Chronology Building in dplR

Andy Bunn       Mikko Korpela

Processed with dplR 1.7.0 in R version 3.5.3 (2019-03-11) on July 15, 2019

Abstract

In this vignette we cover methods for building mean-value chronologies using standardized ring widths.

Contents

1 Introduction 2
   1.1 What Is Covered ........................................ 2
   1.2 Citing dplR and R .......................................... 2

2 Data Sets 3

3 Building Chronologies 5
   3.1 Traditional Chronology ................................. 5
   3.2 Using a Cutoff .......................................... 6
   3.3 Chronology uncertainty ................................. 10
   3.4 Stripping out series by EPS ............................ 11

4 Conclusion 12
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 an illustration of chronology development in dplR.

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:


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 = {2019},
  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")


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

2 Data Sets

Throughout this vignette we will use the onboard data set wa082 which gives the raw ring widths for Pacific silver fir _Abies amabilis_ at Hurricane Ridge in Washington, USA. There are 23 series covering 286 years. The data are plotted in Figure [1]. Note the summary stats below.

```r
> library(dplR)
> data(wa082)
> wa082.sum <- summary(wa082)
> mean(wa082.sum$year)

[1] 197.2609

> mean(wa082.sum$stdev)

[1] 0.410913

> mean(wa082.sum$median)

[1] 0.9756522

> mean(wa082.sum$ar1)
```

3
By the way, if this is all new to you – you should 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.
3 Building Chronologies

Let us make a few chronologies from the wa082 data after detrending each series with a spline that has a frequency response of 50% at a wavelength of 2/3 of each series’s length. Detrending is an enormously complicated area unto itself. We use a spline here but could standardize by several other methods and likely get results that look quite different. There is a reason, after all, that dendrochronologists have been arguing about detrending for decades.

```r
> wa082.rwi <- detrend(wa082, method="Spline")
```

3.1 Traditional Chronology

The simplest way to make a chronology in dplR is with the `crn` function which also has a `plot` method. This defaults to building a mean-value chronology by averaging the rows of the `rwi` data using Tukey’s biweight robust mean (function `tbrm` in dplR). The result is plotted in Figure 2 with a 30-year smoothing spline for visualization.

```r
> wa082.crn <- chron(wa082.rwi, prefix="HUR")
> tail(wa082.crn)

    HURstd samp.depth
1978  0.8766400       22
1979  1.1043443       22
1980  1.0818157       22
1981  1.2447671       22
1982  0.8961536       22
1983  0.9714484       22

> plot(wa082.crn, add.spline=TRUE, nyrs=30)
```

Note that the `chron` function will also compute a residual chronology by “prewhitening” the series before averaging. If the `prewhiten` flag is set to `TRUE`, each series is whitened using `ar` prior to averaging. The residual chronology is thus white noise. Note that the `wa082.std.resid` object has two columns with chronologies as well as the sample depth in a third column. The result is plotted in Figure 3.

```r
> wa082.std.resid <- chron(wa082.rwi, prefix="HUR", prewhiten = TRUE)
> tail(wa082.std.resid)

    HURstd HURres samp.depth
1978  0.8766400  0.8069522       22
1979  1.1043443  1.1348334       22
1980  1.0818157  1.0703302       22
1981  1.2447671  1.2372227       22
1982  0.8961536  0.8695930       22
1983  0.9714484  0.9725288       22

> plot(wa082.std.resid, add.spline=TRUE, nyrs=30)
```
Figure 2: The wa082 chronology.

1980 1.0818157 1.0276662 22
1981 1.2447671 1.2000848 22
1982 0.8961536 0.7757018 22
1983 0.9714484 0.9871302 22

> plot(wa082.std.resid, add.spline=TRUE, nyr=30)

3.2 Using a Cutoff

A relatively simple addition to the traditional chronology is to truncate the chronology when the sample depth gets to a certain threshold. The output from the chron function contains a column called samp.depth which shows the number of series that are average for a particular year. We can use the subset function to modify the chronology. The result is plotted in Figure 4.
Figure 3: The wa082 chronology as the standard chronology and the residual chronology.

```r
> head(wa082.crn)
   HURstd samp.depth
1698 0.9864665    1
1699 0.7960214    1
1700 0.8070109    1
1701 0.5590470    1
1702 0.8436158    2
1703 0.6771966    2

> wa082.trunc <- subset(wa082.crn, samp.depth > 5)
> # and plot
> plot(wa082.trunc, add.spline=T, nyrs=30)
```
Figure 4: The wa082 chronology truncated by sample depth.

It would likely be more robust to recalculate the rwi object by truncating the rw1 file and then making a chronology which could be done by nesting commands via:

```r
> wa082.trunc2 <- chron(detrend(wa082[wa082.crn$samp.depth > 5,],
+   method="Spline"), prefix="HUR")
```

The result in this case is likely to be virtually identical to truncating after calculating the chronology as in Figure 4.

A more interesting and likely more robust approach is to truncate via the subsample signal strength (SSS). The result is plotted in Figure 5.

```r
> wa082.ids <- autoread.ids(wa082)
> sss.cut <- 0.85
> wa082.sss <- sss(wa082.rwi, wa082.ids)
```
> yrs <- time(wa082.crn)
> op <- par(no.readonly=TRUE)
> par(mar = c(2, 2, 2, 2), mgp = c(1.1, 0.1, 0), tcl = 0.25,
+ mcol = c(2, 1), xaxs='i')
> plot(yrs, wa082.crn[, 1], type = "n", xlab = "Year",
+ ylab = "RWI", axes=FALSE)
> cutoff <- max(yrs[wa082.sss < 0.85])
> 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, wa082.crn[, 1], col = "grey50")
> lines(yrs, ffcsaps(wa082.crn[, 1], nyrs = 30), col = "red", lwd = 2)
> axis(1); axis(2); axis(3);
> par(new = TRUE)
> ## Add EPS
> plot(yrs, wa082.sss, type = "l", xlab = "", ylab = "",
+ axes = FALSE, col = "blue")
> abline(h=0.85,col="blue",lty="dashed")
> axis(4, at = pretty(wa082.sss))
> mtext("SSS", side = 4, line = 1.1)
> box()
> ## Second plot is the chronology after the cutoff only
> ## Chronology is rebuilt using just years after cutoff but
> ## the difference by doing it this way rather than just truncating
> ## is essentially nil.
> yr.mask <- yrs > cutoff
> yrs2 <- yrs[yr.mask]
> wa082.rwi2 <- detrend(wa082[yr.mask, ], method="Spline")
> wa082.crn2 <- chron(wa082.rwi2)
> plot(yrs2, wa082.crn2[, 1], type = "n",
+ xlab = "Year", ylab = "RWI", axes=FALSE)
> abline(h = 1, lwd = 1.5)
> lines(yrs2, wa082.crn2[, 1], col = "grey50")
> lines(yrs2, ffcsaps(wa082.crn2[, 1], nyrs = 30),
+ col = "red", lwd = 2)
> axis(1); axis(2); axis(3); axis(4)
> box()
> par(op)

This method calculates a running value for EPS using the \texttt{rwi.stats.running} function with a 30-year window. Then we select a cutoff point where we truncate the chronology when the EPS value dips below a certain point. The cutoff method works in this case because the EPS dips as sample depth de-
clines. However, some feel this is a more objective way of setting a cutoff as opposed to just picking a minimum number of samples.

### 3.3 Chronology uncertainty

Typically we calculate a chronology by taking the average of each year from the \( \text{rwi} \) object. (And that is typically the biweight robust mean.) The function `chron` like pretty much all the functions in `dplR` are relatively simple chunks of code that are used for convenience. We can make our own chronology and get the mean plus two standard errors of the yearly growth (Figure 6). Don’t get stuck in just using the prepackaged functions in `dplR`!

```r
wa082.avg <- apply(wa082.rwi2, 1, mean, na.rm = TRUE)
se <- function(x){
```

Figure 5: The wa082 chronology truncated by EPS.
3.4 Stripping out series by EPS

We want to introduce one other approach that doesn’t deal explicitly with chronology building but can be used to build a better chronology. The `strip.rwl` function uses EPS-based chronology stripping (Fowler and Boswijk 2003) where each series is assessed to see if its inclusion in the chronology improves the EPS. If it does not the series is dropped from the `rwl` object. As we will see in this example two series are excluded which causes a modest improvement in EPS (Figure 7).

```r
> wa082.strip.rwl <- strip.rwl(wa082, ids = wa082.ids)

REMOVE -- Iteration 1: leaving series 712071 out.
EPS improved from 0.88 to 0.882.

REMOVE -- Iteration 2: leaving series 712072 out.
EPS improved from 0.88 to 0.883.

REMOVE -- Iteration 3: no improvement of EPS. Aborting...
REINSERT -- Iteration 1: no improvement of EPS. Aborting...
```
Figure 6: The wa082 chronology with two standard errors.

> wa082.rwi.strip <- detrend(wa082.strip.rwl, method="Spline")
> wa082.crn.strip <- chron(wa082.rwi.strip, prefix = "HUR")
> wa082.crn.strip <- subset(wa082.crn.strip, samp.depth > 5)
> plot(wa082.crn.strip, add.spline=TRUE, nyrs=30)

4 Conclusion

We have tried to introduce a few ways of building chronologies with \texttt{dplR} that are either typical (like truncating by sample depth) or less commonly used. In this vignette we aren’t advocating any particular method but trying to get the users familiar with ways of interacting with the objects that \texttt{dplR} produces. Once the user understands the data structures the rest of \texttt{R} opens up.
Again, we feel that it is important to reiterate that the advantage of using dplR is that it gets the analyst to use R and thus have access to the essentially limitless tool that it provides. Go forth!

References
