Marine exploitation and the arrival of farming: resolving the paradox of the Mesolithic-Neolithic transition in Denmark

Methods for data analysis and visualisation

Contents

Introduction

This document contains the R code used to generate the kernel density and moving average models discussed in the main paper. Some previously unpublished functions are documented in full (Part I), together with the

scrips used to plot the graphs used in the main paper (Part II). Part III contains some additional analyses. The agent-based model discussed in the paper is not included in this markdown, but is documented in the file `ABM.R`.

Part I: Source code

Loading calibration curves and the `rowcal` package

The following script requires the `rowcal` package, which contains code that have been published before by McLaughlin 2019 (`rowcal`, `MCmix`, `MCdensity`) and McLaughlin et al 2021 (`ggr`, `phasedensity`), and their associated S3 methods for analyzing and plotting the objects created by these functions. It is planned to release this package on CRAN in due course, but it can also be downloaded and installed from github at any time using the `devtools` package.

```
# Install rowcal if not present on system
if(!require(devtools)) install.packages('devtools')
if(!require(rowcal)) install_github('rowan-mclaughlin/rowcal')
library(rowcal)
```

Modelling response variables linked to radiocabron data

The various utilities in the `rowcal` package use some form of Monte Carlo resampling to deal with age uncertainty, such as the probability distributions of radiocarbon dates. The function `MCrunmed` visualises how a response variable such as an associated stable isotope measurement changes with time. The basic procedure is:

- for each date, sample an x-axis (i.e. calendar) value from the probability distribution using the inverse transform sampling method
- order the data according to these values
- compute a running median as a scatter-plot smoothing step using the standard R function `runmed` (specifying a k value but otherwise default settings)
- repeat this process a fixed number (default N=100) of times

The probability distribution here is technically a probability mass function, and may be a radiocarbon date, a the posterior probability of an event in a Bayesian model, or a uniform distribution. The function `MCr.as.MCd` is used to convert the output of `MCrunmed` into a regular time grid, where the first column represents time stamps and the subsequent columns contain an approximation of each resampling run. This enables plotting via the `plot.MCd` function. More information about how these functions can be found in the `rowcal` package documentation.

Integrating chronological information from OxCal

Chronological precision can be improved for undated samples associated with material whose age is known via an existing 'OxCal' age model. The following script uses the objects created by the R package `oxcAAR`(Hinz et al 2021), which are resampled and used to explore response variables. In our study, these functions are mainly used to examine trends in published organic residue data.

```
#Reformat OxCal posteriors as a two-column matrix of timestamps and prob.
ox.as.rowcal<-function(x, ox=OxCalModel){
  probs<-ox[[x]]$posterior$prob
  start<-ox[[x]]$posterior$start
  res<-  ox[[x]]$posterior$resolution
  norm<- ox[[x]]$posterior$probNorm
  probs<-probs*norm*res
  return(matrix(c(start+res*seq_along(probs),probs), ncol=2))
}
```

Part II: Data analysis for the main paper

Preparing calibration curves and loading source data

The first step in our analysis is to prepare a series of custom calibration curves to allow for the calibration of samples comprised of proteins derived from both terrestrial and marine biomes:

```
# calculate the mixed curves
mix10<-mixcurves(mix=0.1, delR=-234, error_delR=61)
mix20<-mixcurves(mix=0.2, delR=-234, error_delR=61)
mix30<-mixcurves(mix=0.3, delR=-234, error_delR=61)
mix40<-mixcurves(mix=0.4, delR=-234, error_delR=61)
mix50<-mixcurves(mix=0.5, delR=-234, error_delR=61)
mix60<-mixcurves(mix=0.6, delR=-234, error_delR=61)
mix70<-mixcurves(mix=0.7, delR=-234, error_delR=61)
mix80<-mixcurves(mix=0.8, delR=-234, error_delR=61)
mix90<-mixcurves(mix=0.9, delR=-234, error_delR=61)

# Local marine curve
marineD<-mixcurves(mix=1, delR=-234, error_delR=61)
```

The radiocarbon data is loaded into memory, and a unique identifier generated for each unique site.

```
library('readxl')
dkURL<-'DEMRE.xlsx'   # Change to another local location if desired

dk_raw<-read_xlsx(dkURL, sheet='C-14 dates from shell middens') |> as.data.frame()

# Subset for computable results
goodcurves<-c('intcal','marineD',paste0('mix',seq(10,90,10)))
dk_data<-dk_raw[which(dk_raw$Curve %in% goodcurves &
                        !is.na(dk_raw$`Radiocarbon age (BP)`)),]

# add sitecode based on unique site names
dk_data$sitecode<-as.numeric(factor(dk_data$`Site name`))

# Subset out archaeological (dk) data
dk<-dk_data[which(is.na(dk_data$Natural)),
            c("sitecode","Radiocarbon age (BP)","Radiocarbon error","Curve")]
```

Spatial density

Overal distribution maps

The following script uses two-dimensional kernel density analysis to visualize spatial distributions in the density of sites. The locations of midden sites from the sites and monuments register for Denmark is read in from an external file, as is a shapefile containing the Danish coastline buffered by 5km.

```
library(sf)
library(spatstat)
# Window for analysis
dkb<-sf::st_read("Denmark_UTM_buffered_5km.shp")
```

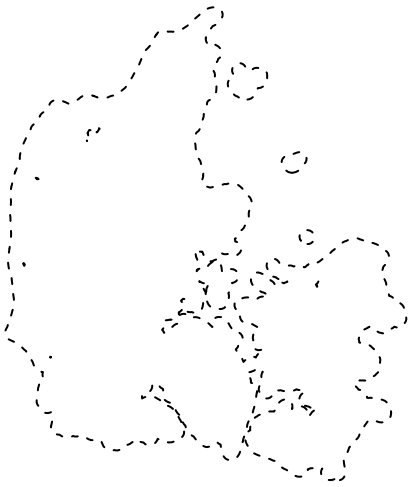
```
## Reading layer `Denmark_UTM_buffered_5km' from data source
##   `/Users/rowan/Dropbox/articles/Danish_shells/Data_paper/Denmark_UTM_buffered_5km.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 1 feature and 24 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 436534.3 ymin: 6044809 xmax: 744077.7 ymax: 6407293
## Projected CRS: WGS 84 / UTM zone 32N
```

```
dk_window<-as.owin(dkb)

# Coastline
dkc<-st_read("Denmark_UTM_outline.shp")
```

```
## Reading layer `Denmark_UTM_outline' from data source
##   `/Users/rowan/Dropbox/articles/Danish_shells/Data_paper/Denmark_UTM_outline.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 1 feature and 24 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 441524.9 ymin: 6049785 xmax: 892734 ymax: 6402308
## Projected CRS: WGS 84 / UTM zone 32N
```

```
# All SMR, all C14, and C14 sites in UTM format
smr<-st_as_sf(read.csv('dk_smr.csv'),
              coords=c("long","lat"), crs=4326) |> st_transform(32632)
c14<-st_as_sf(dk_data[which(!is.na(dk_data$Longitude)),],
              coords=c('Longitude','Latitude'), crs=4326) |> st_transform(32632)
sit<-st_as_sf(unique(dk_data[which(!is.na(dk_data$Longitude)),
                             c('Site name','Longitude','Latitude')]),
              coords=c('Longitude','Latitude'), crs=4326) |> st_transform(32632)

plot(dk_window, lty=2)
```

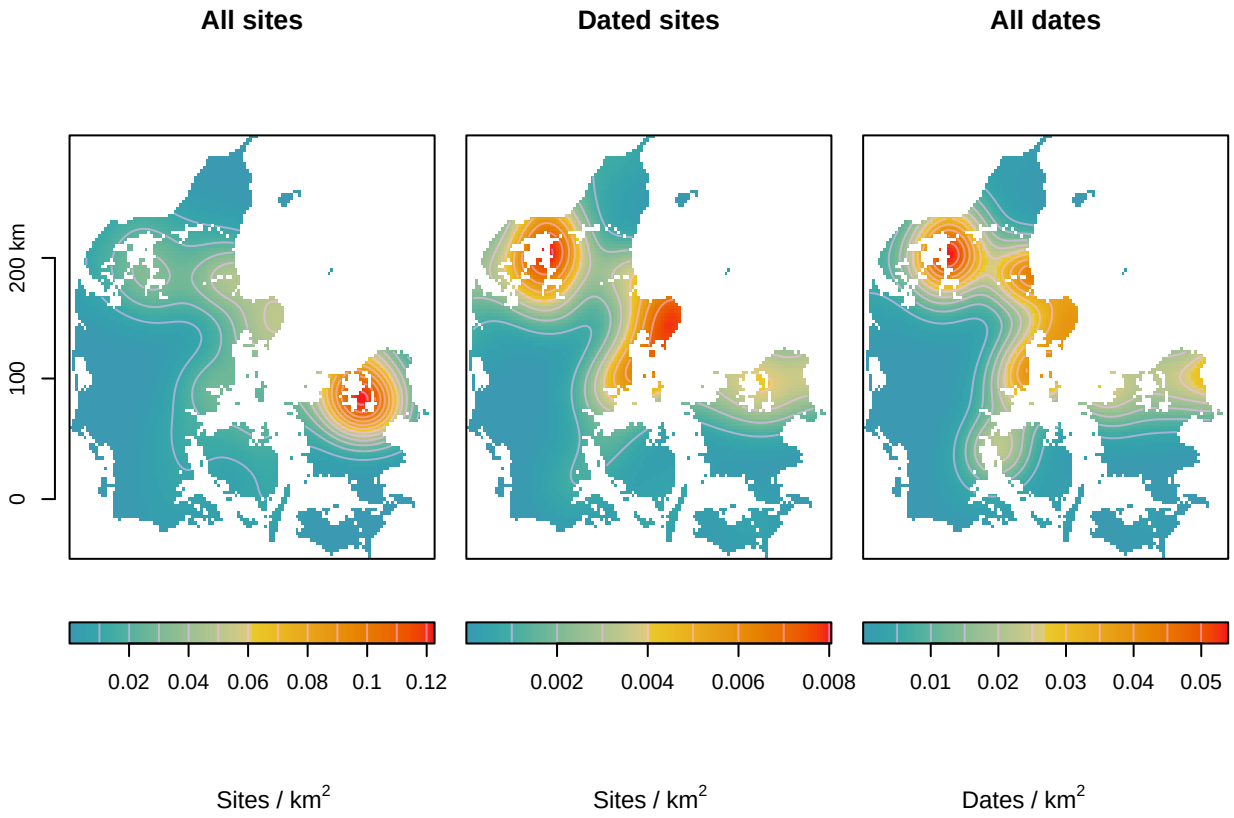
dk_window



```
smrD<-density(rescale(as.ppp(smr$geometry, W=dk_window),1000,'km'), sigma=20)
c14D<-density(rescale(as.ppp(c14$geometry, W=dk_window),1000,'km'), sigma=20)
sitD<-density(rescale(as.ppp(sit$geometry, W=dk_window),1000,'km'), sigma=20)

# Scale masking geometry to km instead of m
dkc$geometry<-dkc$geometry/1000
```

These are visualised as heatmaps together with contours.

```
#pdf('dk_2d_density.pdf', encoding='MacRoman', height=4)


par(mfrow=c(1,3),mar=c(4.1,0,2,0.1),omi=c(0.1,0.3,0,0))
plot(smrD,ribside='bottom',clipwin=as.owin(dkc),addcontour=TRUE,
     main='All sites',col=hcl.colors(128,'Zissou 1'),
     contourargs=list(col='#FFBBFF88', drawlabels=FALSE))
axis(2, at=c(0,100,200)+6100, lab=c(0,100,'200 km'))
plot(sitD,ribside='bottom',clipwin=as.owin(dkc),addcontour=TRUE,
     main='Dated sites',col=hcl.colors(128,'Zissou 1'),
     contourargs=list(col='#FFBBFF88', drawlabels=FALSE))
plot(c14D,ribside='bottom',clipwin=as.owin(dkc),addcontour=TRUE,
     main='All dates',col=hcl.colors(128,'Zissou 1'),
     contourargs=list(col='#FFBBFF88', drawlabels=FALSE))
L<-c(rep(expression("Sites / km"*""^{2}),2),expression("Dates / km"*""^{2}))
mtext(L,outer=TRUE, side=1, adj=c(0.15,0.5,0.85),cex=0.75, line=-2)
```




```
#dev.off()
```

Density within timeslices

Timeslicing radiocarbon data can be problematic because the probability margins of radiocarbon dates often make it ambiguous which 'slice' a sample belongs to. As a visualization stragegy, the following script applies the following algorithm to weight each site's contribution to the density estimate of each slice by the probability that a site dates to the timeframe under consideration:

- for each timeslice
  - a set of 'dates' are drawn from the radiocarbon data
  - these dates are filtered according to the slice in question
  - the 2d density is estimated with a Gaussian kernel (default here is 20km)
  - the process is repeated `Nboot` times and the results averaged

This is quite an expensive process so here `Nboot` is set at 25. Experiments were undertaken using a larger number of bootstraps but these made little difference to the output.

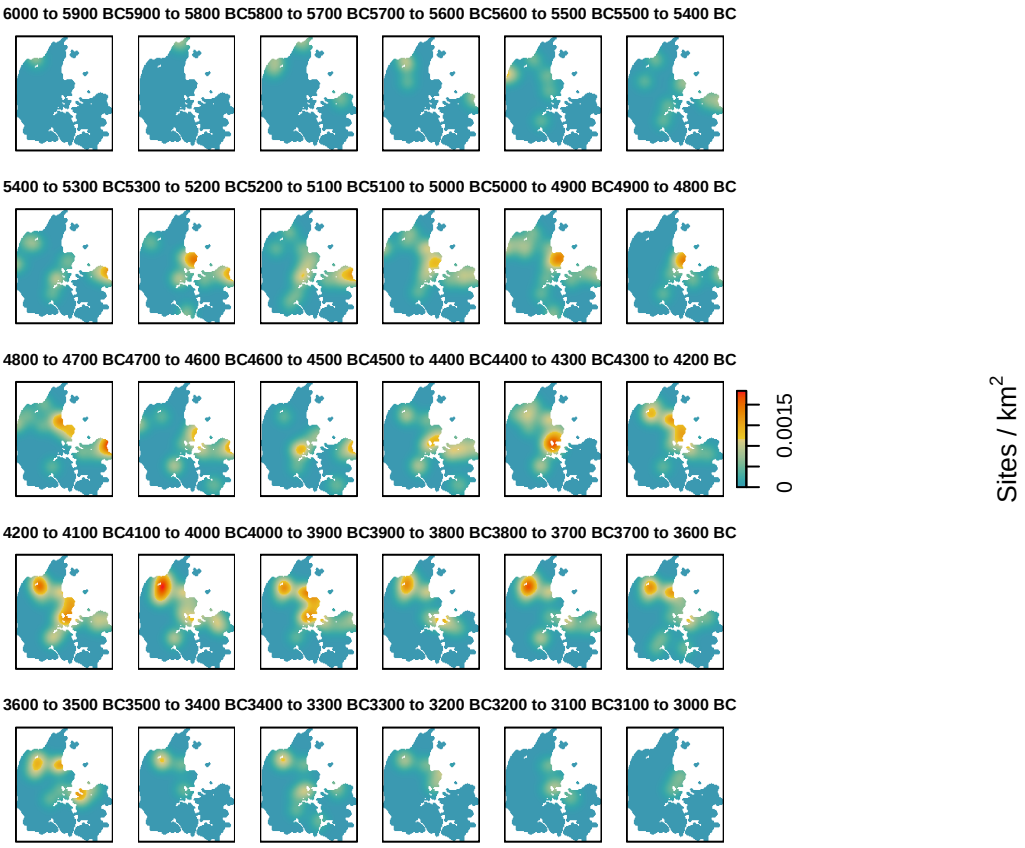
```
# Prepare data
dl<-dk_data[which(!is.na(dk_data$Longitude)),
            c('Radiocarbon age (BP)','Radiocarbon error','Curve',
              'Longitude','Latitude')]
colnames(dl)<-c('BP','SD','curve','x','y')

# Function for extracting a timeslice
```

```
sl<-function(f, t) dl[which(dl$sam>f & dl$sam<=t),c('x','y')]

# Function for calculating the density of each timeslice frame
aframe<-function(from,to,Nboot=25){
  for(i in 1:Nboot){
     dl$sam<-MCmix(dl[,1:3]) #sample from ages
     pts<-unique(sl(from, to)) #extract points
     # Coords converted from LatLong to UTM
     pts<-st_as_sf(pts, coords=c('x','y'), crs=4326) |> st_transform(32632)
     # calculate density, sigma=20km
     fr<-density(as.ppp(pts$geometry, W=dk_window), sigma=20000)
     if(i==1) out<-fr else out<-out+fr # add to average
  }
  return((out/Nboot)*1000000) # output probabilities per sq km
}
```

```
# The frames (slices) are calculated in a loop. Each frame is 100 years.
maplist<-list(); i=0; mids<-seq(-6050,-3050,100)
for(Y in mids){i<-i+1; maplist[[i]]<-aframe(Y-50,Y+50)}
names(maplist)<-paste(abs(mids-50),'to',abs(mids+50),'BC')
```

The `plot.imlist` function in `spatstat` can be used to visualise the results.

```
#pdf('timeslice_density.pdf')
  plot.imlist(maplist[2:31],equal.ribbon = TRUE,
              col=hcl.colors(128,'Zissou 1'),
              cex.main=0.8,mar.panel=c(0.5,0.5,0.5,0.5), main="")
  mtext(expression("Sites / km"*""^{2}), side=4, outer=TRUE, cex=0.8, line=-1)
```




```
#dev.off()
```

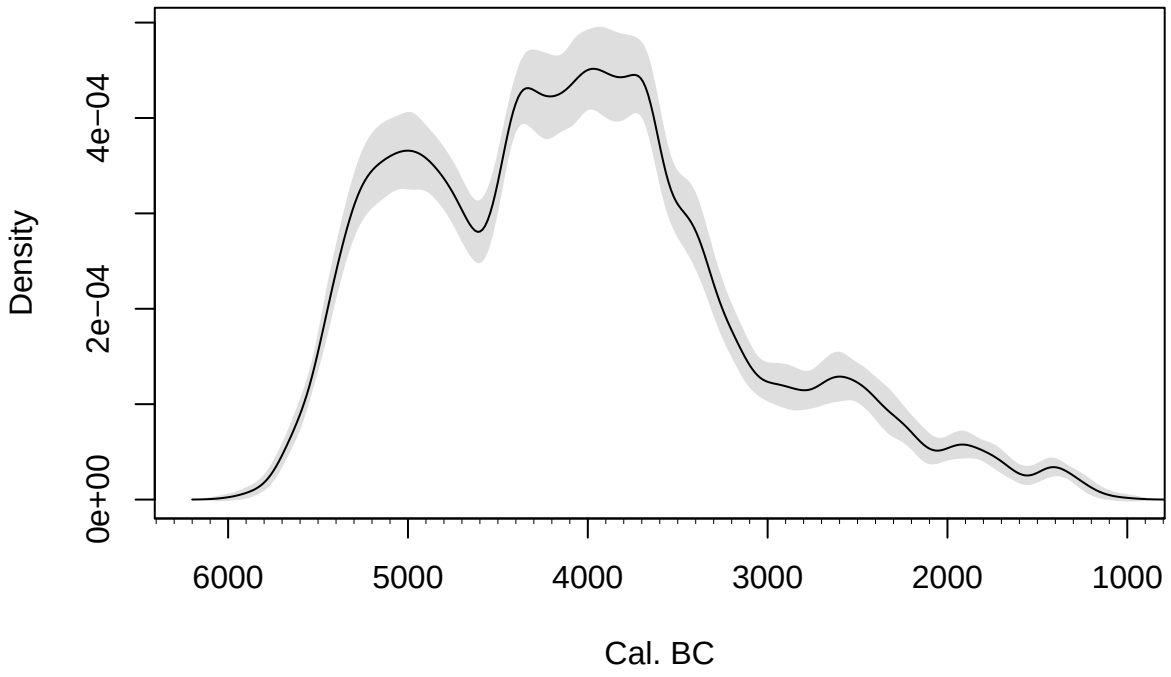
Temporal density

Shell middens

The next step is to calculate the temporal kernel density estimate (KDE) for the shell midden data. This is done using hierarchical clustering to identify unique generational phases (`h=30`) at each site. The bandwidth of the KDE is 75 years.

```
# Main overall density model of dated site phases
cals<-rowcal(dk[,2], dk[,3], dk[,4])
dk_kde<-phasedensity(siteids=dk[,1], dates=cals, bw=75)
dk_spd<-sum(cals)
```

```
# Plot the model
plot(dk_kde, xaxt='n', xlim=c(-6200,-1000))
# Add a ready-formatted x-axis with tickmarks
ax()
```



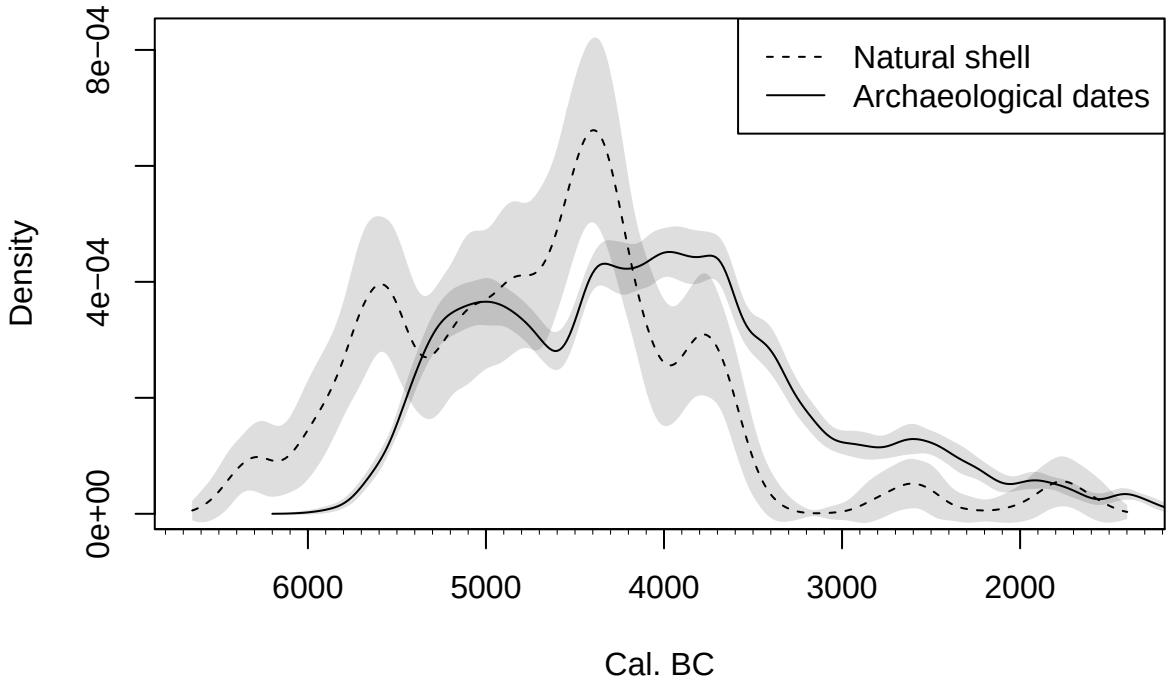
Natural shell banks

Using the same procedure, this is calculated for sites identified as natural banks using only samples of shellfish.

```
shellsp<-c('Ostrea edulis','Mytilus edulis','Cerastoderma edule',
           'Ostrea edulis/Littorina littorea',
           'Cerastoderma edule/Mytilus edulis',
           'Cerastoderma edule/Nassa reticulata',
           'Cerastoderma edule/Littorina littorea',
           'Ostrea edulis/Cerastoderma edule/Nassa reticulata',
           'Ruditapes decussatus/Ostrea edulis/Cerastoderma edule',
           'Ostrea edulis/Cerastoderma edule/Mytilus edulis/Littorina littorea')
cID<-c("sitecode","Radiocarbon age (BP)","Radiocarbon error","Curve")

dk_natural_shell<-dk_data[which(dk_data$Natural == 'Yes' &
                                  dk_data$Species %in% shellsp), cID]
```

```
natural_cals<-rowcal(dk_natural_shell[,2],dk_natural_shell[,3],
                     dk_natural_shell[,4])
dk_natural_shell_kde<-phasedensity(dk_natural_shell[,1], natural_cals,
                                   bw=75,N=250)
dk_natural_shell_spd<-sum(natural_cals)
```

```
# Plot the results juxtaposed with the density of archaeological dates
plot(dk_natural_shell_kde, xaxt='n', lty=2)
ax()
plot(dk_kde, add=T)
legend('topright',lty=c(2,1),legend=c('Natural shell','Archaeological dates'))
```



All marine resources

The following calculates a KDE for all directly-dated shellfish and other marine animals

```
marD<-MCdensity(dl=dk[which(dk$Curve=='marineD'),2:4],bw=75)
marD30<-MCdensity(dl=dk[which(dk$Curve=='marineD'),2:4],bw=30)
```

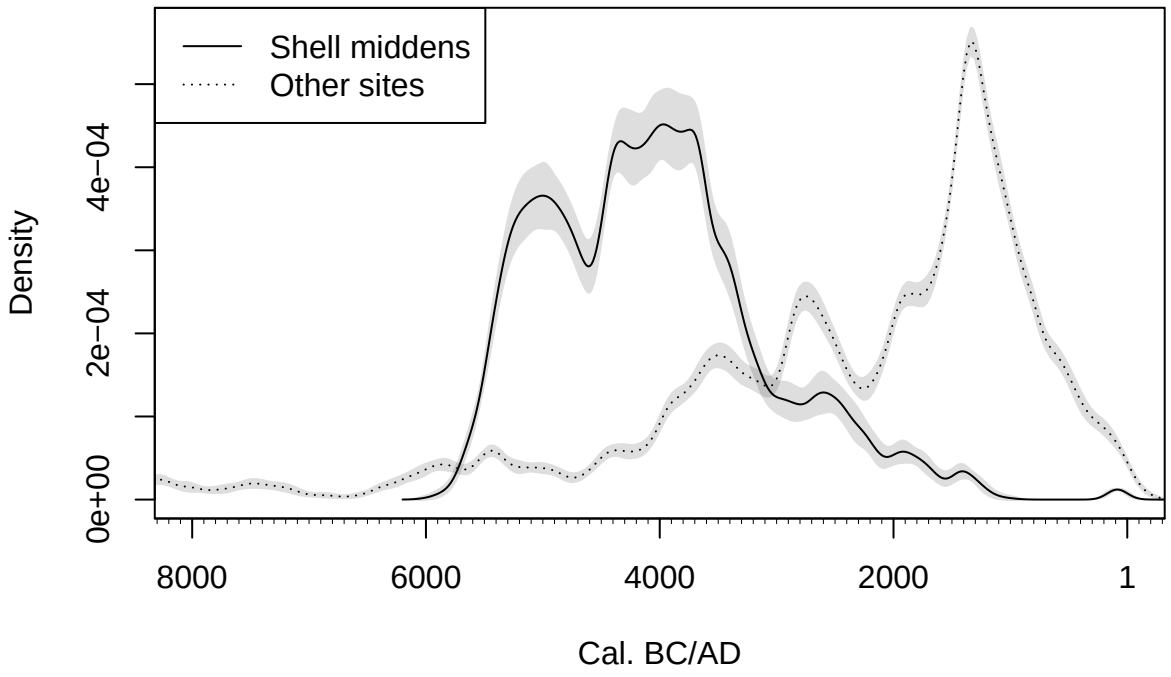
Other archaeological sites in Denmark

This mainly uses data from the p3k14c database, itself a compilation of other data sources (see Bird 2022 and references therein).

```
p3kd<-read_xlsx(dkURL, sheet='C-14 dates non-shell midden') |> as.data.frame()

# Select all dates not in the shell midden data
nonSM<-p3kd[!(p3kd$LabID %in% dk_data$`Lab no.`),]
```

```
# Non-shell midden density model:
nonSMD<-MCdensity(dl=nonSM[,3:4], bw=75)
```

```
# Plot the results juxtaposed with the density of archaeological dates
plot(nonSMD, xaxt='n', lty=3, xlim=c(-8000,0)) #8000 to 1 BC
ax()
plot(dk_kde, add=T)
legend('topleft',lty=c(1,3),legend=c('Shell middens','Other sites'))
```

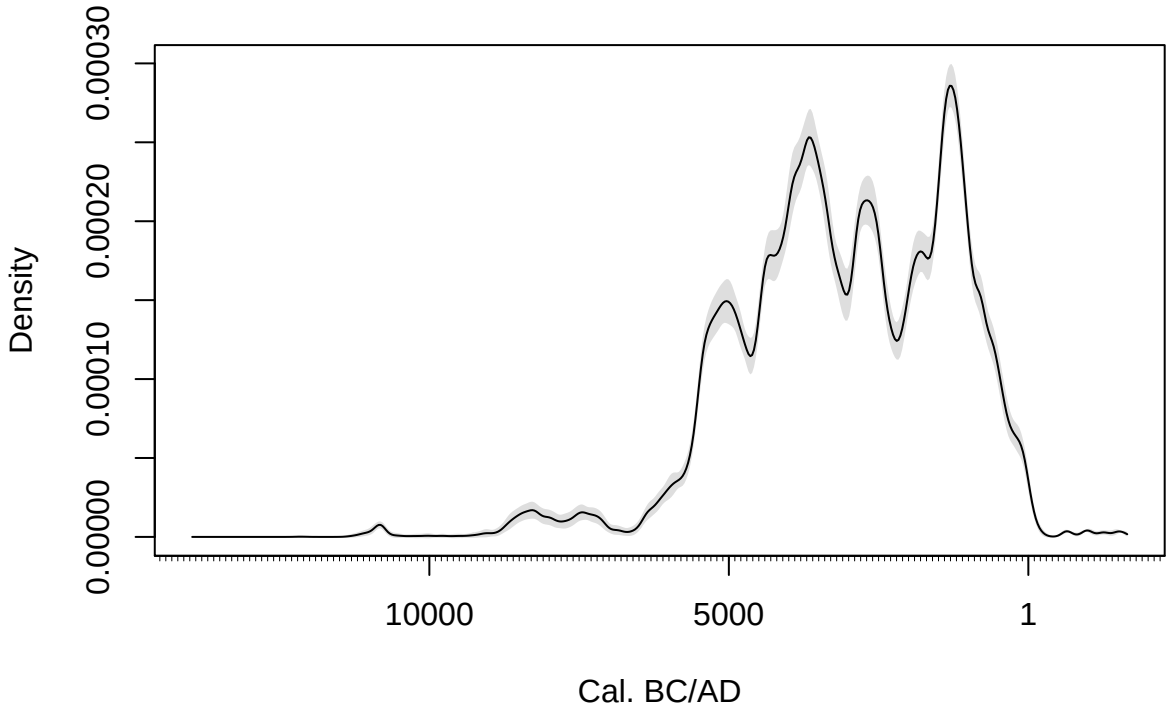


Density model of all sites, shell midden and otherwise

This is produced by concatenating the tables for shell midden and non-shell midden ratiocarbon dates, and producing a `phasedensity` model all the dates.

```
temp<-nonSM[,c(11,3,4)]; temp$Curve<-'intcal'
colnames(temp)<-colnames(dk)
ALL<-rbind(temp,dk)
ALLcal<-rowcal(ALL[,2],ALL[,3],ALL[,4])
ALLD<-phasedensity(ALL[,1],ALLcal, bw=75)
```

```
plot(ALLD, xaxt='n'); ax()
```



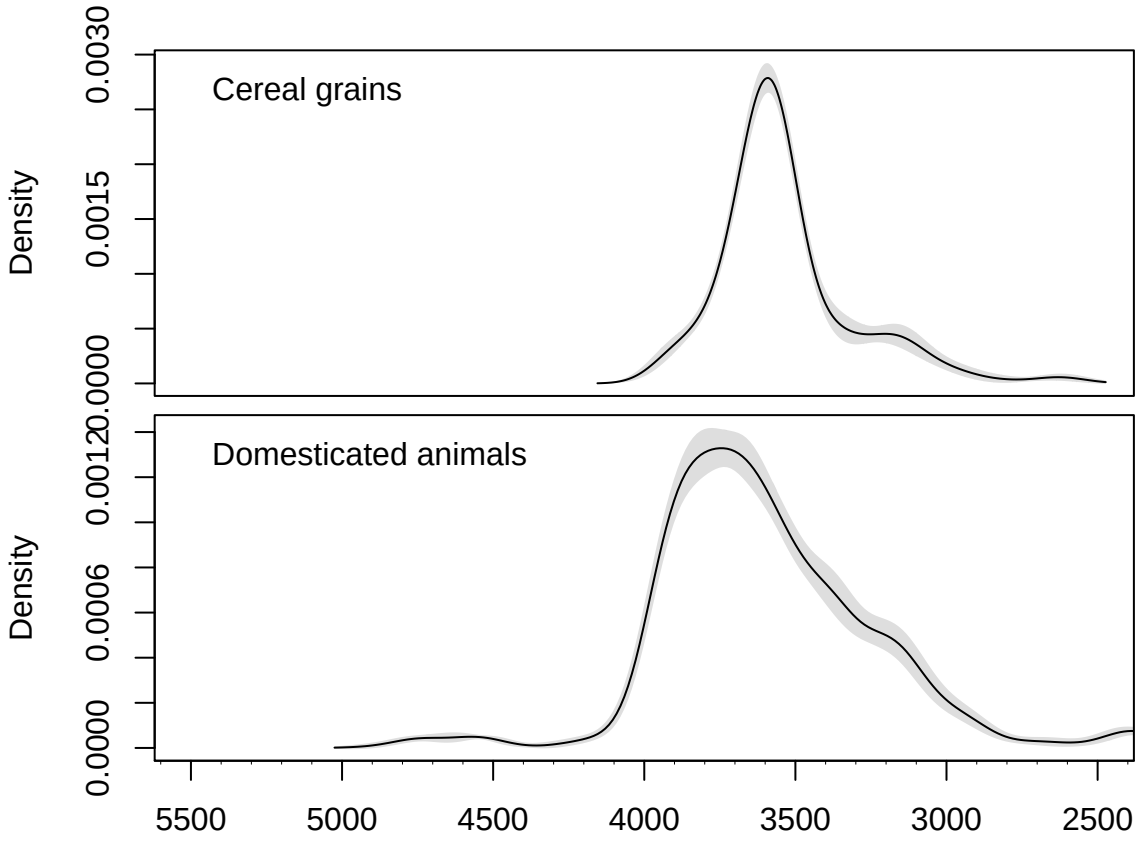
Domesticed animals and cereals

These are relatively straightforward to calculate as we are interested in the temporal frequency of samples rather than site phases, and all samples use the intcal20 curve.

```
# Cereals
cer<-read_xlsx(dkURL, sheet='C-14 dates from cereal grains') |> as.data.frame()
cer<-cer[,c('Site name', 'Radiocarbon age (BP)', 'Radiocarbon error',  'Curve')]
# Domestic animals
da<-read_xlsx(dkURL, sheet='C-14 dates from domesticates') |> as.data.frame()
da<-da[,c('Site name', 'Radiocarbon age (BP)', 'Radiocarbon error',  'Curve')]
```

```
# Calculate the  density models
cerD<-MCdensity(dl=cer[,2:3], bw=75)
daD<-MCdensity(dl=da[,2:3],bw=75)

# All domesticates density model
cn<-c('Site name','Radiocarbon age (BP)','Radiocarbon error')
alldom<-rbind(cer[,cn],da[,cn])
alldomD<-MCdensity(dl=alldom[,2:3],bw=75)
```

```
# Plot the results
par(mfrow=c(2,1), mar=c(0.25,5,0.25,2),omi=c(0.5,0,0.2,0))
plot(cerD, xlim=c(-5500,-2500),xaxt='n')
legend('topleft',pch=NA,legend='Cereal grains',bty='n')
plot(daD, xlim=c(-5500,-2500),xaxt='n')
ax(); legend('topleft',pch=NA,legend='Domesticated animals',bty='n')
```



Point pattern density analysis

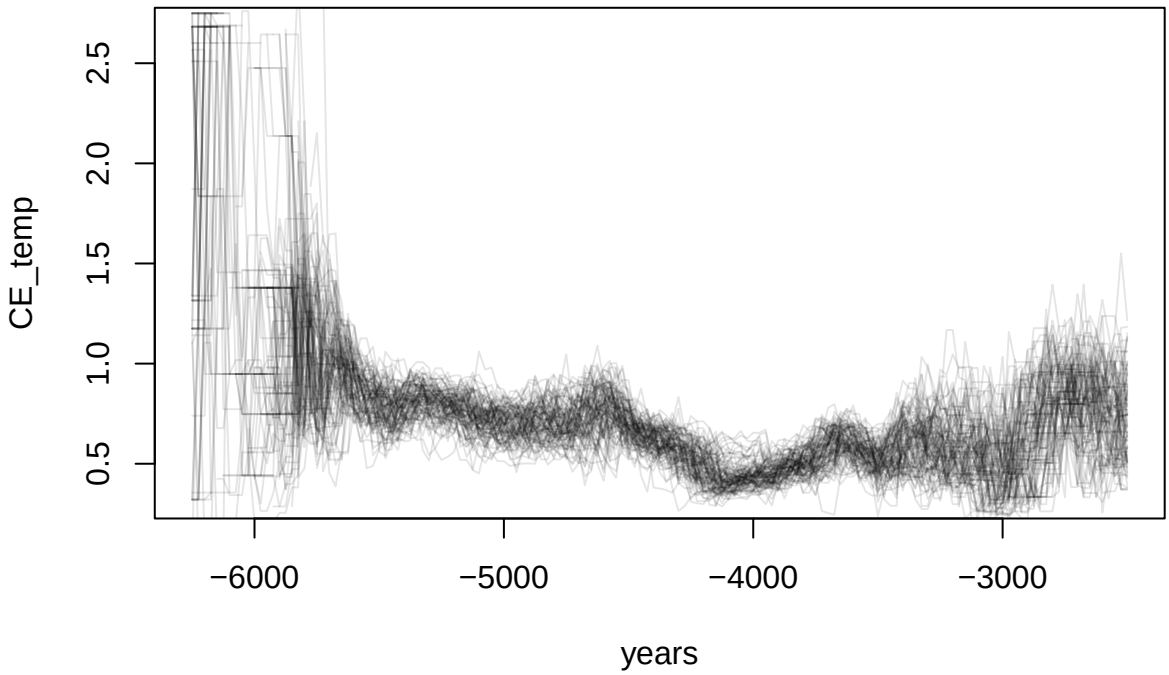
The point-pattern analysis is done within a spatial window calculated from the current danish coastline buffered by 5km. This needs to be stored in the working directory. In the following custom code, the site locations are converted to UTM co-coordinates, and the time series developed using direct calls to the `MCsam` sampling function (see `rowcal` package documentation for more information.)

```
# Clark-Evans R

library(spatstat)
library(sf)
# Need a window for analysis; i.e. the danish coastline buffered by 5km:
dkb<-st_read("~/Dropbox/GISMisc/Denmark/Denmark_UTM_buffered_5km.shp")
```

```
## Reading layer `Denmark_UTM_buffered_5km' from data source
##   `/Users/rowan/Dropbox/GISMisc/Denmark/Denmark_UTM_buffered_5km.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 1 feature and 24 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 436534.3 ymin: 6044809 xmax: 744077.7 ymax: 6407293
## Projected CRS: WGS 84 / UTM zone 32N
```

```
dk_window<-as.owin(dkb)
```

```r
# use sf to convert lat-long to UTM and store results

dk_data_clean<-dk_data[which(!is.na(dk_data$Longitude)),]
c14<-st_as_sf(dk_data_clean,
              coords=c('Longitude','Latitude'), crs=4326) |> st_transform(32632)

dk_data_clean$UTM_E<-st_coordinates(c14)[,1]
dk_data_clean$UTM_N<-st_coordinates(c14)[,2]

# Function to make suitable spatial point pattern 'ppp' object from a datelist,
#      selecting sites in a temporal window defined by `from` and `to`
temporal_ppp <- function(from, to, dl=dk_data_clean, W=dk_window,
                         sp_id=c('UTM_E','UTM_N'),t_id='median')
  as.ppp(unique(dl[which(dl[,t_id] >= from & dl[,t_id] < to),sp_id]), W=W)

# Set up the variables to run the Clark-Evans test through the timeseries
CE_results <- c()
counter <- 1
from <- -6250  # start date
to <- -2500    # end date
by <- 25       # step increment (in years)
size <- 250    # size of moving window (in years)
years <- seq(from=from,to=to,by=by)

# Calibrate dates
ppcals<-rowcal(dk_data_clean[,'Radiocarbon age (BP)'],
               dk_data_clean[,'Radiocarbon error'],
               dk_data_clean[,'Curve'])

allres<-data.frame(Y=years)
for(N in 1:100){
  CE_temp<-c()
  dk_data_clean$MC<-rowcal::MCsam(ppcals)
  for(Y in years)
    CE_temp<-c(CE_temp,clarkevans(temporal_ppp(Y-size/2, Y+size/2,t_id='MC'),
                                  correction='guard',clipregion = dk_window))
  if(N==1) plot(years,CE_temp,col=rgb(0,0,0,0.1), type='l') else
    lines(years,CE_temp,col=rgb(0,0,0,0.1))
  allres<-cbind(allres,CE_temp)
}
```




```
# Mean distance and nearest neighbour routines
# These use spatstat's functions pairdist and nndist, stepped through the
#     data in a similar way to the above

allmeddist<-data.frame(Y=years)
allnndist<-data.frame(Y=years)

# For 100 runs:
for(N in 1:100){
 meddist<-c()
 nndist<-c()
 # Simulate a chronology based on radiocarbon
 dk_data_clean$MC<-MCsam(ppcals)
 # For each year increment in from:to :
 for(Y in years) {
  # Calculate the median distance between each pair of sites:
  meddist<-c(meddist,median(pairdist(temporal_ppp(Y-size/2,Y+size/2,t_id='MC'))))
  # Calculate the median distance to the nearest neighbour:
  nndist<-c(nndist,median(nndist(temporal_ppp(Y-size/2,Y+size/2,t_id='MC'))))
 }
 # Store results
 allmeddist<-cbind(allmeddist,meddist)
 allnndist<-cbind(allnndist,nndist)
}

# For plotting these results the following helper function can be used to
#   to convert to the same format as MCdensity etc
ar.as.MCd<-function(ar, mult=1){
  out<-matrix(NA, nrow=length(ar[[1]]), ncol=length(ar))
  out[,1]<-ar[[1]]
  for(i in 2:length(ar)) out[,i]<-ar[[i]]*mult
  class(out)<-'MCd'
```

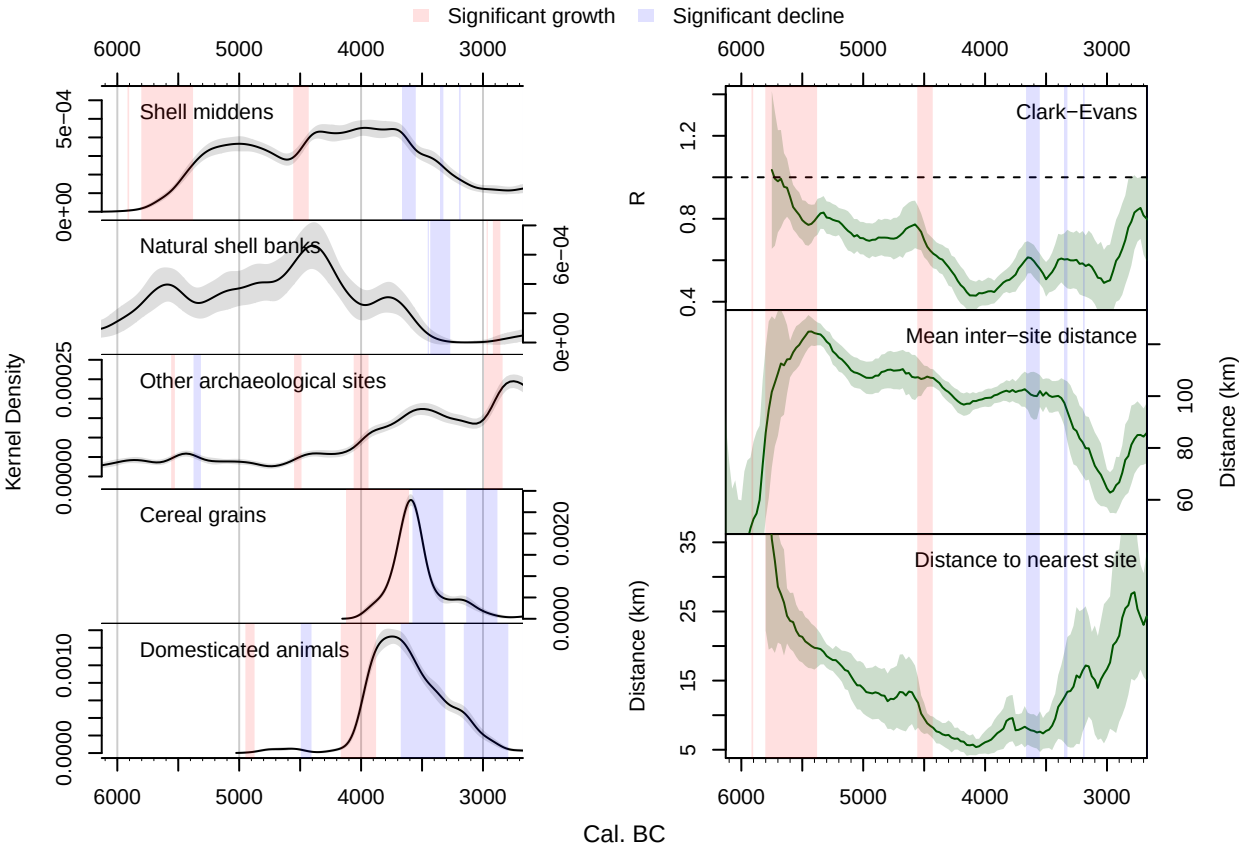
```
  return(out)
}
```

Plotting

The code below plots everything calculated above in one multi-panel plot

```
par(mar=c(0,4,0,4),omi=c(0.5,0,0.5,0),xpd=FALSE)
layout(matrix(c(sort(rep(1:5,3)),sort(rep(6:8,5))),nrow=15))
plot(dk_kde, axes=FALSE, xlim=c(-6000,-2800), ylim=c(-0.00002,0.00065),
     frame.plot=FALSE,ylab='')
  ax(side=3, ticksize = -0.02)
  axis(2); abline(v=seq(-6000,-3000,1000),col='#00000033')
  abline(h=par('usr')[3])
  legend('topleft',pch=NA,legend='Shell middens',bty='n')
  polygon.ggr_sig(ggrsignif(ggr(dk_kde)))
plot(dk_natural_shell_kde, axes=FALSE, xlim=c(-6000,-2800),
     frame.plot=FALSE,ylab='',ylim=c(-0.00005,0.0008))
  axis(4); abline(v=seq(-6000,-3000,1000),col='#00000033')
  abline(h=par('usr')[3])
  legend('topleft',pch=NA,legend='Natural shell banks',bty='n')
  polygon.ggr_sig(ggrsignif(ggr(dk_natural_shell_kde)))
plot(nonSMD,xlim=c(-6000,-2800), xaxt='n', ylim=c(-0.00002,0.0003),
     frame.plot=FALSE,ylab='Kernel Density')
  abline(h=par('usr')[3]); abline(v=seq(-6000,-3000,1000),col='#00000033')
  legend('topleft',pch=NA,legend='Other archaeological sites',bty='n')
  polygon.ggr_sig(ggrsignif(ggr(nonSMD)))
plot(cerD, xlim=c(-6000,-2800),xaxt='n',axes=FALSE, frame.plot=FALSE,ylab='')
  abline(h=par('usr')[3]); abline(v=seq(-6000,-3000,1000),col='#00000033')
  axis(4)
  legend('topleft',pch=NA,legend='Cereal grains',bty='n')
  polygon.ggr_sig(ggrsignif(ggr(cerD)))
plot(daD, xlim=c(-6000,-2800),xaxt='n',frame.plot=FALSE,ylab='')
  ax(ticksize = -0.02); abline(v=seq(-6000,-3000,1000),col='#00000033')
  legend('topleft',pch=NA,legend='Domesticated animals',bty='n')
  polygon.ggr_sig(ggrsignif(ggr(daD)))
plot(ar.as.MCd(allres),xlim=c(-6000,-2800),xaxt='n',col='#005500',
     fill='#00550033',ylab='R',ylim=c(0.4,1.4))
  abline(h=1,lty=2);ax(side=3)
  legend('topright',col=c('#005500'),pch=NA,'Clark-Evans', bty='n')
  polygon.ggr_sig(ggrsignif(ggr(dk_kde)))
plot(ar.as.MCd(allmeddist, mult=0.001),xlim=c(-6000,-2800),xaxt='n',
     ylim=c(50,130),col='#005500',fill='#00550033',ylab='',yaxt='n')
  axis(4)
  legend('topright',col=c('#005500'),pch=NA,'Mean inter-site distance', bty='n')
  polygon.ggr_sig(ggrsignif(ggr(dk_kde)))
plot(ar.as.MCd(allnndist, mult=0.001),xlim=c(-6000,-2800),xaxt='n',
     col='#005500',fill='#00550033',ylim=c(5,35),ylab='Distance (km)' )
  legend('topright',col=c('#005500'),pch=NA,'Distance to nearest site', bty='n')
  ax(ticksize = -0.02)
  polygon.ggr_sig(ggrsignif(ggr(dk_kde)))
mtext(side=1, outer=TRUE, line=2.5, 'Cal. BC', cex=0.75)
mtext(side=4, outer=TRUE, line=-1.33, 'Distance (km)', cex=0.66)
```

```
par(fig=c(0, 1, 0, 1), oma=c(0, 0, 0, 0), mar=c(0, 0, 0, 0), new=TRUE)
plot(0, 0, type='n', bty='n', xaxt='n', yaxt='n')
legend('top', ncol=2, fill=c('#FF000020','#0000FF20'),border=NA,
       legend=c('Significant growth','Significant decline'),bty='n')
```



Animals and their human exploitation

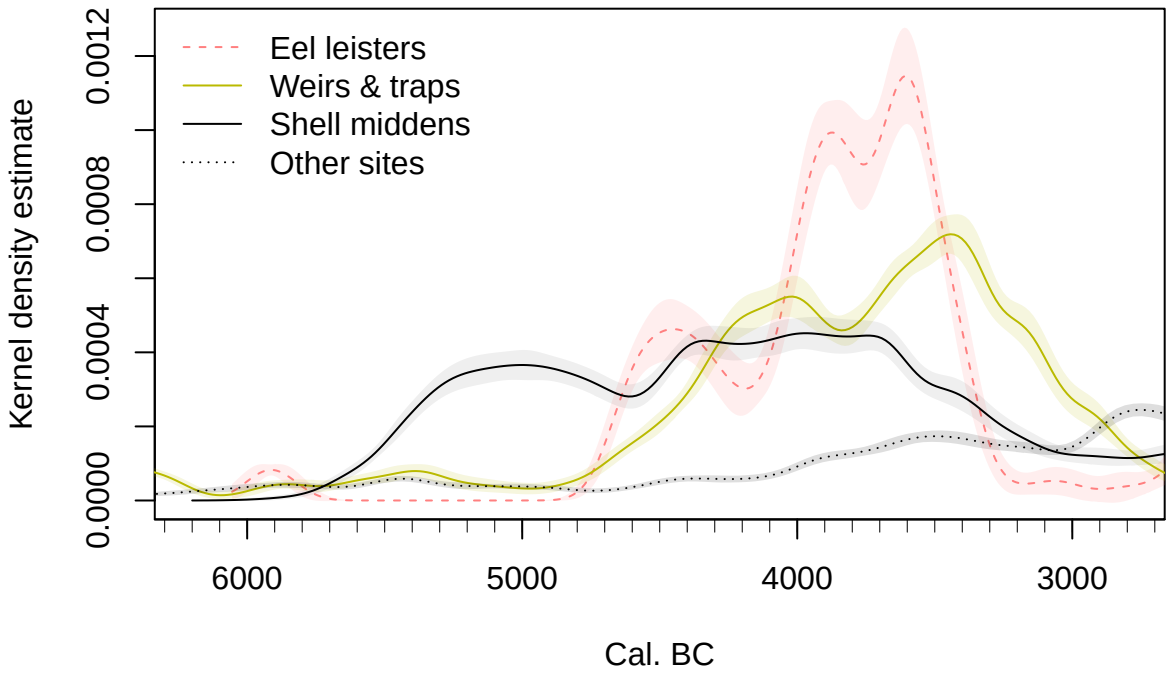
Artefacts and archaeological sites related to fishing

A number of these archaeological finds have been directly radiocarbon dated, so it is possible to compute a KDE to show their temporal distribution. For the traps and weirs, cluster analysis is used to correct for cases where multiple dates are obtained from the same site.

```
fing<-read_xlsx(dkURL,
         sheet='C-14 dates fish traps & weirs') |> as.data.frame()
leis<-read_xlsx(dkURL,
         sheet='C-14 dates bone points etc') |> as.data.frame()
```

```
fingD<-phasedensity(dl=fing[,c("Site name","Radiocarbon age (BP)",
                               "Radiocarbon error")],bw=75)
leisD<-MCdensity(dl=leis[,c("Radiocarbon age (BP)","Radiocarbon error")],bw=75)
```

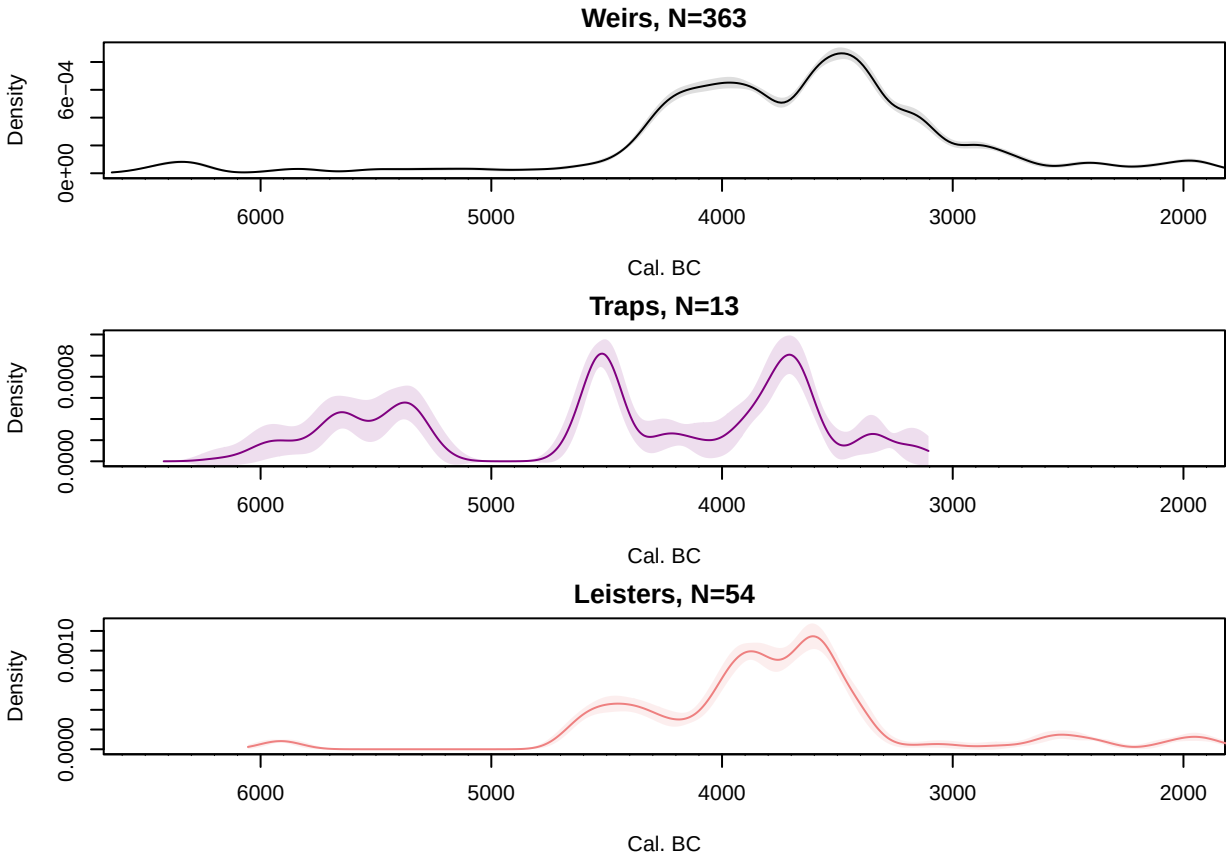
```
# Plot results
plot(leisD, xlim=c(-6200,-2800), lty=2, col='#FF8080', fill='#FF808022',
     xaxt='n',ylab='Kernel density estimate')
  plot(fingD, add=TRUE, col='#baba00', fill='#baba0022')
  plot(dk_kde, add=T, fill='#00000010')
  ax(ticksize = -0.025)
  legend('topleft',lty=c(2,1,1,3),col=c('#FF8080','#baba00',1,1),bty='n',
         legend=c('Eel leisters','Weirs & traps','Shell middens','Other sites'))
  plot(nonSMD, add=T, lty=3)
```



Seperate analysis of fish traps and weirs

The above analysis counts traps and weirs together, but it is equally possible to analyse them seperately:

```
traps<-fing[which(fing$Feature=='Fish trap'),]
weirs<-fing[which(fing$Feature=='Fish weir'),]
trapsD<-MCdensity(dl=traps[,c("Radiocarbon age (BP)","Radiocarbon error")],bw=75)
weirsD<-MCdensity(dl=weirs[,c("Radiocarbon age (BP)","Radiocarbon error")],bw=75)

par(mar=c(4,4,2,1),mfrow=c(3,1),omi=c(0,0,0,0))
plot(weirsD, xlim=c(-6500,-2000),xaxt='n',main=paste0('Weirs, N=',nrow(weirs)))
ax()
plot(trapsD, xlim=c(-6500,-2000), xaxt='n',col='#800080',fill='#80008022',
     main=paste0('Traps, N=',nrow(traps)))
ax()
plot(leisD, xlim=c(-6500,-2000),xaxt='n', col='#EE8080', fill='#EE808022',
     main=paste0('Leisters, N=',nrow(leis)))
ax()
```

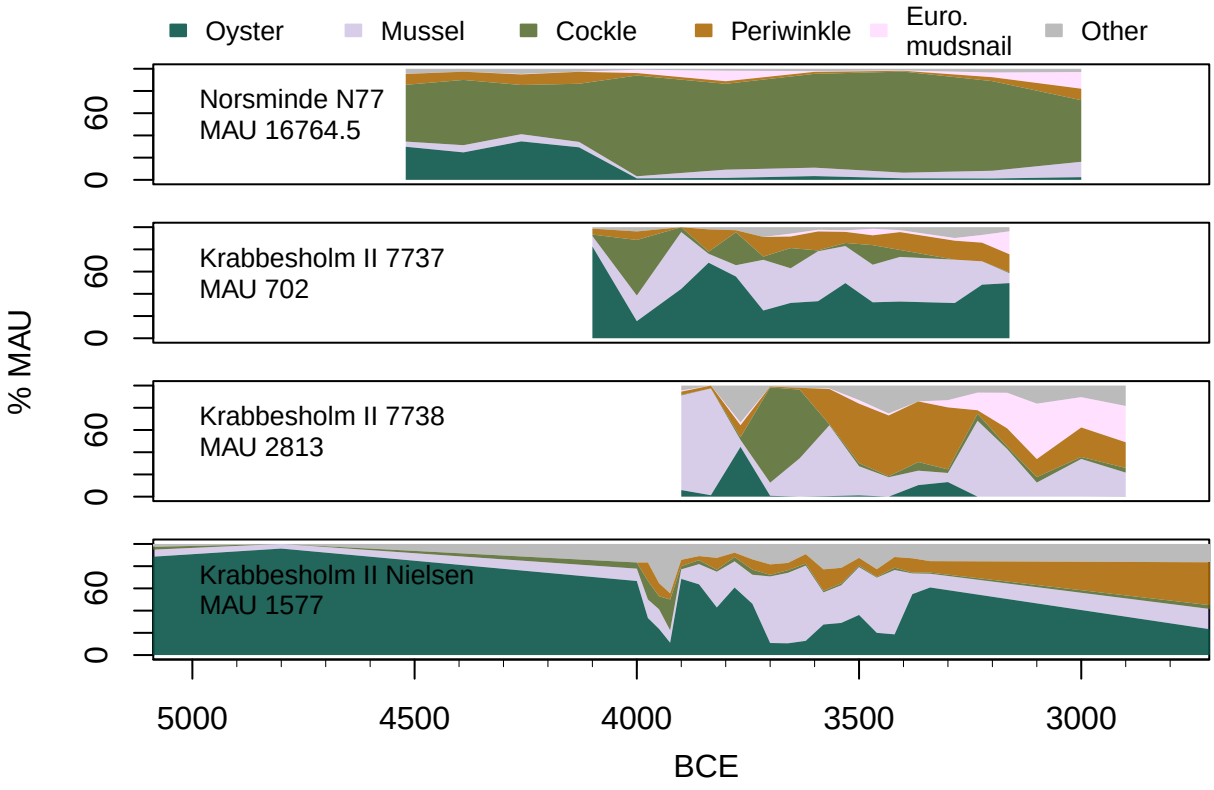


Mollusc shells: number of minium number of animal units (MAU)

Time-series data are available from four cores from two sites, spanning the very end of the Meoslithic period and much of the Neolithic. There is considerable variablity between layers and sites so the moving average models here are run with a higher number of resamples so the resulting estimate is reasonably smooth.

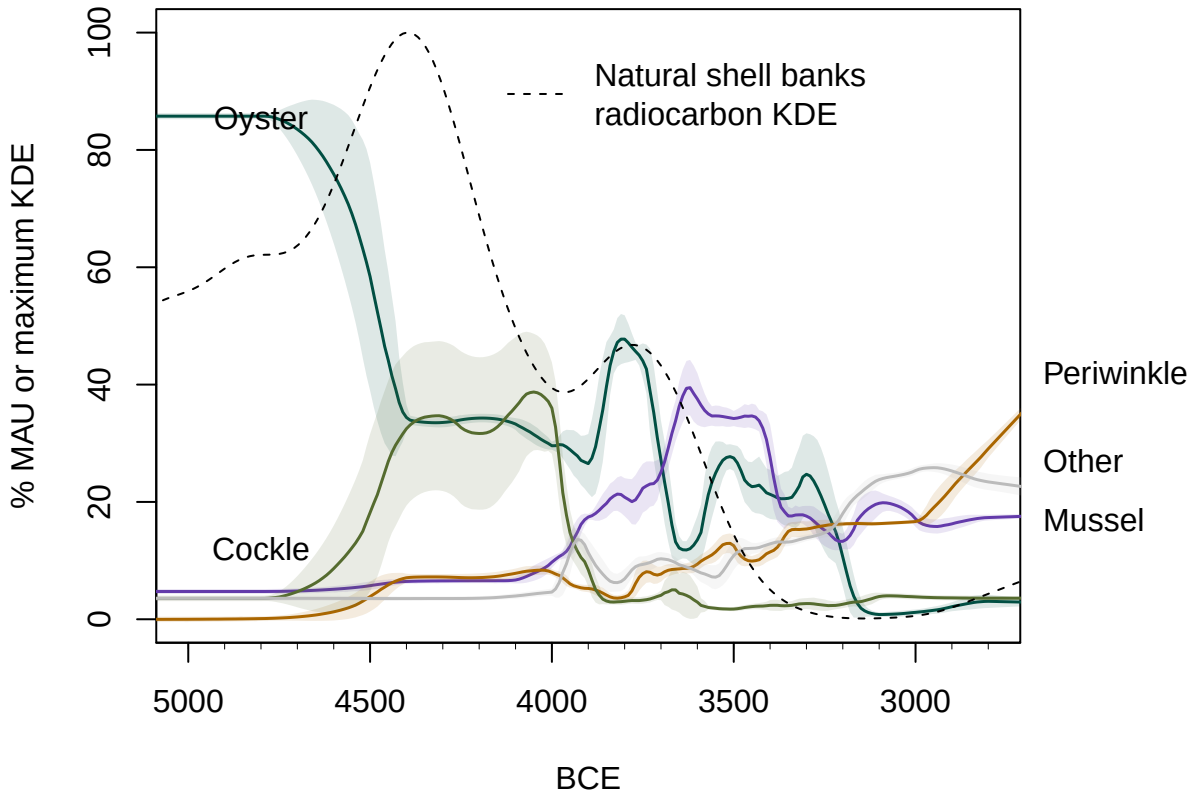
```
MAU<-read_xlsx(dkURL, sheet='Shellfish MAU values',na = '-') |> as.data.frame()
MAU<-MAU[-which(is.na(MAU$`Start (BC)`)),]
MAUothers<-rowSums(MAU[,c("Europ. Mudsnail","Other")])

# Function to plot the various MAU proportions per site
areaf<-function(x,y0,y1,...)
  graphics::polygon(c(x, rev(x)), c(y0, rev(y1)), border = NA,...)

# Colours for taxa
mcols<-c('#014F43DD','#633AAA40','#556B2FDD','#AA6600DD','#FF88FF40','#BBBBBB')
# Custom plotting function
mauplot<-function(corename, ...){
  core<-MAU[which(MAU$Core==corename),]
  plot(-core$`Start (BC)`,rep(100,nrow(core)),col=NA,
       type='l',ylim=c(0,100),xlab='', ylab='', ...)
  A <- core$Oyster; B <- A+core$Mussel; C <- B+core$Cockle
  D <- C+core$Periwinkle; E <- D+core$`Europ. Mudsnail`
  areaf(-core$`Est Start (BC)`, rep(100,nrow(core)), E, col=mcols[6]) #oth
  areaf(-core$`Est Start (BC)`,E, D, col=mcols[5]) #snail
```

```
  areaf(-core$`Est Start (BC)`,D, C, col=mcols[4]) #peri
  areaf(-core$`Est Start (BC)`,C, B,col=mcols[3]) #Coc
  areaf(-core$`Est Start (BC)`,B, A,col=mcols[2]) #muss
  areaf(-core$`Est Start (BC)`,A, rep(0,nrow(core)),col=mcols[1]) #oy
}

# Find unique core names
co<-unique(MAU$Core)

# Calculate total MAU for core
tMAU<-function(core) sum(MAU[which(MAU$Core==core),'Total MAU'])

# Plot each core with a legend
par(mfrow=c(length(co),1),mar=c(0.5,4,0.5,1),omi=c(0.7,0,0.5,0),cex=1)
for(cn in co) {
  mauplot(cn,xlim=c(-5000,-2800),xaxt='n')
  legend('topleft',pch=NA,legend=paste0(cn,'\nMAU ',tMAU(cn)),cex=0.8,bty='n') }
ax(ticksize = -0.06)
mtext('% MAU',side=2, outer=TRUE,line=-1)
mtext('BCE',side=1, outer=TRUE,line=1.75, adj=0.57)
par(fig=c(0,1,0.7,1), oma=c(0,0,0,0), mar=c(0,0,0,0), new=TRUE)
legend('top',fill=mcols, border=mcols, bty='n',
       legend=c('Oyster','Mussel','Cockle','Periwinkle','Euro.\nmudsnail',
                'Other'),ncol=6,cex=0.8)
```




```
# Calculate moving average MAU proportions over time
oyster_mau<-MCrunmed(x=rowunif(x0=-MAU$`Est Start (BC)`,x1=-MAU$`Est End (BC)`),
                     y=MAU$Oyster,k=9, N=250, y_blur=0.5, boot=FALSE) |> MCr.as.MCd()
```

```
## ============================================================================

mussel_mau<-MCrunmed(x=rowunif(x0=-MAU$`Est Start (BC)`,x1=-MAU$`Est End (BC)`),
                 y=MAU$Mussel,k=9, N=250, y_blur=0.5, boot=FALSE) |> MCr.as.MCd()

## ============================================================================

cockle_mau<-MCrunmed(x=rowunif(x0=-MAU$`Est Start (BC)`,x1=-MAU$`Est End (BC)`),
                 y=MAU$Cockle,k=9, N=250, y_blur=0.5, boot=FALSE) |> MCr.as.MCd()

## ============================================================================

other_mau<-MCrunmed(x=rowunif(x0=-MAU$`Est Start (BC)`, x1=-MAU$`Est End (BC)`),
                 y=MAUothers,k=9, N=250, y_blur=0.5, boot=FALSE) |> MCr.as.MCd()

## ============================================================================

pwink_mau<-MCrunmed(x=rowunif(x0=-MAU$`Est Start (BC)`,x1=-MAU$`Est End (BC)`),
             y=MAU$Periwinkle,k=9, N=250, y_blur=0.5, boot=FALSE) |> MCr.as.MCd()

## ============================================================================

# Plot the moving average models
par(mfrow=c(1,1),mar=c(4,4,2,6),omi=c(0,0,0,0))
plot(oyster_mau,xaxt='n', xlab='BCE', xlim=c(-5000,-2800),col='#014F43',
     fill='#014F4322',lwd=1.5, ylab='% MAU or maximum KDE', ylim=c(0,100))
ax()
plot(mussel_mau, add=T,col='#633AAA',fill='#633AAA22',lwd=1.5)
plot(pwink_mau, add=T,col='#AA6600',fill='#AA660022',lwd=1.5)
plot(cockle_mau, add=T,col='#556B2F',fill='#556B2F22',lwd=1.5)
plot(other_mau, add=T, col='#BBBBBB',fill='#BBBBBB22',lwd=1.5)
text(-4800,c(85,12),c('Oyster','Cockle'))
axis(4, at=c(42,27,17),lab=c('Periwinkle','Other','Mussel'),las=2,
     tick=FALSE,line=-0.4)
plot(dk_natural_shell_kde,add=T, fill=NA,smax=TRUE, scalefactor = 100,lty=2)
legend(-4200,100,legend='Natural shell banks\nradiocarbon KDE',lty=2,bty='n')
```




```
# Study moving average of total MAU
mMAU <- function(corename){
  core<-MAU[which(MAU$Core==corename),]
  runmed_MAU<-MCrunmed(x=rowunif(x0=-core$`Est Start (BC)`,
                                 x1=-core$`Est End (BC)`),
                       y=core$`Total MAU`,k=9, N=500, y_blur=0.5, boot=TRUE)
  return(MCr.as.MCd(runmed_MAU))
}

N77_MAU<-mMAU("Norsminde N77")
```

```
## ============================================================================
```

```
K37_MAU<-mMAU("Krabbesholm II 7737")
```

```
## ============================================================================
```

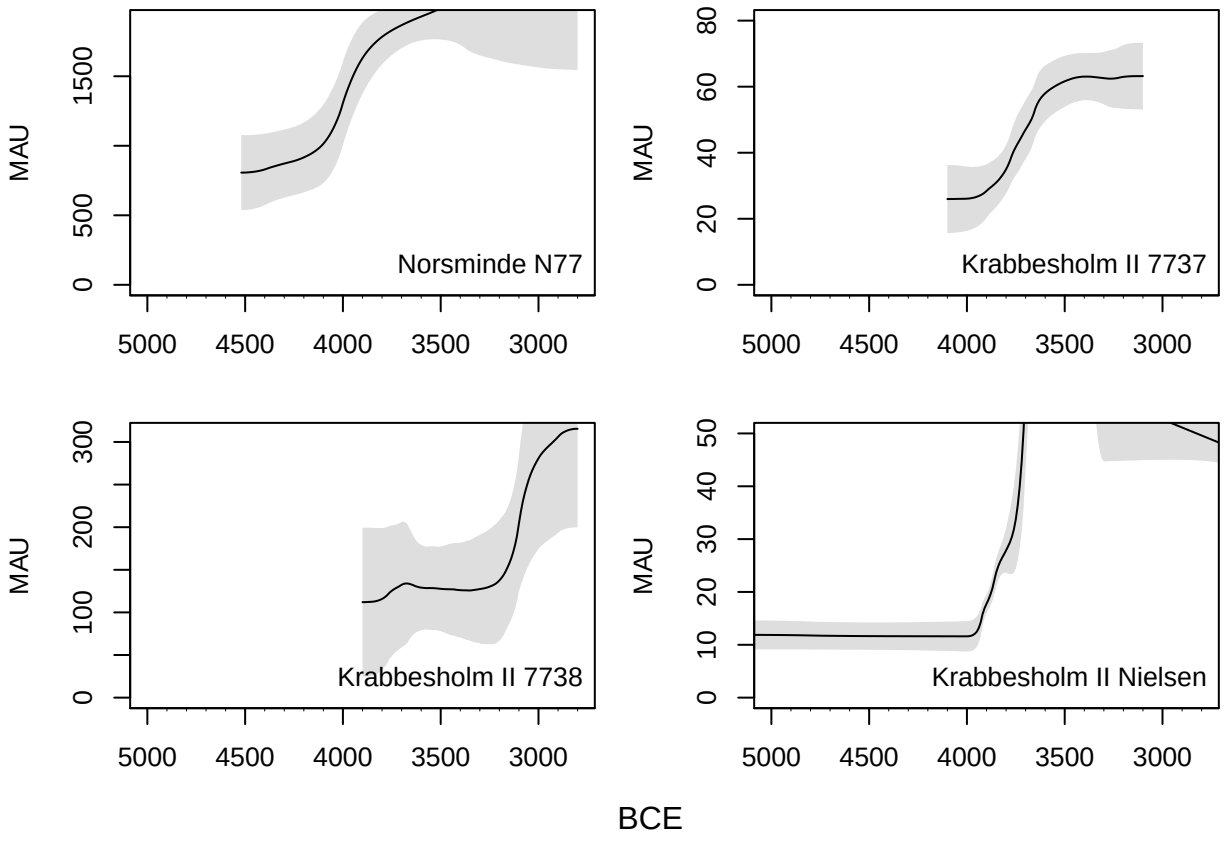
```
K38_MAU<-mMAU("Krabbesholm II 7738")
```

```
## ============================================================================
```

```
KNi_MAU<-mMAU("Krabbesholm II Nielsen")
```

```
## ============================================================================
```

```
par(mfrow=c(2,2), mar=c(3,4,1,1), omi=c(0.2,0,0,0))
plot(N77_MAU, xlim=c(-5000,-2800), xaxt='n',ylim=c(0,1900),ylab='MAU');ax()
legend('bottomright',pch=NA,bty='n',"Norsminde N77")
plot(K37_MAU, xlim=c(-5000,-2800), xaxt='n',ylim=c(0,80),ylab='MAU');ax()
legend('bottomright',pch=NA,bty='n',"Krabbesholm II 7737")
plot(K38_MAU, xlim=c(-5000,-2800), xaxt='n', ylim=c(0,310),ylab='MAU');ax()
legend('bottomright',pch=NA,bty='n',"Krabbesholm II 7738")
plot(KNi_MAU, xlim=c(-5000,-2800), xaxt='n', ylim=c(0,50),ylab='MAU');ax()
legend('bottomright',pch=NA,bty='n',"Krabbesholm II Nielsen")
mtext('BCE',side=1, outer=TRUE,line=0, adj=0.52)
```



Fish and marine mammal bones: number of identified specimens (NISP)

```
mNISP<-read_xlsx(dkURL, sheet='Marine mammal NISP',
                 na=c("","n.d.")) |> as.data.frame()
fNISP<-read_xlsx(dkURL, sheet='Fish NISP') |> as.data.frame()

# Trim any summary stats etc present in spreadsheet
fNISP<-fNISP[which(!is.na(fNISP$Entry)),]
mNISP<-mNISP[which(!is.na(mNISP$Entry)),]

nisp_fish<-MCrunmed(x=rowunif(x0=-fNISP$`Start (BC)`, x1=-fNISP$`End (BC)`),
                    y=as.numeric(unlist(fNISP$NISP)),k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

```
nisp_mm<-MCrunmed(x=rowunif(x0=-mNISP$`Start (BC)`, x1=-mNISP$`End (BC)`),
                  y=mNISP$`Total marine mammal NISP`,k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

```
nisp_om<-MCrunmed(x=rowunif(x0=-mNISP$`Start (BC)`, x1=-mNISP$`End (BC)`),
                  y=mNISP$`Total mammal NISP` -
                    mNISP$`Total marine mammal NISP`,k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

```
nisp_fish_sqm<-MCrunmed(x=rowunif(x0=-fNISP$`Start (BC)`, x1=-fNISP$`End (BC)`),
                    y=fNISP$`NISP/m2`,k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

```
nisp_mm_sqm<-MCrunmed(x=rowunif(x0=-mNISP$`Start (BC)`, x1=-mNISP$`End (BC)`),
                    y=mNISP$`NISP/m2`,k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

Ratio of fish and mammal NISP per site

```
# Marine mammals vs total mammals
nisp_ratio<- mNISP$`Total marine mammal NISP` / mNISP$`Total mammal NISP`
ratio<-MCrunmed(x=rowunif(x0=-mNISP$`Start (BC)`, x1=-mNISP$`End (BC)`),
                y=nisp_ratio,k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

```
# Marine mammals vs fish
aNISP<-merge(fNISP[fNISP$`Site type`=='Shell midden',],
          mNISP[mNISP$`Site type`=='Shell midden',],
          by.x='Site name (site director(s) if multiple entries and/or sample)',
          by.y='Site name (site director(s) if multiple entries and/or sample)')
mm_fish_ratio <- aNISP$`Total marine mammal NISP`/ as.numeric(aNISP$NISP)

mfratio<-MCrunmed(x=rowunif(x0=-aNISP$`Start (BC).x`, x1=-aNISP$`End (BC).x`),
                  y=mm_fish_ratio,k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

Richness (number of taxa per site)

```
frich<-MCrunmed(x=rowunif(x0=-fNISP$`Start (BC)`, x1=-fNISP$`End (BC)`),
                y=fNISP$`No. of taxa per site (richness)`,k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

```
mrich<-MCrunmed(x=rowunif(x0=-mNISP$`Start (BC)`, x1=-mNISP$`End (BC)`),
                y=mNISP$`No. of taxa per site (richness)`,k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

Stable isotope results from marine fauna

Here we are interested detecting any environmental change with time, as a baseline for interpreting human palaeodiet. We have controlled for the influence that tropic position would otherwise have on the results by conducting our analysis separately for three trophic levels: 4.1 (e.g. cod), 3.6 (e.g. eel) and 3.2 (e.g. flounder).

```
aiso<-read_xlsx(dkURL, sheet='Isotopic data of faunal remains')

t3.2<-aiso[which(aiso$`Trophic level position`==3.2),] |> as.data.frame()
t3.6<-aiso[which(aiso$`Trophic level position`==3.6),] |> as.data.frame()
t4.1<-aiso[which(aiso$`Trophic level position`==4.1),] |> as.data.frame()

# Make numeric column for d13C to clean up source data
t3.2$d13C<-NA; for(N in 1:nrow(t3.2))
  try(t3.2[N,'d13C']<-as.numeric(t3.2$` 13C (‰, VPDB)`[[N]]))
t3.6$d13C<-NA; for(N in 1:nrow(t3.6))
  try(t3.6[N,'d13C']<-as.numeric(t3.6$` 13C (‰, VPDB)`[[N]]))
t4.1$d13C<-NA; for(N in 1:nrow(t4.1))
  try(t4.1[N,'d13C']<-as.numeric(t4.1$` 13C (‰, VPDB)`[[N]]))

# remove NA from date in t4.1
t4.1<-t4.1[which(!is.na(t4.1$`Start (BC)`)),]

# Calculate the various moving average models
Ct36<-MCrunmed(x=rowunif(-t3.6$`Start (BC)`, -t3.6$`End (BC)`),
               y=t3.6$d13C, k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

```
Ct41<-MCrunmed(x=rowunif(-t4.1$`Start (BC)`, -t4.1$`End (BC)`),
               y=t4.1$d13C, k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

```
Nt36<-MCrunmed(x=rowunif(-t3.6$`Start (BC)`, -t3.6$`End (BC)`),
               y=t3.6$` 15N (‰, AIR)`, k=51) |> MCr.as.MCd()
```

```
## ================================================================================
```

```
Nt41<-MCrunmed(x=rowunif(-t4.1$`Start (BC)`, -t4.1$`End (BC)`),
               y=t4.1$` 15N (‰, AIR)`, k=51) |> MCr.as.MCd()
```

```
## ============================================================================
```

```
Ct32<-MCrunmed(x=rowunif(-t3.2$`Start (BC)`, -t3.2$`End (BC)`),
               y=t3.2$d13C, k=51) |> MCr.as.MCd()
```

```
## ============================================================================
```

```
Nt32<-MCrunmed(x=rowunif(-t3.2$`Start (BC)`, -t3.2$`End (BC)`),
               y=t3.2$` 15N (‰, AIR)`, k=51) |> MCr.as.MCd()
```

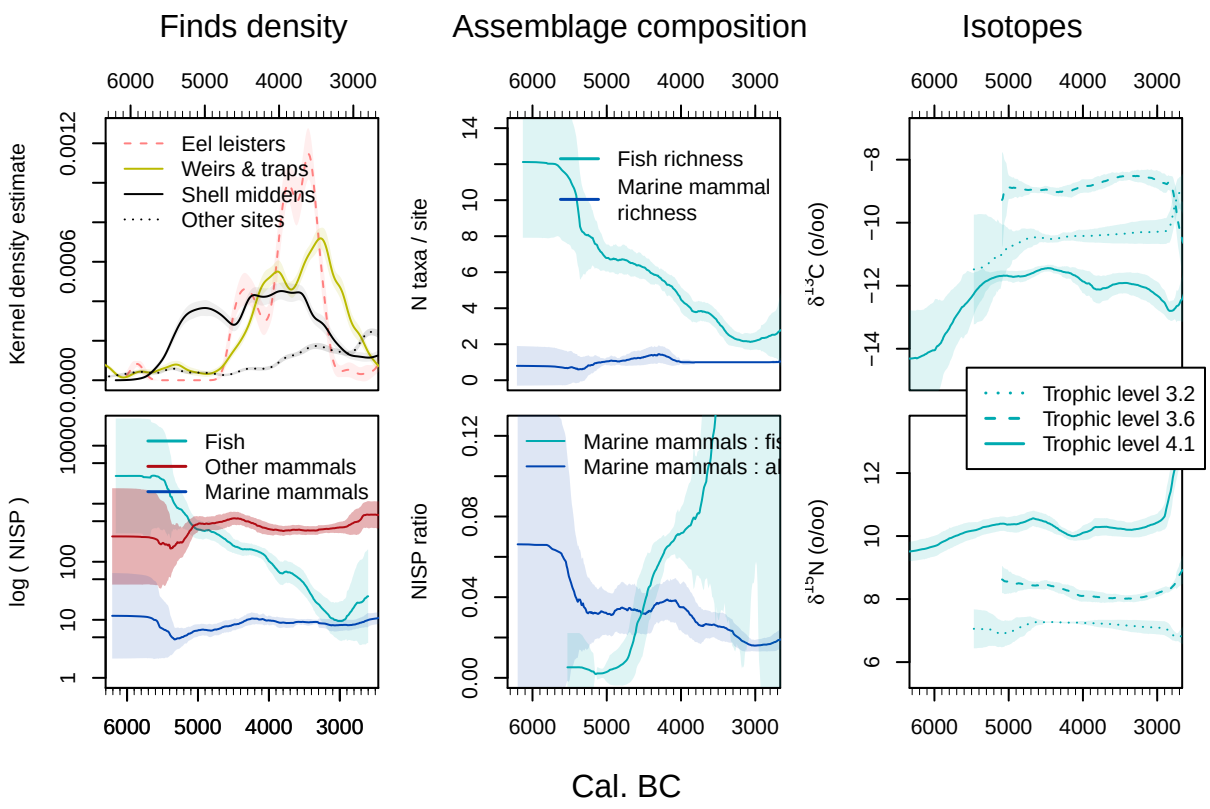
```
## ============================================================================
```

```
# censor 3.6 data pre 5000 BC as the poorly resolved plot is visually misleading
cs<-function(x){o<-x[which(x[,1]>-5100),]; class(o)<-'MCd';o}
Ct36<-cs(Ct36); Nt36<-cs(Nt36)
```

Multipanel plot for animals

```
# pdf('animals.pdf', width=10.5, height=6)

#Expressions for y-axis labels
d13Cpm<-expression(delta * ""^{13} * "C (‰)")
d15Npm<-expression(delta * ""^{15} * "N (‰)")

par(mfcol=c(2,3),mar=c(1,4.1,0,1),omi=c(0.7,0.12,0.7,0.1),xpd=FALSE)

plot(leisD, xlim=c(-6200,-2800), lty=2, col='#FF8080', fill='#FF808022',
     xaxt='n',ylab='Kernel density estimate')
  plot(fingD, add=TRUE, col='#baba00', fill='#baba0022')
  plot(dk_kde, add=T, fill='#00000010')
  ax(side=3,ticksize = -0.025)
  legend('topleft',lty=c(2,1,1,3),col=c('#FF8080','#baba00',1,1),bty='n',
         legend=c('Eel leisters','Weirs & traps','Shell middens','Other sites'))
  plot(nonSMD, add=T, lty=3)

plot(log(nisp_fish),yaxt='n', ylim=c(0,10),col='#00AAB0',fill='#00AAB022',
     ylab='log ( NISP )',xaxt='n');ax()
  axis(2, at=log(c(1,5,10,50,100,500,1000,5000,10000)),
       lab=c(1,NA,10,NA,100,NA,1000,NA,10000) ) # Log axis
  plot(log(nisp_om),add=T,col='#B00B13',fill='#B00B1350')
  plot(log(nisp_mm),add=T,col='#0044B0',fill='#0044B022')
  legend('topright',lty=1,lwd=1.5,col=c('#00AAB0','#B00B13','#0044B0'),
         legend=c('Fish','Other mammals','Marine mammals'),bty='n')
  ax(ticksize = -0.025)

plot(frich,ylab='N taxa / site',xaxt='n',xlim=c(-6200,-2800), ylim=c(0,14),
```

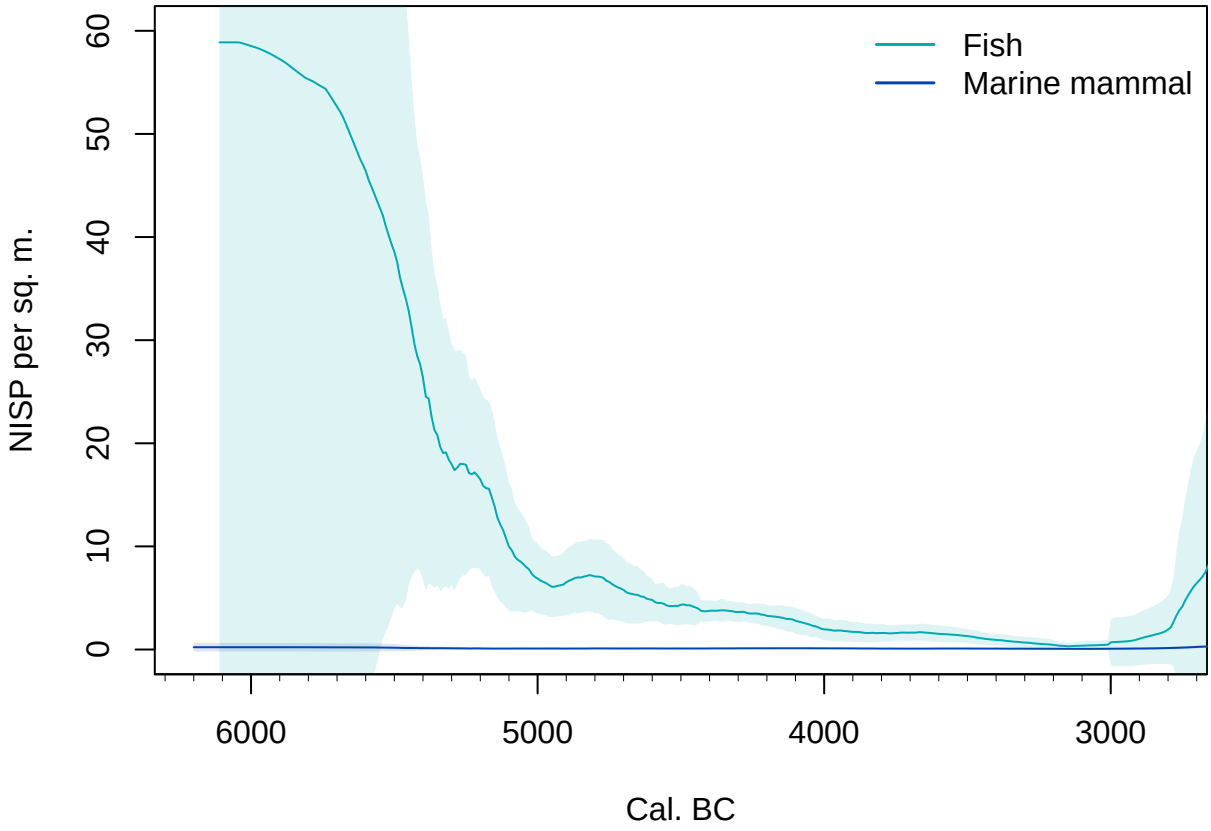
```
     col='#00AAB0',fill='#00AAB022')
  plot(mrich,add=TRUE,col='#0044B0',fill='#0044B022')
  legend('topright',lty=1,lwd=1.5,col=c('#00AAB0','#0044B0'),
         legend=c('Fish richness','Marine mammal\nrichness'),bty='n')
  ax(side=3,ticksize = -0.025)

plot(ratio, ylab='NISP ratio',xaxt='n',xlim=c(-6200,-2800),ylim=c(0,0.125),
     col='#0044B0',fill='#0044B022')
  plot(mfratio, add=TRUE, col='#00AAB0',fill='#00AAB022')
  legend('topleft',lty=1,col=c('#00AAB0','#0044B0'),bty='n',
         legend=c('Marine mammals : fish','Marine mammals : all mammals'))
  ax(ticksize = -0.025)

plot(Ct36,lty=2,ylab=d13Cpm,xaxt='n',xlim=c(-6200,-2800),
     col='#00AAB0',fill='#00AAB022', ylim=c(-15,-7))
  plot(Ct32, add=T,lty=3,col='#00AAB0',fill='#00AAB022' )
  plot(Ct41, add=T,col='#00AAB0',fill='#00AAB022')
  ax(side=3,ticksize = -0.025)

plot(Nt36,lty=2,ylab=d15Npm,xaxt='n',xlim=c(-6200,-2800),
     col='#00AAB0',fill='#00AAB022', ylim=c(5.5,13.5))
  plot(Nt41, add=T,col='#00AAB0',fill='#00AAB022')
  ax(ticksize = -0.025)
  plot(Nt32, add=T, lty=3, col='#00AAB0',fill='#00AAB022')

mtext('Cal. BC',side=1,outer=TRUE,line=2.5,adj = 0.52)
mtext(c('Finds density','Assemblage composition','Isotopes'), side=3,
      outer=TRUE, line=3, adj=c(0.155,0.53,0.885))

par(fig=c(0, 1, 0, 1), oma=c(0, 0, 0, 0), mar=c(0, 0, 0, 0), new=TRUE)
plot(0, 0, type='n', bty='n', xaxt='n', yaxt='n')
legend(0.63,0.12,col=c(rep('#00AAB0',2)),lwd=1.5,lty=c(3,2,1),
      c('Trophic level 3.2','Trophic level 3.6','Trophic level 4.1'),bg='white')
```




```
#dev.off()
```

Plot of NISP per square meter:

```
par(mar=c(4.1,4.1,1,1))
plot(nisp_fish_sqm,ylab='NISP per sq. m.',xaxt='n',xlim=c(-6200,-2800),
     ylim=c(0,60),col='#00AAB0',fill='#00AAB022')
  plot(nisp_mm_sqm,add=TRUE,col='#0044B0',fill='#0044B022')
  legend('topright',lty=1,lwd=1.5,col=c('#00AAB0','#0044B0'),
         legend=c('Fish','Marine mammal'),bty='n')
  ax()
```



Human isotopes

Moving average models for shellmidden from non-shellmidden individuals

The following script extracts the relevant data from the spreadsheet and calls the `MCrunmed` algorithm detailed in Part I. Here, moving models are calculated separately for samples of human bone tissue found from shell middens, and those found from other sites.

```
hiso<-read_xlsx(dkURL,
  sheet='Isotopic data & C-14 humans') |> as.data.frame()

CN<-c('Radiocarbon age (BP)', 'Radiocarbon error','Curve',' 13C (‰, VPDB)',
      ' 15N (‰, AIR)')

# Samples from Denmark NOT from shell middens (dkC and dkN)
dk_iso<-hiso[which( hiso$`Site type` != 'Shell midden' &
              hiso$Curve %in% goodcurves & !is.na(hiso$` 15N (‰, AIR)`)), CN]
colnames(dk_iso)<-c('BP','Error','Curve','d13C','d15N')

# Calculate moving average models
dkC<-MCrunmed(x=rowcal(dl=dk_iso[,c(1:3)]), y=dk_iso[,4], k=21) |> MCr.as.MCd()
```

```
## ============================================================================
```

```
dkN<-MCrunmed(x=rowcal(dl=dk_iso[,c(1:3)]), y=dk_iso[,5], k=21) |> MCr.as.MCd()

## ========================================================================================

# Samples from shell middens (smC and smN)
sm_iso<-hiso[which( hiso$`Site type` == 'Shell midden' &
             hiso$Curve %in% goodcurves & !is.na(hiso$` 15N (%, AIR)`)), CN]

colnames(sm_iso)<-c('BP','Error','Curve','d13C','d15N')
smC<-MCrunmed(x=rowcal(dl=sm_iso[,c(1:3)]), y=sm_iso[,4], k=21) |> MCr.as.MCd()

## ========================================================================================

smN<-MCrunmed(x=rowcal(dl=sm_iso[,c(1:3)]), y=sm_iso[,5], k=21) |> MCr.as.MCd()

## ========================================================================================

# Bootstrapped versions
dkCB<-MCrunmed(x=rowcal(dl=dk_iso[,c(1:3)]), y=dk_iso[,4], k=21,
               boot=TRUE) |> MCr.as.MCd()

## ========================================================================================

dkNB<-MCrunmed(x=rowcal(dl=dk_iso[,c(1:3)]), y=dk_iso[,5], k=21,
               boot=TRUE) |> MCr.as.MCd()

## ========================================================================================

smCB<-MCrunmed(x=rowcal(dl=sm_iso[,c(1:3)]), y=sm_iso[,4], k=21,
               boot=TRUE) |> MCr.as.MCd()

## ========================================================================================

smNB<-MCrunmed(x=rowcal(dl=sm_iso[,c(1:3)]), y=sm_iso[,5], k=21,
               boot=TRUE) |> MCr.as.MCd()

## ========================================================================================
```

Temporal density and median radiocarbon date

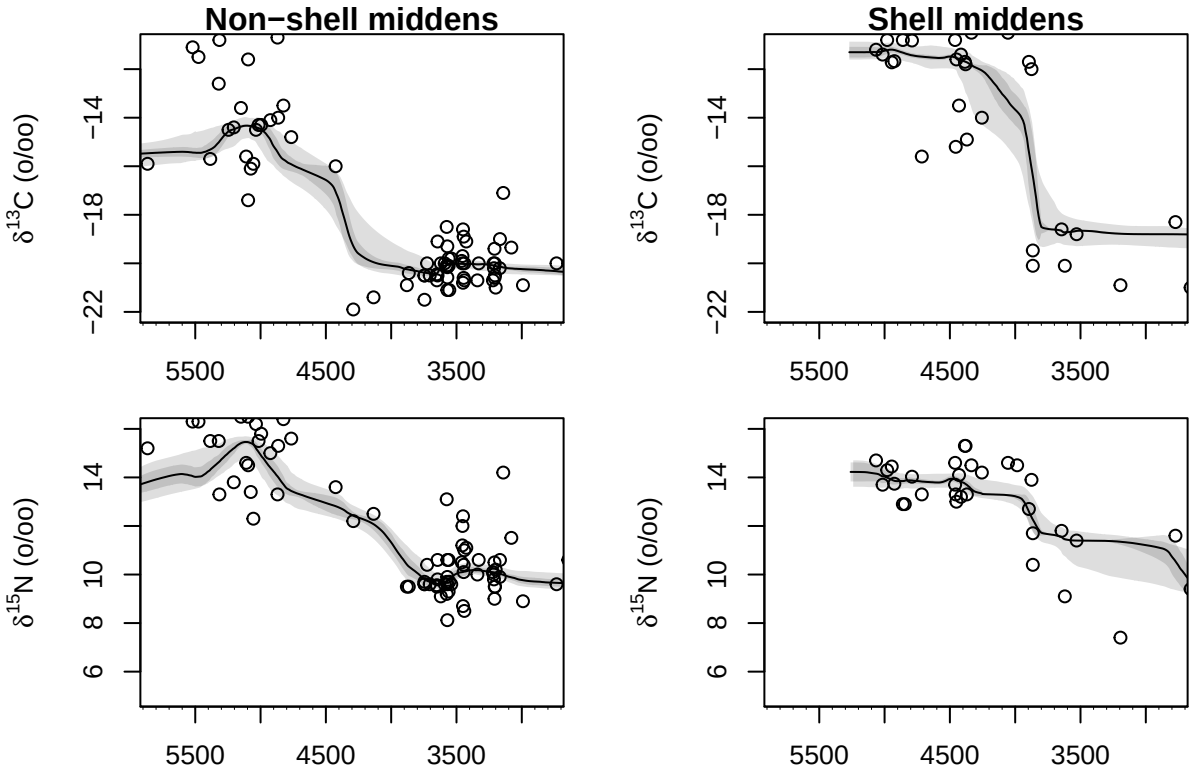
As before, we calculate a KDE for the radiocarbon dated samples. See notes in the `rowcal` package documentaiton for further details on how the median date calculated.

```
dk_iso_kde<-MCdensity(dl=dk_iso[,1:3], bw=75)
sm_iso_kde<-MCdensity(dl=sm_iso[,1:3], bw=75)

dk_iso$median<-findmedian(rowcal(dl=dk_iso[,1:3]))
sm_iso$median<-findmedian(rowcal(dl=sm_iso[,1:3]))
```

Visualising trends in isotopes

In the following plot, both normal and bootstapped versions of the moving average model are presented. The uncertainty of the bootstrapped version, which is of a greater magnitude, reflects the variability in the data as well as its inherant chronological uncertainty.

```
par(mfcol=c(2,2), mar=c(2,4.3,1,2),omi=c(0.2,0,0.3,0))
plot(dkC, ylim=c(-22,-11), xaxt='n',main='Non-shell middens',xlim=c(-5800,-2800)
     ,ylab=d13Cpm); ax()
points(dk_iso$median,dk_iso$d13C)
plot(dkCB, add=TRUE, lty=0)

plot(dkN, ylim=c(5,16), xaxt='n',ylab=d15Npm,xlim=c(-5800,-2800)); ax()
points(dk_iso$median,dk_iso$d15N)
plot(dkNB, add=TRUE, lty=0)

plot(smC, ylim=c(-22,-11), xaxt='n',main='Shell middens',ylab=d13Cpm
     ,xlim=c(-5800,-2800)); ax()
points(sm_iso$median,sm_iso$d13C)
plot(smCB, add=TRUE, lty=0)

plot(smN, ylim=c(5,16), xaxt='n',ylab=d15Npm,xlim=c(-5800,-2800)); ax()
points(sm_iso$median,sm_iso$d15N)
plot(smNB, add=TRUE, lty=0)
```



SIBER analysis

The R package `SIBER` (Jackson 2023) is used here to plot ellipses summarizing each group (shell midden / non shell midden) for two 'communities' (Mesolithic / Neolithic).

```
library(SIBER)
# Group 1 is shell middens, 2 is other. Ecological 'community' is chronology
dk_iso$group=2; dk_iso$community=NA
sm_iso$group=1; sm_iso$community=NA
dk_iso[dk_iso$median > -3800,'community']<-'Neolithic'
dk_iso[dk_iso$median < -3800,'community']<-'Mesolithic'
sm_iso[sm_iso$median > -3800,'community']<-'Neolithic'
sm_iso[sm_iso$median < -3800,'community']<-'Mesolithic'
temp<-rbind(dk_iso, sm_iso)[,c('d13C','d15N','group','community')]
colnames(temp)<-c("iso1", "iso2", "group", "community")
all_iso<-createSiberObject(temp)
#ellipses.posterior <- siberMVN(all_iso, parms, priors)
#group.ML <- groupMetricsML(all_iso)
#colnames(group.ML)<-c('Meso\nother','Meso.\nmidden','Neo.\nother','Neo.\nmidden')
```

Aggregation by genetic cluster

```
cn<-c(CN,
"Cluster group (Allentoft et al. 2024a; Allentoft et al. 2024b; Fischer et al. 2024)")
cl<-c('Farmer_EuropeE_late','HG_EuropeE','HG_EuropeW','PostNeol_EuropeW')
gh<-hiso[
  hiso$`Cluster group (Allentoft et al. 2024a; Allentoft et al. 2024b; Fischer et al. 2024)`
  %in% cl,cn] |> as.data.frame()
gh[,6]<-as.factor(gh[,6])
key<-factor(gh$`Cluster group (Allentoft et al. 2024a; Allentoft et al. 2024b; Fischer et al. 2024)`)
```

###Bayesian food source analysis

The following chuck requires the `simmr` package and its dependencies, hence also an installation of JAGS.

Here we use the average values at 4500 and 3500 BC to explore food sources. Target values were calculated by calling `mean(dk_iso[which(dk_iso$community=='Mesolithic'),'d13C'])`, etc.

```
library(simmr)
# Compile data into list format required by simmr
#1 Meso SM, 2 Meso non-sm, 3 Neo sm, 4 Neo non-sm
targets<-data.frame(d13C=c(-12.4, -15.7,-19.4,-20.1),
                    d15N=c(13.7, 13.9, 9.6,9.9),
                    ID=c('Meso SM','Meso non-SM','Neo SM','Neo non-SM'))
# Plants Kanstrup et al 2014,
sources<-data.frame(source=c('Plants','Animals','Marine'),
                   mean13C=c(-27,-21,-11.5),
                   mean15N=c(4,6,9),
                   sd13C=c(2,1,2),
                   sd15N=c(2,1,2))
TEFs<-data.frame(source=c('Plants','Animals','Marine'),
                 mean13Ce=c(4.8,0.8,0.8),
                 mean15Ne=c(4.6,4,4),
                 sd13Ce=c(0.5,0.5,0.5),
                 sd15Ne=c(1,1,1))

dk_simmr<-simmr_load(mixtures = as.matrix(targets[, 1:2]),
```

```
                     source_names = sources$source,
                     source_means = as.matrix(sources[,2:3]),
                     source_sds = as.matrix(sources[,4:5]),
                     correction_means = as.matrix(TEFs[,2:3]),
                     correction_sds = as.matrix(TEFs[,4:5]),
                     group = as.factor(targets$ID))

mixmod<-simmr_mcmc(dk_simmr)
```

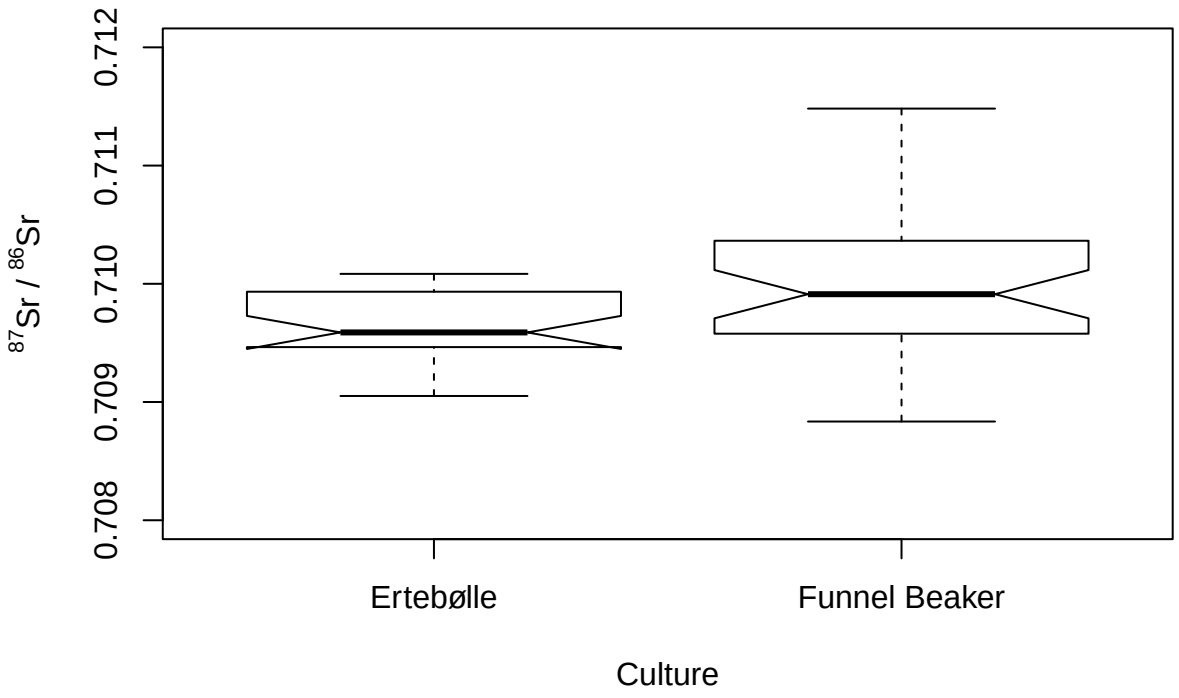
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 2
##    Unobserved stochastic nodes: 5
##    Total graph size: 101
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 2
##    Unobserved stochastic nodes: 5
##    Total graph size: 101
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 2
##    Unobserved stochastic nodes: 5
##    Total graph size: 101
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 2
##    Unobserved stochastic nodes: 5
##    Total graph size: 101
##
## Initializing model
```

```
 post_MSM<-as.data.frame(mixmod$output$`Meso SM`$BUGSoutput$sims.list$p)
post_MnSM<-as.data.frame(mixmod$output$`Meso non-SM`$BUGSoutput$sims.list$p)
 post_NSM<-as.data.frame(mixmod$output$`Neo SM`$BUGSoutput$sims.list$p)
post_NnSM<-as.data.frame(mixmod$output$`Neo non-SM`$BUGSoutput$sims.list$p)
```

```
tx95<-function(x){
  qu<-round(quantile(x,probs =c(0.025,0.975))*100)
  return(paste(qu[1],'to',qu[2]))
}

mixresults<-data.frame(ID=c('Meso SM','Meso non-SM','Neo SM','Neo non-SM'),
          plantmedian=round(c(median(post_MSM$Plants),median(post_MnSM$Plants),
                              median(post_NSM$Plants),median(post_NnSM$Plants))*100),
          plantci=c(tx95(post_MSM$Plants),tx95(post_MnSM$Plants),
                    tx95(post_NSM$Plants),tx95(post_NnSM$Plants)),
          animalmedian=round(c(median(post_MSM$Animals),median(post_MnSM$Animals),
                              median(post_NSM$Animals),median(post_NnSM$Animals))*100),
          animalci=c(tx95(post_MSM$Animals),tx95(post_MnSM$Animals),
                    tx95(post_NSM$Animals),tx95(post_NnSM$Animals)),
          marinemedian=round(c(median(post_MSM$Marine),median(post_MnSM$Marine),
                              median(post_NSM$Marine),median(post_NnSM$Marine))*100),
          marineci=c(tx95(post_MSM$Marine),tx95(post_MnSM$Marine),
                    tx95(post_NSM$Marine),tx95(post_NnSM$Marine)))

# Output results to csv file
# write.csv(mixresults,'~/Desktop/mix_model_results.csv')
```

Strontium isotope variability by cultural group

```
sr<-hiso[which(!is.na(hiso$`87Sr/86Sr`) & hiso$Culture %in% c('Ertebølle','Funnel Beaker')),c('Culture'

sr_yl<-expression(""^{87}*"Sr / "^{86}*"Sr")

plot(as.factor(sr[,1]),sr[,2],outline=FALSE,ylab=sr_yl,xlab='Culture',
     ylim=c(0.708,0.712),col='white',notch=TRUE)
```



Carbon-Nitrogen biplots

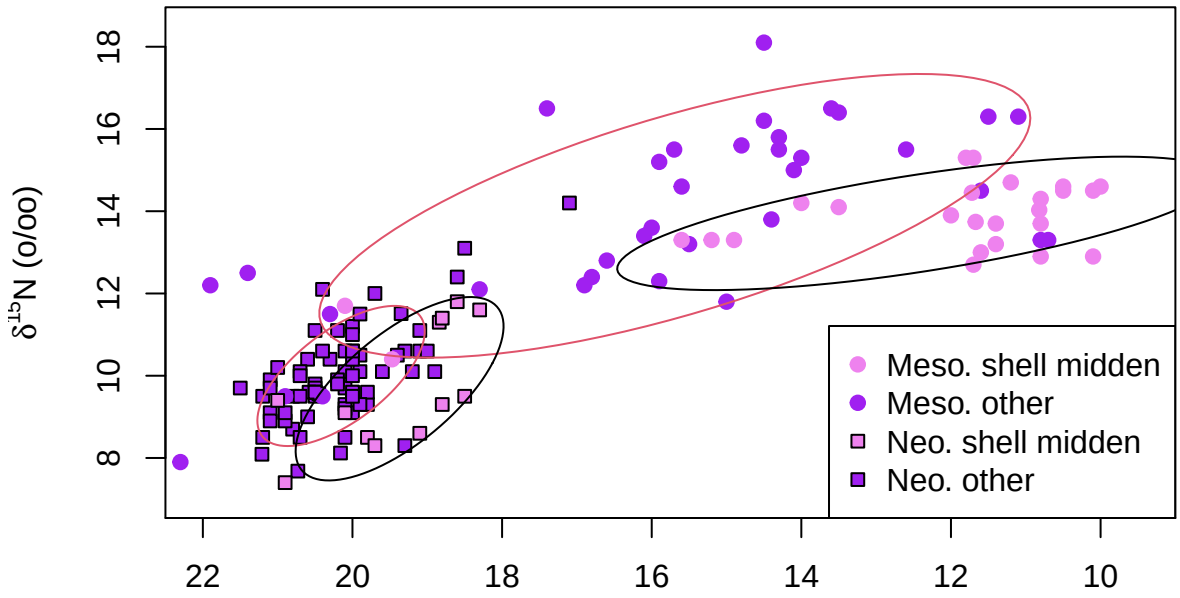
First we prepare some comparitive data from the faunal database:

```
da<-read_xlsx(dkURL, sheet='C-14 dates from domesticates') |> as.data.frame()
bos<-da[which(!is.na(da$` 13C (‰, VPDB)`) &
                da$Species %in% c('Bos taurus','cf. Bos taurus')),
        c(' 13C (‰, VPDB)',' 15N (‰, AIR)')]
cap<-da[which(!is.na(da$` 13C (‰, VPDB)`) &
                da$Species %in% c('Capra sp.','Capra sp./Ovis sp.','Ovis aries',
                                  'Ovis sp.')),
        c(' 13C (‰, VPDB)',' 15N (‰, AIR)')]
sus<-da[which(!is.na(da$` 13C (‰, VPDB)`) &
                da$Species %in% c('Sus domesticus','Sus scrofa f. domestica')),
        c(' 13C (‰, VPDB)',' 15N (‰, AIR)')]
```

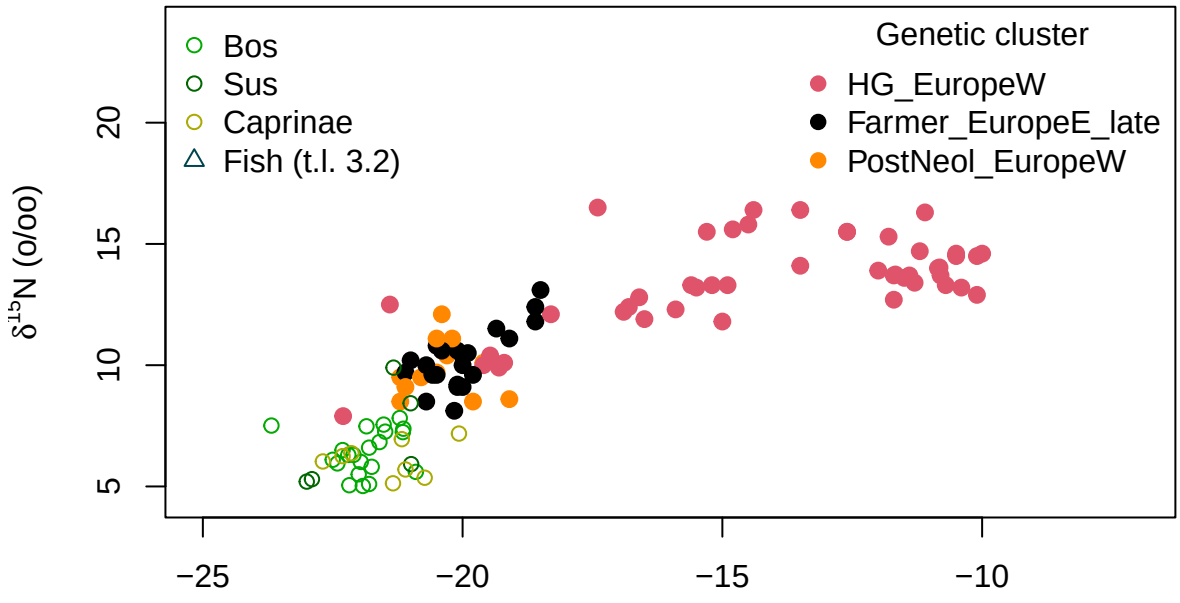
Alternative biplots can be made:

```
par(mfrow=c(1,1))
plot(dk_iso[dk_iso$median > -3800,'d13C'],dk_iso[dk_iso$median > -3800,'d15N'],
     pch=22,bg='purple',xlim=c(-22,-9.5),ylim=c(7,18.5),xlab=NA,
     ylab=d15Npm,xaxt='n')
ax()
points(dk_iso[dk_iso$median < -3800,'d13C'],dk_iso[dk_iso$median < -3800,'d15N'],
       pch=19,col='purple')
points(sm_iso[sm_iso$median > -3800,'d13C'],sm_iso[sm_iso$median > -3800,'d15N'],
       pch=22,bg='violet')
points(sm_iso[sm_iso$median < -3800,'d13C'],sm_iso[sm_iso$median < -3800,'d15N'],
       pch=19,col='violet')
legend('bottomright',pch=c(19,19,22,22),col=c('violet','purple',1,1),
```

```
       pt.bg=c(NA,NA,'violet','purple'),legend=c('Meso. shell midden',
                                  'Meso. other','Neo. shell midden','Neo. other'))
plotGroupEllipses(all_iso,p.interval=0.68)
```




```
gc<-c(1,2,'#FF8800')
plot(gh$` 13C (%, VPDB)`,gh$` 15N (%, AIR)`,col=gc[as.numeric(key)],pch=19,cex=1.1,
     xlab=NA, ylab=d15Npm,xlim=c(-25,-7),ylim=c(4.5,24))
points(bos, pch=1,col='#00AA00')
points(cap, pch=1,col='#006600')
points(sus, pch=1,col='#AAAA00')
points(t3.2$d13C,t3.2$` 15N (%)`,pch=2,col='#004450')

legend('topright',legend=unique(key)[c(2,3,1)],pch=19,col=c(2,1,'#FF8800'),
       title='Genetic cluster',bty='n')
legend('topleft',legend=c('Bos','Sus','Caprinae','Fish (t.l. 3.2)'),pch=c(1,1,1,2),col=c('#00AA00','#00
```

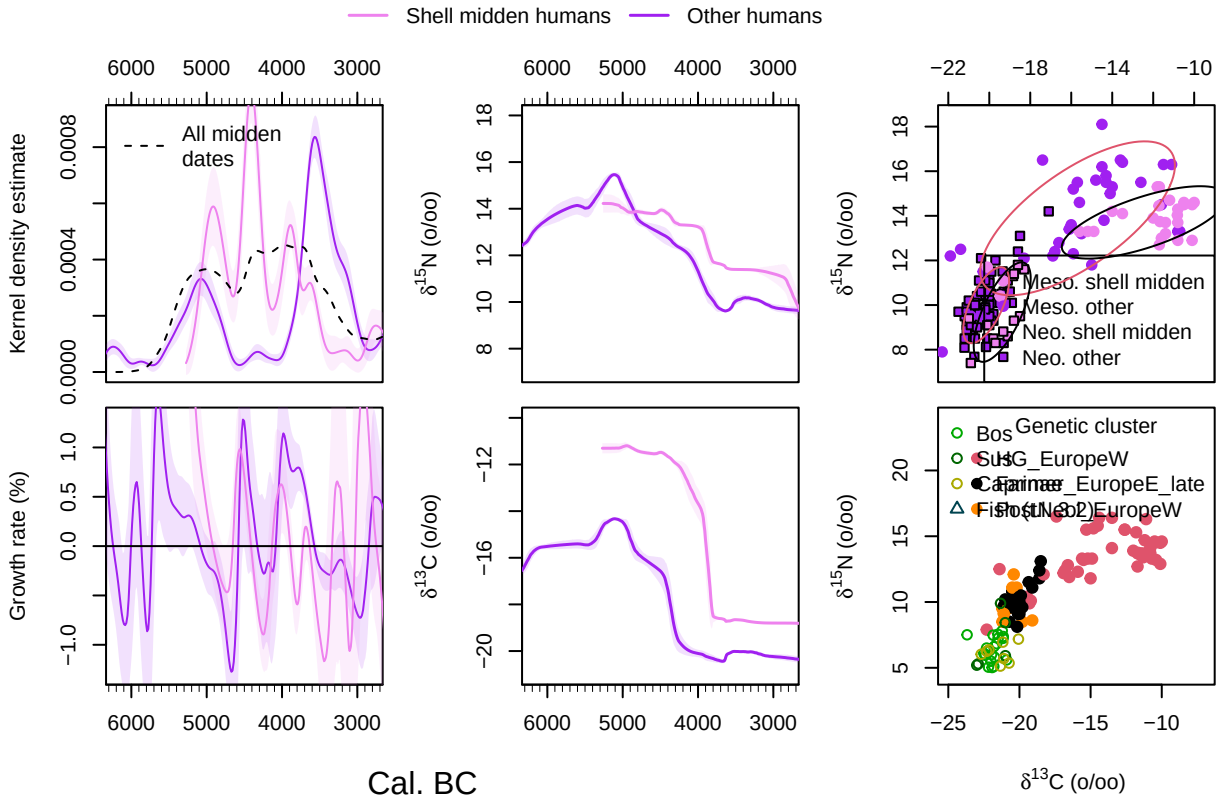


Multi-panel plot for human isotope results

The plot is composed as follows:

```
# pdf('humans.pdf',width=10.5, height=6)

par(mfcol=c(2,3), mar=c(1,4.5,0,1), omi=c(0.75,0,0.6,0))
options(scipen = 999)
plot(dk_iso_kde, xaxt='n', col='purple', fill='#a01ef022', xlim=c(-6200,-2800),
     ylab='Kernel density estimate')
ax(side=3, ticksize = -0.025)
plot(sm_iso_kde, add=TRUE, col='violet', fill='#ee81ee22')
plot(dk_kde, add=T, fill=NA, lty=2)
legend('topleft',lty=2,bty='n','All midden\ndates')

plot(ggr(dk_iso_kde), xaxt='n', col='purple', fill='#a01ef022',
     xlim=c(-6200,-2800), ylim=c(-1.3,1.3), ylab='Growth rate (%)')
ax(ticksize = -0.025)
plot(ggr(sm_iso_kde), add=TRUE, col='violet', fill='#ee81ee22')
abline(h=0)

plot(dkN, xlim=c(-6200,-2800), ylim=c(7,18), xaxt='n',xlab='', col='purple',
     fill='#a01ef022', lwd=1.5, ylab=d15Npm); ax(side=3, ticksize = -0.025)
plot(smN, add=TRUE, col='violet', lwd=1.5, fill='#ee81ee22')

plot(dkC, xlim=c(-6200,-2800), ylim=c(-21,-10),ylab=d13Cpm, xlab='',
     col='purple',fill='#a01ef022', lwd=1.5, xaxt='n'); ax(ticksize = -0.025)
plot(smC, add=TRUE, col='violet', fill='#ee81ee22', lwd=1.5)

plot(dk_iso[dk_iso$median > -3800,'d13C'],dk_iso[dk_iso$median > -3800,'d15N'],
     pch=22,bg='purple',xlim=c(-22,-9.5),ylim=c(7,18.5),xlab=NA,
     ylab=d15Npm,xaxt='n')
axis(3)
points(dk_iso[dk_iso$median < -3800,'d13C'],dk_iso[dk_iso$median < -3800,'d15N'],
       pch=19,col='purple')
points(sm_iso[sm_iso$median > -3800,'d13C'],sm_iso[sm_iso$median > -3800,'d15N'],
       pch=22,bg='violet')
points(sm_iso[sm_iso$median < -3800,'d13C'],sm_iso[sm_iso$median < -3800,'d15N'],
       pch=19,col='violet')
legend('bottomright',pch=c(19,19,22,22),col=c('violet','purple',1,1),
       pt.bg=c(NA,NA,'violet','purple'),legend=c('Meso. shell midden',
                                'Meso. other','Neo. shell midden','Neo. other'))
plotGroupEllipses(all_iso,p.interval=0.68)

gc<-c(1,2,'#FF8800')
plot(gh$`13C (%, VPDB)`,gh$`15N (%, AIR)`,col=gc[as.numeric(key)],pch=19,cex=1.1,
     xlab=NA, ylab=d15Npm,xlim=c(-25,-7),ylim=c(4.5,24))
points(bos, pch=1,col='#00AA00')
points(cap, pch=1,col='#006600')
points(sus, pch=1,col='#AAAA00')
points(t3.2$d13C,t3.2$`15N (%)`,pch=2,col='#004450')

legend('topright',legend=unique(key)[c(2,3,1)],pch=19,col=c(2,1,'#FF8800'),
       title='Genetic cluster',bty='n')
```

```
legend('topleft',legend=c('Bos','Sus','Caprinae','Fish (t.l. 3.2)'),pch=c(1,1,1,2),col=c('#00AA00','#000

mtext('Cal. BC',side=1,line=2.5,outer=TRUE,adj=0.33)
mtext(d13Cpm,side=1,line=2.5,outer=TRUE,adj=0.90,cex=0.66)

par(fig=c(0, 1, 0, 1), oma=c(0, 0, 0, 0), mar=c(0, 0, 0, 0), new=TRUE)
plot(0, 0, type='n', bty='n', xaxt='n', yaxt='n')
legend('top', ncol=2, lty=1, lwd=c(1.5,1.5), col=c('violet','purple'),
       legend=c('Shell midden humans','Other humans'), bty='n')
```




```
#dev.off()
```

Organic residues

Chronological control for the organic residue analysis (ORA) data is provided by an external OxCal model whose .js file is loaded from disc along with the ORA data. This takes some time, so an rData file containing the posteriors has been included in the same respository as this document.

```
library(oxcAAR)
temp<-readOxcalOutput('Robson_et_al.js')
OxCalModel<-parseFullOxcalOutput(temp) # Takes some time
```

Load ORA data from spreadsheet:

```
ora<-read_xlsx(dkURL, sheet='ORA of ceramics', na = "NA")
```

```
## New names:
## * `Fraction` -> `Fraction...37`
## * `13C/12C` -> `13C/12C...38`
## * `13C/12C unc` -> `13C/12C unc...39`
## * `Fraction` -> `Fraction...40`
## * `13C/12C` -> `13C/12C...41`
## * `13C/12C unc` -> `13C/12C unc...42`
## * `13C/12C unc` -> `13C/12C unc...44`
## * `Fraction` -> `Fraction...62`
## * `13C/12C` -> `13C/12C...63`
## * `Fraction` -> `Fraction...64`
```

```
ora<-as.data.frame(ora)
ora$Ware<-factor(ora$Ware)
colnames(ora)[
  which(colnames(ora)=="13C/12C offset fractions (C18:0-C16:0)")]<-'bigD'
# Load chronological control
```

Combining posteriors and uniform estimates of sample dates

The following scipt builds a list in format used by `MCdensity` and `MCrunmed` to enable density estimation and moving average models.

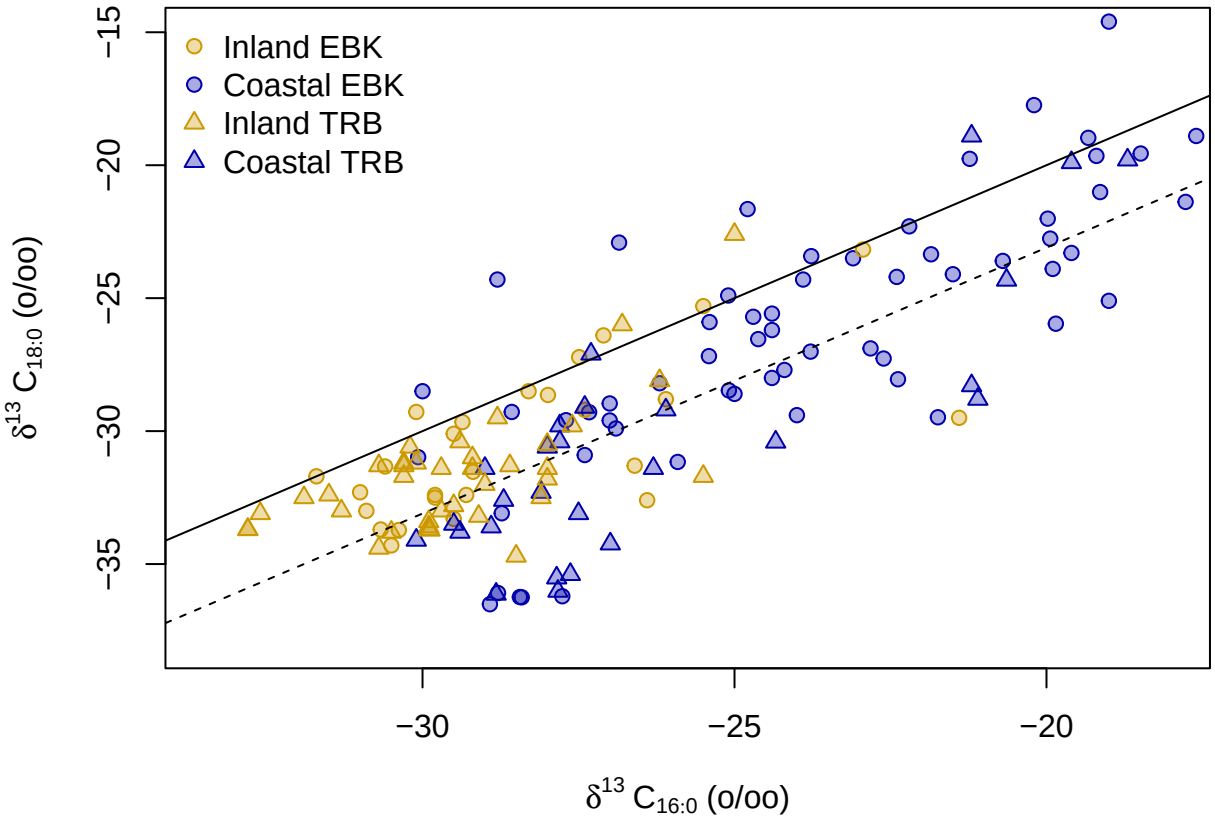
```
ora_dates<-list()
for(i in 1:nrow(ora)) {
  oxx<-ora[i,'Entity in Robson et al. (2021) OxCal model']
  if(is.na(oxx)) ora_dates[[i]]<-rowunif(-ora[i,'Start (BC)'], -ora[i,'End (BC)'])
  else ora_dates[[i]]<-ox.as.rowcal(oxx)
}
class(ora_dates)<-'rowyears'
```

Scatter plot of GC-c-IRMS results

For a scatter plot of these data:

```
CE<-which(ora$`Coastal/inland`=='Coastal' & ora$Ware=='EBK')
IE<-which(ora$`Coastal/inland`=='Inland' & ora$Ware=='EBK')
CT<-which(ora$`Coastal/inland`=='Coastal' & ora$Ware=='TRB')
IT<-which(ora$`Coastal/inland`=='Inland' & ora$Ware=='TRB')

par(mfrow=c(1,1), mar=c(4.3,4.3,1,1), omi=c(0,0,0,0))
plot(ora[CE,]$`13C/12C...38`,ora[CE,]$`13C/12C...41`,bg='#0000AA55',
       col='#0000AA',pch=21, xlim=c(-33.5,-18),ylim=c(-38,-15),
     xlab=expression(delta*""^{13}~"C"*""[16:0]*" (‰)"),
     ylab=expression(delta*""^{13}~"C"*""[18:0]*" (‰)"))
points(ora[IE,]$`13C/12C...38`,ora[IE,]$`13C/12C...41`,bg='#CC990055',
     col='#CC9900',pch=21)
points(ora[CT,]$`13C/12C...38`,ora[CT,]$`13C/12C...41`,bg='#0000AA55',
       col='#0000AA',pch=24)
```

```
points(ora[IT,]$`13C/12C...38`,ora[IT,]$`13C/12C...41`,bg='#CC990055',
     col='#CC9900',pch=24)
legend('topleft',legend=c('Inland EBK','Coastal EBK','Inland TRB',
                          'Coastal TRB'),pch=c(21,21,24,24),
       pt.bg=c('#CC990055','#0000AA55','#CC990055','#0000AA55'),
       col=c('#CC9900','#0000AA','#CC9900','#0000AA'),bty='n')
 abline(0,1,lty=1)
 abline(-3.1,1,lty=2) # to indicate where Delta(C18-C16)=3.1
```




```
# Big delta plot
```

The temporal density of these samples is calculated as follows:

```
# store indexes for coastal and inland sites
C<-which(ora$`Coastal/inland`=='Coastal')
I<-which(ora$`Coastal/inland`=='Inland')

# Temporal Kernel density of these samples
ora_coastalD<-MCdensity(ora_dates[C], bw=75)
ora_inlandD<-MCdensity(ora_dates[I], bw=75)
```

The various running average models are calculated for coastal and inland sites:

```
#delta 13C (16:0 fatty acids)
I_d13C.16.0<-MCrunmed(x=ora_dates[I],y=ora$`13C/12C...38`[I],k=51)
```

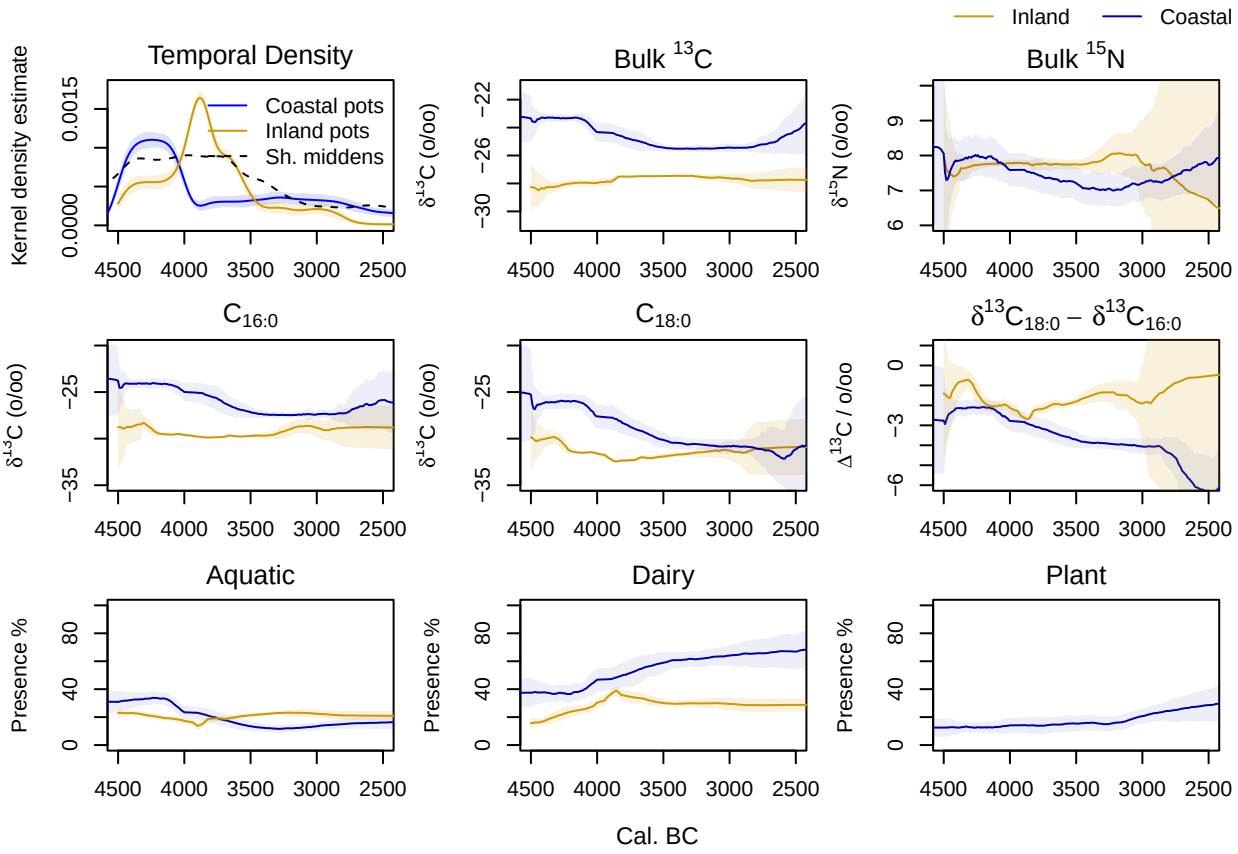
```
## ============================================================================

C_d13C.16.0<-MCrunmed(x=ora_dates[C],y=ora$`13C/12C...38`[C],k=51)

## ============================================================================

#delta 13C (18:0 fatty acids)
I_d13C.18.0<-MCrunmed(x=ora_dates[I],y=ora$`13C/12C...41`[I],k=51)

## ============================================================================

C_d13C.18.0<-MCrunmed(x=ora_dates[C],y=ora$`13C/12C...41`[C],k=51)

## ============================================================================

#Big Delta
I_bigD<-MCrunmed(x=ora_dates[I],y=ora$bigD[I],k=51)

## ============================================================================

C_bigD<-MCrunmed(x=ora_dates[C],y=ora$bigD[C],k=51)

## ============================================================================

#detla 15N bulk
I_d15N.bulk<-MCrunmed(x=ora_dates[I],y=ora$`15N/14N`[I],k=51)

## ============================================================================

C_d15N.bulk<-MCrunmed(x=ora_dates[C],y=ora$`15N/14N`[C],k=51)

## ============================================================================

#delta 13C bult
I_d13C.bulk<-MCrunmed(x=ora_dates[I],y=ora$`13C/12C...63`[I],k=51)

## ============================================================================

C_d13C.bulk<-MCrunmed(x=ora_dates[C],y=ora$`13C/12C...63`[C],k=51)

## ============================================================================

# Biomarkers (using running mean)
C_aq<-MCrunmean(x=ora_dates[C],y= ora$Aquatic[C]*100,k=50)

## ============================================================================
```

```
C_da<-MCrunmean(x=ora_dates[C],y=   ora$Dairy[C]*100,k=50)

## ================================================================================

C_pl<-MCrunmean(x=ora_dates[C],y=   ora$Plant[C]*100,k=50)

## ================================================================================

I_aq<-MCrunmean(x=ora_dates[I],y= ora$Aquatic[I]*100,k=50)

## ================================================================================

I_da<-MCrunmean(x=ora_dates[I],y=   ora$Dairy[I]*100,k=50)

## ================================================================================

I_pl<-MCrunmean(x=ora_dates[I],y=   ora$Plant[I]*100,k=50)

## ================================================================================
```

The multi-panel plot for these results is composed by the following code:

```
ax2<-function() axis(1,at=seq(-4500,-2500,500),lab=-seq(-4500,-2500,500))
BD<-expression(delta*""^{13}*"C"*""[18:0]~-~delta*""^{13}*"C"*""[16:0])

par(mfrow=c(3,3),mar=c(2.3,4.1,2,0.9),omi=c(0.24,0.05,0.2,0),xpd=FALSE)
plot(ora_coastalD,col='blue',fill='#0011AA22', main='Temporal Density',
     ylab='Kernel density estimate',xlim=c(-4500,-2500), ylim=c(0,0.0018),
     font.main=1, xaxt='n')
  ax2()
  plot(ora_inlandD,col='#CC9900',fill='#CC990022', add=TRUE)
  plot(dk_kde, lty=2, fill=NA, add=T, scalefactor=2)
  legend('topright',bty='n',lty=c(1,1,2),col=c('blue','#CC9900',1),
         legend=c('Coastal pots','Inland pots','Sh. middens'))

plot(MCr.as.MCd(I_d13C.bulk),fill='#CC990022', col='#CC9900',
     main=expression("Bulk "^{13}*"C"), ylab=d13Cpm,xaxt='n',
     ylim=c(-31,-21),  xlim=c(-4500,-2500))
  ax2()
  plot(MCr.as.MCd(C_d13C.bulk), add=TRUE, fill='#0000AA11', col='#0000AA')

plot(MCr.as.MCd(I_d15N.bulk),fill='#CC990022', col='#CC9900',
     main=expression("Bulk "^{15}*"N"), ylab=d15Npm,xaxt='n', ylim=c(6,10),
     xlim=c(-4500,-2500))
  ax2()
  plot(MCr.as.MCd(C_d15N.bulk), add=TRUE, fill='#0000AA11', col='#0000AA')

plot(MCr.as.MCd(I_d13C.16.0), main=expression("C"*""[16:0]),  ylab=d13Cpm,
     xaxt='n', ylim=c(-35,-20), fill='#CC990022', col='#CC9900',
```

```
     xlim=c(-4500,-2500))
  ax2()
  plot(MCr.as.MCd(C_d13C.16.0),add=TRUE,fill='#0000AA11', col='#0000AA')

plot(MCr.as.MCd(I_d13C.18.0),main=expression("C"*""[18:0]),
     ylab=d13Cpm,xaxt='n',ylim=c(-35,-20), fill='#CC990022', col='#CC9900',
     xlim=c(-4500,-2500))
  ax2()
  plot(MCr.as.MCd(C_d13C.18.0), add=TRUE,fill='#0000AA11', col='#0000AA')

plot(MCr.as.MCd(C_bigD), main=BD,ylim=c(-6,1),
     ylab=expression(Delta * ""^{13} * "C / ‰"),xaxt='n',fill='#0000AA11',
     col='#0000AA',xlim=c(-4500,-2500))
  ax2()
  plot(MCr.as.MCd(I_bigD), add=TRUE, fill='#CC990022', col='#CC9900')

plot(MCr.as.MCd(C_aq), ylim=c(0,100),fill='#0000AA11', col='#0000AA',
     main='Aquatic',ylab='Presence %',font.main=1,xaxt='n',xlim=c(-4500,-2500))
  ax2()
  plot(MCr.as.MCd(I_aq), add=TRUE,fill='#CC990022', col='#CC9900')

plot(MCr.as.MCd(C_da), ylim=c(0,100),fill='#0000AA11', col='#0000AA',
     main='Dairy',ylab='Presence %',font.main=1,xaxt='n',xlim=c(-4500,-2500))
  ax2()
  plot(MCr.as.MCd(I_da),add=TRUE,fill='#CC990022', col='#CC9900')

plot(MCr.as.MCd(C_pl), ylim=c(0,100),fill='#0000AA11', col='#0000AA',
     main='Plant',ylab='Presence %', font.main=1, xaxt='n',
     xlim=c(-4500,-2500))
  ax2()

mtext('Cal. BC',side=1,line=0.6,adj=0.53,outer=TRUE,cex=0.75)
par(fig=c(0, 1, 0, 1), oma=c(0, 0, 0, 0), mar=c(0, 0, 0, 0), new=TRUE)
plot(0, 0, type='n', bty='n', xaxt='n', yaxt='n')
legend('topright',xpd=TRUE, ncol=2, lty=1, col=c('#CC9900','#0000AA'),legend=c('Inland','Coastal'),bty=
```



Investigating spatial autocorrelation in presence and absence data

The presence / absence of taxa can be used to investigate the effect of spatial distance on the composition of the assemblages at the site. The results indicate very little spatial structure to the composition of the shell middens.

```
zoop<-read_xlsx(dkURL, sheet='Shellfish taxa present') |> as.data.frame()
zoop<-zoop[1:44,-2]
rownames(zoop)<-zoop[,1]
zoop<-t(as.matrix(zoop[,-1]))

require(vegan)
require(ecodist)
```

```
# Extract locations from dates sheet (removing problematic ones)
S<-rownames(zoop)
dsi<-read_xlsx(dkURL, sheet='Sites and dataset index') |> as.data.frame()
sites<-unique(dsi[dsi$`Site name` %in%
                        S,c('Site name','Longitude','Latitude')])
rownames(sites)<-sites[,1]
sites<-sites[,-1]
sites[,1]<-as.numeric(sites[,1]); sites[,2]<-as.numeric(sites[,2])

# define function for the haversine formula
# (This is to avoid co-ordinate conversion etc)
```

```r
haversine<-function (lat1, lat2, lon1, lon2) {
  if (lat1==lat2 & lon1==lon2) D <- 0 else {
   rlat1 <- lat1 * (pi/180)
   rlat2 <- lat2 * (pi/180)
   rlon <- (lon2-lon1) * (pi/180)
   D<-60*(180/pi)*acos(sin(rlat1)*sin(rlat2)+cos(rlat1)*cos(rlat2)*cos(rlon))
   D<-D*1852/1000
  }
  return(D)
}

# Work out which sites mutually occur
SZ<-S[S %in% rownames(sites)]
zoolocs<-sites[SZ,]

#Function to make a table of pairwise distances
GCtable<-function(x, latlong=c('Long.','Lat.')) {
  xlocs<-x[,latlong]
  out<-matrix(0,nrow=nrow(xlocs),ncol=nrow(xlocs))
  for (i in 1:nrow(xlocs))
    for (j in 1:nrow(xlocs))
      if(i!=j)
        out[i,j]<-haversine(xlocs[i,1],xlocs[j,1],xlocs[i,2],xlocs[j,2])
  return(as.dist(out))
}

#Apply haversine distance to each pair of sites
zoo_GC<-GCtable(zoolocs, latlong=c('Longitude','Latitude'))

# Convert coerse the data to numeric datatype if not already
dz<-dim(zoop)
zoop<-matrix(as.numeric(zoop))
dim(zoop)<-dz
rownames(zoop)<-S

# Ecological distance
zoo_jac<-vegdist(zoop[SZ,], 'jaccard')

# Results of mantel tests:
vegan::mantel(zoo_GC, zoo_jac)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## vegan::mantel(xdis = zoo_GC, ydis = zoo_jac)
##
## Mantel statistic r: -0.02648
##       Significance: 0.625
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0976 0.1281 0.1513 0.1758
```

```
## Permutation: free
## Number of permutations: 999
```

```
# Marine mammal mantel
# .csv files containing pre-processed data have been included to reduce the
# amount of data wrangling required

mm<-read.csv('dk_marine_mam.csv',row.names = 1)
mlocs<-mm[,c('Long.','Lat.')]
GC_mm<-matrix(0,nrow=nrow(mlocs),ncol=nrow(mlocs))
for (i in 1:nrow(mlocs)) for (j in 1:nrow(mlocs)) if(i!=j)  GC_mm[i,j]<-haversine(mlocs[i,1],mlocs[j,1]
mm_GC<-as.dist(GC_mm)
mm_jac<-vegdist(mm[,-c(1:4)], 'jaccard')
#summary(mm_jac) mean 0.76
vegan::mantel(mm_GC, mm_jac)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## vegan::mantel(xdis = mm_GC, ydis = mm_jac)
##
## Mantel statistic r: 0.04373
##       Significance: 0.142
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0519 0.0675 0.0800 0.0914
## Permutation: free
## Number of permutations: 999
```

```
# Fish mantel
fish<-read.csv('dk_fish.csv', row.names = 1)
fish_GC<-GCtable(fish)
fish_jac<-vegdist(fish[,-c(1:4)], 'jaccard')
summary(fish_jac)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.6667  0.8235  0.7852  0.9412  1.0000
```

```
vegan::mantel(fish_GC, fish_jac)
```

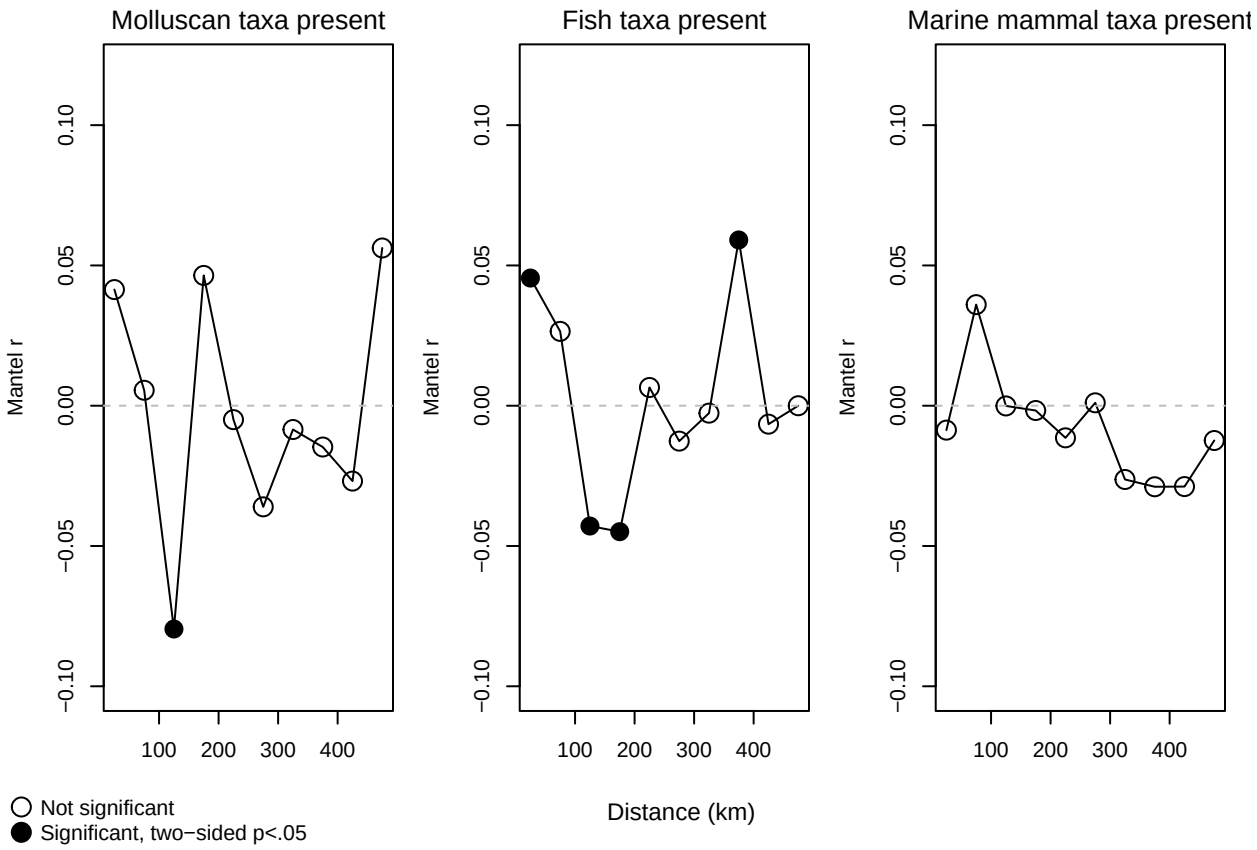
```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## vegan::mantel(xdis = fish_GC, ydis = fish_jac)
##
## Mantel statistic r: 0.02578
##       Significance: 0.211
##
## Upper quantiles of permutations (null model):
```

```
##    90%    95%  97.5%    99%
## 0.0414 0.0578 0.0683 0.0799
## Permutation: free
## Number of permutations: 999
```

```
# Mantel correlograms
brks<-c(seq(0,500,50))
mgram_mm<-  ecodist::mgram(mm_jac,    mm_GC, breaks = brks)
mgram_fish<-ecodist::mgram(fish_jac,fish_GC, breaks = brks)
mgram_zoo <-ecodist::mgram(zoo_jac, zoo_GC, breaks = brks)
```

The following script plots the Mantel correlograms.

```
#pdf('~/Desktop/dk_mantel.pdf',width=10.5, height=3)

par(mfrow=c(1,3), mar=c(2,4,2,1), omi=c(0.5,0,0,0))
plot(mgram_zoo,  ylim=c(-0.1,0.12), ylab='Mantel r',
     main='Molluscan taxa present',font.main=1)
   plot(mgram_fish, ylim=c(-0.1,0.12), ylab='Mantel r',
        main='Fish taxa present',font.main=1)
   plot(mgram_mm,   ylim=c(-0.1,0.12), ylab='Mantel r',
        main='Marine mammal taxa present',font.main=1)
   mtext('Distance (km)', outer=TRUE, adj=0.55, side=1, line=1.5, cex=0.75)

par(fig=c(0, 1, 0, 1), oma=c(0, 0, 0, 0), mar=c(0, 0, 0, 0), new=TRUE)
plot(0, 0, type='n', bty='n', xaxt='n', yaxt='n')
legend('bottomleft', pch=c(1,19),pt.cex=2, bty='n',
       legend=c('Not significant','Significant, two-sided p<.05'))
```




```
#dev.off()
```

Part III Additional data analysis

This section demonstrates some additional analyses that can be performed using the data in the database.

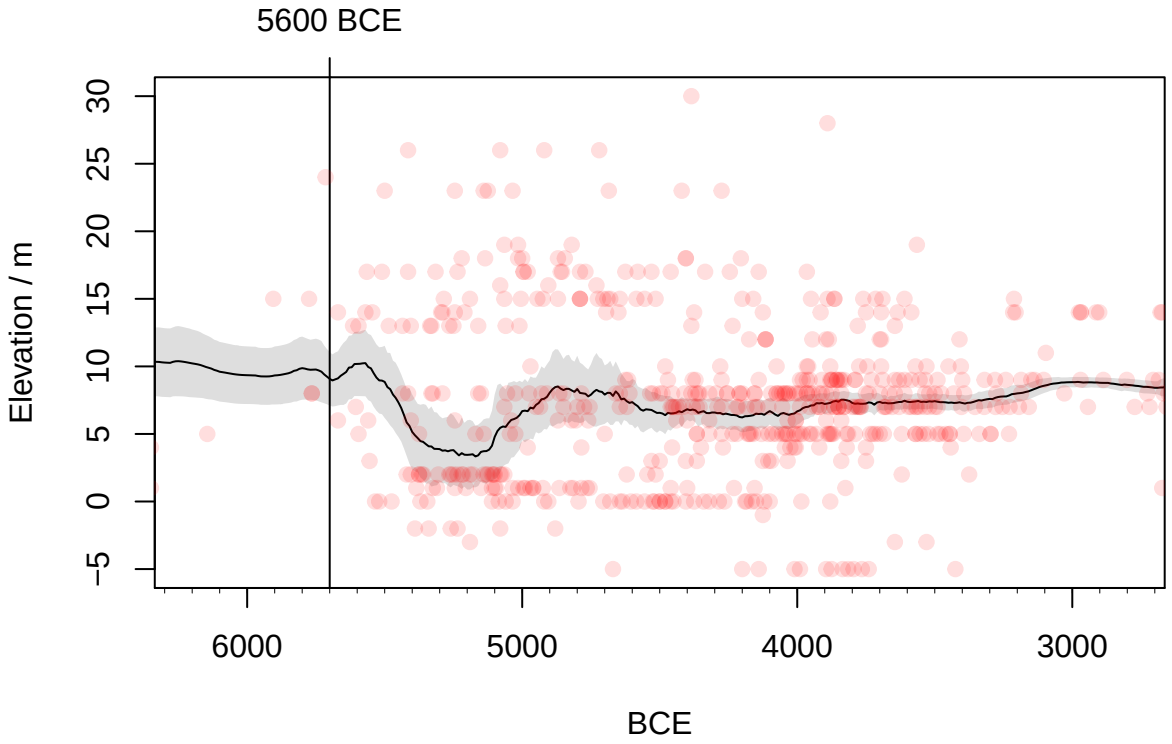
Elevation of sites versus time

Somewhat surprisingly there is no clear pattern in the elevation of sites, although no very low-lying sites are known from before ca. 5700 BC reflecting the loss of the Early Holocene coastline.

```
library(elevatr) # Requres a connection to INTERNET
dk_data$elevation<-get_elev_point(
  data.frame(x=dk_data$Longitude, y=dk_data$Latitude), prj=4326,
  src = 'aws')$elevation
ele_scatter_data<-dk_data[which(!is.na(dk_data$`Radiocarbon age (BP)`)),
        c("Radiocarbon age (BP)","Radiocarbon error","Curve","elevation")]
ele_scatter<-MCrunmed(x=rowcal(dl=ele_scatter_data[,1:3]),
                      y=ele_scatter_data$elevation,k=21) |> MCr.as.MCd()
```

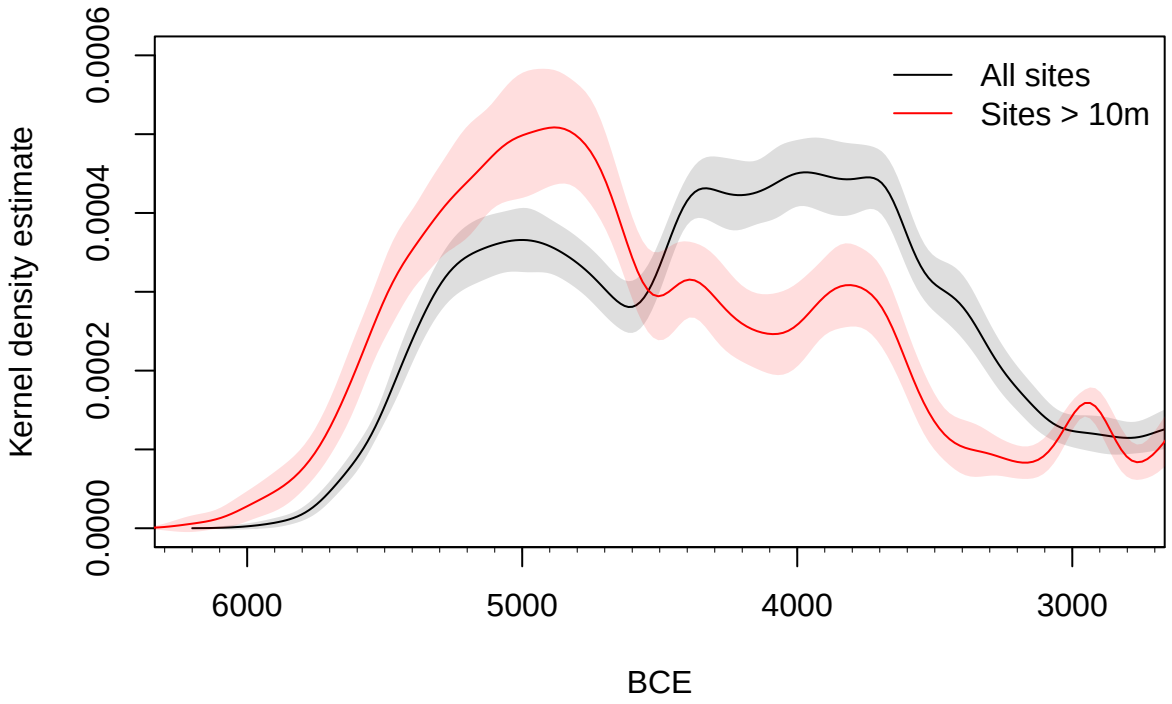
```
## ==============================================================================
```

```
plot(ele_scatter, xaxt='n', xlim=c(-6200,-2800), ylim=c(-5,30),
     ylab='Elevation / m', xlab='BCE')
ax()
abline(v=-5700); axis(3,-5700,'5600 BCE')
points(findmedian(rowcal(dl=ele_scatter_data[,1:3])),ele_scatter_data$elevation,
       pch=19,col='#FF000022')
```



A density model of sites on higher ground can be calculated as follows:

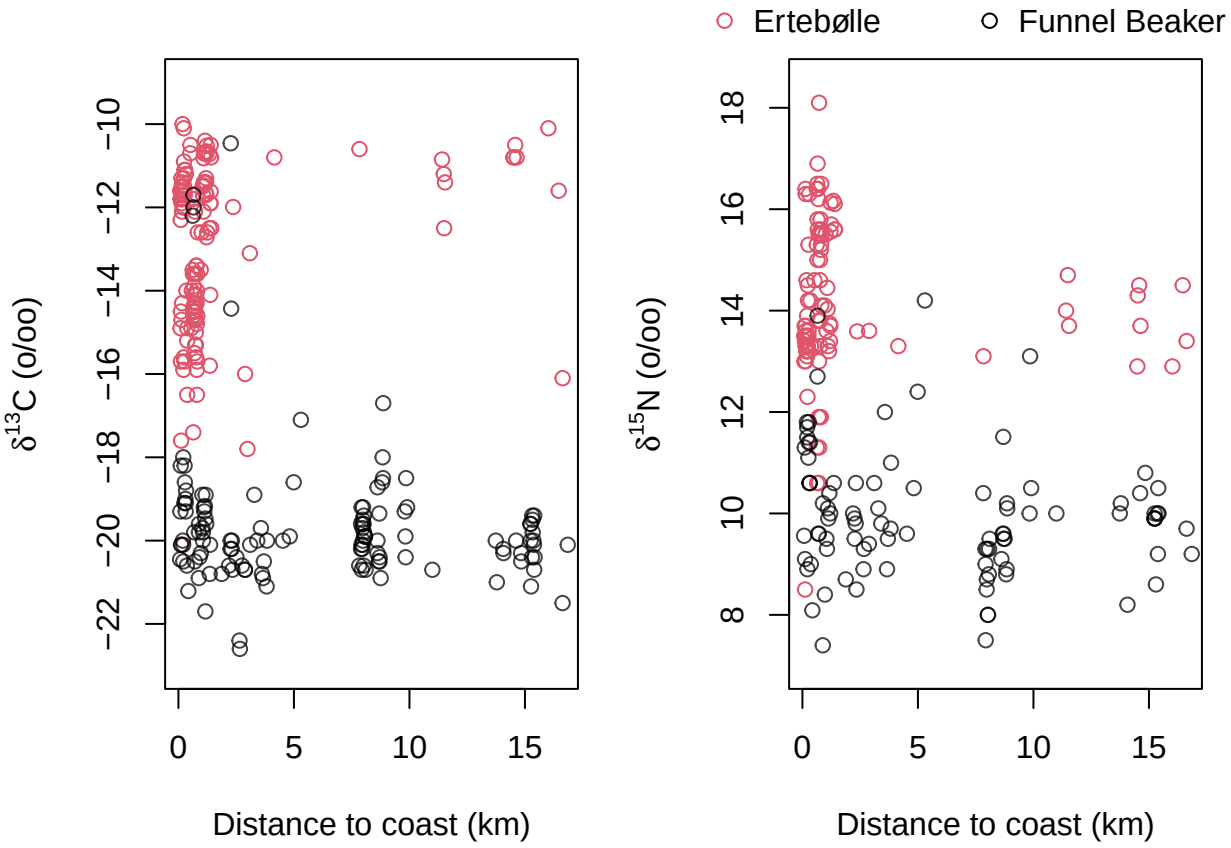
```
dkH<-MCdensity(dl=dk_data[which(dk_data$elevation>10),
                  c('Radiocarbon age (BP)','Radiocarbon error','Curve')],bw=75)

plot(dk_kde,xlim=c(-6200,-2800),ylim=c(0,0.0006),ylab='Kernel density estimate',
     xlab='BCE',xaxt='n')
  ax()
  plot(dkH, add=TRUE, col='red', fill='#FF000022')
  legend('topright',bty='n',lty=c(1,1),col=c(1,'red'),
         legend=c('All sites','Sites > 10m'))
```



Distance to coast vs stable isotope results

There are surprisingly few patterns when we consider the distance to coast.

```
# Jitter data slightly to improve visibility
jiso<-hiso
jiso$`Distance to coast (km)`<-jitter(jiso$`Distance to coast (km)`,
                                      amount = 0.1)

par(mfrow=c(1,2),mar=c(4.1,4.5,2,1),omi=c(0,0,0,0))
plot(jiso[jiso$Culture=='Ertebølle',c('Distance to coast (km)',
                                      ' 13C (‰, VPDB)')],
     ylim=c(-23,-9), ylab=d13Cpm, col=2)
points(jiso[jiso$Culture =='Funnel Beaker',c('Distance to coast (km)',
                                             ' 13C (‰, VPDB)')],col='#000000BB')
plot(jiso[jiso$Culture=='Ertebølle',c('Distance to coast (km)',
                                      ' 15N (‰, AIR)')],
     ylim=c(7,18.5),ylab=d15Npm,col=2)
points(jiso[jiso$Culture =='Funnel Beaker',c('Distance to coast (km)',
                                             ' 15N (‰, AIR)')],col='#000000BB')
par(fig=c(0, 1, 0, 1), oma=c(0, 0, 0, 0), mar=c(0, 0, 0, 0), new=TRUE)
plot(0, 0, type='n', bty='n', xaxt='n', yaxt='n')
legend('topright', ncol=2, pch=1, col=c(2,1),
       legend=c('Ertebølle','Funnel Beaker'), bty='n')
```

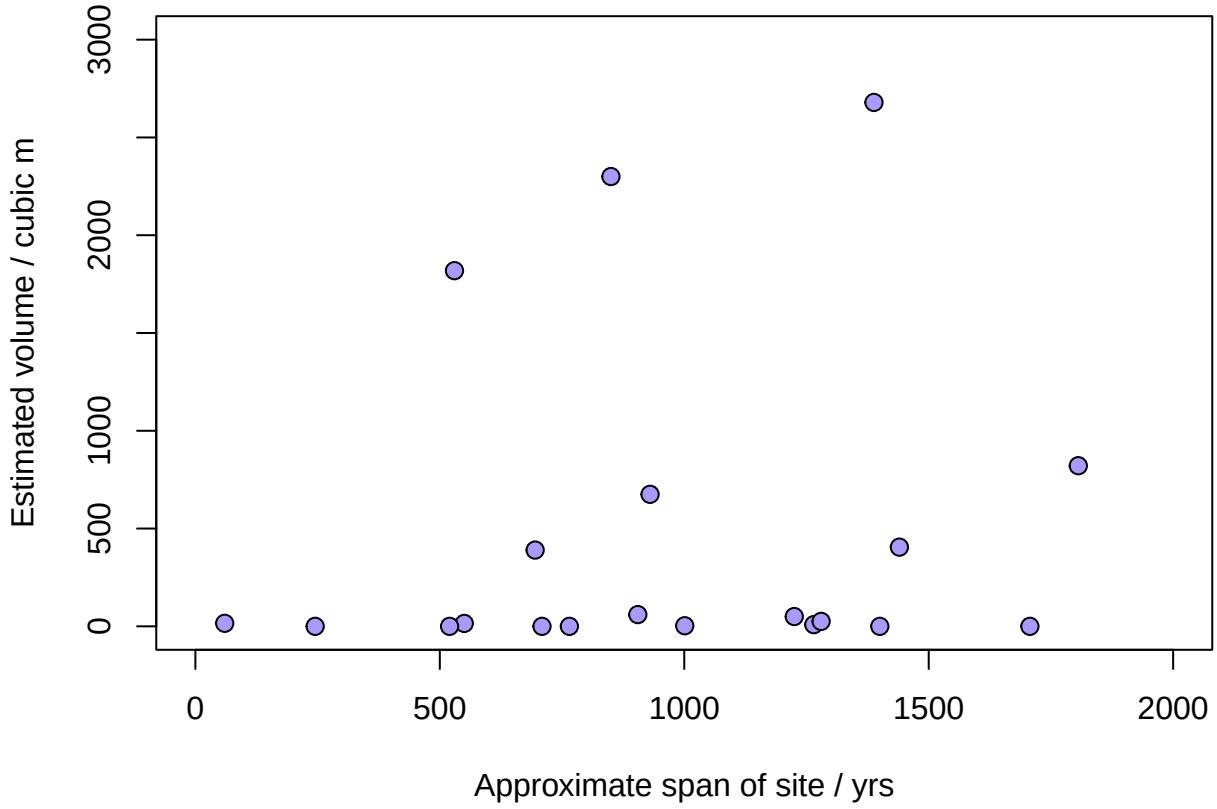


##Size of middens versus their span

There is no clear relationship between these parameters aside from the fact that the larger middens (> 300 cubic m) are known to be occupied for longer than 500 years.

```
dk_data$median<-findmedian(rowcal(dl=dk_data[,
                    c('Radiocarbon age (BP)','Radiocarbon error','Curve')]))

size<-read_xlsx(dkURL,
                 sheet='Shell midden size and volumes') |> as.data.frame()

size$meanvol<-rowMeans(size[,c('Minimum volume','Maximum volume')],
                       na.rm=TRUE)

# Count dates per site (to limit span analysis to well dated sites)

freqs<-as.data.frame(table(dk_data$`Site name`))
colnames(freqs)<-c('Site name','Ndates')

freqs$max=NA
freqs$min=NA
for(N in 1:nrow(freqs))
  freqs$min[N]<-min(dk_data[dk_data$`Site name`==freqs[N,1],'median'])
for(N in 1:nrow(freqs))
  freqs$max[N]<-max(dk_data[dk_data$`Site name`==freqs[N,1],'median'])

par(mfrow=c(1,1),mar=c(4,4,2,1))
size<-merge(size,freqs)
```

```
# sites with 3 or more dates
multidated<-size[which(size$Ndates>2),]
plot(multidated$max-multidated$min,multidated$meanvol,pch=21,bg='#AA99FF',
     xlab='Approximate span of site / yrs',
     ylab='Estimated volume / cubic m',
     xlim=c(0,2000), ylim=c(0,3000),cex=1.2)
```



KDEs by date of publication

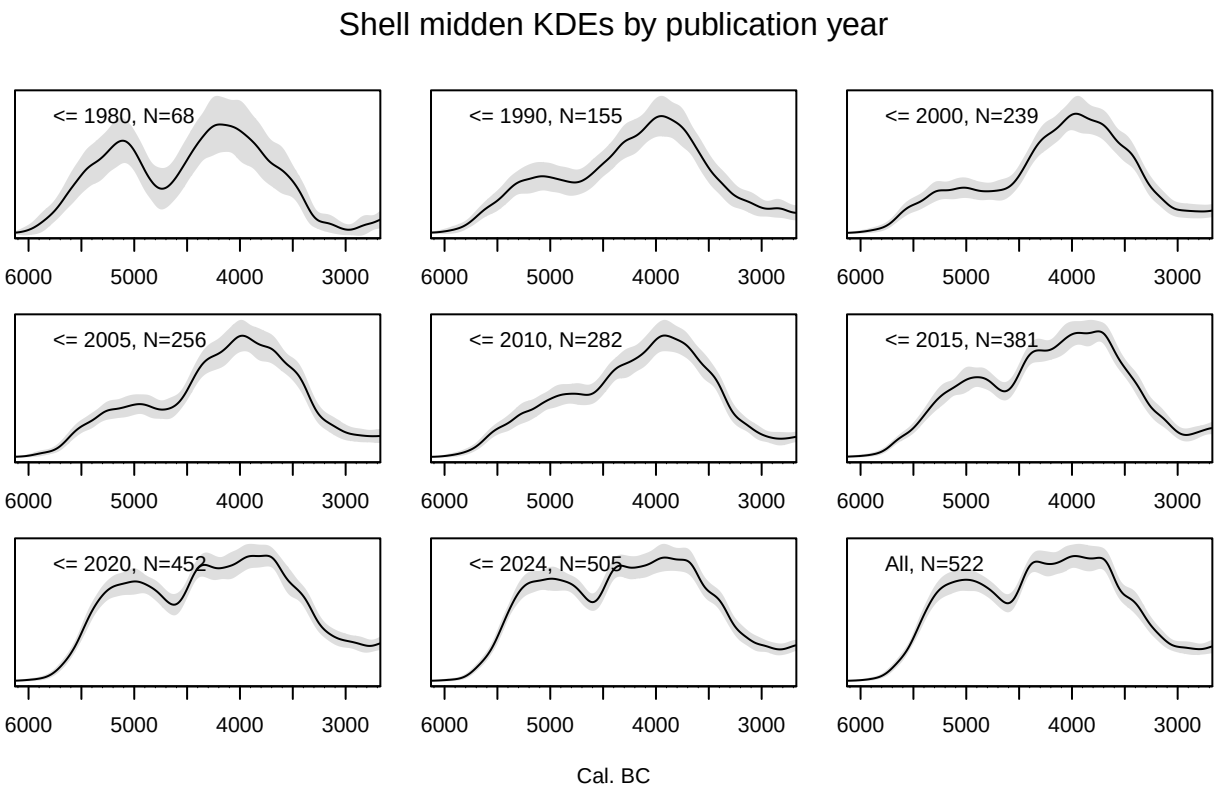
The following plot demonstrates how the accumulation of data over the years has influenced our total knowledge about the temporal density of material from shell middens.

```
# Extract data
dop<-dk_data[which(is.na(dk_data$Natural)), c("sitecode",
  "Radiocarbon age (BP)","Radiocarbon error","Curve","Date of publication")]
dop$`Date of publication`<-as.numeric(dop$`Date of publication`)
# Set unpublished date to the year 3000
dop$`Date of publication`[is.na(dop$`Date of publication`)]<-3000
```

```
dk1980<-phasedensity(dl=dop[which(dop$`Date of publication`<=1980),1:4],bw=75)
dk1990<-phasedensity(dl=dop[which(dop$`Date of publication`<=1990),1:4],bw=75)
dk2000<-phasedensity(dl=dop[which(dop$`Date of publication`<=2000),1:4],bw=75)
dk2005<-phasedensity(dl=dop[which(dop$`Date of publication`<=2005),1:4],bw=75)
dk2010<-phasedensity(dl=dop[which(dop$`Date of publication`<=2010),1:4],bw=75)
dk2015<-phasedensity(dl=dop[which(dop$`Date of publication`<=2015),1:4],bw=75)
```

```
dk2020<-phasedensity(dl=dop[which(dop$`Date of publication`<=2020),1:4],bw=75)
dk2024<-phasedensity(dl=dop[which(dop$`Date of publication`<=2024),1:4],bw=75)
```

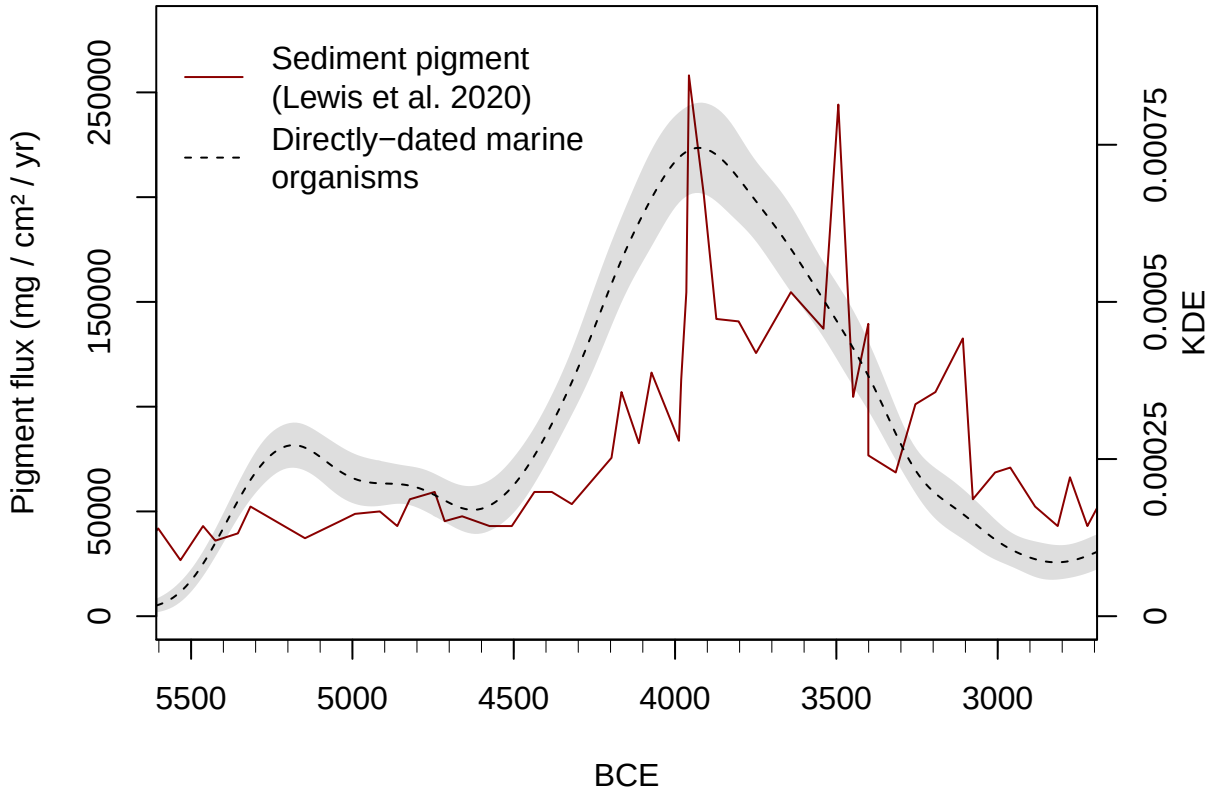
```
par(mfrow=c(3,3),mar=c(2,1,1,1),omi=c(0.5,0,0.5,0))
plot(dk1980, xlim=c(-6000,-2800),xaxt='n',yaxt='n')
  ax()
  legend('topleft',pch=NA,bty='n',paste0('\u2264 1980, N=',nrow(dop[
    which(dop$`Date of publication`<=1980),])))
plot(dk1990, xlim=c(-6000,-2800),xaxt='n',yaxt='n')
  ax()
  legend('topleft',pch=NA,bty='n',paste0('\u2264 1990, N=',nrow(dop[
    which(dop$`Date of publication`<=1990),])))
plot(dk2000, xlim=c(-6000,-2800),xaxt='n',yaxt='n')
  ax()
  legend('topleft',pch=NA,bty='n',paste0('\u2264 2000, N=',nrow(dop[
    which(dop$`Date of publication`<=2000),])))
plot(dk2005, xlim=c(-6000,-2800),xaxt='n',yaxt='n')
  ax()
  legend('topleft',pch=NA,bty='n',paste0('\u2264 2005, N=',nrow(dop[
    which(dop$`Date of publication`<=2005),])))
plot(dk2010, xlim=c(-6000,-2800),xaxt='n',yaxt='n')
  ax()
  legend('topleft',pch=NA,bty='n',paste0('\u2264 2010, N=',nrow(dop[
    which(dop$`Date of publication`<=2010),])))
plot(dk2015, xlim=c(-6000,-2800),xaxt='n',yaxt='n')
  ax()
  legend('topleft',pch=NA,bty='n',paste0('\u2264 2015, N=',nrow(dop[
    which(dop$`Date of publication`<=2015),])))
plot(dk2020, xlim=c(-6000,-2800),xaxt='n',yaxt='n')
  ax()
  legend('topleft',pch=NA,bty='n',paste0('\u2264 2020, N=',nrow(dop[
    which(dop$`Date of publication`<=2020),])))
plot(dk2024, xlim=c(-6000,-2800),xaxt='n',yaxt='n')
  ax()
  legend('topleft',pch=NA,bty='n',paste0('\u2264 2024, N=',nrow(dop[
    which(dop$`Date of publication`<=2024),])))
plot(dk_kde, xlim=c(-6000,-2800),xaxt='n',yaxt='n')
  ax()
  legend('topleft',pch=NA,bty='n',paste0('All, N=',nrow(dk)))
mtext(side=1,outer=T,adj=0.5,cex=0.66,'Cal. BC', line=1)
mtext(side=3,outer=T,adj=0.5,'Shell midden KDEs by publication year', line=1)
```



Palaeo data from Lewis et al 2020

The following plot displays the total sediment pigment flux data from the Tempelkrog core (data taken from Supplementary Figure 4 in Lewis et al 2020), which is compared to the KDE of the shell midden data.

```
par(mar=c(5,4,1,4))
plot(1950-Tempelkrog$y,Tempelkrog$x,type='l',xaxt='n', xlim=c(-5500,-2800),
     ylab='Pigment flux (mg / cm² / yr)',xlab='BCE',col='darkred',
     ylim=c(0,280000))
ax(ticksize=-0.02)
plot(marD,add=T, scalefactor = 300000000, lty=2)
kdescale<-c(0,0.00025,0.0005,0.00075,0.001)
axis(4, at=kdescale*300000000, lab=kdescale)
mtext('KDE',side=4, line=2)
legend('topleft',lty=c(1,2),col=c('darkred',1),bty='n',y.intersp=1.5,
       c('Sediment pigment\n(Lewis et al. 2020)',
         'Directly-dated marine\norganisms'))
```



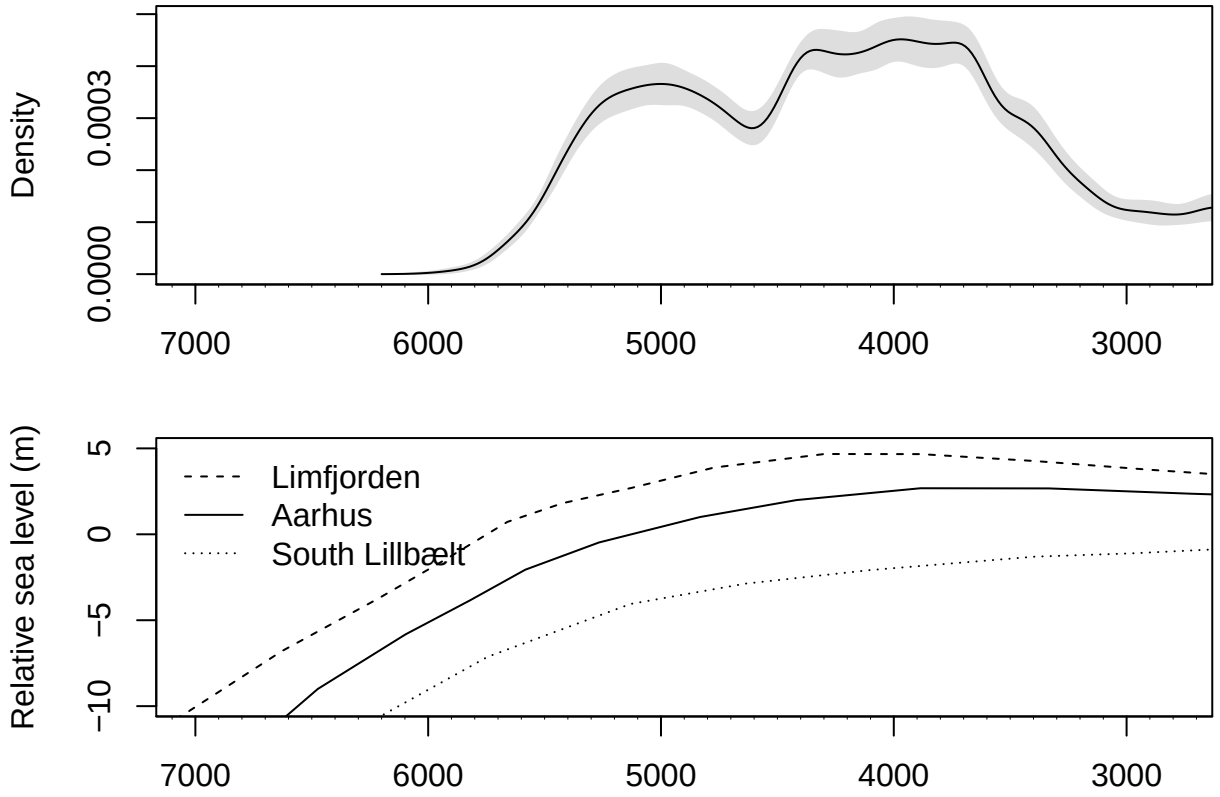

```
#lines(1950-Kilen$y     ,Kilen$x     ,col=2)
#lines(,Tempelkrog$x,col=3)
```

Sea level changes

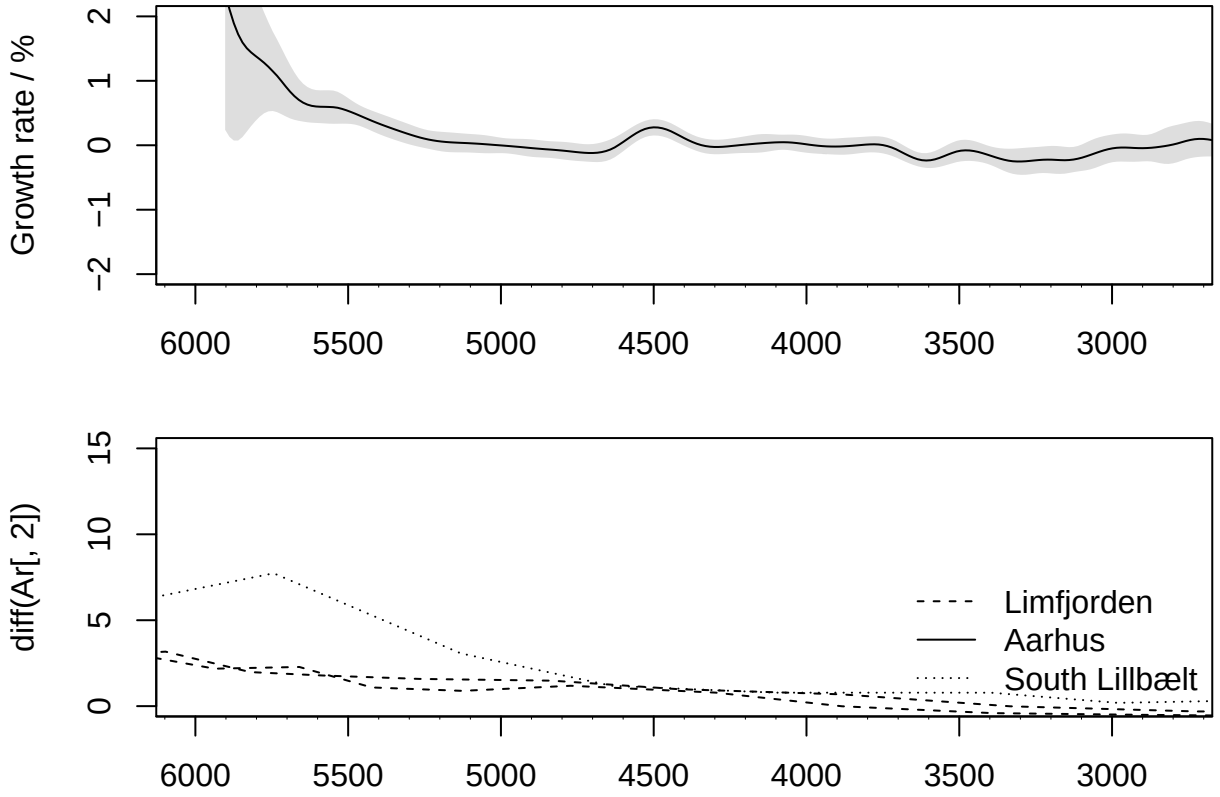
The following data are taken from the Bennike et al 2021 (https://doi.org/10.34194/geusb.v47.6530)

```
Ar<-structure(list(V1 = c(11888.5602435842, 11612.3749456285, 11355.8938668987,
11118.9865158765, 10822.7707699, 10585.7329273597, 10328.7951283167,
10091.3658112223, 9873.51022183558, 9675.35885167464, 9536.1896476729,
9377.31622444541, 9178.83862548934, 9019.76946498478, 8860.70030448021,
8662.41844280122, 8424.1409308395, 8047.67290126142, 7770.5089169204,
7532.88386254894, 7216.57242279252, 6782.10091344063, 6367.65985210962,
5835.18921270118, 5283.47107438017, 4672.83601565898, 3943.9756415833,
3313.63636363637, 2762.04871683341, 2013.61461505003, 1304.45846020009,
733.297085689434, 201.544149630276), V2 = c(-33.5784036537625,
-33.0862875163114, -32.5938451500652, -31.9024032187908, -30.913929969552,
-30.0238147020444, -28.8360156589822, -27.3498803827751, -25.6647455415398,
-23.9792844715093, -22.0941713788604, -20.2093845150065, -18.0272401043932,
-15.8444432361896, -13.6616463679861, -11.7775119617225, -9,
-5.82742496737712, -3.84525880817747, -2.0611135276207, -0.476946498477597,
1.00592648977817, 1.99244236624619, 2.67899086559374, 2.66985645933014,
2.36173336233145, 2.05165289256198, 1.74320356676816, 1.53539582427142,
1.02631578947368, 0.716561548499348, 0.309754240974337, -0.0964006089604172
)), class = "data.frame", row.names = c(NA, -33L))

Li<-structure(list(V1 = c(11651.78338408, 11156.6333188343, 10839.6041757286,
```

```
10443.1056981296, 9967.46411483254, 9471.98782079165, 8976.70726402784,
8580.33927794694, 8164.52805567638, 7887.23357981731, 7609.87385819922,
7372.70552414093, 7076.55502392345, 6721.09612875163, 6247.67290126142,
5833.88429752066, 5341.53979991301, 4849.26054806438, 4199.41278816877,
3411.76598521097, 2880.07829491083, 2289.27794693345, 1698.41235319704,
1166.65941713789, 634.775989560683, 280.230535015229, 24.1409308394977
), V2 = c(-33.0856350587212, -29.219660722053, -26.5427903436277,
-22.873858199217, -18.7095476294041, -14.3468899521531, -10.2822422792518,
-6.81198347107438, -3.73939756415833, -1.55855806872553, 0.721618094823837,
1.81040669856459, 2.69954327968682, 3.88571117877338, 4.67257503262288,
4.66572422792518, 4.26022183558069, 3.75538277511962, 3.14859721618095,
2.34085471944324, 1.83536320139191, 1.3288929969552, 0.921759460635059,
0.515604610700303, 0.308123096998695, 0.103577642453242, 0)),
class = "data.frame", row.names = c(NA, -27L))

SL<-structure(list(V1 = c(11783.3188342758, 11370.5089169204, 10917.8338408004,
10562.8969117007, 9990.03914745542, 9673.4667246629, 9297.65115267508,
8723.55371900827, 7693.84515006525, 7080.99173553719, 6587.60330578513,
6055.06742061766, 5345.19356241845, 4931.27446715964, 4339.88690735103,
3748.62983906046, 3137.73379730318, 2664.83253588517, 1935.64593301436,
1383.7973031753, 970.008699434538, 319.769464984778, -34.9064810787277
), V2 = c(-23.3483579817312, -24.8452588081775, -25.647455415398,
-25.255980861244, -23.0800347977381, -21.0985210961287, -18.9193127446716,
-14.8559699869508, -7.12467377120487, -4.0553501522401, -2.87146585471944,
-2.08558068725532, -1.30263157894737, -1.11080904741192, -0.723249238799477,
-0.534362766420182, -0.445139190952586, -0.452968682035669, -0.26636581122227,
-0.0768268812527193, -0.0836776859504127, -0.0944432361896474,
-0.100315354501957)), class = "data.frame", row.names = c(NA,
-23L))

par(mfrow=c(2,1), mar=c(3,4,1,1))
plot(dk_kde, xlim=c(-7000,-2800),xaxt='n'); ax()
plot(1950-Ar[,1],Ar[,2], type='l',lty=1,xlim=c(-7000,-2800), xaxt='n',
     ylim=c(-10,5),ylab='Relative sea level (m)',xlab='BCE'); ax()
lines(1950-Li[,1],Li[,2], lty=2)
lines(1950-SL[,1],SL[,2], lty=3)
legend('topleft',lty=c(2,1,3), legend=c('Limfjorden','Aarhus','South Lillbælt')
       ,bty='n')
```



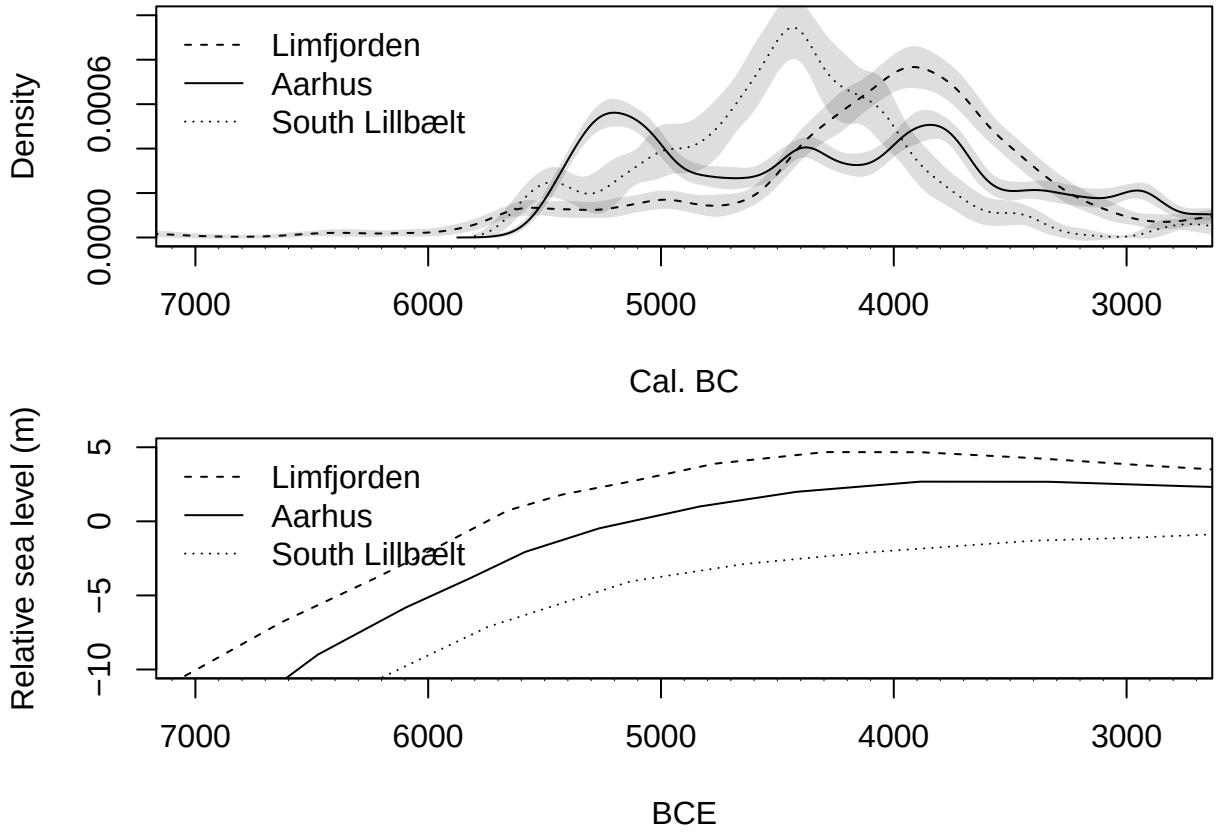

```
# Growth rate compared to rate-of-change of sea level
plot(ggr(dk_kde), xlim=c(-6000,-2800),xaxt='n'); ax()
plot(1950-Ar[-1,1],diff(Ar[,2]), type='l',lty=2,xlim=c(-6000,-2800), xaxt='n',
     ylim=c(0,15)); ax()
lines(1950-Li[-1,1],diff(Li[,2]), lty=2)
lines(1950-SL[-1,1],diff(SL[,2]), lty=3)
legend('bottomright',lty=c(2,1,3), legend=c('Limfjorden','Aarhus','South Lillbælt')
       ,bty='n')
```



These can be compared to regional archaeoloigcal density models as follows:

```
asites<-c('Dyngby I','Dyngby II','Dyngby III','Dyrholmen (natural','Fannerup D',
          'Fannerup E','Fannerup F','Flynderhage',
          'Gammelholm (natural shell bank','Handsted Ådal','Hjarnø Sund','Hjarnø Vesterhoved','Hjortholm
          'Kolind','Kolindsund (natural','Kørup Sø (natural','Lystrup Enge','Mejlgård','Nederst','Norslu
          'Vængesø III','Vængesø IV')

lbsites<-c('Ronæs Skov (redeposited shell midden/natural death assemblage)',
           'Tybrind Vig (natural shell deposit)','Boghoved, Slivsø',
           'Møllegabet II (natural shell deposit?','Selbjerg')

aD<-mixdensity(dk_data[which(dk_data$`Site name` %in% asites), c("Radiocarbon age (BP)","Radiocarbon err

liD<-mixdensity(dk_data[which(dk_data$Region_1=='Limfjord'),
                        c("Radiocarbon age (BP)","Radiocarbon error","Curve")],
                bw=75)

lbD<-mixdensity(dk_data[which(dk_data$`Site name` %in% lbsites),
                        c("Radiocarbon age (BP)","Radiocarbon error","Curve")],
                bw=75)


par(mfrow=c(2,1), mar=c(4,4,1,1))
plot(aD, xlim=c(-7000,-2800),xaxt='n',ylim=c(0,0.001)); ax()
plot(liD, add=T, lty=2)
plot(lbD, add=T, lty=3)
legend('topleft',lty=c(2,1,3), legend=c('Limfjorden','Aarhus','South Lillbælt'),
       bty='n')
```

```r
plot(1950-Ar[,1],Ar[,2], type='l',lty=1,xlim=c(-7000,-2800), xaxt='n',
     ylim=c(-10,5),ylab='Relative sea level (m)',xlab='BCE'); ax()
lines(1950-Li[,1],Li[,2], lty=2)
lines(1950-SL[,1],SL[,2], lty=3)
legend('topleft',lty=c(2,1,3), legend=c('Limfjorden','Aarhus','South Lillbælt'),
       bty='n')
```



References and notes

Bennike, O., Björck, S., Lambeck, K., et al. 2021. The Holocene sea-level history of the northern North Sea: a multi-proxy reconstruction. Geological Survey of Denmark and Greenland Bulletin 47, 1–22. https://doi.org/10.34194/geusb.v47.6530

Lewis, J.P., Ryves, D.B., Rasmussen, P. et al. Marine resource abundance drove pre-agricultural population increase in Stone Age Scandinavia. Nat Commun 11, 2006 (2020). https://doi.org/10.1038/s41467-020-15621-1

McLaughlin, T.R. 2019. On applications of space-time modelling with open-source 14C age calibration. Journal of Archaeological Method and Theory 26, 479–501. https://doi.org/10.1007/s10816-018-9381-3

McLaughlin, T. R., Gómez-Puche, M., Cascalheira, J., Bicho, N. and Fernández-López de Pablo, J. 2021. Late Glacial and Early Holocene human demographic responses to climatic and environmental change in Atlantic Iberia. Philosophical Transactions of the Royal Society B 376, 20190724. https://dx.doi.org/10.1098/rstb.2019.0724

This script by Rowan McLaughlin, Hamilton Institute, Maynooth University, January 2025. email: rowan.mclaughlin@mu.ie