Dendrochronologia 49 (2018) 9–15

Contents lists available at ScienceDirect

Dendrochronologia

journal homepage: www.elsevier.com/locate/dendro

burnr: Fire history analysis and graphics in R

Steven B. Malevich[a,b,*], Christopher H. Guiterman[b], Ellis Q. Margolis[b,c]

[a] Department of Geosciences, University of Arizona, 1040 E. 4th Street, Tucson, AZ 85721, USA
[b] Laboratory of Tree-Ring Research, University of Arizona, 1215 E. Lowell Street, Tucson, AZ 85721, USA
[c] US Geological Survey, Fort Collins Science Center, New Mexico Landscapes Field Station, Santa Fe, NM 87508, USA

ARTICLE INFO

Keywords:
Fire ecology
Statistical software
Event
Visualization
Tree ring
SEA

ABSTRACT

We developed a new software package, burnr, for fire history analysis and plotting in the R statistical programming environment. It was developed for tree-ring fire-scar analysis, but is broadly applicable to other event analyses (e.g., avalanches, frost rings, or culturally modified trees). Our new package can read, write, and manipulate standard tree-ring fire history FHX files, produce fire—demography charts, calculate fire frequency and seasonality statistics, and run superposed epoch analysis (SEA). A key benefit of burnr is that it enables automation of analyses and plotting, especially for large data sets. The package also facilitates creative plotting, mapping, and analyses when combined with the thousands of packages available in R. In this paper, we describe the basic functionality of burnr and introduce users to fire history analyses in R.

1. Introduction

Faced with complicated environmental challenges such as climate change, dendrochronologists are tackling increasingly sophisticated research questions. The dendroecological fire history community has grown rapidly in recent decades, driven by the need to put increases in fire size and severity in a long-term context. Novel insights can often be gained with analyses of large tree-ring data networks (e.g., Cook, 2004) and the proliferation of publicly available data facilitates such efforts. Across the western United States, for example, researchers have built an extensive network of over 1,000 tree-ring fire history sites (Falk et al., 2011), hosted on the International Multi-Proxy Paleofire Database (IMPD). However, performing network-scale analyses is challenging without the ability to read existing data formats (e.g., FHX files) into statistical computer software, such as the popular R language environment (R Development Core Team, 2017). Performing fire-history research in R would facilitate coarse-scale, complex analyses by allowing for automated workflows, easy sharing of data and methods with collaborators or in publications, and the ability to link and analyze metadata associated with fire history data. To facilitate future growth, efficiency, and creativity in dendroecological fire history analyses, we developed burnr.

burnr is a new Open Source package for R, written for users and projects with advanced needs while being accessible for basic use. Specifically, burnr was designed to provide a variety of functions that are simple and flexible, and to bring fire history data to the rich analysis and plotting tools in the R environment. Dendrochronologists are increasingly using R, in part due to the number of powerful packages that have been developed (e.g., Bunn, 2008; Altman et al., 2014; Zang and Biondi, 2015). With flexible base functionality and exposure to R, burnr can act as a foundation from which users can develop their own advanced tools for fire history, or any event-based research – such as frost rings, culturally modified trees, insect outbreaks, flood scarring, or avalanche reconstructions.

The burnr package grew out of methods established by fire history studies that capitalized on the ability of tree rings to record events with annual to sub-annual precision over long time periods (e.g., Dieterich and Swetnam, 1984). Existing tree-ring fire history analysis programs, including FHX2 (Grissino-Mayer, 2001) and FHAES (Brewer et al., 2016), are widely used. We based the burnr functions on these software systems, but by facilitating analyses in R, we hope that users will develop new research and analytical approaches.

In this paper we describe how to install burnr, and we demonstrate its basic use for manipulating and summarizing FHX files, fire—-demography plotting, fire interval analysis, and superposed epoch analysis (SEA). In many of these sections we showcase some of the exciting ways that users can benefit by analyzing fire-history data in R. Each section of this paper explains code and analysis, assuming that the user has installed R and loaded the burnr package (see Section 2). Text in `fixed-width font` indicates code that is typed or output to an interactive R shell. Lines prefixed with " > " denote code entered by the user. Finally, note that source code to recreate this paper's figures is available online (https://github.com/brews/burnr_2018_manuscript_figures). Output presented in this paper is from burnr v0.2.2.

* Corresponding author at: Department of Geosciences, University of Arizona, 1040 E. 4th Street, Tucson, AZ 85721, USA.
E-mail address: malevich@email.arizona.edu (S.B. Malevich).

https://doi.org/10.1016/j.dendro.2018.02.005
Received 22 September 2017; Received in revised form 15 February 2018; Accepted 21 February 2018
Available online 02 March 2018

2. Availability and installation of burnr

The burnr package is freely available as Open Source software through the Comprehensive R Archive Network (CRAN). R can be downloaded and installed directly from the CRAN homepage (https://cran.r-project.org/) or through an alternative R distribution. Users can install burnr from within an R session like any other R package:

```
> install.packages("burnr")
```

The library is then loaded with:

```
> library(burnr)
```

With the package loaded, burnr functionality becomes available to the user. Development releases and source code for the project are available online (https://github.com/ltrr-arizona-edu/burnr). In addition to this paper, support documentation is distributed with the package, including an introduction (https://cran.r-project.org/web/packages/burnr/vignettes/introduction.html).

3. Reading, manipulating, and summarizing FHX files in R

With burnr, fire-history data can be read into R from FHX files. From there, R has the capability to edit, sort, merge, and write data back to FHX files. For example:

```
> library(burnr)
> url <- paste0("https://www1.ncdc.noaa.gov/",
  "pub/data/paleo/firehistory/",
  "firescar/northamerica/")
> pmr <- read_fhx(paste0(url, "uspmr001.fhx"))
> pme <- read_fhx(paste0(url, "uspme001.fhx"))
> pmw <- read_fhx(paste0(url, "uspmw001.fhx"))
> pm_all <- pmr+pme+pmw
> write_fhx(pm_all, "pajarito_mountain.fhx")
```

Here we are reading in three FHX files directly from URLs linking to the online IMPD file server. These data sets are assigned to variables named pmr, pme, and pmw. We then concatenate—or combine—the data with the + operator, producing a single fhx object. Concatenation also checks for duplicate series names before combining the data. The pm_all object is then written into a new FHX file in the current working directory.

Within R, fire history data is stored as a special fhx object, which is similar to a simple R dataframe with three columns: year, series, and rec_type. The first column, year is the datum to the year in which it is recorded; series is the tree or series identifier; rec_type is the tree-derived information stored in FHX files. We found this structure to be relatively efficient for data storage and analysis in R. The fhx object structure is especially effective when combined with tools utilizing a "split-apply-combine" strategy (e.g., Wickham, 2011).

Once loaded in R, several functions are available to analyze FHX data. A quick summary of the individual series in the combined FHX files, pm_all, can be produced with:

```
> series_stats(pm_all)
```

The corresponding output from series_stats() is shown in Table 1. This gives a tree-level summary of the input data.

Table 1
Example series statistics, from series_stats(), for the first five trees in pm_all. The table gives the first, last, and total number of years in the series ("No. yrs"). "Inner" and "Outer" denote how each series starts and ends. Number of scars, injuries ("Inj."), and recording years ("Rec.") are given. Additionally, we give the mean fire interval ("Interval") for each series.

Series	First	Last	No. yrs	Inner	Outer	Scars	Inj.	Rec.	Interval
PMR01	1664	1915	252	pith	outer	19	0	231	10.6
PMR02	1663	1875	213	pith	outer	15	0	190	13.2
PMR03	1676	1979	304	inner	bark	7	9	200	15.8
PMR05	1747	1895	149	inner	outer	2	0	95	40.0
PMR06	1653	1860	208	pith	outer	5	0	176	18.0
				⋮					

4. Plotting fire history and demography

To showcase plotting we included example fire-history data sets within the burnr package. We begin with lgr2, a fire history and demography data set from Los Griegos Peak, New Mexico that contains 11 fire-scarred or fire-injured trees and 15 non-fire-scarred trees of four species (Margolis and Malevich, 2016). Below, we load the example data set to the lgr2 variable as an fhx object, as though we had read the data from an FHX file:

```
> data(lgr2)
```

We can make a fire—demography plot of lgr2 with the basic plot function:

```
> plot(lgr2)
```

The default plot() graphic is shown in Fig. 1. The output from the default plot function is highly customizable, but visually simple. The function includes many options, such as a customizable composite, or "rug", of fire events summarizing all fire events recorded by all series, the ability to add or remove series labels, customizable year-axis range, inclusion of tree injuries, and addition of a legend.

Colored fire—demography plots are one of the more novel features of burnr. In the example below, we add color to differentiate tree species. Tree species are drawn from metadata that can be loaded into R from a comma or tab-delimited file with read.csv() or read.table(). Metadata recording the species of all trees in the Los Griegos data set is included in burnr, and can be loaded with:

```
> data(lgr2_meta)
```

This metadata can be used with the plot function because series names in the fire history lgr2 object match a column (TreeID) in the metadata. Type head(lgr2_meta) to see the first few metadata entries and get an idea of how the data set is organized. A custom fire chart that incorporates species metadata is now created with:

```
> plot(lgr2,
  color_group=lgr2_meta$SpeciesID,
  color_id=lgr2_meta$TreeID,
  ylabels=FALSE, composite_rug=TRUE,
  injury_event=TRUE)
```

The new arguments to plot() produce a fire—demography chart with colors corresponding to tree species and a fire-composite rug along the bottom of the graphic (Fig. 2a). The species information (color_group) is joined with lgr2 through the series names (color_id). We also remove the y-axis labels (ylabels=FALSE) to save plotting space, and add a composite rug that includes injuries with the arguments composite_rug=TRUE and injury_event=TRUE.

Coloring by species in fire—demography charts can be especially helpful to identify successional trends or species transitions related to fire or climate variability. For example, in Fig. 2a, *Pinus ponderosa* (red) was present in the stand but becomes locally absent following a high-severity fire, followed by over a century of fire exclusion that promoted dominance by more mesic species: *Populus tremuloides* and *Pseudotsuga menziesii* (Margolis and Malevich, 2016). Color can be associated with other variables such as tree condition (live or dead) or age classes. It can also represent continuous variables like tree diameter at breast height or elevation.

Advanced edits of fire—demography charts requires a deeper understanding of how burnr creates graphics. Internally, burnr takes advantage of R's advanced grid plotting framework and the ggplot2 package (Wickham, 2016). This helps burnr to create more sophisticated graphics (e.g., with coloring and faceting) that are also modular and customizable. To introduce this, we modify the lgr2 fire—demography chart by adding a legend—with custom placement—and a shaded polygon to highlight the species transition we described previously:

```
> library(ggplot2)
> lgr2_sorted <- sort(lgr2, decreasing=TRUE)
```