Supplement of

Reconstructions of biomass burning from sediment-charcoal records to improve data–model comparisons

Jennifer R. Marlon et al.

Correspondence to: Jennifer R. Marlon (jennmarlon@gmail.com)

The copyright of individual parts of the supplement might differ from the CC-BY 3.0 licence.
# STEP 1 - PREPARE BASE FIGURES
rm(list=ls())
library(devtools)
library(paleofire)
library(GCD)

#install_github("paleofire","paleofire", ref="daily")
#load_all("~/Work/research/GPWG/paleofire/")

# WORKING DIRECTORY
setwd('~/methods/charcoal/GCD v3.0 Paper figures/')

plotdata.file = './Data/All_GCDv1.1_rawplots_BB21k_2015-09-18.rds'
#plotdata.file = './Data/All_GCDv1.1_rawplots_BB1k_2015-09-18.rds'
TR.file = './Data/All_GCDv1.1_Transformed_BB21k.rds'
#TR.file = './Data/All_GCDv1.1_Transformed_BB1k.rds'
TR.mode = 0 #1 # 0==Run transformation, save result for later, 1==use saved data

# Figure file name base.
# - Can include a path (otherwise goes in working directory); all directories must exist.
# - Year designation and file suffix will be added automatically
# - Set to NULL to only print to screen

fig.base.name = './1perPg/GCDv3-SI-21k_'
# fig.base.name = './1perPg/GCDv3-SI-1k_
# fig.base.name = NULL # Use for paper figures

# Base map ('coasts' or 'countries')
base.map = 'coasts'

# Grid resolution and extent (in degrees)
grd.res = 5
grd.ext = c(-180,180,-90,90) # c(lonmin, lonmax, latmin, latmax)

# Composite params
tarAge = seq(0,21000,1000) #increments = 1000 (Figs. 6 & 7 gridded maps)
tarAge = seq(-50,1050,100) #increments = 100
hw = 250 #10 (v3 Figs. 6 & 7 paper setting)
binhw = 500 #50 #20 (v3 Figs. 6 & 7 paper setting) ### THIS CHANGES THE MAP INTERVALS & FILE NAMES
n.boot = 1000 #1000 #(v3 Fig. 6 paper setting)

# Dot size parameters
cx.mult = 1.5
cx.minsize = 0.4

# Projection for maps
# Unprojected
proj4 = "+proj=longlat"

# Robinson
# proj4 = "+proj=robin +lon_0=0 +x_0=0 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"

# Gall-Peters equal-area projection
# proj4 = "+proj=cea +lon_0=0 +lat_ts=45 +x_0=0 +y_0=0 +ellps=WGS84 +units=m +no_defs"

# Lambert equal-area
# proj4 = '+proj=laea +lat_0=30 +lon_0=0 +x_0=0 +y_0=0'

# Mercator
# proj4 = "+proj=merc +lon_0=0 +k_0=1 +x_0=0 +y_0=0"

# ------------------------ End Options
# ----- Make dir
if(!is.null(fig.base.name))
   dir.create(dirname(fig.base.name), recursive=T, showWarnings=F)

# ----- Transform records (slow, which is why TR.mode=1 is added)
if(TR.mode==0) {
  # New transform
  id = pfSiteSel() # Select all sites
  TR = pfTransform(id, method=c("MinMax","Box-Cox","Z-Score"),
                   BasePeriod=c(200,21000),QuantType="INFL") #Fig. 6
  TR = pfTransform(id, method=c("MinMax","Box-Cox","Z-Score"),
                   BasePeriod=c(200,1000),QuantType="INFL") #Fig. BB1k
  saveRDS(TR, file=TR.file)
} else {
  # load existing
  TR = readRDS(TR.file)
}

# ----- Run pfDotMap
dotmap = pfDotMap ( TR=TR, tarAge=tarAge, binhw=binhw, hw=hw,
                    n.boot=n.boot,
                    fig.base.name=fig.base.name, base.map=base.map,
                    grd.res=grd.res, grd.ext=c(-180,180,-90,90), proj4=proj4,
                    cx.minsize=cx.minsize, cx.mult=cx.mult
                  )

saveRDS(dotmap, plotdata.file)

# STEP 2 – MAKE PLOTS
rm(list=ls())
library(rgdal)
library(rworldmap)

# ---------------
setwd(~methods/charcoal/GCD v3.0 Paper figures/)
dotmap = readRDS('./Data/All_GCDv1.1_rawplots_BB1k_2015-08-18.rds')
outdir = './PaperFigs_pfCompositeLF/'
y.lim = c(-70,80)
x.lim = c(-180,180)

### extract 1000-year slices desired
#grd = list(dotmap$sp.grd[[1]], dotmap$sp.grd[[7]], dotmap$sp.grd[[22]])
#site = list(dotmap$sp.sites[[1]], dotmap$sp.sites[[7]], dotmap$sp.sites[[22]])
# n.bin = length(grd)
# picked 1,6,11 first

#grd = list(dotmap$sp.grd[[1]], dotmap$sp.grd[[2]], dotmap$sp.grd[[11]])
#site = list(dotmap$sp.sites[[1]], dotmap$sp.sites[[2]], dotmap$sp.sites[[11]])
# n.bin = length(grd)
Load base map
proj4 = proj4string(grd[[1]])
data(countriesCoarse)  # A dataset in rworldmap used in the plots below
data(coastsCoarse)  # An alternative base map. Needs one fix:
countriesCoarse = spTransform(countriesCoarse, CRS(proj4))
coastsCoarse = spTransform(coastsCoarse, CRS(proj4))

dir.create(outdir, recursive=T, showWarnings=F)
file.plot = paste0(outdir, 'Mean.pdf')
file.legend = paste0(outdir, 'Mean_v3_legend.pdf')

cols = c("#0571B0","#92C5DE","#F4A582","#CA0020")  # modified from colorbrewer
# cols = rev(c( rgb(1.000,0.250,0.000), rgb(1.000,0.501,0.144),
# rgb(1.000,0.740,0.376), rgb(1.000,0.924,0.694), rgb(0.887,1.000,1.000),
# rgb(0.607,0.918,1.000), rgb(0.376,0.792,1.000), rgb(0.194,0.630,1.000) ))
# cols = c( rgb(0,0,1), rgb(0.194,0.630,1), rgb(0.376,0.792,1), grey(0.95),
# rgb(1,0.74,0.376), rgb(1,0.501,0.144), rgb(1,0,0) )
cuts = c(-1.75,-1.25,-0.75,-0.25,0.25,0.75,1.25,1.75)
# cuts = c(-1.5,-0.9,-0.3,0.3,0.9,1.5)  # Defines range and resolution of color scale
# cuts = c(-3,-1.8,-0.6,0,0.6,1.8,3)  # Defines range and resolution of color scale
cx.sizes = c(0.75,1)

panel.labels = rep("","3")  # c("Present","6 ka","21 ka")
grid.grey = grey(0.8)

mp = list()
for(i in 1:n.bin) {
  sp.grd = grd[[i]]

  # Assign symbol size based on whether CI contain 0
  cx = ifelse(sp.grd$CI.lower>0 | sp.grd$CI.upper<0, max(cx.sizes), min(cx.sizes))

  # The previous line will produce NA for cells with n=1 since CI are undefined. Give these "non-significant" symbol size by default.
  cx[which(sp.grd$sitesPerCell==1)] = min(cx.sizes)

  # Create plot object (actually plotted later)
  mp[[i]] =
spplot(sp.grd, 'mean.CHAR', xlim=x.lim, ylim=y.lim,
cuts=cuts, colorkey=T, col.regions=cols, cex=cx, edge.col=grey(0.7), lwd=0.1,
sp.layout=list(
  list("sp.lines",coastsCoarse,col=grid.grey,lwd=0.3),
  list("sp.polygons",countriesCoarse,col=grid.grey,lwd=0.3),
  list("sp.lines",gridlines(sp.grd),col=grid.grey, lwd=0.3),
  list("sp.text",c(-150,-50), panel.labels[i], fontface=2)),
par.settings=list(
  layout.widths=list(left.padding=3, right.padding=3),
  layout.heights=list(top.padding=-3, bottom.padding=-3)),
scales=list(alternating=0,tck=-0.5)
) # End spplot
if(i==1) mp.legend = mp[[i]]
mp[[i]]$legend = NULL
names(mp.legend$legend) = "bottom"
mp.legend$legend$bottom$args$key$space="bottom"

save.plot = T
if(save.plot) pdf(file.plot, width=17.5/2.54, height=11)
print(mp[[1]], position=c(0,0.635,1,0.905), panel.width=list(17.2,"cm"),
panel.height=list(17.5*0.42,"cm"), more=T)
print(mp[[2]], position=c(0,0.365,1,0.635), panel.width=list(17.2,"cm"),
panel.height=list(17.5*0.42,"cm"), more=T)
print(mp[[3]], position=c(0,0.095,1,0.365), panel.width=list(17.2,"cm"),
panel.height=list(17.5*0.42,"cm"), more=F)
if(save.plot) dev.off()

if(save.plot) {
  pdf(file.legend, width=20, height=5)
  print(mp.legend, position = c(0,0.0,1,1), panel.width=list(17.2,"cm"),
  panel.height=list(17.5*0.42,"cm"))
  dev.off()
}

# ----------------------------- NSITES PLOT -------------------------------
file.plot = paste0(outdir, 'Nsites.pdf')
file.legend = paste0(outdir, 'Nsites_legend.pdf')

cols = grey(0.2) # Can be replaced by a vector if different colors are desired

cuts = c(0.9,1.9,9.9,1000) # Where to divide symbol sizes
cx.legend = c("1", "2-9", "10+") # legend text
cx.key = c(0.3,0.4,0.5)
n.cx = length(cuts)-1 # number of bins represented
mp = list()
for(i in 1:n.bin) {
  sp.grd = grd[[i]]
  
  cx = cx.key[ cut(sp.grd$sitesPerCell, cuts, labels=F) ]

  # Create plot object (actually plotted later)
  mp[[i]] =
    spplot(sp.grd, 'sitesPerCell', xlim=x.lim, ylim=y.lim,
       cex=cx, cex.key=cx.key, legendEntries=cx.legend, cuts=cuts,
       col.regions=cols, edge.col="transparent",
       sp.layout=list(
         list("sp.lines",coastsCoarse,col=grid.grey,lwd=0.3),
         list("sp.polygons",countriesCoarse,col=gridgrey,lwd=0.3),
         list("sp.lines",gridlines(sp.grd),col=gridgrey, lwd=0.3),
         list("sp.text",c(-150,-50), panel.labels[i], fontface=2, cex=0.7)),
       par.settings=list(
         layout.widths=list(left.padding=-3, right.padding=-3),
         layout.heights=list(top.padding=-3, bottom.padding=-3)),
       scales=list(alternating=0,tck=-0.5)
    )
  if(i==1) mp.legend = mp[[i]]
  mp[[i]]$legend = NULL
}

save.plot = T
if(save.plot) pdf(file.plot, width=8.5/2.54, height=5.3)

print(mp[[1]], position=c(0,0.635,1,0.905), panel.width=list(8.25,"cm"),
      panel.height=list(8.25*0.42,"cm"), more=T)
print(mp[[2]], position=c(0,0.365,1,0.635), panel.width=list(8.25,"cm"),
      panel.height=list(8.25*0.42,"cm"), more=T)
print(mp[[3]], position=c(0,0.095,1,0.365), panel.width=list(8.25,"cm"),
      panel.height=list(8.25*0.42,"cm"), more=F)
if(save.plot) dev.off()

if(save.plot) {
  pdf(file.legend, width=8.5/2.54, height=5)
  print(mp.legend, position = c(0,0,1,1), panel.width=list(8.25,"cm"),
       panel.height=list(8.25*0.42,"cm"))
  dev.off()
}
# NSITES PLOT

file.plot = paste0(outdir, 'Ncells.pdf')
file.legend = paste0(outdir, 'Ncells_legend.pdf')

cols = grey(0.2)  # Can be replaced by a vector if different colors are desired

cuts = c(0.9,1.9,3.9,100)  # Where to divide symbol sizes

cx.legend = c("1", "2-3", "4+")  # legend text

cx.key = c(0.3,0.4,0.5)

n.cx = length(cuts) - 1  # number of bins represented

# ind.non0 = which(cx>0)  # Don't want to change size 0 (== not plotted)
# cx[ind.non0] = cx[ind.non0] + cx.minsize - min(cx[ind.non0])

mp = list()
for(i in 1:n.bin) {
  sp.site = site[[i]]

  cx = cx.key[ cut(sp.site$cellsPerSite, cuts, labels=F) ]

  # Create plot object (actually plotted later)
  mp[[i]] =
  spplot(sp.site, 'cellsPerSite', xlim=xlim, ylim=ylim,
       cex=cx, cex.key=cx.key, legendEntries=cx.legend, cuts=cuts,
       col.regions=cols, edge.col="transparent",
       sp.layout=list(
         list("sp.lines",coastsCoarse,col=grid.grey,lwd=0.3),
         list("sp.polygons",countriesCoarse,col=grid.grey,lwd=0.3),
         list("sp.lines",gridlines(sp.grd),col=grid.grey, lwd=0.3),
         list("sp.text",c(-150,-50), panel.labels[i], fontface=2, cex=0.7)),
       par.settings=list(
         layout.widths=list(left.padding=-3, right.padding=-3),
         layout.heights=list(top.padding=-3, bottom.padding=-3)),
       scales=list(alternating=0,tck=0.5))

  )

  if(i==1) mp.legend = mp[[i]]
  mp[[i]]$legend = NULL
}

save.plot = T
if(save.plot) pdf(file.plot, width=8.5/2.54, height=5.3)

print(mp[[1]], position=c(0.0,0.635,1,0.905), panel.width=list(8.25,"cm"),
      panel.height=list(8.25*0.42,"cm"), more=T)
print(mp[[2]], position=c(0,0.365,1,0.635), panel.width=list(8.25,"cm"),
panel.height=list(8.25*0.42,"cm"), more=T)
print(mp[[3]], position=c(0,0.095,1,0.365), panel.width=list(8.25,"cm"),
panel.height=list(8.25*0.42,"cm"), more=F)
if(save.plot) dev.off()

if(save.plot) {
  pdf(file.legend, width=8.5/2.54, height=5)
  print(mp.legend, position = c(0,0,1,1), panel.width=list(8.25,"cm"),
panel.height=list(8.25*0.42,"cm"))
  dev.off()
}
Charcoal Influx z-Scores: 300−400 BP

Number of sites per grid cell

Number of grid cells influenced by each site

- 1
- 2−5
- 6−10
- 11−20
- >20
Charcoal Influx z-Scores: 400–500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 600–700 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 700–800 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 800–900 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 4500–5500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 6500–7500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 8500–9500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 9500–10500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 10500–11500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 11500–12500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 12500–13500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 13500–14500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx $z$-Scores: 14500–15500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 15500–16500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 16500–17500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 17500–18500 BP

Number of sites per grid cell

Number of grid cells influenced by each site
Charcoal Influx z-Scores: 19500–20500 BP

Number of sites per grid cell

Number of grid cells influenced by each site

- 1
- 2–5
- 6–10
- 11–20
- >20

-2
-1
0
1
2
Charcoal Influx $z$-Scores: 20500–21500 BP

- Number of sites per grid cell
  - 1
  - 2–5
  - 6–10
  - 11–20
  - >20

- Number of grid cells influenced by each site
  - 1
  - 2
  - 3
  - 4
  - >4