An Introduction to dplR

Andy Bunn Mikko Korpela

Processed with dplR 1.7.2 in R version 4.0.3 (2020-10-10) on January 29, 2021

Abstract

This document describes basic features of dplR by following the initial steps that an analyst might follow when working with a new tree-ring data set. The vignette starts with reading in ring widths and plotting them. We describe a few of the available methods for detrending and then show how to extract basic descriptive statistics. We show how to build and plot a simple mean-value chronology. We also show how to build a chronology using the expressed population signal from the detrended ring widths as an example of how more complicated analysis can be done using dplR.

Contents

1 Introduction **2**
1.1 What is Covered 2
1.2 Citing dplR and R 2

2 Working with Ring-Width Data **3**
2.1 Reading Data 3
2.2 Describing and Plotting Ring-Width Data 4

3 Detrending **5**
3.1 Common Detrending Methods 7
3.2 Other Detrending Methods 10

4 Descriptive Statistics **10**

5 Building a Mean Value Chronology **13**

6 Prospectus **13**

1 Introduction

1.1 What is Covered

The Dendrochronology Program Library in R (dplR) is a package for dendrochronologists to handle data processing and analysis. This document gives just a brief introduction of some of the most commonly used functions in dplR. There is more detailed information available in the help files and in the literature (Bunn, 2008, 2010).

In this vignette, we will walk through the most basic activities of working with tree-ring data in roughly the order that a user might follow. E.g., reading data, detrending, chronology building, and doing preliminary exploratory data analysis via descriptive statistics.

1.2 Citing dplR and R

The creation of dplR is an act of love. We enjoy writing this software and helping users. However, neither of us is among the idle rich. Alas. We have jobs and occasionally have to answer to our betters. There is a nifty `citation` function in R that gives you information on how to best cite R and, in many cases, its packages. We ask that you please cite dplR and R appropriately in your work. This way when our department chairs and deans accuse us of being dilettantes we can point to the use of dplR as a partial excuse.

```
> citation()

To cite R in publications use:

  R Core Team (2020). R: A language and environment
  for statistical computing. R Foundation for
  Statistical Computing, Vienna, Austria. URL
  https://www.R-project.org/.

A BibTeX entry for LaTeX users is

  @Manual{,
    title = {R: A Language and Environment for Statistical Computing},
    author = {{R Core Team}},
    organization = {R Foundation for Statistical Computing},
    address = {Vienna, Austria},
    year = {2020},
    url = {https://www.R-project.org/},
  }
```

```
We have invested a lot of time and effort in creating
R, please cite it when using it for data analysis. See
also 'citation("pkgname")' for citing R packages.

> citation("dplR")

Bunn AG (2008). "A dendrochronology program library in
R (dplR)." _Dendrochronologia_, *26*(2), 115-124. ISSN
1125-7865, doi: 10.1016/j.dendro.2008.01.002 (URL:
https://doi.org/10.1016/j.dendro.2008.01.002).

Bunn AG (2010). "Statistical and visual crossdating in
R using the dplR library." _Dendrochronologia_,
*28*(4), 251-258. ISSN 1125-7865, doi:
10.1016/j.dendro.2009.12.001 (URL:
https://doi.org/10.1016/j.dendro.2009.12.001).

  Andy Bunn, Mikko Korpela, Franco Biondi, Filipe
  Campelo, Pierre Mérian, Fares Qeadan and Christian
  Zang (2021). dplR: Dendrochronology Program Library
  in R. R package version 1.7.2.
  https://github.com/AndyBunn/dplR

To see these entries in BibTeX format, use
'print(<citation>, bibtex=TRUE)', 'toBibtex(.)', or
set 'options(citation.bibtex.max=999)'.
```

2 Working with Ring-Width Data

2.1 Reading Data

There are, alas, many different ways that tree-ring data are digitally stored. These range in sophistication from the simple (and commonly used) Tucson/decadal format file of ring widths to the more complex (but richer) TRiDaS format. We generally refer to these as `rwl` objects for "ring width length" but there is no reason these cannot be other types of tree-ring data (e.g., density).

The workhorse function for getting tree-ring data into R is dplR's `read.rwl` function. This function reads files in `"tucson"`, `"compact"`, `"tridas"`, and `"heidelberg"` formats. The onboard `rwl` data sets in dplR (i.e., `co021`, `ca533`, `gp.rwl`) were all imported into R using this function.

Throughout this vignette we will use the onboard data set `ca533` which gives the raw ring widths for bristlecone pine *Pinus longaeva* at Campito Mountain in California, USA. There are 34 series spanning 1358 years.

These objects are structured very simply as a `data.frame` with the series in columns and the years as rows. The series IDs are the column names and the years are the row names (both stored as characters). For instance, using the Campito Mountain ring widths:

```
> library(dplR)
> data(ca533) # the result of ca533 <- read.rwl('ca533.rwl')
> dim(ca533) # 1358 years and 34 series

[1] 1358   34

> colnames(ca533) # the series IDs

 [1] "CAM011" "CAM021" "CAM031" "CAM032" "CAM041" "CAM042"
 [7] "CAM051" "CAM061" "CAM062" "CAM071" "CAM072" "CAM081"
[13] "CAM082" "CAM091" "CAM092" "CAM101" "CAM102" "CAM111"
[19] "CAM112" "CAM121" "CAM122" "CAM131" "CAM132" "CAM141"
[25] "CAM151" "CAM152" "CAM161" "CAM162" "CAM171" "CAM172"
[31] "CAM181" "CAM191" "CAM201" "CAM211"

> head(rownames(ca533)) # the first few years

[1] "626" "627" "628" "629" "630" "631"

> class(ca533) # note "rwl" class as well as "data.frame"

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

2.2 Describing and Plotting Ring-Width Data

Once a `rwl` data set has been read into R, there are a variety of ways to describe and visualize those data. For instance, we can run a quick report using `rwl.report` that summarizes the `rwl` data and gives a few descriptive stats and also tells the user where the series have absent rings (zeros) and so on (more descriptive statistics like rbar and EPS are described below). We can also plot a `rwl` object by showing either the segments arranged over time as straight lines or as a "spaghetti plot." The `rwl` objects have a generic S3 method for `plot` and `summary`. See Figure 1.

```
> rwl.report(ca533)

Number of dated series: 34
Number of measurements: 23276
Avg series length: 684.5882
Range:  1358
Span:  626 - 1983
```

```
Mean (Std dev) series intercorrelation: 0.6294255 (0.0859289)
Mean (Std dev) AR1: 0.7092647 (0.09811876)
-------------
Years with absent rings listed by series
    Series CAM011 -- 1753 1782
    Series CAM031 -- 1497 1500 1523 1533 1540 1542 1545 1578 1579 1580 1655 1668 167
    Series CAM032 -- 1497 1523 1579 1654 1670 1681 1782
    Series CAM051 -- 1475
    Series CAM061 -- 1497 1523 1542 1545 1547 1579 1654 1655 1668 1670 1672 1782 185
    Series CAM062 -- 1542 1545 1547 1548 1579 1654 1655 1670 1672 1782 1836 1857 185
    Series CAM071 -- 1269 1497 1498 1523 1542 1547 1578 1579 1612 1655 1656 1668 167
    Series CAM072 -- 1218 1497 1498 1523 1533 1538 1542 1545 1546 1547 1571 1579 158
    Series CAM081 -- 1218 1336
    Series CAM082 -- 1362 1858 1865
    Series CAM091 -- 1655 1669 1670 1782 1858
    Series CAM092 -- 1624 1654 1655 1670 1672 1675 1677 1690 1703 1705 1707 1708 171
    Series CAM101 -- 1782 1783 1899 1929
    Series CAM102 -- 1669 1690 1782 1858 1899 1929
    Series CAM111 -- 1542
    Series CAM112 -- 1542
    Series CAM121 -- 1093 1218 1254 1361 1365 1460 1462 1468 1473 1475 1492 1497 154
    Series CAM122 -- 1117 1133 1147 1177 1218 1254 1361 1475 1497 1670
    Series CAM131 -- 1361
    Series CAM151 -- 1670 1703
    Series CAM161 -- 1523
    Series CAM162 -- 1618 1624 1641
    Series CAM181 -- 1450 1523
    Series CAM191 -- 1475 1497 1523 1533 1542 1558 1571 1578 1618 1655 1668 1670 167
    Series CAM201 -- 1523
    Series CAM211 -- 645 762 809 847 924 957 1014 1118 1123 1133 1147 1189 1350 1384
234 absent rings (1.005%)
-------------
Years with internal NA values listed by series
    None

> plot(ca533, plot.type="spag")
```

3 Detrending

Analysts typically (but not always) detrend a `rwl` data set to create an object containing ring-width index (rwi) values. The dplR package contains most standard detrending methods including detrending via splines, fitting

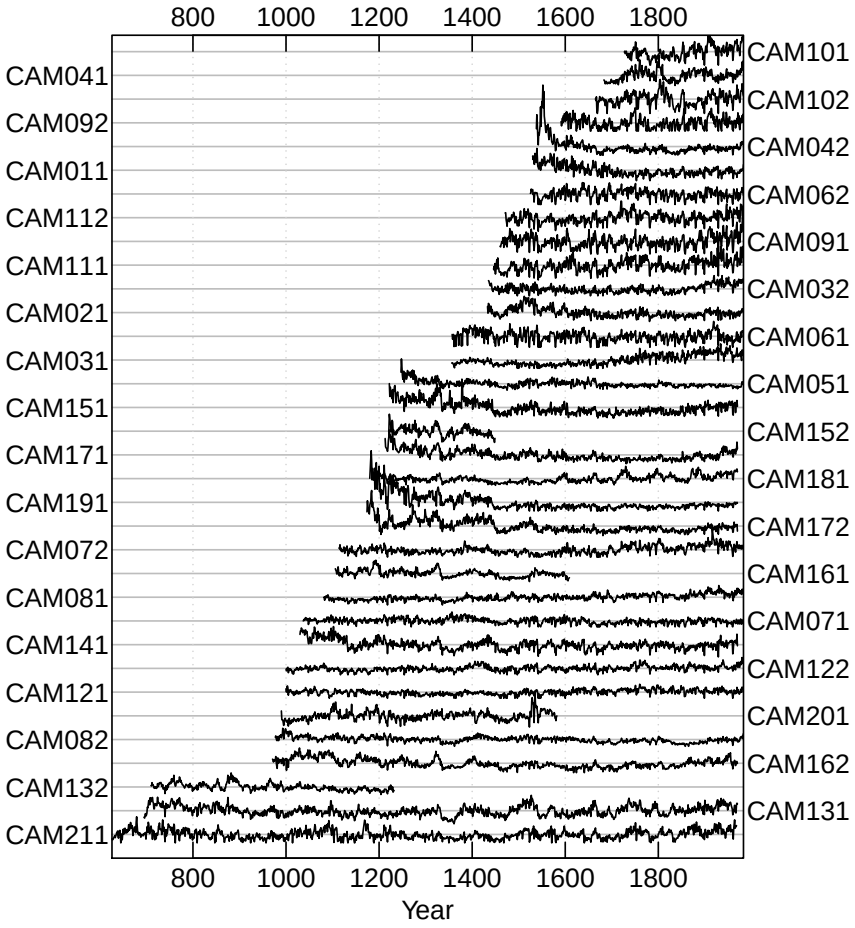

Figure 1: A spaghetti plot of the Campito Mountain ring widths.

negative exponential curves, and so on. There are also dplR functions for less commonly used detrending methods like regional curve standardization.

By the way, if this is all new to you – you should stop reading this vignette and proceed immediately to a good primer on dendrochronology like Fritts (2001). This vignette is not intended to teach you about how to do tree-ring analysis. It is intended to teach you how to use the package.

A rwi object has the same basic properties as the `rwl` object from which it is made. I.e., it has the same number of rows and columns, the same names, and so on. The difference is that each series has been standardized by dividing the ring widths against a growth model (e.g., a stiff spline, a negative exponential, etc.). This gives each series a mean of one (thus referred to as "indexed") and allows a chronology to be built (next section). As `read.rwl` is the primary function for getting data into R, `detrend` is the

primary function for standardizing `rwl` objects (but see `cms`, `rcs`, `bai.in`, and `bai.out` as well).

3.1 Common Detrending Methods

As any dendrochronologist will tell you, detrending is a dark art. In dplR we have implemented some of the standard tools for detrending but all have drawbacks. In all of the methods, the detrending is the estimation and removal of the low frequency variability that is due to biological or stand effects. The standardization is done by dividing each series by the growth trend to produce units in the dimensionless ring-width index (RWI). Much of the text that follows is modified from the help page of `detrend`.

Probably the most common method for detrending is what is often called the "conservative" approach of attempting to fit a negative exponential curve to a series. In the dplR implementation the `"ModNegExp"` method of `detrend` attempts to fit a classic nonlinear model of biological growth of the form $f(t) = a\exp(bt) + k$, where the argument of the function is time, using `nls`. See Fritts (2001) for details about the parameters. If a suitable nonlinear model cannot be fit (function is non-decreasing or some values are not positive) then a linear model is fit using `lm`. That linear model can have a positive slope unless `pos.slope` is `FALSE` in which case the series is standardized by its mean (method `"Mean"` in `detrend`).

For instance every series in the `ca533` object can be detrended at once via:

```
> ca533.rwi <- detrend(rwl = ca533, method = "ModNegExp")
```

This saves the results in `ca533.rwi` which is a `data.frame` with the same dimensions as the `rwl` object `ca533` and each series standardized.

```
> dim(ca533)

[1] 1358   34

> dim(ca533.rwi)

[1] 1358   34

> names(ca533)

 [1] "CAM011" "CAM021" "CAM031" "CAM032" "CAM041" "CAM042"
 [7] "CAM051" "CAM061" "CAM062" "CAM071" "CAM072" "CAM081"
[13] "CAM082" "CAM091" "CAM092" "CAM101" "CAM102" "CAM111"
[19] "CAM112" "CAM121" "CAM122" "CAM131" "CAM132" "CAM141"
[25] "CAM151" "CAM152" "CAM161" "CAM162" "CAM171" "CAM172"
[31] "CAM181" "CAM191" "CAM201" "CAM211"
```

```
> names(ca533.rwi)

 [1] "CAM011" "CAM021" "CAM031" "CAM032" "CAM041" "CAM042"
 [7] "CAM051" "CAM061" "CAM062" "CAM071" "CAM072" "CAM081"
[13] "CAM082" "CAM091" "CAM092" "CAM101" "CAM102" "CAM111"
[19] "CAM112" "CAM121" "CAM122" "CAM131" "CAM132" "CAM141"
[25] "CAM151" "CAM152" "CAM161" "CAM162" "CAM171" "CAM172"
[31] "CAM181" "CAM191" "CAM201" "CAM211"

> colMeans(ca533.rwi, na.rm=TRUE)

   CAM011    CAM021    CAM031    CAM032    CAM041    CAM042
0.9995596 1.0000454 1.0000000 0.9999635 1.0000000 1.0011776
   CAM051    CAM061    CAM062    CAM071    CAM072    CAM081
1.0002397 0.9999085 0.9999751 0.9999606 0.9999973 1.0000000
   CAM082    CAM091    CAM092    CAM101    CAM102    CAM111
0.9999579 0.9999768 0.9995803 1.0000000 1.0000000 1.0000000
   CAM112    CAM121    CAM122    CAM131    CAM132    CAM141
1.0000000 0.9999843 1.0000000 0.9997957 0.9984926 0.9999348
   CAM151    CAM152    CAM161    CAM162    CAM171    CAM172
0.9995000 0.9999264 1.0004105 0.9994063 0.9997201 0.9997682
   CAM181    CAM191    CAM201    CAM211
0.9999855 0.9953371 1.0000000 0.9998410
```

When `detrend` is run on an `rwl` object the function loops through each series. It does this by calling a different function (`detrend.series`) for each column in the `rwl` object. But, a user can also call `detrend.series` and it is useful to do so here for educational purposes.

Let us detrend a single series and apply more than one detrending method when we call it. We will call `detrend.series` using the verbose mode so that we can see the parameters applied for each method. The `detrend.series` function produces a plot by default (Figure 2).

```
> series <- ca533[, "CAM011"] # extract the series
> names(series) <- rownames(ca533) # give it years as rownames
> series.rwi <- detrend.series(y = series, y.name = "CAM011",
+                              verbose=TRUE)

Verbose output: CAM011
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 Options
 make.plot      TRUE
 method(s)1     c("Spline", "ModNegExp", "Mean", "Ar", "Friedman", "ModHugershoff"
 method(s)2     )
 nyrs           NULL
```

```
f              0.5
pos.slope      FALSE
constrain.nls  never
verbose        TRUE
return.info    FALSE
wt             default
span           cv
bass           0
difference     FALSE

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Zero years in input series:
1753 1782

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Detrend by ModNegExp.
Trying to fit nls model...
nls coefs
a:  0.66109686
b: -0.01184415
k:  0.31793386

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Detrend by ModHugershoff.
Trying to fit nls model...
nls coefs
a: 0.45550003
b: 0.15419657
g: 0.01532032
d: 0.32391578

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Detrend by spline.
Spline parameters
nyrs = 304, f = 0.5

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Detrend by mean.
Mean =  0.4395859

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Detrend by prewhitening.
Call:
ar(x = y[idx.goody])
```

```
Coefficients:
      1        2        3        4        5        6
 0.3884   0.1393   0.0002   0.0838   0.1321   0.0613
      7        8        9       10       11       12
 0.0381  -0.1255   0.0366  -0.0996  -0.0102   0.0147
     13       14       15       16       17       18
 0.0879   0.0104   0.0639  -0.0132   0.0151  -0.0044
     19       20       21       22       23
-0.0539   0.1240  -0.0302  -0.0545   0.1370

Order selected 23  sigma^2 estimated as  0.02092

 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Detrend by FriedMan's super smoother.
 Smoother parameters
 span = cv, bass = 0
 wt = default

 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Zero years in Ar series:
 1618
```

Often, a user will want to interactively detrend each series and fit a negative exponential curve to one series and a spline to another. This can be done via the `i.detrend` and `i.detrend.series` functions. See their help pages for details.

3.2 Other Detrending Methods

There are other detrending methods that are less commonly used but have distinct theoretical advantages. These include regional curve standardization (function `rcs`), C-Method Standardization (function `cms`), and converting measurements of ring widths to basal area increment (functions `bai.in` and `bai.out`). See help pages for further information.

4 Descriptive Statistics

The `rwl.report` function above gave a very cursory look at the `rwl` data. Let's look at some common (and not-so common) descriptive statistics of a `rwl` object. The `rwl.stats` function is typically used on raw ring widths (the `rwl` object) and produces summary statistics. Here are summary statistics on the first five series in `ca533`.

```
> rwl.stats(ca533[1:5]) # can also be run via summary(ca533)
```

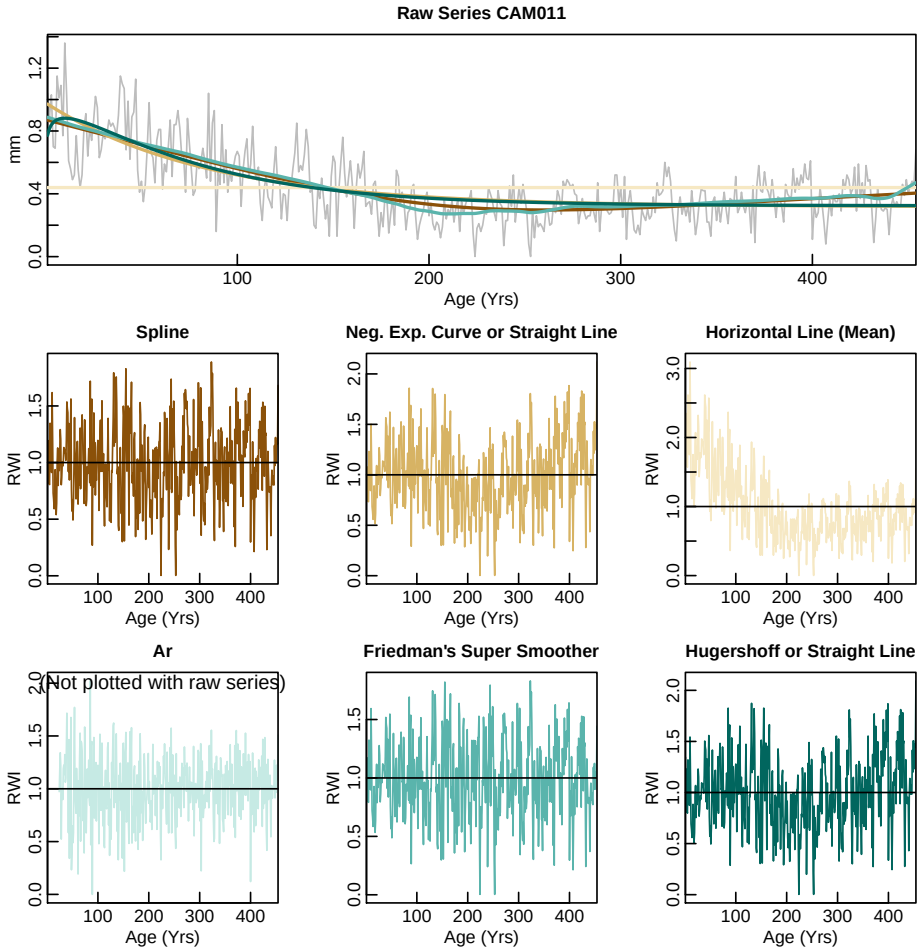

Figure 2: Detrending a single series via multiple methods.

	series	first	last	year	mean	median	stdev	skew	gini	ar1
1	CAM011	1530	1983	454	0.440	0.40	0.222	1.029	0.273	0.696
2	CAM021	1433	1983	551	0.424	0.40	0.185	0.946	0.237	0.701
3	CAM031	1356	1983	628	0.349	0.29	0.214	0.690	0.341	0.808
4	CAM032	1435	1983	549	0.293	0.26	0.163	0.717	0.309	0.661
5	CAM041	1683	1983	301	0.526	0.53	0.223	0.488	0.238	0.690

These are common summary statistics like mean, median, etc. but also statistics that are more specific to dendrochronology like the first-order autocorrelation (`ar1`) and mean sensitivity (`sens1` and `sens2`). We would be remiss if we did not here mention that mean sensitivity is actually a terrible statistic that should rarely, if ever, be used (Bunn *et al.*, 2013).

It is also easy in dplR to compute commonly used descriptive statistics that describe the correlation between series (both within and between tree

correlations) as well as the expressed population signal and signal-to-noise ratio for a data set. These are done in dplR using the `rwi.stats` function so-named because these statistics are typically (but not always) carried out on detrended and standardized ring widths (rwi). If a data set has more than one core taken per tree this information can be used in the calculations to calculate within vs. between tree correlation. The function `read.ids` is used to identify which trees have multiple cores.

```
> ca533.ids <- read.ids(ca533, stc = c(3, 2, 3))
> rwi.stats(ca533.rwi, ca533.ids, prewhiten=TRUE)

  n.cores n.trees    n n.tot n.wt n.bt rbar.tot rbar.wt
1      34      21 11.7   523   13  510    0.444   0.603
  rbar.bt c.eff rbar.eff   eps    snr
1   0.439 1.448    0.501 0.922 11.749
```

There is (at least) one other way of looking at the average interseries correlation of a data set. The `interseries.cor` function in dplR gives a measure of average interseries correlation that is different from the rbar measurements from `rwi.stats`. In this function, correlations are calculated serially between each tree-ring series and a master chronology built from all the other series in the `rwl` object (leave-one-out principle). The average of those correlations is sometimes called the "overall interseries correlation." This number is typically higher than `rbar.tot`. We are showing just the first five series and the mean for all series here:

```
> ca533.rho <- interseries.cor(ca533.rwi, prewhiten=TRUE,
+                              method="spearman")
> ca533.rho[1:5, ]

        res.cor p.val
CAM011 0.5358390     0
CAM021 0.6759898     0
CAM031 0.5257833     0
CAM032 0.6265193     0
CAM041 0.4906602     0

> mean(ca533.rho[, 1])

[1] 0.6368288
```

Again, if these concepts are unknown to you statistically look at some of the canonical works in dendrochronology like Cook *et al.* (1990) and Fritts (2001) as well as more recent works like Hughes *et al.* (2011).

5 Building a Mean Value Chronology

After detrending, a user will typically build a chronology by averaging across the years of the rwi object. In dplR the function for doing this is `chron` which by default uses Tukey's biweight robust mean (an average that is unaffected by outliers).

```
> ca533.crn <- chron(ca533.rwi, prefix = "CAM")
```

This object has the same number of rows as the rwi object that was used as the input and two columns. The first gives the chronology and the second the sample depth (the number of series available in that year).

```
> dim(ca533.rwi)

[1] 1358   34

> dim(ca533.crn)

[1] 1358    2
```

An object produced by `chron` has a generic S3 method for plotting which calls the `crn.plot` function (which has many arguments for customization). Here we will just make a simple plot of the chronology with a smoothing spline added. See Figure 3.

```
> plot(ca533.crn, add.spline=TRUE, nyrs=20)
```

6 Prospectus

In general this vignette aims to give a very cursory overview of basic tasks that most dendrochronologists will want to be aware of. Know that we are just scratching the surface of what dplR is capable of. As a small example, here is a way that a user might decide to truncate a chronology based on the expressed population signal. See Figure 4.

```
> def.par <- par(no.readonly=TRUE)
> eps.cut <- 0.85 # An arbitrary EPS cutoff for demonstration
> ## Plot the chronology showing a potential cutoff year
> ## based on EPS. Running stats on the rwi with a window.
> foo <- rwi.stats.running(ca533.rwi, ca533.ids,
+                          window.length = 80)
> yrs <- time(ca533.crn)
> bar <- data.frame(yrs = c(min(yrs), foo$mid.year, max(yrs)),
+                   eps = c(NA, foo$eps, NA))
> par(mar = c(2, 2, 2, 2), mgp = c(1.1, 0.1, 0), tcl = 0.25,
```

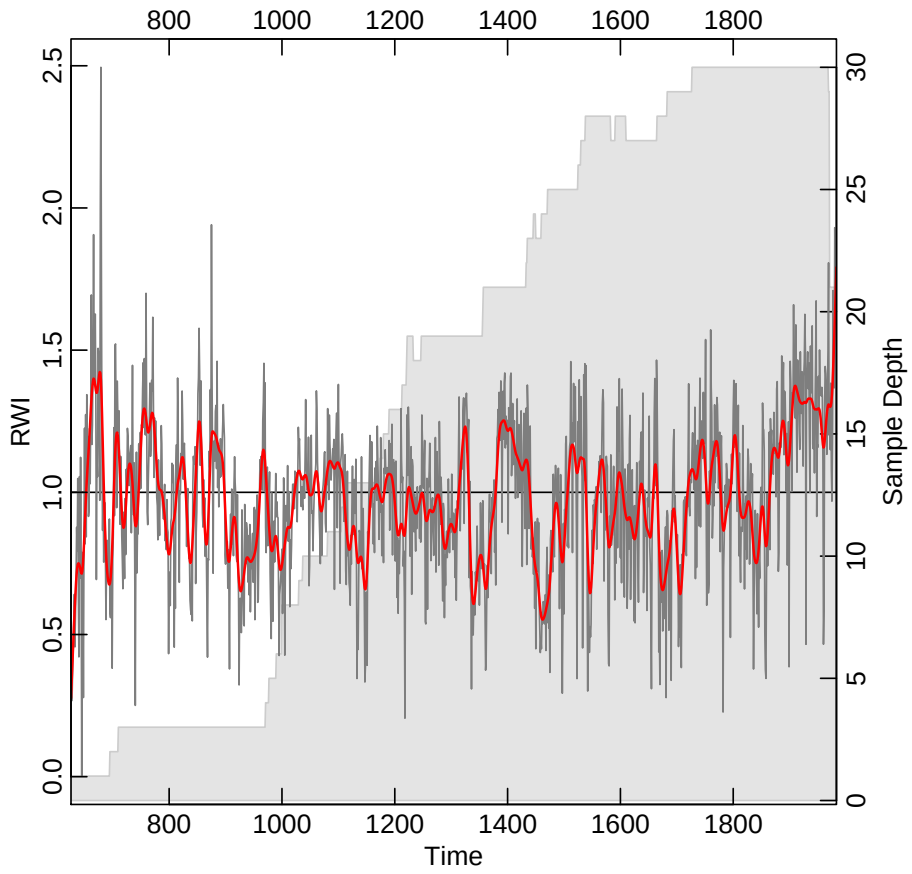

Figure 3: Campito Mountain chronology with 20-year smoothing spline.

```
+     mfcol = c(2, 1), xaxs='i')
> plot(yrs, ca533.crn[, 1], type = "n", xlab = "Year",
+      ylab = "RWI", axes=FALSE)
> cutoff <- max(bar$yrs[bar$eps < eps.cut], na.rm = TRUE)
> xx <- c(500, 500, cutoff, cutoff)
> yy <- c(-1, 3, 3, -1)
> polygon(xx, yy, col = "grey80")
> abline(h = 1, lwd = 1.5)
> lines(yrs, ca533.crn[, 1], col = "grey50")
> lines(yrs, ffcsaps(ca533.crn[, 1], nyrs = 32), col = "red",
+       lwd = 2)
> axis(1); axis(2); axis(3);
> par(new = TRUE)
> ## Add EPS
```

```
> plot(bar$yrs, bar$eps, type = "b", xlab = "", ylab = "",
+      axes = FALSE, pch = 20, col = "blue")
> axis(4, at = pretty(foo$eps))
> mtext("EPS", side = 4, line = 1.1)
> box()
> ## Second plot is the chronology after the cutoff only
> ## Chronology is rebuilt using just years after cutoff but
> ## that difference is essentially nil.
> yr.mask <- yrs > cutoff
> yrs2 <- yrs[yr.mask]
> ca533.crn2 <- chron(ca533.rwi[yr.mask, ])
> plot(yrs2, ca533.crn2[, 1], type = "n",
+      xlab = "Year", ylab = "RWI", axes=FALSE)
> abline(h = 1, lwd = 1.5)
> lines(yrs2, ca533.crn2[, 1], col = "grey50")
> lines(yrs2, ffcsaps(ca533.crn2[, 1], nyrs = 32),
+       col = "red", lwd = 2)
> axis(1); axis(2); axis(3); axis(4)
> box()
> par(def.par)
```

We hope that this vignette helps users cover introductory data handling and processing using dplR and R. As we noted above we are just providing a short introduction as to what is possible in dplR. There are many other functions in dplR that will help users analyze tree rings. These include a host of functions for statistical cross dating as well as spectral and wavelet analysis. We will cover those in future vignettes.

References

Bunn AG (2008). "A dendrochronology program library in R (dplR)." *Dendrochronologia*, **26**(2), 115–124. ISSN 11257865. doi:10.1016/j.dendro.2008.01.002. URL http://linkinghub.elsevier.com/retrieve/pii/S1125786508000350.

Bunn AG (2010). "Statistical and visual crossdating in R using the dplR library." *Dendrochronologia*, **28**(4), 251–258. ISSN 11257865. doi:10.1016/j.dendro.2009.12.001. URL http://linkinghub.elsevier.com/retrieve/pii/S1125786510000172.

Bunn AG, Jansma E, Korpela M, Westfall RD, Baldwin J (2013). "Using simulations and data to evaluate mean sensitivity (ζ) as a useful statistic in dendrochronology." *Dendrochronologia*, **31**(3), 250–254. ISSN 11257865. doi:10.1016/j.dendro.2013.01.004. URL http://www.sciencedirect.com/science/article/pii/S1125786513000295.

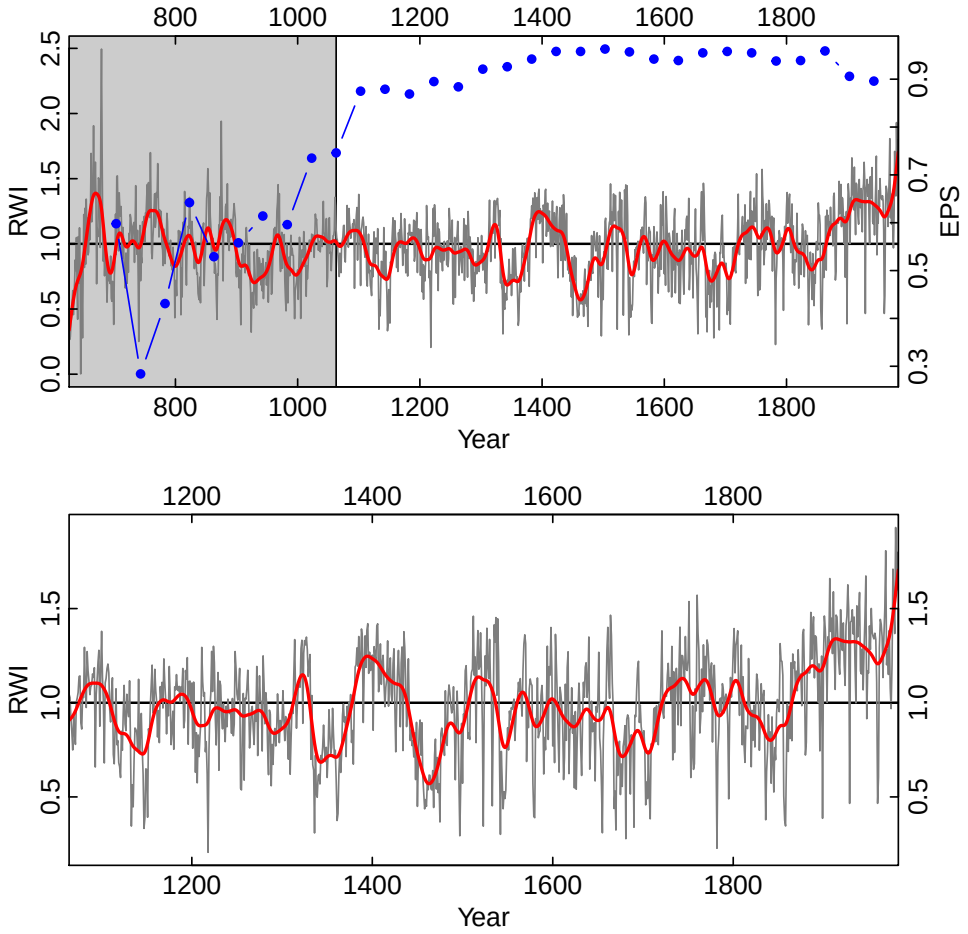

Figure 4: Campito Mountain chronology using an EPS cutoff.

Cook E, Briffa K, Shiyatov S, Mazepa V (1990). "Tree-Ring Standardization and Growth-Trend Estimation." In E Cook, L Kairiukstis (eds.), *Methods of Dendrochronology: Applications in the Environmental Sciences*, pp. 104–123. Kluwer, Dordrecht. ISBN 978-0792305866.

Fritts HC (2001). *Tree Rings and Climate*. The Blackburn Press. ISBN 1930665393. URL http://www.amazon.com/Tree-Rings-Climate-H-Fritts/dp/1930665393.

Hughes MK, Swetnam TW, Diaz HF (eds.) (2011). *Dendroclimatology*, volume 11 of *Developments in Paleoenvironmental Research*. Springer Netherlands, Dordrecht. ISBN 978-1-4020-4010-8. doi:10.1007/978-1-4020-5725-0. URL http://www.springerlink.com/index/10.1007/978-1-4020-5725-0.