

LinkedEarth - Neotoma - P418 Use Case

Isabel & age uncertainty

Allow me to paint a picture for you. A user (let's call her Isabel) is interested in investigating the impact of time uncertainty on a pollen-based temperature reconstruction at Basin Pond. The temperature data are available at LinkedEarth, but the geochronologic data and depth data are not archived there. The pollen, depth and geochronologic data are available at Neotoma, but the temperature data are not. Isabel wants to combine these two datasets, and then use the GeoChronR package to do age-uncertain visualization.

This is a story of how she might do that.

1. Isabel decides to search for the Basin Pond data geographically, and creates a box around the site.

```
locpoly <- matrix(c(-70,-71,-71,-70,45,45,44,44),ncol=2,byrow = F)
```

2. She then converts that array to GeoJSON FeatureCollection, so she can query GeoDex

```
#install.packages("geojson")
library(geojson)
```

```
##
```

```
## Attaching package: 'geojson'
```

```
## The following object is masked from 'package:graphics':
```

```
##
```

```
##      polygon
```

```
poly <- geojson::polygon(paste0('{ "type": "Polygon", "coordinates": [[' , "[" ,locpoly[1,1] ,", " ,locpoly
```

```
feature <- geojson::feature(poly)
feature_collection <- geojson::featurecollection(feature)
feature_collection_str <- geo_pretty(feature_collection)
```

3. She then sets options for spatial searching, the domain and request URL details. Please see <http://geodex.org/swagger-ui/> for a complete description of the web service call formats.

```
search_type <- "spatial/search/object"
domain_url <- "http://geodex.org/api/v1/"
request_url <- paste(domain_url, search_type, sep="")
```

4. She creates an R list data structure to hold URL submission parameters

```
params_list <- list(geowithin = feature_collection_str)
```

5. Then makes a call to the Geodex RESTful web service using the requests package.

```
library(httr)
r <- GET(request_url, query = params_list)
results <- content(r, "text", encoding = "ISO-8859-1")
```

6. OK! The first results. She takes a look at what her spatial search returned.

```
library(rjson)
results_json <- fromJSON(results)
features <- results_json$features
number_of_results <- length(features)
```

```
first_result <- features[[1]]
url <- first_result$properties$URL
```

7. Since it's a spatial search, she decides to sort results by distance to polygon

```
library(geosphere)
#find nearest coord in each feature
nearestInFeature <- function(feats){
  if(is.list(feats$geometry$coordinates)){
    coords <- matrix(unlist((feats$geometry$coordinates)),ncol=2,byrow = T)
  }else{
    coords <- matrix((feats$geometry$coordinates),ncol=2,byrow = T)
  }
  md =try(min(geosphere::distGeo(coords,locpoly[1,])))
  if(is.numeric(md)){
    return(md)
  }else{
    return(NA)
  }
}

nearestDist <- sapply(results_json$features,nearestInFeature)

#sort by nearest
results_json$features <- results_json$features[order(nearestDist)]
allUrls <- sapply(results_json$features,FUN = function(x){x$properties$URL})
geometry <- first_result$geometry$type
coordinates <- first_result$geometry$coordinates[[1]]

#print first 10 results
print(allUrls[1:10])
```

```
## [1] "http://data.neotomadb.org/datasets/1490/"
## [2] "http://data.neotomadb.org/datasets/190/"
## [3] "http://data.neotomadb.org/datasets/1509/"
## [4] "http://data.neotomadb.org/datasets/1489/"
## [5] "http://data.neotomadb.org/datasets/1493/"
## [6] "http://data.neotomadb.org/datasets/1492/"
## [7] "http://wiki.linked.earth/NAM-BasinPond.Gajewski.1988"
## [8] "http://data.neotomadb.org/datasets/237/"
## [9] "http://data.neotomadb.org/datasets/1494/"
## [10] "http://wiki.linked.earth/NAM-ElephantMountain.Conkey.1994"
```

She recognizes those results! Numbers 2 & 7 are what she's looking for.

8. Now she explores the metadata for the LinkedEarth result to find the download link, doing another query to geodex

```
dataset_url <- allUrls[7]

search_type <- "graph/details"
```

9. She creates the URL for the RESTful web service call.

```
domain_url <- "http://geodex.org/api/v1/"

request_url <- paste(domain_url, search_type, sep="")
```

10. Creates and executes the GET parameter for this call.

```
params_list <- list(r = dataset_url)
r <- GET(request_url, query = params_list)
results <- content(r, "text", encoding = "ISO-8859-1")
```

11. And prints the results in a useful way.

```
results_json <- fromJSON(results)
print(paste("URI =", results_json$S))
```

```
## [1] "URI = t3522624"
```

```
print(paste("Alternate Name =", results_json$Aname))
```

```
## [1] "Alternate Name = "
```

```
print(paste("Name =", results_json$Name))
```

```
## [1] "Name = "
```

```
print(paste("URL =", results_json$URL))
```

```
## [1] "URL = http://wiki.linked.earth/NAM-BasinPond.Gajewski.1988"
```

```
print(paste("Description =", results_json$Description))
```

```
## [1] "Description = "
```

```
print(paste("Citation =", results_json$Citation))
```

```
## [1] "Citation = t3522626"
```

```
print(paste("Data Published =", results_json$Datepublished))
```

```
## [1] "Data Published = "
```

```
print(paste("Dataset Download Link =", results_json$Curl))
```

```
## [1] "Dataset Download Link = http://wiki.linked.earth/wiki/index.php/Special:WTLiPD?op=export&lipdid"
```

```
print(paste("Keywords =", results_json$Keywords))
```

```
## [1] "Keywords = paleoclimate, climate"
```

```
print(paste("License =", results_json$License))
```

```
## [1] "License = "
```

12. Aha! There's the download URL she needs! She'll use the LiPD-utilities package to load that dataset into R

```
#devtools::install_github("nickmckay/lipd-utilities", subdir = "R")
library(lipdR)
L2 <- readLipd(results_json$Curl)
```

```
## Please enter the dataset name for this file (Name.Location.Year) :
```

```
## [1] "reading: Downloads.lpd"
```

13. Now to get her hands on those Neotoma data. She needs the site ID from the Neotoma metadata, and performs another GeoDex query to achieve this...

```
dataset_url <- allUrls[2]
```

... creating the GET parameter for this call ...

```
params_list <- list(r = dataset_url)
```

... and executing the RESTful web service call.

```
r <- GET(request_url, query = params_list)
results <- content(r, "text", encoding = "ISO-8859-1")
```

14. She prints these results in a user friendly way.

```
results_json <- fromJSON(results)
print(paste("URI =", results_json$S))
```

```
## [1] "URI = t3611777"
```

```
print(paste("Alternate Name =", results_json$Aname))
```

```
## [1] "Alternate Name = "
```

```
print(paste("Name =", results_json$Name))
```

```
## [1] "Name = "
```

```
print(paste("URL =", results_json$URL))
```

```
## [1] "URL = http://data.neotomadb.org/datasets/190/"
```

```
print(paste("Description =", results_json$Description))
```

```
## [1] "Description = "
```

```
print(paste("Citation =", results_json$Citation))
```

```
## [1] "Citation = "
```

```
print(paste("Data Published =", results_json$Datepublished))
```

```
## [1] "Data Published = "
```

```
print(paste("Dataset Download Link =", results_json$Curl))
```

```
## [1] "Dataset Download Link = "
```

```
print(paste("Keywords =", results_json$Keywords))
```

```
## [1] "Keywords = "
```

```
print(paste("License =", results_json$License))
```

```
## [1] "License = https://creativecommons.org/licenses/by/4.0/deed.en_US"
```

TBD: This search needs needs to return a site.id or a site name.

15. Now she can load in Neotoma data as a LiPD object too!

```
#devtools::install_github("nickmckay/geochronr")
library(geoChronR)
library(neotoma)
```

```
##
```

```
## Attaching package: 'neotoma'
```

```
## The following object is masked from 'package:lipdR':
##
##      get_table
L = neotoma2Lipd(site = 234) #PULL THIS FROM QUERY RESULTS!

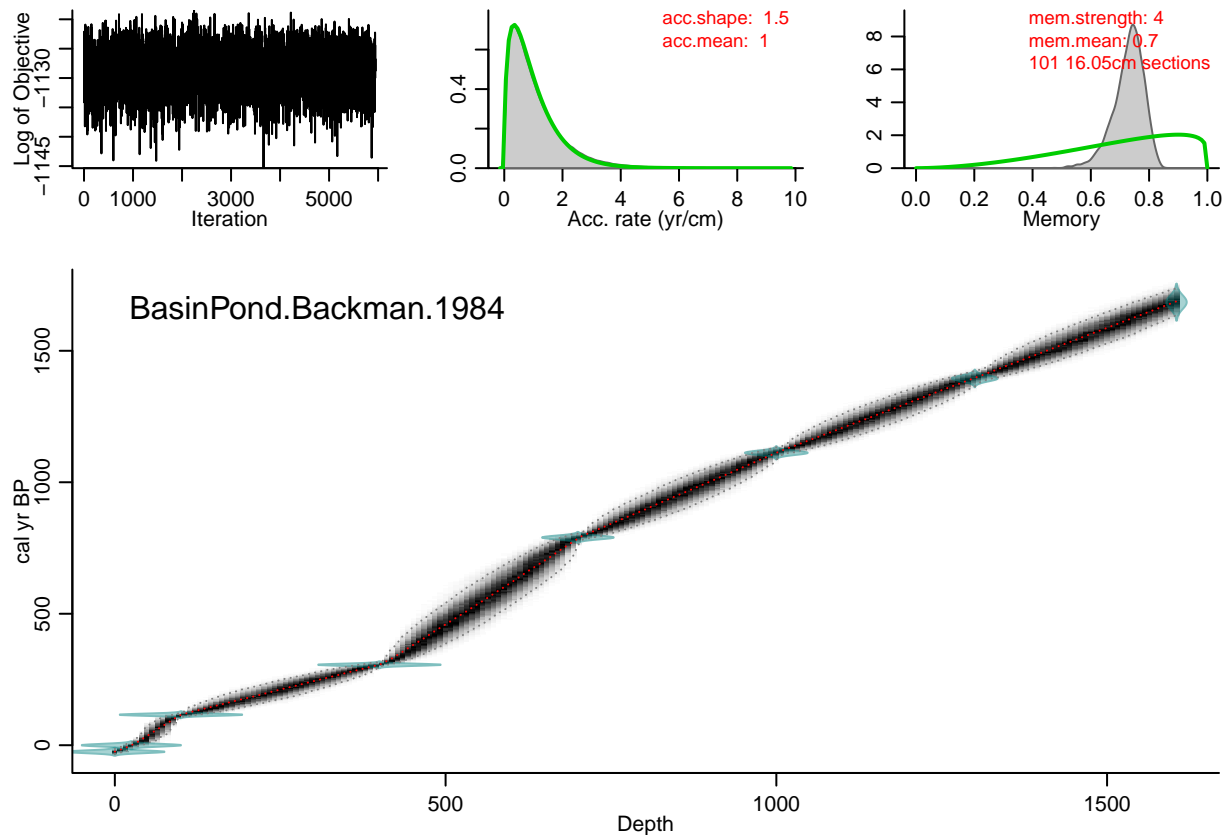
## The API call was successful, you have returned 1 record.
## API call was successful. Returned record for Basin Pond
## The API call was successful, you have returned 6 records.
## API call was successful. Returned chronology.
## API call was successful. Returned chronology.
## API call was successful. Returned chronology.

#estimate uncertainty from range
L$chronData[[2]]$measurementTable[[1]] = estimateUncertaintyFromRange(L$chronData[[2]]$measurementTable
detach("package:geojson",character.only = TRUE) #deal with polygon conflict!
```

16. Now Isabel uses the geochronologic data from LinkedEarth to create a new age model in Bacon.

```
L = runBacon(L,which.chron = 2,baconDir = "~/Dropbox/MacBacon/",modelNum = 2,
            remove.rejected = FALSE,labIDVar="chron.control.id", age14CVar = NULL, age14CuncertaintyVar=

## [1] "Looking for laboratory ID..."
## [1] "Found it! Moving on..."
## [1] "Looking for radiocarbon ages..."
## [1] "Looking for radiocarbon age uncertainty..."
## [1] "Looking for calibrated ages..."
## [1] "Found it! Moving on..."
## [1] "Looking for calibrated age uncertainty..."
## [1] "Found it! Moving on..."
## [1] "Looking for depth..."
## [1] "Found it! Moving on..."
## [1] "Looking for radiocarbon reservoir age offsets (deltaR)..."
## [1] "can also use radiocarbon reservoir ages if need be..."
## [1] "Looking for radiocarbon reservoir age uncertainties..."
## [1] "Looking for column of reject ages, or ages not included in age model"
## [1] 8
##      id age error depth cc dR dSTD ta tb
## 1 330  -25     5     0 0 0     0 33 34
## 2 331   0     5    25 0 0     0 33 34
## 3 332 116     4   100 0 0     0 33 34
## 4 333 306     4   400 0 0     0 33 34
## 5 334 790     7   700 0 0     0 33 34
## 6 335 1112    8  1000 0 0     0 33 34
## 7 336 1395   11  1300 0 0     0 33 34
## 8 337 1685   26  1605 0 0     0 33 34
## Hi there, welcome to Bacon for Bayesian age-depth modelling
## Using calibration curve specified within the .csv file,
```

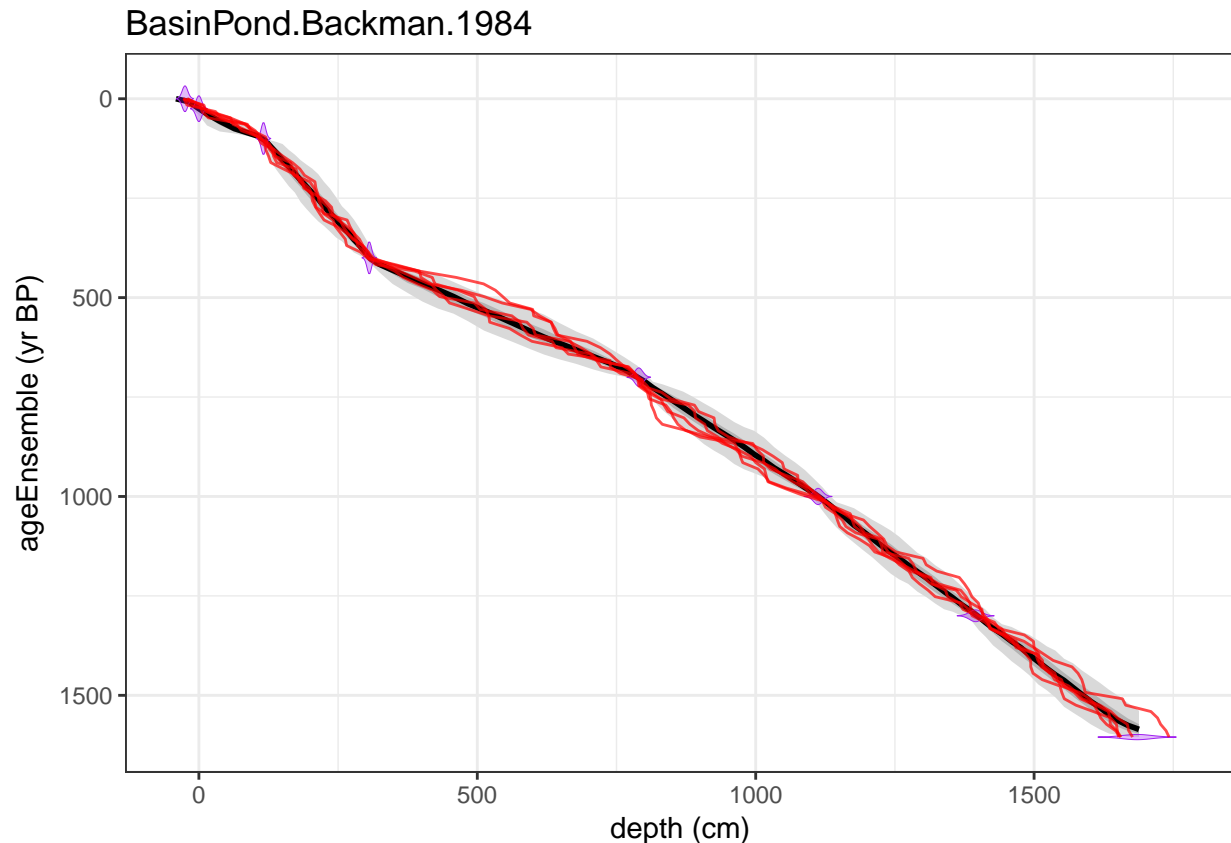


```
## Warning, this will take quite some time to calculate. I suggest increasing d.by to, e.g. 10
## Mean 95% confidence ranges 92.6 yr, min. 16.8 yr at 101 cm, max. 178.8 yr at 562 cm
##
## [1] "taking a short break..."
```

17. And geoChronR to visualize her new age model and a handful of ensemble members

```
plotChron(L,chron.number = 2,model.num = 2)
```

```
## [1] "Found it! Moving on..."
## [1] "Found it! Moving on..."
```



18. Next, she takes the temperature data from the LinkedEarth file, and ports it into the Neotoma-derived object, and maps the age ensemble onto the depths associated with the temperature values.

```
L$paleoData[[1]]$measurementTable[[1]]$temperature <- L2$paleoData[[1]]$measurementTable[[1]]$temperature
L <- mapAgeEnsembleToPaleoData(L = L,which.chron = 2,which.paleo = 1,which.model = 2,which.ens = 1,which)

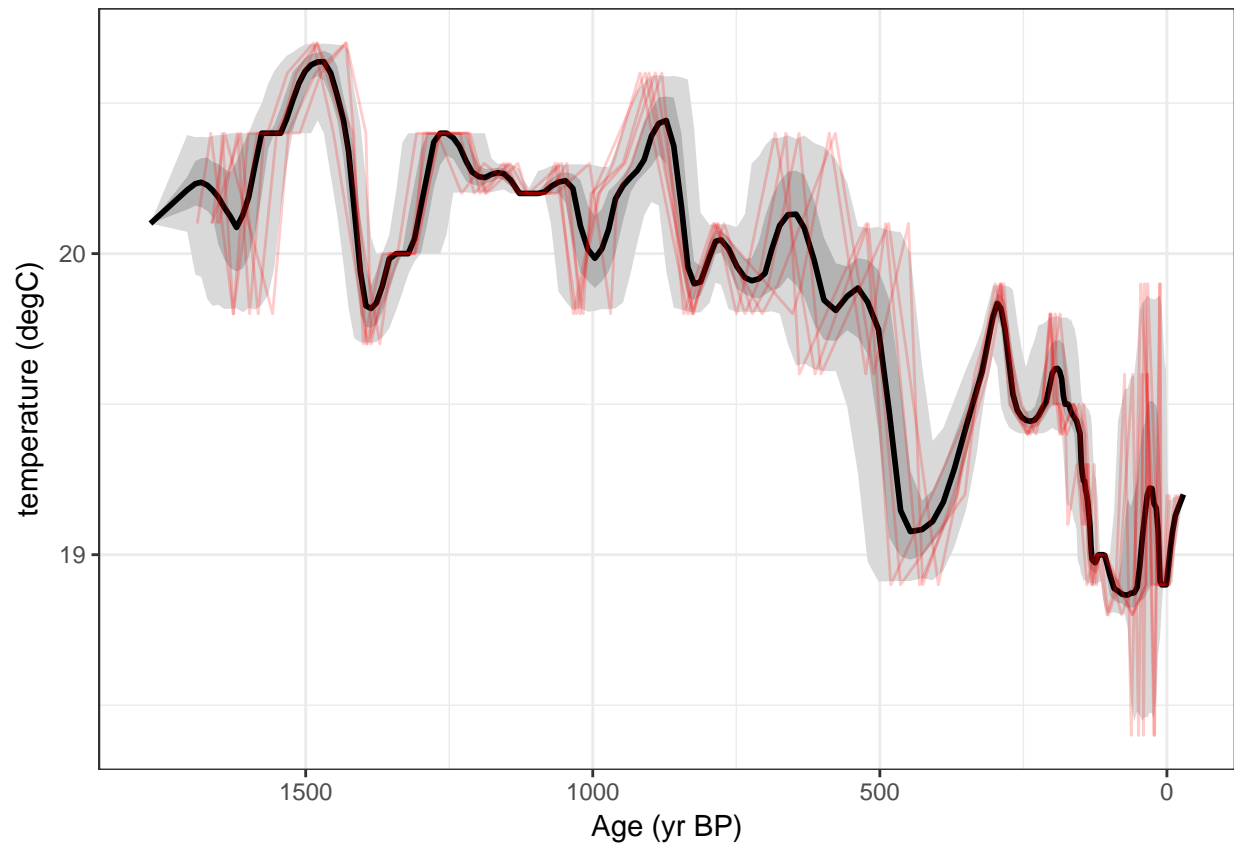
## [1] "BasinPond.Backman.1984"
## [1] "Looking for age ensemble...."
## [1] "Found it! Moving on..."
## [1] "Found it! Moving on..."
## [1] "getting depth from the paleodata table..."
## [1] "Found it! Moving on..."
```

19. Finally, Isabel creates a plot that shows the impact of age uncertainty on the temperature dataset.

```
library(ggplot2)
temp <- selectData(L,"temperature")

## [1] "Found it! Moving on..."
ageEnsemble <- selectData(L,"ageEnsemble")

## [1] "Found it! Moving on..."
plot <- plotTimeseriesEnsRibbons(ageEnsemble,temp)
plot <- plotTimeseriesEnsLines(ageEnsemble,temp,color = "red",add.to.plot = plot,maxPlotN = 5)+
  scale_x_reverse("Age (yr BP)")
print(plot)
```



The end.