

Chapter2Code

Reilly Brennan

2025-07-15

```
##Start: Load Packages
##Part 1. Parasite Species Richness###
Load Data: - Read in csv with records of parasites from literature review - Create coordinate object with each record and then buffer with 0.5 degrees around each point - Sort by unique records to take a look at the list
par<- read.csv("ColiPar.csv", header=TRUE) #has coordinates lat/long for olivacea sequences (all, cytb
wgscoor<- par
coordinates(wgscoor) <- ~Longitude + Latitude
proj4string(wgscoor)<- CRS("+proj=longlat +datum=WGS84")
raster:::shapefile(wgscoor, "ColiPar.shp", overwrite=T)
#->buffered in qGIS and imported below (0.5 degrees)

sort(unique(par$Parasite))

## [1] "ANDV"
## [2] "Barreropsylla excelsa"
## [3] "Bartonella"
## [4] "Borrelia"
## [5] "Capillaria"
## [6] "Chiliopsylla allophyla allophyla"
## [7] "Cryptosporidium"
## [8] "Ctenoparia inopinata"
## [9] "Ctenoparia topali"
## [10] "Cuterebra apicalis"
## [11] "Giardia"
## [12] "Gigantolaelaps wolffsohni"
## [13] "Haemogamasus alongipilis"
## [14] "Hectopsylla cypha"
## [15] "Hemoplasma"
## [16] "Hepatozoon"
## [17] "Hoplopleura travassosi"
## [18] "Hymenolepis"
## [19] "Ixodes sigelos "
## [20] "Ixodes stilesi"
## [21] "Litomosoides pardinasi"
## [22] "Lukoschus maresi"
## [23] "Mysolaelaps microspinosis "
## [24] "Neotyphloceras chilensis"
## [25] "Neotyphloceras crassispina crassispina"
## [26] "Physaloptera"
## [27] "Physaloptera calnuensis"
```

```

## [28] "Pterygodermatites (Paucipectines)"
## [29] "Quadraseta chiloensis"
## [30] "Sphinctopsylla ares"
## [31] "Strongylidia"
## [32] "Syphacia"
## [33] "Tetrapsyllus rhombus"
## [34] "Tetrapsyllus tantillus"
## [35] "Trypanosoma cruzi"

```

Import buffered file of parasite points and use phyloraster package to consider species richness

```

coli_par <- terra::vect("./ColiParBuff/ColiParBuff.shp")

r2_colipar <- shp2rast(coli_par, sps.col = "Parasite", ymask = FALSE, background = 0,
                         resolution = 0.5) # convert to raster with phyloraster
layer_colipar<- as.data.frame(r2_colipar, xy= T) #create df with raster data
srcoli<- rast.sr(x=r2_colipar) #calculate species richness using phyloraster function

```

Create S. America for map overlay

```

Chile<- read_sf("./CHL_adm/CHL_adm0.shp")
Argentina <-read_sf("./ARG_adm/ARG_adm0.shp")
SA <- st_union(Chile, Argentina)

```

```

## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries

```

Match the parasite species richness raster to the extent and resolution of South America

```

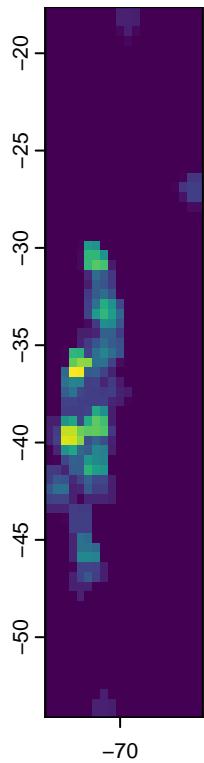
test2_par <- terra::crop(srcoli, SA)

ext <- floor(ext(SA))
nr <- rast(ext, res = res(srcoli))
values(nr) <- 0
testnew <- terra::mask(nr,SA)

tn_ras<-rast(testnew)
t2_ras<-rast(test2_par)

res(tn_ras) <- 0.5
res(t2_ras)<-0.5
crs(t2_ras)<-crs(tn_ras)
testnew<- resample(testnew, tn_ras)
test2_par<-resample(test2_par, t2_ras)
plot(test2_par)

```



```
plot_par <- mosaic(test2_par, testnew, fun="max")
srSA <- terra::mask(plot_par, SA)
```

ggplot map for parasite richness

```
par_plot_df<- as.data.frame(srSA, xy=TRUE)
par_plot<- subset(par_plot_df, SR!=0)
par_zero<- subset(par_plot_df, SR==0)

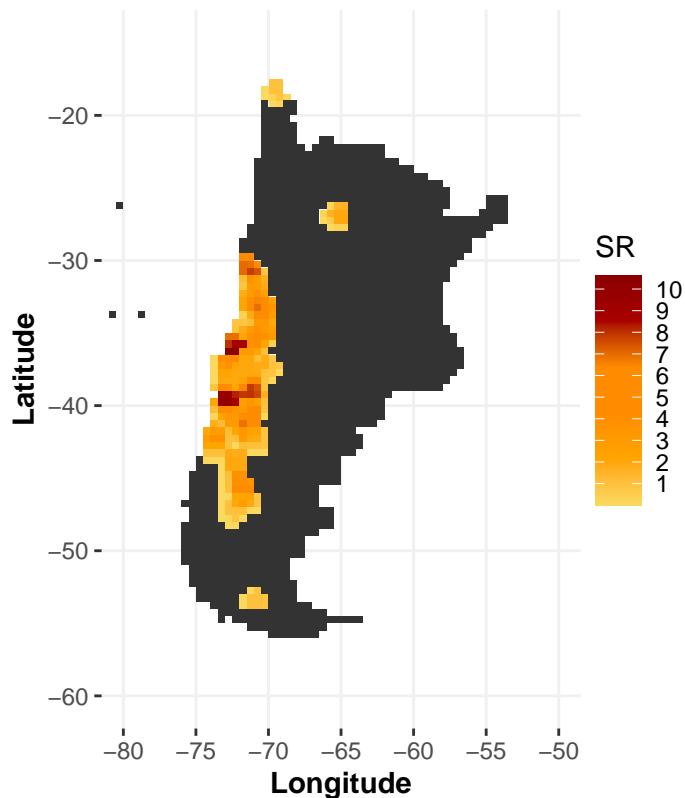
xlim<-c(-80, -50)
ylim<-c(-60, -15)

parmap <- ggplot(data = par_plot) +
  geom_raster(aes(x = x, y = y, fill = `SR`)) +
  scale_fill_gradientn(colours = c("#fada5fff", "orange", "darkorange", "darkorange", "#aa0000", "red4"),
                        breaks = c(1,2,3,4,5,6,7,8,9,10),
                        labels= c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10"))+
  geom_raster(data=par_zero, aes(x = x, y = y), show.legend = F)+
  ggtitle(expression(paste(bold("Parasite Species Richness of"), bolditalic (" O. longicaudatus")))) +
  theme(legend.position = "right", legend.box.spacing = unit(0, "pt"), panel.background = element_rect(),
        panel.grid = element_line(color = "gray94"), plot.title = element_text(hjust = 0.5, size=10),
        axis.title = element_text(face="bold"))
  ) +
  coord_sf(xlim=xlim, ylim=ylim) +
  labs(x="Longitude", y= "Latitude")
parmap

## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.
## Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.
```

```
## Raster pixels are placed at uneven horizontal intervals and will be shifted  
## i Consider using `geom_tile()` instead.
```

Parasite Species Richness of *O. longicaudatus*



Make data frame of parasite species richness for

data analysis

```
dfa<- as.data.frame(srcoli, xy=T)
```

##Part 2. ANDV Phylogeny

###Phylogenetic Tree Building- This can be skipped as the .tree file is provided. However, a new tree can be built. The ML algorithm may find a tree slightly different than the one provided in the .tree file, but it won't significantly change the results.

Load the sequence file ANDV.fasta and explore the sequences:

```
fa_andv <- readDNAStringSet("ANDV.fasta")  
len <- seqlengths(fa_andv)  
len<-as.data.frame(len)  
min(len$len)
```

```
## [1] 371  
max(len$len)
```

```
## [1] 943  
##function we will use throughout ##  
alignment2Fasta <- function(alignment, filename) {  
  sink(filename)  
  
  n <- length(rownames(alignment))  
  for(i in seq(1, n)) {
```

```

    cat(paste0('>', rownames(alignment)[i]))
    cat('\n')
    the.sequence <- toString(unmasked(alignment)[[i]])
    cat(the.sequence)
    cat('\n')
}

sink(NULL)
}

```

Align the sequences. The msa package works with ClustalW, ClustalOmega, and MUSCLE alignment algorithms. Save the alignment Trim the alignment and convert it to DNAbin object

```

seqs_andv<- readDNAStringSet("ANDV.fasta")#fasta file classsic format with about 180 cytb seqs

align_andv<- msaMuscle(seqs_andv) #align sequences, there are different parameters here that can be ch

alignment2Fasta(align_andv, 'alignmentANDV_may7.fasta') #my function

aligned_andv<- readFasta('alignmentANDV_may7.fasta')
trimANDV_align<- msaTrim(aligned_andv, gap.end = 0.1, gap.mid = 0.9)
trimANDV_align <- msa2mat(trimANDV_align) # for use with ape::as.DNAbin(msa.mat)
trimANDV_align<- ape::as.DNAbin(trimANDV_align)

```

Create phydat, and test models of substitution. Build Phylogenetic Tree. Save tree.

```

andv_dat<-as.phyDat(trimANDV_align)

andvmodels<- modelTest(andv_dat)

fit_andv<- pml_bb(andvmodels, control = pml.control(trace = 0), rearrangements="stochastic")

bs_andv <- bootstrap.pml(fit_andv, bs=1000, optNni=TRUE,trees=TRUE,
                           control = pml.control(trace = 0))

plotBS(midpoint(fit_andv$tree), p = .5, type="p", digits=2, main="Ultrafast bootstrap")

plotBS(midpoint(fit_andv$tree), bs, p = 50, type="p", main="Standard bootstrap")

plotBS(midpoint(fit_andv$tree), bs, p = 50, type="p", digits=0, method = "TBE", main="Transfer bootstrap"
treeandv<- fit_andv$tree
plot(fit_andv$tree)
plot(treeandv)

treebs_andv<-plotBS(fit_andv$tree, bs_andv, type= "n")
plot(treebs_andv)
write.tree(treebs_andv, "ANDVTree_BS.tree")

```

Calculating Phylogenetic Richness in Geographic Space Get shapefile with points for each data from occurrence matrix

```

andv_points<- read.csv("ColilargoANDVdata.csv", header=TRUE) #has coordinates lat/long for cytb coli
andv_points[is.na(andv_points)] <- 0
wgscoor<-andv_points
coordinates(wgscoor) <- ~ Longitude+ Latitude
proj4string(wgscoor)<- CRS("+proj=longlat +datum=WGS84")

```

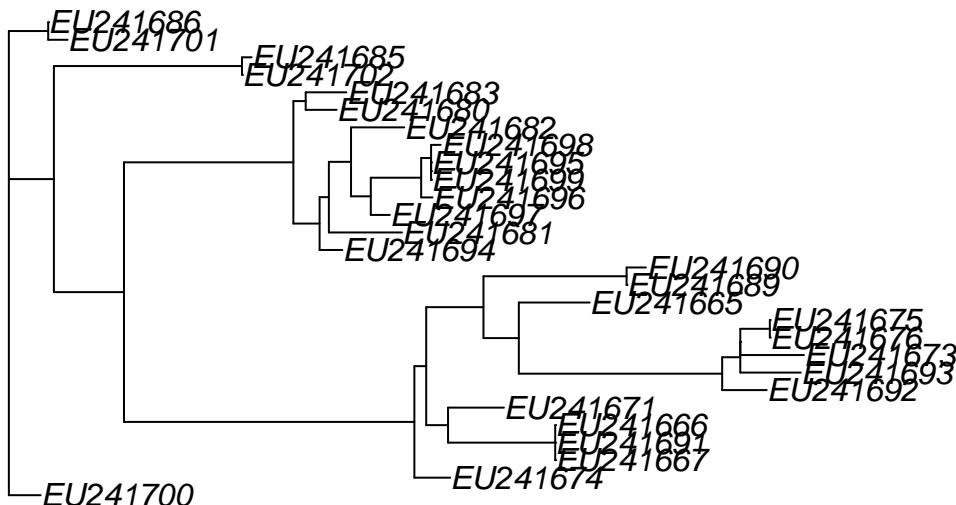
```
raster::shapefile(wgscoor, "ANDV.shp", overwrite=T)
#buffered in qGIS and imported below (0.5 degrees)
```

Phylo geo map just to take a fun look at where the ANDV seqs are coming from

```
treeandv<-read.tree("ANDVTree_BS.tree")
tipsandv<- read.csv("andvtips.csv", header=TRUE) # a table to change the tip labels so they will match
pos_id <-match(treeandv$tip.label,tipsandv$old ) #element position

treeandv$tip.label <- tipsandv$new[pos_id]

plot(treeandv)
```



```
andv_points<- read.csv("ColilargoANDVdata.csv", header=TRUE)
```

```
andv_coords<-data.frame(tip= c(andv_points$Accession),
                          lat= c(andv_points$Latitude),
                          lon= c(andv_points$Longitude))
row.names(andv_coords)<- andv_coords$tip
andv_coords$tip<-NULL
```

```
map_andv<- phylo.to.map(treeandv, andv_coords, rotate= T, plot=F,
                           direction="rightwards", regions="Chile")
```

```
## objective: 370
```

```
## objective: 318
```

```
## objective: 318
```

```
## objective: 142
```

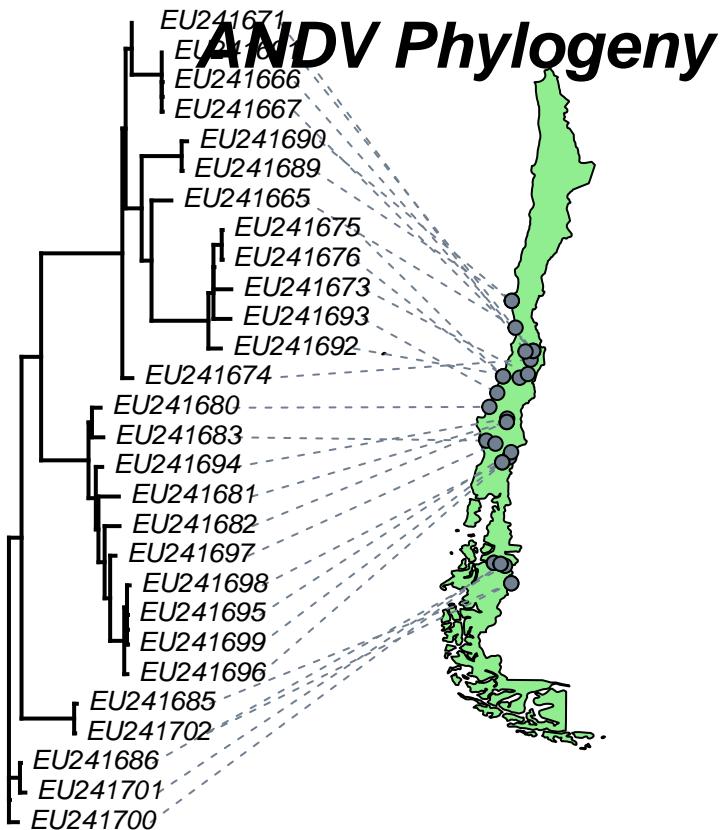
```
## objective: 142
```

```
## objective: 138
```

```
## objective: 134
```

```
## objective: 134
```

```
## objective: 132
```



Import the buffered points, calculate phylogenetic diversity of ANDV

```
shpandv <- terra::vect("ANDVbuff/ANDVbuff.shp") # collection points with buffers
r2ANDV <- shp2rast(shpandv,sps.col = "Accession", ymask = FALSE, background = 0, resolution =0.5) # turn into raster

treeandv<- read.tree("ANDVTree_BS.tree") #load phylogenetic tree
```

```

tipsandv<- read.csv("andvtips.csv", header=TRUE) # a table to change the tip labels so they will match

pos_id_andv <-match(treeandv$tip.label,tipsandv$old ) #element position
treeandv$tip.label <- tipsandv$new[pos_id_andv]

dataprep_andv<- phylo.pres(x=r2ANDV, tree=treeandv, pruning= "tree") # take the data points for which t
pdr_andv <- rast.pd(x = dataprep_andv$x, edge.path = dataprep_andv$edge.path,
branch.length = dataprep_andv$branch.length) # get phlyo diversity with function from ph

```

Normalize pd values by number of individuals

```

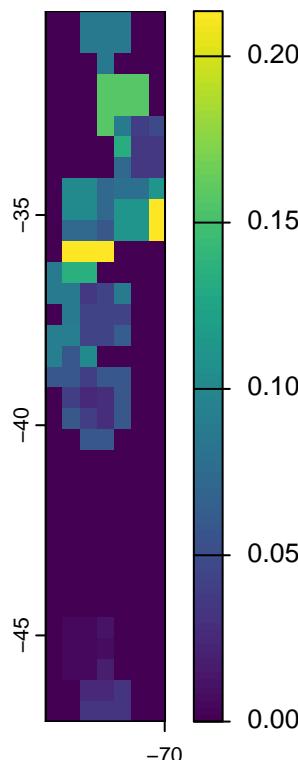
#create df from raster layer
layer_andv<- as.data.frame(r2ANDV, xy=T)

#create df from PD values
dfandv_coli<- as.data.frame(pdr_andv$PD, xy=T)
colnames(dfandv_coli)[3]<-"PDandv"

#merge and divide PD by # of individuals
seqANDV<- merge(dfandv_coli, layer_andv, by=c("x", "y"))
seqandv_sum<-seqANDV
seqandv_sum$SeqSum <- rowSums(seqandv_sum[,4:33] )
seqANDV$PDandv<- seqandv_sum$PDandv/seqandv_sum$SeqSum

seqANDV[is.na(seqANDV)]<-0
PD_ANDV<- data.frame(seqANDV[1:3])
r_PDandv<- rast(PD_ANDV, type="xyz")
plot(r_PDandv)

```

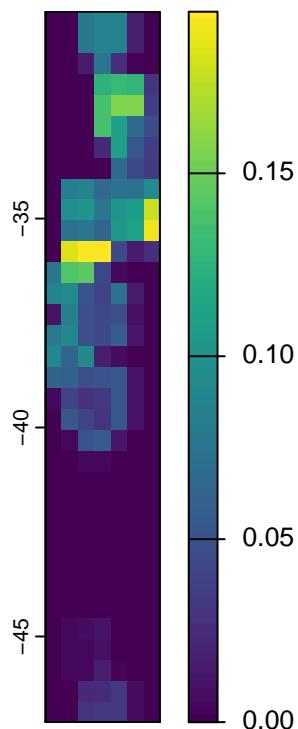


Phylogenetic diversity of ANDV Map

```
test2 <- terra::crop(r_PDandv$PDandv, SA)
ext <- floor(ext(SA))
nr <- rast(ext, res = res(r_PDandv$PDandv))
values(nr) <- 0
testnew <- terra::mask(nr,SA)

tn_ras<-rast(testnew)
t2_ras<-rast(test2)

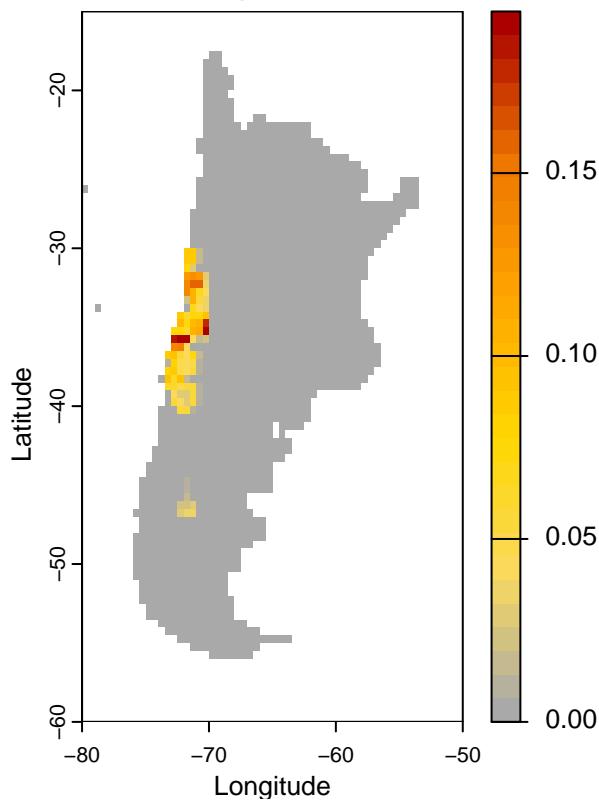
res(tn_ras) <- 0.5
res(t2_ras)<-0.5
crs(t2_ras)<-crs(tn_ras)
testnew<- resample(testnew, tn_ras)
test2<-resample(test2, t2_ras)
plot(test2)
```



```
plot<- terra::mosaic(test2,testnew, fun="max")

PDplot_andv <- terra::mask(plot,SA)
color<-colorRampPalette(c("darkgray", "#fada5fff", "gold", "orange","darkorange2" ,"#aa0000"))
plot(PDplot_andv,
     main= "ANDV Phylogenetic Diversity", ext=c(-80, -50, -60, -15), xlab="Longitude", ylab="Latitude",
     col=color(30))
```

ANDV Phylogenetic Diversity



Phylogenetic diversity of ANDV Map GGPLOT

```
andv_plot_df<- as.data.frame(PDplot_andv, xy=TRUE)
andv_plot<- subset(andv_plot_df, PDandv!=0)
andv_zero<- subset(andv_plot_df, PDandv==0)

xlim<-c(-80, -50)
ylim<-c(-60, -15)

andvmap<-ggplot(data = andv_plot) +
  geom_raster(aes(x = x, y = y, fill = `PDandv`)) +
  scale_fill_gradientn(colours = c("#fada5fff", "orange", "darkorange2", "#aa0000", "red4"),
                       breaks = c(0.05, 0.1, 0.15, 0.2, max(andv_plot$PDandv)),
                       labels= c("0.05", "0.1", "0.15", "0.2", "0.22"))+
  geom_raster(data=andv_zero, aes(x = x, y = y), show.legend = F)+
  theme(legend.position = "right", legend.box.spacing = unit(0, "pt"), panel.background = element_rect(
    panel.grid = element_line(color = "gray94"),
    axis.title = element_text(face="bold"),
    plot.title = element_text(hjust = 0.5, size=10)) +
  coord_sf(xlim=xlim, ylim=ylim) +
  labs(x="Longitude", y= "Latitude")+
  ggtitle(expression(paste(bold("Phylogenetic Diversity of ANDV")))))
```

##Part 3. Colilargo Phylo

###Phylogenetic Tree Building- This can be skipped as the .tree file is provided. However, a new tree can be built. The ML algorithm may find a tree slightly different than the one provided in the .tree file, but it won't significantly change the results. Align the sequences. The msa package works with ClustalW,

ClustalOmega, and MUSCLE alignment algorithms. Here I am using MUSCLE.

```
seqs_coli<- readDNAStringSet("CytBColi_4-22.fasta")
align_coli<- msaMuscle(seqs_coli) #align sequences, there are different parameters here that can be changed
alignment2Fasta(align_coli, 'alignment_july.fasta') #my function
aligned<- readFasta('alignment_july.fasta')
trim_align<- msaTrim(aligned, gap.end = 0.1, gap.mid = 0.9)
```

Build Phylogenetic Tree

```
#convert data type
trim_align <- msa2mat(trim_align) # for use with ape::as.DNAbin(msa.mat)
trim_align<- ape::as.DNAbin(trim_align)
colidat<-as.phyDat(trim_align)

#model testing
colimodels<- modelTest(colidat)
#model fit
fit_coli <- pml_bb(colimodels, control = pml.control(trace = 0))
#bootstrap
bs_coli <- bootstrap.pml(fit_coli, bs=1000, optNni=TRUE,trees=TRUE,
                           control = pml.control(trace = 0))

treebs<-plotBS(fit_coli$tree, bs_coli, type= "n",p=0.5)
plot(treebs)
plotBS(midpoint(fit_coli$tree), p = .5, type="p", digits=2, main="Ultrafast bootstrap")

plotBS(midpoint(fit_coli$tree), bs, p = 50, type="p", main="Standard bootstrap")

plotBS(midpoint(fit_coli$tree), bs, p = 50, type="p", digits=0, method = "TBE", main="Transfer bootstrap")

plotBS(midpoint(fit_coli$tree), bs_coli, p = 50, type="p", main="Standard bootstrap")

write.tree(treebs, "ColiTreeJuly_bs.tree")
write.tree(fit_coli$tree,"ColiTreeJuly.tree" )

###Calculating Phylogenetic Richness in Geographic Space
```

Get shapefile with points for each data from occurrence matrix

```
cyt<- read.csv("CytBData.csv", header=TRUE) #has coordinates lat/long for cytb coli
cyt[is.na(cyt)] <- 0
wgscoor<- cyt
coordinates(wgscoor) <- ~Longitude+Latitude
proj4string(wgscoor)<- CRS("+proj=longlat +datum=WGS84")
raster::shapefile(wgscoor, "CytBColi.shp", overwrite=T)
#buffered in qGIS and imported below (0.5 degrees)
```

Colilargo Phyloraster

```
shpcoli <- terra::vect("CytColiBuff/CytColiBuff.shp") # collection points with buffers
r2 <- shp2rast(shpcoli,sps.col = "Accession", ymask = FALSE, background = 0, resolution =0.5) # turn shapefile into raster
tree<- read.tree("ColiTreeJuly.tree") #load phylogenetic tree

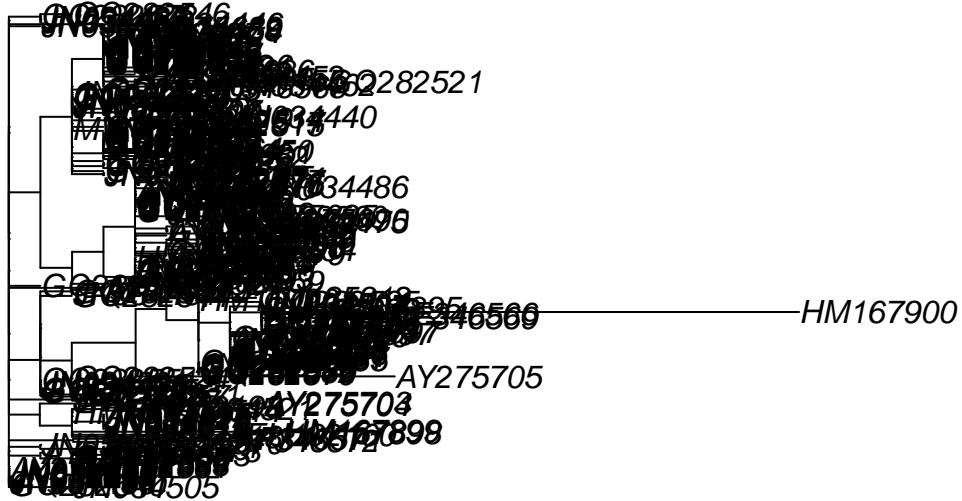
tips<- read.csv("tips_april23.csv", header=T)
# make tips match
pos_id <-match(tree$tip.label, tips$old)#element position
```

```

colitree<-tree
colitree$tip.label <- tips$new[pos_id] #here sorting by pos_id

plot(colitree)

```

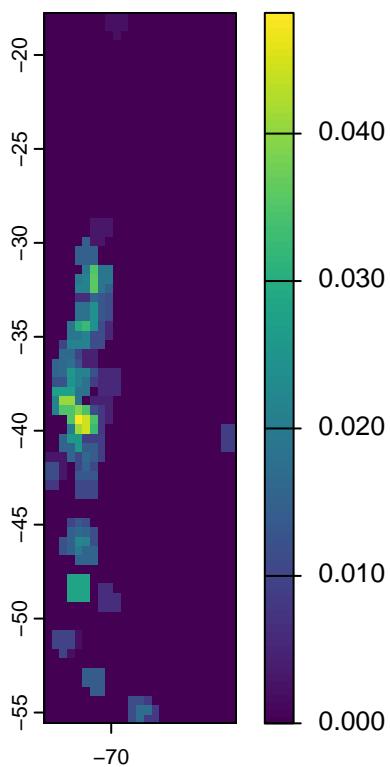


```

##prid for phyloraster##
dataprep<- phylo.pres(x=r2, tree=colitree, pruning= "tree") # take the data points for which there is p

ed <- rast.ed(dataprep$x, colitree)
pdr <- rast.pd(x = dataprep$x, edge.path = dataprep$edge.path,
                branch.length = dataprep$branch.length) # get phlyo diversity with function from phylora
plot(pdr$PD)

```



Normalize pd values by number of individuals

```

#create df from raster layer of colilargo occ.
layer_coli<- as.data.frame(r2, xy= T)

#create dfs from PD values
dfpd_coli<-as.data.frame(pdr$PD, xy=T)

mean(dfpd_coli$PD[dfpd_coli$PD != 0])

## [1] 0.01469698

#merge and divide PD by # of individuals
seqPD<- merge(dfpd_coli, layer_coli, by=c("x", "y"))
seqPDsum<-seqPD
seqPDsum$SeqSum <- rowSums(seqPDsum[,4:241] )
seqPD$PD<- seqPDsum$PD/seqPDsum$SeqSum

PD_colilargo<-data.frame(seqPD[1:3])
max(PD_colilargo$PD)

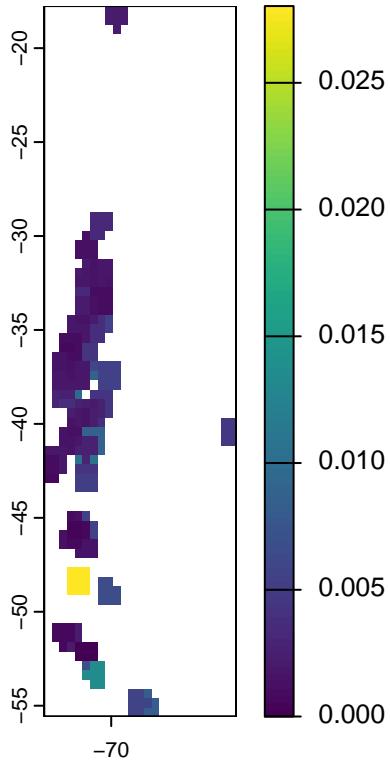
## [1] NaN

min(PD_colilargo$PD[PD_colilargo$PD != 0])

## [1] NA

r_PDcolilargo<- rast(PD_colilargo, type="xyz")
plot(r_PDcolilargo)

```



Make map pretty use shp files of chile and arg

```

Chile<- read_sf("./CHL_adm/CHL_adm0.shp")
Argentina <-read_sf("./ARG_adm/ARG_adm0.shp")

```

```

SA <- st_union(Chile, Argentina)

## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries

Phylogenetic diversity Colilargo Map

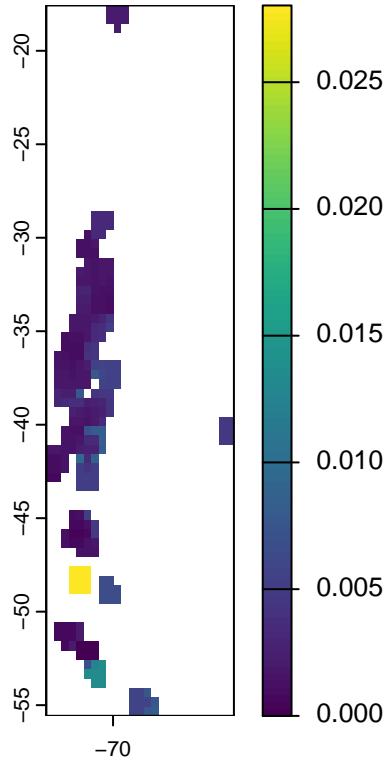
test2 <- terra::crop(r_PDcolilargo$PD, SA)

ext <- floor(ext(SA))
nr <- rast(ext, res = res(test2))
raster::values(nr) <- 0
testnew <- terra::mask(nr, SA)

tn_ras<-rast(testnew)
t2_ras<-rast(test2)

res(tn_ras) <- 0.5
res(t2_ras)<-0.5
crs(t2_ras)<-crs(tn_ras)
testnew <- resample(testnew, tn_ras)
test2<-resample(test2, t2_ras)
plot(test2)

```



```

plot<- mosaic(test2,testnew, fun="max")

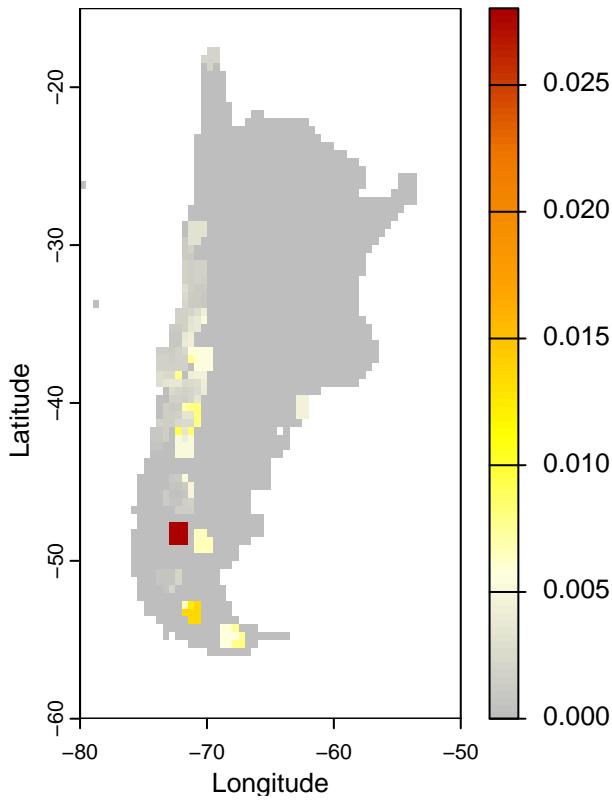
PDplot <- terra::mask(plot,SA)

color<-colorRampPalette(c("gray","lightyellow","yellow", "orange", "darkorange2","#aa0000"))
plot(PDplot,
     main= "Colilargo Phylogenetic Diversity", ext=c(-80, -50, -60, -15), xlab="Longitude", ylab="Latitude")

```

```
col=color(100))
```

Colilargo Phylogenetic Diversity



Plotting Phylo Div Colilargo GGPLOT

```
pd_plot_df<- as.data.frame(PDplot, xy=TRUE)
pd_plot<- subset(pd_plot_df, PD!=0)
max<-subset(pd_plot_df, PD=0.1147386)
```

```
## Warning: In subset.data.frame(pd_plot_df, PD = 0.1147386) :
##   extra argument 'PD' will be disregarded
```

```
pdzero<- subset(pd_plot_df, PD==0)
xlim<-c(-80, -50)
ylim<-c(-60, -15)
```

```
pdmap<-ggplot(data = pd_plot) +
  geom_raster(aes(x = x, y = y, fill = `PD`)) +
  scale_fill_gradientn(colours = c("#fada5fff", "orange", "darkorange2", "#aa0000", "red4" ),
                      breaks = c( 0.001, 0.01, 0.02, 0.03, 0.04),
                      labels= c( "0.001", "0.01", "0.02", "0.03", "0.04"))+
  geom_raster(data=pdzero, aes(x = x, y = y), show.legend = F)+

  theme(legend.position = "right", legend.box.spacing = unit(0, "pt"), panel.background = element_rect(
    fill="white", color="black"),
    panel.grid = element_line(color = "gray94"),
    axis.title = element_text(face="bold"),
    plot.title = element_text(hjust = 0.5, size=10)) +
  coord_sf(xlim=xlim, ylim=ylim) +
```

```

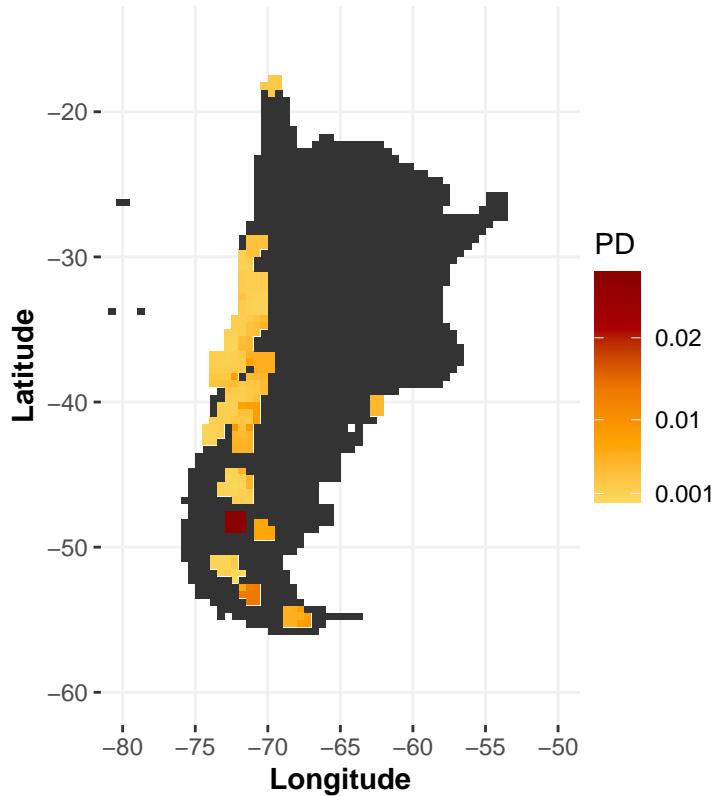
  labs(x="Longitude", y= "Latitude")+
  ggtitle(expression(paste(bold("Phylogenetic Diversity of "), bolditalic("O. longicaudatus")))))
pdmap

## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.

## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.
## Raster pixels are placed at uneven horizontal intervals and will be shifted
## i Consider using `geom_tile()` instead.

Phylogenetic Diversity of O. longicaudatus

```



```
##Part 4. Combine ANDV, Colilargo, and parasite species richness
```

```
Combine PD, SR, latitude
```

```

layer_coli<- as.data.frame(r2, xy= T)

##with unique coords
dfpd_coli<- as.data.frame(r_PDcolilargo$PD, xy=T)
dfpar_coli<- as.data.frame(srcoli, xy=T)
dfandv_coli<- as.data.frame(r_PDandv$PDandv, xy=T)
colnames(dfandv_coli)[3]<- "PDandv"

dfpar_coli$SR[dfpar_coli$SR==0]<-NA
dfpd_coli$PD[dfpd_coli$PD==0]<-NA
dfandv_coli$PDandv[dfandv_coli$PDandv==0]<-NA

```

```

#with matched standard coordinates
pdcoli<-as.data.frame(PDplot, xy=T)
srcoli<-as.data.frame(srSA, xy=T)
andvcoli<-as.data.frame(PDplot_andv, xy=T)
colnames(andvcoli)[3]<-"PDandv"
srcoli$SR[srcoli$SR==0]<-NA
pdcoli$PD[pdcoli$PD==0]<-NA
andvcoli$PDandv[andvcoli$PDandv==0]<-NA

colilargo<- merge(pd_plot, andv_plot, by=c("x", "y"))

dfnew<- merge(pdcoli, srcoli, by= c("x", "y"))
dfnew<- merge(dfnew, andvcoli, by=c("x", "y"))
dfssub<- subset(dfnew, ! (is.na(PD) & (is.na(SR)) & is.na(PDandv))) ### this is for climate

write.csv(dfssub, "climate_df.csv")

subdf_coli<- subset(dfnew, PD!=0 & PDandv!=0) ##NO NA FOR ANDV AND COLI

write.csv(dfnew, "dfnew.csv")

```

##Part 5. Nucleotide and Haplotype Diversity of Colilargo

Load Sequence data and cluster all sequences

```

cyt<- read.csv("CytBData.csv", header=TRUE) #info for the sequences
coor<- as.data.frame(cbind(cyt$Longitude, cyt$Latitude))
clusters<- balanced_clustering(coor, 17, method="centroid") #237/17=
cluster<- as.data.frame(clusters)
cyt$cluster<- cluster

```

Align and trim sequences

```

fa <- readDNAStringSet("CytBColi.fasta")
len <- seqLengths(fa)
len<-as.data.frame(len)
write.csv(len, "lengths.csv")
len<- read.csv("lengths.csv")
sublen<- subset(len, len>800)

subfa<- fa[c(which(names(fa) %in% sublen$X))]

align<-msaMuscle(subfa)
alignment2Fasta(align, 'alignment.fasta') #my function

aligncon<- readFasta('alignment.fasta')

trimmed_align<- msaTrim(alignment, gap.end = 0.1, gap.mid = 0.9)

msa.mat <-as.data.frame(msa2mat(trimmed_align) )
write.csv(msa.mat, "msa.mat.csv")
msa.mat<-read.csv("msa.mat.csv", header=TRUE)
tips<- read.csv("msalabels.csv", header=TRUE) # a table to change the tip labels
msa.mat$X[msa.mat$X %in% tips$old] <- tips$new

## Warning in msa.mat$X[msa.mat$X %in% tips$old] <- tips$new: number of items to

```

```

## replace is not a multiple of replacement length
msa.mat$X[msa.mat$X %in% cyt$Accession] <- cyt$cluster$clusters

## Warning in msa.mat$X[msa.mat$X %in% cyt$Accession] <- cyt$cluster$clusters:
## number of items to replace is not a multiple of replacement length
msa.mat$X<-as.character(msa.mat$X)

Caluclate haplotype and nucleotide diversity by cluster
##### HAPLOTYPE DIVERSITY BY CLUSTER #####
hap<-list()
for(i in 1:length(unique(msa.mat$X))){
  name = unique(msa.mat$X)[i]
  df= filter(msa.mat, X==name)
  mat= as.matrix(df)
  bin = ape::as.DNAbin(mat)
  div= hap.div(bin)
  hap[[i]]<-div
}

## Warning in haplotype.DNAbin(x): some sequences of different lengths were
## assigned to the same haplotype
hapdf<-do.call(rbind.data.frame, hap)
hapdf$cluster<- row.names(hapdf)

##### NUCLEOTIDE DIVERSITY BY CLUSTER #####
nuc<-list()
for(i in 1:length(unique(msa.mat$X))){
  name = unique(msa.mat$X)[i]
  df= filter(msa.mat, X==name)
  mat= as.matrix(df)
  bin = ape::as.DNAbin(mat)
  div= nuc.div(bin)
  nuc[[i]]<-div
}

nucdf<-do.call(rbind.data.frame, nuc)
nucdf$cluster<- row.names(nucdf)

```

Now we want to connect the nucleotide and haplotype diversity to values of parasite species richness, ANDV phylogenetic richness, and environmental variables. Need to match lat long from parasites to pd to map back. This finds the lat long in parasites that is closest to pd and then assigns the parasite richness to that coordinate

```

seqPD2<-subset(seqPD, PD!=0) ## seqPD is from line 429 ;
for (i in 4:ncol(seqPD2)) {
  seqPD2[[i]] <- ifelse(seqPD2[[i]] == 1, seqPD2$PD, 0) ##replace binary vals with PD val
}

pdacc<-(seqPD2[,4:ncol(seqPD2)])
pdacc<-as.data.frame(cbind(unlist(pdacc, use.names = FALSE), rep(names(pdacc), each = nrow(pdacc))))
pdacc<-subset(pdacc, V1 !=0)
write.csv(seqPD2, "SeqPD.csv")

pdacc<- pdacc %>% group_by(V2) %>% top_n(1, V1) %>% distinct(V1, V2, .keep_all = TRUE) #get rid of dupl

```

```

colnames(pdacc)[1] <-"trait"
colnames(pdacc)[2] <-"label"

tree<- read.tree("ColiTreeApril23_bs.tree") #load phylogenetic tree

tips<- read.csv("tips_april23.csv", header=T)
# make tips match
pos_id <-match(tree$tip.label, tips$old)#element position
tree$tip.label <- tips$new[pos_id] #here sorting by pos_id

pdtable <- full_join(tree, pdacc, by = 'label')
pd <- as_tibble(pdtable)
pd$trait[is.na(pd$trait)] <- 0
pd$trait<- as.numeric(pd$trait)
pdtree<- as.treedata(pd)

dfpd <- reshape2::melt(pdacc, id = "label")
dfpd$value<- as.numeric(dfpd$value)

```

Parasites

```

seqPD2<- merge(dfpd_coli, layer_coli, by=c("x", "y"))
#match the lat and long from the parasites and pd
new<-dfpar_coli %>%
  rename(Latpar=x, Lonpar=y) %>%
  crossing(seqPD2) %>%
  mutate(dist=sqrt((x-Latpar)^2+(y-Lonpar)^2)) %>%
  group_by(x, y) %>%
  filter(dist==min(dist)) %>%
  ungroup() %>%
  select(x, y, PD, SR)

#merge the dbs
sr_seq<- merge(new, seqPD2, by=c("x", "y") )
sr_seq<- subset(sr_seq, PD.x !=0 | SR !=0)
for (i in 4:ncol(seqPD2)) {
  seqPD2[[i]] <- ifelse(seqPD2[[i]] == 1, seqPD2$PD, 0) ##replace binary vals with PD val
}
for (i in 6:ncol(sr_seq)) {
  sr_seq[[i]] <- ifelse(sr_seq[[i]] == 1, sr_seq$SR, 0) ##replace binary vals with Sr val
}

spacc<-(sr_seq[,6:ncol(sr_seq)])
spacc<-as.data.frame(cbind(unlist(spacc, use.names = FALSE), rep(names(spacc), each = nrow(spacc))))
spacc<-subset(spacc, V1 !=0)

spacc<- spacc %>% group_by(V2) %>% top_n(1, V1) %>% distinct(V1, V2, .keep_all = TRUE)
colnames(spacc)[1] <-"trait"
colnames(spacc)[2] <-"label"

df <- reshape2::melt(spacc, id = "label")
df$value<- as.numeric(df$value)## use df for parasite species richness

```

ANDV pd

```
seqPD2<- merge(dfpd_coli, layer_coli, by=c("x", "y"))
#match the lat and long from the parasites and pd
new_andv<-dfandv_coli %>%
  rename(LatPD=x, LonPD=y) %>%
  crossing(seqPD2) %>%
  mutate(dist=sqrt((x-LatPD)^2+(y-LonPD)^2)) %>%
  group_by(x, y) %>%
  filter(dist==min(dist)) %>%
  ungroup() %>%
  select(x, y, PDandv)

#merge the dbs
andv_seq<- merge(new_andv, seqPD2, by=c("x", "y") )
andv_seq<- subset(andv_seq, PD !=0 | PDandv !=0)

for (i in 5:ncol(andv_seq)) {
  andv_seq[[i]] <- ifelse(andv_seq[[i]] == 1, andv_seq$PDandv, 0) ##replace binary vals with Sr val
}

andv_acc<-(andv_seq[,5:ncol(andv_seq)])
andv_acc<-as.data.frame(cbind(unlist(andv_acc, use.names = FALSE), rep(names(andv_acc), each = nrow(andv_acc)), subset(andv_acc, V1 !=0)

andv_acc<- andv_acc %>% group_by(V2) %>% top_n(1, V1) %>% distinct(V1, V2, .keep_all = TRUE)
colnames(andv_acc)[1] <- "trait"
colnames(andv_acc)[2] <- "label"

tree<- read.tree("ColiTreeApril23_bs.tree") #load phylogenetic tree

tips<- read.csv("tips_april23.csv", header=T)
# make tips match
pos_id <-match(tree$tip.label, tips$old)#element position
colitree<-tree
colitree$tip.label <- tips$new[pos_id] #here sorting by pos_id
tips<-as.data.frame(tree$tip.label)
colnames(tips)[1] <- "label"
#dfed<- merge(edacc, tips, by=c("label"))
df_andv <- reshape2::melt(andv_acc, id = "label")
df_andv$value<- as.numeric(df_andv$value)
## use df for parasite species richness
```

Total Data frame with points traced back to accession coordinates

```
###df merge all ####
tot<- merge(dfpd, df, by= c("label"), all=T)
total_df<- merge(tot, df_andv, by=c("label"), all=T)

total_df$variable.x<- NULL
total_df$variable.y<-NULL
total_df$variable<-NULL

colnames(total_df)[2]<-"ColiPD"
colnames(total_df)[3]<-"ColiParasite"
```

```

colnames(total_df)[4] <- "andvPD"
colnames(cyt)[4] <- "label"

dftotal <- merge(total_df, cyt, by=c("label"), all=T)
write.csv(dftotal, "dftotal.csv") #traced to acc

get clim data for accession df
geodata::geodata_path("/Users/rbrennan/Library/Mobile Documents/com~apple~CloudDocs/Documents/VT/Project")
r <- geodata::worldclim_global(res=10, var="bio")
r <- r[[c(1,12)]]
names(r) <- c("Temp", "Prec")
coords <- data.frame(x=dftotal$Longitude, y=dftotal$Latitude)
points <- vect(coords, geom=c("x", "y"), crs ="+proj=longlat +datum=WGS84")

values <- terra::extract(r, points, xy=TRUE) #get the temp and precip for each point of diversity (gene)
df_acc_clim <- cbind.data.frame(values, dftotal) # same order
df_acc_clim$X <- NULL
write.csv(df_acc_clim, "df_acc_clim.csv")

Climate analysis by cluster
#sr_cluster <- merge(cyt, df_acc_clim, by=c('label'))

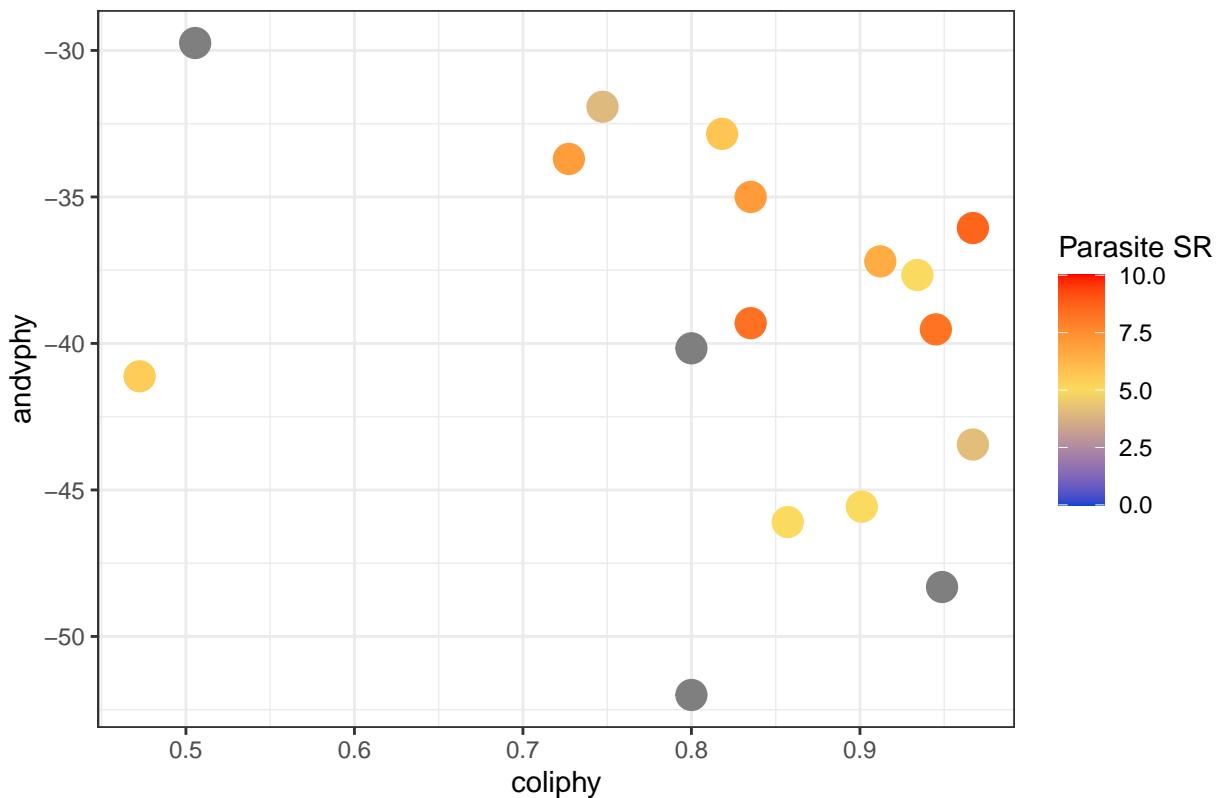
sum_sr <- df_acc_clim %>% group_by(cluster$clusters) %>% summarize(sumpar = mean(ColiParasite), latitude= mean(Longitude),
                                                                     longitude= mean(Longitude),
                                                                     coliphy = mean(ColiPD), andvphy = mean(temp=mean(Temp), prec=mean(Prec)))

colnames(sum_sr)[1] <- "cluster"
sr_hap <- merge(sum_sr, hapdf, by=c("cluster"))
colnames(sr_hap)[9] <- "hap"
coli_total <- merge(sr_hap, nucdf, by=c("cluster"))
colnames(coli_total)[10] <- "nuc"

ggplot(coli_total, aes(x=hap, y=latitude, color=sumpar)) + theme_bw() +
  labs(title="Haplotype Diversity by latitude",
       x="coliphy", y = "andvphy") +
  scale_color_gradient2(low="#0f46d1ff", mid="#fada5fff", high= "red", limits=c(0,10), midpoint= 5, na

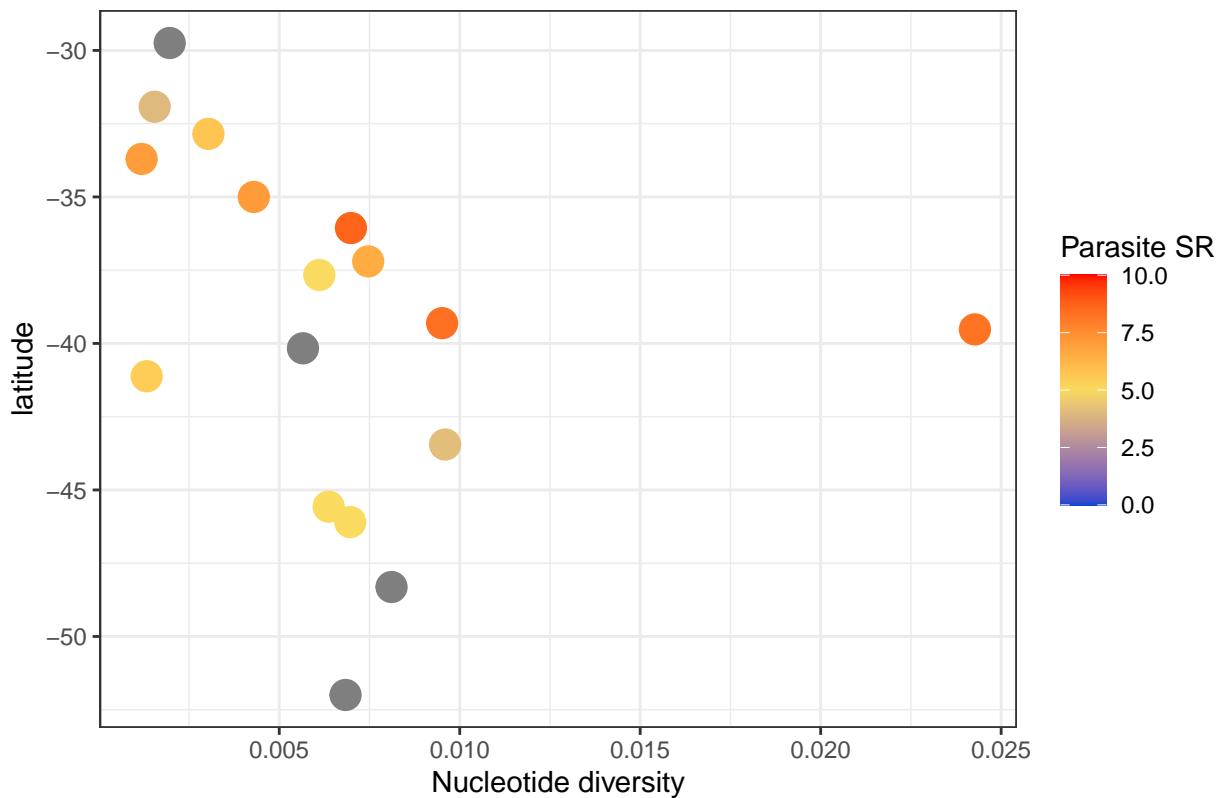
```

Haplotype Diversity by latitude



```
ggplot(coli_total, aes(x=nuc, y=latitude, color= sumpar)) + theme_bw() +  
  labs(title="Nucleotide Diversity by latitude",  
       x="Nucleotide diversity", y = "latitude") +  
  scale_color_gradient2(low="#0f46d1ff", mid="#fada5fff",high= "red",limits=c(0,10),midpoint= 5, na
```

Nucleotide Diversity by latitude



```
##Part 6. Build geographic and environmental space
get GBIF occurences for Oligoryzomys longicaudatus
species<- "Oligoryzomys longicaudatus"
gbif_taxon_keys <-
species %>% # use only first 1000 names for testing
name_backbone_checklist() %>% # match to backbone
filter(!matchType == "NONE") %>% # get matched names
pull(usageKey)

occ_download(
pred_in("taxonKey", gbif_taxon_keys),
format = "SIMPLE_CSV")

clean occurrences
data <- occ_download_get('0000269-250121130708018') %>%
  occ_download_import()

## Download file size: 0.23 MB
## file exists & overwrite=FALSE, not overwriting...
# select columns of interest
dat <- data %>%
  dplyr::select(species, decimalLongitude,
                decimalLatitude, countryCode, individualCount,
                gbifID, family, taxonRank, coordinateUncertaintyInMeters,
                year, basisOfRecord, institutionCode)
```

```

# remove records without coordinates
dat <- dat %>% filter(!is.na(decimalLongitude)) %>% filter(!is.na(decimalLatitude))
dat$countryCode <- countrycode(dat$countryCode,
                                origin = 'iso2c',
                                destination = 'iso3c')

#flag problems
dat <- data.frame(dat)
flags <- clean_coordinates(x = dat,
                            lon = "decimalLongitude",
                            lat = "decimalLatitude",
                            countries = "countryCode",
                            species = "species",
                            tests = c("capitals", "centroids",
                                      "equal", "zeros", "countries"))

## Testing coordinate validity
## Flagged 0 records.

## Testing equal lat/lön
## Flagged 0 records.

## Testing zero coordinates
## Flagged 0 records.

## Testing country capitals
## Flagged 0 records.

## Testing country centroids
## Flagged 0 records.

## Testing country identity
## Flagged 92 records.

## Flagged 92 of 4615 records, EQ = 0.02.

#Exclude problematic records
dat_cl <- dat[flags$.summary,] ## these the good ones

#The flagged records
dat_fl <- dat[!flags$.summary,]

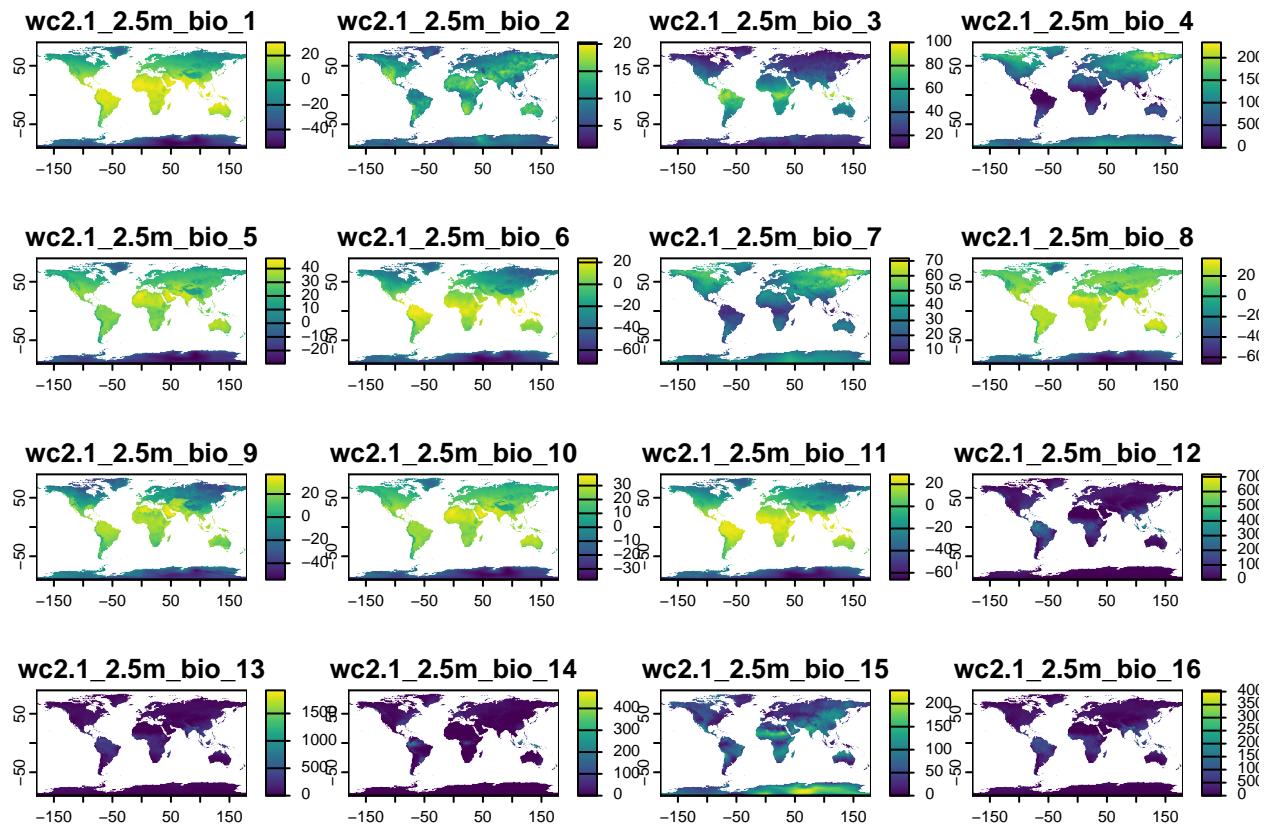
```

get world clim for occ data from GBIF

```

geodata::geodata_path("/Users/rbrennan/Library/Mobile Documents/com~apple~CloudDocs/Documents/VT/Projec
r<-geodata::worldclim_global(res=2.5, var="bio")
plot(r)

```



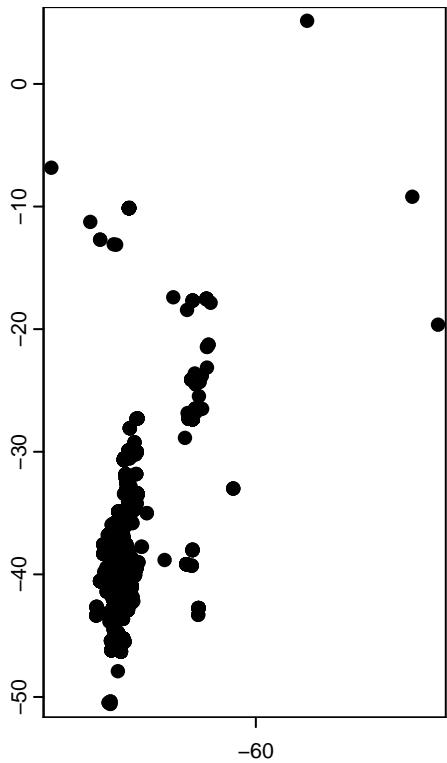
```

r<- r[[c(1,12)]]
names(r) <- c("Temp", "Prec")

names(dat_cl)[2] <- "x"
names(dat_cl)[3] <- "y"

coords<- data.frame(x=dat_cl$x, y=dat_cl$y)
points<-vect(coords, geom=c("x", "y"), crs ="+proj=longlat +datum=WGS84")
plot(points)

```



```

values <- terra::extract(r, points, xy=T) #extract clim variables from 'r' for 'points'
occ_bio <- cbind.data.frame(values, dat_cl) # same order
# where x.1 and y.1 are the original latlong and x and y are the closest from worldclim

write.csv(occ_bio, "occ_bio.csv")

```

Plot gbif occurrences and calculate mean temp and precip and difference from mean for all gbif occurrences

```

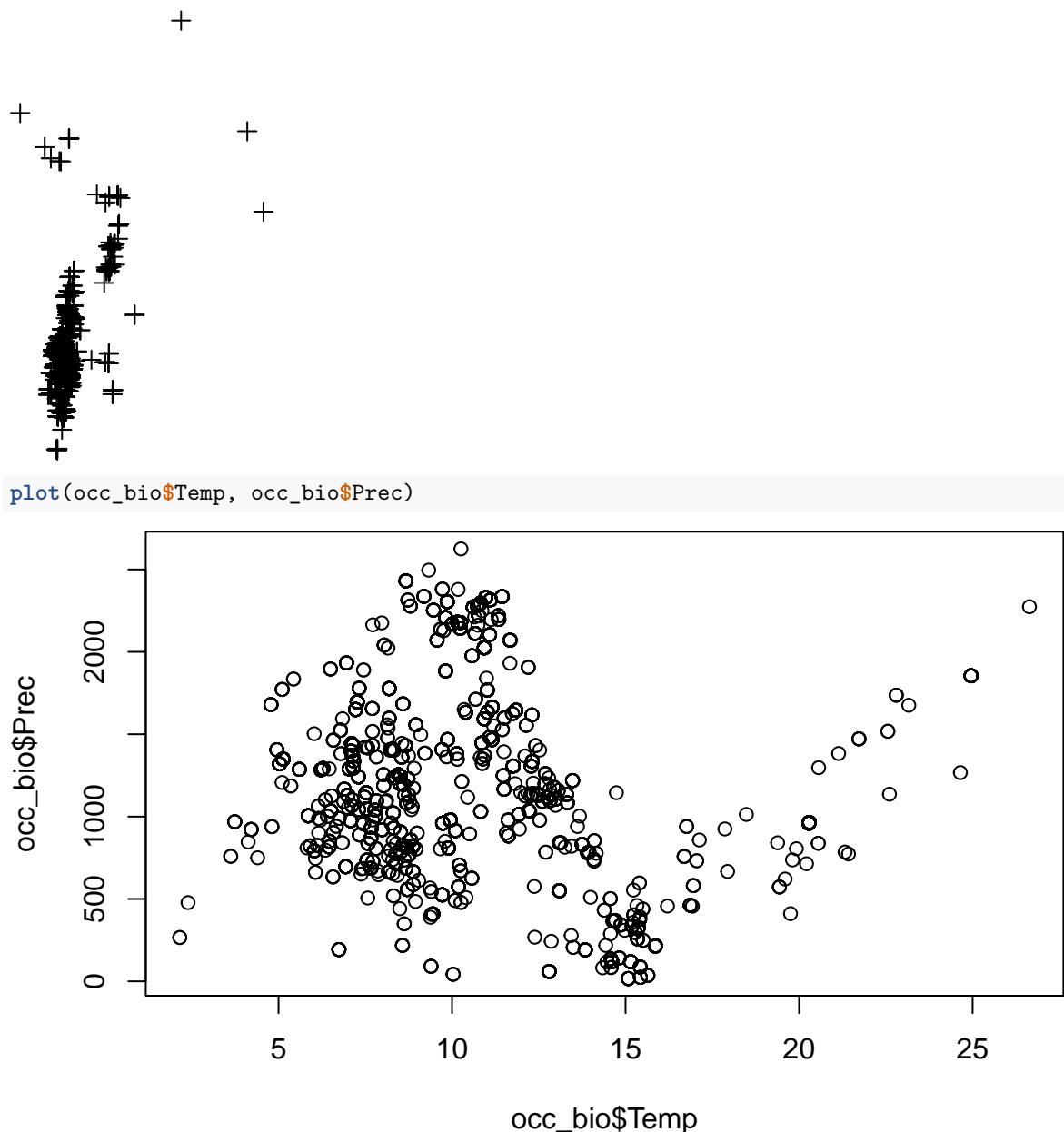
occ_bio<-read.csv("occ_bio.csv", header=T)
wallace<- data.frame(scientific_name=occ_bio$species, longitude= occ_bio$x.1, latitude=occ_bio$y.1)

sapply(wallace, anyNA)

## scientific_name      longitude      latitude
##          FALSE          FALSE          FALSE

write.csv(wallace, "ollo_wallace.csv") ##occurrences for wallace, to be added to lit review records
occ_bio$X<-NULL
occ_bio$ID<-NULL
occ_bio$species<-gsub(" ","_", occ_bio$species )
#rename columns
names(occ_bio)[3]<- "LonW"
names(occ_bio)[4]<- "LatW"
names(occ_bio)[6]<- "Longitude"
names(occ_bio)[7]<- "Latitude"
#check data by plotting
wgscoor<- occ_bio
coordinates(wgscoor) <- ~Longitude+Latitude
proj4string(wgscoor)<- CRS("+proj=longlat +datum=WGS84")
plot(wgscoor)

```



Calculate Center of range using gbif.range

```
# Download
obs_pt = get_gbif(sp_name = "Oligoryzomys longicaudatus",
                  basis = c("OBSERVATION", "HUMAN_OBSERVATION", "MACHINE_OBSERVATION"))

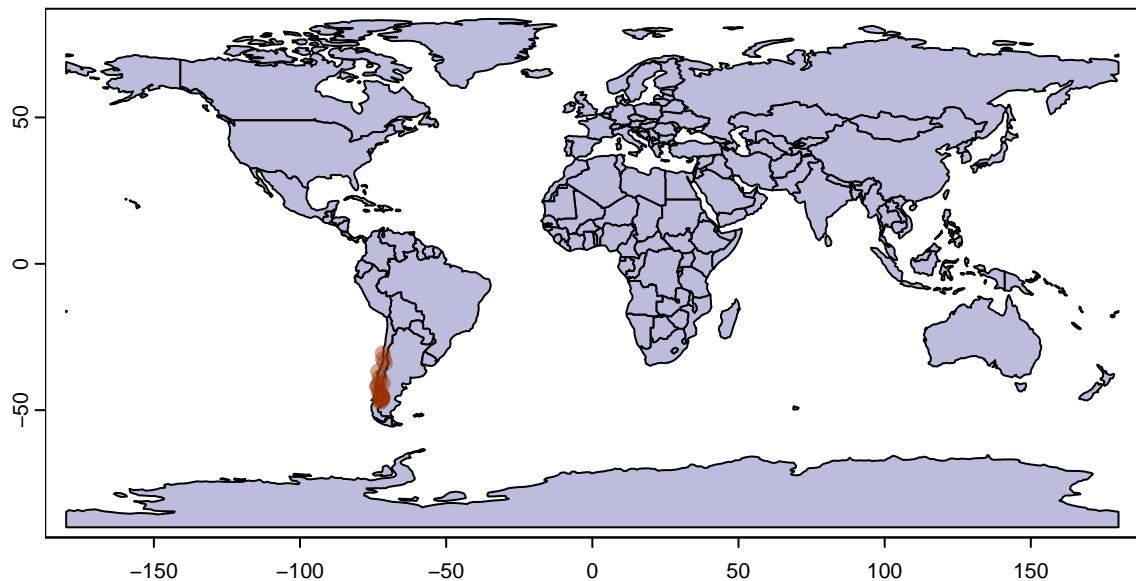
## >>>>> Total number of records: 4838
## ...GBIF records of Oligoryzomys longicaudatus : download of all records starting...
## 100 %...
## ---> Grain filtering...
## Records removed: 2408
## ---> Removal of duplicated records...
## Records removed: 2096
## ---> Removal of absence records...
## Records removed: 0
```

```

## ---> Basis of records selection...
## Records removed: 283
## ---> Time period selection...
## Records removed: 0
## ---> Removal of identical xy records...
## Records removed: 0
## ---> Removal of wrong lon/lat converted records...
## Records removed: 0
## ---> Removal of raster centroids...
## Records removed: 0

# Plot species records
countries = vect(ne_countries(type = "countries",returnclass = "sf"))
plot(countries,col = "#bcbddc")
points(obs_pt[,c("decimalLongitude","decimalLatitude")],pch=20,col="#99340470",cex=1.5)

```



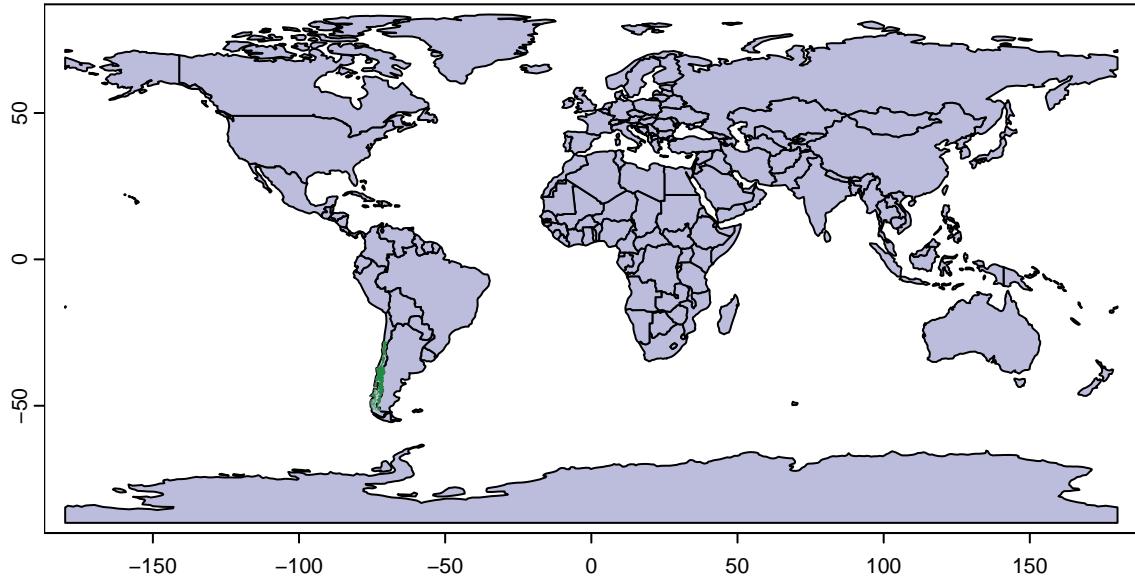
```

# Download ecoregion and read
eco_terra = read_bioreg(bioreg_name = "eco_terra", save_dir = NULL)
# Range
range_coli = get_range(occ_coord = obs_pt,
                      bioreg = eco_terra,
                      bioreg_name = "ECO_NAME")

## ## Start of computation for species: Oligoryzomys longicaudatus ##
## 3 outlier's from 23 | proportion from total points: 13%
## bioregion 1 of 3 : Chilean Matorral
## bioregion 2 of 3 : Magellanic Subpolar Forests
## bioregion 3 of 3 : Valdivian Temperate Forests
## |-----|-----|-----|-----|=====

plot(countries,col = "#bcbddc")
plot(range_coli,col = "#238b45",add = TRUE,axes = FALSE,legend = FALSE)

```



```
colirange <- terra::as.polygons(range_coli) #have to change to calculate centroid
colirange<-sf::st_as_sf(colirange)

cent<- st_centroid(st_geometry(colirange)) #calculate centroid
coords_cent <- st_coordinates(cent)
```

get world clim data for the parasite/genetic occurrences

```
geodata::geodata_path("/Users/rbrennan/Library/Mobile Documents/com~apple~CloudDocs/Documents/VT/Project"
r<-geodata::worldclim_global(res=10, var="bio")
```

```
r<- r[[c(1,12)]]
names(r) <- c("Temp","Prec")
```

```
subdf<- read.csv("climate_df.csv", header=T) # from 537 this has the Phylo Div, ANDV div, and Parasite ...
subdf[1]<-NULL
coords<- data.frame(x=subdf$x, y=subdf$y)
points<-vect(coords, geom=c("x","y"), crs ="+proj=longlat +datum=WGS84")
```

```
values <- terra::extract(r, points, xy=TRUE) #get the temp and precip for each point of diversity (gene
occ_bio_par <- cbind.data.frame(values, subdf) # same order
occ_bio_par[4]<-NULL
occ_bio_par[4]<-NULL
```

Calculate the distance to range centroid geo

```
##CENTROID from gbif.range GEOGRAPHIC##
x <- c(coords_cent[, "X"])
y <- c(coords_cent[, "Y"])

centdf<- as.matrix(data.frame(x,y))
occdf<- as.matrix(data.frame(occ_bio_par$x, occ_bio_par$y))

dist2cent<-distm(occdf,centdf, fun = distHaversine)
occ_bio_par$distcent<- dist2cent
```

```

##Part 7. Suitability
Suitability logistic

Chile<- read_sf("./CHL_adm/CHL_adm0.shp")
Argentina <-read_sf("./ARG_adm/ARG_adm0.shp")
SA <- st_union(Chile, Argentina)

## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries

##Transform suitability tif from maxent into map and dataframe
suit<- rast(x="maxent_gbiflit_accepted_block/Oligoryzomys_longicaudatus_fc.L_rm.0.5_logistic.tif")
test2 <- terra::crop(suit, SA)
ext <- floor(ext(SA))
nr <- rast(ext, res = res(test2))
raster::values(nr) <- 0
testnew <- terra::mask(nr,SA)

tn_ras<-rast(testnew)
t2_ras<-rast(test2)

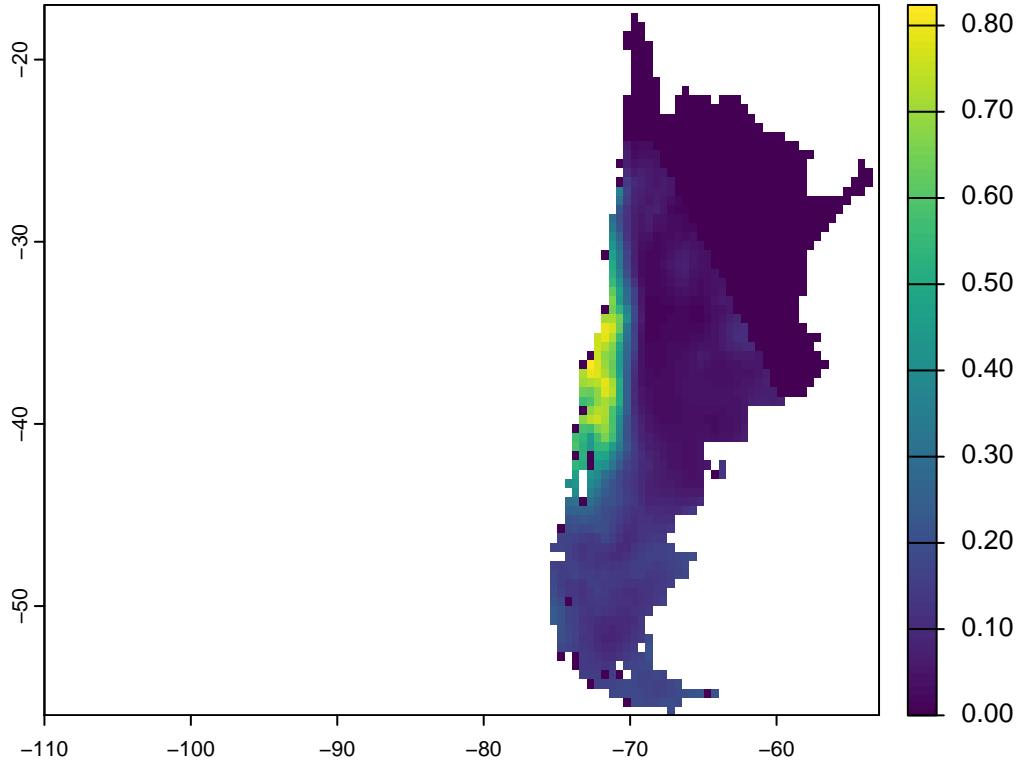
res(tn_ras) <- 0.5
res(t2_ras)<-0.5
crs(t2_ras)<-crs(tn_ras)
testnew <- resample(testnew, tn_ras)
test2<-resample(test2, t2_ras)

plot(test2)

```

plot_suit<- mosaic(test2,testnew, fun="max")

```
suit_plot <- terra::mask(plot_suit, SA)
plot(suit_plot)
```



```
suit_df<- as.data.frame(suit_plot, xy=TRUE)
names(suit_df)[3]<-"suit"
```

Binary map

```
# Load original binary raster
bin <- rast("maxent_gbiflit_accepted_block/Oligoryzomys_longicaudatus_fc.L_rm.0.5_p10.tif")

# Binarize in case of float values
bin[] <- ifelse(bin[] >= 0.5, 1, 0)

# Crop and mask to SA shape
bin_crop <- crop(bin, SA)
bin_mask <- terra::mask(bin_crop, SA)

# MATCH GRID: Use suit_plot as the template (has correct res, extent, crs, alignment)
template <- suit_plot # already aligned to SA, 0.5 deg

# --- KEY STEP: aggregate BEFORE resample to preserve sparse 1s ---
# Estimate aggregation factor between bin and template resolution
fact_x <- round(res(template)[1] / res(bin_mask)[1])
fact_y <- round(res(template)[2] / res(bin_mask)[2])
fact <- c(fact_x, fact_y)

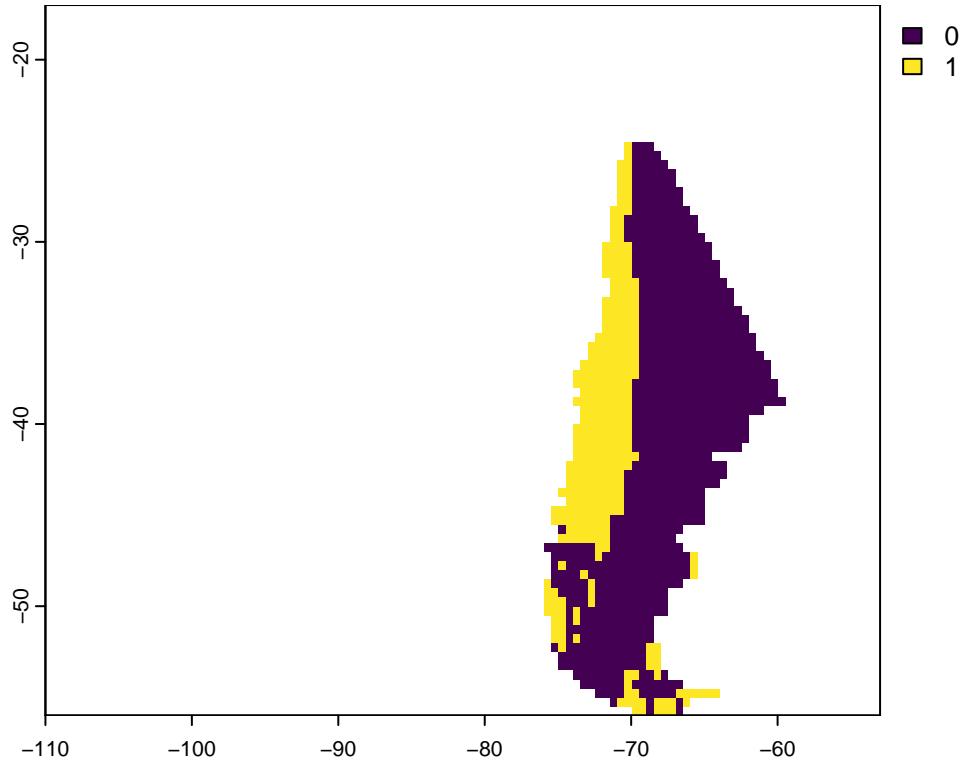
# Aggregate with max: if any underlying pixel is 1, keep 1
bin_agg <- aggregate(bin_mask, fact = fact, fun = "max", na.rm = TRUE)
```

```

# Align to template exactly (grid matching)
bin_res <- resample(bin_agg, template, method = "near")

# Final mask (in case of slight edge artifacts)
bin_plot <- terra::mask(bin_res, SA)
plot(bin_plot)

```



```

# Convert to df
bin_df <- as.data.frame(bin_plot, xy = TRUE)
names(bin_df)[3] <- "binary"

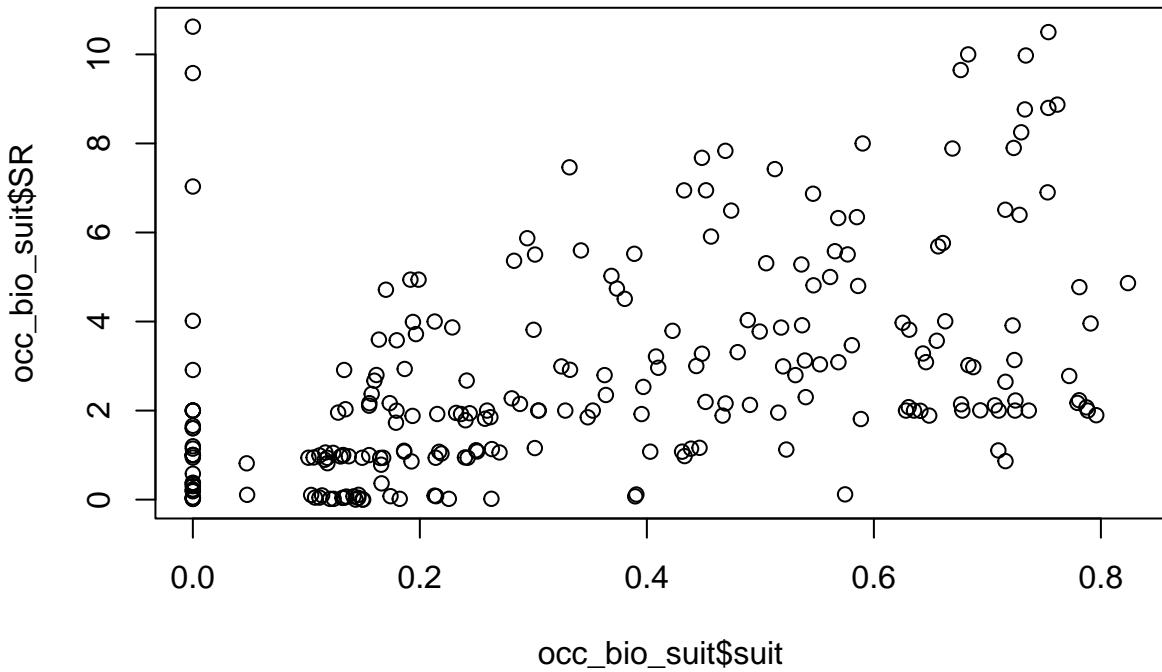
```

```

database merge
#####
occ_bio_suit<- merge(suit_df, occ_bio_par, by=c("x", "y"))
#occ_bio_suit2<- read.csv("occ_bio_suit.csv", header=T) ##original file for next steps of plotting and
occ_bio_suit$X<-NULL
occ_bio_suit2<- merge(occ_bio_suit, bin_df, by=c("x", "y"))

plot(occ_bio_suit$suit, occ_bio_suit$SR)

```



```
suit_plot_nozero<- subset(suit_df, suit!=0)
suitzero<- subset(suit_df, suit == 0)
```

Plots

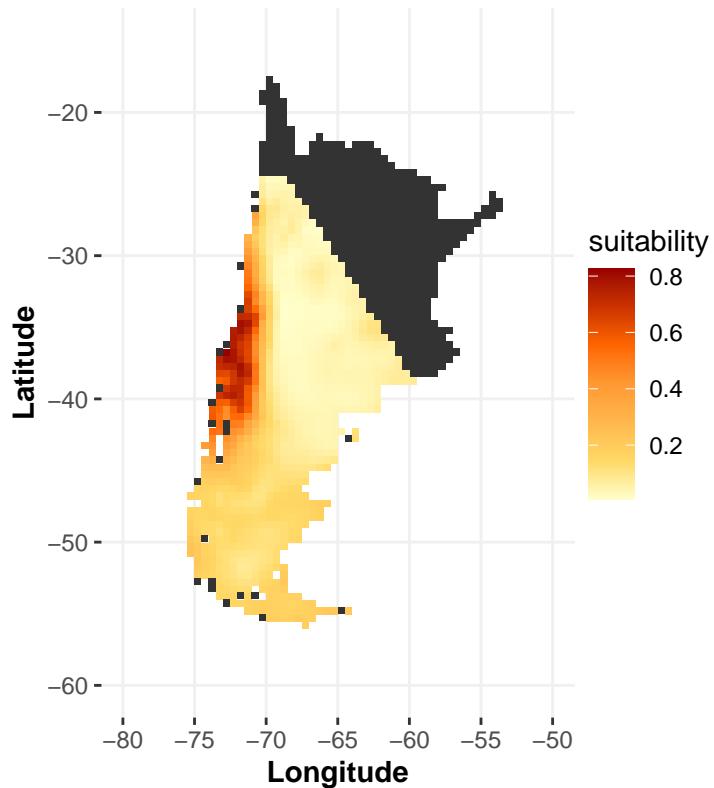
```
xlim<-c(-80, -50)
ylim<-c(-60, -15)
suit_map<-ggplot(data = suit_plot_nozero) +
  geom_raster(aes(x = x, y = y, fill = `suit`)) +
  scale_fill_gradientn(colours = c( "#FFFFCC", "#FFD966", "#FFB84D",
    "#FF9933", "#FF6600", "#CC3300", "#990000"),
    breaks = c(0, 0.2, 0.4, 0.6, 0.8, 1),
    labels= c("0", "0.2", "0.4", "0.6", "0.8", "1"),
    name="suitability")+
  geom_raster(data=suitzero, aes(x = x, y = y), show.legend = F)+

  theme(legend.position = "right",legend.box.spacing = unit(0, "pt"), panel.background = element_rect(
    panel.grid = element_line(color = "gray94"),
    axis.title = element_text(face="bold"),
    plot.title = element_text(hjust = 0.5)) +
  coord_sf(xlim=xlim, ylim=ylim) +
  labs(x="Longitude", y= "Latitude")+
  ggtitle(expression(paste(bold("Suitability of "), bolditalic("O. longicaudatus")))))

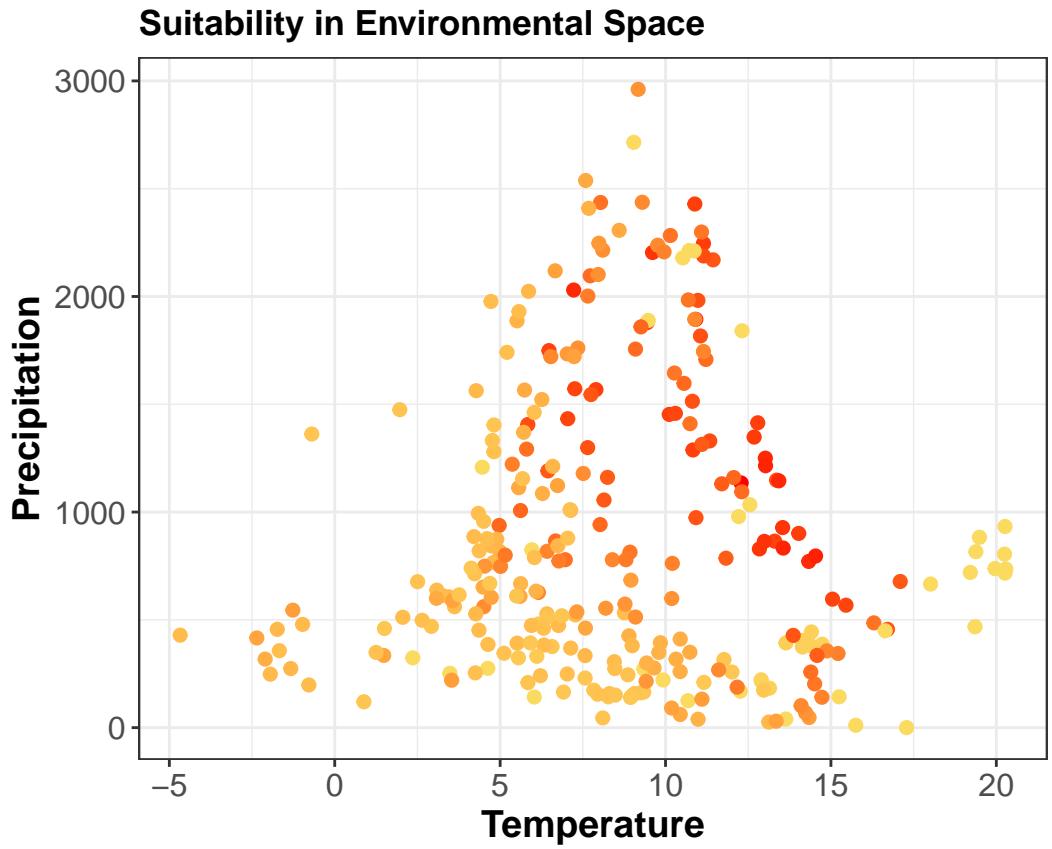
envSuit<- ggplot(occ_bio_suit, aes(x=Temp, y=Prec, color=suit)) + theme_bw()+
  labs(title="Suitability in Environmental Space",
    x="Temperature", y = "Precipitation", size=100) + scale_color_gradient2(mid="#fada5fff", high="#")
  axis.title=element_text(size=14,face="bold"), plot.title = element_text(size=13, face="bold"))

suit_map
```

Suitability of *O. longicaudatus*



envSuit



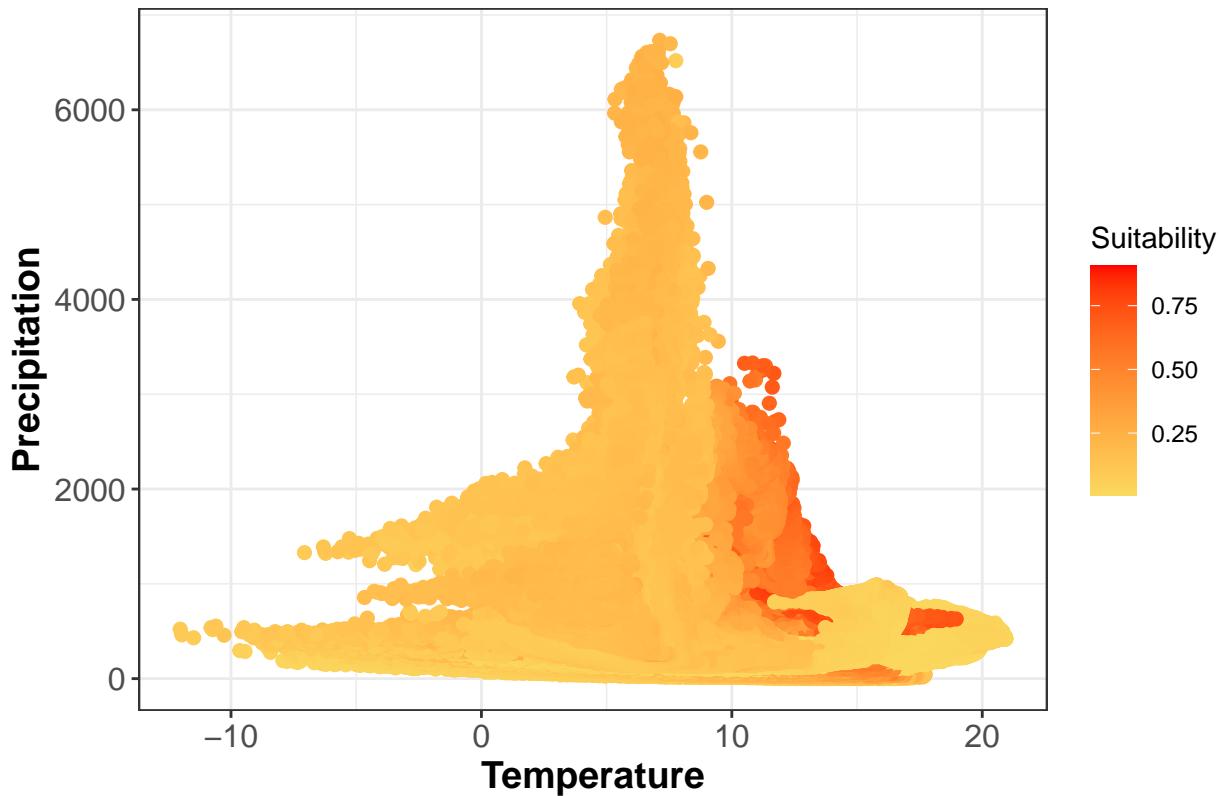
plots from wallace data direct

```
wallace_df<- as.data.frame (suit, xy=TRUE)
names(wallace_df)[3]<-"suit"
geodata::geodata_path("/Users/rbrennan/Documents/BePhyNE")
r<-geodata::worldclim_global(res=2.5, var="bio")
r<- r[[c(1,12)]]
names(r) <- c("Temp", "Prec")

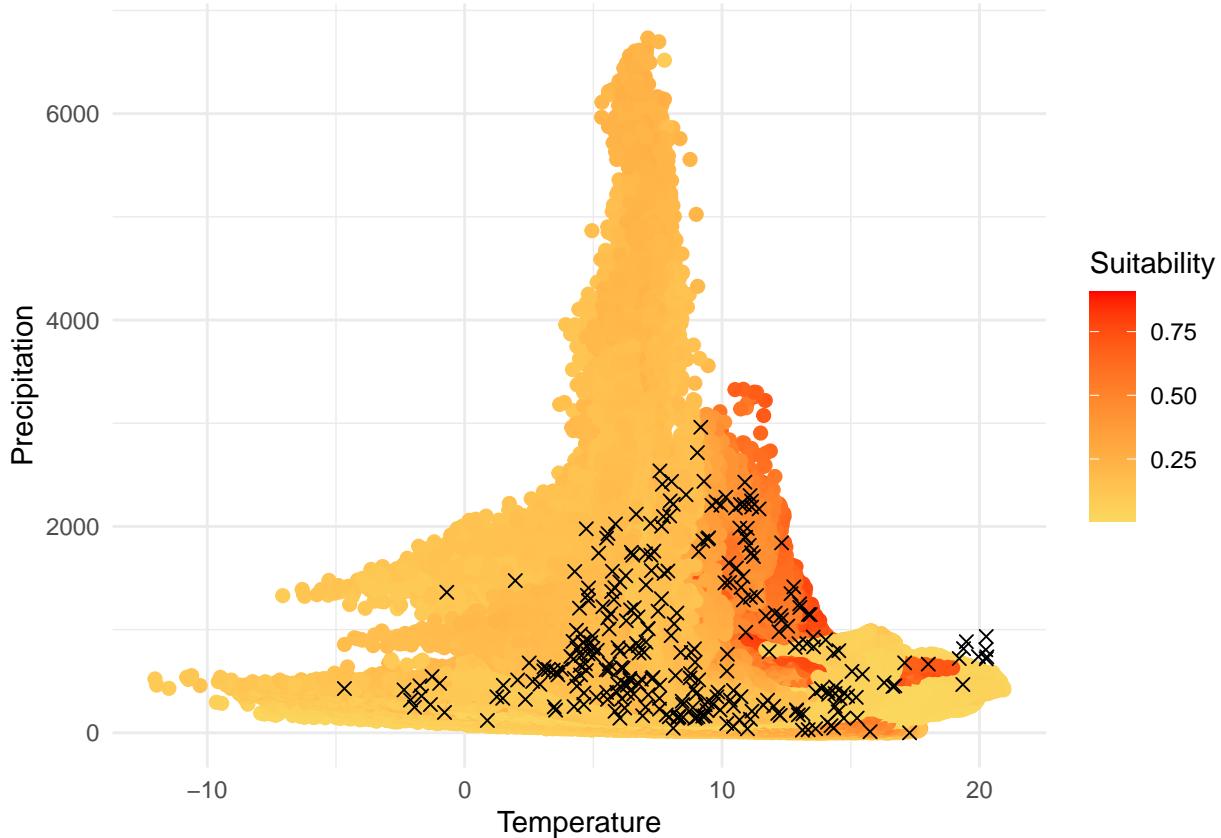
coords<- data.frame(x=wallace_df$x, y=wallace_df$y)
points<-vect(coords, geom=c("x", "y"), crs ="+proj=longlat +datum=WGS84")

values <- terra::extract(r, points, xy=TRUE) #extract clim variables from 'r' for 'points'
wallace_values <- cbind.data.frame(values, wallace_df)
wallace_values[6]<-NULL
wallace_values[6]<-NULL
wallace_env<- ggplot(wallace_values, aes(x=Temp, y=Prec, color=suit)) + theme_bw()+
  labs(title="Habitat Suitability in Environmental Space", x="Temperature", y = "Precipitation", size=10,
       geom_point(size=2) + theme(axis.text=element_text(size=12),
       axis.title=element_text(size=14,face="bold")), plot.title = element_text(size=13, face="bold"))
wallace_env
```

Habitat Suitability in Environmental Space



```
plot<-ggplot() +
  # Plot wallace_values points with color by suit
  geom_point(data = wallace_values, aes(x = Temp, y = Prec, color = suit), size = 2) +
  # Overlay occ_bio_suit as black "x" points
  geom_point(data = occ_bio_suit, aes(x = Temp, y = Prec), shape = 4, color = "black", size = 2) +
  # Optional: continuous color scale and styling
  scale_color_gradient2(mid="#fada5fff", high="red",name="Suitability") +
  theme_minimal() +
  labs(x = "Temperature", y = "Precipitation", color = "Suitability")
plot
```



south america

```
library(rmapshaper)
Chile<- read_sf("./CHL_adm/CHL_adm0.shp")
Argentina <-read_sf("./ARG_adm/ARG_adm0.shp")
SA <- st_union(Chile, Argentina)

## Warning: attribute variables are assumed to be spatially constant throughout
## all geometries
SA_simple<- rmapshaper::ms_simplify(SA, keep = 0.05)
```

Dist to gbif centroid from thinned data only gbif data used for first suitability analysis, from Chile/Arg in accepted range

```
gbif_thin<- read.csv("Oligoryzomys_longicaudatus_user_raw_GBIF_ChileArg.csv", header=T)
coords<- data.frame(x=gbif_thin$longitude, y=gbif_thin$latitude)
points_gbif<-vect(coords, geom=c("x","y"), crs ="+proj=longlat +datum=WGS84")

points_gbif_sf <- sf::st_as_sf(points_gbif)

# Plot South America with points
mapgbif<-ggplot() +
  geom_sf(data = SA_simple, fill = "gray95", color = "black") +
  geom_sf(data = points_gbif_sf, color = "red", size = 0.5) +
  coord_sf(xlim = c(-85, -50), ylim = c(-60, -15), expand = FALSE) +
  theme_minimal() +
  labs(title = "GBIF Accepted over Chile and Argentina",
       x = "Longitude", y = "Latitude")
```

```

values <- terra::extract(r, points_gbif, xy=TRUE) #get the temp and precip for each point of diversity
gbif_thin<- cbind.data.frame(values, gbif_thin) # same order

centroid_gbif<- data.frame(Temp = mean(gbif_thin$Temp, na.rm = TRUE),
                             Prec = mean(gbif_thin$Prec, na.rm = TRUE))

# Combine the centroid and occ_bio_suit into a single matrix
combined_gbif <- rbind(centroid_gbif, occ_bio_suit[, c("Temp", "Prec")])

# Use dist() to calculate Euclidean distances
dists_gbif <- dist(combined_gbif, method = "euclidean")

# Extract the distances from the centroid (row 1) to all other rows
# dist() returns the lower triangle of the distance matrix as a vector,
# so we need to extract only the distances from the first row to the rest.
# The first n-1 elements of the result will be those.
dist_to_gbif <- as.matrix(dists_gbif)[-1, 1]

# Add to the data frame
occ_bio_suit$dist_to_gbif <- dist_to_gbif

```

dist to my points lit points only

```

litrev<- read.csv("Genetic_Parasite_Coords.csv", header=T)
litrev$Longitude<-as.numeric(litrev$Longitude)

```

Warning: NAs introduced by coercion

```

coords<- data.frame(x=litrev$Longitude, y=litrev$Latitude)
points_lit<-vect(coords, geom=c("x", "y"), crs ="+proj=longlat +datum=WGS84")

```

```

points_lit_sf <- sf::st_as_sf(points_lit)

```

Plot South America with points

```

map_lit<-ggplot() +
  geom_sf(data = SA_simple, fill = "gray95", color = "black") +
  geom_sf(data = points_lit_sf, color = "red", size = 0.5) +
  coord_sf(xlim = c(-85, -50), ylim = c(-60, -15), expand = FALSE) +
  theme_minimal() +
  labs(title = "Points from literature review",
       x = "Longitude", y = "Latitude")

```

```

values <- terra::extract(r, points_lit, xy=TRUE) #get the temp and precip for each point of diversity ()
litrev<- cbind.data.frame(values, litrev) # same order
#### dif to my POINTS
centroid_lit<- data.frame(Temp = mean(litrev$Temp, na.rm = TRUE),
                           Prec = mean(litrev$Prec, na.rm = TRUE))

# Combine the centroid and occ_bio_suit into a single matrix
combined_lit<- rbind(centroid_lit, occ_bio_suit[, c("Temp", "Prec")])

```

```

# Use dist() to calculate Euclidean distances
dists_lit <- dist(combined_lit, method = "euclidean")

# Extract the distances from the centroid (row 1) to all other rows
# dist() returns the lower triangle of the distance matrix as a vector,
# so we need to extract only the distances from the first row to the rest.
# The first n-1 elements of the result will be those.
dist_to_lit <- as.matrix(dists_lit)[-1, 1]

# Add to the data frame
occ_bio_suit$dist_to_lit <- dist_to_lit

```

dist to realized accepted lit and gbif

```

allacc<- read.csv("Oligoryzomys_longicaudatus_processed_occs.csv", header=T)

coords_allacc<- data.frame(x=allacc$longitude, y=allacc$latitude)
points_allacc<-vect(coords_allacc, geom=c("x","y"), crs ="+proj=longlat +datum=WGS84")

points_allacc_sf <- sf::st_as_sf(points_allacc)

# Plot South America with points
map_allacc<-ggplot() +
  geom_sf(data = SA_simple, fill = "gray95", color = "black") +
  geom_sf(data = points_allacc_sf, color = "red", size = 0.5) +
  coord_sf(xlim = c(-85, -50), ylim = c(-60, -15), expand = FALSE) +
  theme_minimal() +
  labs(title = "Points from GBIF and Lit in accepted range",
       x = "Longitude", y = "Latitude")

```

values <- terra::extract(r, points_allacc, xy=TRUE) #get the temp and precip for each point of diversity

```

allacc<- cbind.data.frame(values, allacc) # same order
### dif to my POINTS
centroid_allacc<- data.frame(Temp = mean(allacc$Temp, na.rm = TRUE),
                                Prec = mean(allacc$Prec, na.rm = TRUE))

# Combine the centroid and occ_bio_suit into a single matrix
combined_allacc<- rbind(centroid_allacc, occ_bio_suit[, c("Temp", "Prec")])

# Use dist() to calculate Euclidean distances
dists_allacc <- dist(combined_allacc, method = "euclidean")

# Extract the distances from the centroid (row 1) to all other rows
# dist() returns the lower triangle of the distance matrix as a vector,
# so we need to extract only the distances from the first row to the rest.
# The first n-1 elements of the result will be those.
dist_to_allacc <- as.matrix(dists_allacc)[-1, 1]

# Add to the data frame
occ_bio_suit$dist_to_allacc <- dist_to_allacc

```

dist to full realized centroid: lit points plus all Gbif chile arg

```

full_rec<-read.csv("Oligoryzomys_longicaudatus_processed_occs_GBIF_Lit_ChileArg.csv", header=T)

coords<- data.frame(x=full_rec$longitude, y=full_rec$latitude)
points_full<-vect(coords, geom=c("x","y"), crs ="+proj=longlat +datum=WGS84")

points_full_sf <- sf::st_as_sf(points_full)

# Plot South America with points
map_full<-ggplot() +
  geom_sf(data = SA_simple, fill = "gray95", color = "black") +
  geom_sf(data = points_full_sf, color = "red", size = 0.5) +
  coord_sf(xlim = c(-85, -50), ylim = c(-60, -15), expand = FALSE) +
  theme_minimal() +
  labs(title = "GBIF and Lit Review",
       x = "Longitude", y = "Latitude")

values <- terra::extract(r, points_full, xy=TRUE) #get the temp and precip for each point of diversity
full_rec<- cbind.data.frame(values, full_rec) # same order

centroid_realized<- data.frame(Temp = mean(full_rec$Temp, na.rm = TRUE),
                                   Prec = mean(full_rec$Prec, na.rm = TRUE))

# Combine the centroid and occ_bio_suit into a single matrix
combined_realized<- rbind(centroid_realized, occ_bio_suit[, c("Temp", "Prec")])

# Use dist() to calculate Euclidean distances
dists_real <- dist(combined_realized, method = "euclidean")

# Extract the distances from the centroid (row 1) to all other rows
# dist() returns the lower triangle of the distance matrix as a vector,
# so we need to extract only the distances from the first row to the rest.
# The first n-1 elements of the result will be those.
dist_to_realized <- as.matrix(dists_real)[-1, 1]

# Add to the data frame
occ_bio_suit$dist_to_realized <- dist_to_realized

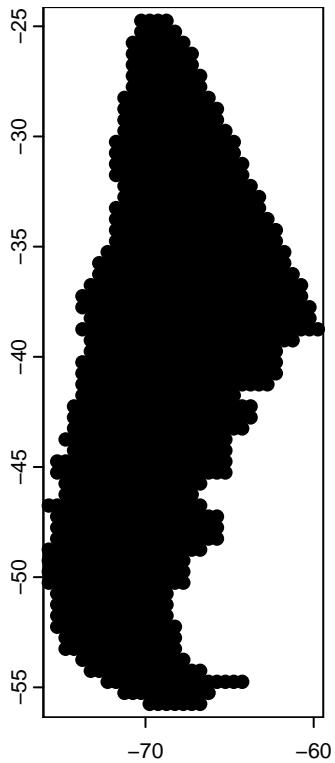
```

Binary environmental space and fundamental centroid

```

coords_bin<- data.frame(x=bin_df$x, y=bin_df$y)
points_bin<-vect(coords_bin, geom=c("x","y"), crs ="+proj=longlat +datum=WGS84")
plot(points_bin)

```



```

values_bin <- terra::extract(r, points_bin, xy=TRUE) #extract clim variables from 'r' for 'points'
bin_values <- cbind.data.frame(values_bin, bin_df)
bin_values[6]<-NULL
bin_values[6]<-NULL
bin1<- subset(bin_values, binary!=0)

bin_env<- ggplot(bin1, aes(x=Temp, y=Prec, color=binary)) + theme_bw()+
  labs(title="Binary Suitability in Environmental Space", x="Temperature", y = "Precipitation", size=10)
  geom_point(size=2) + theme(axis.text=element_text(size=12),
    axis.title=element_text(size=14,face="bold"), plot.title = element_text(size=13, face="bold"))

# Define centroid as a one-row data frame or matrix
centroid <- data.frame(Temp = mean(bin_values$Temp, na.rm = TRUE),
  Prec = mean(bin_values$Prec, na.rm = TRUE))

# Combine the centroid and occ_bio_suit into a single matrix
combined <- rbind(centroid, occ_bio_suit[, c("Temp", "Prec")])

# Use dist() to calculate Euclidean distances
dists <- dist(combined, method = "euclidean")

# Extract the distances from the centroid (row 1) to all other rows
# dist() returns the lower triangle of the distance matrix as a vector,
# so we need to extract only the distances from the first row to the rest.
# The first n-1 elements of the result will be those.
dist_to_centroid <- as.matrix(dists)[-1, 1]

# Add to the data frame

```

```

occ_bio_suit$dist_to_centroid <- dist_to_centroid

distance to fundamental from big Suit analysis

bin2 <- rast("maxent_gbiflit_all_block/Oligoryzomys_longicaudatus_fc.L_rm.0.5_p10.tif")

# Binarize in case of float values
bin2[] <- ifelse(bin2[] >= 0.5, 1, 0)

# Crop and mask to SA shape
bin_crop2 <- crop(bin2, SA)
bin_mask2 <- terra::mask(bin_crop2, SA)

# MATCH GRID: Use suit_plot as the template (has correct res, extent, crs, alignment)
template <- suit_plot # already aligned to SA, 0.5 deg

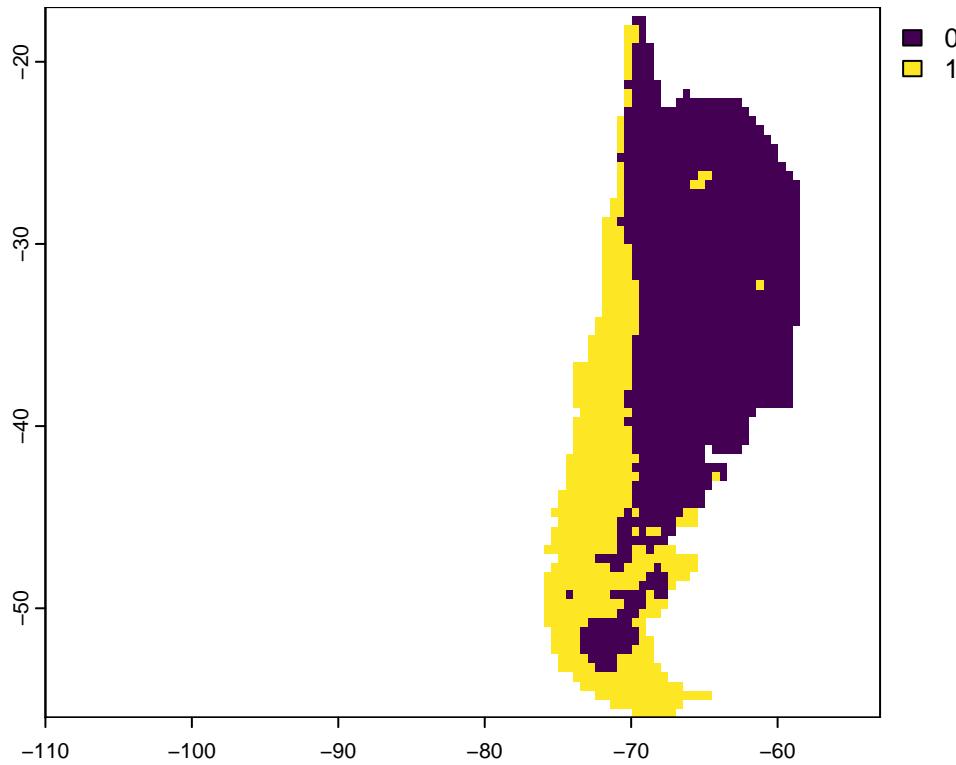
# --- KEY STEP: aggregate BEFORE resample to preserve sparse 1s ---
# Estimate aggregation factor between bin and template resolution
fact_x2<- round(res(template)[1] / res(bin_mask2)[1])
fact_y2 <- round(res(template)[2] / res(bin_mask2)[2])
fact2 <- c(fact_x2, fact_y2)

# Aggregate with max: if any underlying pixel is 1, keep 1
bin_agg2 <- aggregate(bin_mask2, fact = fact, fun = "max", na.rm = TRUE)

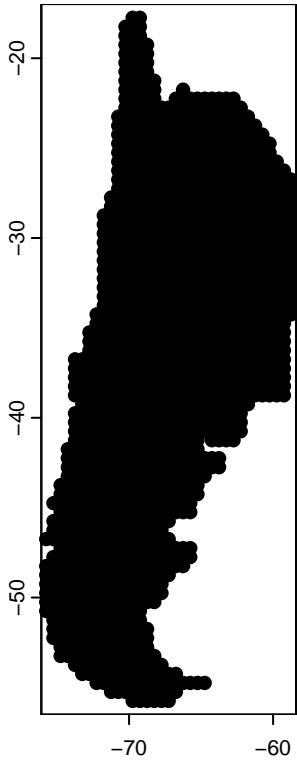
# Align to template exactly (grid matching)
bin_res2 <- resample(bin_agg2, template, method = "near")

# Final mask (in case of slight edge artifacts)
bin_plot2 <- terra::mask(bin_res2, SA)
plot(bin_plot2)

```



```
# Convert to df
bin_df2 <- as.data.frame(bin_plot2, xy = TRUE)
names(bin_df2)[3] <- "binary"
#####
coords_bin2<- data.frame(x=bin_df2$x, y=bin_df2$y)
points_bin2<-vect(coords_bin2, geom=c("x","y"), crs ="+proj=longlat +datum=WGS84")
plot(points_bin2)
```



```

values_bin2 <- terra::extract(r, points_bin2, xy=TRUE) #extract clim variables from 'r' for 'points'
bin_values2 <- cbind.data.frame(values_bin2, bin_df2)
bin_values2[6]<-NULL
bin_values2[6]<-NULL
bin2<- subset(bin_values2, binary!=0)

bin_env2<- ggplot(bin2, aes(x=Temp, y=Prec, color=binary)) + theme_bw()+
  labs(title="Binary Suitability in Environmental Space", x="Temperature", y = "Precipitation", size=10)
  geom_point(size=2) + theme(axis.text=element_text(size=12),
    axis.title=element_text(size=14,face="bold"), plot.title = element_text(size=13, face="bold"))

# Define centroid as a one-row data frame or matrix
centroid2 <- data.frame(Temp = mean(bin_values2$Temp, na.rm = TRUE),
  Prec = mean(bin_values2$Prec, na.rm = TRUE))

# Combine the centroid and occ_bio_suit into a single matrix
combined2 <- rbind(centroid2, occ_bio_suit[, c("Temp", "Prec")])

# Use dist() to calculate Euclidean distances
dists2 <- dist(combined2, method = "euclidean")

# Extract the distances from the centroid (row 1) to all other rows
# dist() returns the lower triangle of the distance matrix as a vector,
# so we need to extract only the distances from the first row to the rest.
# The first n-1 elements of the result will be those.
dist_to_centroid2 <- as.matrix(dists2)[-1, 1]

# Add to the data frame

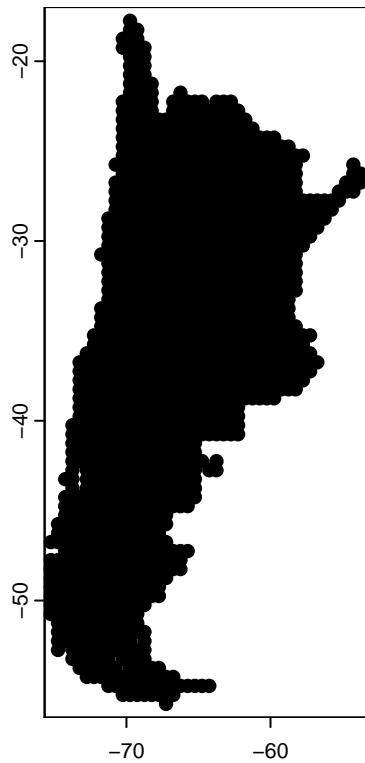
```

```
occ_bio_suit$dist_to_centroid2 <- dist_to_centroid2
```

those good good plots

```
geodata::geodata_path("/Users/rbrennan/Documents/BePhyNE")
r<-geodata::worldclim_global(res=2.5, var="bio")
r<- r[[c(1,12)]]
names(r) <- c("Temp", "Prec")

coords<- data.frame(x=suit_df$x, y=suit_df$y)
points<-vect(coords, geom=c("x", "y"), crs ="+proj=longlat +datum=WGS84")
plot(points)
```



```
values <- terra::extract(r, points, xy=TRUE) #extract clim variables from 'r' for 'points'
suit_tp <- cbind.data.frame(values, suit_df)
suit_tp[6]<-NULL
##continuous

suit_tp <- suit_tp[order(suit_tp$suit), ]

library(ggplot2)
library(ggnewscale)

centroid_all <- rbind(
  data.frame(centroid, source = "Fundamental Centroid"),
  data.frame(centroid_allacc, source = "Realized Centroid"),
  data.frame(centroid_realized, source= "All Data Realized Centroid"),
  data.frame(centroid2, source= "All Data Fundamental Centroid"),
  data.frame(centroid_lit, source = "Literature Records Centroid"),
  data.frame(centroid_gbif, source="GBIF Records Centroid")
```

```

)

# Prepare centroid source factor
centroid_all$source <- factor(centroid_all$source,
                                levels = c("Fundamental Centroid",
                                          "Realized Centroid",
                                          "All Data Realized Centroid",
                                          "All Data Fundamental Centroid",
                                          "Literature Records Centroid",
                                          "GBIF Records Centroid"))

# Split suitability
suit_zero <- subset(suit_tp, suit == 0)
suit_nonzero <- subset(suit_tp, suit > 0)

cont2<-ggplot() +
  # Suitability == 0 (gray), shown with legend
  geom_point(data = suit_zero, aes(x = Temp, y = Prec, color = "Not Suitable"),
             size = 2) +
  scale_color_manual(
    name= NULL,
    values = c("Not Suitable" = "gray")
  ) +
  new_scale_color() +
  # Suitability > 0 (yellow to red gradient)
  geom_point(data = suit_nonzero, aes(x = Temp, y = Prec, color = suit), size = 2) +
  scale_color_gradient2(mid = "#fada5fff", high = "red", name = "Suitability") +
  new_scale_color() +
  # Literature X points
  geom_point(data = occ_bio_suit, aes(x = Temp, y = Prec, shape = "Literature Record"),
             color = "black", size = 2) +
  # Centroids: shapes + fill color
  geom_point(data = centroid_all, aes(x = Temp, y = Prec, shape = source, fill = source),
             size = 5, stroke = 0.8) +
  # Shape legend (combined for X and centroids)
  scale_shape_manual(
    name = "Record Type",
    values = c("Literature Record" = 4,
              "Fundamental Centroid" = 21,
              "Realized Centroid" = 24,
              "All Data Realized Centroid"= 25,
              "All Data Fundamental Centroid"= 15,
              "Literature Records Centroid" = 22,
              "GBIF Records Centroid"= 23 )
  ) +
  # Fill color for centroids (no name to avoid duplication)

```

```

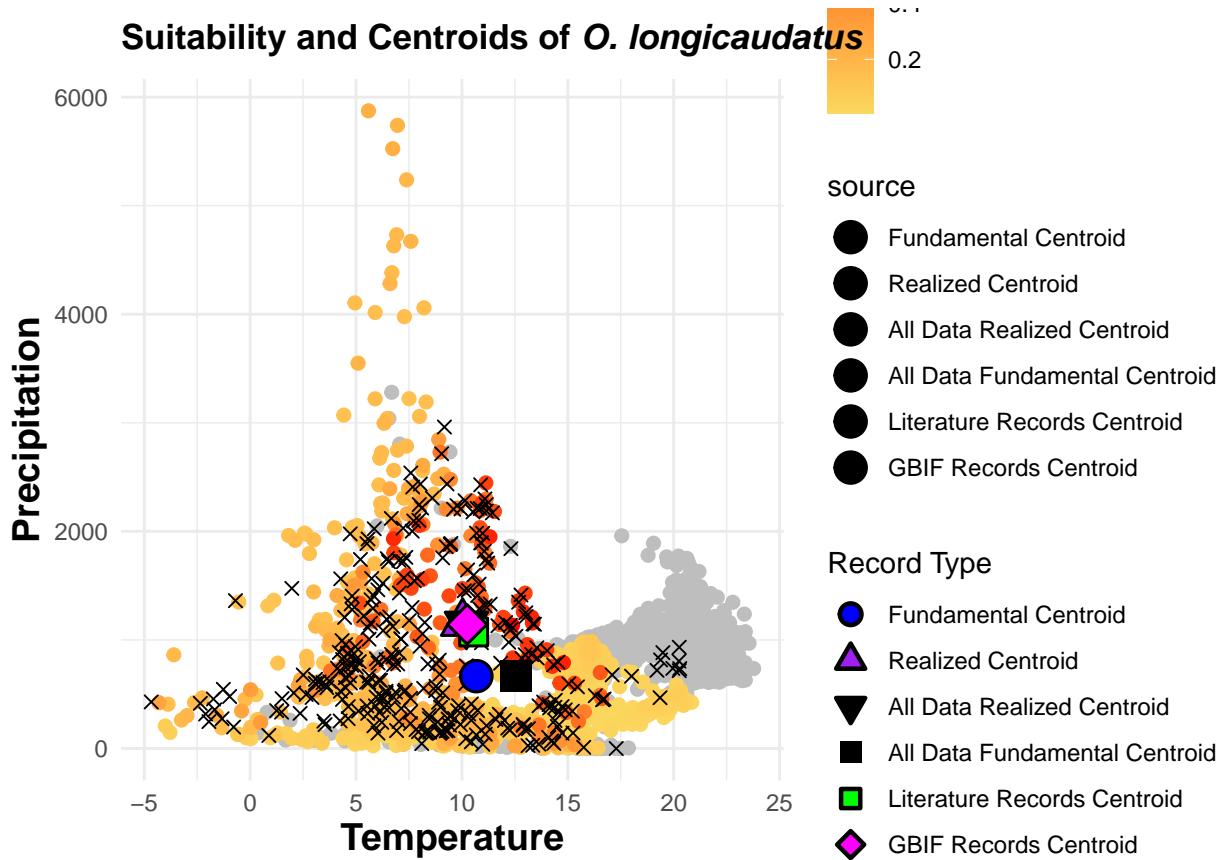
scale_fill_manual(
  values = c(
    "Fundamental Centroid" = "blue",
    "Realized Centroid" = "purple",
    "All Data Realized Centroid" = "black",
    "All Data Fundamental Centroid" = "lightblue",
    "Literature Records Centroid" = "green",
    "GBIF Records Centroid" = "magenta"
  )
) +
guides(
  shape = guide_legend(override.aes = list(
    fill = c( "blue", "purple", "black", "lightblue" , "green", "magenta" , NA),
    color = "black",
    stroke = 1.2,
    size = 3
  )))
) +
theme_minimal() +
theme(axis.title.x = element_text(size=14, face="bold", colour = "black"),
  axis.title.y = element_text(size=14, face="bold", colour = "black"))+
labs(x = "Temperature", y ="Precipitation")+
ggtitle(expression(paste(bold("Suitability and Centroids of ")), bolditalic("O. longicaudatus"))))

cont2

## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 13 rows containing missing values or values outside the scale range
## (`geom_point()`).

```



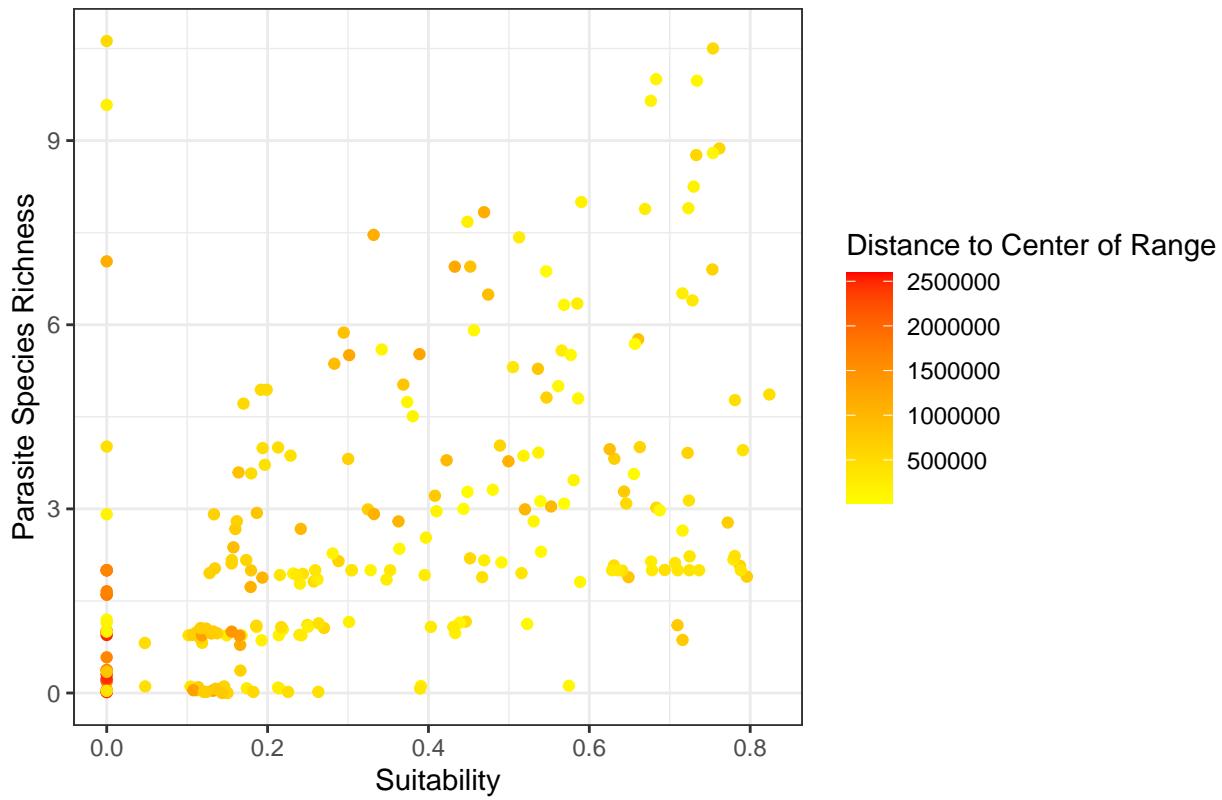
```
##Part 8. Final analyses and plots plots
```

```
###
```

```
SR<-ggplot(occ_bio_suit, aes(x=suit, y=SR, color=distcent)) + theme_bw() +
  labs(title="Predictors of Parasite Species Richness",
       x="Suitability", y = "Parasite Species Richness") +
  scale_color_gradient2(low="blue",mid="yellow", high="red",name="Distance to Center of Range") +
  SR
```

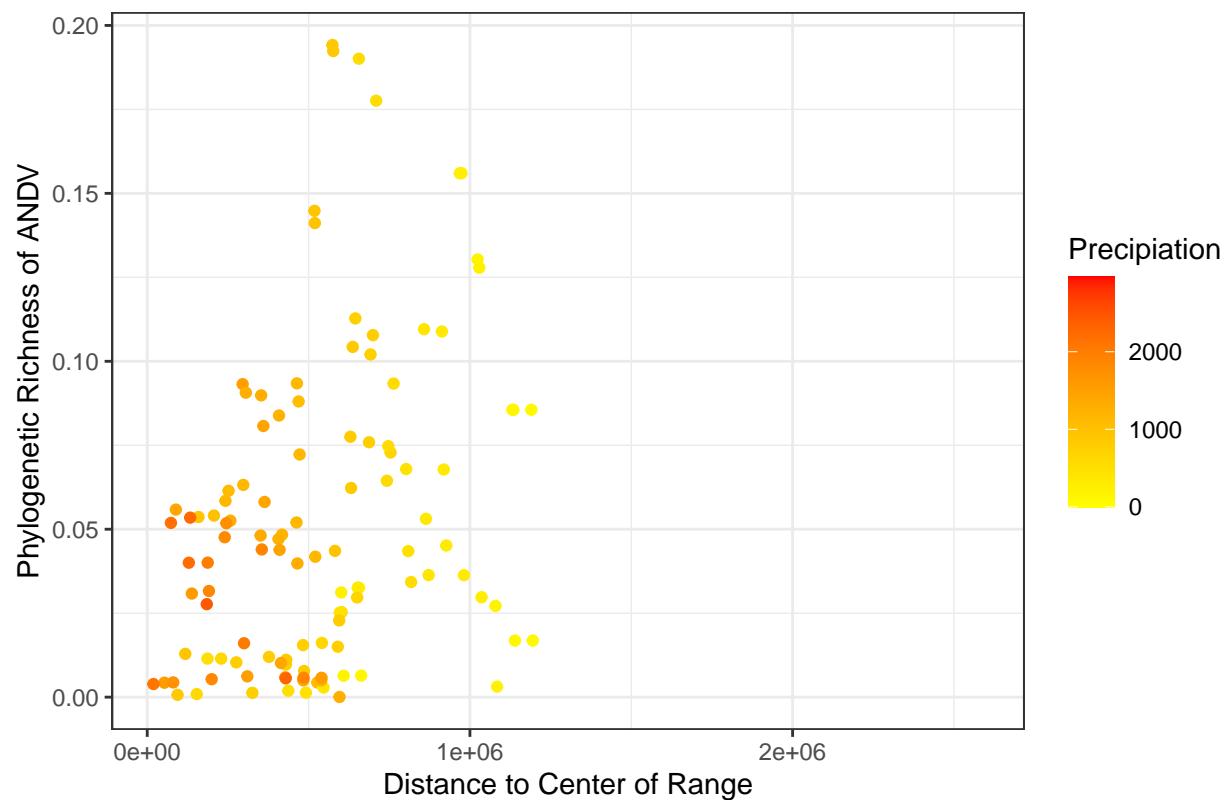
```
## Warning: Removed 51 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Predictors of Parasite Species Richness



```
ANDV<- ggplot(occ_bio_suit, aes(x=distcent, y=PDandv, color=Prec )) + theme_bw()+
  labs(title="Predictors of ANDV Phylogenetic Richness",
       x="Distance to Center of Range", y = "Phylogenetic Richness of ANDV") +
  scale_color_gradient2(low="blue",mid="yellow", high="red",name="Precipitation") + geom_point()
ANDV
## Warning: Removed 179 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

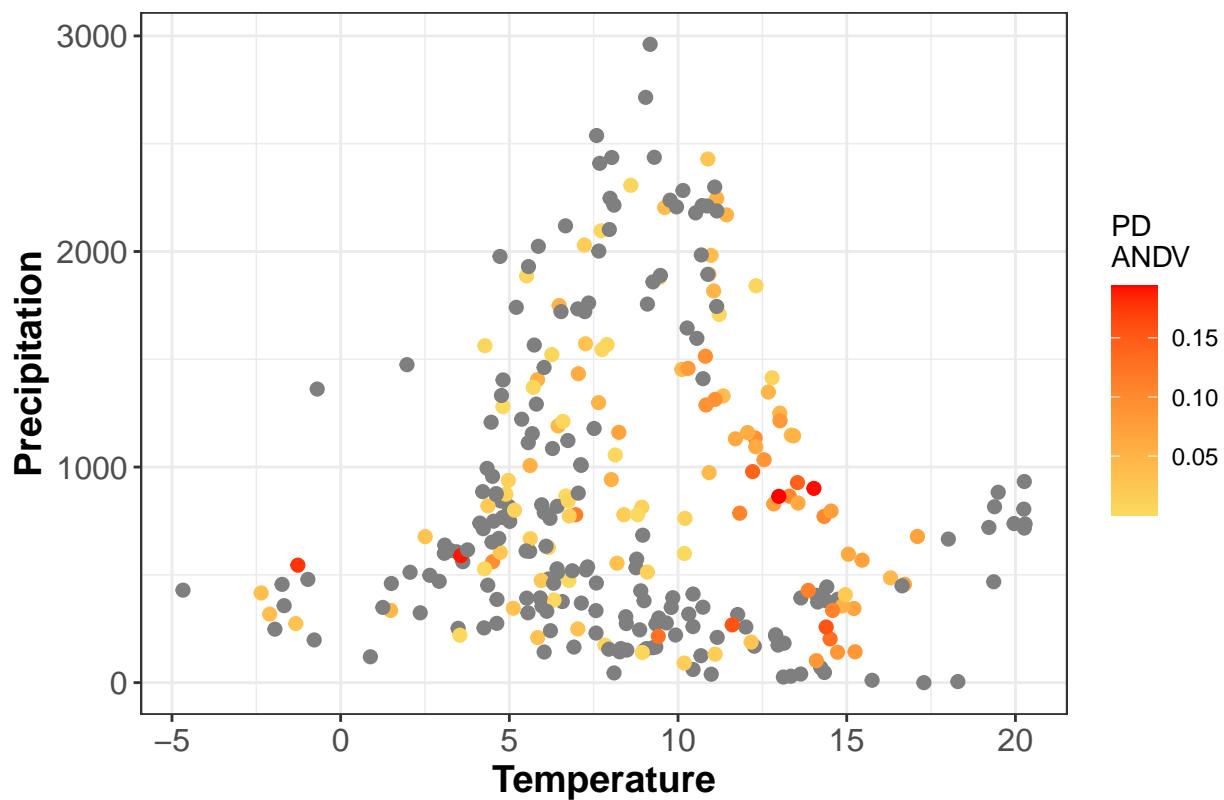
Predictors of ANDV Phylogenetic Richness



```
#####
envANDV<- ggplot(occ_bio_par, aes(x=Temp, y=Prec, color=PDandv)) + theme_bw()+
  labs(title="Phylogenetic Diversity of ANDV in Environmental Space",
       x="Temperature", y = "Precipitation", size=100) + scale_color_gradient2(mid="#fada5fff", high="",
      axis.title=element_text(size=14,face="bold"), plot.title = element_text(size=13, face="bold"))
envANDV

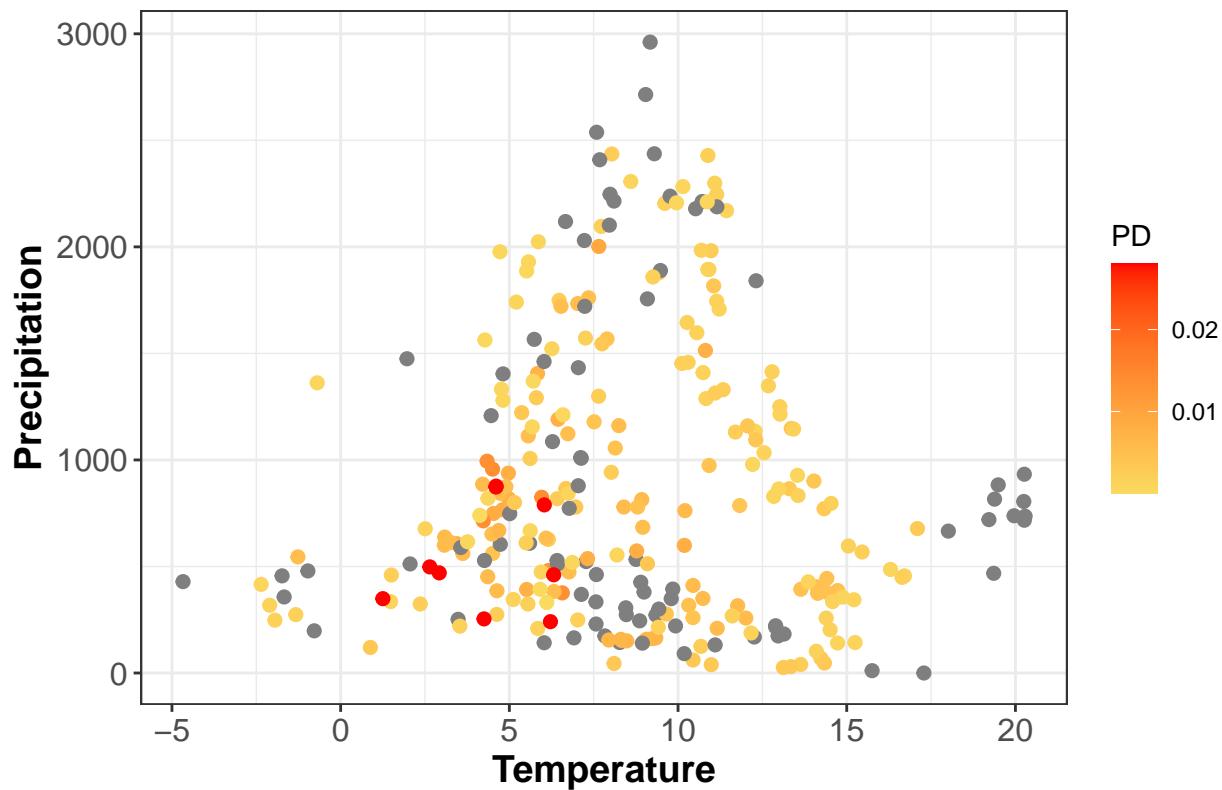
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Phylogenetic Diversity of ANDV in Environmental Space

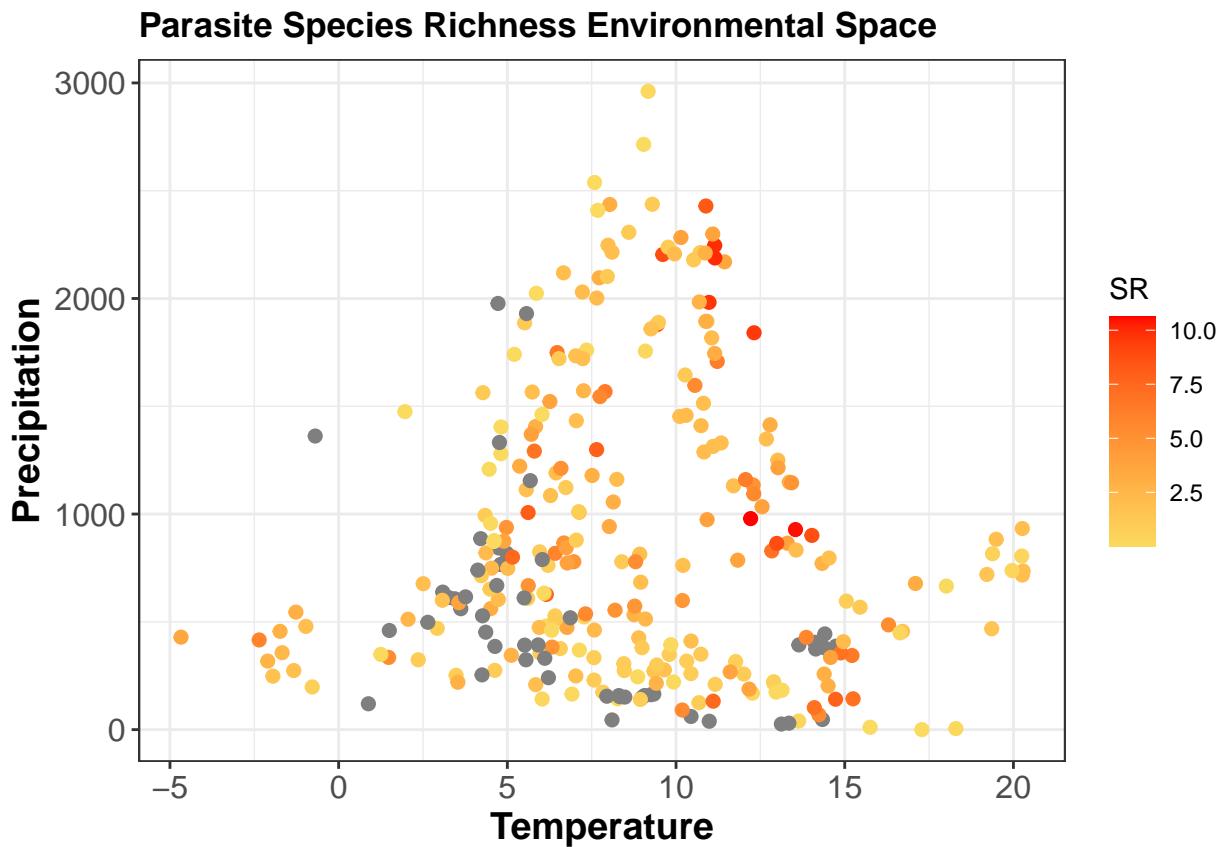


```
envColi<- ggplot(occ_bio_suit, aes(x=Temp, y=Prec, color=PD)) + theme_bw()+
  labs(x="Temperature", y = "Precipitation", size=100) +
  ggttitle(expression(paste(bold("Phylogenetic Diversity of "), bolditalic("O. longicaudatus"))))+
  scale_color_gradient2(mid="#fada5fff", high="red",name="PD") + geom_point(size=2) + theme(axis.text=element_text(size=14,face="bold"), plot.title = element_text(size=13, face="bold"))
envColi
```

Phylogenetic Diversity of *O. longicaudatus*

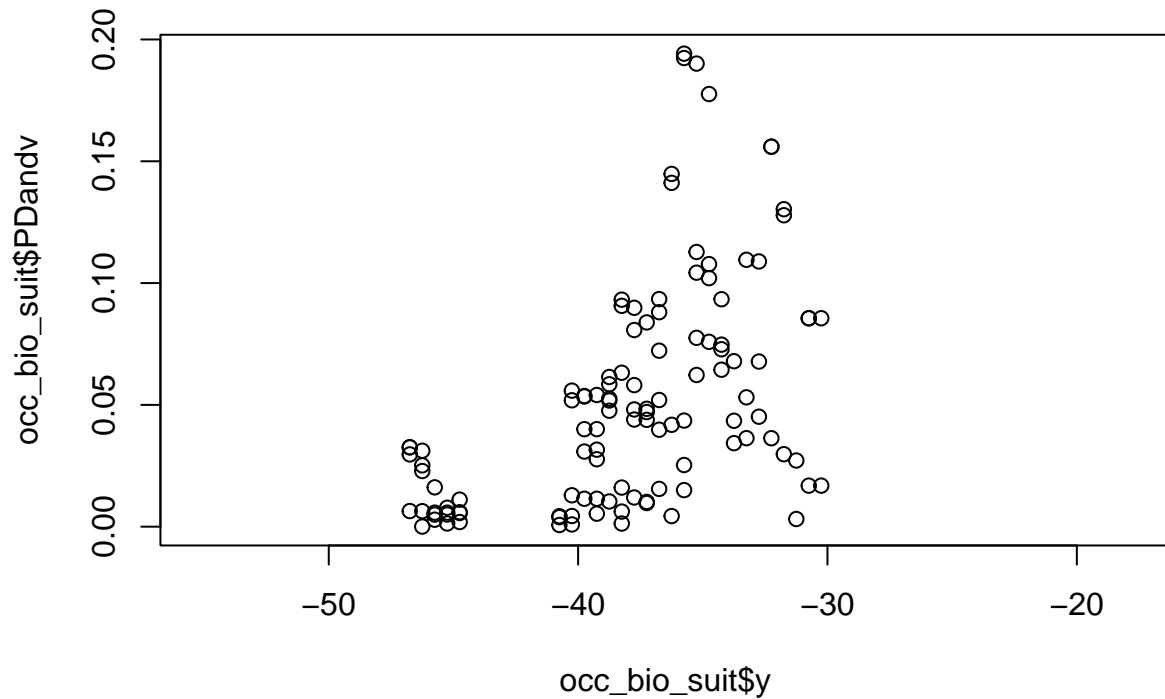


```
envSR<- ggplot(occ_bio_par, aes(x=Temp, y=Prec, color=SR)) + theme_bw()+
  labs(title="Parasite Species Richness Environmental Space",
       x="Temperature", y = "Precipitation", size=100) + scale_color_gradient2(mid="#fada5fff", high="#d95f36")
axis.title=element_text(size=14,face="bold"), plot.title = element_text(size=13, face="bold"))
envSR
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

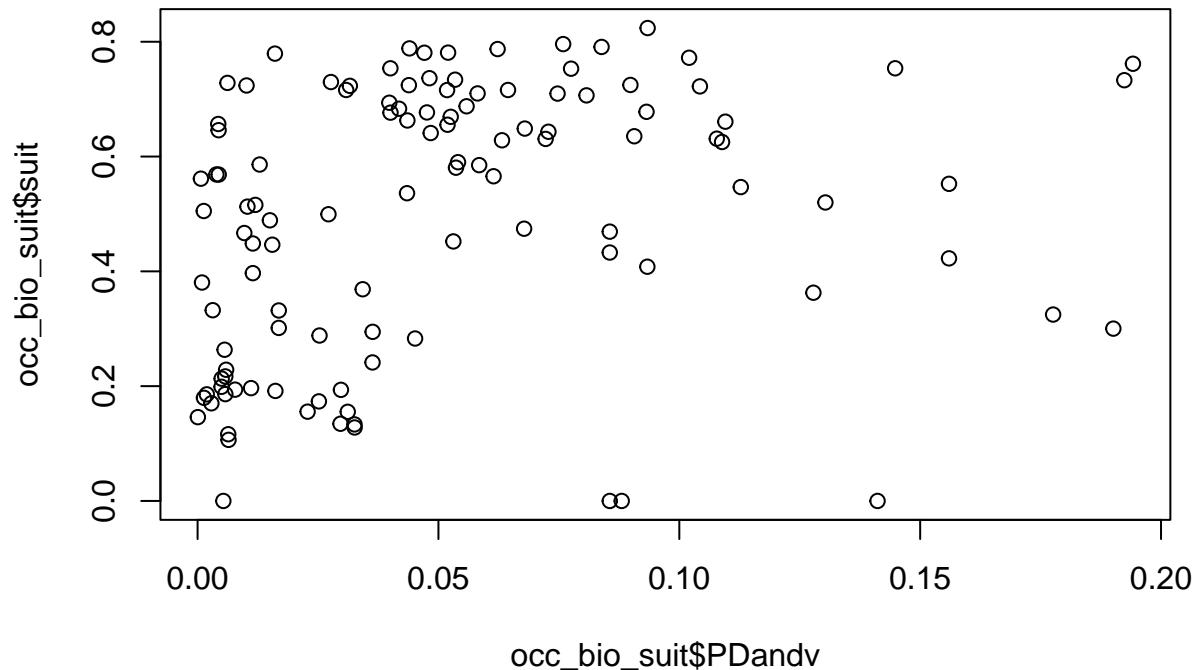


Correlation tests

```
plot(occ_bio_suit$y, occ_bio_suit$PDandv)
```



```
plot(occ_bio_suit$PDandv,occ_bio_suit$suit)
```



`occ_bio_suit$PDandv`

```
##check normality
```

```
shapiro.test(occ_bio_suit$Temp)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
```

```
## data: occ_bio_suit$Temp
```

```
## W = 0.9873, p-value = 0.01114
```

```
shapiro.test(occ_bio_suit$Prec)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
```

```
## data: occ_bio_suit$Prec
```

```
## W = 0.90842, p-value = 2.309e-12
```

```
shapiro.test(occ_bio_suit$PD)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
```

```
## data: occ_bio_suit$PD
```

```
## W = 0.59874, p-value < 2.2e-16
```

```
shapiro.test(occ_bio_suit$SR)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
```

```
## data: occ_bio_suit$SR
```

```
## W = 0.87914, p-value = 6.011e-13
```

```

shapiro.test(occ_bio_suit$PDandv)

##
##  Shapiro-Wilk normality test
##
## data: occ_bio_suit$PDandv
## W = 0.88336, p-value = 5.606e-08
shapiro.test(occ_bio_suit$distcent)

##
##  Shapiro-Wilk normality test
##
## data: occ_bio_suit$distcent
## W = 0.90255, p-value = 8.167e-13
shapiro.test(occ_bio_suit$suit)

##
##  Shapiro-Wilk normality test
##
## data: occ_bio_suit$suit
## W = 0.90729, p-value = 1.883e-12

##cor tests
cor.test(occ_bio_suit$PDandv,occ_bio_suit$PD, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PDandv, occ_bio_suit$PD, method =
## "spearman", : Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data: occ_bio_suit$PDandv and occ_bio_suit$PD
## S = 186644, p-value = 0.8026
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.02493142

cor.test(occ_bio_suit$PDandv,occ_bio_suit$suit, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PDandv, occ_bio_suit$suit, method =
## "spearman", : Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data: occ_bio_suit$PDandv and occ_bio_suit$suit
## S = 150212, p-value = 1.643e-05
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.3916183

cor.test(occ_bio_suit$PDandv,occ_bio_suit$Temp, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PDandv, occ_bio_suit$Temp, method =

```

```

## "spearman", : Cannot compute exact p-value with ties
##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PDandv and occ_bio_suit$Temp
## S = 120019, p-value = 4.981e-09
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.5139064

cor.test(occ_bio_suit$PDandv,occ_bio_suit$Prec, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PDandv, occ_bio_suit$Prec, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PDandv and occ_bio_suit$Prec
## S = 281887, p-value = 0.1327
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.141682

cor.test(occ_bio_suit$PDandv,occ_bio_suit$distcent, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PDandv, occ_bio_suit$distcent, :
## Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PDandv and occ_bio_suit$distcent
## S = 169663, p-value = 0.0007017
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.312842

cor.test(occ_bio_suit$PDandv,occ_bio_suit$dist_to_centroid, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PDandv, occ_bio_suit$dist_to_centroid,
## : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PDandv and occ_bio_suit$dist_to_centroid
## S = 278326, p-value = 0.1773
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.1272575

cor.test(occ_bio_suit$PDandv,occ_bio_suit$dist_to_realized, method = "spearman", use = "complete.obs")

```

```

## Warning in cor.test.default(occ_bio_suit$PDandv, occ_bio_suit$dist_to_realized,
## : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PDandv and occ_bio_suit$dist_to_realized
## S = 276528, p-value = 0.2035
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.1199793

cor.test(occ_bio_suit$PDandv,occ_bio_suit$SR, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PDandv, occ_bio_suit$SR, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PDandv and occ_bio_suit$SR
## S = 213659, p-value = 0.2398
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.1114707

##sr
cor.test(occ_bio_suit$SR,occ_bio_suit$PD, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$SR, occ_bio_suit$PD, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$SR and occ_bio_suit$PD
## S = 918611, p-value = 7.57e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.3456743

cor.test(occ_bio_suit$SR,occ_bio_suit$suit, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$SR, occ_bio_suit$suit, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$SR and occ_bio_suit$suit
## S = 973734, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.5877575

```

```

cor.test(occ_bio_suit$SR,occ_bio_suit$Temp, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$SR, occ_bio_suit$Temp, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$SR and occ_bio_suit$Temp
## S = 2010144, p-value = 0.02042
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## 0.1489799

cor.test(occ_bio_suit$SR,occ_bio_suit$Prec, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$SR, occ_bio_suit$Prec, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$SR and occ_bio_suit$Prec
## S = 1962664, p-value = 0.008397
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## 0.1690812

cor.test(occ_bio_suit$SR,occ_bio_suit$distcent, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$SR, occ_bio_suit$distcent, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$SR and occ_bio_suit$distcent
## S = 2988172, p-value = 2.95e-05
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.2650803

cor.test(occ_bio_suit$SR,occ_bio_suit$dist_to_centroid, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$SR, occ_bio_suit$dist_to_centroid, :
## Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$SR and occ_bio_suit$dist_to_centroid
## S = 2412499, p-value = 0.7409
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
##
```

```

## -0.02136191
cor.test(occ_bio_suit$SR,occ_bio_suit$dist_to_realized, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$SR, occ_bio_suit$dist_to_realized, :
## Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$SR and occ_bio_suit$dist_to_realized
## S = 2838693, p-value = 0.001602
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.2017966

## PD
cor.test(occ_bio_suit$PD,occ_bio_suit$PD, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PD, occ_bio_suit$PD, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PD and occ_bio_suit$PD
## S = 0, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 1

cor.test(occ_bio_suit$PD,occ_bio_suit$suit, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PD, occ_bio_suit$suit, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PD and occ_bio_suit$suit
## S = 1818331, p-value = 0.00971
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## -0.178084

cor.test(occ_bio_suit$PD,occ_bio_suit$Temp, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PD, occ_bio_suit$Temp, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PD and occ_bio_suit$Temp
## S = 1843689, p-value = 0.004669
## alternative hypothesis: true rho is not equal to 0

```

```

## sample estimates:
##          rho
## -0.1945129

cor.test(occ_bio_suit$PD,occ_bio_suit$Prec, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PD, occ_bio_suit$Prec, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PD and occ_bio_suit$Prec
## S = 1916468, p-value = 0.0004098
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.2416661

cor.test(occ_bio_suit$PD,occ_bio_suit$distcent, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PD, occ_bio_suit$distcent, method =
## "spearman", : Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PD and occ_bio_suit$distcent
## S = 1425298, p-value = 0.2694
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## 0.07655962

cor.test(occ_bio_suit$PD,occ_bio_suit$dist_to_centroid, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PD, occ_bio_suit$dist_to_centroid, :
## Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PD and occ_bio_suit$dist_to_centroid
## S = 2016673, p-value = 6.016e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.3065884

cor.test(occ_bio_suit$PD,occ_bio_suit$dist_to_realized, method = "spearman", use = "complete.obs")

## Warning in cor.test.default(occ_bio_suit$PD, occ_bio_suit$dist_to_realized, :
## Cannot compute exact p-value with ties

##
## Spearman's rank correlation rho
##
## data: occ_bio_suit$PD and occ_bio_suit$dist_to_realized
## S = 1611672, p-value = 0.5242

```

```

## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.04419092
##scaled multivariate##
df_scaled <- as.data.frame(scale(occ_bio_suit[, -c(1:2)]))

model_sr <- lm(SR ~ PD + suit + Temp+ Prec+ distcent + dist_to_centroid, data = df_scaled)
summary(model_sr)

##
## Call:
## lm(formula = SR ~ PD + suit + Temp + Prec + distcent + dist_to_centroid,
##      data = df_scaled)
##
## Residuals:
##       Min     1Q   Median     3Q    Max 
## -1.5683 -0.6648 -0.2160  0.5564  3.4624
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.047014  0.081876  0.574 0.566670    
## PD          -0.282629  0.086838 -3.255 0.001398 ** 
## suit         0.343700  0.098316  3.496 0.000618 *** 
## Temp        -0.003888  0.095480 -0.041 0.967570    
## Prec         -0.200601  0.186653 -1.075 0.284189    
## distcent    -0.158846  0.111976 -1.419 0.158057    
## dist_to_centroid 0.052131  0.154633  0.337 0.736481  
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8968 on 153 degrees of freedom
##   (133 observations deleted due to missingness)
## Multiple R-squared:  0.2401, Adjusted R-squared:  0.2103 
## F-statistic: 8.056 on 6 and 153 DF,  p-value: 1.438e-07

model_pdandv <- lm(PDandv ~ PD + suit + Temp+ Prec+ distcent +dist_to_centroid , data = df_scaled)
summary(model_pdandv)

##
## Call:
## lm(formula = PDandv ~ PD + suit + Temp + Prec + distcent + dist_to_centroid,
##      data = df_scaled)
##
## Residuals:
##       Min     1Q   Median     3Q    Max 
## -1.4046 -0.5112 -0.1733  0.3233  3.0621
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.05597   0.20984   0.267 0.79026    
## PD          0.04746   0.30416   0.156 0.87633    
## suit        0.30676   0.11810   2.597 0.01087 *  
## Temp        0.22529   0.10768   2.092 0.03906 * 

```

```

## Prec          0.17214   0.28022   0.614  0.54047
## distcent     0.89857   0.29893   3.006  0.00338 **
## dist_to_centroid -0.20023   0.18775  -1.066  0.28888
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8115 on 96 degrees of freedom
##   (190 observations deleted due to missingness)
## Multiple R-squared:  0.3497, Adjusted R-squared:  0.309
## F-statistic: 8.603 on 6 and 96 DF,  p-value: 1.732e-07
model_pd <- lm(PD ~ suit + Temp+ Prec+ distcent +SR +dist_to_centroid, data = df_scaled)
summary(model_pd)

##
## Call:
## lm(formula = PD ~ suit + Temp + Prec + distcent + SR + dist_to_centroid,
##      data = df_scaled)
##
## Residuals:
##    Min      1Q  Median      3Q      Max
## -1.2707 -0.3504 -0.1078  0.1844  3.6035
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.01184   0.07379  0.160  0.8728
## suit        -0.10697  0.09158 -1.168  0.2446
## Temp        -0.16720  0.08490 -1.970  0.0507 .
## Prec         0.24988  0.16747  1.492  0.1377
## distcent     0.02210  0.10146  0.218  0.8279
## SR           -0.22910  0.07039 -3.255  0.0014 **
## dist_to_centroid -0.37508  0.13593 -2.759  0.0065 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8075 on 153 degrees of freedom
##   (133 observations deleted due to missingness)
## Multiple R-squared:  0.2361, Adjusted R-squared:  0.2061
## F-statistic: 7.881 on 6 and 153 DF,  p-value: 2.081e-07

regression plots
SR <- ggplot(occ_bio_suit, aes(x = suit, y = SR, color = distcent)) +
  theme_bw() +
  # Scatter points
  geom_point() +
  # Pearson correlation line + confidence interval
  geom_smooth(method = "lm", se = TRUE, color = "black") +
  # Axis limits based on data max
  scale_x_continuous(limits = c(0, max(occ_bio_suit$suit, na.rm = TRUE))) +
  scale_y_continuous(limits = c(0, max(occ_bio_suit$SR, na.rm = TRUE))) +
  # Color gradient for distance to center

```

```

scale_color_gradient2(
  low = "blue", mid = "yellow", high = "red",
  name = "Distance to \nCenter of Range"
) + 

# Axis labels and plot title
labs(
  title = "Predictors of Parasite Species Richness",
  x = "Suitability",
  y = "Parasite Species Richness"
) + 

# Bold axis and title text
theme(
  plot.title = element_text(face = "bold", size = 14, hjust = 0.5),
  axis.title = element_text(face = "bold", size = 12),
  axis.text = element_text(face = "bold", size = 10)
)
SR

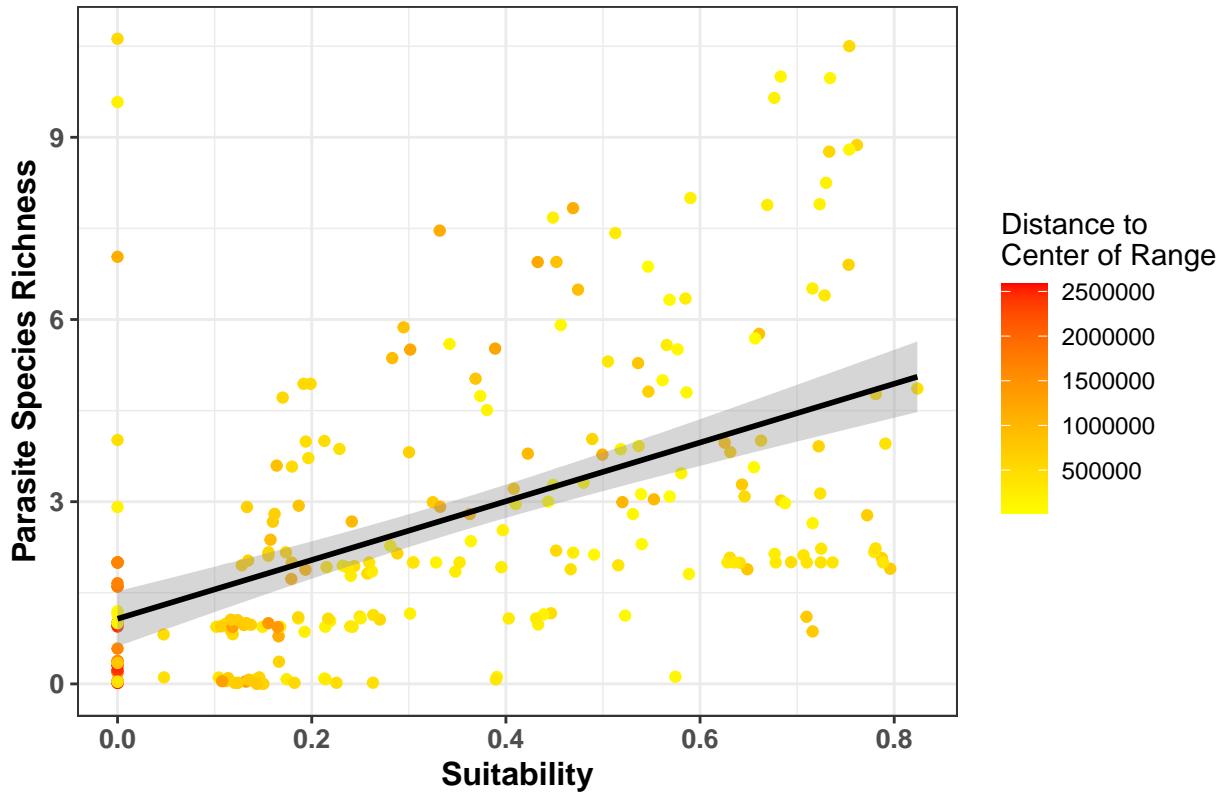
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 51 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 51 rows containing missing values or values outside the scale range
## (`geom_point()`).

```

Predictors of Parasite Species Richness



```

#~~~~~
ANDV<-
ggplot(occ_bio_suit, aes(x = distcent, y = PDandv, color = Prec)) +
  theme_bw() +
  # Scatter points
  geom_point() +
  # Pearson correlation line + confidence interval
  geom_smooth(method = "lm", se = TRUE, color = "black") +
  # Axis limits based on data max
  scale_x_continuous(limits = c(0, 1250000)) +
  scale_y_continuous(limits = c(0, max(occ_bio_suit$PDandv, na.rm = TRUE))) +
  # Color gradient for distance to center
  scale_color_gradient2(
    low = "blue", mid = "yellow", high = "red",
    name = "Precipitation"
  ) +
  # Axis labels and plot title
  labs(
    title = "Predictors of ANDV Phylogenetic Richness",
    x = "Distance to Range Centroid",
    y = "Phylogenetic Richness of ANDV"
  ) +
  # Bold axis and title text
  theme(
    plot.title = element_text(face = "bold", size = 14, hjust = 0.5),
    axis.title = element_text(face = "bold", size = 12),
    axis.text = element_text(face = "bold", size = 10)
  )
ANDV

## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 179 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 179 rows containing missing values or values outside the scale range
## (`geom_point()`).

```

Predictors of ANDV Phylogenetic Richness

