

Historical and Ace2 PCR results, *Culex pipiens* complex, Barr + Lit + This

Norah Saarman

2024-02-27

R build: Geospatial 4.4.0

Import species/hybrid countsAll per site

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

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

Create a dataframe of countsAll

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

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

Pie charts on a map

Convert countsAll to proportions (frequency):

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

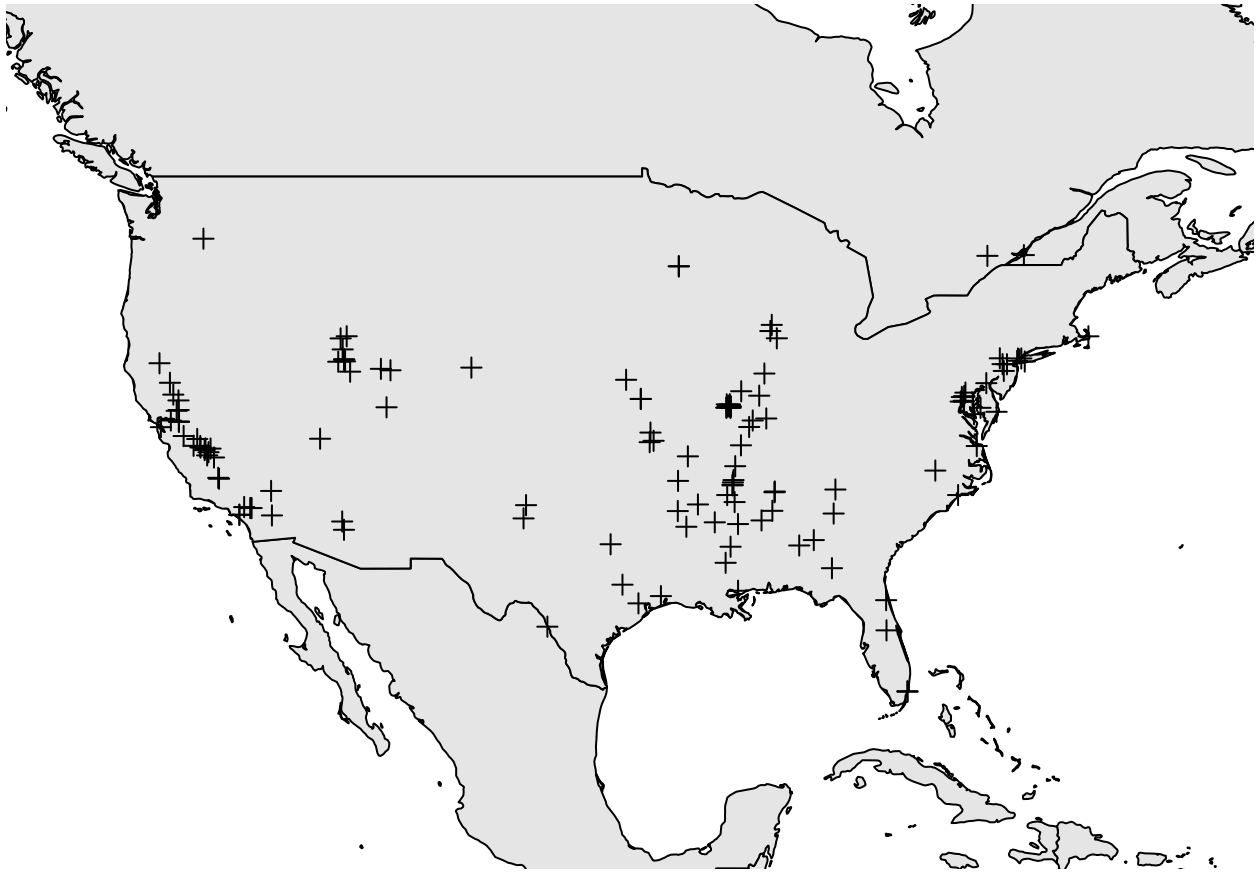
One pie chart at a time, to check code

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(countsAllDf[,c("longitude", "latitude")])

#plot coordinates onto map
map("usa")
map(add = T, col = "grey90", fill = TRUE)
points(coord, col="black", cex=1, pch=3)
```



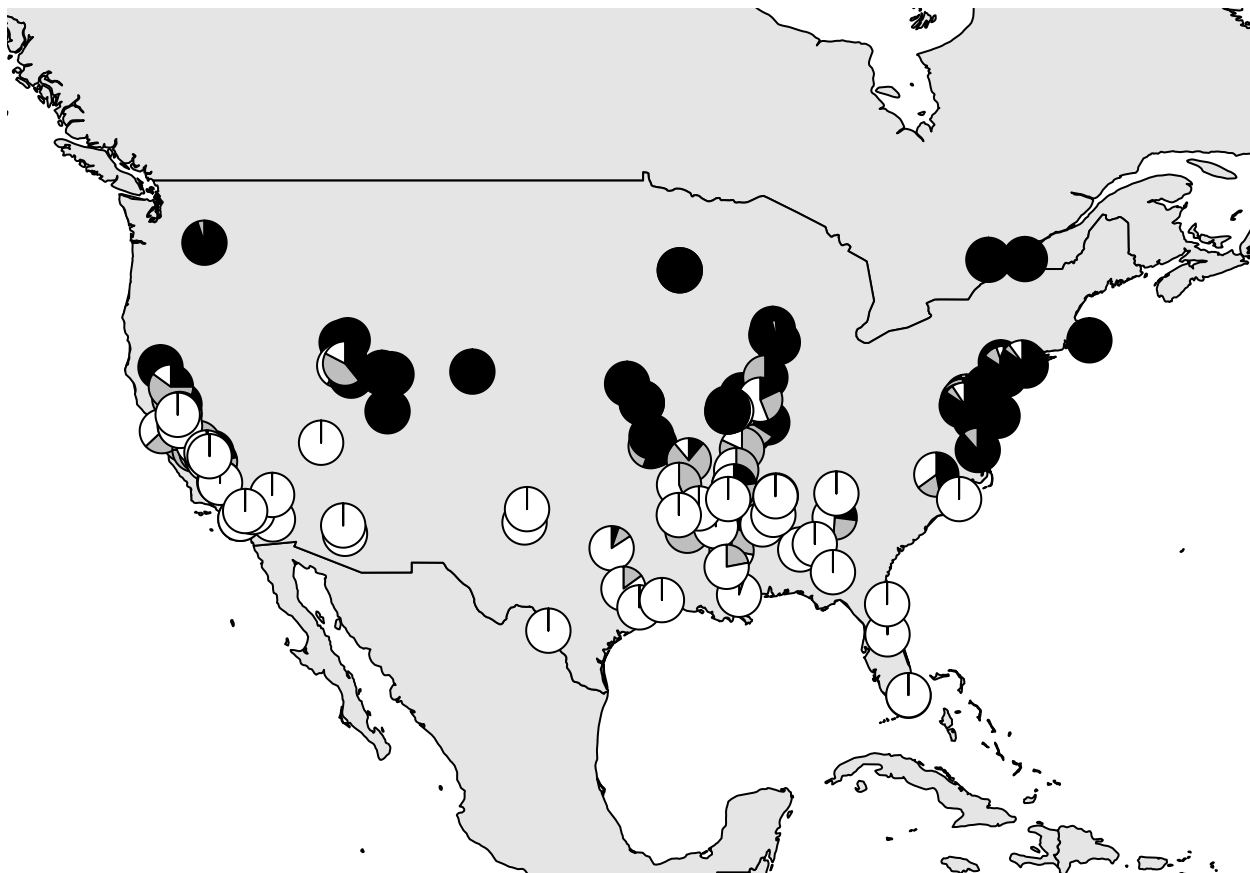
Add pies to map following: “<http://membres-timc.imag.fr/Olivier.Francois/Conversion.R>”

```
# Open PDF device
#pdf("../figs/ace2_pies_ThisBarrLit.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(countsAllDf$h_index))
plot_order <- rev(c(order(countsAllDf$h_index)[countsAllDf$year > 2021], order(countsAllDf$h_index)[countsAllDf$year <= 2021]))
plot_order <- c(rev(order(countsAllDf$h_index)[countsAllDf$year <= 2021]), rev(order(countsAllDf$h_index)[countsAllDf$year > 2021]))

# 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 with small dots for fixed sites... but doesn't look great.

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

# Add flag for fixed sites and what type of fixed (pp or qq)
freqs_fixed <- freqsDf[, 1] == 1 | freqsDf[, 3] == 1
freqs_fixed_type <- ifelse(freqsDf[, 1] == 1, "pp",
                           ifelse(freqsDf[, 3] == 1, "qq", NA))

# Plot map with expanded margins
map("usa")
map(add = TRUE, col = "grey90", fill = TRUE)

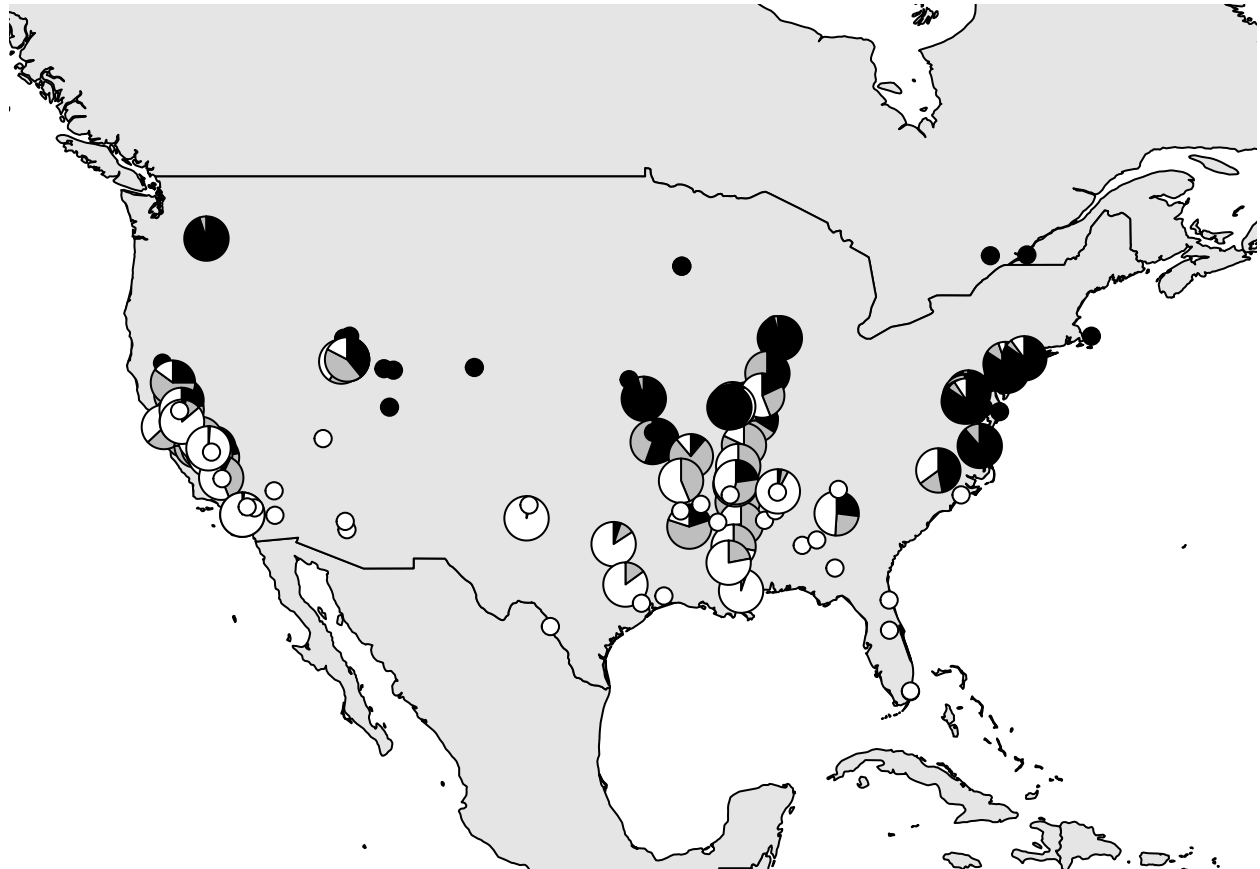
for (i in plot_order) {
  if (freqs_fixed[i]) {
    if (freqs_fixed_type[i] == "pp") {
      # Black dot
      points(coord[i,1], coord[i,2], pch = 21, bg = "black", col = "black", cex = 1.2)
    } else if (freqs_fixed_type[i] == "qq") {
      # White dot with black border
      points(coord[i,1], coord[i,2], pch = 21, bg = "white", col = "black", cex = 1.2)
    }
  } else {

```

```

    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 )
  }
}

```



1. Load and Process Species Distribution Models (SDMs)

```

# Load MaxEnt SDMs
sdm_pipiens <- raster("../gis/culex_pipiens_meansuitability.nc")

## Loading required namespace: ncdf4

sdm_quinque <- raster("../gis/culex_quinquefasciatus_meansuitability.nc")

# Threshold to binary
threshold <- 0.5
sdm_pipiens_bin <- sdm_pipiens >= threshold
sdm_quinque_bin <- sdm_quinque >= threshold

```

2. Convert SDMs to Polygons

```
# Raster to terra
sdm_papiens_v <- terra::rast(sdm_papiens_bin)
sdm_quinque_v <- terra::rast(sdm_quinque_bin)

# Raster to polygons
poly_papiens <- terra::as.polygons(sdm_papiens_v, dissolve = TRUE)
poly_quinque <- terra::as.polygons(sdm_quinque_v, dissolve = TRUE)

# Terra to sf
poly_papiens_sf <- st_as_sf(poly_papiens)
poly_quinque_sf <- st_as_sf(poly_quinque)

# Filter to presence only
names(poly_papiens_sf)[1] <- "presence"
names(poly_quinque_sf)[1] <- "presence"
poly_papiens_sf <- poly_papiens_sf %>% filter(presence == 1)
poly_quinque_sf <- poly_quinque_sf %>% filter(presence == 1)
```

3. Project to Albers Equal Area and Calculate Overlap

```
# Define CRS
aea_crs <- st_crs("+proj=aea +lat_1=29.5 +lat_2=45.5 +lat_0=23 +lon_0=-96")

# Project
poly_papiens_sf <- st_transform(poly_papiens_sf, aea_crs)
poly_quinque_sf <- st_transform(poly_quinque_sf, aea_crs)

# Overlap
overlap_sf <- st_intersection(poly_papiens_sf, poly_quinque_sf)

# Calculate areas (km²)
area_papiens_km2 <- sum(st_area(poly_papiens_sf)) / 1e6
area_quinque_km2 <- sum(st_area(poly_quinque_sf)) / 1e6
area_overlap_km2 <- sum(st_area(overlap_sf)) / 1e6
```

4. Clip to North America and Reproject to WGS84

```
# US base map
us_states <- st_as_sf(map("usa", plot = FALSE, fill = TRUE))
us_states <- st_transform(us_states, crs = aea_crs)

# Clip extent
bbox_na <- st_as_sfc(st_bbox(c(xmin = -170, xmax = -50, ymin = 5, ymax = 85), crs = st_crs(4326)))
bbox_na_sf <- st_transform(bbox_na, crs = aea_crs)

# Clip
poly_papiens_sf <- st_intersection(st_make_valid(poly_papiens_sf), bbox_na_sf)
poly_quinque_sf <- st_intersection(st_make_valid(poly_quinque_sf), bbox_na_sf)
overlap_sf <- st_intersection(st_make_valid(overlap_sf), bbox_na_sf)
```

```

# Reproject to WGS84
poly_papiens_ll <- st_transform(poly_papiens_sf, 4326)
poly_quinque_ll <- st_transform(poly_quinque_sf, 4326)
overlap_ll <- st_transform(overlap_sf, 4326)

```

6. Merge Nearby Overlapping zones

```

# Reproject overlap
overlap_ll_proj <- st_transform(overlap_ll, aea_crs)

# Make valid and extract polygons
overlap_valid <- st_make_valid(overlap_ll_proj)
overlap_polygons <- st_collection_extract(overlap_valid, "POLYGON")
overlap_parts <- st_cast(overlap_polygons, "POLYGON")

# Buffer outward
buffer_dist_meters <- 25000
overlap_buffered <- st_buffer(overlap_parts, dist = buffer_dist_meters)
overlap_buffered <- st_make_valid(overlap_buffered)

# Merge touching patches
overlap_combined <- st_union(overlap_buffered)
overlap_combined <- st_make_valid(overlap_combined)

# Buffer inward
overlap_combined <- st_buffer(overlap_combined, dist = -buffer_dist_meters)
overlap_combined <- st_make_valid(overlap_combined)

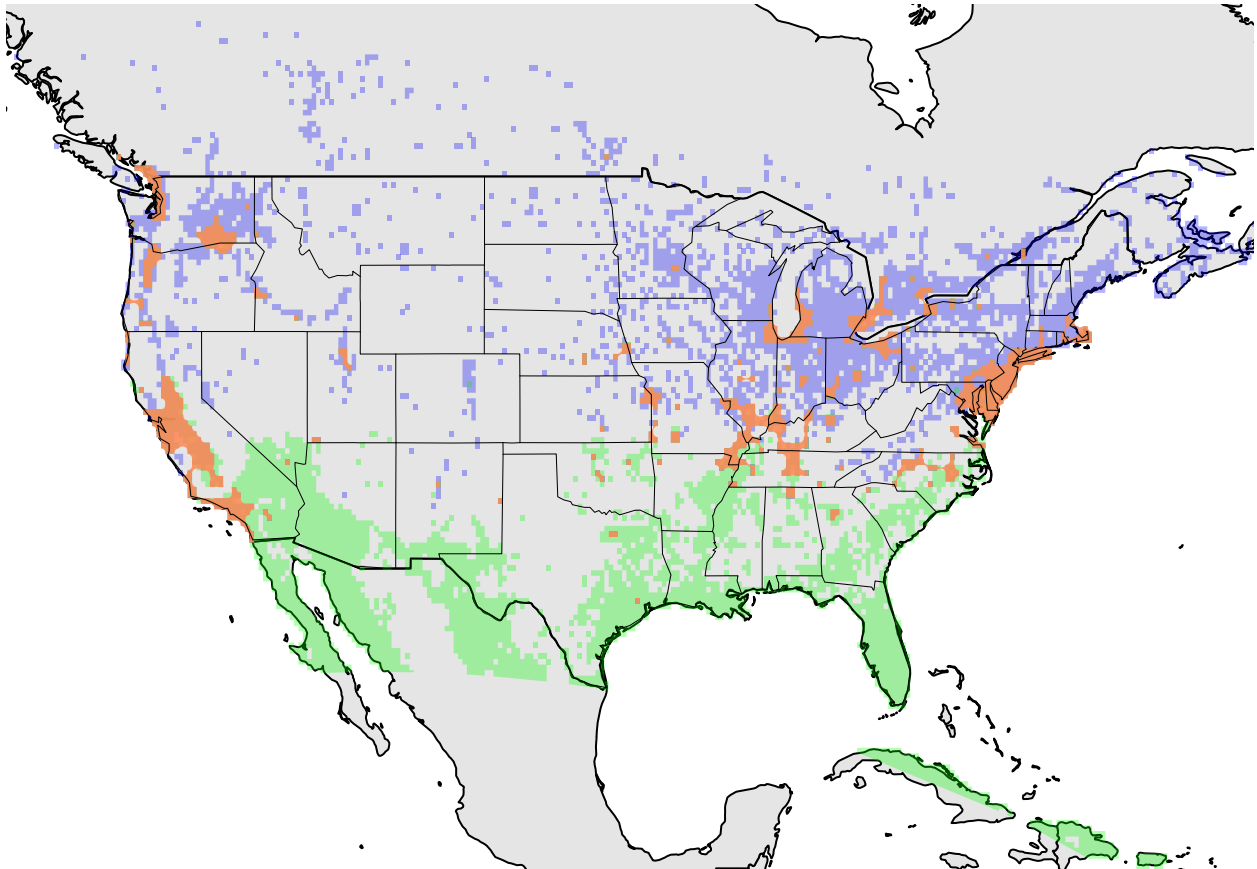
# Finalize
overlap_combined <- st_cast(overlap_combined, "MULTIPOLYGON")
overlap_combined <- st_transform(overlap_combined, 4326)

#pdf("../figs/overlap_SDM_ThisBarrLit.pdf", width = 8, height = 6)
par(xpd = NA)
map("usa")
map(add = TRUE, col = "grey90", fill = TRUE)

plot(st_geometry(poly_papiens_ll), col = rgb(0, 0, 1, 0.3), border = NA, add = TRUE)
plot(st_geometry(poly_quinque_ll), col = rgb(0, 1, 0, 0.3), border = NA, add = TRUE)
plot(st_geometry(overlap_combined), col = adjustcolor("#fc8d59", alpha.f = 0.9), border = NA, add = TRUE)

map("state", add = TRUE, col = "black", lwd = 0.5)

```



```
# Add sampling points, but looks messy:
# points(coord,col="black",cex=1,pch=3)

#dev.off()
```

Predicted level of overlapping suitable habitat?

```
# -----
# Quantify predicted overlap around each sample point
# -----

# Step 1: Make sample points an sf object
samples_sf <- st_as_sf(countsAllDf, coords = c("longitude", "latitude"), crs = 4326)

# Step 2: Reproject sample points to Albers Equal Area CRS
samples_sf <- st_transform(samples_sf, crs = aea_crs)

# Step 3: Buffer each point by 30 km (30,000 meters)
sample_buffers <- st_buffer(samples_sf, dist = 30000)

# Step 4: Prepare overlap_combined layer in same CRS
overlap_combined_proj <- st_transform(overlap_combined, crs = aea_crs)

# Step 5: Calculate proportion of buffer area that overlaps
predicted_overlap_prop <- sapply(1:nrow(sample_buffers), function(i) {
  buf <- sample_buffers[i,]
  intersection <- st_intersection(buf, overlap_combined_proj)
```

```

if (nrow(intersection) == 0) {
  # No overlap
  overlap_area <- 0
} else {
  # Sum all overlapping parts
  overlap_area <- sum(st_area(intersection))
}

buffer_area <- st_area(buf)

# Return proportion
as.numeric(overlap_area) / as.numeric(buffer_area)
})

# Step 6: Add predicted overlap to countsAllDf
countsAllDf$predicted_overlap <- predicted_overlap_prop

```

Predicted H-index

Now, add directionality with 0 = quinq, 1 = pip, to estimate “predicted_h_index”

Predicted h_index based on **Current** SDM habitat MaxEnt models also included, although I wonder about the logic of this, since climate HAS changed?

```

# -----
# New: Predict directional hybrid index from habitat around each sampling site
# -----

# Step 1: Prepare the pipiens-only and quinque-only polygons
# (We already have poly_pipiens_ll and poly_quinque_ll, but need to transform)
poly_pipiens_proj <- st_transform(poly_pipiens_ll, crs = aea_crs)
poly_quinque_proj <- st_transform(poly_quinque_ll, crs = aea_crs)
overlap_combined_proj <- st_transform(overlap_combined, crs = aea_crs) # already done above

# Step 2: Create pip-only and quinque-only polygons (remove overlap area)
pip_only <- st_difference(poly_pipiens_proj, overlap_combined_proj)
quinque_only <- st_difference(poly_quinque_proj, overlap_combined_proj)

# Step 3: Calculate areas for each sample buffer
library(units) # make sure units package is loaded

## udunits database from /usr/share/xml/udunits/udunits2.xml
predicted_h_index <- sapply(1:nrow(sample_buffers), function(i) {
  buf <- sample_buffers[i, ]

  pip_intersect <- st_intersection(buf, pip_only)
  quinque_intersect <- st_intersection(buf, quinque_only)
  overlap_intersect <- st_intersection(buf, overlap_combined_proj)

  pip_area <- if (nrow(pip_intersect) == 0) units::set_units(0, "m^2") else sum(st_area(pip_intersect))
  quinque_area <- if (nrow(quinque_intersect) == 0) units::set_units(0, "m^2") else sum(st_area(quinque_intersect))
  overlap_area <- if (nrow(overlap_intersect) == 0) units::set_units(0, "m^2") else sum(st_area(overlap_intersect))

```



```

total_area <- pip_area + cinque_area + overlap_area

if (as.numeric(total_area) == 0) {
  return(NA) # no habitat found
} else {
  pred_h <- (pip_area + 0.5 * overlap_area) / total_area
  return(as.numeric(pred_h)) # strip units at the end
}
})
# Step 4: Add predicted_h_index to countsAllDf
countsAllDf$predicted_h_index <- predicted_h_index

# Step 5: Quick check and save
(summary_table <- countsAllDf[, c("site", "latitude", "longitude", "h_index", "predicted_overlap", "pre

##
## 001.ByroWA.2023 001.ByroWA.2023 46.19300 -119.89900 0.97 0.841121564
## 002.StAnQC.1944 002.StAnQC.1944 45.45600 -73.63100 1.00 0.361104399
## 003.OttaON.1957 003.OttaON.1957 45.42090 -75.69010 1.00 0.192520104
## 004.StPaMN.1953 004.StPaMN.1953 44.94970 -93.09310 1.00 0.311265050
## 005.StPaMN.1953 005.StPaMN.1953 44.94970 -93.09310 1.00 0.311265050
## 006.GreaIL.1944 006.GreaIL.1944 42.30930 -87.84970 1.00 0.687216478
## 007.CookIL.2023 007.CookIL.2023 42.03176 -87.93087 1.00 0.999322243
## 008.CachUT.2023 008.CachUT.2023 41.79696 -111.82005 1.00 0.000000000
## 009.BarnMA.2023 009.BarnMA.2023 41.79362 -69.99427 1.00 0.824884441
## 010.TrumIL.2012 010.TrumIL.2012 41.70490 -87.56440 0.97 0.794448383
## 011.BoxEUT.2024 011.BoxEUT.2024 41.69945 -112.16386 1.00 0.000000000
## 012.OgdeUT.2024 012.OgdeUT.2024 41.20359 -112.04803 1.00 0.475093146
## 013.OxfoNJ.1953 013.OxfoNJ.1953 40.80320 -74.98960 1.00 0.008691163
## 014.FortNY.2007 014.FortNY.2007 40.79600 -73.77800 0.87 1.000000000
## 015.SaltUT.2023 015.SaltUT.2023 40.74805 -111.96788 0.38 0.485085123
## 015.SaltUT.2018 015.SaltUT.2018 40.74805 -111.96788 0.93 0.485085123
## 016.SaltUT.2021 016.SaltUT.2021 40.74750 -111.96760 0.61 0.484910810
## 017.GoveNY.1953 017.GoveNY.1953 40.69150 -74.01240 1.00 1.000000000
## 018.SaltUT.1957 018.SaltUT.1957 40.66320 -111.91030 1.00 0.470535723
## 019.FreeNY.1953 019.FreeNY.1953 40.65760 -73.58360 1.00 0.895548023
## 020.BensUT.2021 020.BensUT.2021 40.64940 -112.29680 0.30 0.086471138
## 021.ReddCA.2003 021.ReddCA.2003 40.58040 -122.37710 1.00 0.000000000
## 022.HuntNJ.2023 022.HuntNJ.2023 40.53959 -74.83462 0.89 0.427789251
## 023.SomeNJ.2023 023.SomeNJ.2023 40.53358 -74.58610 0.90 0.695388858
## 024.FortCO.2023 024.FortCO.2023 40.38428 -104.78940 1.00 0.000000000
## 025.UteTUT.2024 025.UteTUT.2024 40.32890 -109.89110 1.00 0.000000000
## 026.VernUT.2024 026.VernUT.2024 40.26180 -109.35110 1.00 0.000000000
## 027.TrenNJ.2007 027.TrenNJ.2007 40.23300 -74.76600 0.98 0.987008509
## 028.ProvUT.2024 028.ProvUT.2024 40.20065 -111.62731 1.00 0.186683924
## 029.ChamIL.2005 029.ChamIL.2005 40.11700 -88.26500 0.80 0.000000000
## 030.SeneKS.1951 030.SeneKS.1951 39.83420 -96.06420 1.00 0.000000000
## 031.ChicCA.1992 031.ChicCA.1992 39.70000 -121.79000 0.55 0.790033112
## 032.NewaDE.1953 032.NewaDE.1953 39.68280 -75.75160 1.00 0.676133581
## 033.RaymIL.1939 033.RaymIL.1939 39.31950 -89.57200 1.00 0.000000000
## 034.ClarMD .2020 034.ClarMD .2020 39.25640 -76.92760 0.98 0.891844198
## 035.SuttCA.2023 035.SuttCA.2023 39.16647 -121.59845 0.72 0.914493895
## 036.EffiIL.2005 036.EffiIL.2005 39.11800 -88.55700 0.31 0.000000000
## 037.Old MD.2020 037.Old MD.2020 39.09920 -77.00180 0.95 0.961456368

```

## 038.RockMD.2023	038.RockMD.2023	39.05775	-77.13055	0.93	0.874845423
## 039.LawrKS.1955	039.LawrKS.1955	38.97190	-95.23590	0.98	0.422164069
## 040.LawrKS.1951	040.LawrKS.1951	38.97190	-95.23590	1.00	0.422164069
## 041.LincCA.2023	041.LincCA.2023	38.90447	-121.30633	0.98	0.855749203
## 042.DistDC.2019	042.DistDC.2019	38.89130	-77.02600	0.88	1.000000000
## 043.FairVA.2007	043.FairVA.2007	38.85400	-77.19500	0.92	0.959820119
## 044.AnacDC.2020	044.AnacDC.2020	38.84720	-77.01180	0.89	1.000000000
## 045.BadeMO.1952	045.BadeMO.1952	38.72870	-90.21780	0.92	0.966786790
## 046.StAnMO.1952	046.StAnMO.1952	38.72810	-90.38790	1.00	0.882733698
## 047.NortMO.1952	047.NortMO.1952	38.70490	-90.21740	1.00	0.933525874
## 048.UnivMO.1952	048.UnivMO.1952	38.65690	-90.31030	1.00	0.799896752
## 049.RichMO.1952	049.RichMO.1952	38.62830	-90.31910	1.00	0.731860647
## 050.StLoMO.1952	050.StLoMO.1952	38.62800	-90.19100	0.73	0.791288629
## 051.StLoMO.1953	051.StLoMO.1953	38.62800	-90.19100	0.08	0.791288629
## 052.EastIL.1944	052.EastIL.1944	38.62690	-90.15970	0.95	0.788980118
## 053.EastIL.1943	053.EastIL.1943	38.62690	-90.15970	1.00	0.788980118
## 054.MoabUT.2024	054.MoabUT.2024	38.59597	-109.57376	1.00	0.000000000
## 055.KirkMO.1952	055.KirkMO.1952	38.58010	-90.40690	1.00	0.562205677
## 056.KirkMO.1952	056.KirkMO.1952	38.58010	-90.40690	1.00	0.562205677
## 057.CambMA.1945	057.CambMA.1945	38.57150	-76.07630	1.00	0.992826084
## 058.CahoIL.1943	058.CahoIL.1943	38.56620	-90.17940	0.50	0.654560902
## 059.SacrCA.1947	059.SacrCA.1947	38.47320	-121.29810	0.41	0.895758193
## 060.KingCA.1957	060.KingCA.1957	38.43640	-121.40830	0.00	0.992391977
## 061.BainMD.1945	061.BainMD.1945	38.39180	-75.17350	1.00	0.677493194
## 062.PatuMD.1945	062.PatuMD.1945	38.28680	-76.44360	1.00	0.384456761
## 063.CarmIL.1941	063.CarmIL.1941	38.09090	-88.15860	1.00	0.838270062
## 064.OaklCA.1992	064.OaklCA.1992	38.00000	-121.73800	0.70	1.000000000
## 065.BentIL.2005	065.BentIL.2005	38.00000	-88.92400	0.54	0.999558899
## 066.StocCA.1957	066.StocCA.1957	37.95770	-121.29080	0.50	1.000000000
## 067.SanJCA.1957	067.SanJCA.1957	37.93730	-121.27740	0.08	1.000000000
## 068.CrabIL.1942	068.CrabIL.1942	37.70770	-89.12470	1.00	0.673832475
## 069.AlamCA.1992	069.AlamCA.1992	37.70000	-122.30000	0.51	0.972419966
## 070.FronKS.1951	070.FronKS.1951	37.45560	-94.68910	1.00	0.362863363
## 071.NewmCA.1992	071.NewmCA.1992	37.30000	-121.01700	0.62	1.000000000
## 072.StGeUT.2023	072.StGeUT.2023	37.17900	-113.32000	0.00	0.304021108
## 073.ChowCA.1991	073.ChowCA.1991	37.16700	-120.25000	0.11	0.799629650
## 074.JoplMO.1952	074.JoplMO.1952	37.08420	-94.51330	0.85	0.531934812
## 075.BaxtKS.1957	075.BaxtKS.1957	37.02360	-94.73520	0.78	0.334707893
## 076.SikeMO.2005	076.SikeMO.2005	36.88200	-89.58700	0.41	0.844358543
## 077.FireCA.1952	077.FireCA.1952	36.85880	-120.45600	0.50	1.000000000
## 078.MadeCA.1992	078.MadeCA.1992	36.85000	-120.07800	0.45	0.976789348
## 079.NorfVA.1957	079.NorfVA.1957	36.84940	-76.29000	0.94	0.653304608
## 080.KermCA.1992	080.KermCA.1992	36.78300	-120.06600	0.35	1.000000000
## 081.FresCA.1953	081.FresCA.1953	36.73940	-119.78480	0.01	0.922491087
## 082.CentCA.2000	082.CentCA.2000	36.73390	-119.49830	0.67	0.483149003
## 083.ReedCA.2000	083.ReedCA.2000	36.66670	-119.83330	0.34	0.980749996
## 084.SelmCA.1953	084.SelmCA.1953	36.57080	-119.61210	0.00	0.796428028
## 085.FresCA.2005	085.FresCA.2005	36.42700	-119.68900	0.27	0.976260559
## 086.BullAR.1957	086.BullAR.1957	36.38400	-92.58160	0.50	0.000000000
## 087.VisaCA.1992	087.VisaCA.1992	36.33300	-119.30300	0.39	0.480897839
## 088.BlytAR.2005	088.BlytAR.2005	35.94000	-89.91100	0.32	0.746968916
## 089.RaleNC.2007	089.RaleNC.2007	35.74300	-78.62600	0.56	0.395767799
## 092.BakeCA.1992	092.BakeCA.1992	35.41600	-119.06600	0.26	0.725368806
## 093.BakeCA.1953	093.BakeCA.1953	35.37390	-119.01950	0.00	0.729994833

## 094.ShelTN.2002	094.ShelTN.2002	35.31520	-90.00620	0.56	0.308644133
## 095.RussAR.1957	095.RussAR.1957	35.27840	-93.13380	0.22	0.000000000
## 096.MempTN.2002	096.MempTN.2002	35.21220	-90.03840	0.38	0.441559622
## 097.MempTN.2002	097.MempTN.2002	35.11310	-90.05720	0.36	0.447261866
## 098.ShelTN.2005	098.ShelTN.2005	35.07700	-90.06000	0.49	0.446873395
## 099.LakeGA.1955	099.LakeGA.1955	34.88900	-84.26130	0.00	0.000000000
## 100.SanBCA.1951	100.SanBCA.1951	34.82530	-116.08330	0.00	0.000000000
## 101.FlorAL.1954	101.FlorAL.1954	34.79980	-87.67730	0.05	0.000000000
## 102.ShefAL.1953	102.ShefAL.1953	34.76510	-87.69860	0.00	0.000000000
## 103.TuniMS.1945	103.TuniMS.1945	34.63990	-90.36240	0.00	0.000000000
## 104.CaLeNC.1957	104.CaLeNC.1957	34.63970	-77.34180	0.00	0.000000000
## 105.BateMS.2005	105.BateMS.2005	34.32400	-89.94500	0.44	0.000000000
## 106.PineAR.1957	106.PineAR.1957	34.21570	-92.01400	0.00	0.000000000
## 107.PlaiTX.1953	107.PlaiTX.1953	34.18480	-101.70680	0.00	0.000000000
## 108.CucaCA.1957	108.CucaCA.1957	34.09920	-117.60220	0.00	0.888801175
## 109.RedlCA.1951	109.RedlCA.1951	34.05500	-117.18270	0.00	0.556155598
## 110.LomaCA.1957	110.LomaCA.1957	34.05380	-117.26110	1.00	0.687196135
## 111.WinfAL.1957	111.WinfAL.1957	33.92820	-87.81630	0.00	0.000000000
## 112.GurdAR.1957	112.GurdAR.1957	33.91820	-93.14880	0.00	0.000000000
## 113.AtlaGA.2007	113.AtlaGA.2007	33.80700	-84.36000	0.39	0.666059179
## 114.OranCA.1953	114.OranCA.1953	33.75060	-117.87220	0.01	1.000000000
## 115.OranCA.1953	115.OranCA.1953	33.75060	-117.87220	0.01	1.000000000
## 116.RiveCA.1957	116.RiveCA.1957	33.72200	-116.03720	0.00	0.319824055
## 117.LubbTX.1953	117.LubbTX.1953	33.58560	-101.84700	0.02	0.000000000
## 118.ColuMS.1944	118.ColuMS.1944	33.49570	-88.42730	0.00	0.000000000
## 119.PhoeAZ.2024	119.PhoeAZ.2024	33.44844	-112.07414	0.00	0.000000000
## 120.GreeMS.1957	120.GreeMS.1957	33.41110	-91.06360	0.00	0.000000000
## 121.VaidMS.2005	121.VaidMS.2005	33.33300	-89.75300	0.36	0.000000000
## 122.ElDoAR.1953	122.ElDoAR.1953	33.21150	-92.66500	0.50	0.000000000
## 123.MariAZ.2023	123.MariAZ.2023	33.07362	-111.97377	0.00	0.000000000
## 124.DalFTX.2024	124.DalFTX.2024	32.42235	-96.93698	0.11	0.000000000
## 125.AubuAL.1957	125.AubuAL.1957	32.60990	-85.48080	0.00	0.000000000
## 126.MontAL.1957	126.MontAL.1957	32.36700	-86.30060	0.00	0.000000000
## 127.JackMS.2005	127.JackMS.2005	32.31100	-90.17400	0.14	0.000000000
## 128.BrooMS.2005	128.BrooMS.2005	31.58600	-90.45200	0.11	0.000000000
## 129.BakeGA.1953	129.BakeGA.1953	31.34390	-84.45550	0.00	0.000000000
## 130.CollTX.2023	130.CollTX.2023	30.60044	-96.26893	0.08	0.000000000
## 131.SlidLA.2024	131.SlidLA.2024	30.32700	-89.74900	0.03	0.000000000
## 132.BeauTX.1954	132.BeauTX.1954	30.08600	-94.10180	0.00	0.000000000
## 133.AnasFL.2023	133.AnasFL.2023	29.90311	-81.40074	0.00	0.000000000
## 134.HousTX.1957	134.HousTX.1957	29.75890	-95.36770	0.00	0.235793000
## 135.EaglTX.1953	135.EaglTX.1953	28.70840	-100.50390	0.00	0.000000000
## 136.OrlaFL.1957	136.OrlaFL.1957	28.54210	-81.37900	0.00	0.000000000
## 137.MiDaFL.2023	137.MiDaFL.2023	25.80141	-80.19909	0.00	0.000000000
## 138.MiamFL.1957	138.MiamFL.1957	25.77420	-80.19360	0.00	0.000000000
##	predicted_h_index				
## 001.ByroWA.2023		0.5208546			
## 002.StAnQC.1944		0.8184089			
## 003.OttaON.1957		0.9036682			
## 004.StPaMN.1953		0.8443658			
## 005.StPaMN.1953		0.8443658			
## 006.GreaIL.1944		0.5584620			
## 007.CookIL.2023		0.5003389			
## 008.CachUT.2023		1.0000000			

## 009.BarnMA.2023	0.5000000
## 010.TrumIL.2012	0.5425679
## 011.BoxEUT.2024	1.0000000
## 012.OgdeUT.2024	0.7051933
## 013.OxfoNJ.1953	0.9954287
## 014.FortNY.2007	0.5000000
## 015.SaltUT.2023	0.7010417
## 015.SaltUT.2018	0.7010417
## 016.SaltUT.2021	0.7010898
## 017.GoveNY.1953	0.5000000
## 018.SaltUT.1957	0.6879793
## 019.FreeNY.1953	0.5000000
## 020.BensUT.2021	0.9321020
## 021.ReddCA.2003	1.0000000
## 022.HuntNJ.2023	0.7861053
## 023.SomeNJ.2023	0.6523055
## 024.FortCO.2023	1.0000000
## 025.UteTUT.2024	1.0000000
## 026.VernUT.2024	1.0000000
## 027.TrenNJ.2007	0.5064957
## 028.ProvUT.2024	0.8433461
## 029.ChamIL.2005	1.0000000
## 030.SeneKS.1951	NA
## 031.ChicCA.1992	0.4068169
## 032.NewaDE.1953	0.6619332
## 033.RaymIL.1939	1.0000000
## 034.ClarMD .2020	0.5540776
## 035.SuttCA.2023	0.4925264
## 036.EffiIL.2005	1.0000000
## 037.Old MD.2020	0.5192716
## 038.RockMD.2023	0.5586669
## 039.LawrKS.1955	0.6634387
## 040.LawrKS.1951	0.6634387
## 041.LincCA.2023	0.4591176
## 042.DistDC.2019	0.5000000
## 043.FairVA.2007	0.5200899
## 044.AnacDC.2020	0.5000000
## 045.BadeMO.1952	0.5166066
## 046.StAnMO.1952	0.5583703
## 047.NortMO.1952	0.5332371
## 048.UnivMO.1952	0.5984681
## 049.RichMO.1952	0.6295175
## 050.StLoMO.1952	0.6043557
## 051.StLoMO.1953	0.6043557
## 052.EastIL.1944	0.6055099
## 053.EastIL.1943	0.6055099
## 054.MoabUT.2024	NA
## 055.KirkMO.1952	0.6968042
## 056.KirkMO.1952	0.6968042
## 057.CambMA.1945	0.5000000
## 058.CahoIL.1943	0.6727195
## 059.SacrCA.1947	0.4697888
## 060.KingCA.1957	0.4961960
## 061.BainMD.1945	0.5000000

## 062.PatuMD.1945	0.5000000
## 063.CarmIL.1941	0.5023048
## 064.OaklCA.1992	0.5000000
## 065.BentIL.2005	0.5002206
## 066.StocCA.1957	0.5000000
## 067.SanJCA.1957	0.5000000
## 068.CrabIL.1942	0.6216752
## 069.AlamCA.1992	0.5000000
## 070.FronKS.1951	0.5000000
## 071.NewmCA.1992	0.5000000
## 072.StGeUT.2023	0.4154164
## 073.ChowCA.1991	0.5000000
## 074.JoplMO.1952	0.3894218
## 075.BaxtKS.1957	0.2937069
## 076.SikeMO.2005	0.4221798
## 077.FireCA.1952	0.5000000
## 078.MadeCA.1992	0.5000000
## 079.NorfVA.1957	0.3727732
## 080.KermCA.1992	0.5000000
## 081.FresCA.1953	0.4811033
## 082.CentCA.2000	0.3687577
## 083.ReedCA.2000	0.4919062
## 084.SelmCA.1953	0.4146860
## 085.FresCA.2005	0.4881303
## 086.BullAR.1957	0.0000000
## 087.VisaCA.1992	0.2709736
## 088.BlytAR.2005	0.3734854
## 089.RaleNC.2007	0.2218396
## 092.BakeCA.1992	0.4386534
## 093.BakeCA.1953	0.4614086
## 094.ShelTN.2002	0.1543237
## 095.RussAR.1957	NA
## 096.MempTN.2002	0.2207813
## 097.MempTN.2002	0.2236339
## 098.ShelTN.2005	0.2234390
## 099.LakeGA.1955	NA
## 100.SanBCA.1951	0.0000000
## 101.FlorAL.1954	0.0000000
## 102.ShefAL.1953	0.0000000
## 103.TuniMS.1945	0.0000000
## 104.CaLeNC.1957	0.0000000
## 105.BateMS.2005	0.0000000
## 106.PineAR.1957	0.0000000
## 107.PlaiTX.1953	NA
## 108.CucaCA.1957	0.4444006
## 109.RedlCA.1951	0.3239416
## 110.LomaCA.1957	0.3695397
## 111.WinfAL.1957	NA
## 112.GurdAR.1957	0.0000000
## 113.AtlaGA.2007	0.3330306
## 114.OranCA.1953	0.5000000
## 115.OranCA.1953	0.5000000
## 116.RiveCA.1957	0.1599133
## 117.LubbTX.1953	0.0000000

```
## 118.ColuMS.1944      0.0000000
## 119.PhoeAZ.2024      0.0000000
## 120.GreeMS.1957      0.0000000
## 121.VaidMS.2005      0.0000000
## 122.ElDoAR.1953      0.0000000
## 123.MariAZ.2023      0.0000000
## 124.DalFTX.2024      0.0000000
## 125.AubuAL.1957      0.0000000
## 126.MontAL.1957      0.0000000
## 127.JackMS.2005      0.0000000
## 128.BrooMS.2005      0.0000000
## 129.BakeGA.1953      0.0000000
## 130.CollTX.2023      0.0000000
## 131.SlidLA.2024      0.0000000
## 132.BeauTX.1954      0.0000000
## 133.AnasFL.2023      0.0000000
## 134.HousTX.1957      0.1178991
## 135.EaglTX.1953      0.0000000
## 136.OrlaFL.1957      0.0000000
## 137.MiDaFL.2023      0.0000000
## 138.MiamFL.1957      0.0000000
```

```
# Save as CSV
```

```
# write.csv(summary_table, file = "../data/summary_hindex_prediction.csv", row.names = FALSE)
```

Plot h_index versus predicted_h_index, latitude, etc...

By Zone: H-index versus Predicted H-index and Latitude

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

```
##
## Attaching package: 'patchwork'

## The following object is masked from 'package:terra':
##
## area

## The following object is masked from 'package:raster':
##
## area
```

```
# Define hybrid zones:
```

```
countsAllDf$zone <- NULL
countsAllDf$zone <- case_when(
  grepl("CA|OR|WA", countsAllDf$site) ~ "West Coast",
  grepl("NJ|MD|MA|DE|CT|NY|VA|FL|QC|DC|TN|GA|NC", countsAllDf$site) ~ "East Coast",
  grepl("TX|LA|IL|LA|MN|ON|KS|MO|AR|AL|MS", countsAllDf$site) ~ "Central",
  grepl("UT|CO|AZ", countsAllDf$site) ~ "Mtn/Southwest"
)
```

```
# Control order by making it a factor
```

```

countsAllDf$zone <- factor(countsAllDf$zone, levels = c("West Coast", "Mtn/Southwest", "Central", "East

# Define zone color palette
zone_colors <- c(
  "Mtn/Southwest" = "#d73027",
  "Central" = "#6a3d9a",
  "West Coast" = "#e6b800",
  "East Coast" = "#1f78b4"
)

# Create shared color scale
zone_scale <- scale_color_manual(values = zone_colors, name = "Zone:")

# Shared theme for both plots
shared_theme <- theme_minimal(base_size = 10) +
  theme(
    plot.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom", # <-- Legend at bottom
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 8),
    plot.margin = margin(10, 10, 10, 10)
  )

# -----
# H-index vs Predicted H-index
# -----

r2_pred <- countsAllDf %>%
  group_by(zone) %>%
  summarise(r2 = summary(lm(predicted_h_index ~ h_index))$r.squared)

p2 <- ggplot(countsAllDf, aes(x = h_index, y = predicted_h_index, color = zone)) +
  geom_point(size = 3, alpha = 0.5, shape = 16) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, linewidth = 1.2) +
  geom_text(data = r2_pred, aes(x = 0.05, y = 0.95 - as.numeric(zone) * 0.05,
    label = paste0("R2 = ", round(r2, 2)), color = zone),
    hjust = 0, size = 3.5, inherit.aes = FALSE) +
  theme_minimal() +
  labs(x = "H-index", y = "Predicted H-index (from SDM)", title = "Predicted H-index", color = "Zone:")
  xlim(0, 1) +
  ylim(0, 1) +
  scale_color_manual(values = zone_colors)
  guides(color = guide_legend(override.aes = list(alpha = 1, shape = NA, linetype = 1)))

## <Guides[1] ggproto object>
##
## colour : <GuideLegend>

# -----
# H-index vs Latitude
# -----

r2_lat <- countsAllDf %>%
  group_by(zone) %>%
  summarise(r2 = summary(lm(latitude ~ h_index))$r.squared)

```

```
p1 <- ggplot(countsAllDf, aes(x = h_index, y = latitude, color = zone)) +
  geom_point(size = 3, alpha = 0.5, shape = 16) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, linewidth = 1.2) +
  geom_text(data = r2_lat, aes(x = 0.05, y = max(countsAllDf$latitude, na.rm = TRUE) - as.numeric(zone),
    label = paste0("R2 = ", round(r2, 2)), color = zone),
    hjust = 0, size = 3.5, inherit.aes = FALSE) +
  theme_minimal() +
  labs(x = "H-index", y = "Latitude", title = "Latitude", color = "Zone:") +
  xlim(0, 1) +
  scale_color_manual(values = zone_colors)
guides(color = guide_legend(override.aes = list(alpha = 1, shape = NA, linetype = 1)))
```

```
## <Guides[1] ggproto object>
```

```
##
```

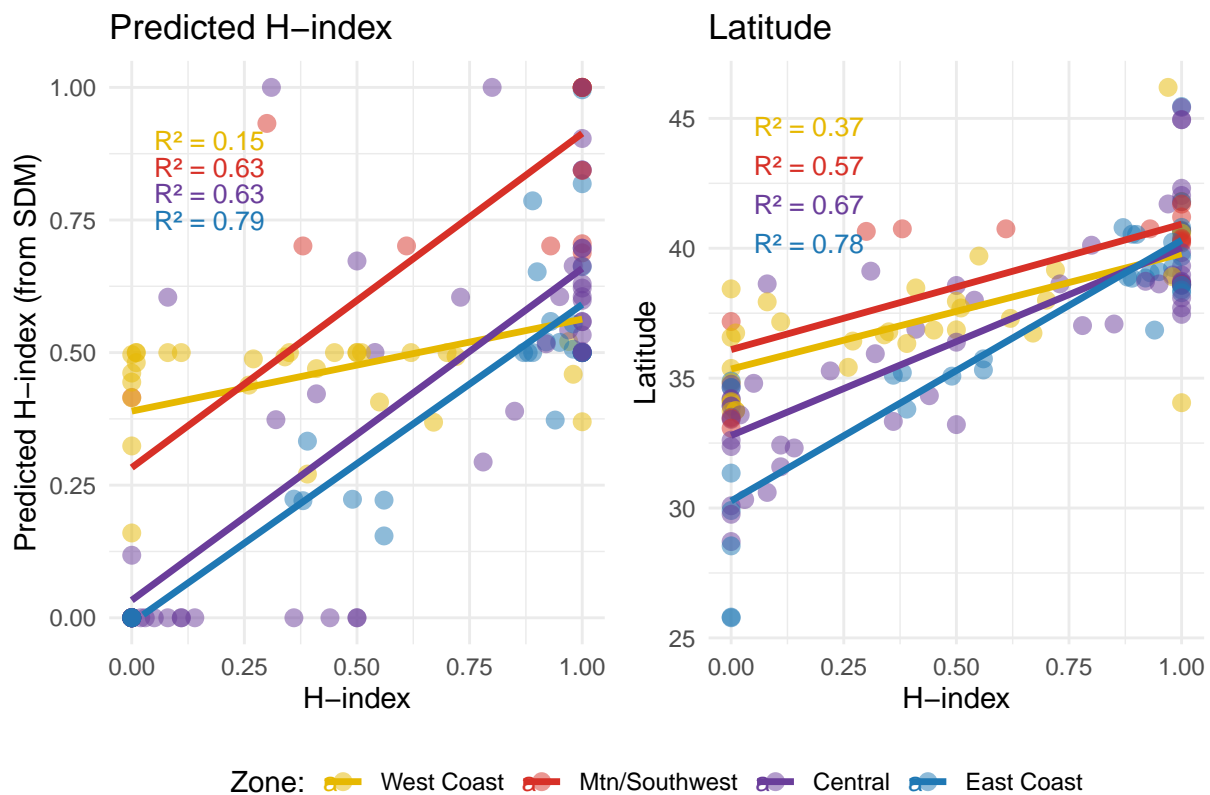
```
## colour : <GuideLegend>
```

```
# -----
# Combine plots with unified legend at bottom
# -----
```

```
# Define the plot object
```

```
final_plot <- (p2 | p1) + plot_layout(guides = 'collect') & theme(legend.position = "bottom")
```

```
final_plot
```



```
# Save as PDF with specific width and height (in inches)
```

```
#ggsave("../figs/h_index_lm_all_by_zone.pdf", plot = final_plot, width = 7, height = 4, units = "in")
```


By Year: H-index vs Predicted H-index and Latitude

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

countsAllDf <- countsAllDf %>%
  mutate(time_frame = case_when(
    year >= 1939 & year <= 1961 ~ "1940-1960",
    year >= 1989 & year <= 2019 ~ "1990-2019",
    year >= 2020 & year <= 2024 ~ "2020-2024",
    TRUE ~ "Other"
  ))

# Optional: Make time_frame a factor to control order
countsAllDf$time_frame <- factor(countsAllDf$time_frame,
                                 levels = c("1940-1960", "1990-2019", "2020-2024"))

time_frame_colors <- c("1940-1960" = "#A7A9D1", # muted lavender/blue
                      "1990-2019" = "#78A9CF", # moderate blue
                      "2020-2024" = "#80CBC4") # aqua-green, clearly distinct

time_frame_scale <- scale_color_manual(values = time_frame_colors, name = "Time Frame:")

# Shared theme
shared_theme <- theme_minimal(base_size = 10) +
  theme(
    plot.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom",
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 8),
    plot.margin = margin(10, 10, 10, 10)
  )

# -----
# H-index vs Predicted H-index
# -----

r2_pred <- countsAllDf %>%
  group_by(time_frame) %>%
  summarise(r2 = summary(lm(predicted_h_index ~ h_index))$r.squared)

p2 <- ggplot(countsAllDf, aes(x = h_index, y = predicted_h_index, color = time_frame)) +
  geom_point(size = 3, alpha = 0.5, shape = 16) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, linewidth = 1.2) +
  geom_text(data = r2_pred, aes(x = 0.05, y = 0.95 - as.numeric(time_frame) * 0.05,
                                label = paste0("R² = ", round(r2, 2)), color = time_frame),
            hjust = 0, size = 3.5, inherit.aes = FALSE) +
  shared_theme +
  labs(x = "H-index", y = "Predicted H-index (from SDM)", title = "Predicted H-index", color = "Time Frame") +
  xlim(0, 1) +
  ylim(0, 1) +
  time_frame_scale
```

```

# -----
# H-index vs Latitude
# -----

r2_lat <- countsAllDf %>%
  group_by(time_frame) %>%
  summarise(r2 = summary(lm(latitude ~ h_index))$r.squared)

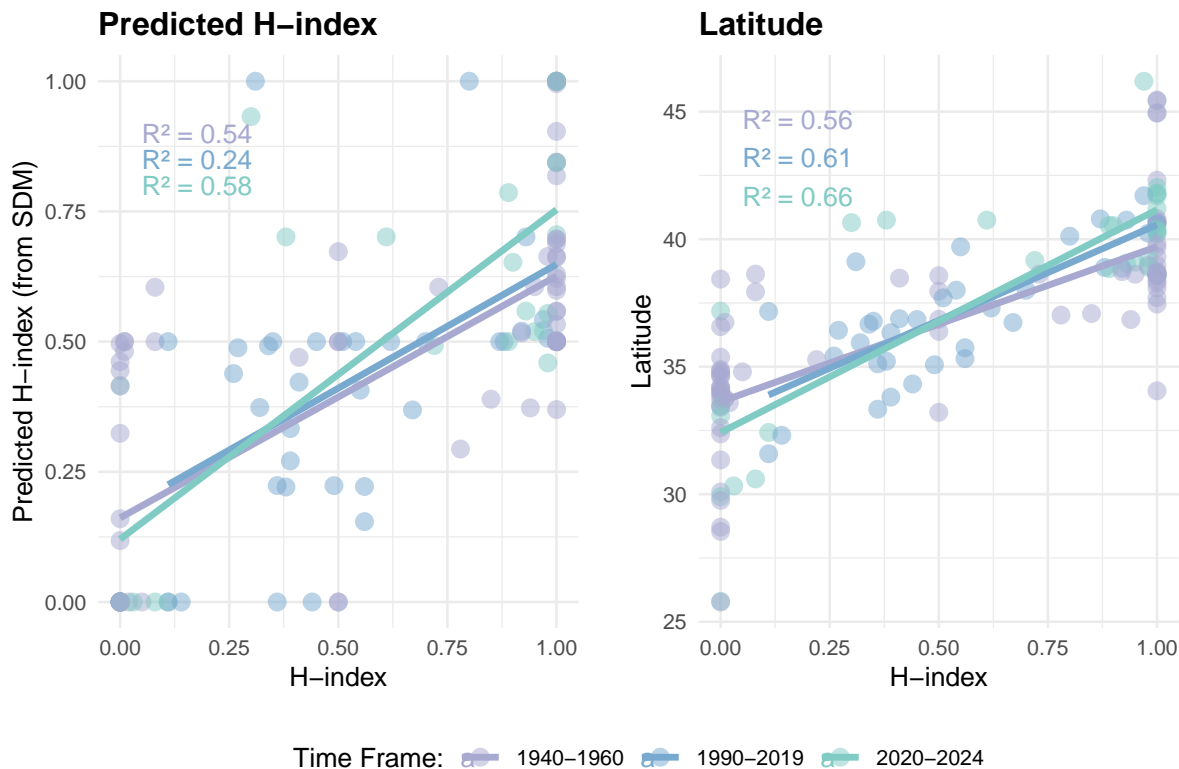
p1 <- ggplot(countsAllDf, aes(x = h_index, y = latitude, color = time_frame)) +
  geom_point(size = 3, alpha = 0.5, shape = 16) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, linewidth = 1.2) +
  geom_text(data = r2_lat, aes(x = 0.05, y = max(countsAllDf$latitude, na.rm = TRUE) - as.numeric(time_frame),
                                label = paste0("R2 = ", round(r2, 2)), color = time_frame),
            hjust = 0, size = 3.5, inherit.aes = FALSE) +
  shared_theme +
  labs(x = "H-index", y = "Latitude", title = "Latitude", color = "Time Frame:") +
  xlim(0, 1) +
  time_frame_scale

# -----
# Combine plots with unified legend at bottom
# -----

# Define the plot object
final_plot <- (p2 | p1) + plot_layout(guides = 'collect') & theme(legend.position = "bottom")

final_plot

```



```
# Save as PDF with specific width and height (in inches)
#ggsave("../figs/h_index_lm_all_by_time.pdf", plot = final_plot, width = 7, height = 4, units = "in")
```

ANCOVA (Analysis of Covariance)

This allows you to test:

Whether slopes differ significantly by zone vs. time.

Whether overall fit is better when grouping by zone vs. by time.

You can compare models using AIC or adjusted R^2 .

```
# Model 1: Zone as factor
mod_zone <- lm(latitude ~ h_index * zone, data = countsAllDf)

# Model 2: Time Frame as factor
mod_time <- lm(latitude ~ h_index * time_frame, data = countsAllDf)

# Compare model fits
AIC(mod_zone, mod_time)
```

```
##           df           AIC
## mod_zone   9 606.2853
## mod_time   7 634.8123
```

```
summary(mod_zone)
```

```
##
## Call:
## lm(formula = latitude ~ h_index * zone, data = countsAllDf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.7368 -1.3649 -0.2123  1.2204  6.5356
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      35.3504     0.5674  62.299 < 2e-16 ***
## h_index           4.4403     1.1366   3.906 0.000150 ***
## zoneMtn/Southwest  0.7480     1.2113   0.618 0.537990
## zoneCentral      -2.5602     0.7207  -3.552 0.000534 ***
## zoneEast Coast   -5.0792     0.9214  -5.512 1.85e-07 ***
## h_index:zoneMtn/Southwest  0.3864     1.7441   0.222 0.825009
## h_index:zoneCentral    2.7914     1.3134   2.125 0.035466 *
## h_index:zoneEast Coast  5.5966     1.4757   3.792 0.000228 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.134 on 129 degrees of freedom
## Multiple R-squared:  0.6899, Adjusted R-squared:  0.6731
## F-statistic:    41 on 7 and 129 DF,  p-value: < 2.2e-16
```

```
summary(mod_time)
```

```
##
```

```
## Call:
## lm(formula = latitude ~ h_index * time_frame, data = countsAllDf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.8521 -1.3168  0.0515  1.0134  5.7474
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      33.6263     0.4111  81.800 <2e-16 ***
## h_index           6.0822     0.6029  10.088 <2e-16 ***
## time_frame1990-2019 -0.5602     1.0190  -0.550  0.5835
## time_frame2020-2024 -1.2066     0.9115  -1.324  0.1879
## h_index:time_frame1990-2019  1.4157     1.6986   0.833  0.4061
## h_index:time_frame2020-2024  2.6501     1.2068   2.196  0.0298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.385 on 131 degrees of freedom
## Multiple R-squared:  0.6068, Adjusted R-squared:  0.5918
## F-statistic: 40.43 on 5 and 131 DF, p-value: < 2.2e-16
```

Repeat with SDM predicted H-index

```
# Model with zone
mod_pred_zone <- lm(predicted_h_index ~ h_index * zone, data = countsAllDf)
summary(mod_pred_zone)
```

```
##
## Call:
## lm(formula = predicted_h_index ~ h_index * zone, data = countsAllDf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38964 -0.09067 -0.02205  0.07831  0.77307
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.38964     0.04690   8.308 1.50e-13 ***
## h_index           0.17351     0.09394   1.847  0.06715 .
## zoneMtn/Southwest -0.10653     0.10017  -1.064  0.28963
## zoneCentral       -0.35659     0.06098  -5.848 4.20e-08 ***
## zoneEast Coast    -0.39905     0.07919  -5.039 1.63e-06 ***
## h_index:zoneMtn/Southwest  0.45668     0.14571   3.134  0.00216 **
## h_index:zoneCentral   0.45191     0.10979   4.116 7.00e-05 ***
## h_index:zoneEast Coast  0.42657     0.12434   3.431  0.00082 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1764 on 123 degrees of freedom
## (6 observations deleted due to missingness)
## Multiple R-squared:  0.6763, Adjusted R-squared:  0.6578
## F-statistic: 36.71 on 7 and 123 DF, p-value: < 2.2e-16
```

```
# Model with time frame
mod_pred_time <- lm(predicted_h_index ~ h_index * time_frame, data = countsAllDf)
summary(mod_pred_time)
```

```
##
## Call:
## lm(formula = predicted_h_index ~ h_index * time_frame, data = countsAllDf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39367 -0.16157 -0.06731  0.14973  0.67969
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.16157    0.03932   4.109 7.14e-05 ***
## h_index          0.46421    0.05654   8.210 2.33e-13 ***
## time_frame1990-2019  0.01136    0.09302   0.122  0.903
## time_frame2020-2024 -0.04102    0.08343  -0.492  0.624
## h_index:time_frame1990-2019 0.01119    0.15431   0.073  0.942
## h_index:time_frame2020-2024 0.16857    0.11110   1.517  0.132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2156 on 125 degrees of freedom
## (6 observations deleted due to missingness)
## Multiple R-squared:  0.5084, Adjusted R-squared:  0.4887
## F-statistic: 25.85 on 5 and 125 DF,  p-value: < 2.2e-16
```

```
# Compare model fits
AIC(mod_pred_zone, mod_pred_time)
```

```
##              df          AIC
## mod_pred_zone  9 -73.05279
## mod_pred_time  7 -22.33187
```

```
summary(mod_pred_zone)
```

```
##
## Call:
## lm(formula = predicted_h_index ~ h_index * zone, data = countsAllDf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38964 -0.09067 -0.02205  0.07831  0.77307
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.38964    0.04690   8.308 1.50e-13 ***
## h_index          0.17351    0.09394   1.847  0.06715 .
## zoneMtn/Southwest -0.10653    0.10017  -1.064  0.28963
## zoneCentral       -0.35659    0.06098  -5.848 4.20e-08 ***
## zoneEast Coast    -0.39905    0.07919  -5.039 1.63e-06 ***
## h_index:zoneMtn/Southwest 0.45668    0.14571   3.134  0.00216 **
## h_index:zoneCentral  0.45191    0.10979   4.116 7.00e-05 ***
## h_index:zoneEast Coast  0.42657    0.12434   3.431  0.00082 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1764 on 123 degrees of freedom
## (6 observations deleted due to missingness)
## Multiple R-squared:  0.6763, Adjusted R-squared:  0.6578
## F-statistic: 36.71 on 7 and 123 DF,  p-value: < 2.2e-16
```

```
summary(mod_pred_time)
```

```
##
## Call:
## lm(formula = predicted_h_index ~ h_index * time_frame, data = countsAllDf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39367 -0.16157 -0.06731  0.14973  0.67969
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.16157    0.03932   4.109 7.14e-05 ***
## h_index           0.46421    0.05654   8.210 2.33e-13 ***
## time_frame1990-2019  0.01136    0.09302   0.122  0.903
## time_frame2020-2024 -0.04102    0.08343  -0.492  0.624
## h_index:time_frame1990-2019 0.01119    0.15431   0.073  0.942
## h_index:time_frame2020-2024 0.16857    0.11110   1.517  0.132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2156 on 125 degrees of freedom
## (6 observations deleted due to missingness)
## Multiple R-squared:  0.5084, Adjusted R-squared:  0.4887
## F-statistic: 25.85 on 5 and 125 DF,  p-value: < 2.2e-16
```