pcrecon: An R Package for Nested Principal Component Regression Climate Reconstructions

Laura Smith1\*, Nicholas Nagle1, Stockton Maxwell2, Daniel Hocking3

*1 University of Tennessee; Knoxville, Tennessee, 37996, USA* *2 Radford University; Radford, Virginia, 24142, USA* *3 NOAA Greater Atlantic Regional Fisheries Office; Gloucester, MA, 01930 USA*

*\* Corresponding Author Email:* [*lsmit224@vols.utk.edu*](mailto:lsmit224@vols.utk.edu)

# Abstract

We present pcrecon, a new software package for the development of nested principal component regression climate reconstructions from tree ring data in R. The use of the R statistical programming environment for this application can improve replicability and transparency, which are both crucial to the goals of the paleoclimate community.

# Introduction

## Principal Component Regression Climate Reconstructions

Principal Component Regression (PCR) has been used extensively for decades to develop long-running climate reconstructions from tree rings (Jacoby & Cook 1981, Peters et al. 1981, Cook et al. 1999, Williams et al. 2020).Principal Component Analysis (PCA) reduces the dimensionality of predictors, in this case tree ring chronologies, down to a few principal eigenvectors which represent the most shared variance among the many datasets. A linear regression model is used to estimate the relationship between the eigenvectors (predictor) and climate (predictand).

This technique allows for the inclusion of data from multiple sites and tree species while limiting multicollinearity among related predictors. This process can help to parse the local, autocorrelated ecological noise related to forest succession, competition, and disturbance from desired regional climate signal.An assumption inherent to these models is that the signal shared among many trees across the included region is related to climate (Fritz 1976). PCR has been of particular importance in reconstructing climate of the eastern U.S., where climate-growth relationships are more complex due to the temperate climate and highly disturbed forests (Alexander et al., 2019; Cook, Meko, Stahle, & Cleaveland, 1999; Maxwell et al., 2017, p. Harley et al. (2017); Maxwell, Hessl, Cook, & Pederson, 2011; Pederson et al., 2012).

A limitation of PCA is it requires equal length vectors, and therefore a nesting technique is applied such that a new model is developed for for each common period, and the shortest chronology is removed from the dataset for each subsequent nest. *explain this better*

## Software for Analysis of Paleoclimate Proxy Data

Due to the time consuming nature of data collection and the specialized statistical methods used for the analysis of tree-ring data, the paleoclimate community often relies on the open sharing of data and software. This is evidenced by the popularity and frequent use of resources such as the International Tree Ring Data Bank (ITRDB, NOAA), the Dendrochronology Program Library (Richard Holmes/University of Arizona), the Lamont-Doherty dendrochronology software repository (website), and R packages such as dplR (Bunn 2008) and treeclim (Zang 2015).

Incredible efforts have been made over the last 40 years in the development of statistical techniques for tree-ring data analysis and associated software which allow users to standardize series, develop chronologies, cross-date, correlate to climate field, estimate climate reconstructions, and much more. These programs revolutionized dendrochronolgy, and their free availability to the community is a testament to the importance of open communication and collaboration in science. Many of these programs utilize an interactive user interface and are based in FORTRAN, which is a powerful and versatile language well suited to scientific computing. They efficiently perform complex calculations, preventing the user from having to program them.

PcReg (<https://www.ldeo.columbia.edu/tree-ring-laboratory/resources/software>), a program developed by Ed Cook and *honestly I don’t remember who programmed it and that information isn’t documented anywhere on the Lamont site* has become the standard for many researchers seeking to make estimates of past climates from tree rings using PCR. Like other programs designed for tree-ring analysis (ARSTAN, COFECHA, EDRM), PcReg handles the antiquated fixed-width formats (i.e. Tuscon and chronology formats) that are commonly used to store tree-ring data. Different versions have varying capabilities, but the current publicly available version of PCreg requires users to run each individual nest and splice them together.

## R Environment

R is a powerful statistical programming language and software environment which has several advantages over common FORTRAN or MATLAB based programs. R is free and open source, in addition to being extremely versatile in its application for many different types of statistical analyses, data processing, simulations, etc. It has a large global user-base across a wide variety of disciplines, many of whom are actively engaged in various wikis, newsletters, and web forums (such as stack exchange) which create a community of support through active dialogue and problem solving.

In contrast to the commonly used FORTRAN programs for tree-ring analysis, R utilizes the command-line rather than an interactive user interface. On it’s surface, this may seem like a disadvantage to some users. However, most common tree-ring packages in R are well-documented with vignettes that provide worked examples. There are great workflow advantages in having repeatable scripts that can be adjusted and rerun all within Rstudio, without having to open a seperate text file, close and reopen the program file, and re-run that logfile. It’s also simple to reorganize and do any number of manipulations to the data (i.e. filtering, transformations, etc), as well as many types of analyses (tree-ring specific or not), and produce publication quality visualizations, all in one environment. Thanks to the work of Andy Bunn and others on package dplR (2012), fixed width format files can be read in to R as a dataframe or matrix, making all manner of analysis on these data more accessible.

The implications for reproducibility in using script-based code are perhaps the most important aspect. Peer-reviewed journals are increasingly requiring data and analysis to be provided as part of the manuscript submission. This is not feasible with the many output files that are rendered in programs like PCreg or ARSTAN, and the outputs are not universally very readable outside of those individuals who work with them frequently. A standard, versatile, common, and cross-disciplinary environment for analysis is important for clear communication and reproducibility of results.

In the case of PcReg, while the methods are well-documented in a number of publications, there is no documentation for the use of the program. The interactive interface is not intuitive, therefore its use has become something of an oral tradition, where an experienced researchers share how to use the program through workshops or individual interactions. While there is nothing inherently wrong with this approach, the strict documentation requirements for inclusion in the CRAN repository are an additional advantage of working within the R environment. *and maybe it doesn’t end up on CRAN initially?*

# Pcrecon Package

# Functions

There are three main functions in the pcreg package for making reconstruction estimates: load\_clim(), eval\_clim() and pcreg().

* load\_clim() takes monthly climate data and selects individual months, averages, or sums across selected months (previous year through current year, -12:12). Input is a dataframe or matrix object containing climate data, months of interest, and method for summarizing. Output is an S3 list which includes the climate variable and the ar model if prewhitening is indicated.
* eval\_clim selects chronologies for inclusion in the PC regression based on correlation with the climate variable. Inputs are the S3 list as created by the load\_clim() function and a dataframe containing tree-ring chronology data in which each column is a chronology. Output includes a dataframe of correlation coefficients for each chronology and the target climate variable, a dataframe of chronologies selected for inclusion in the model, and a dataframe containing the start and end year of each selected chronology.
* pcreg() calculates principal component eigenvectors and linear regression model estimates, then makes predictions for each nest and splices them together. Output includes: climate reconstruction, model statistics and validation statistics for each nest, PCA objects for each nest, and lm objects for each nest.

Additional functions are used to retrieve metadata from ITRDB downloaded tree ring files, filter tree rings for inclusion based on spatial location or correlation to target climate, and to upload and summarize climate data from various formats. These include:

* parse\_json
* filter\_spat
* filter\_cor
* load\_clim

*Should I go into descriptions of these? Or leave for the vignette rather than publication?*

# Worked Example

# Data

Tree ring data downloaded form the ITRDB using the FedData package (include code to procure? used a bounding box) Climate data is USGS Streamflow gage data for gauge #347432, downloaded from waterdata.usgs.gov

# Code

Streamflow data from gauge 08380500 (Gallinas Creek near Montezuma), Latitude 35°39’07.18“, Longitude 105°19’07.79”

Location of Stream gauge on Gallinas Creek near Montezuma

Location of Stream gauge on Gallinas Creek near Montezuma

Tree ring data from ITRDB

Bounding Box for retrieving ITRDB tree ring chronologies

Bounding Box for retrieving ITRDB tree ring chronologies

Typical workflow for this type of analysis often begins with downloading proxy data from a repository, or uploading the users own data. Tree ring data downloaded from the ITRDB is accompanied by a folder of metadata files in JavaScript Object Notation (JSON) format. The parse\_json() function can be used to creates a table which includes: lat/lon, chronology length, species, elevation, and NOAA ID for each chronology. The user indicates the directory location for the folder containing the .json files using the “dir” argument. When center = TRUE, the mean between the minimum/maximum elevation, northernmost/southernmost latitude, and easternmost/westernmost longitude is reported.

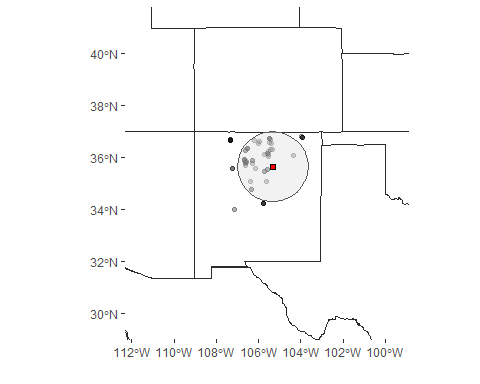
’> metadata <- parse\_json(dir = “extdata/metadata”, center = TRUE)

| ID | NOAADataTableId | earliestYear | mostRecentYear | timeUnit | speciesCode | scientificName | chron\_length | lat | lon | elev |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NM581 | 19237 | 1520 | 2003 | AD | PIED | Pinus edulis Engelm. | 483 | 36.5333 | -106.0167 | 2300 |
| NM582 | 19238 | 1575 | 2003 | AD | PIPO | Pinus ponderosa Douglas ex C. Lawson | 428 | 36.6167 | -105.9833 | 2500 |
| NM588 | 29778 | 620 | 2011 | AD | PIPO | Pinus ponderosa Douglas ex C. Lawson | 1391 | 36.2820 | -106.6450 | 2525 |

Having these metadata in a table allows the user to filter based on variables such as location, species, elevation, etc. The pcrecon package contains two functions for filtering chronologies or other datasets based on spatial location. The first, filter\_rad(), selects chronology IDs that fall within a given radius from a point. The arguments for this function are: dataframe containing site ID, latitude, and longitude, the lat/lon of the center point, and the radius in km. The output of this function is a character vector of chronology IDs that fall within that radius.

Here, we will filter within a 150km radius of the streamgauge, and save the output as an object called select\_crns\_rad.

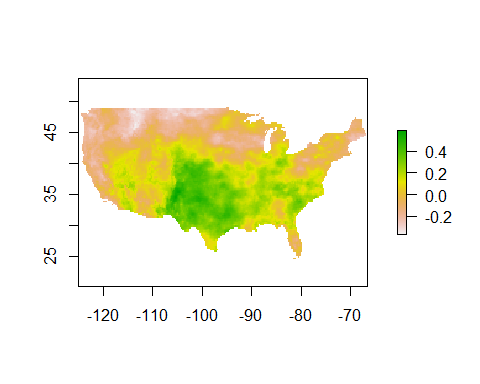
`> select\_crns\_rad <- filter\_rad(x = metadata, cent\_lat = 35.65, cent\_lon = -105.32, radius = 150, plot = TRUE)’



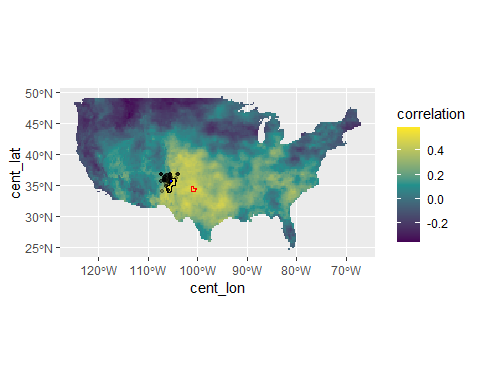
Another option for filtering chronologies spatially is to select those within a given climate footprint using the filter\_foot() function. This method is most often used when reconstructing streamflow. Chronologies are selected within an area where the pearson correlation (r) between streamflow and precipitation or soil moisture meets a given threshhold (cite harley, s. maxwell, j. maxwell). This function requires a raster file. In this example, a .nc file created in the KNMI climate explorer (<https://climexp.knmi.nl/start.cgi>) is used, which contains correlation coefficients between PRISM 4km gridcell precipitation data (cite) and Gallinas Creek streamflow.

We will select chronologies that are in an area where correlation between streamflow and precipitation is 0.5 or higher (r = 0.5). This is a higher threshold than is typically used for the purpose of keeping the dataset small.

*explain why radius is used here*



> footprint <- raster::raster(system.file("extdata/gallinas\_cf.nc", package = "pcrecon", mustWork = TRUE)) > select\_crns\_fp <- filter\_foot(x = metadata, footprint = footprint, r = 0.5, cent\_lat = 35.65, cent\_lon = -105.32, radius = 150, plot = TRUE)



filter\_foot and filter\_rad return a character vector containing IDs of the chronologies that should be included in the next step of the analysis, which is to evaluate and filter based on correlation between chronologies and the target climate variable. The load\_crns function reads those .crn files from a folder, such as those downloaded from the ITRDB or produced using ARSTAN. The user can select which measurement and chronology types associated with each ID to read in. The default is the standard ring width chronology. Other options include:

Code Measurement Type (type\_measure)

D Total Ring Density

E Earlywood Width

I Earlywood Density

L Latewood Width

N Minimum Density

R Ring Width

T Latewood Density

X Maximum Density

P Latewood Percent

Code Chronology Type (type\_crn)

A ARSTND

P Low Pass Filter

R Residual

S Standard

W Re-Whitened Residual

N Measurements Only

The output is a dataframe in which columns contain chronologies and rows are ring width observations

Next, I’ll read in the streamflow data I downloaded from <https://waterdata.usgs.gov/nwis>. A big advantage of using R for these analyses is that any intermediary procedures (plotting, transformations) can be done in one environment and saved in a script for posterity. Here, I will take a look at the distribution and autoregressive order of the streamflow data.

These data are highly skewed. I’ll do a log transform to try to make things a little more symmetric:

The load\_clim function aggregates the climate by calculating the mean or sum over particular months of interest. You can also select individual months. This function requires that these data be in 13 column (year and months 1:12) or long (3 column - “ID”, “month”, and “value”) format. See examples above for how to use name() and/or pivot\_longer to easily reformat to compatible data structures.

eval\_clim() calculates correlations between the climate variable and the chronologies. Here, the user indicates the calibration/validation period for the intended model, the window over which to correlate the chronologies and climate, leads/lags, correlation test type, alpha value, and a correlation cutoff for selecting chronologies. Weighting features will be added later. See function documentation for options.

The output of eval\_clim() is an S3 object of class “PCreg\_data”. This object contains lots of necessary pieces that feed into the final regression model function. You can view the the components of the object using the “$” operator:

The pcreg() function takes the information from the previous step in the form of the PCreg\_data object, and performs the principal component regression estimates for each nest. The user indicates the selection threshold for PCs, the window over which to calculate PCs, and the window over which to scale the variance.

The output of this function is an object of class PCreg\_recon, and you can call the pieces of it as was done above. Output includes model estimates, validation and calibration period statistics (RE, CE, R2), as well as linear model and Principal Component Analysis objects for each nest, and autoregressive model statistics for climate and chronologies if AR prewhitening is used.

Since we log transformed the target climate before running the model, let’s exponentiate the reddened values and take a look:

### Comparison to PCreg

### Citations

Harley, Grant L, Justin T Maxwell, Evan Larson, Henri D Grissino-Mayer, Joseph Henderson, and Jean Huffman. 2017. “Suwannee River Flow Variability 1550–2005 CE Reconstructed from a Multispecies Tree-Ring Network.” *Journal of Hydrology* 544: 438–51. <https://doi.org/10.1016/j.jhydrol.2016.11.020>.