can be traced back before this. Other influential early papers are Fraedrich (1986), Vautard and Ghil (1989), Vautard *et al.* (1992), and Allen and Smith (1996). Development of SSA was paralleled by Danilov and Zhigljavsky (1997) in the former USSR. References to recent publications in a wide range of different areas can be found in Zhigljavsky (2010) and by various authors in two issues of *Statistics and its Interface* (2010, 2017). The review paper by Ghil *et al.* (2002) gives an extensive list of references to earlier work.

The descriptions given here follow Golyandina *et al.* (2001), Golyandina and Zhigljavsky (2013), and Golyandina *et al.* (2018) which additionally provide many further examples from a variety of disciplines; the last additionally gives an extensive list of related research articles. SSA provides inter alia a means of identifying three major components of the time series to be analyzed. Long-term variations ("trends") form a component which typically is difficult to predict without some form of model or understanding of the underlying process. Fourier analysis usually reflects this long-term behavior as a rise in amplitude as frequency decreases. Noise elements are those with no particular pattern, generally weak, and often related to the observation process. Most importantly we wish to extract the "signal," manifested a some form of periodic variation, as both an end in itself and a pre-requisite for further analysis, for example non-linear time series analysis (NLTSA) or where further analysis requires a stationary data series. NLTSA has been used, albeit based on different techniques, in astronomical literature by Kollath (1990) and Buchler *et al.* (1996), and others in the context of giant variable stars. Methodology for NLTSA based on SSA analysis, and code, is provided by Huffaker *et al.* (2017) and code therein forms the basis of the code in this paper.

We describe in detail two methods of SSA along with code implementing these approaches. The code in Appendix A.1 and A.2 closely follows the mathematical methods, while the code in Appendix A.3 calls more efficient (but black box) library code. We illustrate the methods by application to two long-term astronomical time series—visual observations of the magnitude of the semi-regular variable S Per, and observed minus calculated times of minimum for the eclipsing binary RZ Cas, together with an analysis of high-frequency CCD/DSLR magnitude observations stripping out a signal which is far weaker than the noise in the data and revealing δ Scuti-type variations.

In this paper we use the R (R Foundation 2018a) statistical programming language and CRAN (R Foundation 2018b) libraries and in particular the function "ssa" in the R library "Rssa" (R Foundation 2018c). In the Appendix we provide code adapted from Huffaker *et al.* (2017) to perform the analysis. RStudio (2018) provides a convenient user interface to the R code and many of the charts below are taken directly from the RStudio platform.

2. Methodology

We use two different approaches ("1d-ssa" and "Toeplitz-ssa") to the construction of the "trajectory matrix" and the "lagged correlation matrix" after which reconstruction of the series follows the same process.

2.1. Decomposition—"1d-ssa"

A data series taken at equal time points is first adjusted by removing the average value and may then be represented by a set of numbers $(O_1, O_2, \ldots, O_n)$ where n is 900 for example. A first column $(O_1, O_2, \ldots, O_k)$, a second column $(O_2, O_3, \ldots, O_k, O_{k+1})$ and a third column $(O_3, O_4, \ldots, O_k, O_{k+1}, O_{k+2})$ (and so on) can rather trivially be produced starting one observation later (a "delay" on one) for some $k < n$. Each of these rows is called a "lagged" or "Takens" vector and stacking m (for example, 400) such columns next to each other produces what is termed the "trajectory matrix," X, where in our example X has 400 columns and $k = 501$ rows ($k = n - m + 1$ so that all the data are used). In this paper we take m to be a little under half the length of the time series; general advice is that m should be sufficiently large that we capture the main features of the data but less than half the length of the time series. In addition if a strong periodic signal is present m should be a multiple of the period. Golyandina and Zhigljavsky (2013) gives many examples where some choices of the embedding dimension are however very different from half the length of the series. Step 1

in the "SVD code" in the Appendix creates the trajectory matrix after reading in the data. (The data file should be formatted in column(s) as a csv file with the first row naming the column(s).)

The transpose of the trajectory matrix multiplied by the matrix gives an $m \times m$ matrix, $S = X^T X$, whose terms are covariances of the observations and is called the "lagged correlation matrix" and where m is called the "embedding dimension." Step 2 creates the lagged correlation matrix, and code is given in Appendix A.1.

2.2. Decomposition—"Toeplitz-ssa"

The series must be approximately stationary for Toeplitz decomposition (Golyandina and Zhigljavsky (2013) section 2.5.3). The first column of the trajectory matrix is the entire series, the second column as above starts at O_2 but pads the end with zero, the third column starts at O_3 and has two zeros at the end and so on. The lagged correlation matrix is calculated not as above but from the formula

$$S_{ij} = \sum_{t=1}^{t=n-|i-j|} O_t \, O_{t-|i-j|} / (n-|i-j|) \quad (1)$$

The lagged correlation matrix again has *m* eigenvalues and eigenvectors. The alternative code is given in Appendix A.2.

2.3. Singular value decomposition—"SVD"

Any matrix of the form of S has *m* eigenvectors (see, for example, Lang 2013), EV1<=i<=m such that EVi multiplied by S simply stretches the EVi by a factor Li (the "eigenvalue") but doesn't change its direction. These eigenvectors also have the property that they are perpendicular to each other so define axes in m-dimensional space. We sort the eigenvectors in order from strongest eigenvalue to the weakest. The vectors

$$V_i = XE_i / \sqrt{L_i} \quad (2)$$

(the eigenvalue term being introduced merely for normalization) are a projection of the time series of observations onto that eigenvector axis. The relative strength associated with each component is L_i / L where L is the sum of the eigenvalues. Step 3 performs these calculations and in our example under 1d-ssa V is a 501-length vector, whereas under Toeplitz it has length 900. Step 3 also writes the eigenvalues to a file and plots the relative magnitudes on a log scale.

The trajectory matrix decomposes into $X = X_1 + ... X_m$ where

$$X_i = V_i E_i^T \sqrt{L_i} \quad (3)$$

and (each X_i has rank 1 and), $[\sqrt{L_i}, E_i, V_i]$ is referred to as the *i*th eigentriple of the SVD of X. Step 4 calculates the decomposition.

2.4. Series reconstruction

The values in each X_i are then averaged across "anti-diagonals" (row + column = constant) to give a time series component $\{x_i\}$ of the signal where in both decomposition cases the component has length *n* (900 in our example). Note that in the case of 1d-ssa decomposition the averaging is over 400 values for $400 <= i <= 500$, whereas under Toeplitz the averaging stops at row 900 of each X. Step 5 of the "SVD code"calculates the individual averaged series, produces a graphic of the correlations between the m different time series, produces a graphic of a portion of the time series, and writes the series to a data file. The user selects how many series to plot in the user input section—typically starting with 40 or so then refining to 10 or 20.

The "reconstructed signal(s)" we choose for further analysis is a sum of a subset of the component signals where the signals meet certain requirements—not being part of the noise, having similar periodicity (trend or high frequency variation) and being sufficiently independent from other signals, as illustrated below. The graphical results help in this decision making process: time series which are highly correlated should be grouped together, time series which have no correlation with other signals can be treated as a separate signal; time series with very different periodicities / patterns may be better treated separately.

The more efficient Rssa package can be used instead of the above code to perform the same calculations and graphical analysis, and is also illustrated in Appendix A.3.

3. S Per magnitude variability

S Per (GSC 03698-03073) is an M4.5-7Iae C spectral type (Wenger *et al.* 2000) Src (Kiss *et al.* 2006) variable star with period(s) variously identified as 813 ± 60 (Kiss *et al.* 2006); 822 (Samus *et al.* 2017); 745, 797, 952, 2857 (Chipps *et al.* 2004). The strong color causes significant differences in the estimation of magnitude by visual observers arising from observer dependent color response, the Purkinje effect (Purkinje 1825), local atmospheric conditions, altitude of the star at the time of observation and other factors, giving rise to a significant level of noise related to the observation process—"extrinsic" noise. In addition there may be "intrinsic" random variability caused by the star and its environment—for example, matter thrown off by the star may form a non-uniform cloud causing variation over time in scattering of the light away from the observer. Data are taken from the BAA (2018) and the AAVSO (Kafka 2018) databases, and from the VSOLJ (2018) database prior to 2000. We restrict our attention to observations made by experienced observers (defined as those reporting over 100 observations of the star).

Figure 1a shows visually estimated magnitudes from experienced observers starting from JD 2423000 grouped into 878 40-day buckets. The buckets contained two empty buckets and values were estimated by linear interpolation from neighboring observations. Had the number of missing points been substantial, then a more sophisticated interpolation method, such as in Kondrashev and Ghil (2006), would be appropriate.

Applying 1d-ssa analysis Figure 1b shows a sharp drop in strength after the fourth eigenvector and another after the twelfth EV. A "scree test" (Cattell 1965a, 1965b) is often used to decide where signal ends and noise starts but in the case of very noisy data (for example visual magnitude observations of narrow range red variables) there may nevertheless be an uncorrelated but weak signal present after the strong presence of the noise begins and such a signal should not be ignored. Figure

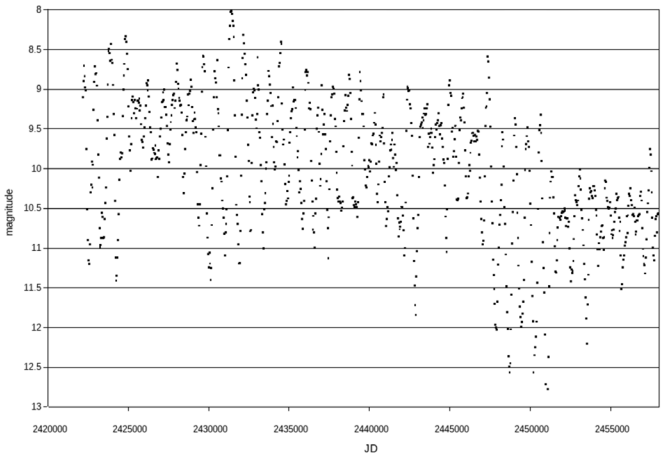

Figure 1a. S Per visual magnitude estimates by experienced observers averaged in 40-day buckets.

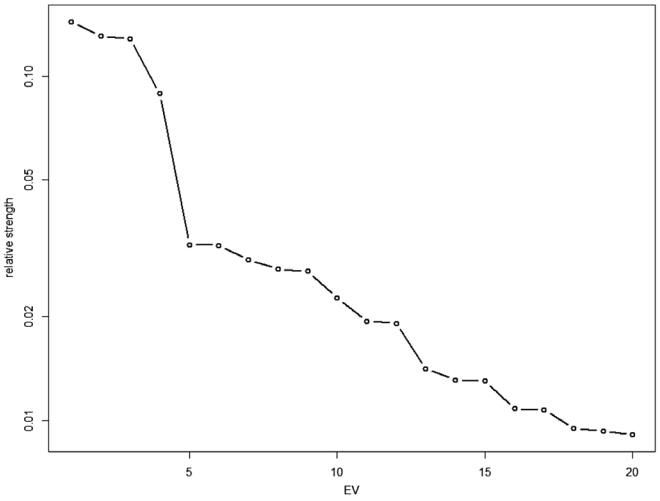

Figure 1b. S Per EV norms using 1d-ssa.

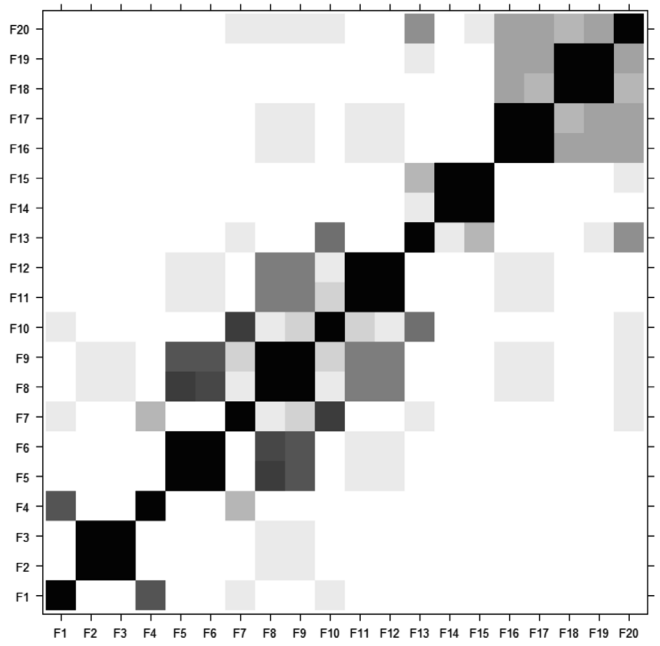

Figure 1c. Per EV series correlations, strong correlation being indicated by a more solid shade.

1c shows EVs 1 and 4 strongly correlated with each other but not the rest; EVs 2 and 3 strongly correlated but again not with the rest; EVs 7, 10, and 13 are largely detached from the rest while remaining EVs 5 to 12 form a block. Figure 1d shows similar behavior in EVs 1 and 4, and similar behavior in EVs 2 and 3, and EVs 5, 6, 8, 9, 11, 12.

Figure 1e shows the data together with the reconstructed signals from EVs 1 and 4, and from EVs 2 and 3 (the mean magnitude has been added back to these series). It is clear that EVs 1 and 4 (dashed line) reflect long-term trends (in particular coping with a shift in magnitude described below) while EVs 2 and 3 represent a 799-day oscillation (calculated separately). The figure shows (solid line) the reconstructed signal from EVs 2,3,5,6,8,9,11,12 which shows virtually the same periodicity as EVs 2 and 3 alone.

It is manifestly clear that the time series is non-stationary, there being a marked fall in brightness starting around 2447000 and being maintained. The relatively abrupt change in magnitude is discussed in Chipps *et al.* (2004), and Sabin and Zijlstra (2006) identify similar abrupt changes in other long-period variable stars. We make the following adjustment in order to produce a time-series which is closer to a stationary one. The adjusted magnitude at time t, m_t, is given by

$$m_t = raw_t - (t - T_t) \times H_t - K_t \quad (4)$$

where *raw* is the observed magnitude and the parameters H and K are given in Table 1.

Table 1. H and K parameters.

Time Period (JD)	*H*	*K*
< 2447000	0	0
2447000 to 2448750	4.87–04	0
2448750 to 2458093 (end)	0	0.853

Figure 1f shows the same data series as Figure 1a after adjustment described above.

Applying Toeplitz decomposition Figure 1g shows a clearer distinction between different eigenvectors, and following similar logic to above we group EVs 1–4, and 5–7 and the reconstructed series are shown in Figure 1h. EV 1–4 has a strong period at 815 days.

4. RZ Cas period variability and δ Scuti variation

4.1. Period variability

RZ Cas (GSC 04317-01793) is a semi-detached Algol-type binary comprising a primary A3V star (Duerbeck and Hänel 1979) and a carbon (Abt and Morrell 1995) or K01V (Maxted *et al.* 1994; Rodriguez *et al.* 2004) star which fills its Roche lobe. Times of minimum (tmin) were taken from the Lichtenknecker database (Frank and Lichtenknecker 1987) and compared with expected times using a linear ephemeris and a period of 1.19525031 days chosen to minimize the variance of the differences of observed tmin minus calculated tmin (O–C). Data was bucketed into 100-day blocks with a small number of missing values linearly interpolated between neighboring values

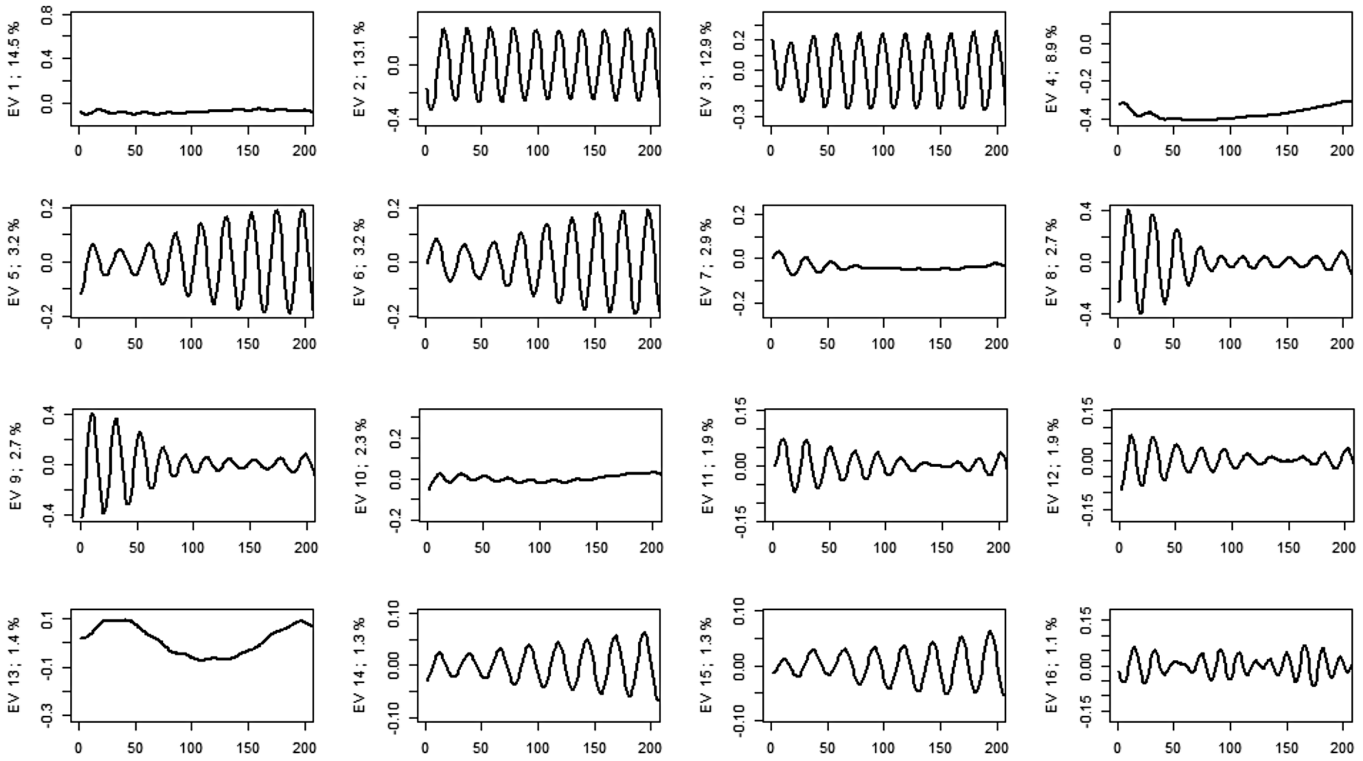

Figure 1d. S Per 1d-ssa first 16 individual EV time series, initial 200 data points – to identify the broad type of pattern.

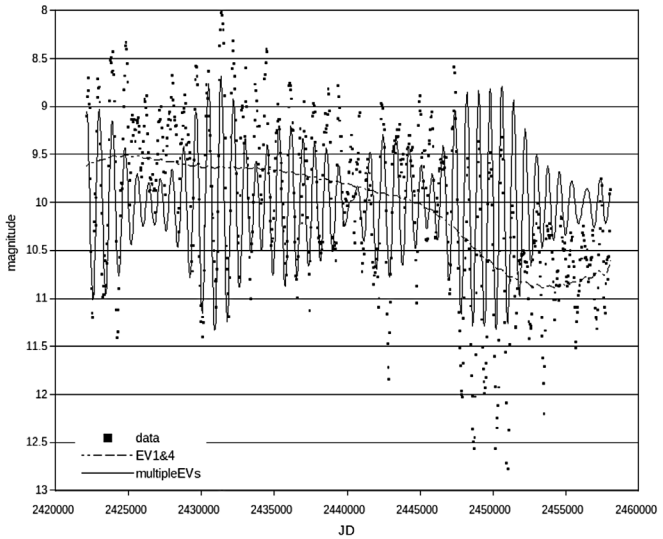

Figure 1e. S Per unadjusted data and reconstructed signals from EV groups using 1d-ssa.

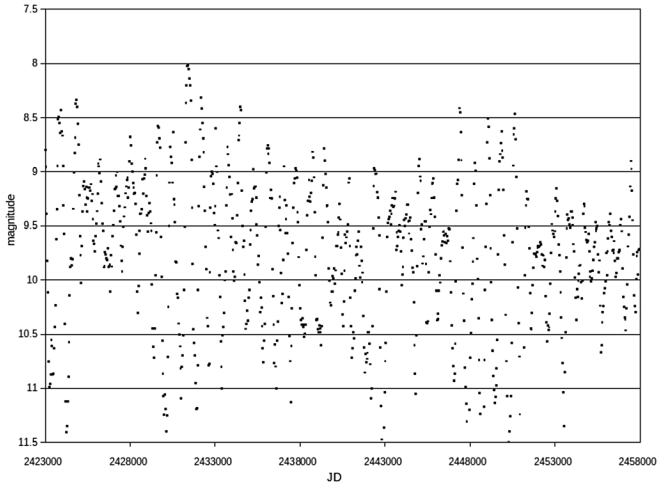

Figure 1f. S Per adjusted data.

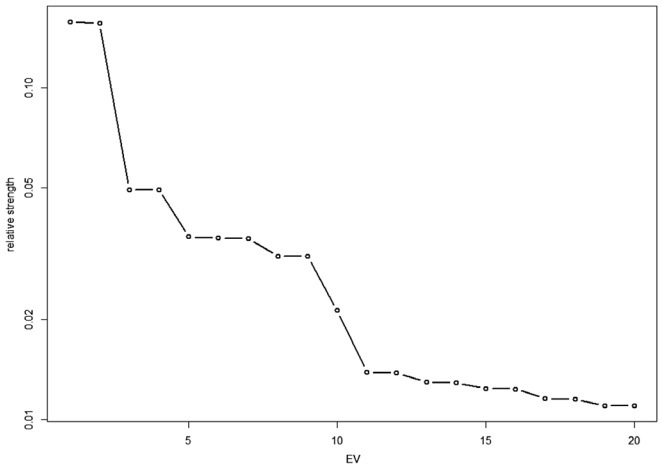

Figure 1g. S Per EV norms using Toeplitz decomposition.

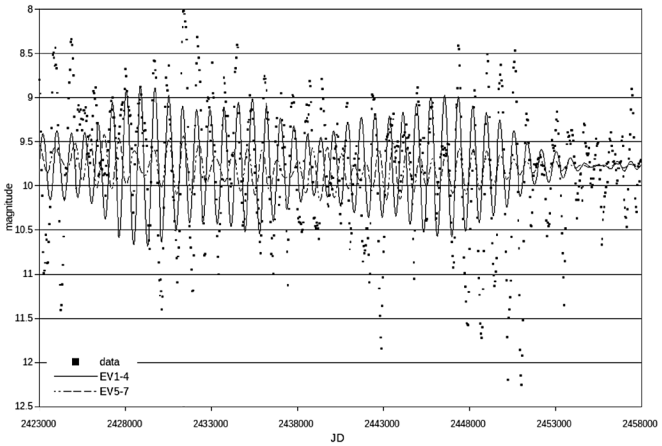

Figure 1h. S Per adjusted data and reconstructed signals from EV groups.

and is shown in Figure 2a.

1d-ssa decomposition was applied, and inspection showed a typical noise pattern after EV6. The correlation matrix and time series charts (and eigenvalue magnitudes) and are shown in Figures 2b (first 10 EVs) and 2c (first 9 EVs).

From these two charts we see EVs 1 and 2 have a small correlation with EV 5 and similar periodicity (and comprise over 90% of the data variation), EVs 3 and 4 are largely separate (comprising 5%) and a similar period but shorter than EVs 1 and 2, EVs 6 and 7 (and several others also with significant correlation) seem to be correcting for abnormalities at the start of the period, and EV 6 and beyond may be regarded as the noise. Figure 2d shows the data and reconstructed signals.

The signal EV3–4 is intriguing: it should be borne in mind that the reconstructed signal is merely an average of the original data series (albeit a very complicated one)—at no point are harmonics used in the calculation, yet this signal is at first glance similar to a sine wave with period just of approximately 23 years. A closer look shows the amplitude of the signal is decreasing and the wavelength is not constant, although this could be a corruption caused by the original noisy data. Furthermore it is known (for example Allen and Smith 1996; Greco *et al.* 2015) that noise other than white noise—in particular noise related to an autoregressive process—can generate spurious periodicities. Further testing, which is beyond the scope of the current paper,

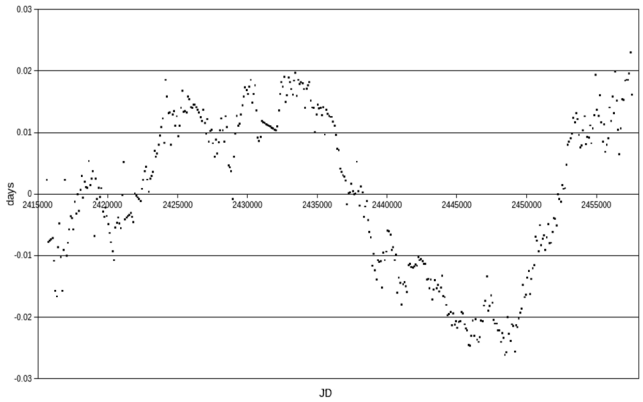

Figure 2a. RZ Cas O-C bucketed into 100-day intervals.

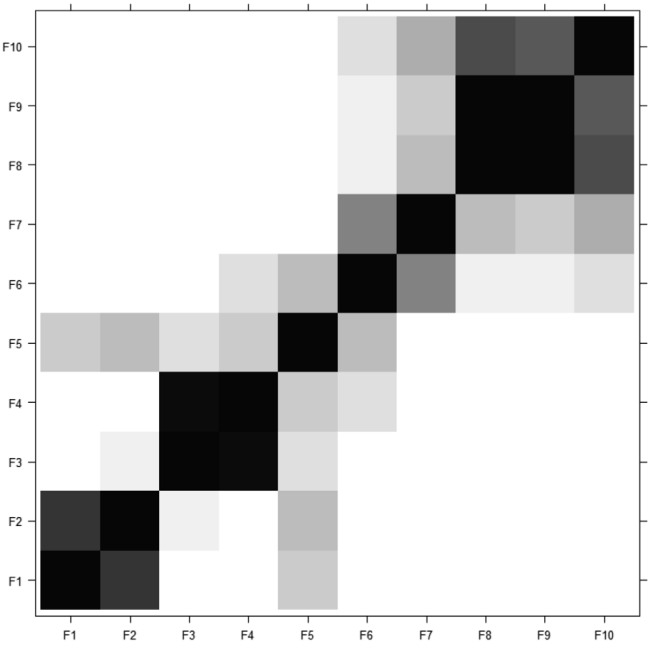

Figure 2b. RZ Cas O-C first 10 EV correlations.

is required to determine whether the observed signal has arisen by chance or from a more complicated underlying non-linear process. If this was indeed harmonic and caused by an orbiting third body then the semi-amplitude of the signal implies an orbit for the main components about a center of mass of the system, and the joint masses of the eclipsing stars would imply a mass of under 0.2 solar mass for an orbiting body.

4.2. δ Scuti-type variation

High frequency CCD observations by G. Samolyk from the AAVSO database were analyzed as follows. Differences from model magnitudes (using a Wilson-Devinney eclipse model (see, for example, Kallrath and Milone 2009)) were calculated and analyzed using 1d-ssa decomposition (Figures 3a and 3b). EVs1–4 and 7, 8 represent the slow deviations, and a relatively strong EV5–6 is independent of other signals apart from the weak 14 and 15, and shows a clear periodicity of 22.4 minutes—in good agreement with Ohshima *et al.* (2001) and Rodriguez *et al.* (2004). The very high frequency variations in EVs 9–12 may be instrumentation related. Figure 3c shows the

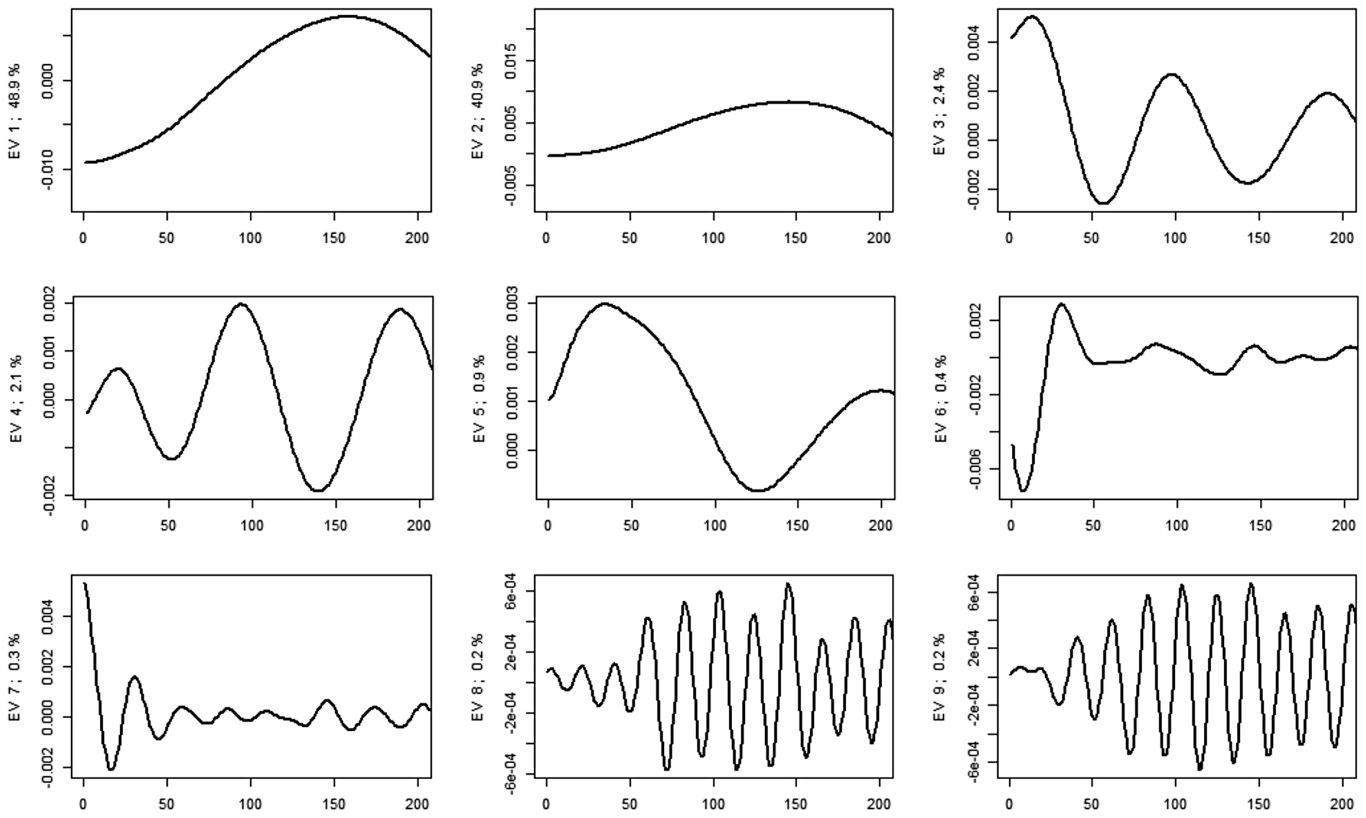

Figure 2c. RZ Cas first 9 individual EV time series, initial 200 data points—to identify the broad type of pattern.

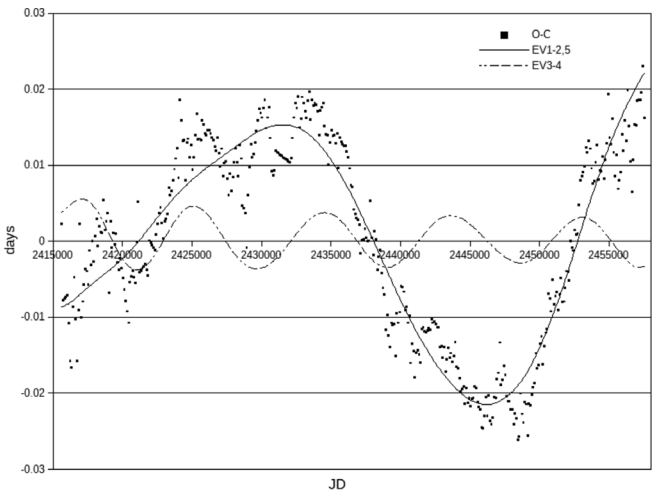

Figure 2d. RZ Cas O–C and reconstructed signals from EV groups.

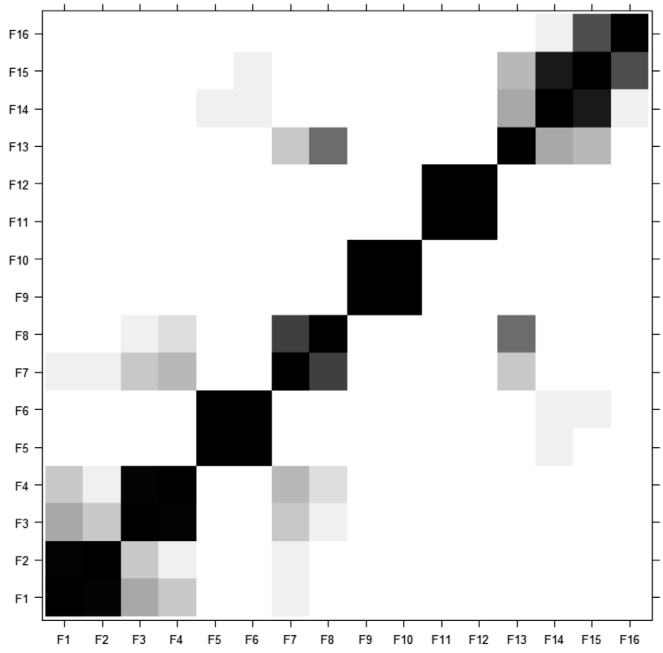

Figure 3a. RZ Cas high-frequency CCD data, EV correlation matrix, first 16 EVs.

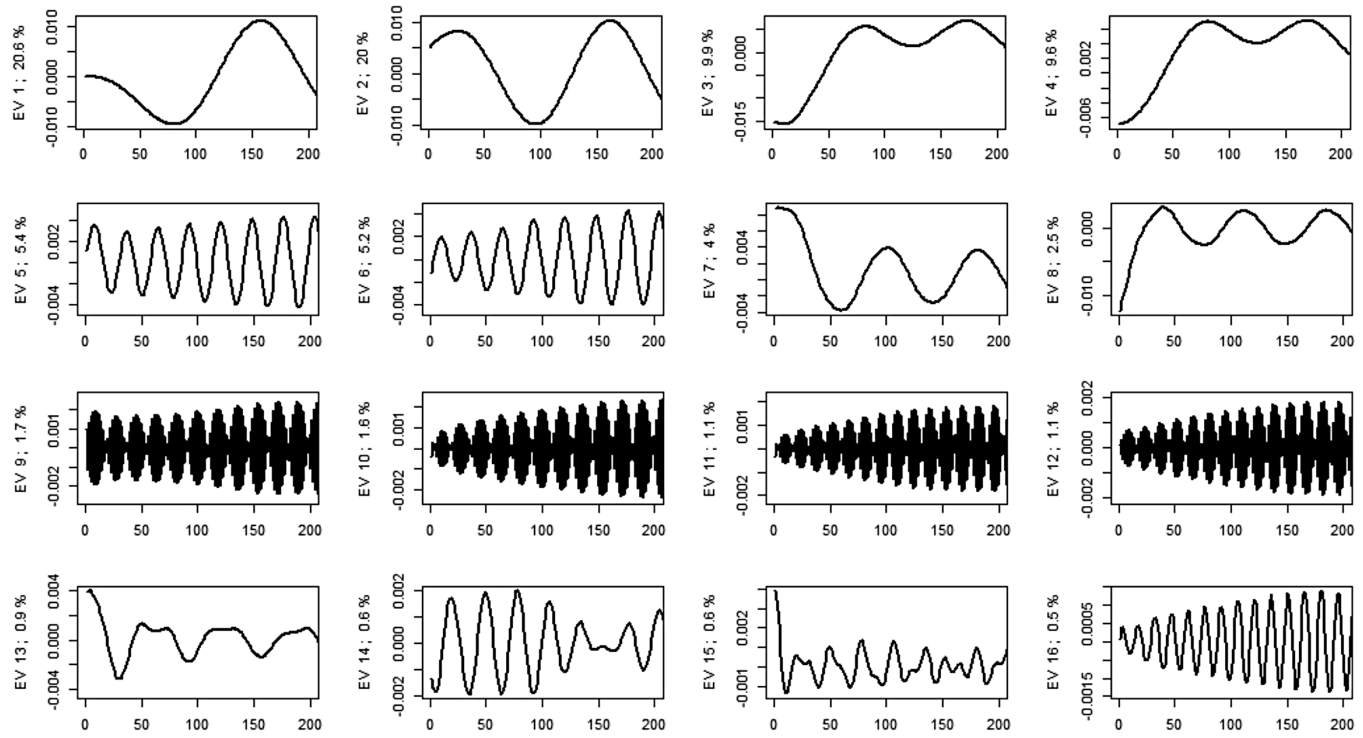

Figure 3b. RZ Cas high-frequency CCD data, first 16 EV time series.

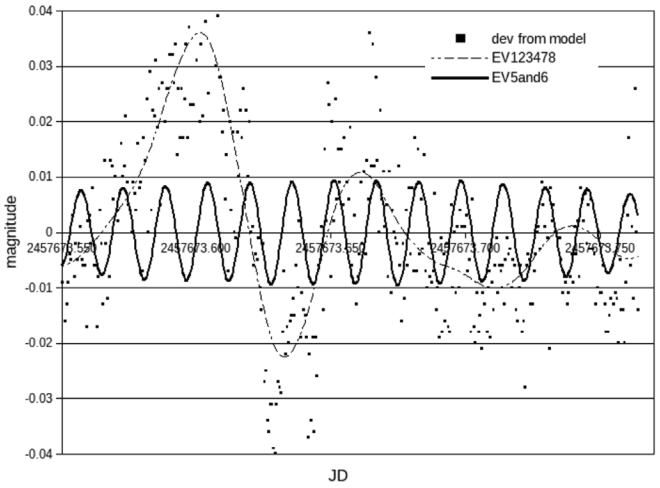

Figure 3c. RZ Cas high-frequency CCD data and reconstructed signal from EV groups.

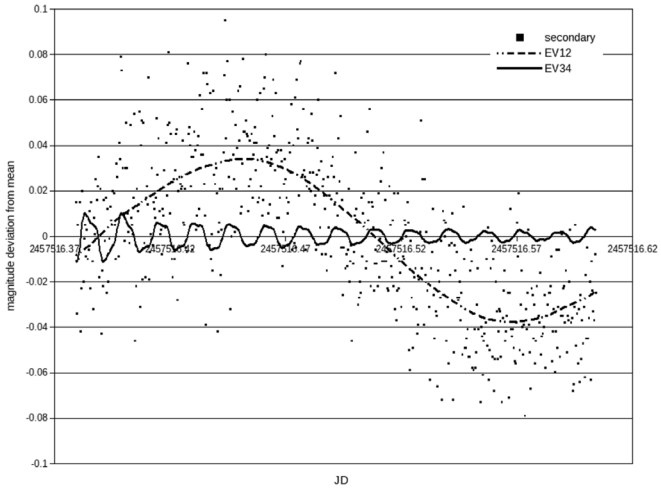

Figure 3d. RZ Cas high-frequency DSLR data and reconstructed signals from EV groups.

data and reconstructed signal.

A second series of relatively noisy DSLR data shows virtually the same periodicity—EV34 shows periodicity of 21.1 minutes. DSLR data by Screech from the BAA database taken through a secondary minimum have been analyzed simply by removing the mean then applying 1d-ssa decomposition and analysis, the result being shown in Figure 3d.

5. Conclusions

In this paper we have given code corresponding closely to the formulae behind singular spectrum analysis as well as code to call the corresponding more efficient "black box" functionality in the Rssa R code package. We have shown how to use the key intermediate output—eigenseries correlation plots and eigenseries time series plots together with their relative strengths—to reconstruct meaningful time series components—trend, periodic and residual noise—of the original time series. As an interactive data driven method this is more revealing, and capable of extracting more information, that typical model driven methods. The S Per data example is one of simple periodicity discovery and is included to illustrate SSA and its application. The RZ Cas O–C series discovers a periodic signal in the times of minimum and hints at a possible third body, while the high frequency data illustrate how the δ Scuti variability can be extracted from relatively noisy data exhibiting strong long-term variation throughout the data sample.

6. Acknowledgements

The author is grateful to the AAVSO, the BAA, and the VSOLJ for providing the observational data used in this analysis. The author would also like to thank Prof. Ray Huffaker for helpful suggestions and encouragement during the production of this paper, and an anonymous referee whose suggestions significantly improved the paper.

References

Abt, H. A., and Morrell, N. 1995, *Astrophys. J., Suppl. Ser.*, **99**, 135.

Allen, M. R., and Smith, L. A. 1996, J. *Climate*, **9**, 3373.

British Astronomical Association Variable Star Section. 2018, BAAVSS online database (http://www.britastro.org/vssdb/).

Broomhead, D. S., and King, G. P. 1986a, *Phys. D: Nonlinear Phenomena*. **20**, 217.

Broomhead, D. S., and King, G. P. 1986b, in *Nonlinear Phenomena and Chaos*, ed. S. Sakar, Adam Hilger, Bristol, 113.

Buchler, J. R., Kollath, Z., Serre, T., and Mattei, J. 1996, *Astrophys. J.*, **462**, 489.

Cattell, R. B. 1965a, *Biometrics*, **21**, 190,

Cattell, R. B. 1965b, *Biometrics*, **21**, 405.

Chipps, K. A., Stencel, R. E., and Mattei, J. A. 2004, *J. Amer. Assoc. Var. Star Obs.*, **32**, 1.

Danilov, D. and Zhigljavsky, A., eds. 1997, *Principal Components of Time Series: the "Caterpillar" Method*, St. Petersburg Univ. Press, Saint Petersburg (in Russian).

Duerbeck, H. W., and Hänel, A. 1979, *Astron. Astrophys. Suppl. Ser.*, **38**, 155.

Fraedrich, K. 1986, *J. Atmos. Sci.*, **4**, 419.

Frank, P. and Lichtenknecer, D. 1987, *BAV Mitt.*, No. 47, 1 (Lichtenknecker database http://www.bav-astro.eu/index.php/veroeffentlichungen/service-for-scientists/lkdb-engl).

Ghil, M., *et al.* 2002, *Rev. Geophys.*, **40**, 1.

Golyandina, N., Korobeynikov, A., and Zhigljavsky, A. 2018, *Singular Spectrum Analysis with R*, Springer-Verlag, Berlin, Heidelberg.

Golyandina, N., Nekrutkin, V., and Zhigljavsky, A. 2001, *Analysis of Time Series Structure*, CRC Press, Boca Raton, FL.

Golyandina, N., and Zhigljavsky, A. 2013, *Singular Spectrum Analysis for Time Series*, Springer-Verlag, Berlin, Heidelberg.

Greco, G., *et al.* 2015, *Astrophys. Space Sci. Proc.*, **42**, 105.

Huffaker, R., Bittelli, M., and Rosa, R. 2017, *Non-linear Time Series Analysis with R*, Oxford Univ. Press, Oxford.

Kafka, S. 2018, variable star observations from the AAVSO International Database (https://www.aavso.org/aavso-international-database).

Kallrath, J., and Milone, E. F. 2009, *Eclipsing Binary Stars: Modeling and Analysis*, Springer-Verlag, Berlin, Heidelberg.

Kiss, L. L., Szabo, Gy. M., and Bedding, T. R. 2006, *Mon. Not. Roy. Astron. Soc.*, **372**, 1721.

Kollath, Z. 1990, *Mon. Not. Roy. Astron. Soc.*, **247**, 377.

Kondrashov, D., and Ghil, M. 2006, *Geophys.*, **13**, 151.

Lang, S. 2013, *Linear Algebra*, 3rd ed., Springer-Verlag, Berlin, Heidelberg.

Maxted, P. F. L., Hill, G., and Hilditch, R. W. 1994, *Astron. Astrophys.*, **282**, 821.

Ohshima, O., *et al.* 2001, *Astron. J.*, **122**, 418.

Purkinje, J. E. 1825, *Neue Beiträge zur Kenntniss des Sehens in Subjectiver Hinsicht*, Reimer, Berlin, 109.

The R Foundation for Statistical Computing. 2018a, R: a language and environment for statistical computing (https://www.R-project.org).

The R Foundation for Statistical Computing. 2018b, CRAN: The Comprehensive R Archive Network (https://cran.r-project.org/mirrors.html).

The R Foundation for Statistical Computing. 2018c, Rssa: a collection of methods for sungular spectrum analysis (http://cran.r-project.org/web/packages/Rssa).

Rodriguez, E., *et al.* 2004, *Mon. Not. Roy. Astron. Soc.*, **347**, 1317.

RStudio. 2018, RSTUDIO software (https://www.rstudio.com).

Sabin, L., and Zijlstra, A. A. 2006, *Mem. Soc. Astron. Ital.*, 77, 933.

Samus N. N., Kazarovets, E. V., Durlevich, O. V., Kireeva, N. N., and Pastukhova E. N. 2017, *Astron. Rep.*, **61**, 80, *General Catalogue of Variable Star*s: version GCVS 5.1 (http://www.sai.msu.su/gcvs/gcvs/index.htm).

Variable Star Observers League in Japan. 2018, VSOLJ variable star observation database (http://vsolj.cetus-net.org/database.html).

Various authors. 2010, *Statistics and its Interface*, **3** (No. 3).

Various authors. 2017, *Statistics and its Interface*, **10** (No. 1).

Vautard, M., and Ghil, M. 1989, *Phys. D: Nonlinear Phenomena*, **35**, 395.

Vautard, M., Yiou, P., and Ghil, M. 1992, *Phys. D: Nonlinear Phenomena*, **58**, 95.

Wenger, M., *et al.* 2000, *Astron. Astrophys., Suppl. Ser.*, **143**, 9.

Zhigljavsky, A. 2010, *Statistics and its Interface*, **3**, 255.

Appendix A: code examples

A.1. SVD code

Notes:

1. We recommend the use of "RSTUDIO" (2018) which provides a simple and highly efficient way of handling R code and results.

2. The user needs to set the path according to where the R system has been installed—see the code comment below.

3. The packages "tseriesChaos" and "Rssa" need to be installed from the "install" tab under "packages" in RSTUDIO.

4. The code should be saved as "XXXX.R" in the "User Defined Function" subdirectory of R when "XXXX" is a user chosen name.

5. Steps 1–6 are present to show what is going on behind the scenes in Step 7—in practical use only Step 7 is needed.

6. Comments are in italics, code in bold.

```
# Code: Basic SSA - matrix decomposition and grouping
rm(list=ls(all=TRUE))

# DEFINE YOUR PATH HERE
setwd("C:/Users/Geoff/Documents/R/data")
# END DEFINE YOUR PATH HERE

# User-defined function for averaging of minor diagonals—from Huffaker et al. (2017) code 6.6
diag.ave<-function(mat, rowCount, colCount) {
  hold<-matrix(0,(rowCount+(colCount-1)))
  for(i in 1:(rowCount+(colCount-1))) {
  if(i==1) {d<-mat[1,1]}

  if(i>1 & i<=colCount) {d<-diag(mat[i:1,1:i])}

  if(i>colCount & i<=rowCount) {d<-diag(mat[i:(i-(colCount-1)),1:colCount])}

  if(i>rowCount & i<(rowCount+(colCount-1))) {
  d<-diag(mat[rowCount:(i-(colCount-1)),(i-(rowCount-1)):colCount])}

  if(i==(rowCount+(colCount-1))) {d<-mat[rowCount,colCount]}

  d.ave<-mean(d) #average minor diagonals
  hold[i,]<-d.ave
  } #end loop
  return(hold)
} #end function

# START START START START START START START START START START START START
# USER INPUT USER INPUT USER INPUT USER INPUT USER INPUT USER INPUT USER

# Read in data
ts<-read.csv("RZ Cas O minus C.csv")
x<-ts$OmCadj #x has ndata rows and 1 col

# dimension (number of columns) of the trajectory matrix
L = 200
# choose the number of eigenvectors (and reconstructed series) required
outputVecCount = 20

# end USER INPUT USER INPUT USER INPUT USER INPUT USER INPUT USER INPUT

# step 1: construct trajectory matrix

library(tseriesChaos)
TM = embedd(x,L,1) #1=delay
ndata=length(TM[,1])

# step 2: lagged covariance matrix

lagCM = t(TM) %*% TM

# step 3: eigensystem of lagCM

eigensys = eigen(lagCM,symmetric=TRUE)
eigenvals = eigensys$values
eigenvecs = eigensys$vectors
eigenSet = cbind(eigenvals,eigenvecs)

orderedSet = order(eigenSet[,1],decreasing=TRUE)

ES = eigenSet[order(eigenSet[,1],decreasing=TRUE),] #sort in order of eigenvalues

# calculate relative strength of EVs

sumLambdas = sum(eigenvals)
relativeEV = matrix(0,nrow=outputVecCount,ncol=1)
for (i in 1:outputVecCount) {relativeEV[i] = abs(ES[i,1])/sumLambdas}

write(relativeEV[1:outputVecCount], file = "BasicSSAdata.csv",
  ncolumns = outputVecCount,append = FALSE, sep = ",")

# PLOT: relative eigenvalue plots

plot(relativeEV[1:outputVecCount],log="y",type="b",col="black",lwd=2)

# calculate left eigenvectors of the trajectory matrix

left = matrix(0,nrow=ndata,ncol=outputVecCount)
for(i in 1:outputVecCount){
  left[,i] = TM %*% ES[,1+i]/sqrt(ES[i,1])
}


# step 4: now get the decomposition of the TM (trajectory matrices projected
# on important eigenvectors)

X = array(1:ndata*L*outputVecCount,dim=c(ndata,L,outputVecCount))
for(i in 1:outputVecCount){
 X[,,i] = sqrt(ES[i,1]) * left[,i] %*% t(ES[,1+i])
}

# step 5: reconstructed individual time-series (diagonal averaging)

actualNdata = ndata+L-1
recon = matrix(0,nrow=actualNdata,ncol=outputVecCount)
for (i in 1:outputVecCount) {
 recon[,i] = diag.ave(X[,,i],ndata,L)
}

# PLOT: plot of correlations

w<-cor(recon,y=NULL,use="everything",method="pearson")
library(corrplot)
corrplot(w,method="square")

# PLOT: miniplot of recon time series related to each EV

plotRow = round(sqrt(outputVecCount))
par(mfrow=c(plotRow,outputVecCount/plotRow))
for (i in 1:outputVecCount){
  plot(recon[,i],xlim=c(1,200),xlab="",
  ylab=paste("series ",toString(i),"; ",toString(round(1000*relativeEV
[i])/10),"%"),
  type="l",col="black",lwd=2) #plot 1st 20 time series for 200 periods
}
```

```
# write time series output

   write(t(recon), file = "BasicSSAdata.csv",#tmp
   ncolumns = outputVecCount,append = TRUE, sep = ",")
```

A.2. Toeplitz code

Steps 1 and 2 in the above are replaced with the following:

```
#step 1: construct trajectory matrix
zero = seq(0,0,length.out=ndata) #used for padding
TM = matrix(0,nrow=ndata,ncol=L)
TM = cbind(x,append(x[2:ndata], zero[1:1],after=ndata-1))
for(j in 3:L){
   TM = cbind(TM,append(x[j:ndata],zero[1:j-1],after=ndata-j+1))
}

#step 2: lagged covariance matrix
lagCM = matrix(data=NA,nrow=L,ncol=L)
for(i in 1:L){
   for(j in 1:L){
   xsum = 0
   for (t in 1:(ndata-abs(i-j))){
   xsum = xsum + x[t]*x[(t+abs(i-j))]
   }
   xsum = xsum / (ndata-abs(i-j))
   lagCM[i,j] <- xsum
   }
}
```

A.3. Rssa code

```
#Code 6.9 from Huffaker et al. (2017), SSA: matrix decomposition and grouping
diagnostics
rm(list=ls(all=TRUE))

#Read in data
setwd("C:/Users/Geoff/Documents/R/data")
ts<-read.csv("RZ Cas O minus C.csv");
x<-ts$OmCadj
n = length(x)

#SSA Decomposition
#load Rssa R library from Install Packages
library(Rssa)
L=200
s<-ssa(x,L,kind="1d-ssa") #run Rssa 1d-ssa
#s<-ssa(x,L,kind="toeplitz-ssa") # alternatively run Rssa Toeplitz-ssa

#Run grouping diagnostics to group eigentriplets
   #First visual diagnostic: Eigenspectrum
plot(s,numvalues=20,col="black",lwd=2) #plot 1st 20 largest
eigenvalues<-plot(s,numvalues=20,col="black",lwd=2)

   #Second visual diagnostic: Eigenvector plots
plot(s,type="vectors",idx=1:20,xlim=c(1,200),col="black",lwd=2) #plot
1st 20 for 300 periods

   #Weighted correlation matrix
plot(w<-wcor(s,groups=c(1:19))) #1st 20 eigentriplets
w.corr.res<-wcor(s,groups=c(1:20)) #table for 1st 10 eigentriplets

# write time series output
r.1<-reconstruct(s,groups=li
st(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20))
recon.1<-r.1$F1
recon.2<-r.1$F2
recon.3<-r.1$F3
recon.4<-r.1$F4
recon.5<-r.1$F5
recon.6<-r.1$F6
recon.7<-r.1$F7
recon.8<-r.1$F8
recon.9<-r.1$F9
recon.10<-r.1$F10
recon.11<-r.1$F11
recon.12<-r.1$F12
recon.13<-r.1$F13
recon.14<-r.1$F14
recon.15<-r.1$F15
recon.16<-r.1$F16
recon.17<-r.1$F17
recon.18<-r.1$F18
recon.19<-r.1$F19
recon.20<-r.1$F20

tmp = vector("numeric",20)
write(c(1:20), file = "BasicRssadata.csv",ncolumns = 20,append = FALSE,
sep = ",")
for (i in 1:n) {
   tmp = c(recon.1[i],recon.2[i],recon.3[i],recon.4[i],recon.5[i],
   recon.6[i],recon.7[i],recon.8[i],recon.9[i],recon.10[i],
   recon.11[i],recon.12[i],recon.13[i],recon.14[i],recon.15[i],
   recon.16[i],recon.17[i],recon.18[i],recon.19[i],recon.20[i])
   write(t(tmp), file = "BasicRssadata.csv",ncolumns = 20,append = TRUE,
sep = ",")
}
```