Available online at www.sciencedirect.com
ScienceDirect
Dendrochronologia 28 (2010) 251–258
DENDROCHRONOLOGIA
www.elsevier.de/dendro

TECHNICAL NOTE

Statistical and visual crossdating in R using the dplR library

Andrew G. Bunn*

Environmental Sciences, Huxley College, Western Washington University, Bellingham, WA 98225-9181, United States

Received 26 September 2009; accepted 17 December 2009

Abstract

I demonstrate new functionality for the Dendrochronology Program Library in R (dplR) that allows for flexible statistical crossdating of tree-ring data. Using a well-dated ring-width file, I give examples of how dplR can be used to examine correlations between each series and a master chronology according to overlapping time periods (segments) specified by the user; examine moving correlations of suspect series; and compute cross-correlation functions to identify specific dating issues. I also show how automatically generated skeleton plots can be used to visually crossdate. Much of the terminology and approach used for crossdating in dplR will be familiar to users of COFECHA.

Keywords: Statistical software; Moving correlation; Cross-correlation; Skeleton plot; COFECHA

Introduction

The advent of computer-based tools has greatly advanced the field of dendrochronology. The Dendrochronology Program Library (Holmes, 1992) and ARSTAN (Cook and Holmes, 1996) brought new rigor to the field and are widely used by tree-ring scientists worldwide. The program COFECHA (Holmes, 1983) brought statistical crossdating into common practice and is employed by the International Tree-Ring Data Bank as the standard quality-control tool.

The Dendrochronology Program Library in R (dplR) (Bunn, 2008) is an open-source package used within the R statistical programming environment (R Development Core Team, 2009). The use of dplR makes it easier for dendrochronologists to use R as their primary analytic environment especially in conjunction with other tree-ring specific packages (e.g., bootRes; Zang, 2009). Here I describe functions that allow graphical crossdating using dplR version 1.3 (Bunn, 2010).

New crossdating functions and examples

I use Schulman's Mesa Verde Douglas Fir (*Pseudotsuga menziesii*) data from the International tree-ring data bank (Schulman, 1963), to demonstrate how dplR can be used to crossdate tree-ring series. The Mesa Verde data are exquisitely well dated with 35 series spanning the 13th century to the mid-20th century. The average series length is 565 years ($\sigma = 157$), autocorrelation at 1-year averages 0.60 ($\sigma = 0.16$), and mean sensitivity, calculated according to Eq. (2) in Biondi and Qeadan (2008), is 0.61($\delta = 0.11$). Each series is correctly dated and the average series correlation to the master chronology is 0.85 ($\sigma = 0.04$). These numbers, and indeed all the calculations and figures presented here, are reproducible using the code in Appendix A.

A common issue in crossdating is the presence of a missing ring. To demonstrate crossdating in R, I randomly chose a series from the Mesa Verde ring-width dataset and corrupted

*Tel.: +1 360 650 4252; fax: +1 360 650 7284.
E-mail address: andrew.bunn@wwu.edu.

1125-7865/$ – see front matter © 2010 Istituto Italiano di Dendrocronologia. Published by Elsevier GmbH. All rights reserved.
doi:10.1016/j.dendro.2009.12.001

it by deleting an existing measurement. The code below reads in the Mesa Verde data (included with dplR) and corrupts series 641143 by deleting the 325th measurement:

```
R>data(co021)

R>dat=co021

R>tmp=dat$'641143'

R>tmp=c(NA,tmp[-325])

R>dat$'641143'=tmp
```

The R object `dat` now contains the one misdated series. I will demonstrate how to locate the approximate location of that dating problem. First, the dplR function `corr.rwl.seg()` looks at the correlation of each series in a tree-ring dataset according to user-specified segment lengths and creates a plot where suspect series are flagged if the correlation to the master chronology is below a user-defined critical value. The correlation measure is Spearman's rank correlation coefficient (ρ) by default. In this function each series is removed from the dataset and a master chronology is calculated as the mean of the remaining series (using Tukey's biweight robust mean by default). Each series is prewhitened by default to remove autocorrelation using an autoregressive model where the parameters and complexity are selected by AIC. Users can opt to detrend each series using a Hanning filter to remove low-frequency variability prior to, or

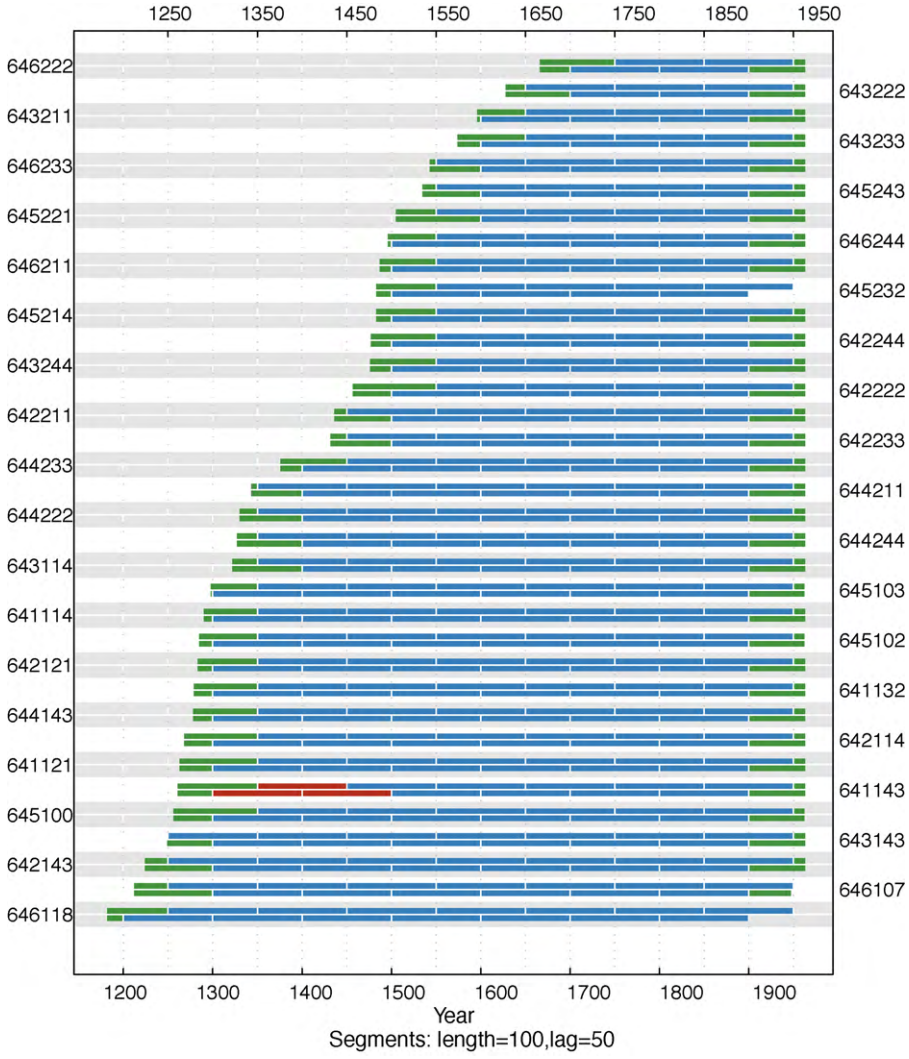

Fig. 1. Each segment of each series in Schulman's Mesa Verde dataset is shown and colored by correlation with the master chronology. Each series is represented by two courses of lines with the bottom course adhering to the bottom axis timeline and the top course matching the upper axis timeline (100-year segments lagged by 50 years). Segments are colored according to their correlation. Blue segments correlate well to the master chronology (p-values less or equal to the user-set critical value) while potential dating problems are indicated by the red segments (p-values greater than the user-set critical value). Green lines show segments that do not completely overlap the time period and for which no correlations are calculated. Series 641143 shows poor correlation beginning in the 16th century. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

in lieu of, prewhitening as well. The default arguments to `corr.rwl.seg()` can be seen using the `args()` function in R:

```
R>args(corr.rwl.seg)
function (rwl, seg.length = 50,
bin.floor = 100, n = NULL,
prewhiten = TRUE, pcrit = 0.05,
biweight = TRUE, make.plot =
TRUE, ...)
```

This indicates that the function `corr.rwl.seg()` requires a ring-width object (rwl) but all the other arguments have default values (e.g., the segment length, whether to make a plot, whether to use Tukey's biweight robust mean, *etc*.) that can be changed by the user and are detailed in the function's help page. Fig. 1 is produced using the object `dat` as defined above:

```
R>rwl.100=corr.rwl.seg(rwl=dat,seg.
length=100)
```

The argument `seg.length` indicates that a segment length of 100 years is used and each segment is lagged by half the segment length (50 years) to create overlap. Series 641143 falls below the default critical probability $p = 0.05$ going back in time from the 1400–1500 segment. The 1450–1550 segment is not flagged. The results are stored in the `rwl.100` object which gives a correlation matrix with ρ and p-value for each series and each segment, the overall correlation and p-value are given for each series in its entirety as well as the average correlation for each segment. If any series are flagged (e.g., 641143) the series id and a list of questionable segments are returned as well.

The function `corr.series.seg()` can be used to look more closely at the correlation values for individual series. As above, the series and master are optionally filtered and the series is compared against the master chronology. The default arguments to `corr.series.seg()` can be seen using the `args()` function as described above. Fig. 2 shows the graphical output of the following:

```
R>flagged=dat$'641143'

R>names(flagged)=rownames(dat)

R>dat$'641143'=NULL

R>seg.100=corr.series.seg(rwl=dat,

series=flagged,

    seg.length=100)
```

Here, the questionable series 641143 is assigned to the object `flagged` and given its dates as attributes (names). That series is then removed from the master chronology `(dat)`. The series `flagged` is compared against a chronology of all the other series from the Mesa Verde dataset. The correlations are returned by segment (specified with a length of 100 years, lagged by 50 years) and a centered running correlation with a window equal to the segment length is also returned and plotted. The object `seg.100` stores the results. As Fig. 2 shows, the correlation between this series and the master begins to deteriorate prior to 1500 AD.

A key issue to explore at this point is to determine whether the flagged series and segments are the result of a dating errors or just a period of low correlation from, say, suppression in that individual tree. We can use cross-correlation functions at this point to determine if the problem with the series stems from misdating (i.e., is it better dated at another point in time?). The function `ccf.series.rwl()` calculates a cross-correlation function between a series and a master chronology according to segment (as described above) with a user-specified window that gives the maximum lag at which to calculate the correlations. As above, the default arguments to `ccf.series.rwl()` can be seen using the `args()` function. Fig. 3 shows the results of the cross-correlation:

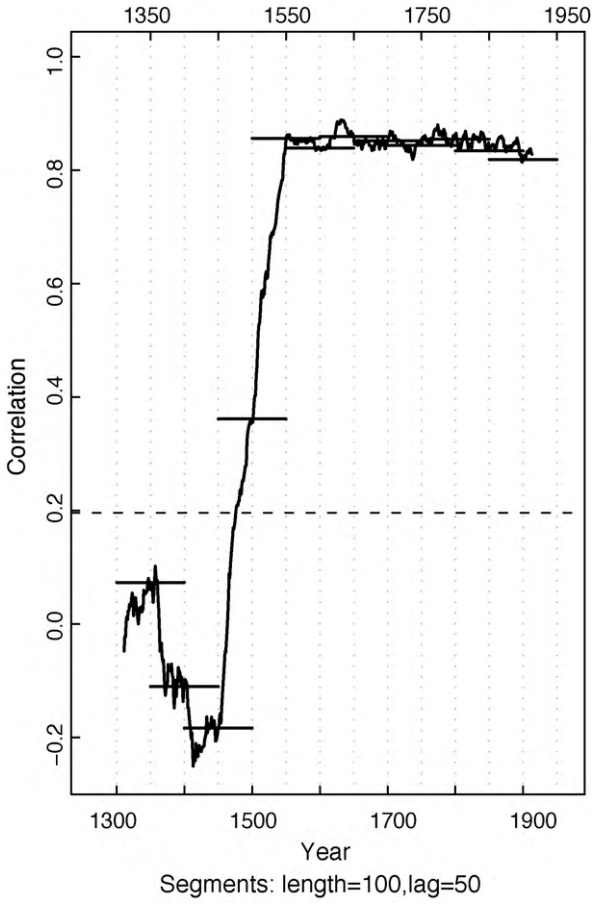

Fig. 2. Correlations between series 641143 and the master chronology are shown with horizontal lines according to the specified segments (100-year segments lagged by 50 years). A centered running correlation with a length of 100 years complements the segment correlations. The user-specified critical level is show with a dashed line. Series 641143 begins to lose correlation with the master at 1450–1550 AD and is no longer significantly correlated prior to 1500 AD.

```
R>ccf.100=ccf.series.rwl(rwl=dat,series=flagged,
     seg.length=100)
```

This indicates that the core is well dated (lag = 0) until the 1450–1550 segment. A lag of −1 is persistent prior to then indicating a dating problem. Because we now know that the core is well dated until 1500 AD we can begin to target that area in the cross-correlation function with a shorter segment length of 30 years as shown in Fig. 4:

```
R>win=1390:1600
R>dat.yrs=as.numeric(rownames(dat))
R>dat.trunc=dat[dat.yrs%in%win,]
R>flagged.yrs=as.numeric(names(flagged))
R>flagged.trunc=flagged[flagged.yrs%in%win]
R>names(flagged.trunc)=rownames(dat.trunc)
R>ccf.30=ccf.series.rwl(rwl=dat.trunc,
series=flagged.trunc,
     seg.length=30)
```

It now appears that the dating on series 641143 is good at 1505–1535 AD, at a lag of −1 between 1475 and 1505, and deteriorating in the 1490–1520 AD segment.

One more easily performed analysis is possible using the function `skel.plot()` in dplR. This creates a skeleton plot by calculating departures from high frequency growth for each year by comparing that year to the surrounding 3 years (i.e., $t-1$, t, $t+1$). These departures are assigned a relative scale of 1–10 as described in the function's help page. A Hanning filter is used to remove low-frequency variation. To compare series 641143 to the master chronology:

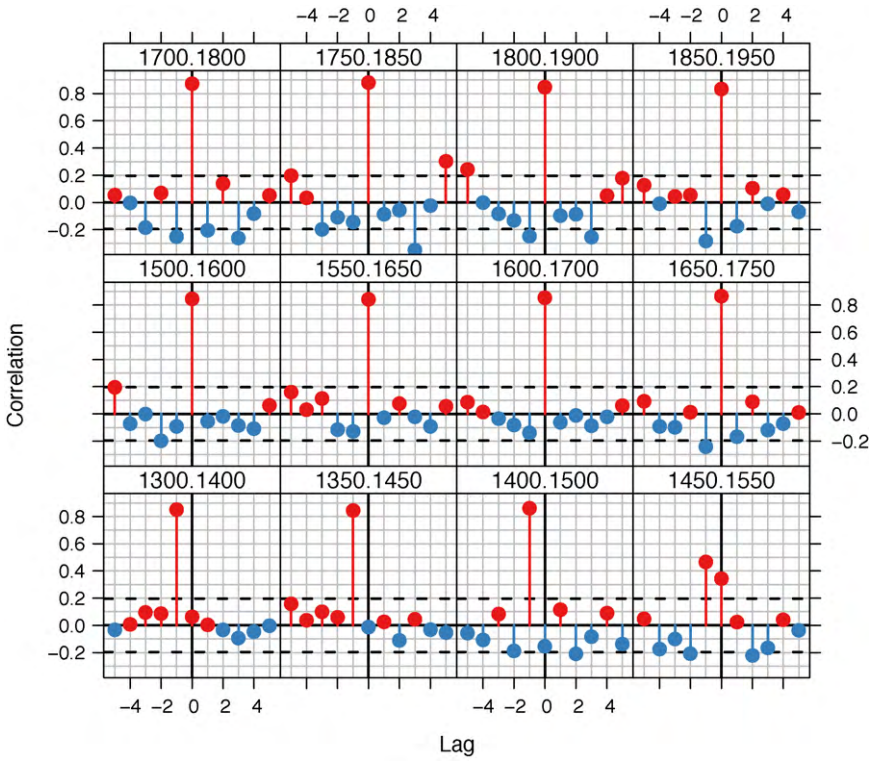

Fig. 3. Cross-correlations between series 641143 and the master chronology are shown for each segment (100-year segments lagged by 50 years). The series correlates well at lag 0 from 1500 to 1600 AD onward but at lag −1 prior to 1500 AD. The time period from 1450 to 1550 AD appears suspect with correlation split between lag 0 and lag −1. The user-specified critical value for correlation is shown by the horizontal dashed lines.

```
R>win=1440:1559
R>dat.trunc=dat[dat.yrs%in%win,]
R>dat.yrs.trunc=dat.yrs[dat.yrs%in%win]
R>flagged.trunc=flagged[flagged.yrs%in%win]
R>flagged.yrs.trunc=flagged.yrs
  [flagged.yrs%in%win]
R>flagged.name='641143'
R>skel.plot(rw.vec=flagged.trunc,yr.
   vec=flagged.yrs.trunc,
     sname=flagged.name,master=FALSE)
R>skel.plot(rw.vec=rowMeans(dat.trunc,na.
            rm=T),
     yr.vec=dat.yrs.trunc,sname='CO021',
     master=TRUE)
```

Fig. 5 compares the master skeleton plot to the series skeleton plot for the 120 years after 1440 AD. This confirms the result of the cross-correlation analysis (Fig. 4) and indicates a date shift near or at the year 1500 AD. In fact, the marker years in 1495 AD and 1497 AD are the first line up with the plot from 641143 at lag −1 which is then well matched after the 1505 marker year seen in both series 641143 and in the master chronology.

At this point the analyst has a strong inclination to examine series 641143 closely at 1500 AD. Indeed, deleting the 325th year from series 641143 created a "missing ring" in the year 1500 AD and thus a dating lag of −1 prior to that point.

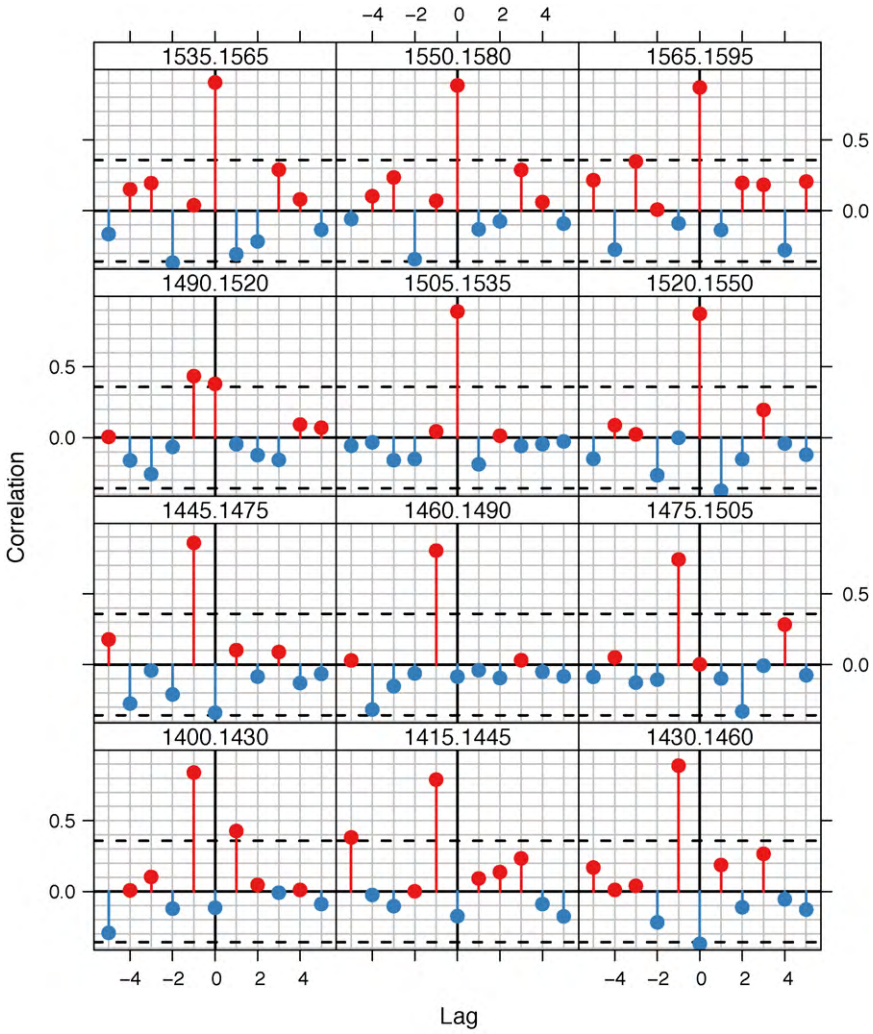

Fig. 4. Cross-correlations between series 641143 and the master chronology are shown for each segment as in Fig. 3 but for a narrowed time window 1400–1600 using 20-year segments lagged by 10 years). As above, the series correlates well at lag 0 from 1500 AD onwards but at lag −1 prior to 1500 AD. The time period from 1490 to 1520 AD appears suspect with correlation split between lag 0 and lag +1.

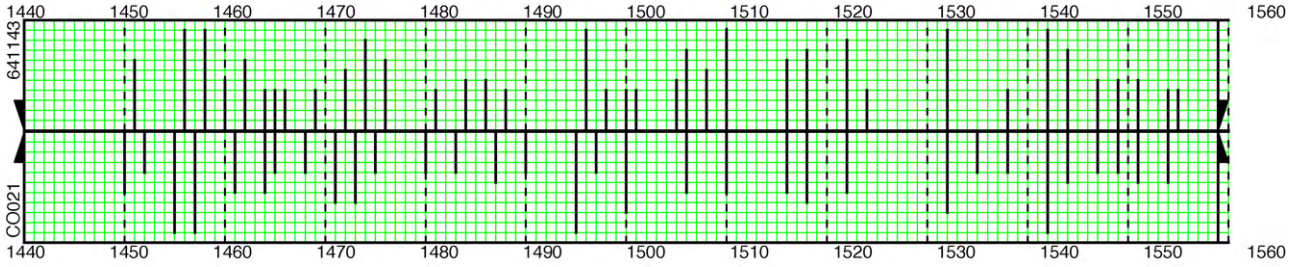

Fig. 5. An automatically generated skeleton plot compares series 641143 against the master chronology from 1440 to 1560 AD. As above, there is strong correspondence between marker years after 1500 and marker years in 641143 precede the master chronology from 1440 to 1498 by 1 year.

Conclusion

Crossdating with the aid of correlation analyses has been widespread in dendrochronology since the adoption of COFECHA (Holmes, 1983) and is a welcome addition to a discipline that becomes more technologically and statistically savvy every year. The addition of statistical crossdating greatly expands the uses of dplR and will help R become a primary environment for those wishing to perform analysis in one software package. However, as powerful as R is as a computing environment, successful crossdating is ultimately dependent on the experience of the analyst and statistical sophistication should never supplant the judgment of an experienced dendrochronologist. Wood should always be inspected carefully and crossdated based on human, and not computer, expertise.

Availability

There are hundreds of contributed packages to R of which dplR is one. New users of R can download and install the base software from the Comprehensive R Archive Network website: http://cran.r-project.org/. Once R is installed and running the dplR package can be installed and loaded:

```
R>install.packages('dplR')

R>library(dplR)
```

Users with older versions of dplR can update their installations via:

```
R>update.packages()
```

The help pages for all functions including the new crossdating functions described above can seen viewed with their embedded examples:

```
R>?corr.rwl.seg

R>?corr.series.seg

R>?ccf.series.rwl

R>?skel.plot
```

Acknowledgements

I would like to acknowledge support from the National Science Foundation (ARC-0612346, ATM-0629172, and OPP-0732477). L. Berner and C. Robertson provided helpful tests of the new functions. Users of dplR have provided bug fixes and suggestions. The quality of this manuscript improved due to suggestions by one anonymous reviewer.

Appendix A.

```
# start with a clean workspace

rm(list=ls())

# load the library

library(dplR)

# load the data

data(co021)

# figures from the paper
```

Appendix A (*Continued*)

```
dat=co021

# useful stats on the data - sensitivity, ar1, etc.

sum.stats=rwl.stats(dat)

# not run

# average series correlation to the master chronology

# tmp=corr.rwl.seg(co021,make.plot=F)

# mean(tmp$overall[,1]); sd(tmp$overall[,1])


# create a missing ring by deleting a year of

# growth in a random series

tmp=dat$'641143'

tmp=c(NA,tmp[-325])

dat$'641143'=tmp


# figure 1

# see the arguments possible for the corr.rwl.seg function

args(corr.rwl.seg)

# fig 1

rwl.100=corr.rwl.seg(dat,seg.length=100)


# take the misdated series and remove it

# from the rwl object

flagged=dat$'641143'

names(flagged)=rownames(dat)

dat$'641143'=NULL
```

Appendix A (*Continued*)

```
# figure 2

seg.100=corr.series.seg(rwl=dat,series=flagged,seg.length=100)


# figure 3

ccf.100=ccf.series.rwl(rwl=dat,series=flagged,seg.length=100)

# figure 4

win=1390:1600

dat.yrs=as.numeric(rownames(dat))

dat.trunc=dat[dat.yrs%in%win,]

flagged.yrs=as.numeric(names(flagged))

flagged.trunc=flagged[flagged.yrs%in%win]

names(flagged.trunc)=rownames(dat.trunc)

ccf.30=ccf.series.rwl(rwl=dat.trunc,series=flagged.trunc,

seg.length=30)

# figure 5 (two figures are merged in illustrator for paper)

win=1440:1559

dat.trunc=dat[dat.yrs%in%win,]

dat.yrs.trunc=dat.yrs[dat.yrs%in%win]

flagged.trunc=flagged[flagged.yrs%in%win]

flagged.yrs.trunc=flagged.yrs[flagged.yrs%in%win]

flagged.name='641143'

skel.plot(rw.vec=flagged.trunc,yr.vec=flagged.yrs.trunc,

sname=flagged.name,master=FALSE)

skel.plot(rw.vec=rowMeans(dat.trunc,na.rm=T), yr.vec=dat.yrs.trunc,

sname='CO021',master=TRUE)
```

References

Biondi, F., Qeadan, F., 2008. Inequality in paleorecords. Ecology 89, 1056–1067.

Bunn, A.G., 2008. A dendrochronology program library in R (dplR). Dendrochronologia 26, 115–124.

Bunn, A.G., 2010. dplR: Dendrochronology Program Library in R. R package version 1.3. URL http://www.R-project.org.

Cook, E.R., Holmes, R.L., 1996. Users Manual for Program ARSTAN. Laboratory of Tree-Ring Research, University of Arizona, Tucson, USA.

Holmes, R.L., 1992. Dendrochronology Program Library, Instruction and Program Manual (January 1992 update). Laboratory of Tree-Ring Research, University of Arizona, Tucson, USA.

Holmes, R.L., 1983. Computer assisted quality control in tree-ring dating and measurement. Tree-Ring Bulletin 43, 69–78.

R Development Core Team, 2009. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Schulman, E., 1963. Schulman Old Tree No. 1 Data Set. IGBP PAGES/World Data Center for Paleoclimatology Data Contribution Series 1983-CO021.RWL. NOAA/NCDC Paleoclimatology Program, Boulder, Colorado, USA.

Zang, C., 2009. bootRes: The bootRes Package for Bootstrapped Response and Correlation Functions. R package version 0.1. URL http://www.R-project.org.