The HiDA Paleo Climate Challenge

Moritz Johannes Kirschner, Team EOF Brudi

4/2020

Outline

Thank you for the opportunity to participate in this challenge. I've tried to answer the following questions using the data supplied:

1. Can you find out which periods were dominated by climate forcing without knowing the exact dates of volcanic events or low solar activity? We know the solution to this question because we can match our simulations with time series data for changes in solar activity and volcanic outbreaks at the time, but we would like to be able to tell just from the model.
2. Can you tell how the spatial patterns change for periods with strong volcanic activity? In other words do all regions in the two simulations show the same temperature decline or maybe even increase? Please consider that we see greater impact of factors other than volcanos and solar activity in so-called extratropical regions, the closer we move towards the poles, north of 30°N and south of 30°S. It would also be interesting to test whether tropical regions show a more coherent temperature response pattern in R1 and R2 simulations compared to extratropical regions.

I've decided to tackle the questions one-by-one, but in the other way around. Section 1 will tackle question 2 and vice versa.

For these analysis, I've used R and CDOs, with mostly standard packages. For spectral estimation, I've also used the Earth System Diagnostics research group's PaleoSpec package: https://github.com/EarthSystemDiagnostics/paleospec

Section 1: Spatial patterns change for periods with strong volcanic activity

For this section, I've implemented an analysis found in *Impact of explosive volcanic eruptions on the main climate variability modes*, by Swingedouw et al., 2017, which uses standardized anomalies to neatly visualize the spatial response to high aerosol forcing.

```r
library(maps)
library(ncdf4)
library(stringi)
library(RColorBrewer)
```

I first define some helper functions which will be used later.

```r
#, echo=FALSE

getStdAnom <- function(path, runs, ind.volc.active)
{
  data <- list()
  data[["mean.active"]] <- list()
  data[["mean"]] <- list()
  data[["sd"]] <- list()
```

```r
  for(run in runs)
  {
    nc <- nc_open(paste0(path,"T2m_",run,"_ym_1stMill.nc"))
    raw <- ncvar_get(nc, "T2m")
    nc_close(nc)

    data[["mean.active"]][[run]] <- apply(raw[,,ind.volc.active], c(1,2), mean, na.rm=T)
    data[["mean"]][[run]] <- apply(raw[,,], c(1,2), mean, na.rm=T)
    data[["sd"]][[run]] <- apply(raw[,,], c(1,2), sd, na.rm=T)
  }
  rm(raw)
  gc()

  return(data)
}

plot_matrix <- function(fld, lons, lats, col, zlim)
{
  lons.image <- lons-180
  lats.image <- rev(lats)

  l <- length(lons)
  lons.data.index <- c((l/2+1):l,1:(l/2))
  lats.data.index <- rev(1:length(lats))

  plot(0,type="n",xlim=range(lons)-180, ylim=range(lats), axes=F,
       xlab="", ylab="", yaxs="i", xaxs="i")
  image(lons.image, lats.image,
        fld[lons.data.index,lats.data.index],
        zlim = zlim, col = col, add=T)
  map("world",interior=F,add=T)
}

plot_colorbar <- function(zlim, col = heat.colors(12), add.axis=TRUE, axis.pos=1, cex=1, ...)
{
  breaks <- seq(zlim[1], zlim[2], length.out=(length(col)+1))
  poly <- vector(mode="list", length(col))
  for(i in seq(poly)){
    poly[[i]] <- c(breaks[i], breaks[i+1], breaks[i+1], breaks[i])
  }
  if(axis.pos %in% c(1,3)){ylim<-c(0,1); xlim<-range(breaks)}
  if(axis.pos %in% c(2,4)){ylim<-range(breaks); xlim<-c(0,1)}
  plot(1,1,t="n",ylim=ylim, xlim=xlim, axes=FALSE, xlab="", ylab="", xaxs="i", yaxs="i", ...)
  for(i in seq(poly)){
    if(axis.pos %in% c(1,3)){
      polygon(poly[[i]], c(0,0,1,1), col=col[i], border=NA)
    }
    if(axis.pos %in% c(2,4)){
      polygon(c(0,0,1,1), poly[[i]], col=col[i], border=NA)
    }
  }
  box()
  if(add.axis) {
```

```r
    axis(axis.pos,labels=F)
    mtext(axTicks(axis.pos),side=axis.pos, at=axTicks(axis.pos),font=2,line=2,cex=cex)
  }
}
```

Let's do some time validation first

```r
# Get forcing and time axis
time <- list()
ts.forcing <- list()
for (i in 1:2)
{
  forcing <- c("Solar_forcing","Volc_Forc_AOD")[i]
  ncvar.forcing <- c("TSI","AOD")[i]
  nc <- nc_open(paste0(forcing,"_1st_mill.nc"))
  ts.forcing[[forcing]] <- ncvar_get(nc, ncvar.forcing)
  time[[forcing]] <- ncvar_get(nc, "time")
  nc_close(nc)
}

# Get model data time axis
for (run in c("R1","R2"))
{
  nc <- nc_open(paste0("T2m_",run,"_ym_1stMill.nc"))
  time[[run]] <- ncvar_get(nc, "time")
  nc_close(nc)
}

# print(time)

# It looks like all data is saved yearly but the model data prints the year means as YY-07-16
# We can just cut the last 4 digits
for (i in 1:length(time))
{
  # stop if any timestep in forcing time axis does not end in 1010
  if(i %in% c(1,2))
    stopifnot(!any(substr(stri_reverse(as.character(time[[i]])),1,4)!="1010"))

  # stop if any timestep in forcing time axis does not end in 0716
  if(i %in% c(3,4))
    stopifnot(!any(substr(stri_reverse(as.character(time[[i]])),1,4)!="6170"))

  time[[i]] <- floor(time[[i]]*0.0001)
}

print(paste("All timesteps equal to each other?",
            !any(apply(simplify2array(time), 1, function(x) length(unique(x))) != 1)))
```

```
## [1] "All timesteps equal to each other? TRUE"
```

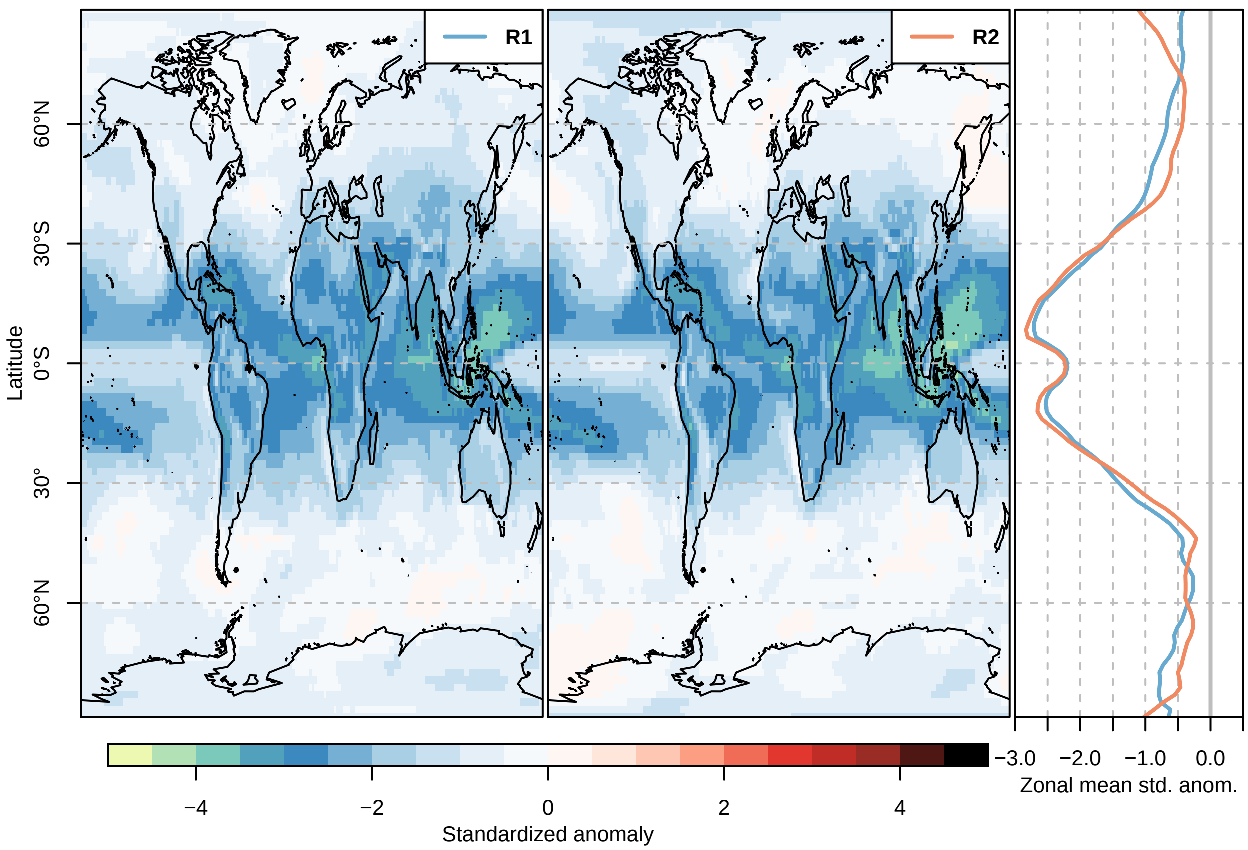
Generate the data.

```r
# Generate standardized anomalies
runs <- c("R1","R2")

data <- getStdAnom("./", runs, ts.forcing$Volc_Forc_AOD > 0.15)
```

And plot it.

```
# Get grid
nc <- nc_open("T2m_R1_ym_1stMill.nc")
lons <- ncvar_get(nc, "lon")
lats <- ncvar_get(nc, "lat")
nc_close(nc)

# Plot setup
lineCol <- list("R1"="#67A9CF", "R2"="#EF8A62")
col <- colorRampPalette(rev(c("black","#962D26","#de2d26","#fc9272","#fee0d2","white",
                              "#deebf7","#9ecae1","#3182bd","#7fcdbb","#edf8b1")))
nbreakpoints <- 20
zlim <- c(-5,5)

letters <- c("a","b","c")
i.letter <- 0

latgrid <- seq(-90,90,30)

runs <- c("R1","R2")

# Start to plot
cex <- 0.7

layout(matrix(c(1:3,c(4,4,0)),2,3,byrow=T),widths=c(1,1,0.5),heights=c(1,0.1))
par(oma=c(1,3,0,0), mar=c(1,0,1,0.2), cex=cex)

zonal_fld <- list()
for (run in runs)
{

  # Compute std anom and save zonal mean for later
  fld <- ((data$mean.active[[run]]-data$mean[[run]])/data$sd[[run]])
  zonal_fld[[run]] <- apply(fld, 2, mean)

  # Plotting using helper function above
  plot_matrix(fld, lons, lats, col=col(nbreakpoints), zlim=zlim)
  abline(h=latgrid, col="grey", lty=2, lwd=1)

  legend("topright", legend=run, text.font=2, lwd=2, col=lineCol[[run]], bg="white")

  if(run == runs[1])
  {
    axis(2, at=latgrid, labels=paste0(abs(latgrid),"°",c("N","N","","S","S")))
    mtext(side=2, "Latitude", cex=cex, line=2)
  }

  box()
}

plot(0,type="n",xlim=c(-3,0.5), ylim=range(lats), axes=F, xlab="", ylab="", yaxs="i", xaxs="i")
abline(h=latgrid, v=seq(-3,0.5,0.5), col="grey", lty=2, lwd=1)
```

```r
abline(v=0, col="grey", lty=1, lwd=2)
for (run in runs)
{
  lines(zonal_fld[[run]], lats, col=lineCol[[run]],lwd=2)
}
axis(1)
mtext(side=1, "Zonal mean std. anom.", cex=cex, line=2)
box()

par(mar=c(2,1,0,1))
plot_colorbar(zlim, col=col(nbreakpoints), add.axis=F)
axis(1)
mtext(side=1, "Standardized anomaly", cex=cex, line=2)
```



As we can see, our analysis showed only minimal differences in response to high aerosol forcing in the two simulations R1 and R2.

Section 2: Which periods were dominated by climate forcing

For this analysis, I couldn't rely on the forcing time series supplied. Hence I thought this would be a good first project for me to work with Empirical Orthogonal Functions (EOFs). However I was not able extract useful information for the solar forcing from the time series.

The idea is to compute EOFs on a masked ensemble mean of the two simulations. We can use an ensemble mean since we know the same forcing was used and the simulation times are synchronized. This way, we increase the signal (forcing) -to-noise (weather) in our time series. We are masking areas above and below

30°N and 30°S since areas in those regions are highly variable on longer than annual timescales (sea ice retreat and growth probably) and any forcing signal will not be picked up by the EOFs. The fact that solar and volcanic forcing effects low-latitudes the most work in our favor here.

We then analyze the EOF coefficient time series and find that the first two are highly similar to the AOD time series. We could therefore reconstruct period dominated by volcanic forcing by these time series. Unfortunately, we were not able to find any other EOF coefficient time series which could be interpreted as the solar forcing. To search for these, we looked for peaks in the spectra 10 year region.

```r
library(PaleoSpec)
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
detrend <- function(y)
{
  fit <- lm(y ~ index(y))
  return(y - index(y)*fit$coefficients[2] - fit$coefficients[1])
}

plot_mean_conf <- function(x, y, upper, lower, col, lty)
{
  polygon(c(x, rev(x))
          , c(lower, rev(upper))
          , col=rgb(t(col2rgb(col))
                    , alpha=100
                    , maxColorValue=255)
          , border = NA)
  lines(x, y
        , type='l'
        , lw=2
        , lty=lty
        , col=rgb(t(col2rgb(col))
                  , alpha=255
                  , maxColorValue=255))
}

plot_spectra <- function(tslist, lty, col)
{
  if(!is.list(tslist)) tslist <- list(tslist)

  sp <- lapply(tslist, function(x)
    {
      LogSmooth(SpecMTM(ts(detrend(x), start=1, deltat=1)),removeLast = 80)
    }
  )

  if(missing(col)) col <- 1:length(sp)
  if(missing(lty)) lty <- rep(1,length(sp))

  xlim <- 1/c(500,2)
```

```r
  ylim <- range(sapply(sp, function(x) c(x$lim.1,x$lim.2)))

  plot(xlim, ylim, log="xy", axes=FALSE, xlab="", ylab="", type="n")
  grid()

  for (i in 1:length(sp))
  {
    plot_mean_conf(sp[[i]]$freq,sp[[i]]$spec,
                   sp[[i]]$lim.2,sp[[i]]$lim.1,
                   lty=lty[i], col=col[i])
  }
  axis(1, at=axTicks(1), labels=1/axTicks(1))
  axis(2)
  mtext(side=1, "Period [years]", line=2)
  mtext(side=2, "Power Spectral Density", line=2)
  abline(v=0, col="grey", lwd=2)
  box()
}

generate_eof <- function(path.file, savedir, standardize.anom=T)
{
  if(!dir.exists(savedir)) system(paste("mkdir",savedir))

  # Define paths
  filename <- sub('\\..*$', '', basename(path.file))

  if(standardize.anom)
  {
    # Compute standardized anomaly
    path.anom <- paste0(savedir,"/",filename,"_stdanomaly.nc")
    system(paste("cdo div -sub",path.file,"-timmean",path.file,
                        "-timstd",path.file,
                      path.anom))
  } else {
    # Compute anomaly
    path.anom <- paste0(savedir,"/",filename,"_anomaly.nc")
    system(paste("cdo sub",path.file,"-timmean",path.file,path.anom))
  }

  # Compute eof
  system(paste0("export CDO_WEIGHT_MODE=off; cdo eof,40 ",path.anom," ",
                  savedir,"/",filename,"_eval.nc ",
                  savedir,"/",filename,"_eof.nc"))

  # Compute eofcoeff
  system(paste0("export CDO_WEIGHT_MODE=off; ",
                "cdo eofcoeff ",savedir,"/",filename,"_eof.nc ",
                  path.anom," ",
                  savedir,"/",filename,"_eofcoeff_"))
}
```

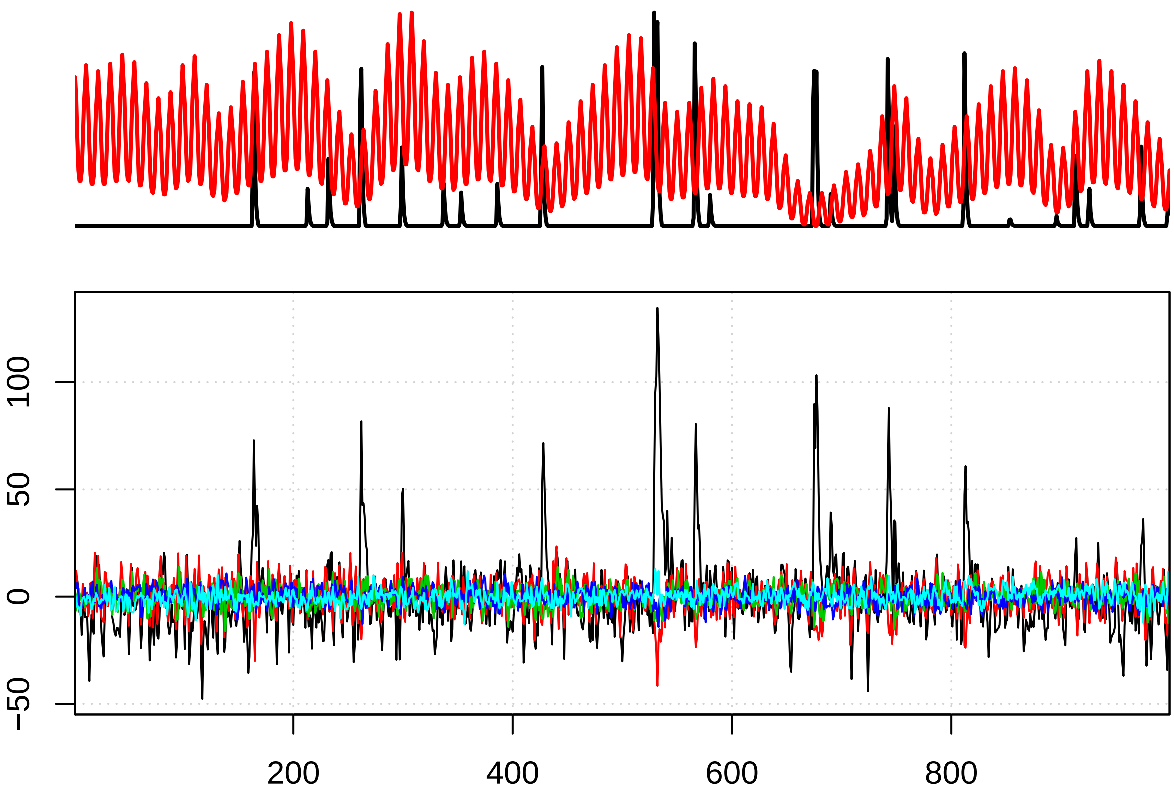
```r
runs <- c("R1","R2")

for (run in runs)
```

```r
{
  path.file <- paste0("T2m_",run,"_ym_1stMill.nc")

  filename <-sub('\\..*$', '', basename(path.file))

  system(paste0("cdo setclonlatbox,0,-180,180,30,90 ",
                "-setclonlatbox,0,-180,180,-90,-30 ",
                path.file," ",filename,"_masked.nc"))
}

system(paste("cdo ensmean",
              "T2m_R1_ym_1stMill_masked.nc",
              "T2m_R2_ym_1stMill_masked.nc",
              "T2m_ym_1stMill_masked_ensmean.nc"))

dir <- "eof_masked_ensmean"
if(!dir.exists(dir))
{
  generate_eof("T2m_ym_1stMill_masked_ensmean.nc",
              dir, standardize.anom = F)
}
```

```r
ncvarname <- "T2m"
dir <- "eof_masked_ensmean/"

filename.eofcoeffs <- list.files(dir,pattern="eofcoeff_")

eofcoeffs <- list()
for (i in 1:length(filename.eofcoeffs))
{
  nc <- nc_open(paste0(dir,filename.eofcoeffs[i]))
  eofcoeffs[[filename.eofcoeffs[i]]] <- ncvar_get(nc, ncvarname)
  nc_close(nc)
}

nc <- nc_open("Solar_forcing_1st_mill.nc")
TSI <- ncvar_get(nc, "TSI")
nc_close(nc)

nc <- nc_open("Volc_Forc_AOD_1st_mill.nc")
AOD <- ncvar_get(nc, "AOD")
nc_close(nc)
```

```r
ind <- 1:5

x <- 1:length(AOD)
xlim <- range(x)
ylim <- range(unlist(eofcoeffs[ind]))

omega <- 1/26
lowpass <- c(1)#Lowpass(omega, 1/omega+1)

layout(matrix(1:2,2,1),heights=c(0.5,1),widths=1)
par(mar=c(0.5,3,1,1), oma=c(0,0,0,0))
```

```r
#plot forcing
plot(x,filter(AOD,lowpass), type="l", lwd=2, col=1, axes=F, ylab="", xaxs="i")
par(new=T)
plot(x,filter(TSI,lowpass), type="l", lwd=2, col=2, axes=F, ylab="",xlab="", xaxs="i")

par(mar=c(3,3,1,1))
plot(xlim, ylim, xlab="Year", ylab="T2m eofcoeff", type="n", xaxs="i")
grid()

for (i in ind)
{
  lines(x,filter(eofcoeffs[[i]],lowpass), lty=1, col=i)
}
abline(v=0, col="grey", lwd=2)
box()
```




```r
omega <- 1/26
lowpass <- Lowpass(omega, 1/omega+1)

coeffcorr <- sapply(eofcoeffs, function(x)
  {
    (ccf(na.contiguous(filter(TSI,lowpass)),
      na.contiguous(filter(x,lowpass)),
      lag=0, plot=F)$acf)[1,1,1]
  }
)
coeffcorr <- rev(sort(coeffcorr))
```

```
which.max(coeffcorr)
```

```
## T2m_ym_1stMill_masked_ensmean_eofcoeff_00013.nc
##                                                1
```

```
nc <- nc_open("Solar_forcing_1st_mill.nc")
T2m <- ncvar_get(nc, "TSI")
nc_close(nc)

plot_spectra(eofcoeffs[1:10], col=1:10)
```