# Historical and Ace2 PCR results, Culex pipiens complex

Norah Saarman

2024-02-27

R build: Geospatial 4.4.0

## Import species/hybrid counts per site

```
counts <- read.csv("../data/Barr1957_plusLit_plusThisStudy.txt", sep = "\t")
countsSp <- SpatialPoints(coords = cbind(counts$long,counts$lat))
```

Genotypes in p/q notation: Cx. pipiens = pp Cx. quinquefasciatus = qq

## Create a dataframe of counts

```
# start data frame and name fields
countsDf <- as.data.frame(counts[,c(2,1,4,5,6,9,10,7,8)])
names(countsDf) <- c("locality","site","pp","pq","qq","latitude","longitude","year","h_index")

# name rows
rownames(countsDf)<- countsDf$site
```

## Pie charts on a map

Convert counts to proportions (frequency):

```
freqsDf <- as.data.frame(countsDf[,c("pp","pq","qq")])
freqsDf <- as.matrix.data.frame(t(apply(freqsDf, 1, function(row) row / sum(row))))
```

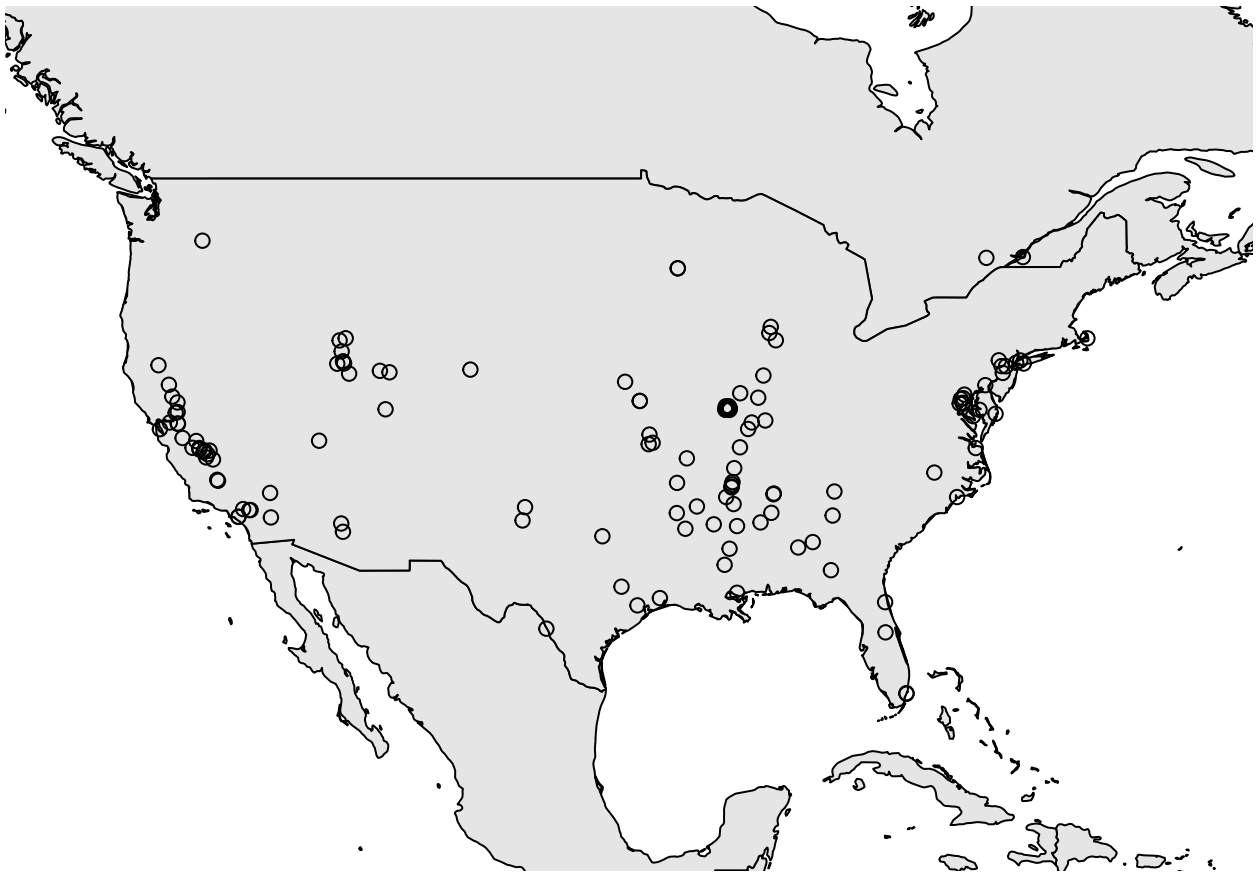
One pie chart at a time, to check code

Plot points on map to check data:

```
# Set xpd to NA to allow for plotting in the margins
par(xpd = NA)

#create and plot coord = long, lat
coord <- as.data.frame(countsDf[,c("longitude","latitude")])

#plot coordinates onto map
map("usa")
map(add = T, col = "grey90", fill = TRUE)
points(coord,col="black",cex=1)
```



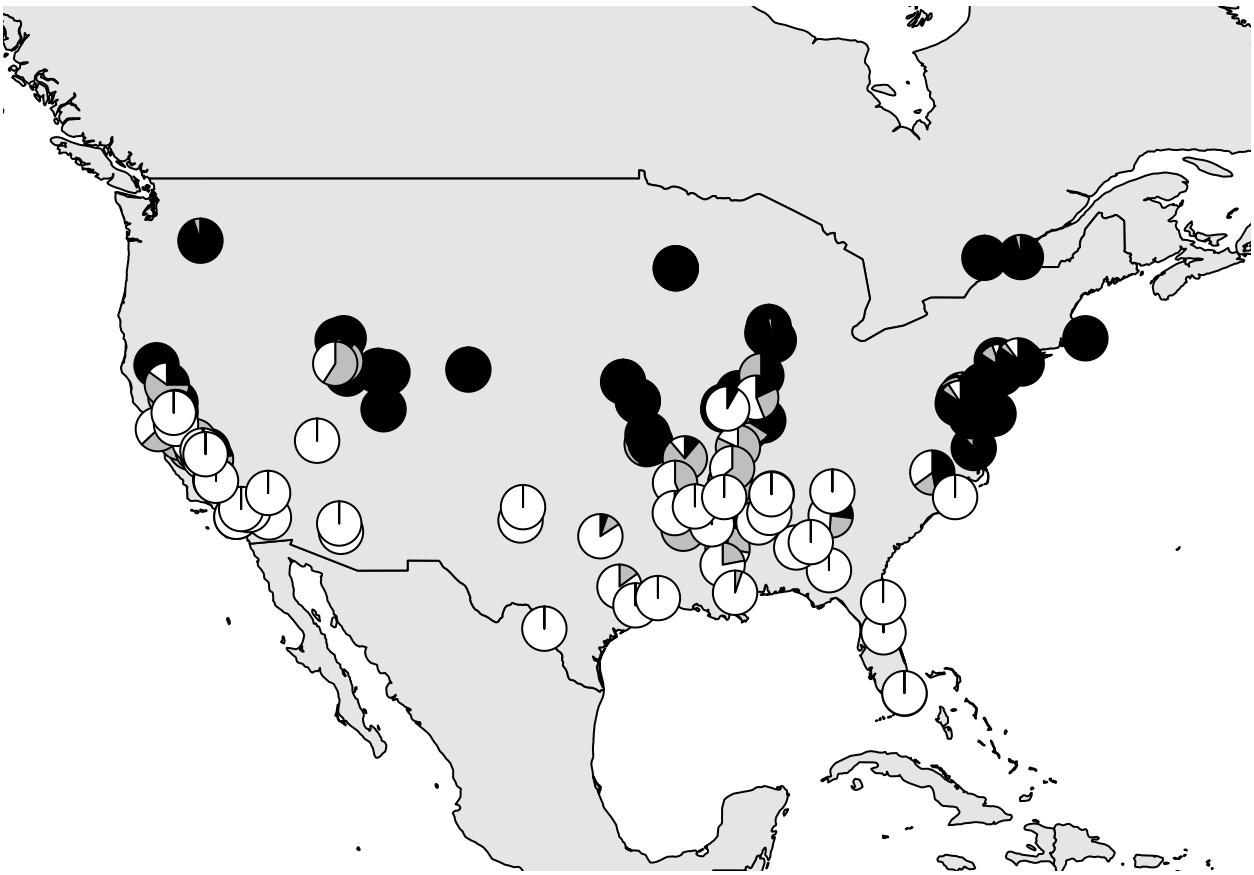
Add pies to map following: "http://membres-timc.imag.fr/Olivier.Francois/Conversion.R"

```
# Open PDF device
#pdf("../figs/ace2_pies_BarrLitThis.pdf", width = 8, height = 6)

# Set xpd to NA to allow for plotting in the margins
par(xpd = NA)

# Determine plot order by descending h_index
plot_order <- rev(order(countsDf$h_index))

# plot pies onto map
map("usa")
map(add = T, col = "grey90", fill = TRUE)
for (i in plot_order){
  add.pie(z = freqsDf[i,],
          x = coord[i,1],
          y = coord[i,2],
          clockwise=TRUE,
          labels = "",
          col = c("black","grey","white"),
          cex = 1, radius = 1 )
}
```



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

Plot latitude vs h-index excluding 0s and 1s:

```
library(ggplot2)

df <- countsDf[countsDf$h_index>0 & countsDf$h_index<1,]

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

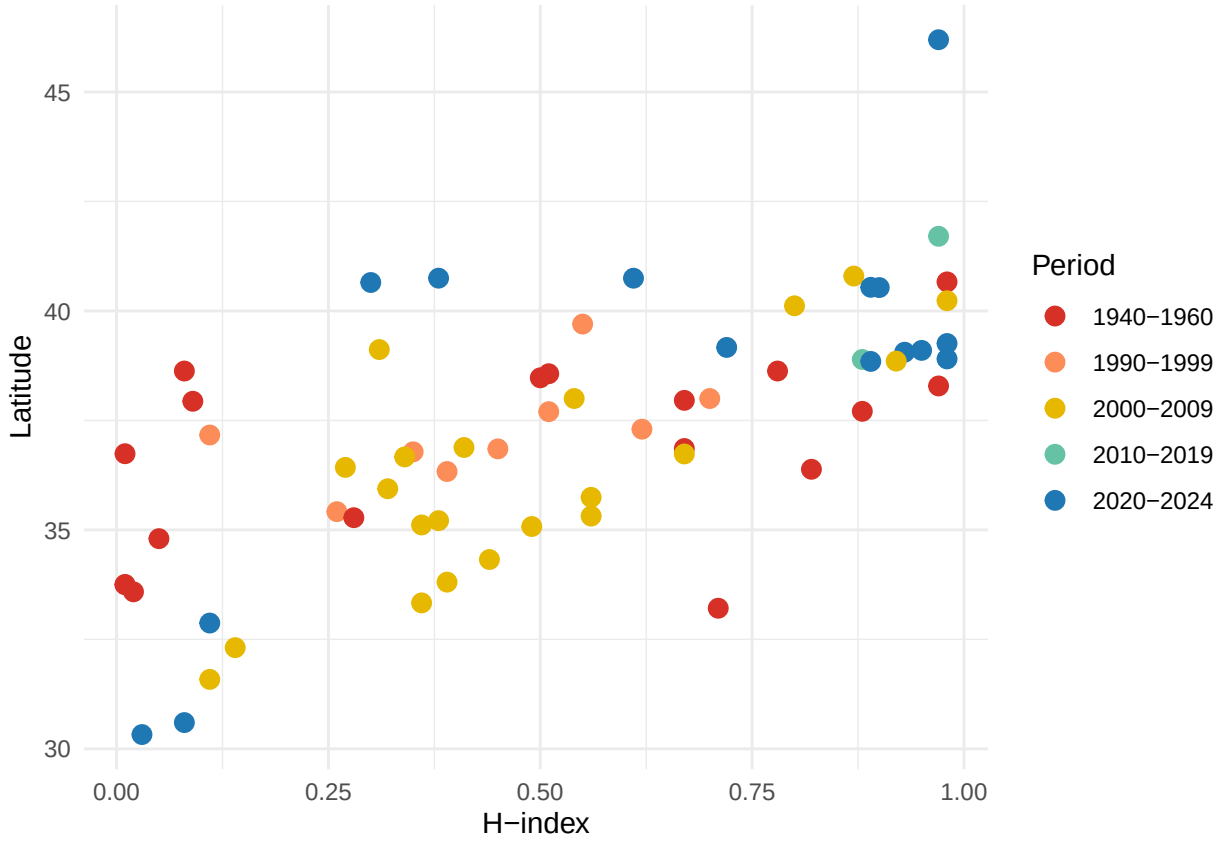
```
df <- df %>%
  mutate(
    period = case_when(
      year >= 1940 & year <= 1960 ~ "1940-1960",
      year >= 1990 & year <= 1999 ~ "1990-1999",
      year >= 2000 & year <= 2009 ~ "2000-2009",
      year >= 2010 & year <= 2019 ~ "2010-2019",
      year >= 2020 & year <= 2024 ~ "2020-2024",
```

```
      TRUE ~ "Other"  # fallback for unexpected years
    )
  )

# Optional: custom colors to match the legend in your image
period_colors <- c(
  "1940-1960" = "#d73027",  # red
  "1990-1999" = "#fc8d59",  # orange
  "2000-2009" = "#e6b800",  # yellowish
  "2010-2019" = "#66c2a5",  # greenish teal
  "2020-2024" = "#1f78b4"   # blue
)

p <- ggplot(df, aes(x = h_index, y = latitude, color = period)) +
  geom_point(size = 3) +
  scale_color_manual(values = period_colors) +
  theme_minimal() +
  labs(x = "H-index", y = "Latitude", color = "Period")

p
```




```
# Save to PDF
#ggsave("../figs/hindex_latitude_by_period.pdf", plot = p, width = 8, height = 6)
```

Add a line of best fit per period

```
library(ggplot2)
library(dplyr)
```

```r
# Example: assuming your df has h_index, latitude, and period

ggplot(df, aes(x = h_index, y = latitude, color = period)) +
  geom_point(size = 2, alpha = 0.7) +
  geom_smooth(method = "lm",
              formula = y ~ poly(x, 2),  # quadratic
              se = FALSE,                # remove confidence interval
              linewidth = 1.2) +
  scale_color_manual(values = c(
    "1940-1960" = "#d73027",
    "1990-1999" = "#fc8d59",
    "2000-2009" = "#e6b800",
    "2010-2019" = "#66c2a5",
    "2020-2024" = "#1f78b4"
  )) +
  theme_minimal() +
  labs(x = "H-index", y = "Latitude", color = "Period")
```

```
## Warning: Failed to fit group 4.
## Caused by error in `poly()`:
## ! 'degree' must be less than number of unique points
```