Timescale-specfic variance ratio (tsvr) package vignette

Lei Zhao, Shaopeng Wang, Daniel Reuman

The `tsvr` package provides an implementation of a timescale-specific extension and generalization of the variance ratio of Peterson (1975). The variance ratio is used commonly in community ecology. The extension implemented in the `tsvr` package is described in detail by Zhao et al. (In prep). The `tsvr` package supports that paper and provides an implemetation of the tools developed there for anyone to use. The mathematical formulas for the variance ratio and extensions are detailed elsewhere (Peterson 1975; L. Hallett et al. 2014; Zhao et al. In prep). The mathematics are also summarized here, but the main purpose of this vignette is to provide a decription of how to use the `tsvr` package.

1 Preparing data

A typical dataset for analysis using `tsvr` is an $N \times T$ matrix of nonnegative numeric values where rows correspond to species in a community (so the number of species is N) and columns correspond to evenly spaced times during which sampling was conducted (so the number of times sampling was conducted is T). Matrix entries may be densities, or percent cover values for plant species within a quadrat, or biomasses, or other measures of abundance of the species. For instance:

```
library(tsvr)
class(JRGdat)
```

```
## [1] "data.frame"
```

```
names(JRGdat)
```

```
##  [1] "year"                        "agoseris.heterophylla"
##  [3] "astragalus.gambelianus"      "bombycilaena.californica"
##  [5] "brodiaea.sp"                 "bromus.berteroanus"
##  [7] "bromus.hordeaceus"           "calandrinia.ciliata"
##  [9] "calycadenia.multiglandulosa" "castilleja.densiflora"
## [11] "chlorogalum.pomeridianum"    "crassula.connata"
## [13] "elymus.multisetus"           "epilobium.brachycarpum"
## [15] "hemizonia.congesta"          "hesperevax.sparsiflora"
## [17] "lasthenia.californica"       "layia.platyglossa"
## [19] "lepidium.nitidum"            "lotus.wrangelianus"
## [21] "microseris.douglasii"        "nassella.pulchra"
## [23] "plantago.erecta"             "poa.secunda"
## [25] "trifolium.albopurpureum"     "trifolium.sp"
## [27] "vulpia.microstachys"
```

```
d<-t(as.matrix(JRGdat[,2:dim(JRGdat)[2]]))
dim(d)
```

```
## [1] 26 28
```

Here `d` is a 26×28 matrix containing percent cover measurements for each year from 1983 to 2010 for 26 species which occurred in a 1 m^2 plot in the Jasper Ridge Biological Preserve serpentine grassland site. Species names are in the row names of `JRGdat`, which is an example dataset embedded in the `tsvr` package. Documentation for the dataset can be viewed via `?JRGdat`. See (L. Hallett et al. 2014) for details of the Jasper Ridge ecosystem and of these and other related data.

Standard implementations of Fourier transforms require time series consisting of measurements taken at evenly spaced times, with no missing data. The core functions provided in `tsvr` make these same assumptions and throw an error if data are missing. The user is left to decide on and implement a reasonable way of filling missing data, if data are missing. We have previously used the simple approach of replacing missing values in a time series by the median of the non-missing values in the time series (Sheppard et al. 2016). This approach, and other related simple procedures (Sheppard et al. 2016), seem unlikely to artefactually produce significant synchrony, or coherence relationships with other variables, but rely on the percentage of missing data being fairly low and may obscure detection of synchrony or significant coherence relationships if too many data are missing. For applications which differ meaningfully from the prior work for which the tools of this package were developed (Zhao et al. In prep), different ways of filling missing data may be more appropriate.

The timescale-specific variance ratio techniques which are the focus of this package use Fourier methods to decompose by timescale the classic variance ratio and related quantities. Detrending and variance standardization across time series, techniques which are often applied before doing Fourier analysis, may not be approriate except in cases for which it makes sense to calculate the classic variance ratio and related quantities after performing those techniques.

2 The classic variance ratio and related quantities

Let $x_i(t)$ be a measure of the population of species i at time t, for $i = 1, \ldots, N$ and $t = 1, \ldots, T$. Define $x_{\text{tot}}(t) = \sum_{i=1}^{N} x_i(t)$ to be the total of all species populaton measures. Define CV^2_{com} to be the square of the coefficient of variation of $x_{\text{tot}}(t)$ over time, i.e., $\text{var}_t(x_{\text{tot}}(t))/(\text{mean}_t(x_{\text{tot}}(t)))^2$, where var_t and mean_t represent variance and mean computed through time. This equals $\sum_{i,j} \text{cov}_t(x_i(t), x_j(t))/(\text{mean}_t(x_{\text{tot}}(t)))^2$, where cov_t is covariance through time. The abbreviation "com" in CV^2_{com} stands for "community" since CV^2_{com} is the squared coefficient of variation of the whole-community population. Define $\text{CV}^2_{\text{com\_ip}}$ to be the value of CV^2_{com} that would pertain if the population dynamics of different species were independent, so that $\text{cov}_t(x_i(t), x_j(t)) = 0$ for all $i \neq j$. The abbreviation "ip" stands for "independent populations". Thus $\text{CV}^2_{\text{com\_ip}} = \sum_i \text{cov}_t(x_i(t), x_i(t))/(\text{mean}_t(x_{\text{tot}}(t)))^2 = \sum_i \text{var}_t(x_i(t))/(\text{mean}_t(x_{\text{tot}}(t)))^2$. The classic variance ratio is defined as $\varphi = \text{CV}^2_{\text{com}}/\text{CV}^2_{\text{com\_ip}}$, so that $\text{CV}^2_{\text{com}} = \varphi \text{CV}^2_{\text{com\_ip}}$. A variance ratio greater than 1 suggests "synchronous" dynamics of the species comprising the community, so that community variability (CV^2_{com}) is greater than it would be if populations were independent ($\text{CV}^2_{\text{com\_ip}}$). A variance ratio less than 1 suggests "compensatory" dynamics of the species comprising the community (i.e., increases/decreases in some species are partly compensated for by decreases/increases in others), so that community variability is less than it would be if populations were independent.

The quantities CV^2_{com}, $\text{CV}^2_{\text{com\_ip}}$ and φ can be computed using the `vreq_classic` function in the `tsvr` package, we here do so for the dataset `d` of the previous section, which we will continue to use throughout this vignette:

```r
res<-vreq_classic(d)
class(res)
```

```
## [1] "vreq_classic" "vreq"         "list"
```

```r
names(res)
```

```
## [1] "com"     "comnull" "vr"
```

```r
summary(res)
```

```
## class: vreq_classic
## com: 0.04695127
## comnull: 0.1028208
## vr: 0.4566318
```

```r
print(res)
```

```
## Object of class vreq_classic:
##  CVcom2: 0.0469512705521134
##  CVcomip2: 0.102820846939826
##  classic vr: 0.456631820778433
```

```
all.equal(res$com,res$comnull*res$vr)
```

```
## [1] TRUE
```

The `vreq_classic` S3 class, of which `vreq_classic` is the generator function, inherits from the generic S3 class `vreq` (generator function `vreq`) and the `list` class, and has the three slots `com`, `comnull`, and `vr`. These slots correspond to CV^2_{com}, $\text{CV}^2_{\text{com\_ip}}$ and φ. Both the `vreq` and `vreq_classic` classes have `print` and `summary` methods and `set_*` and `get_*` methods where `*` represents any of the class slot names. See documentation for `vreq`, `vreq_classic`, `vreq_methods`, `vreq_classic_methods`, `setget_methods` for details. The "classic" in `vreq_classic` references the fact that this version of the variance ratio is the original version used in community ecology (Peterson 1975), and is probably still the most commonly used. Alternative versions have been proposed (Loreau and Mazancourt 2008) - see the next section of this vignette.

3 The Loreau-de Mazancourt variance ratio and related quantities

Define $\text{CV}^2_{\text{pop}} = \left(\sum_i \sqrt{\text{var}_t(x_i(t))}\right)^2 / (\text{mean}_t(x_{\text{tot}}(t)))^2$. Another version of the variance ratio has been proposed by Loreau and de Mazancourt: $\varphi_{\text{LdM}} = \text{CV}^2_{\text{com}}/\text{CV}^2_{\text{pop}}$. Thus $\text{CV}^2_{\text{com}} = \varphi_{\text{LdM}}\text{CV}^2_{\text{pop}}$. See (Loreau and Mazancourt 2008) for details. The `tsvr` package implements the Loreau-de Mazancourt approach:

```
res<-vreq_LdM(d)
class(res)
```

```
## [1] "vreq_LdM" "vreq"     "list"
```

```
names(res)
```

```
## [1] "com"     "comnull" "vr"
```

```
summary(res)
```

```
## class: vreq_LdM
## com: 0.04695127
## comnull: 0.836276
## vr: 0.05614327
```

```
print(res)
```

```
## Object of class vreq_LdM:
##  CVcom2: 0.0469512705521134
##  CVpop2: 0.836276015586239
##  LdM vr: 0.0561432704956868
```

```
all.equal(res$com,res$comnull*res$vr)
```

```
## [1] TRUE
```

```
all.equal(res$com,vreq_classic(d)$com)
```

```
## [1] TRUE
```

The `vreq_LdM` S3 class, of which `vreq_LdM` is the generator function, inherits from the generic S3 class `vreq` and the `list` class, and has slots `com`, `comnull`, and `vr`. These slots correspond to CV^2_{com}, CV^2_{pop} and φ_{LdM}. The class `vreq_LdM` has `print` and `summary` methods and `set_*` and `get_*` methods where `*` represents any of the class slot names. See documentation for `vreq_LdM`, `vreq_LdM_methods`, and `setget_methods` for details.

4 The timescale-specific classic variance ratio and related quantities

Next we describe a timescale-specific extension of the classic variance ratio, and its related quantities, that uses spectral methods, a set of standard statistical tools in ecology and other fields. The power spectrum of the time series $x_i(t)$, here denoted $s_{ii}(\sigma)$ and defined for timescales $\sigma = T/(T-1), T/(T-2), \ldots, T/2, T$, decomposes $\mathrm{var}_t(x_i(t))$ by timescale in that $s_{ii}(\sigma)$ is nonnegative, will tend to be larger for timescales on which $x_i(t)$ shows greater variation through time, and $\sum_\sigma s_{ii}(\sigma) = \mathrm{var}_t(x_i(t))$. Likewise, the cospectrum $s_{ij}(\sigma)$ decomposes $\mathrm{cov}_t(x_i(t), x_j(t))$ by timescale in a similar way, being larger for timescales on which the two time series predominantly covary.

We define a timescale-specific generalization of $\mathrm{CV}^2_{\mathrm{com}}$ as $\mathrm{CV}^2_{\mathrm{com}}(\sigma) = \sum_{i,j} s_{ij}(\sigma)/(\mathrm{mean}_t(x_{\mathrm{tot}}(t)))^2$. It is straightforward to see that $\sum_\sigma \mathrm{CV}^2_{\mathrm{com}}(\sigma) = \mathrm{CV}^2_{\mathrm{com}}$, so $\mathrm{CV}^2_{\mathrm{com}}(\sigma)$ decomposes $\mathrm{CV}^2_{\mathrm{com}}$ by timescale. $\mathrm{CV}^2_{\mathrm{com}}(\sigma)$ reveals to what extent variation on each timescale contributes to community variability. Likewise, $\mathrm{CV}^2_{\mathrm{com\_ip}} = \sum_i \mathrm{var}_t(x_i(t))/(\mathrm{mean}_t(x_{\mathrm{tot}}(t)))^2$ can be decomposed by timescale as $\mathrm{CV}^2_{\mathrm{com\_ip}}(\sigma) = \sum_i s_{ii}(\sigma)/(\mathrm{mean}_t(x_{\mathrm{tot}}(t)))^2$. Finally, we define a timescale-specific version of the classic variance ratio as the quotient of these two quantities, i.e., $\varphi_{ts}(\sigma) = \left(\sum_{i,j} s_{ij}(\sigma)\right) / \left(\sum_i s_{ii}(\sigma)\right)$, so that $\mathrm{CV}^2_{\mathrm{com}}(\sigma) = \varphi_{ts}(\sigma)\mathrm{CV}^2_{\mathrm{com\_ip}}(\sigma)$. The timescale-specific variance ratio quantifies the extent to which species' oscillations are synchronous (> 1) or compensatory (< 1) on a timescale-by-timescale basis. For further details, see (Zhao et al. In prep).

The quantities $\mathrm{CV}^2_{\mathrm{com}}(\sigma)$, $\varphi_{ts}(\sigma)$ and $\mathrm{CV}^2_{\mathrm{com\_ip}}(\sigma)$ can be computed using the `tsvr` package:

```r
res<-tsvreq_classic(d)
class(res)
```

```
## [1] "tsvreq_classic" "tsvreq"         "list"
```

```r
names(res)
```

```
## [1] "ts"      "com"     "comnull" "tsvr"    "wts"
```

```r
summary(res)
```

```
## class: tsvreq_classic
## ts_start: 1.037037
## ts_end: 28
## ts_length: 27
## com_length: 27
## comnull_length: 27
## tsvr_length: 27
## wts_length: 27
```

```r
print(res)
```

```
## Object of class tsvreq_classic:
##  ts, a length 27 numeric vector: 1.04 1.08 1.12 ... 9.33 14 28
##  CVcom2, a length 27 numeric vector: 0.00221 0.000986 0.00179 ... 0.00179 0.000986 0.00221
##  CVcomip2, a length 27 numeric vector: 0.00927 0.00291 0.018 ... 0.018 0.00291 0.00927
##  tsvr, a length 27 numeric vector: 0.238 0.339 0.0995 ... 0.0995 0.339 0.238
##  wts, a length 27 numeric vector: 0.0901 0.0283 0.175 ... 0.175 0.0283 0.0901
```

```r
all.equal(res$com,res$tsvr*res$comnull)
```

```
## [1] TRUE
```

```r
all.equal(sum(res$com),vreq_classic(d)$com)
```

```
## [1] TRUE
```

```r
all.equal(sum(res$comnull),vreq_classic(d)$comnull)
```

```
## [1] TRUE
```

Here, `ts` is a vector of timescales $(T/(T-1), T/(T-2), \dots, T/2, T)$, and the other elements are vectors of the same length containing timescale-specific information. The element `com` contains $\text{CV}^2_{\text{com}}(\sigma)$, `comnull` contains $\text{CV}^2_{\text{com\_ip}}(\sigma)$, and `tsvr` contains $\varphi_{ts}(\sigma)$. The `tsvreq_classic` S3 class inherits from the generic class `tsvreq` and from the `list` class.

The timescale-specific variance ratio, $\varphi_{ts}(\sigma)$, is not a decomposition of φ_{ts}, i.e., summing $\varphi_{ts}(\sigma)$ across timescales does not yield φ. However, defining $w(\sigma) = \sum_i s_{ii}(\sigma)/(\sum_i \text{var}_t(x_i(t)))$, one can show $\sum_\sigma w(\sigma) = 1$, so the $w(\sigma)$ are weights, and $\sum_\sigma w(\sigma)\varphi_{ts}(\sigma) = \varphi$. So φ is a weighted average of the values of $\varphi_{ts}(\sigma)$ across timescales. The `wts` field of a `tsvreq_classic` object contains $w(\sigma)$.

```
sum(res$wts)
```

```
## [1] 1
```

```
res2<-vreq_classic(d)
all.equal(sum(res$tsvr),res2$vr)
```

```
## [1] "Mean relative difference: 0.9794545"
```

```
all.equal(sum(res$wts*res$tsvr),res2$vr)
```

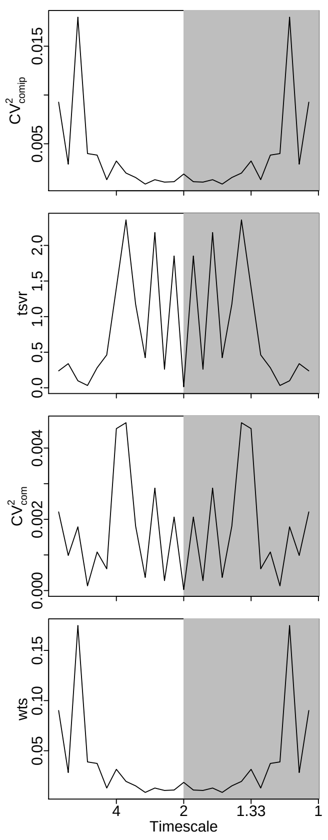
```
## [1] TRUE
```

The plot method for the `tsvreq_classic` class displays the various components as functions of timescale:

```
plot(res,filename="Tsvreq_classic_demo")
```

```
## pdf
##   2
```

```
knitr::include_graphics("Tsvreq_classic_demo.pdf")
```



The plots are symmetric about the middle, because Fourier transforms of real-valued time series have this property. The middle timescale is associated with the Nyquist frequency. The gray shading is a reminder of the symmetry - one should typically interpret the left, un-grayed side of plots. Both symmetric sides are plotted because sums must be computed over all displayed timescales to equal the corresponding frequency-nonspecific analogues. Additionally, non-smoothed Fourier transforms are used so that sums will exactly equal frequency-nonspecific analogues. Unsmoothed Fourier spectra and cospectra are jagged, so peaks of plots should not be given undue interpretive weight unless smoothing or averaging over timescales (see below) or significance testing is performed.

5 The lack of a timescale-specific Loreau-de Mazancourt variance ratio

It is difficult to envision an analogous timescale-specific version of the Loreau-de Mazancourt approach. The quantity $\mathrm{CV}^2_{\mathrm{pop}} = \left(\sum_i \sqrt{\mathrm{var}_t(x_i(t))}\right)^2 / (\mathrm{mean}_t(x_{\mathrm{tot}}(t)))^2$ cannot be decomposed by replacing the variances in the numerator by power spectra because of the square root.

6 Aggregating the timescale-specific classic variance ratio and related quantities to timescale bands

If Ω is a set of timescales, and defining $\mathrm{CV}^2_{\mathrm{com}}(\Omega) = \sum_{\sigma \in \Omega} \mathrm{CV}^2_{\mathrm{com}}(\sigma)$, $\mathrm{CV}^2_{\mathrm{com\_ip}}(\Omega) = \sum_{\sigma \in \Omega} \mathrm{CV}^2_{\mathrm{com\_ip}}(\sigma)$, and $\bar{\varphi}_{ts}(\Omega) = \frac{\sum_{\sigma \in \Omega} \varphi_{ts}(\sigma) w(\sigma)}{\sum_{\sigma \in \Omega} w(\sigma)}$, it has been shown (Zhao et al. In prep) that $\mathrm{CV}^2_{\mathrm{com}}(\Omega) = \bar{\varphi}_{ts}(\Omega)\mathrm{CV}^2_{\mathrm{com\_ip}}(\Omega)$. Aggregating over timescales mitigates the jaggedness resulting from unsmoothed Fourier transforms (see above). The `tsvr` package provides tools for aggregating over any collection of timescales:

```
res<-tsvreq_classic(d)
aggresLong<-aggts(res,res$ts[res$ts>=4])
aggresShort<-aggts(res,res$ts[res$ts<4])
class(aggresLong)
```

```
## [1] "vreq_classic_ag" "vreq"            "list"
```

```
names(aggresLong)
```

```
## [1] "com"     "comnull" "vr"      "ts"
```

```
print(aggresLong)
```

```
## Object of class vreq_classic_ag:
##  CVcom2: 0.0227087709439967
##  CVcomip2: 0.0850891648065253
##  vr: 0.266882052440303
##  ts: 1.04 1.08 1.12 ... 9.33 14 28
```

```
print(aggresShort)
```

```
## Object of class vreq_classic_ag:
##  CVcom2: 0.0242424996081167
##  CVcomip2: 0.0177316821333008
##  vr: 1.36718555103062
##  ts: 1.4 1.47 1.56 ... 2.8 3.11 3.5
```

The `aggts` function is the generator function for the `vreq_classic_ag` class, which inherits from the `vreq` class and from `list`.

Note that the best way to specify the timescales over which to aggregate is by using conditions such as `aggts(res,res$ts[res$ts<4])`. If you type in timescales, e.g., `aggts(res,c(1.03,1.07,1.12))`, and the timescales do not match *exactly* with timescales in `res$ts`, they will be removed. No error or warning will be triggered unless there are no remaining timescales. The reason for this is, the code removes all timescales that are not among the canonical Fourier timescales less than 2, the Nyquist timescale. Remaining timescales are then reflected about the Nyquist timescale to account for the symmetry of Fourier transforms. This setup makes it possible to specify, say, aggregation over timescales less than 4 by `aggts(res,res$ts[res$ts<4])` instead of `aggts(res,res$ts[res$ts<4 & res$ts>4/3])` (which gives the same results if run, but is an inconvenient format with which to specify timescales). In other words, only timescales greater than or equal to the Nyquist timescale (2) need be specified in the argument `ts`, and the symmetric timescales on the other side of the Nyquist timescale are included automatically. See also the documentation for `aggts`.

7 Acknowledgements

This material is based upon work supported by the National Science Foundation under grant numbers 1714195 and 1442595, by the James S McDonnell Foundation, and by a working group grant from the Long Term Ecological Research Network Communications Office of the National Center for Ecological Analysis and Synthesis. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these funders. We thank all users of the package who have reported or will later report ways in which the package could be improved.

References

Hallett, LM, JS Hsu, EE Cleland, SL Collins, TL Dickson, EC Farrer, LA Gherardi, et al. 2014. "Biotic Mechanisms of Community Stability Shift Along a Precipitation Gradient." *Ecology* 95: 1693–1700.

Loreau, M, and C de Mazancourt. 2008. "Species Synchrony and Its Drivers: Neutral and Nonneutral Community Dynamics in

Fluctuating Environments." *American Naturalist* 172: E48–66.

Peterson, CH. 1975. "Stability of Species and of Community for the Benthos of Two Lagoons." *Ecology* 56: 958–65.

Sheppard, LW, J Bell, R Harrington, and DC Reuman. 2016. "Changes in Large-Scale Climate Alter Spatial Synchrony of Aphid Pests." *Nature Climate Change* 6: 610–13.

Zhao, L, S Wang, L Hallett, A Rypel, MCN Castorani, LG Shoemaker, KL Kottingham, K Suding, and DC Reuman. In prep. "Decomposition of the Variance Ratio Illuminates Timescale-Specific Population and Community Variabilty." *In Prep*.