# Inferences from large sets of radiocarbon dates: software and method - Electronic Supplementary Material

E.Crema & A.Bevan

13 August 2020

## Introduction

This Rmarkdown document includes R scripts for generating all figures in the article *Inferences from large sets of radiocarbon dates: software and method*, as well as supplementary notes, figures and scripts.

## Setup

All analyses and figures require the installation of the *rcarbon* package version 1.4.0 and additional data supplied in the ESM. To install *rcarbon* type the following command in the R console:

```r
install.packages("rcarbon")
```

Once the package is downloaded and installed, load *rcarbon* by typing:

```r
library(rcarbon)
```

and read/load all relevant datasets:

```r
# IntCal20 Calibration Curve from Reimer et al 2020
intcal20 <- read.csv('http://intcal.org/curves/intcal20.14c',
                     encoding="UTF-8",skip=11,header=F)
colnames(intcal20) <- c("BP","CRA","Error","D14C","Sigma")

# ShCal20 Calibration Curve from Hogg et al 2020
shcal20 <- read.csv('http://intcal.org/curves/shcal20.14c',
                    encoding="UTF-8",skip=11,header=F)
colnames(shcal20) <- c("BP","CRA","Error","D14C","Sigma")

# Eastern Meditteranean 14C Dataset from Roberts et al 2018
data(emedyd) # this is available within the rcarbon package
emedyd <- subset(emedyd, CRA<=17000&CRA>=8000)
southLevant <- subset(emedyd,Region=='1') #

# Sahara 14C Dataset from Manning and Timpson 2014:
sahara <- read.csv('./data/Manning_and_Timpson2014.csv',header=TRUE)
sahara <- subset(sahara,Yrs.BP<=17000&Yrs.BP>=8000)

# Brazilian 14C Dataset from Bueno et al 2013:
brazil <- read.csv('./data/Bueno_etal_2013.csv',header=TRUE)
brazil <- subset(brazil,OccCRA<=17000&OccCRA>=8000)
```

```
# EUROEVOL 14C Datasets from Manning et al 2016:
data(euroevol) # this is available within the rcarbon package
data(ewdates) # subset of dates in England and Wales
data(ewowin) # window of analysis for the English and Welsh subset
```

# Main Article Figures

## Figure 1

The figure below displays the consequence of thinning and binning in the summation of radiocarbon dates. Panel *a* shows a summed probability distribution of dates from Grime's Graves in Britain (n=130 dates) as well as smoothed version with a rolling mean of 50 years and three example dates. Panel *b* shows the smoothed raw density with of a randomly 'thinned' dataset of just ten dates from the same site, while panels *c* to *e* show binned datasets at clustering cut-offs of h=50, 100 and 200 years respectively.

```
grimes <- euroevol[grepl("S2072", euroevol$SiteID),]
grimesc <- calibrate(x=grimes$C14Age, errors=grimes$C14SD, ids=grimes$C14ID,
                     normalised=FALSE,verbose = FALSE)
workingstartBP <- 6000
workingendBP <- 2000
gspd0 <- spd(x=grimesc, timeRange=c(workingstartBP,workingendBP),
            datenormalised=TRUE,verbose = FALSE)
threedates <- grimes[grimes$LabCode %in% c("BM-1022","BM-1033","BM-1066"),]
threedatesc <- calibrate(x=threedates$C14Age, errors=threedates$C14SD, ids=threedates$C14ID,
                         normalised=FALSE,verbose=FALSE)


## Thinned version
n <- 10
inds <- thinDates(ages=grimes$C14Age,errors=grimes$C14SD,grimes$SiteID,size=n,method="random",
                 seed=99)
grimesthin <- grimes[inds,]
grimesthinc <- calibrate(x=grimesthin$C14Age, errors=grimesthin$C14SD, ids=grimesthin$C14ID,
                         normalised=FALSE,verbose = FALSE)
gspdthin <- spd(x=grimesthinc, timeRange=c(workingstartBP,workingendBP), datenormalised=TRUE,
               verbose = FALSE)


## Binned version
bins50 <- binPrep(sites=grimes$SiteID, ages=grimes$C14Age, h=50)
bins100 <- binPrep(sites=grimes$SiteID, ages=grimes$C14Age, h=100)
bins200 <- binPrep(sites=grimes$SiteID, ages=grimes$C14Age, h=200)
gspd50 <- spd(x=grimesc, bins=bins50, timeRange=c(workingstartBP,workingendBP),
             datenormalised=TRUE,verbose = FALSE)
gspd100<- spd(x=grimesc, bins=bins100, timeRange=c(workingstartBP,workingendBP),
             datenormalised=TRUE,verbose = FALSE)
gspd200 <- spd(x=grimesc, bins=bins200, timeRange=c(workingstartBP,workingendBP),
              datenormalised=TRUE,verbose = FALSE)


# plot
layout(matrix(c(1,1,2,3,4,5), 3, 2, byrow=TRUE), widths=c(3,3), heights=c(3,2,2))

# Panel a
```

```r
par(mar=c(4, 4, 1, 0.5))
plot(gspd0, calendar="BCAD", fill.p="chocolate1")
plot(gspd0, runm=50, calendar="BCAD", add=TRUE, fill.p=NA, border="chocolate4")
plot(threedatesc[1],add=TRUE,calendar='BCAD', col='black')
plot(threedatesc[2],add=TRUE,calendar='BCAD', col='black')
plot(threedatesc[3],add=TRUE,calendar='BCAD', col='black')

legend("topright", bty="n", col=c("chocolate1","chocolate4","black"),
       legend=c("raw SPD", "smoothed (50-yr running mean)", "3 example dates"),
       lwd=c(5,1,5), cex=0.8)
text(-3800, 0.02, "a", font=2)


# Panel b
par(mar=c(0.5, 3, 0.5, 0.5))

plot(gspd0, xaxt="n", yaxt="n", runm=50, fill.p=NA, border="white",
     spdnormalised=TRUE)
plot(gspdthin, xaxt="n", yaxt="n", runm=50, fill.p="aquamarine",
     spdnormalised=TRUE, add=TRUE)
plot(gspd0, xaxt="n", yaxt="n", runm=50, fill.p=NA, border="chocolate4",
     spdnormalised=TRUE, add=TRUE)
legend("topright", bty="n", col=c("chocolate4","aquamarine"),
       legend=c("original SPD",
                paste("Thinned to ", nrow(grimesthin), " random dates", sep="")),
       lwd=c(1,5), cex=0.7)
text(BCADtoBP(-3800), 0.00015, "b", font=2)


# Panel c
plot(gspd0, xaxt="n", yaxt="n", runm=50, fill.p=NA, border="white",
     spdnormalised=TRUE)
plot(gspd50, xaxt="n", yaxt="n", runm=50, fill.p="bisque3",
     spdnormalised=TRUE, add=TRUE)
plot(gspd0, xaxt="n", yaxt="n", runm=50, fill.p=NA, border="chocolate4",
     spdnormalised=TRUE, add=TRUE)
legend("topright", bty="n", col=c("chocolate4","bisque3"),
       legend=c("original SPD", "50 year bins"), lwd=c(1,5), cex=0.7)
text(BCADtoBP(-3800), 0.00015, "c", font=2)


# Panel d
plot(gspd0, xaxt="n", yaxt="n", runm=50, fill.p=NA, border="white",
     spdnormalised=TRUE)
plot(gspd100, xaxt="n", yaxt="n", runm=50, fill.p="bisque3",
     spdnormalised=TRUE, add=TRUE)
plot(gspd0, xaxt="n", yaxt="n", runm=50, fill.p=NA, border="chocolate4",
     spdnormalised=TRUE, add=TRUE)
legend("topright", bty="n", col=c("chocolate4","bisque3"),
       legend=c("original SPD", "100 year bins"), lwd=c(1,5), cex=0.7)
text(BCADtoBP(-3800), 0.00015, "d", font=2)


# Panel e
plot(gspd0, xaxt="n", yaxt="n", runm=50, fill.p=NA, border="white",
     spdnormalised=TRUE)
plot(gspd200, xaxt="n", yaxt="n", runm=50, fill.p="bisque3",
```
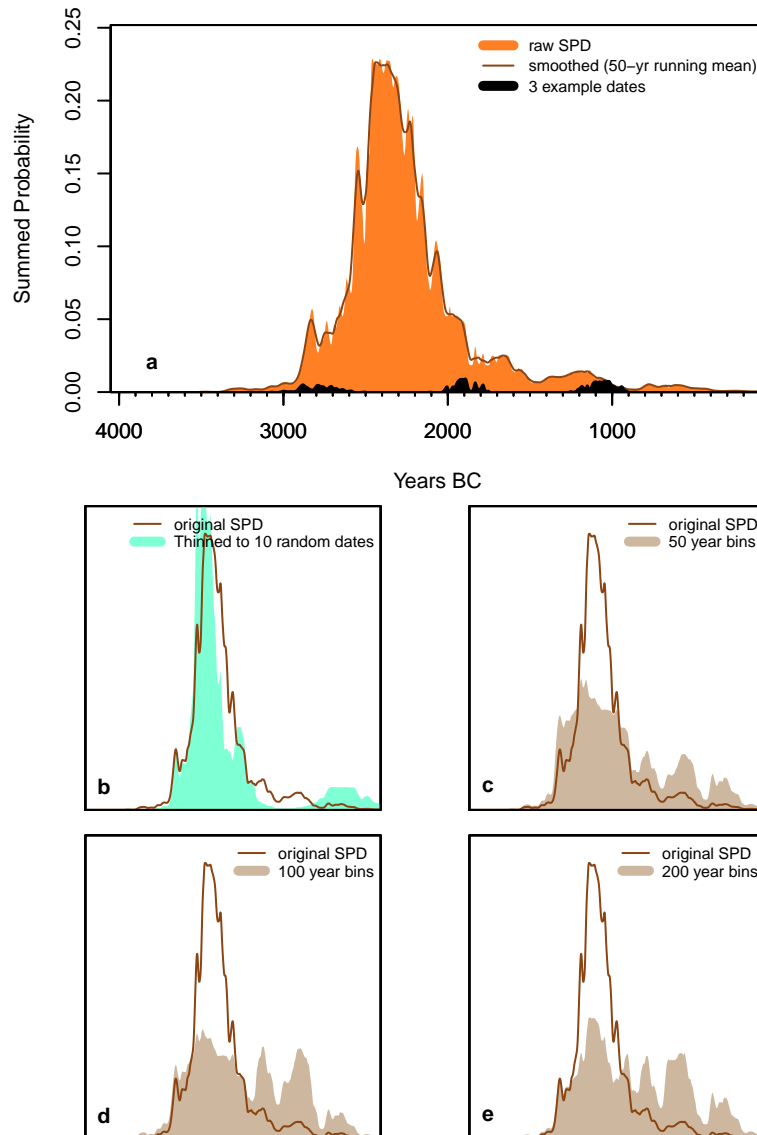
```
        spdnormalised=TRUE, add=TRUE)
plot(gspd0, xaxt="n", yaxt="n", runm=50, fill.p=NA, border="chocolate4",
        spdnormalised=TRUE, add=TRUE)
legend("topright", bty="n", col=c("chocolate4","bisque3"),
        legend=c("original SPD", "200 year bins"), lwd=c(1,5), cex=0.7)
text(BCADtoBP(-3800), 0.00015, "e", font=2)
```



**Figure 2**

The code below showcase the impact of normalised and unnormalised calibration on individual dates (panels *a* and *b*) and on summed probability distributions (panels *c* - *e*). SPDs are based on empirical data from three distinct geographic regions (Southern Levant, Sahara, and Brazil). Notice that artificial spikes matching steeper portions of the calibration curve are evident when the summation is based on normalised dates.

```
## Calibrate Dates
southlevant.x <- calibrate(southLevant$CRA,southLevant$Error,
                            normalised=F,verbose=F,calCurves='intcal20')
```

```r
sahara.x <- calibrate(sahara$Yrs.BP,sahara$STD,
                      normalised=F,verbose=F,calCurves='intcal20')
brazil.x <- calibrate(brazil$CRA,brazil$Error,
                      normalised=F,verbose=F,calCurves='shcal20')

## Minimal binning to control for sites with many dates.
southlevant.bins <- binPrep(sites=southLevant$SiteName, ages=southLevant$CRA, h=50)
sahara.bins <- binPrep(sites=sahara$Sitename, ages=sahara$Yrs.BP, h=50)
brazil.bins <- binPrep(sites=brazil$SiteName, ages=brazil$CRA, h=50)

## Summation
southlevant.spd.nnorm <- spd(southlevant.x, bins=southlevant.bins,
                             timeRange=c(16000,9000),verbose=F)
sahara.spd.nnorm <- spd(sahara.x, bins=sahara.bins,
                        timeRange=c(16000,9000),verbose=F)
brazil.spd.nnorm <- spd(brazil.x, bins=brazil.bins,
                        timeRange=c(16000,9000),verbose=F)
southlevant.spd.norm <- spd(southlevant.x, bins=southlevant.bins,
                            timeRange=c(16000,9000),verbose=F,datenormalised=T)
sahara.spd.norm <- spd(sahara.x, bins=sahara.bins,
                       timeRange=c(16000,9000),verbose=F,datenormalised=T)
brazil.spd.norm <- spd(brazil.x, bins=brazil.bins,
                       timeRange=c(16000,9000),verbose=F,datenormalised=T)

## Prepare Panels
fmat <- matrix(c(rep(1,3),rep(2,3),rep(3:5,each=2)),nrow=6,ncol=2)
layout(fmat,widths=c(0.4,0.8),heights=c(1.2,1,0.8,0.8,1,1.2))

# Panel a
x1 <- calibrate(3523,45,normalised = F,verbose=F)
plot(x1,type='auc',cex.lab=1,cex.axis=1)
title("a",line=-2)

# Panel b
x2 <- calibrate(4274,45,normalised = F,verbose=F)
plot(x2,type='auc',cex.lab=1,cex.axis=1)
title("b",line=-2)


# Panel c
par(yaxs="i")
par(xaxs="i")
par(mar=c(0,6,6,1))
plot(southlevant.spd.nnorm,fill.p=NA, xlim=c(14000,9000),
     ylim=c(0,0.25), xaxt="n",yaxt='n',cex.lab=1,cex.axis=0)
rect(x=c(9509.948,10228.445,11241.710,12690)+100,
     xright=c(9509.948,10228.445,11241.710,12690)-100,
     ybottom=rep(-100,4),ytop=rep(100,5),border=NA,col='orange')
plot(southlevant.spd.nnorm,fill.p='bisque3',ylim=c(0,0.25), xaxt="n", add=TRUE)
plot(southlevant.spd.norm,type='simple', xaxt="n",add=TRUE)
lines(intcal20[intcal20$BP<=14000 & intcal20$BP>=9000,"BP"],
      reScale(intcal20[intcal20$BP<=14000 & intcal20$BP>=9000,"CRA"])*par('usr')[4],
      col="red",lwd=2)
```

```
legend("topleft",
       legend=c("Unnormalised SPD","Normalised SPD", "IntCal20","ShCal20","Areas with spikes"),
       col=c('bisque3','black',"red","blue","orange"),
       lwd=c(5,1,1,1,5))
title("c",line=-2)

# Panel d
par(mar=c(0,6,0,1))
plot(sahara.spd.nnorm,fill.p=NA, xlim=c(14000,9000),
     ylim=c(0,0.35), xaxt="n",yaxt='n',cex.lab=1)
rect(x=c(9509.948,10228.445,11241.710,12690)+100,
     xright=c(9509.948,10228.445,11241.710,12690)-100,
     ybottom=rep(-100,4),ytop=rep(100,5),border=NA,col='orange')
plot(sahara.spd.nnorm,fill.p='bisque3',ylim=c(0,0.35), xaxt="n", add=TRUE)
plot(sahara.spd.norm,type = 'simple', xaxt="n",add=TRUE)
lines(intcal20[intcal20$BP<=14000 & intcal20$BP>=9000,"BP"],
      reScale(intcal20[intcal20$BP<=14000 & intcal20$BP>=9000,"CRA"])*par('usr')[4],
      col="red",lwd=2)
title("d",line=-2)
title(ylab="Summed Probability",cex.lab=1.5,line=0.5)

#Panel e
par(mar=c(6,6,0,1))
plot(brazil.spd.nnorm,fill.p=NA, xlim=c(14000,9000), ylim=c(0,0.125),
     xaxt="n",yaxt='n',cex.lab=1,cex.axis=0)
rect(x=c(9509.948,10228.445,11241.710,12690)+100,
     xright=c(9509.948,10228.445,11241.710,12690)-100,
     ybottom=rep(-100,4),ytop=rep(100,5),border=NA,col='orange')
plot(brazil.spd.nnorm,fill.p='bisque3',ylim=c(0,0.125), xaxt="n", add=TRUE)
plot(brazil.spd.norm,type = 'simple', xaxt="n", add=TRUE)
axis(side=1, at=seq(14000,9000,-1000), labels=seq(14000,9000,-1000), las=2, cex.axis=1)
lines(shcal20[shcal20$BP<=14000 & shcal20$BP>=9000,"BP"],
      reScale(shcal20[shcal20$BP<=14000 & shcal20$BP>=9000,"CRA"])*par('usr')[4],
      col="blue",lwd=2)
title("e",line=-2)
```
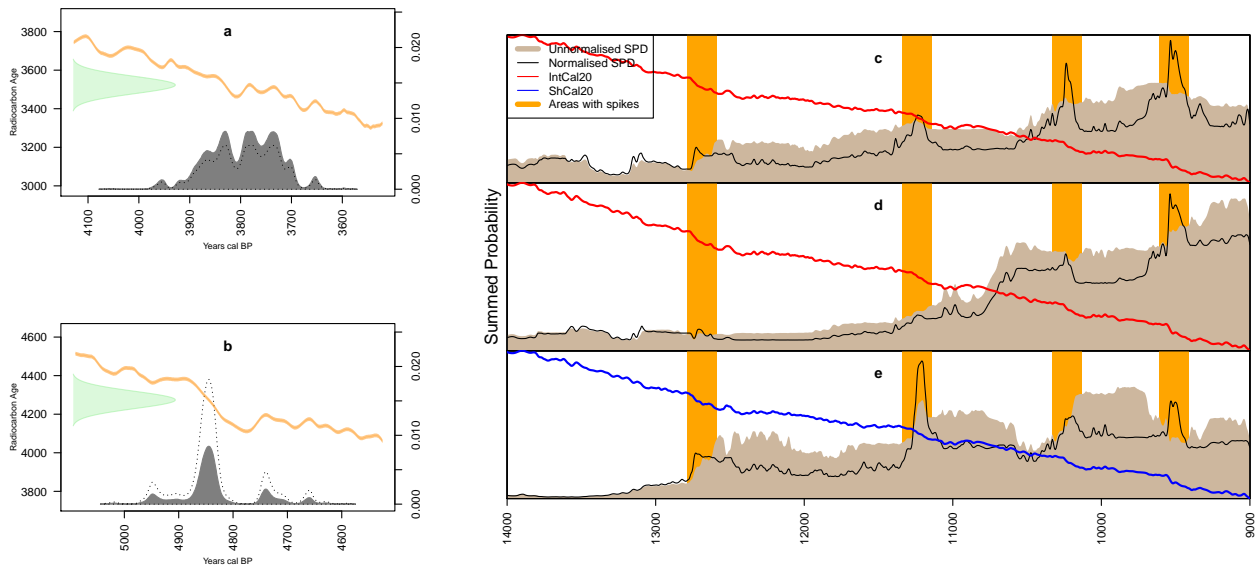
## Figure 3

The script below compares empirical data (`southLevant`) and simulated samples (from a fitted exponential model) using four different methods `calsample` realisations of (a) normalised and (b) unnormalised dates, and `uncalsample` realisations of (c) normalised and (d) unnormalised dates.

```r
# set number of cores for parallel processing:
ncores <- 10

# Calibrate data with and without normalisation
southlevant.x <- calibrate(southLevant$CRA,southLevant$Error,
                           normalised=TRUE,verbose=FALSE)
southlevant.x2 <- calibrate(southLevant$CRA,southLevant$Error,
                            normalised=FALSE,verbose=FALSE)

# 50 years bin
sl.bins <- binPrep(sites = southLevant$SiteName, ages = southLevant$CRA, h = 50)

# Fit and simulate theoretical SPDs using the calsample method:
test.cal.nn = modelTest(southlevant.x2, bins = sl.bins, errors=southLevant$Error,
                        timeRange=c(16000,9000), nsim=1000,
                        ncores=ncores, method='calsample',
                        normalised = FALSE,runm=50,verbose=FALSE)
test.cal.n <- modelTest(southlevant.x, bins = sl.bins, errors=southLevant$Error,
                        timeRange=c(16000,9000), nsim=1000,
                        ncores=ncores, method='calsample',
                        normalised = TRUE,runm=50,verbose=FALSE)

# Fit and simulate theoretical SPDs using the uncalsample method:
test.uncal.nn <- modelTest(southlevant.x2, bins = sl.bins,
                           errors=southLevant$Error,timeRange=c(16000,9000),
                           nsim=1000, ncores=ncores, method='uncalsample',
                           normalised = FALSE,runm=50,verbose=FALSE)
test.uncal.n <- modelTest(southlevant.x, bins = sl.bins,
                          errors=southLevant$Error,timeRange=c(16000,9000),
                          nsim=1000, ncores=ncores, method='uncalsample',
                          normalised = TRUE,runm=50,verbose=FALSE)

# Display Results
par(mfrow=c(2,2))
plot(test.cal.n,lwd.obs = 1.5)
title('a')
plot(test.cal.nn,lwd.obs = 1.5)
title('b')
plot(test.uncal.n,lwd.obs = 1.5)
title('c')
plot(test.uncal.nn,lwd.obs = 1.5)
title('d')
```
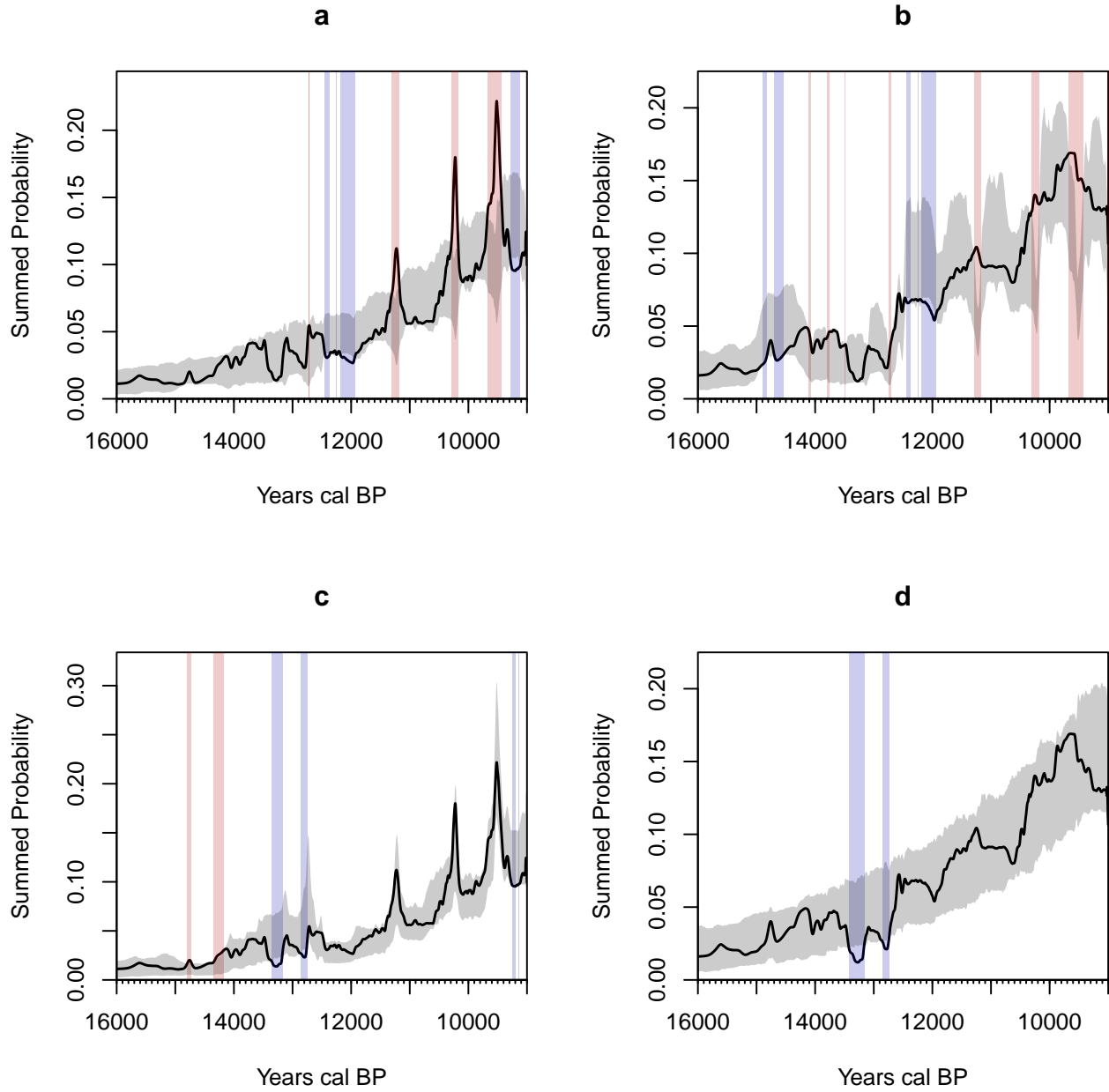
**a**

**b**

**c**

**d**

## Figure 4

The script below shows an example of mark permutation test, replicating the analyses first carried out in Roberts et al 2018. Notice that the method takes into account differences in sampling intensity by generating wider simulation envelopes for sets with smaller sample sizes. Further details on the technique can be found in Crema et al 2016.

```
# prepare data and generate the panregional for levant
levant.emedyd <- subset(emedyd,Region%in%c(1,2))

alldatesLevant.emedyd <- calibrate(x=levant.emedyd$CRA,
                                    errors=levant.emedyd$Error, calCurves='intcal20',
                                    normalised=FALSE,verbose=FALSE)
bins <- binPrep(sites=levant.emedyd$SiteName, ages=levant.emedyd$CRA, h=50)
```

```r
alldates.spd <- spd(alldatesLevant.emedyd,bins=bins,timeRange=c(16000,9000),
                    runm=200,spdnormalised=T,verbose=F)

#execute the permutation test
result <- permTest(x=alldatesLevant.emedyd, bins=bins,
                   marks=levant.emedyd$Region, timeRange=c(16000,9000), runm=200,
                   nsim=1000,spdnormalised=T,verbose=FALSE)

# plot results
par(mfrow=c(2,1))
plot(result,focalm='1',lwd=2,main="Southern Levant")
plot(alldates.spd,add=T,type='simple',lty=2,col='grey27')
plot(result,focalm='2',lwd=2,main="Northern Levant")
plot(alldates.spd,add=T,type='simple',lty=2,col='grey27')
```

**Southern Levant**

Summed Probability

Years cal BP
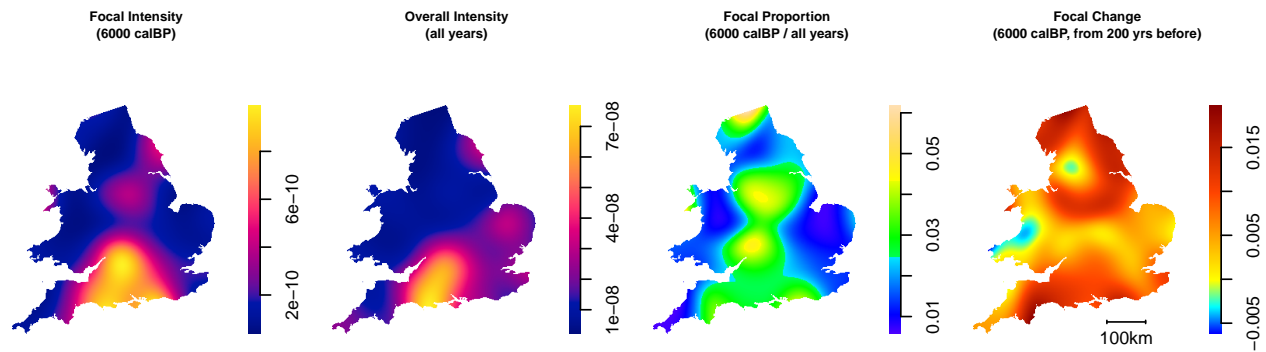
**Northern Levant**

Summed Probability

Years cal BP
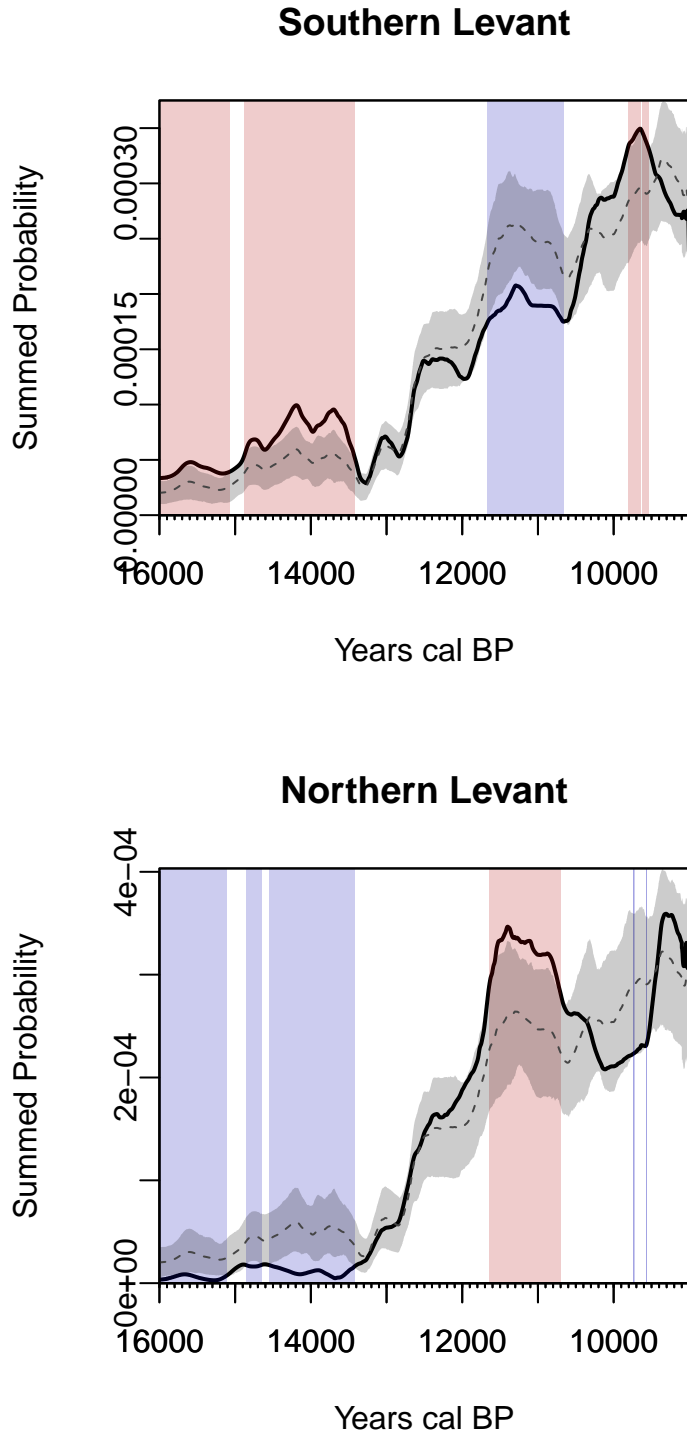
**Figure 5**

The script below generates a sample output of a spatio-temporal kernel density analysis of English and Welsh dates from the Euroevol dataset (Manning et al 2016). The four panels show (from left to right): (a) the spatio-temporal intensity for the focal year 6000 calBP; b) the overall spatial intensity of all Neolithic dates (8000-4000 calBP); (c) the relative risk surface of 6000 cal BP (i.e the ratio between *a* and *b*; and (d) a measure of the spatial pattern of change, mostly growth, from 6200 calBP to 6000 calBP.

```
x <- calibrate(x=ewdates$C14Age, errors=ewdates$C14SD, normalised=FALSE,verbose=FALSE)
## Create centennial timeslices (also with site binning)
bins1 <- binPrep(sites=ewdates$SiteID, ages=ewdates$C14Age, h=50)
stkde1 <- stkde(x=x, coords=ewdates[,c("Eastings", "Northings")],
                win=ewowin, sbw=40000, cellres=2000, focalyears=seq(6500, 5000, -100),
                tbw=100, bins=bins1, backsight=200, outdir=tempdir(),amount=1,verbose=FALSE)
## Plot an example of all four basic outputs for 6000 calBP
par(mar=c(0.5, 0.5, 2.5, 2))
plot(stkde1, 6000, type="all")
## Add scale bar
arrows(x0 = 478616, x1= 478616+100000, y0=40187, y1=40187, code=3, length=0.01, angle=90)
text(x=478616+50000,y=187,"100km")
```



Focal Intensity (6000 calBP) · Overall Intensity (all years) · Focal Proportion (6000 calBP / all years) · Focal Change (6000 calBP, from 200 yrs before)

# Figure 6

The script below shows the typical workflow for the spatial permutation test (from data preparation to visualisation). The panel on the left shows the mean geometric growth rate between 6300-6100 to 6100-5900 calBP, whilst the panel on the right show locations with significant negative or positive deviations (using p-values and q-values) from the expected null hypothesis of spatial stationarity in growth rates. Further details on this method can be found in Crema et al 2017.

```
library(maptools)
```

```
## Loading required package: sp
```

```
## Checking rgeos availability: TRUE
```

```
x <- calibrate(x=ewdates$C14Age, errors=ewdates$C14SD, normalised=FALSE,verbose=FALSE)
bins1 <- binPrep(sites=ewdates$SiteID, ages=ewdates$C14Age, h=50)

# Create SpatialPoints
sites <- unique(data.frame(id=ewdates$SiteID,
                           east=ewdates$Eastings,north=ewdates$Northings))
rownames(sites) <- sites$id
sites <- sites[,-1]
sp::coordinates(sites) <- c("east","north")
sp::proj4string(sites) <- sp::CRS("+init=epsg:27700")

#Compute distance matrix
d <- sp::spDists(sites,sites,longlat=FALSE)
#Compute spatial weights
w <- spweights(d/1000,h=40)
```

```
# Define temporal blocks
breaks <- seq(6300,5900,-200)

# Spatial Permutation Test
res <- sptest(calDates=x,bins=bins1,timeRange=c(6300,5900),locations=sites,
              permute="locations",nsim=1000,
              breaks=breaks,spatialweights=w,verbose=FALSE)

# Plot Results
par(mar=c(1,1,1,1),mfrow=c(1,2))
base=as(ewowin,"SpatialPolygons")
sp::proj4string(base) <- sp::CRS("+init=epsg:27700")
xlim = bbox(base)[1,]
ylim = bbox(base)[2,]
xlim[1] = xlim[1]-100000

plot(base,col="antiquewhite3",border="antiquewhite3",xlim=xlim,ylim=ylim)
plot(res,index=1,option="raw",add=TRUE,legend=TRUE,legSize=0.8,
     location="topleft",baseSize=0.9)
plot(base,col="antiquewhite3",border="antiquewhite3",xlim=xlim)
plot(res,index=1,option="test",add=TRUE,legend=TRUE,legSize=0.8,
     location="topleft",baseSize=0.9)

# Add scale bar
arrows(x0 = 478616, x1= 478616+100000, y0=40187, y1=40187, code=3, length=0.01, angle=90)
text(x=478616+50000,y=187,"100km")
```
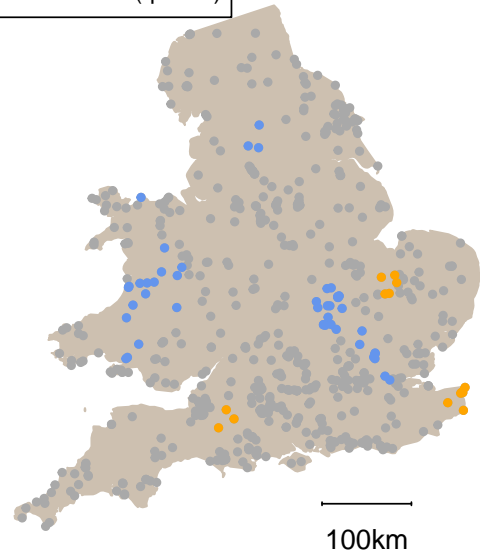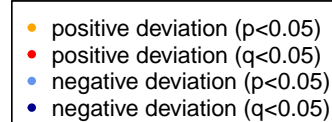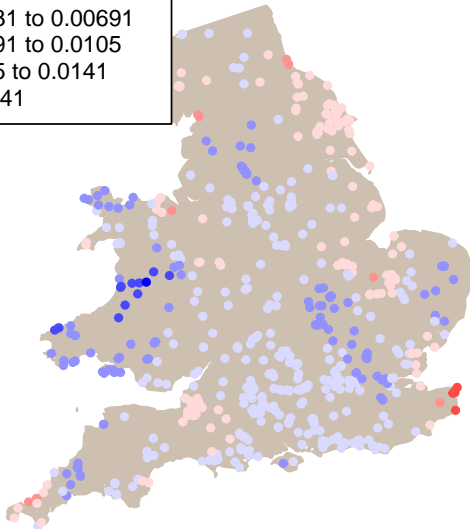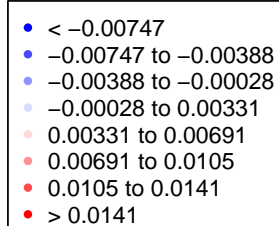


12

# Supplementary Figures

## Supplementary Figure 1

The script below compares OxCal's SUM function (executed via the *oxcAAR* package, Hinz et al 2018) with *rcarbon*'s `spd()` function.

```r
## oxcal via oxcAAR
library(oxcAAR)
quickSetupOxcal()
```

```
## Oxcal doesn't seem to be installed. Downloading it now:
```

```
## Oxcal download to /export/home/erc62/rcarbon_paper_esm/manuscript figures and esm/intcal20 version su
```

```
## Oxcal path set!
```

```
## NULL
```

```r
my_dates <- R_Date(southLevant$LabID, southLevant$CRA, southLevant$Error)
my_sum <- oxcal_Sum(my_dates)
my_result_file <- executeOxcalScript(my_sum)
my_result_text <- readOxcalOutput(my_result_file)
my_result_data <- parseFullOxcalOutput(my_result_text)
res <- my_result_data[nrow(southLevant)+3]

n <- length(res$`ocd[1]`$likelihood$prob) #number of probabilities
PrDens <- res$ocd[[6]][[8]]
resolution <- res$ocd[[6]][[7]]
PrDens <- PrDens/sum(PrDens)/resolution
calBPs <- res$ocd[[6]][[6]]+seq(from=0,by=resolution,length.out=n)-1950

## Compute SPD in rcarbon using a wider timeRange to control edge effect
southlevant.spd3 <- spd(southlevant.x,verbose=F,timeRange=c(21000,8000))

## Compare rcarbon and oxcal
plot(southlevant.spd3,spdnormalised=T,xlim=c(21000,8000),fill.p = 'grey55')
lines(abs(calBPs), PrDens,col="red",lty=1,lwd=1.5)
legend("topleft",legend=c("rcarbon (normalised)","OxCal"),
       col=c("grey55","red"),lwd=c(15,1.5),lty=c(1,1))
```
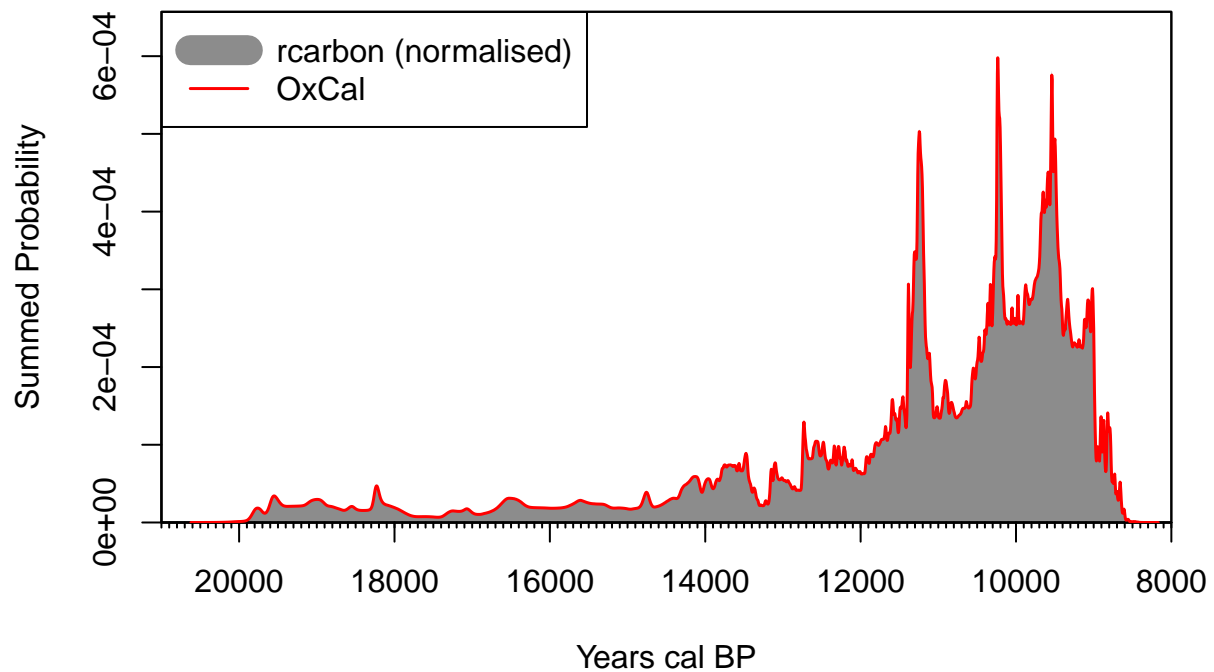
## Supplementary Figure 2

The script below visually compares the summation of unnormalised calibrated dates (a) to a curve produced by first summing uncalibrated Gaussians conventional radiocarbon age distributions (each of weight 1) and then calibrating them in one go (b).

```r
data(emedyd)
d <- subset(emedyd,Region==1&CRA<10000&CRA>7000)

# Create SPD using uncalibrated dates
sumUncalMat <- matrix(NA,nrow=length(13000:4000),ncol=nrow(d))
for (i in 1:nrow(d))
{
  sumUncalMat[,i] <- dnorm(13000:4000,mean=d$CRA[i],sd=d$Error[i])
}

PrDensNN <- apply(sumUncalMat,1,sum)
PrDens <- PrDensNN/sum(PrDensNN)
sumUncal <- data.frame(CRA=13000:4000,PrDens=PrDens)
sumUncal <- as.UncalGrid(sumUncal)

# Calibrate the entire uncalibrated SPD
sumCal <- calibrate(sumUncal,eps=0, spdnormalised=TRUE, verbose=FALSE)

# Calibrate dates without normalisation
x <- calibrate(d$CRA,d$Error,normalised=FALSE,verbose=FALSE)
# Create SPD
SPDnn <- spd(x,timeRange=c(12000,8000),spdnormalised=TRUE,verbose=FALSE)

# Plot and Compare
plot(SPDnn)
```
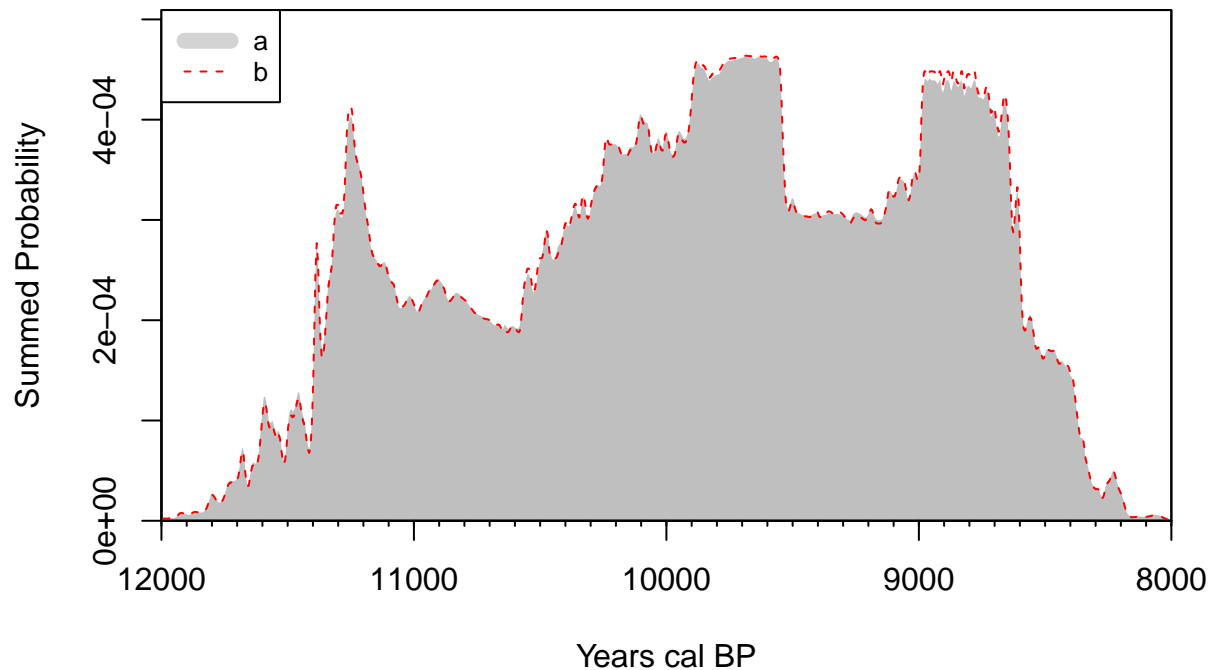
```
lines(sumCal$calBP,sumCal$PrDens,col="red",type="l",lwd=1,lty=2)
legend("topleft",bg="white",legend=c("a","b"),col=c("lightgrey","red"),
        lwd=c(8,1),lty=c(1,2),cex=0.8)
```



## Supplementary Figure 3

The figure below showcase how the simple visual inspection of SPDs can be misleading as it hides the uncertainty associated with individual dates. The script generates two hypothetical datasets, both with the same sample size (10 dates) and covering the same temporal range of analyses but with different degrees of uncertainty associated with each sample. In the dataset a the uncertainty of each date is described by uniform distribution with parameters $t_a$ and $t_b$, with the difference between $t_a$ and $t_b$ held constant to 10 units of time. Furthermore, the dates are abutting, i.e. $t_b$ of the first date correspond to the $t_a$ of the second date. The uncertainty associated with dates in the dataset b is also described with a uniform distribution, but this time $t_b - t_a$ covers the entire temporal range of analysis (100 time units). The summed probability of the two datasets are identical (flat over the time range of analysis, see figures in the left panel), but in the first case, we have a much higher confidence of constant density over time compared to dataset b. This can be shown by creating a composite kernel density estimates (CKDE) usinig randomly drawn dates from each sample. The figures on the right panel show this using 100 iterations, and showcase how the greater uncertainty associated with dataset b. Similar approaches for radiocarbon dates have been proposed in the literature (e.g. Brown 2017) and can be carried out using the ckde() function in *rcarbon*.

```
# Dataset a: low individual uncertainties.
# Each date has a time-span of 10 years with a probability of 0.1 for each year.
a <- matrix(0,nrow=100,ncol=10)

for (i in 0:9)
{
  a[(1+i*10):((i+1)*10),i+1] <- 0.1
}

# Dataset b: high individual uncertainties.
```

```r
# Each date has a time-span of 100 years with a probability of 0.01 for each year.
b <- matrix(0.01,nrow=100,ncol=10)


#Comparison of SPDs
par(mfcol=c(2,2))
plot(1:100,apply(a,1,sum),main="SPD dataset a",type="l",
     ylab="Summed Probability",xlab="time")
plot(1:100,apply(b,1,sum),main="SPD dataset b",type="l",
     ylab="Summed Probability",xlab="time")

#CKDE based on random dates from each sample, 100 repetitions
set.seed(1)
plot(density(apply(a,2,function(x){sample(1:100,size=1,prob=x)}),bw=10),
     xlim=c(1,100),ylim=c(0,0.02),col=rgb(0,0,0,0.25),
     main="CKDE dataset a",xlab="time")
replicate(100,lines(density(apply(a,2,function(x)
{sample(1:100,size=1,prob=x)}),bw=10),col=rgb(0,0,0,0.2)))
#CKDE based on random dates from each sample, 100 repetitions
plot(density(apply(b,2,function(x){sample(1:100,size=1,prob=x)}),bw=10),
     xlim=c(1,100),ylim=c(0,0.025),col=rgb(0,0,0,0.2),
     main="CKDE dataset b",xlab="time")
replicate(100,lines(density(apply(b,2,function(x)
{sample(1:100,size=1,prob=x)}),bw=10),col=rgb(0,0,0,0.2)))
```
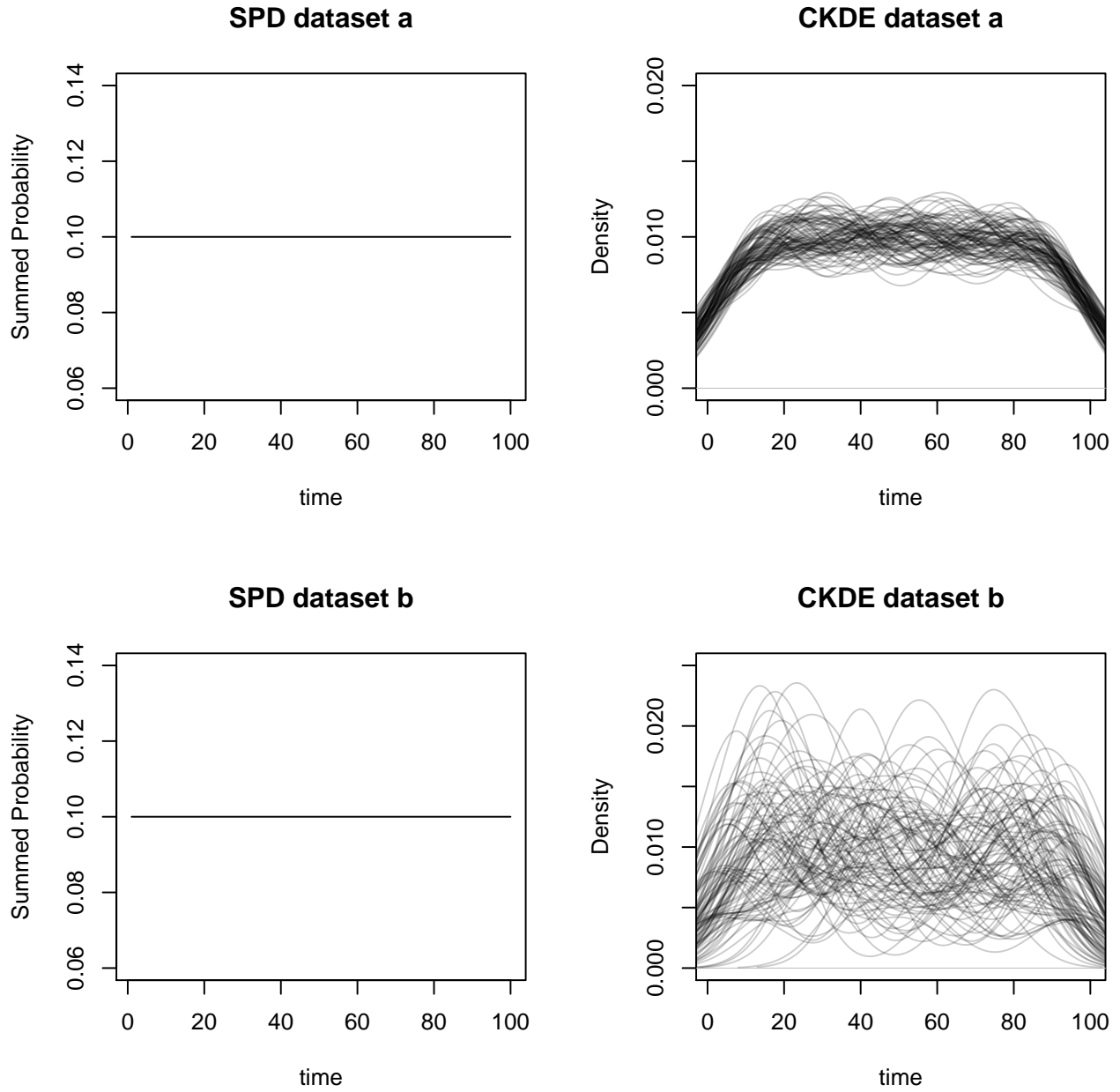
## Supplementary Figure 4

The script below compares observed and expected SPD (panel *a*) and growth rates (panel *b*) of British and Welsh dates for the interval 8,000 - 4,000 calBP (n.bins=1461, n.dates=2327). The rates of change are computed using he default settings of the formula $(d_t/d_{t-\Delta t})^{1/\Delta t-1}$, where $d_t$ is the summed probability of calibrated radiocarbon dates at a point in time $t$, $d_{t-\Delta t}$ is the summed probability probability at the "backsight" year, in this case 200 years prior to each focal year (i.e. $\Delta t = 200$). Notice how the significant departure from the null hypothesis are different depending on whether the density or the rate of change in density is being evaluated.

```
ncores <- 10 #number of cores for parallel processing
x.ew <- calibrate(x=ewdates$C14Age, errors=ewdates$C14SD, normalised=FALSE,verbose=FALSE)
bins.ew <- binPrep(sites=ewdates$SiteID, ages=ewdates$C14Age, h=50)
```

```
ew.mod <- modelTest(x.ew,errors=ewdates$C14SD,bins=bins.ew,timeRange=c(8000,4000),
                    nsim=1000,runm=50,ncores=ncores,backsight=100,verbose=FALSE)
```

```
## Warning in modelTest(x.ew, errors = ewdates$C14SD, bins = bins.ew, timeRange =
## c(8000, : edgeSize reduced
```
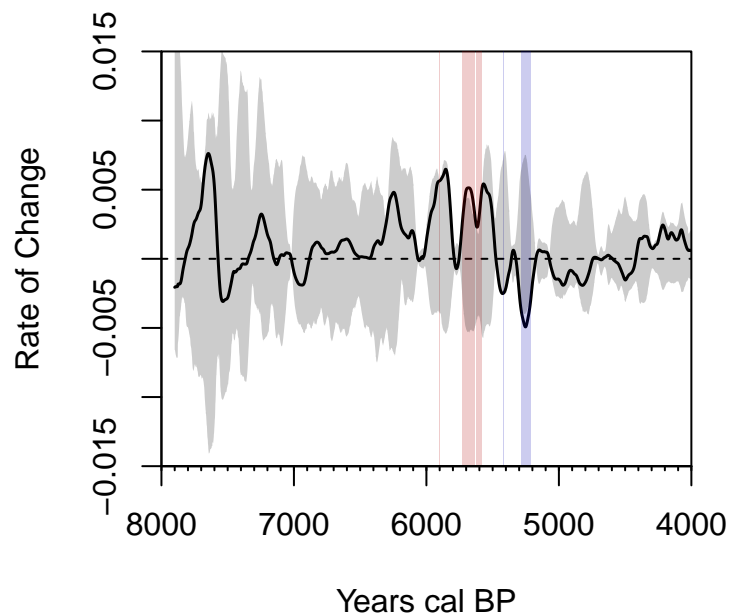
```
par(mfrow=c(2,1))
plot(ew.mod,lwd.obs = 1.5,
     main = paste0("a (global P-value: ", round(ew.mod$pval,3),")"))
plot(ew.mod,lwd.obs=1.5, type='roc',
     main = paste0("b (global P-value: ", round(ew.mod$pval.roc,3),")"),
     ylim=c(-0.015,0.015))
```

**a (global P−value: 0.001)**
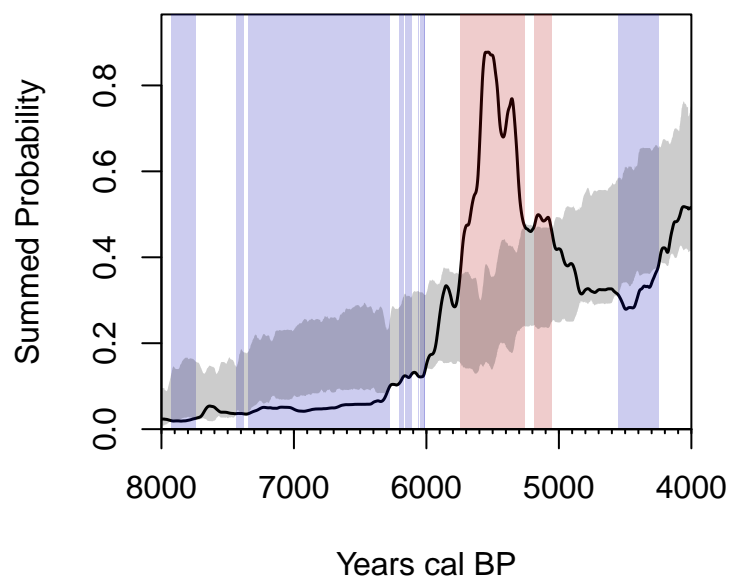


**b (global P−value: 0.488)**



# References

Brown, W. A. (2017). The past and future of growth rate estimation in demographic temporal frequency analysis: Biodemographic interpretability and the ascendance of dynamic growth models. *Journal of Archaeological Science, 80*, 96–108.

Bueno, L., Dias, A. S., & Steele, J. (2013). The Late Pleistocene/Early Holocene archaeological record in

Brazil: A geo-referenced database. Quaternary International, 301, 74–93.

Crema, E. R., Bevan, A., & Shennan, S. (2017). Spatio-temporal approaches to archaeological radiocarbon dates. Journal of Archaeological Science, 87(Supplement C), 1–9.

Crema, E. R., Habu, J., Kobayashi, K., & Madella, M. (2016). Summed Probability Distribution of 14 C Dates Suggests Regional Divergences in the Population Dynamics of the Jomon Period in Eastern Japan. PLOS ONE, 11(4), e0154809.

Hinz, M. Schmid, C., Knitter, D., & Tietze, C. (2018). oxcAAR: Interface to 'OxCal' Radiocarbon Calibration. R package version 1.0.0.

Manning, K., & Timpson, A. (2014). The demographic response to Holocene climate change in the Sahara. Quaternary Science Reviews, 101, 28–35.

Manning, K., Colledge, S., Crema, E., Shennan, S., & Timpson, A. (2016). The Cultural Evolution of Neolithic Europe. EUROEVOL Dataset 1: Sites, Phases and Radiocarbon Data. Journal of Open Archaeology Data, 5(0).

Reimer, P. J., Austin, W.E., Bard, E., Bayliss, A., et al. (2020). The IntCal20 Northern Hemispher Radiocarbon Age Calibration Curve (0–55 CAL kBP) Radiocarbon, 55(4), 1869–1887.

Roberts, N., Woodbridge, J., Bevan, A., Palmisano, A., Shennan, S., & Asouti, E. (2018). Human responses and non-responses to climatic variations during the last Glacial-Interglacial transition in the eastern Mediterranean. Quaternary Science Reviews, 184, 47–67.