Neotoma paper

Simon Goring

24 September, 2014

neotoma: A Programmatic Interface to the Neotoma Paleoecological Database

Abstract:

Paleoecological data is integral to ecological analysis, providing an opportunity to study interactions between communities and abiotic environments at centennial to millenial time scales. Paleoecological analysis also allows us to observe changes in ecological processes that occur infrequently, such as megadroughts, hurricanes, rapid climatic change and volcanic eruptions. Paleoecological inference also can allow us to understand ecological processes in the absence of widespread antrhopogenic influence.

The R package `neotoma` obtains and manipulates data from the Neotoma Paleoecological Database (Neotoma: http://www.neotomadb.org). Neotoma is a searchable repository for multiproxy paleoecological records spanning the past 5 million years, providing the opportunity to explore biogeographic patterns from the Pliocene to the present. `neotoma` searches the Neotoma Database for datasets using search keys that can include location, taxon or dataset type (*e.g.*, pollen, mammal, ostrocode) using the database's Application Programming Interface (API). The package returns a set of nested metadata associated with the site, including the full assemblage record, dataset specific information (age range of samples, date of accession into Neotoma, site principle investigator) and site specific information (location, site name and description). `neotoma` also provides a set of tools to allow cross-site analysis, including the ability to standardize taxonomies using standard taxonomies from the published literature (limited to pollen taxa), or user provided taxonomies.

Here we illustrate how the `neotoma` package can be used in a paleoecological workflow using examples from the published literature, for both plant and mammal taxa.

Introduction

Paleoecological data are used to understand patterns and drivers of biogeographical, climatic and evolutionary change at multiple spatial and temporal scales. Paleoecoinformatics (Brewer et al. 2012; Uhen et al. 2013) is increasingly providing tools to researchers across disciplines to access and use large datasets spanning thousands of years. These datasets may be used to provide better insight into patterns of biomass burning (Marlon et al. 2013), regional vegetation change (Blois et al. 2013; Blarquez, Carcaillet, et al. 2014) or temporal change in sedimentation rates (Goring et al. 2012). The increasing interest in uniting ecological and paleoecological data to understand modern ecological patterns and responses to climate change (Fritz et al. 2013; Behrensmeyer & Miller 2012; Dietl & Flessa 2011) means that efforts to unite these two, seemingly independedent data-streams will rely, in part, on more robust tools to access and synthesize paleoecological data.

The Neotoma Paleoecological Database is the result of longstanding collaboration between the European Pollen Database and the North American Pollen Database (Grimm et al. 2013). The database framework was generalized from pollen data to accomodate additional macro- and microfossil data and geochemistry information, including loss-on-ignition and isotope records. Constituent databases include the European, Indo-Pacific, Latin American, North American Pollen Database, FAUNMAP (Pliocene to Quaternary mammal fossils in the continental United States), the North American Non-Marine Ostrocode Database, and the Diatom Paleolimnology Data Cooperative. Work is underway to include the North American Fossil Beetle Database, testate amoeba records, the North American Plant Macrofossil Database and the Digital Archaeological

Record, thus further expanding the data that can be accomodated by Neotoma. Through the use of data stewards - domain-specific experts who can check for inaccuracies, upload and manage data records - Neotoma can support high quality assurance for each of the constituent data types, and receive feedback from research communities involved with each specific data type (Grimm et al. 2013).

The authors of the paleoecological database Neotoma have also developed Application Programming Interfaces, which allow users to query the database using the internet and properly formed URLs. For example, the URL: http://api.neotomadb.org/v1/apps/geochronologies/?datasetid=8 will return all geochronological data for a single record associated with the datasetid provided.

The analysis of paleoecological data is commonly performed using the statistical software R (R Core Team 2014). There are several R packages that are particularly useful in a paleoecological workflow including `analogue` (Simpson & Oksanen 2013; Simpson 2007) and `rioja` (Juggins 2013) for paleoenvironmental reconstruction, `Bchron` for radiocarbon dating and age-depth modeling (Parnell 2014) and `paleofire` to access and analyse charcoal data (Blarquez, J. R. Marlon, et al. 2014). Notwithstanding these packages, the use of extensive paleoecological resources within R has traditionally relied on *ad hoc* methods of obtaining and importing data. This has meant reliance on datasets such as those from the NOAA Paleoclimate Repository or the North American Modern Pollen Database, and on the distribution of individual datasets from author to analyst.

With an increasing push to provide paleoecological publications that include numerically reproducible results (e.g., Goring et al. (2012); Gill et al. (2013); Goring et al. (2013)) it is important to develop user-friendly tools that both allow analysts to directly access dynamic datasets and support reproducible workflows. The rOpenSci project - one venue that allows developers to provide such tools - has provided a number of tools that can directly interact with application programmatic interfaces (APIs) to access data from a number of databases including rfishbase (FishBase: (Boettiger et al. 2012) and taxize (Encyclopedia of Life, iPlant/Taxosaurus and others: (Chamberlain & Szöcs 2013) among others.

The `neotoma` package addresses concerns regarding data access and workflow reproducibility by providing users with tools that allow paleoecologists to query, download, organize, and summarize data from the Neotoma database using R. First we describe the `neotoma` package, then we illustrate how `neotoma` can simplify the paleoecological workflow and allow users to perform research that is critical to understanding paleoecological change in the Pleistocene in an open and reproducible manner.

The `neotoma` package

`neotoma` is an R package that acts as an interface between a large dynamic database (the Neotoma Paleoecological Database: http://neotomadb.org) and statistical tools in R. The `neotoma` package uses an API to send data requests to Neotoma, and then forms data objects that can interact with existing packages such as `analogue` (Simpson & Oksanen 2013) and `rioja` (Juggins 2013), that are used for environmental reconstruction, manipulation and presentation of paleoecological data. The `neotoma` package also includes tools to standardize pollen data across sample sites using either a standard or user-defined set of pollen taxa.

Data in the neotoma package is represented in three main classes named `"site"`, `"dataset"`, and `"download"`. A `"site"` is the most basic form of spatial information, it is a representation of all data points, as a special class of `data.frame`. A `"site"` contains site names, locations and, when supplied, site descriptions, along with a unique `siteID`. Individual sites can be associated with one or more `"dataset"`s. A `"dataset"` is a special type of `list` that includes a `"site"` class for each dataset, but also includes information about the particular dataset, including the data type, the principle investigator, the submission date to Neotoma and the date that the information was accessed from Neotoma using the R package. The `"dataset"` also includes a unique `dataset.id` that can be used to access the full `"download"`. A `"download"` contains both `"site"` and `"dataset"` information, but it also contains the full data object for the dataset it references, whether it is pollen, ostrocode, mammal or other data.

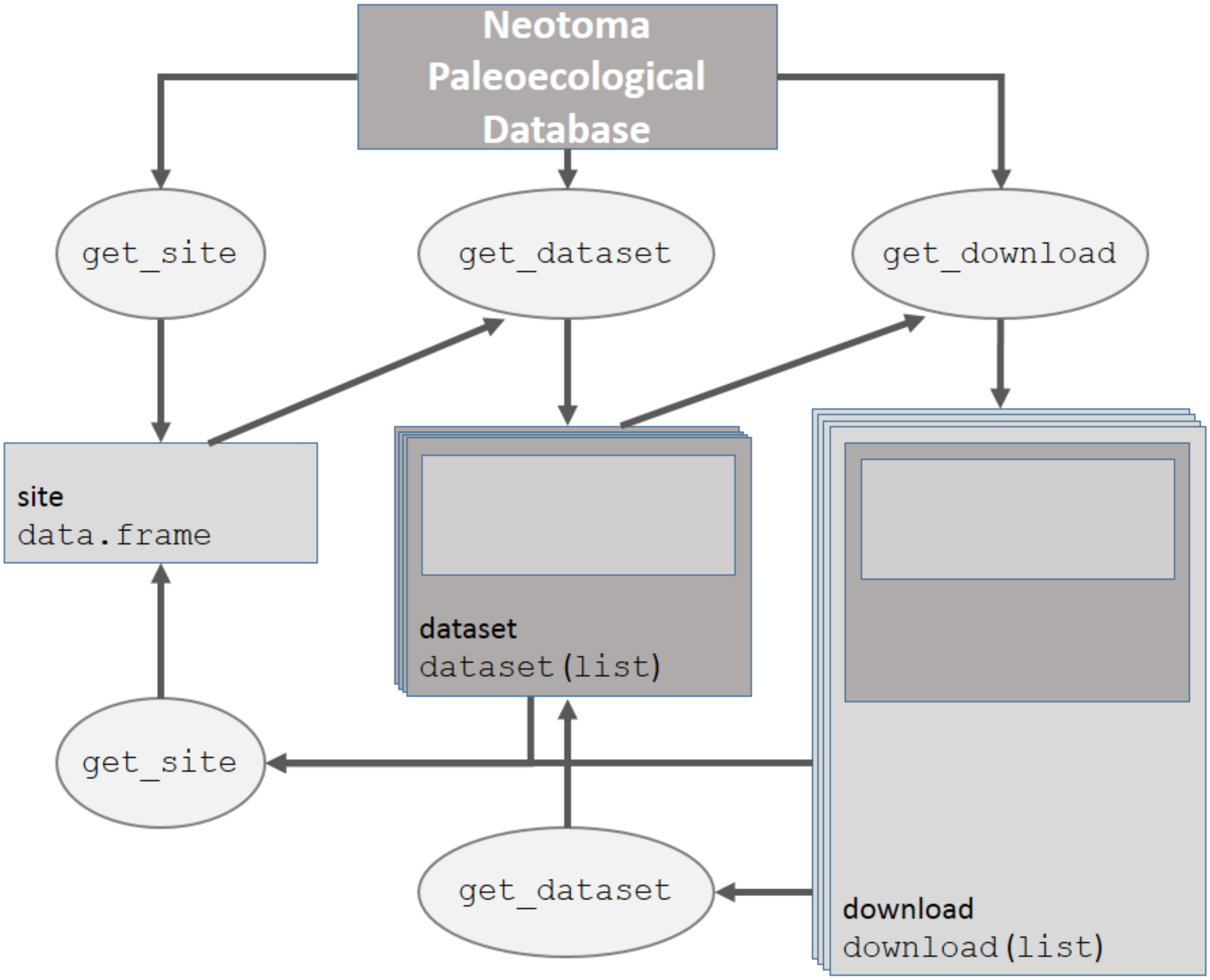


Figure 1. *How the main data objects relate to one another in the `neotoma` package, and the helper functions used to move from one data type to another.*

Each of these objects, `"site"`, `"dataset"` and `"download"` can be obtained using direct calls to the API, or using functions defined in the `neotoma` package, as desribed below.

Examples

Here we present several examples that both introduce users to the `neotoma` package, and highlight how `neotoma` can be used in a paleoecological worklfow. We beging with a simple example in which we compare change in Alnus between two sites, followed by two more involved examples where we look at Pine migration and mammal distributions.

A simple example

A researcher is interested in finding the pollen record for Marion Lake, in British Columbia (Mathewes 1973) and comparing the change in Alnus pollen to pollen from Louise Pond (Pellatt & Mathewes 1997) on Haida G'Waii, further north. We search for specific sites by name using `get_site()`, and make use of the wildcard `%` to catch all sites whose site names begin with `Marion Lake` or `Louise Pond`:

```r
library("neotoma")
library("analogue")
```

```
  Loading required package: vegan
  Loading required package: permute

  Attaching package: 'permute'

  The following object is masked from 'package:devtools':

      check

  Loading required package: lattice
  This is vegan 2.0-10
  This is analogue 0.14-0
```

```r
library("knitr")
marion <- get_site(sitename = "Marion Lake%")
```

```
  The API call was successful, you have returned  1 records.
```

```r
louise <- get_site(sitename = "Louise Pond%")
```

```
  The API call was successful, you have returned  1 records.
```

`get_site()` returns an object of class `"site"`, which is a data frame with columns `siteid`, `sitename`, `lat`, `long`, `elev`, `description`, `long_acc`, and `lat_acc`. Here we queried the database for site based on sitenames, but alternatively we could have queried the database for sites within a geographical bounding box, or by geopolitical region. Each row of the object returned by `get_site` represents a unique site, and provides enough descriptive data to plot site locations and understand the spatial extent of a site. Using the class assignment `"site"` allows objects returned by `get_site()` to be recognized by other functions, so that site information can easily be used to obtain datasets or whole data downloads. Sites, conceptually, are containers for datasets. Generally it's better to search for a neotoma dataset. The `neotoma` package allows you to use almost all of the same search terms in `get_dataset()` as in `get_site()`, and returns a more complete description of the datasets available, however, `get_site()` is the only method by which you can search by site name.

To get the `"dataset"` for these records we can simplify the workflow by `rbind()`ing the two site records, and then using `get_dataset()` directly (Figure 1):

```r
western.sites <- rbind(marion, louise)
western.data <- get_dataset(western.sites)
```

A `"dataset"` is a `list` with additional class `"dataset"`. The dataset contains site information, but it also contains information about the specific dataset, so the site data is nested within the dataset. The use of a specific `"dataset"` class allows us to easily move between `get_dataset()`, `get_site()` and `get_download()` by using S3 methods in R.

To obtain the actual pollen records we use `get_download()` directly on the dataset returned from our earlier `get_dataset()` call (Figure 1):

```
western.dl <- get_download(western.data)
```

```
API call was successful. Returned record for Marion Lake(CA:British Columbia)
API call was successful. Returned record for Louise Pond

Warning:
Modifiers are absent from the lab objects Lycopodium tablets, Lycopodium spike, Sample quantity.
get_download will use uniqueidentifiers to resolve the problem.
```

`get_download()` returns an object of class `"download"`, which is a `list` of length equal to the number of sites returned. In most cases `get_download()` will return a message for an individual core as it is running, that can be turned off using the argument `verbose = FALSE`.

Both the `get_download()` and `get_dataset()` functions also record the time at which the API was accessed. Because of the large size of most `"download"`s there is a special print function that limits output size, however, the objects remain `lists` and can be manipulated as such in R.

A single `"download"` object is a `list` with six components:

```
names(western.dl[[1]])
```

```
[1] "metadata"     "sample.meta"  "taxon.list"   "counts"
[5] "lab.data"     "chronologies"
```

The `metadata` component is equivalent to a `"dataset"` returned by `get_dataset()`. The `sample.meta` component is where the core depth and age information is stored. The actual chronologies are stored in `chronologies`. If a core has a single age model then `chronologies` has a length of one. Some cores have multiple chronologies and these are added to the list. The default chronology is always represented in `sample.meta`, and is always the first `chronology`. If you choose to build your own chronology using `Bacon` (Blaauw & Christen 2011) or another method you can obtain the chronological controls for the core using `get_chroncontrol()` with the chronology ID in either `sample.meta` or any one of the `chronologies` objects. While the chronological controls used to build a chronology may vary across chronologies for a single site, the default model contains the most accurate chronological control data.

The `taxon.list` component is a critical part of the `"download"` object. It lists the taxa found in the core, as well as any laboratory data, along with the units of measurement and taxonomic grouping. This is important information for determining which taxa make it into pollen percentages. The `counts` are the actual count or percentage data recorded for the core. The `lab.data` component contains information about any spike used to determine concentrations, sample quantities and, in some cases, charcoal counts.

We have 2 records downloaded, one for Marion Lake and one for Louise Pond. Pollen taxonomy can vary substantially across cores often depending on researcher skill, or changing taxonomies for species, genera or families over time. This shifting taxonomy is often problematic to deal with. The `neotoma` package implements a taxonomic standardization function which allows users to standardize taxa to one of four published taxonomies for the United States and Canada. While this function can be helpful in many cases it should also be used with care. The aggregation table is accessible using `data(pollen.equiv)` and the function to compile the data is called `compile_taxa()`. It can accomodate either the internal translation table provided with the package, or a user defined translation table.

In this case we are interested in the percentage of *Alnus* in the core, so we can compile the taxa to the most straightforward taxonomy, ‘P25’ from Gavin et al. (2003). The first record downloaded is Marion Lake. We can see in the `"download"` the `taxon.table` has 5 columns:

```
head(western.dl[[1]]$taxon.list)
```

	TaxonName	VariableUnits	VariableElement
2	Unknown (monolete)	NISP	spore
29	Unknown	NISP	pollen/spore
3	Myrica	NISP	pollen
4	Poaceae	NISP	pollen
5	Alnus	NISP	pollen
6	Tsuga mertensiana	NISP	pollen

Table 1: Table continues below

	VariableContext	TaxaGroup
2		Unidentified palynomorphs
29		Unidentified palynomorphs
3		Vascular plants
4		Vascular plants
5		Vascular plants
6		Vascular plants

Once we apply `compile_taxa()` to the dataset using the 'P25' compiler:

```
western.comp <- compile_taxa(western.dl, list.name = "P25")
names(western.comp) <- c("marion", "louise")
```

we can see that the `taxon.table` now has an extra column (note that we've removed several columns to improve readability here).

This block shows just the code necesssary but skips the formatting parts needed to make the table play nice with pandoc

```
head(western.comp[[1]]$taxon.list[, c(1, 5, 6)])
```

	TaxonName	TaxaGroup	compressed
2	Unknown (monolete)	Unidentified palynomorphs	Other
29	Unknown	Unidentified palynomorphs	Other
3	Myrica	Vascular plants	Other

	TaxonName	TaxaGroup	compressed
4	Poaceae	Vascular plants	Poaceae
5	Alnus	Vascular plants	Alnus
6	Tsuga mertensiana	Vascular plants	Tsuga

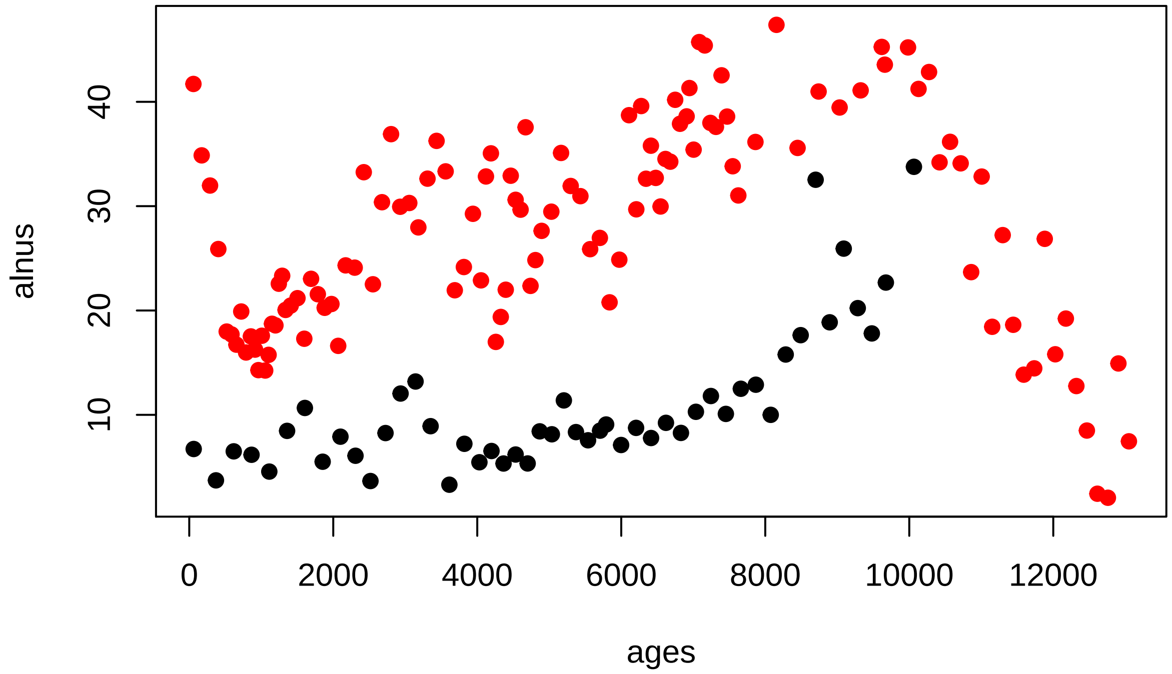
`compile_taxa()` returns an object that looks exactly like the `"download"` object passed to it, however, the `taxon.list` data frame gains a column named `compressed` that links the original taxonomy to the revised taxonomy. This acts as an important check for researchers who choose to use this package for large-scale analysis. Here we see that all the spore types listed have been lumped into a single *Other*. The `compile_taxa()` function can also accept user-defined tables for aggregation if the provided compilations are not acceptable.

In this case the counts look reasonable, and the synonomy appears to have been applied correctly (although we're really only interested in *Alnus*). We now transform our `counts` into percentages to standardize across cores. We can see what happens with *Alnus* on the west coast of North America during the Holocene:

```r
marion.alnus <- tran(x = western.comp$marion$counts, method = 'percent')[,'Alnus']
louise.alnus <- tran(x = western.comp$louise$counts, method = 'percent')[,'Alnus']

alnus.df <- data.frame(alnus = c(marion.alnus, louise.alnus),
                      ages  = c(western.comp$marion$sample.meta$Age,
                                western.comp$louise$sample.meta$Age),
                      site = c(rep('Marion', length(marion.alnus)),
                               rep('Louise', length(louise.alnus))))

plot(alnus ~ ages, data = alnus.df, col = alnus.df$site, pch = 19)
```



In this example we see that Marion Lake (red) maintains much higher proportions of *Alnus* throughout it's history, and has a rapid increase in *Alnus* pollen during the historical period. This rapid shift in the last 200 years is likely as a result of rapid colonization by pioneer *Alnus rubra* following forest clearance in the lower mainland of British Columbia.

Pinus migration following the last Glacial Maximum

Macdonald and Cwynar (1991) used pollen percentage data for *Pinus* to map the northward migration of lodgepole pine (*Pinus contorta* var *latifolia*) following glaciation. In their study a cutoff of 15% *Pinus* pollen is associated with presence at pollen sample sites. Recent work by Strong and Hills (Strong & Hills 2013) has remapped the migration front using a lower pollen proportion (5%) and more sites. Here, we attempt to replicate the analysis as an example both of the strengths of the package and limitations of paleoinformatic approaches.

To begin we must define a spatial bounding box and a set of taxa of interest. Strong and Hills (2013) use a region approximately bounded by 54°N to the south and 65°N to the North, and from 110°W to 130°W. We can use `get_site()` to find all sites within a bounding box:

```
# install.packages('ggmap', 'ggplot2', 'reshape2', 'plyr', 'Bchron',
# 'gridExtra')
library("ggmap")
library("ggplot2")
library("reshape2")
library("plyr")
library("Bchron")
library("gridExtra")
all.sites <- get_site(loc = c(-140, 45, -110, 65))
```

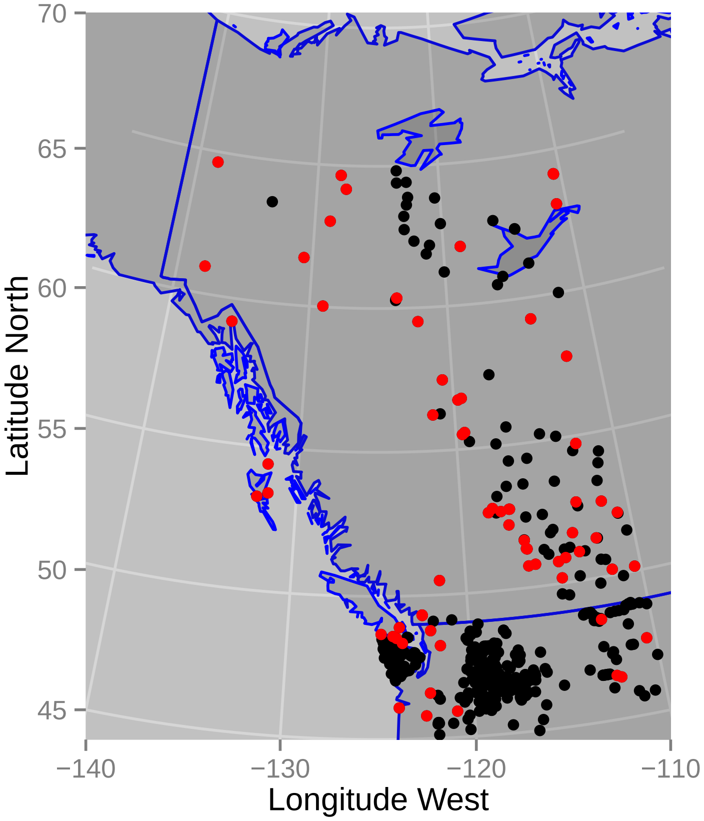
```
  The API call was successful, you have returned  443 records.
```

Our code above returns a total of 443 sites. While `get_site` is similar to `get_dataset()`, `get_dataset()` can also limit the type of dataset, either by looking for specific taxa, or by describing the dataset type (*e.g.*, `'pollen'` or `'mammal'`). In the next example we will look for all taxa beginning with *Pinus* in a pollen dataset within a bounding box corresponding to Washington State in the USA and British Columbia and the Yukon Territory in Canada. We use the `*` wildcard to indicate any and all taxa with *Pinus* in their name:

```
all.datasets <- get_dataset(loc = c(-140, 45, -110, 65), datasettype = "pollen",
    taxonname = "Pinus*")
```

The API tells us we now have 69 datasets from the original 443 sites. Many of the samples that are not included are pollen surface samples, or vertebrate fauna, meaning pollen core data comprises much less than half of the records. Regardless, we now know that there is pollen core data from 69 sites and we can plot those sites over our original 443.

```
map <- map_data("world")
ggplot(data = data.frame(map), aes(long, lat)) + geom_polygon(aes(group = group),
    color = "blue", alpha = 0.2) + geom_point(data = all.sites, aes(x = long,
    y = lat)) + geom_point(data = get_site(all.datasets), aes(x = long, y = lat),
    color = 2) + xlab("Longitude West") + ylab("Latitude North") + coord_map(projection = "albers",
    lat0 = 40, lat1 = 65, xlim = c(-140, -110), ylim = c(45, 70))
```



So we see that there are a number of sites in the interior of British Columbia that have no core pollen. For many of these cores pollen records exist. This is an obvious limitation of the use of large datasets. While many dataset have been entered into Neotoma, a large number have yet to make their way into the repository. An advantage of the API-based analysis however is that analysis using Neotoma can be updated continuously as new sites are added.

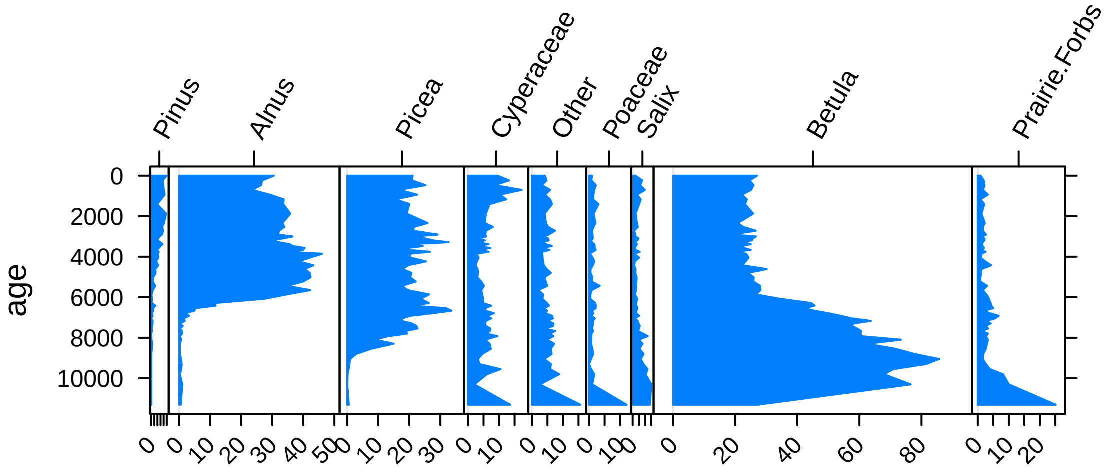
Let's get the data for each of the cores we have. Because we have assigned the `"dataset"` class to the output of `get_dataset()` it means that the function `get_download()` can immediately recognize the `"dataset"` object and extract the dataset ID from it to use in obtaining full records from the API:

```r
# This step may be time consuming when you run it, particularly if you have
# a slow internet connection.
if (!file.exists("all.downloads.Rdata")) {
    all.downloads <- get_download(all.datasets, verbose = FALSE)
    save(all.downloads, file = "all.downloads.Rdata")
} else {
    load(file = "all.downloads.Rdata")
}
```

For our purposes we are really only interested in the percentage of *Pinus* in the core, so we can again compile the taxa using the 'P25' taxonomy (Gavin et al. 2003).

In this case the synonomy (not shown) appears to have been applied correctly (although we're really only interested in *Pinus*). We now transform our `counts` into percentages to standardize across cores. We can see what a single core looks like:

```r
compiled.cores <- compile_taxa(all.downloads, "P25")
core.pct <- data.frame(tran(compiled.cores[[1]]$counts, method = "percent"))
core.pct$age <- compiled.cores[[1]]$sample.meta$Age
# Eliminate taxa with no samples greater than 4%.
core.pct <- chooseTaxa(core.pct, max.abun = 4)
# Plotted using the Stratiplot function in 'analogue', very naive plotting
# for demonstration.
Stratiplot(age ~ ., core.pct, sort = "wa", type = "poly")
```

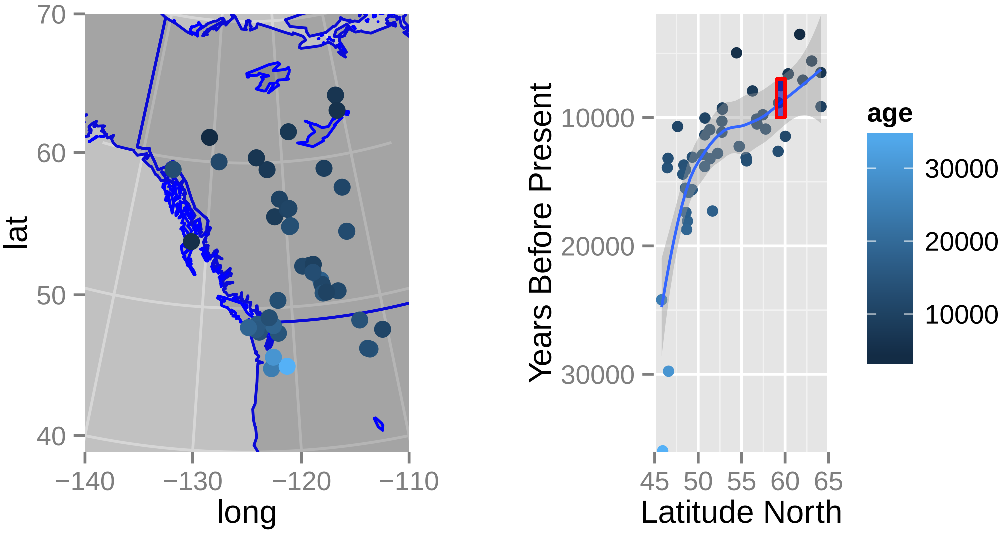


Andy Lake (Szeicz et al. 1995) shows changes through time, particularly for *Betula* and *Alnus*, but little *Pinus* pollen.

Pollen data is found in the `counts` component of the `"download"`. We want to determine which sample has the first local *Pinus* presence in each core using a cutoff of 5% (Strong & Hills 2013). Programmatically we can find which rows in the *Pinus* column have presence over 5% and then find the highest row number since age increases with row number.

```
top.pinus <- function(x) {
    # Convert the core data into proportions.
    x.pct <- tran(x$counts, method = "proportion")
    # Cores must span at least 5000 years (and have non NA dates), otherwise
    # they date the arrival of Pinus too late!
    old.enough <- max(x$sample.meta$Age) > 5000 & !all(is.na(x$sample.meta$Age))
    # Find the highest row index associated with Pinus presence over 5%
    oldest.row <- ifelse(any(x.pct[, "Pinus"] > 0.05 & old.enough), max(which(x.pct[,
        "Pinus"] > 0.05)), 0)
    # return a data frame with site name and locations, and then the age and
    # date type associated with the oldest recorded Pinus presence.  We preserve
    # date type since some records have ages in radiocarbon years.
    if (oldest.row > 0) {
        return(data.frame(site = x$metadata$site.data$sitename, lat = x$metadata$site.data$lat,
            long = x$metadata$site.data$long, age = x$sample.meta$Age[oldest.row],
            date = x$sample.meta$AgeType[oldest.row]))
    } else {
        return(NULL)
    }
}
# Apply the function 'top.pinus' to each core (here we use the plyr function
# ldply so we can pass in a list (compiled.cores) and return a data.frame.
summary.pinus <- ldply(compiled.cores, top.pinus)
# We need to calibrate dates that are recorded in radiocarbon years.  In
# most cases we have no idea what the uncertainty was.  For this example I
# am simply assuming a 100 year SD for calibration.  This is likely too
# small for some earlier dates, but we use it as an example here:
radio.years <- summary.pinus$date %in% "Radiocarbon years BP"
# BChronCalibrate is a function in the BChron package:
calibrated <- BchronCalibrate(summary.pinus$age[radio.years], ageSds = rep(100,
    sum(radio.years, na.rm = TRUE)), calCurves = rep("intcal13", sum(radio.years,
    na.rm = TRUE)))
# calibrated contains the full calibration curve for each date, we want the
# weighted mean:
wmean.date <- function(x) sum(x$ageGrid * x$densities/sum(x$densities))
summary.pinus$age[radio.years] <- sapply(calibrated, wmean.date)
summary.pinus <- na.omit(summary.pinus)
summary.pinus <- summary.pinus[!((summary.pinus$age < 2000) & (summary.pinus$long <
    -130)), ]
# We're using a loess curve here but the curve can be improved by using a
# monotone spline.
regress <- ggplot(summary.pinus, aes(x = lat, y = age)) + geom_point(aes(color = age),
    size = 2) + scale_y_reverse(expand = c(0, 100)) + xlab("Latitude North") +
    ylab("Years Before Present") + geom_smooth(n = 40, method = "loess") + geom_rect(aes(xmin = 59,
    xmax = 60, ymin = 7000, ymax = 10000), color = 2, fill = "blue", alpha = 0.01)
mapped <- ggplot(data = data.frame(map), aes(long, lat)) + geom_polygon(aes(group = group),
    color = "blue", alpha = 0.2) + geom_point(data = summary.pinus, aes(x = long,
    y = lat, colour = age), size = 3) + coord_map(projection = "albers", lat0 = 40,
    lat1 = 65, xlim = c(-140, -110), ylim = c(40, 70)) + theme(legend.position = "none")
```

```
grid.arrange(mapped, regress, nrow = 1)
```



And so we see a clear pattern of migration by *Pinus* in northwestern North America. These results match up broadly with the findings of Strong and Hills (Strong & Hills 2013) who suggest that *Pinus* reached a northern extent between 59 and 60°N at approximately 7 - 10kyr as a result of geographic barriers.

Mammal Distributions in the Pleistocene

Graham et al. (Graham et al. 1996) look for patterns of change in mammal distributions through the Pleistocene to modern era using fossil assemblages collated from FAUNMAP. Using involved data analysis methods, they show that mammal species have responded in a Gleasonian manner to climate change since the late-Pleistocene. Their results indicate some species migrating northward in response to warming climates, others staying relatively stable and some moving southward. Since FAUNMAP has been incorporated into Neotoma we aim to replicate tests of species distributional changes in a straightforward manner to demonstrate the utility of `neotoma` in analysing mammal distributions and change through time.

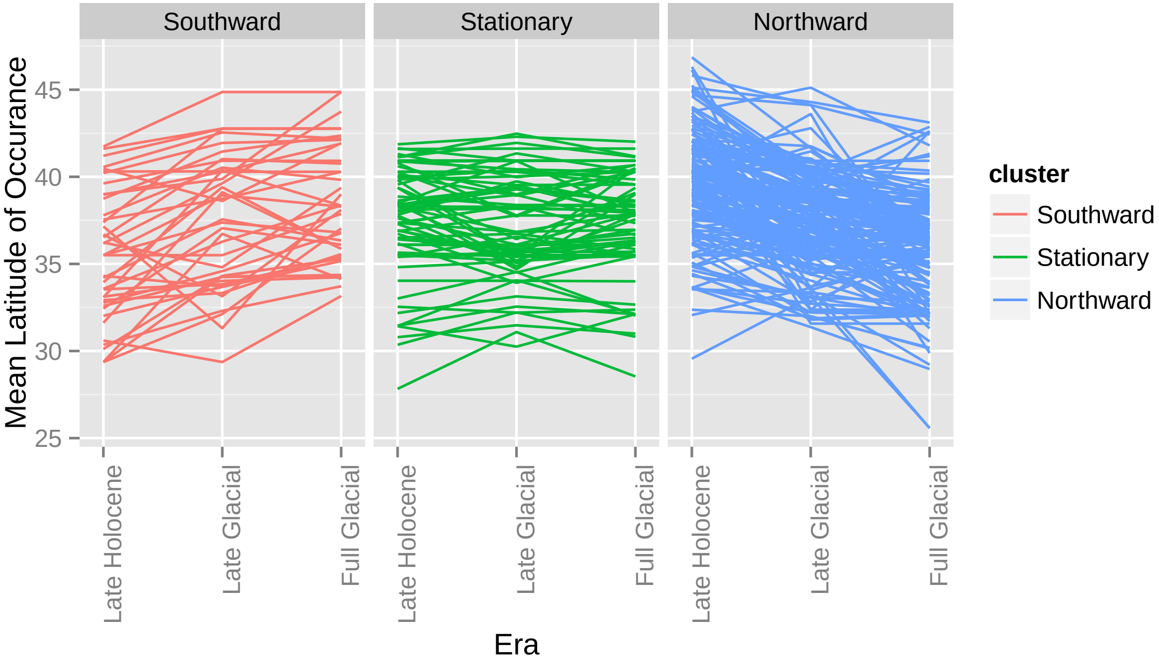
First we need to obtain all fossil assemblages from Neotoma for vertabeate fauna,

```
# Bounding box is effectively the continental USA, excluding Alaska.
mam.set <- get_dataset(datasettype = "vertebrate fauna", loc = c(-125, 24, -66,
    49.5))
# Calling this many sites can be very time consuming.  It takes
# approximately an hour to run fully.
if (!file.exists("mam.dl.Rdata")) {
    mam.dl <- get_download(dataset = mam.set)
    save(mam.dl, file = "mam.dl.Rdata")
} else {
    load("mam.dl.Rdata")
}
```

So, now we have all the sites, we need to bin them into time periods as in Graham et al. (Graham et al. 1996). To do that we first need to build a large table with time and `xy` coordinates for each site. Time data in `sample.meta` is not the same as for for pollen data, where many pollen sites contain an age (often mean age) and upper and lower bounds. Most mammal sites have younger and older bounds, but no estimates of exact age. In this case we take a short-cut and simply average the younger and older bounds to save the reader from having to examine too much code.

```r
compiled.mam <- compile_downloads(mam.dl)
# We assign time bins to the data.  The command findInterval should tell us
# if it is in an inteval equivalent to the Modern (0 - 500ybp), Late
# Holocene (500 - 4000ybp), Early-Mid Holocene (4kyr - 10kyr), Late Glacial
# (10kyr - 15kyr), Full Glacial (15kyr - 20kyr) or Late Pleistocene
# (20kyr+).
time.bins <- c(500, 4000, 10000, 15000, 20000)
# This is not the best option, age bounds cross our pre-defined bins,
# however solving this is more complex than this example requires.
mean.age <- rowMeans(compiled.mam[, c("ageold", "ageyoung", "age")], na.rm = TRUE)
interval <- findInterval(mean.age, time.bins)
periods <- c("Modern", "Late Holocene", "Early-Mid Holocene", "Late Glacial",
    "Full Glacial", "Late Pleistocene")
compiled.mam$ageInterval <- periods[interval + 1]
# The melt and dcast commands are in reshape2
mam.melt <- melt(compiled.mam, measure.vars = 10:(ncol(compiled.mam) - 1), na.rm = TRUE,
    factorsAsStrings = TRUE)
mam.melt$ageInterval <- factor(mam.melt$ageInterval, levels = periods)
mam.lat <- dcast(data = mam.melt, variable ~ ageInterval, value.var = "lat",
    fun.aggregate = mean, drop = TRUE)[, c(1, 3, 5, 6)]
# We only want taxa that appear at all time periods:
mam.lat <- mam.lat[rowSums(is.na(mam.lat)) == 0, ]
# Group the samples based on the range & direction (N vs S) of migration.
mam.lat$grouping <- factor(findInterval(mam.lat[, 2] - mam.lat[, 4], c(-11,
    -1, 1, 20)), labels = c("Southward", "Stationary", "Northward"))
mam.lat.melt <- melt(mam.lat)
colnames(mam.lat.melt)[2:3] <- c("cluster", "Era")
```

```r
ggplot(mam.lat.melt, aes(x = Era, y = value)) + geom_path(aes(group = variable,
    color = cluster)) + facet_wrap(~cluster) + scale_x_discrete(expand = c(0.1,
    0)) + ylab("Mean Latitude of Occurance") + theme(axis.text.x = element_text(angle = 90,
    hjust = 1))
```



So we can see that at this basic analytic scale species are not uniformly responding to climatic warming following deglaciation. These findings in essence echo those of Graham et al. (Graham et al. 1996) who showed that taxon response is largely individualistic. While we do see the pre-ponderance of migration is northward, a number of taxa show little migratory response and a number show southward migration. For the sake of brevity, we do not consider movement to the west or east, and ignore potential issues associated with the complex topography of the mountainous west. Regardless, it is clear that the use of `neotoma` can support research that is reproducible and robust.

Conclusion

The increasing pressure to develop large-scale databases requires the development of tools that can access the data and can leave reproducible analyses so that others can build from and verify results.

Here we present the `neotoma` package for R (R Core Team 2014) and use examples from the literature to show its utility. `neotoma` joins a number of other existing packages that are designed either to exploit exisiting paleoecological datasets (**???**) or to manipulate paleoecological data (Simpson & Oksanen 2013; Simpson 2007; Juggins 2013). The `neotoma` package itself is available either from the CRAN repository, or from GitHub where ongoing development continues with help from the public.

The use of the Neotoma database continues to expand, and here we provide researchers with the tools to move analytics to an open framework using R (R Core Team 2014) so that methods can be fully visible. *Add text here about reproducibility, check the Uhen papers & stuff too.*

References

Behrensmeyer, A.K. & Miller, J.H., 2012. Building links between ecology and paleontology using taphonomic studies of recent vertebrate communities. In *Paleontology in ecology and conservation*. Springer, pp. 69–91.

Blaauw, M. & Christen, J.A., 2011. Flexible paleoclimate age-depth models using an autoregressive gamma process. *Bayesian Analysis*, 6(3), pp.457–474.

Blarquez, O. et al., 2014. Disentangling the trajectories of alpha, beta and gamma plant diversity of North American boreal ecoregions since 15,500 years. *Frontiers In Paleoecology*, 2, p.6.

Blarquez, O. et al., 2014. paleofire: an r package to analyse sedimentary charcoal records from the global charcoal database to reconstruct past biomass burning. *Computers & Geosciences*.

Blois, J.L. et al., 2013. Modeling the climatic drivers of spatial patterns in vegetation composition since the Last Glacial Maximum. *Ecography*, 36(4), pp.460–473.

Boettiger, C., Lang, D. & Wainwright, P., 2012. rfishbase: exploring, manipulating and visualizing FishBase data from R. *Journal of Fish Biology*, 81(6), pp.2030–2039.

Brewer, S., Jackson, S.T. & Williams, J.W., 2012. Paleoecoinformatics: applying geohistorical data to ecological questions. *Trends in Ecology & Evolution*, 27(2), pp.104–112.

Chamberlain, S.A. & Szöcs, E., 2013. taxize: taxonomic search and retrieval in R. *F1000Research*, 2.

Dietl, G.P. & Flessa, K.W., 2011. Conservation paleobiology: putting the dead to work. *Trends in Ecology & Evolution*, 26(1), pp.30–37.

Fritz, S.A. et al., 2013. Diversity in time and space: wanted dead and alive. *Trends in Ecology & Evolution*, 28(9), pp.509–516.

Gavin, D.G. et al., 2003. A statistical approach to evaluating distance metrics and analog assignments for pollen records. *Quaternary Research*, 60(3), pp.356–367.

Gill, J.L. et al., 2013. Linking abundances of the dung fungus sporormiella to the density of bison: implications for assessing grazing by megaherbivores in palaeorecords. *Journal of Ecology*, 101(5), pp.1125–1136.

Goring, S. et al., 2013. Pollen assemblage richness does not reflect regional plant species richness: a cautionary tale. *Journal of Ecology*, 101(5), pp.1137–1145.

Goring, S. et al., 2012. Deposition times in the northeastern United States during the Holocene: establishing valid priors for Bayesian age models. *Quaternary Science Reviews*, 48, pp.54–60.

Graham, R.W. et al., 1996. Spatial response of mammals to late quaternary environmental fluctuations. *Science*, 272, pp.1601–1606.

Grimm, E. et al., 2013. Databases and their application.

Juggins, S., 2013. *rioja: Analysis of Quaternary science data*, Available at: http://www.staff.ncl.ac.uk/staff/stephen.juggins/.

MacDonald, G. & Cwynar, L.C., 1991. Post-glacial population growth rates of *pinus contorta* ssp. *latifolia* in western Canada. *The Journal of Ecology*, pp.417–429.

Marlon, J.R. et al., 2013. Global biomass burning: a synthesis and review of holocene paleofire records and their controls. *Quaternary Science Reviews*, 65, pp.5–25.

Mathewes, R.W., 1973. A palynological study of postglacial vegetation changes in the University Research Forest, southwestern British Columbia. *Canadian Journal of Botany*, 51(11), pp.2085–2103.

Parnell, A., 2014. *Bchron: Radiocarbon dating, age-depth modelling, relative sea level rate estimation, and non-parametric phase modelling*, Available at: http://CRAN.R-project.org/package=Bchron.

Pellatt, M.G. & Mathewes, R.W., 1997. Holocene tree line and climate change on the Queen Charlotte Islands, Canada. *Quaternary Research*, 48(1), pp.88–99.

R Core Team, 2014. *R: A language and environment for statistical computing*, Vienna, Austria: R Foundation for Statistical Computing. Available at: http://www.R-project.org/.

Simpson, G.L., 2007. Analogue methods in palaeoecology: Using the analogue package. *Journal of Statistical Software*, 22(2), pp.1–29.

Simpson, G.L. & Oksanen, J., 2013. *analogue: Analogue and weighted averaging methods for palaeoecology*, Available at: http://cran.r-project.org/package=analogue.

Strong, W.L. & Hills, L.V., 2013. Holocene migration of lodgepole pine (*pinus contorta* var. *latifolia*) in southern Yukon, Canada. *The Holocene*, 23(9), pp.1340–1349.

Szeicz, J.M., MacDonald, G.M. & Duk-Rodkin, A., 1995. Late Quaternary vegetation history of the central Mackenzie Mountains, Northwest Territories, Canada. *Palaeogeography, Palaeoclimatology, Palaeoecology*, 113(2), pp.351–371.

Uhen, M.D. et al., 2013. From card catalogs to computers: databases in vertebrate paleontology. *Journal of Vertebrate Paleontology*, 33(1), pp.13–28.