merge_datasets.R

brunn

2022-10-21

```
library(ggplot2)
require(data.table)
```

```
## Loading required package: data.table
```

```
library(dplyr)
```

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

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```
library(tidyr)
require(parallel)
```

```
## Loading required package: parallel
```

```
library(bdc)
library(taxadb)
library(traitdataform)
library(pbapply)


source("R/clean_taxa_functions.r")

##########################
# Load bishifts
bs_v1 <- fread('Data/biov1_fixednames.csv')
```

```r
bs_v1$spp <- gsub("_"," ",bs_v1$sp_name_std_v1)

bs_v2 <- fread('Data/biov2_fixednames.csv')
bs_v2$spp <- gsub("_"," ",bs_v2$sp_name_std_v2)

# Load genetic data
mt <- fread('Data/mtdna.csv')
ms <- fread('Data/msat.csv')
gen_d <- fread('Data/Deposited_data_genetic_diversity_dekort2021.csv')
gen_d$spp <-  gsub("_"," ",gen_d$Species)
gen_lf <- fread('Data/MacroPopGen_Database_final_areas_Lawrence_Fraser_2020.csv')
gen_lf$spp <-  gsub("_"," ",gen_lf$G_s)

gen_sps <- unique(c(mt$spp, ms$spp, gen_d$spp, gen_lf$spp))

bsi_v1 <- intersect(bs_v1$spp, gen_sps)
bsi_v2 <- intersect(bs_v2$spp, gen_sps)

bsi_tot <- union(bsi_v1,bsi_v2)

length(gen_sps) # 1897 spp with gen data before fixing sci names
```

```
## [1] 1897
```

```r
length(bsi_v1) # 317 species match with bioshifts v1 before fixing sci names
```

```
## [1] 317
```

```r
length(bsi_v2) # 327 species match with bioshifts v2 before fixing sci names
```

```
## [1] 327
```

```r
length(bsi_tot) # 366 species match before fixing sci names
```

```
## [1] 366
```

```r
# Break down

tot1 <- data.frame(N_species_with_gen_data = length(gen_sps),
                   v1_matches = length(bsi_v1),
                   v2_matches = length(bsi_v2),
                   v3_matches = length(bsi_tot))
tot1
```

```
##   N_species_with_gen_data v1_matches v2_matches v3_matches
## 1                    1897        317        327        366
```

```
##############################
# Fix names at the genetic data

# Species to find

mycols <- c("species","scientificName","kingdom","phylum","class","order","family","db_code")
splist <- data.frame(matrix(ncol = length(mycols), nrow = length(gen_sps)))
names(splist) <- mycols

splist$reported_name_fixed <- gen_sps
tofind_ <- splist[which(is.na(splist$scientificName)),]
tofind_ <- unique(tofind_$reported_name_fixed)

tofind <- data.frame(matrix(nrow = length(tofind_), ncol = 8))
names(tofind) = c("scientificName", "kingdom", "phylum", "class", "order", "family", "db", "db_code")

tofind <- data.frame(species = tofind_, tofind)

tofind <- tofind %>%
    mutate(across(everything(), as.character))



## GBIF
togo <- tofind[which(is.na(tofind$scientificName)),]

mycols <- c("speciesKey", "kingdom","phylum","class","order","family", "species")

# retrieve sp names
cl <- makeCluster(detectCores()-2)
clusterExport(cl, c("togo","standardize_taxa"))

gbif_names <- pblapply(togo$species, function(x){
    try(standardize_taxa(data.frame(verbatimScientificName = x),
                         fuzzy = FALSE,
                         silent = TRUE))
}, cl = cl)

stopCluster(cl)

rem <- sapply(gbif_names, class)
if(any(rem=="try-error")){
    rem <- which(rem=="try-error")
    gbif_names <- gbif_names[-rem]
}
gbif_names <- rbindlist(gbif_names)

if(any(is.na(gbif_names$scientificName))){
    gbif_names <- gbif_names[-which(is.na(gbif_names$scientificName)),]
}
gbif_names <- gbif_names[which(gbif_names$taxonRank=='species'),]

# remove duplicates
```

```r
if(any(duplicated(gbif_names$verbatimScientificName))){
    gbif_names <- gbif_names[-which(duplicated(gbif_names$verbatimScientificName)),]
}

any(!gbif_names$original_search %in% splist$species)
```

```
## [1] FALSE
```

```r
cat("--- Summary ---\n",
    "N taxa:",nrow(togo),"\n",
    "N taxa found:",nrow(gbif_names), "\n",
    "N taxa not found:", nrow(togo)-nrow(gbif_names))
```

```
## --- Summary ---
##  N taxa: 1897
##  N taxa found: 1855
##  N taxa not found: 42
```

```r
gbif_names <- gbif_names[,c("verbatimScientificName","scientificName","kingdom","phylum","class","order
names(gbif_names) <- c("species","scientificName","kingdom","phylum","class","order","family","db_code"
gbif_names$db <- "gbif"
gbif_names$db_code <- gsub("http://www.gbif.org/species/","",gbif_names$db_code)
gbif_names$db_code <- paste("GBIF:",gbif_names$db_code,sep = "")

# Feed
tofind <- tofind %>%
    rows_patch(gbif_names,
               by = "species")

gc()
```

```
##            used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  3506354 187.3    6419295 342.9  4368994 233.4
## Vcells 12192758  93.1   21003785 160.3 14462465 110.4
```

```r
## Add found species names to the splist


all(tofind$species %in% splist$reported_name_fixed)
```

```
## [1] TRUE
```

```r
if(any(is.na(tofind$scientificName))){
    found <- tofind[-which(is.na(tofind$scientificName)),]
} else {
    found = tofind
}


for(i in 1:length(found$species)){
```

```
    tofill <- unique(which(splist$reported_name_fixed == found$species[i]))
    splist$scientificName[tofill] <- found$scientificName[i]
    splist$kingdom[tofill] <- found$kingdom[i]
    splist$phylum[tofill] <-found$phylum[i]
    splist$class[tofill] <- found$class[i]
    splist$order[tofill] <- found$order[i]
    splist$family[tofill] <- found$family[i]
    splist$db[tofill] <- found$db[i]
    splist$db_code[tofill] <- found$db_code[i]
}

any(!splist$scientificName == splist$reported_name_fixed)
```

```
## [1] TRUE
```

```
splist$spp = splist$reported_name_fixed

### Fix names

any(!mt$spp %in% gen_sps)
```

```
## [1] FALSE
```

```
any(!mt$spp %in% splist$spp)
```

```
## [1] FALSE
```

```
any(!ms$spp %in% gen_sps)
```

```
## [1] FALSE
```

```
any(!ms$spp %in% splist$spp)
```

```
## [1] FALSE
```

```
any(!gen_d$spp %in% gen_sps)
```

```
## [1] FALSE
```

```
any(!gen_d$spp %in% splist$spp)
```

```
## [1] FALSE
```

```
any(!gen_lf$spp %in% gen_sps)
```

```
## [1] FALSE
```

```r
any(!gen_lf$spp %in% splist$spp)
```

```
## [1] FALSE
```

```r
for(i in 1:nrow(mt)){
    mt$spp_new[i] <- splist$scientificName[which(splist$spp == mt$spp[i])]
}
for(i in 1:nrow(ms)){
    ms$spp_new[i] <- splist$scientificName[which(splist$spp == ms$spp[i])]
}
for(i in 1:nrow(gen_d)){
    gen_d$spp_new[i] <- splist$scientificName[which(splist$spp == gen_d$spp[i])]
}
for(i in 1:nrow(gen_lf)){
    gen_lf$spp_new[i] <- splist$scientificName[which(splist$spp == gen_lf$spp[i])]
}

##############################

# merge data again

bsmt_v1 <- intersect(gsub("_"," ",bs_v1$sp_name_std_v1), mt$spp_new)
bsms_v1 <- intersect(gsub("_"," ",bs_v1$sp_name_std_v1), ms$spp_new)

bsmt_v2 <- intersect(gsub("_"," ",bs_v2$sp_name_std_v2), mt$spp_new)
bsms_v2 <- intersect(gsub("_"," ",bs_v2$sp_name_std_v2), ms$spp_new)

totspp_v1 <- union(bsmt_v1, bsms_v1)
totspp_v2 <- union(bsmt_v2, bsms_v2)

totspp <- union(totspp_v1, totspp_v2)

tot <- union(mt$spp_new, ms$spp_new)

length(tot) # 378 spp with gen data after fixing sci names
```

```
## [1] 378
```

```r
length(totspp_v1) # 74 species match after fixing sci names
```

```
## [1] 74
```

```r
length(totspp_v2) # 96 species match after fixing sci names
```

```
## [1] 96
```

```r
length(totspp) # 100 species match after fixing sci names
```

```
## [1] 100
```

```
gen_sps <- unique(c(mt$spp_new, ms$spp_new, gen_d$spp_new, gen_lf$spp_new))

bsi_v1 <- intersect(bs_v1$spp, gen_sps)
bsi_v2 <- intersect(bs_v2$spp, gen_sps)

bsi_tot <- union(bsi_v1,bsi_v2)

length(gen_sps) # 1821 spp with gen data after fixing sci names
```

```
## [1] 1821
```

```
length(bsi_v1) # 345 species match with bioshifts v1 after fixing sci names
```

```
## [1] 345
```

```
length(bsi_v2) # 351 species match with bioshifts v2 after fixing sci names
```

```
## [1] 351
```

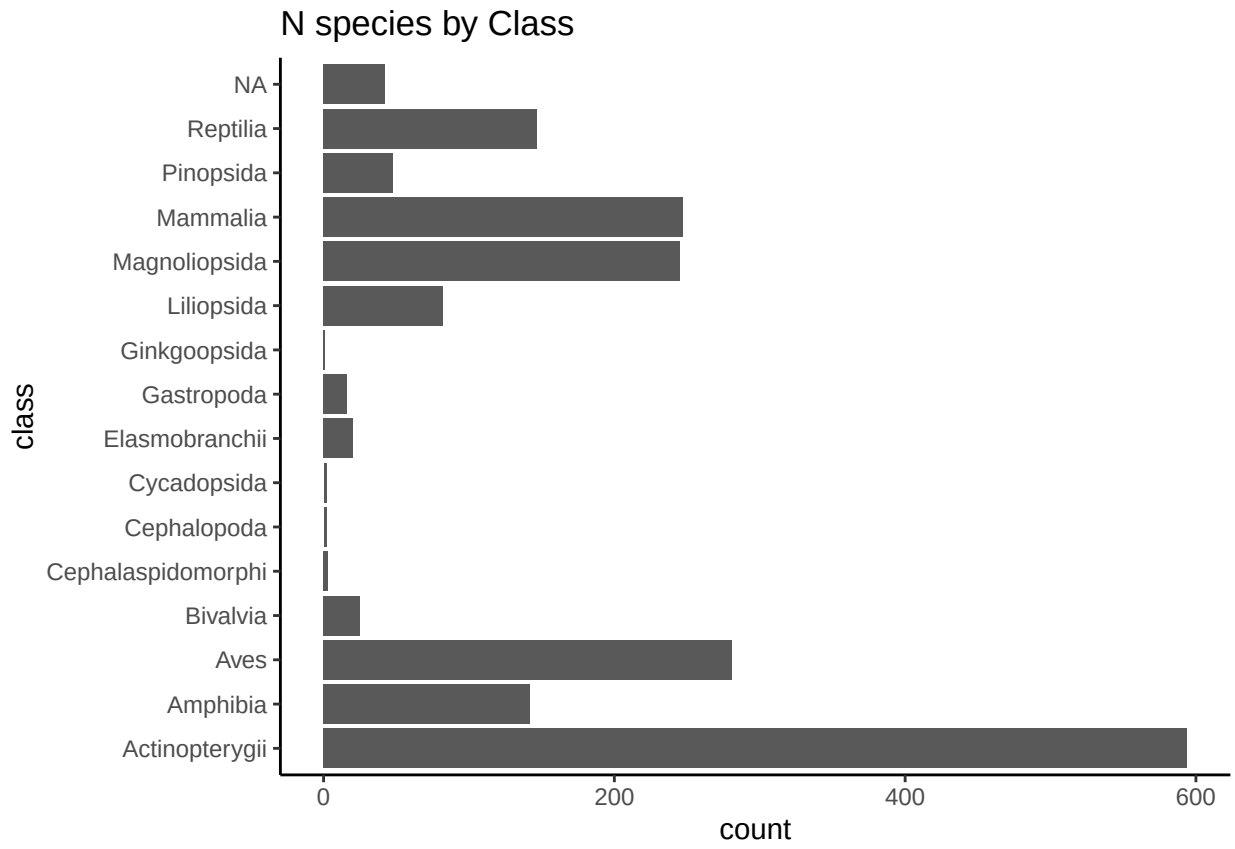
```
length(bsi_tot) # 395 species match after fixing sci names
```

```
## [1] 395
```

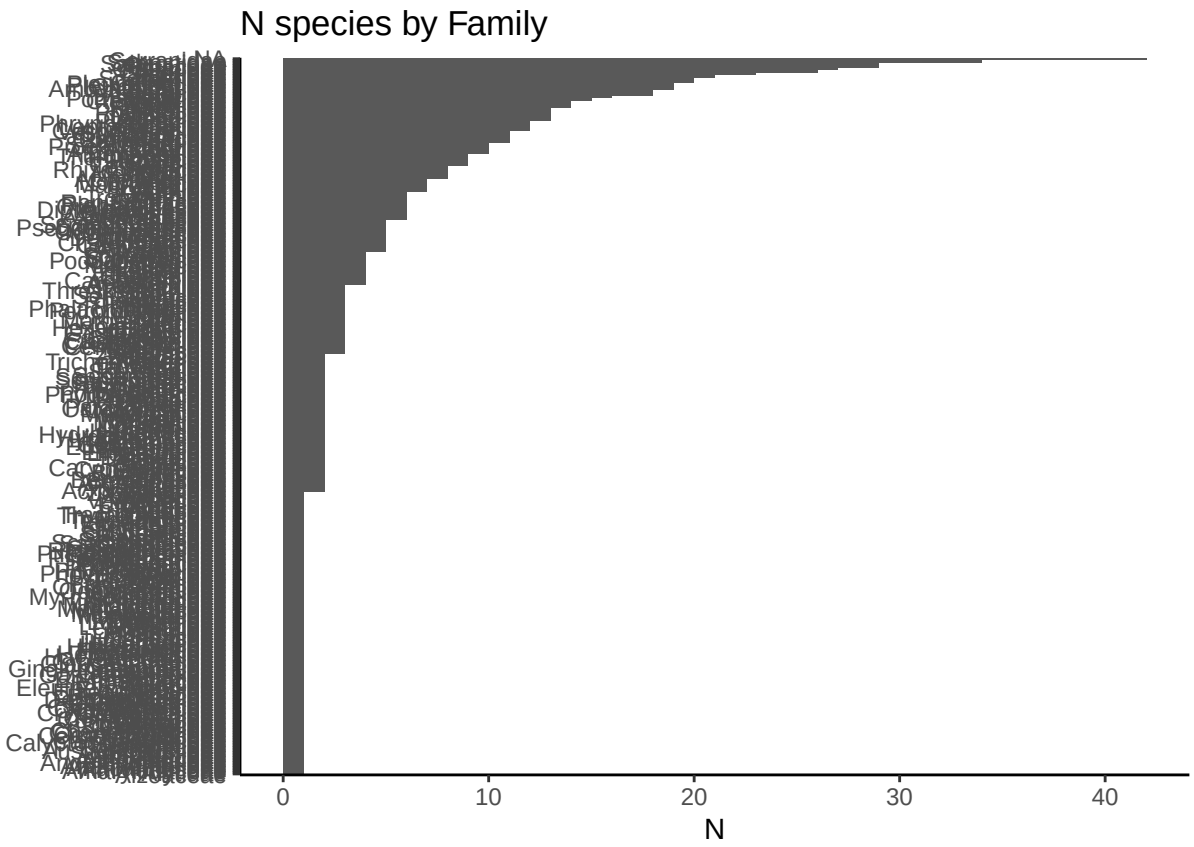
```
# Break down

tot2 <- data.frame(N_species_with_gen_data = length(gen_sps),
                   v1_matches = length(bsi_v1),
                   v2_matches = length(bsi_v2),
                   v3_matches = length(bsi_tot))

total <- data.frame(cbind(t(tot1),t(tot2)))
names(total) <- c("Before_tax_harmonization","After_tax_harmonization")
total
```

```
##                          Before_tax_harmonization After_tax_harmonization
## N_species_with_gen_data                     1897                     1821
## v1_matches                                   317                      345
## v2_matches                                   327                      351
## v3_matches                                   366                      395
```

```
# Which species?

ggplot(splist, aes(x=class))+
    ggtitle("N species by Class")+
    geom_bar()+
    theme_classic()+
    coord_flip()
```




```
toplot <- splist %>%
    group_by(family) %>%
    summarise(N = length(family))
level_order <- toplot$family[order(toplot$N)]

ggplot(toplot, aes(x=factor(family, level = level_order), y = N))+
    ggtitle("N species by Family")+
    geom_col()+
    theme_classic()+
    coord_flip()+
    xlab("")
```

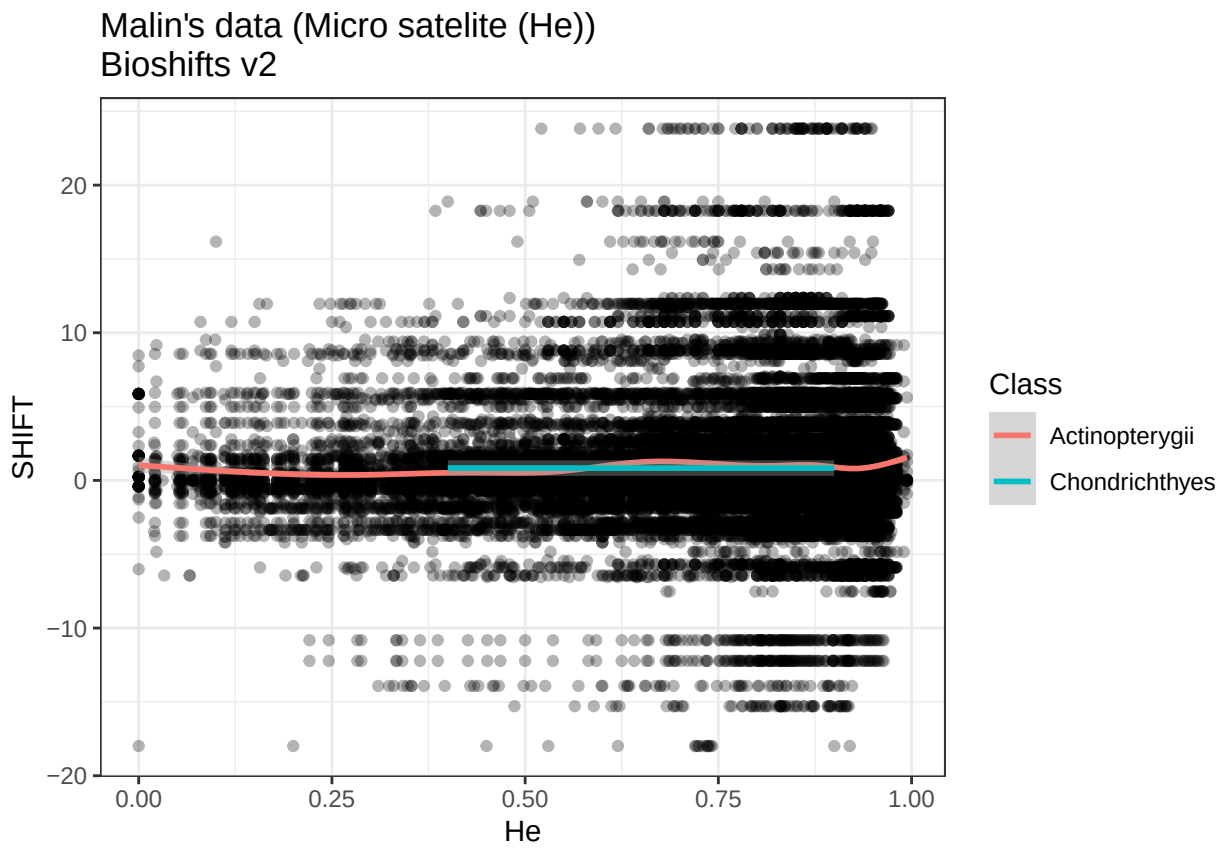



```
####################
# Explore

bs_v2$SHIFT <- as.numeric(bs_v2$`Conversion to km/dec`)
```

```
## Warning: NAs introduced by coercion
```

```
bs_v1$sp_name_std_v1 <- gsub("_"," ",bs_v1$sp_name_std_v1)
bs_v2$sp_name_std_v2 <- gsub("_"," ",bs_v2$sp_name_std_v2)

# Malin's data
bs_v1_ms <- merge(bs_v1, ms, by.x = "sp_name_std_v1", by.y = "spp_new")
bs_v1_mt <- merge(bs_v1, mt, by.x = "sp_name_std_v1", by.y = "spp_new")

bs_v2_ms <- merge(bs_v2, ms, by.x = "sp_name_std_v2", by.y = "spp_new", allow.cartesian=TRUE)
bs_v2_mt <- merge(bs_v2, mt, by.x = "sp_name_std_v2", by.y = "spp_new")

ggplot(bs_v1_ms, aes(x = He, y = SHIFT))+
    ggtitle("Malin's data (Micro satelite (He))\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = Class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

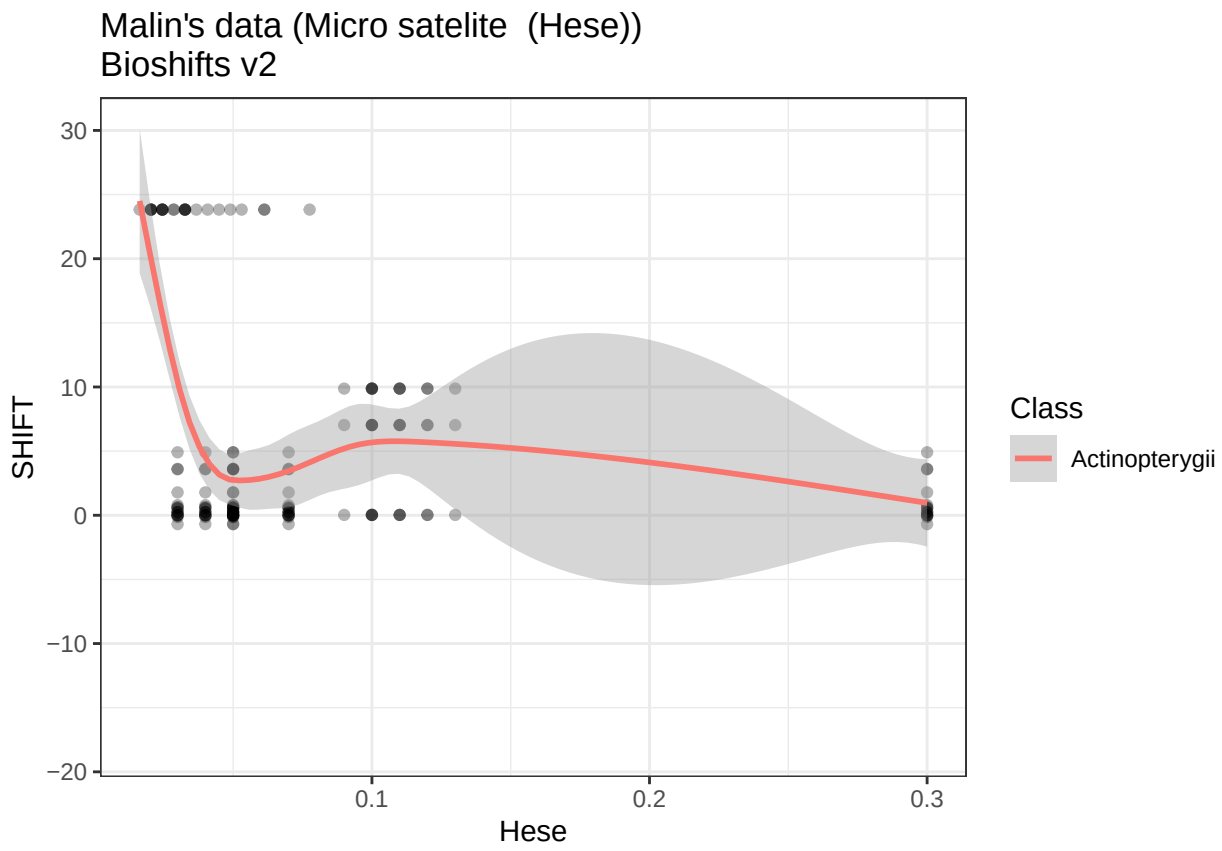



```
ggplot(bs_v1_ms, aes(x = Hese, y = SHIFT))+
    ggtitle("Malin's data (Micro satelite  (Hese))\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = Class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 38601 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 38601 rows containing missing values (geom_point).
```

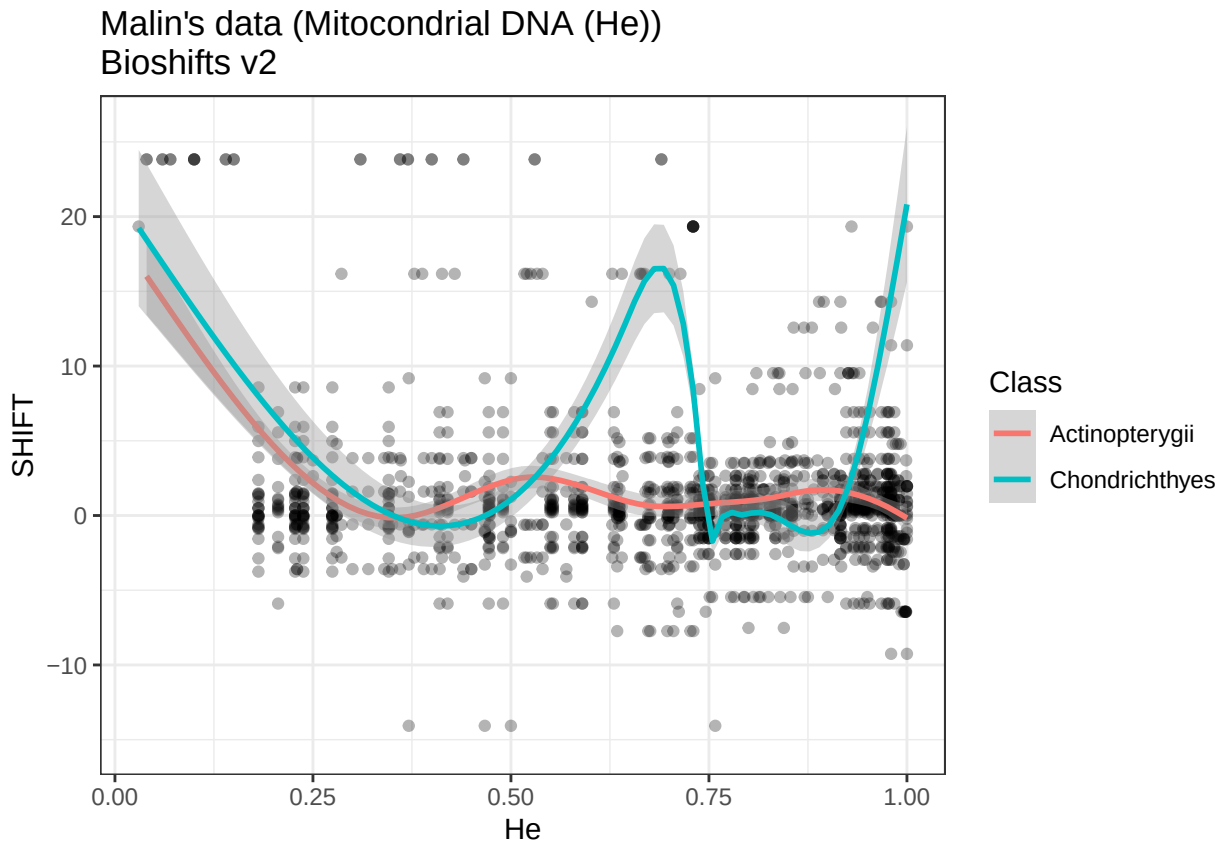



```
ggplot(bs_v1_mt, aes(x = He, y = SHIFT))+
    ggtitle("Malin's data (Mitocondrial DNA (He))\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = Class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 48 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 48 rows containing missing values (geom_point).
```

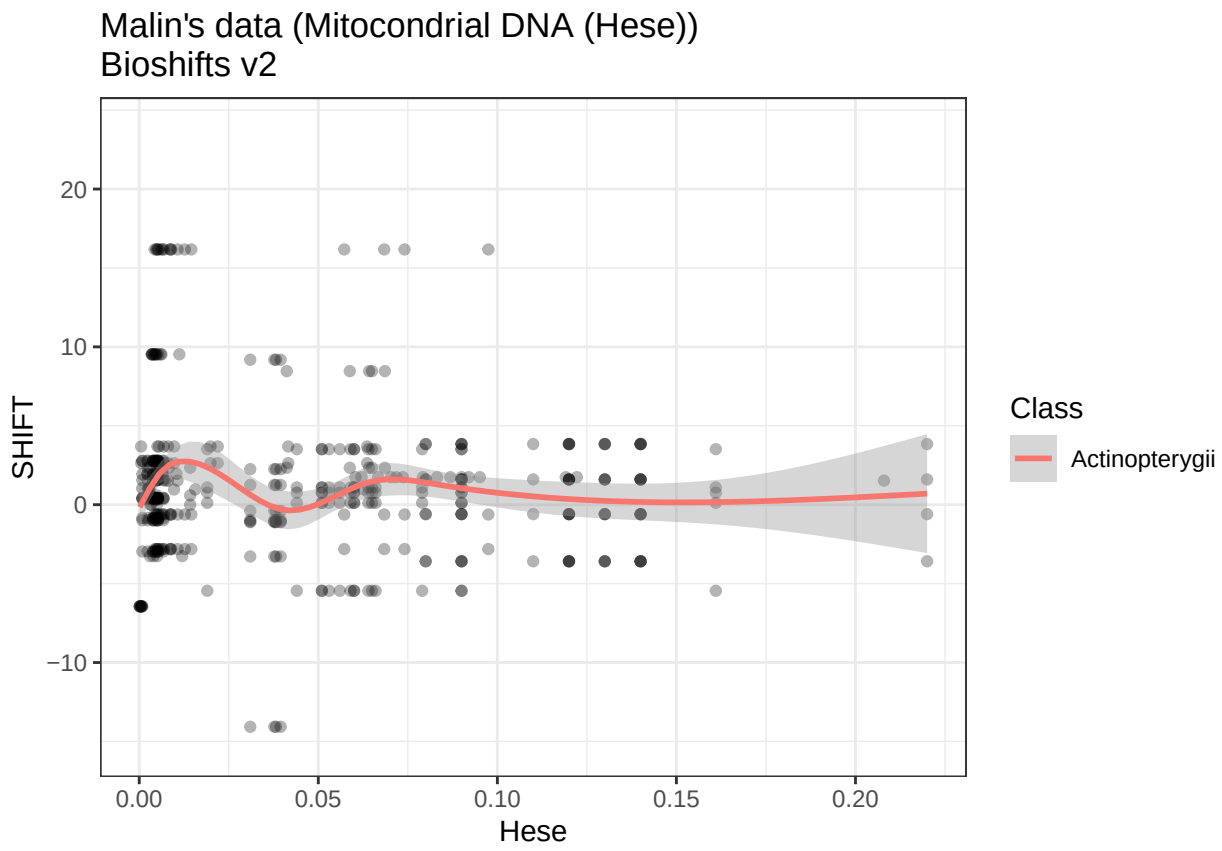



```
ggplot(bs_v1_mt, aes(x = Hese, y = SHIFT))+
    ggtitle("Malin's data (Mitocondrial DNA (Hese))\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = Class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 1128 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1128 rows containing missing values (geom_point).
```

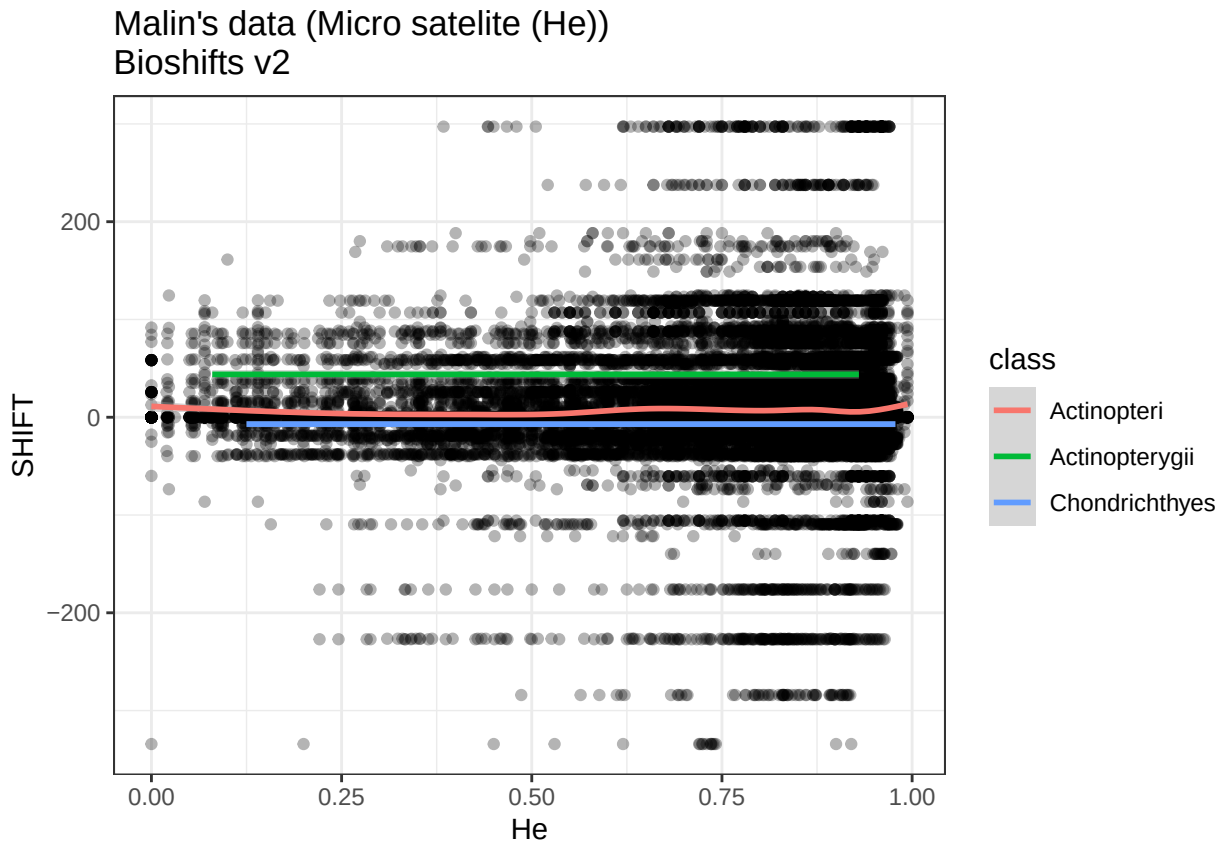



```
ggplot(bs_v2_ms, aes(x = He, y = SHIFT))+
    ggtitle("Malin's data (Micro satelite (He))\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 12623 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 12623 rows containing missing values (geom_point).
```

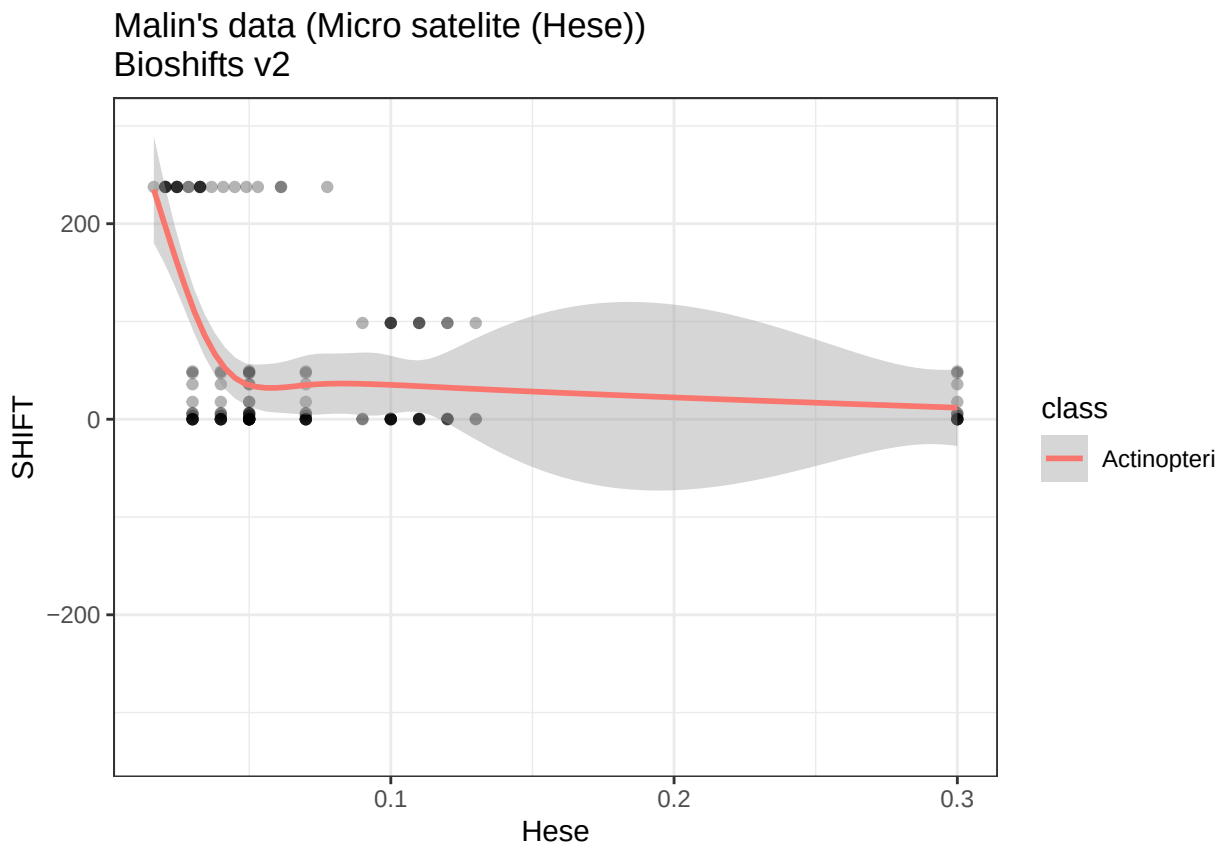



```
ggplot(bs_v2_ms, aes(x = Hese, y = SHIFT))+
    ggtitle("Malin's data (Micro satelite (Hese))\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 51737 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 51737 rows containing missing values (geom_point).
```

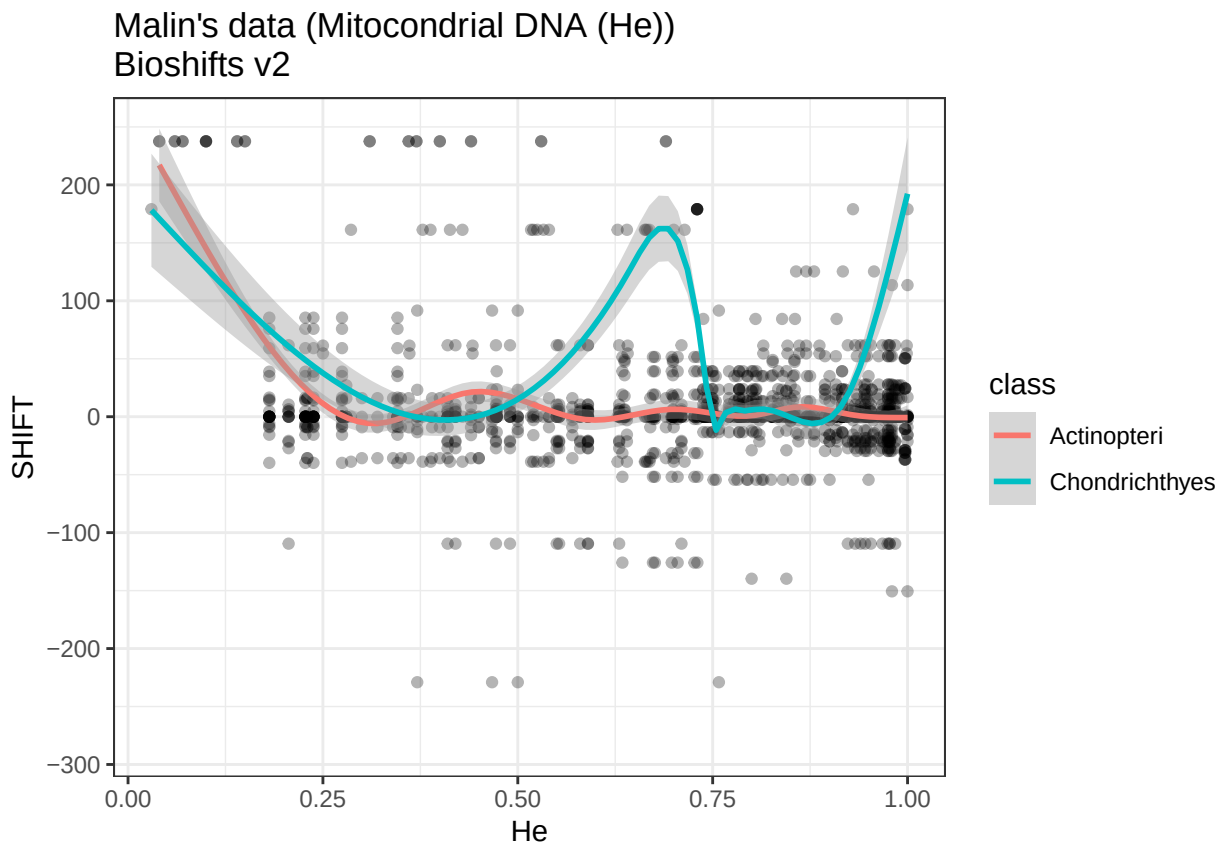



```
ggplot(bs_v2_mt, aes(x = He, y = SHIFT))+
    ggtitle("Malin's data (Mitocondrial DNA (He))\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 853 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 853 rows containing missing values (geom_point).
```

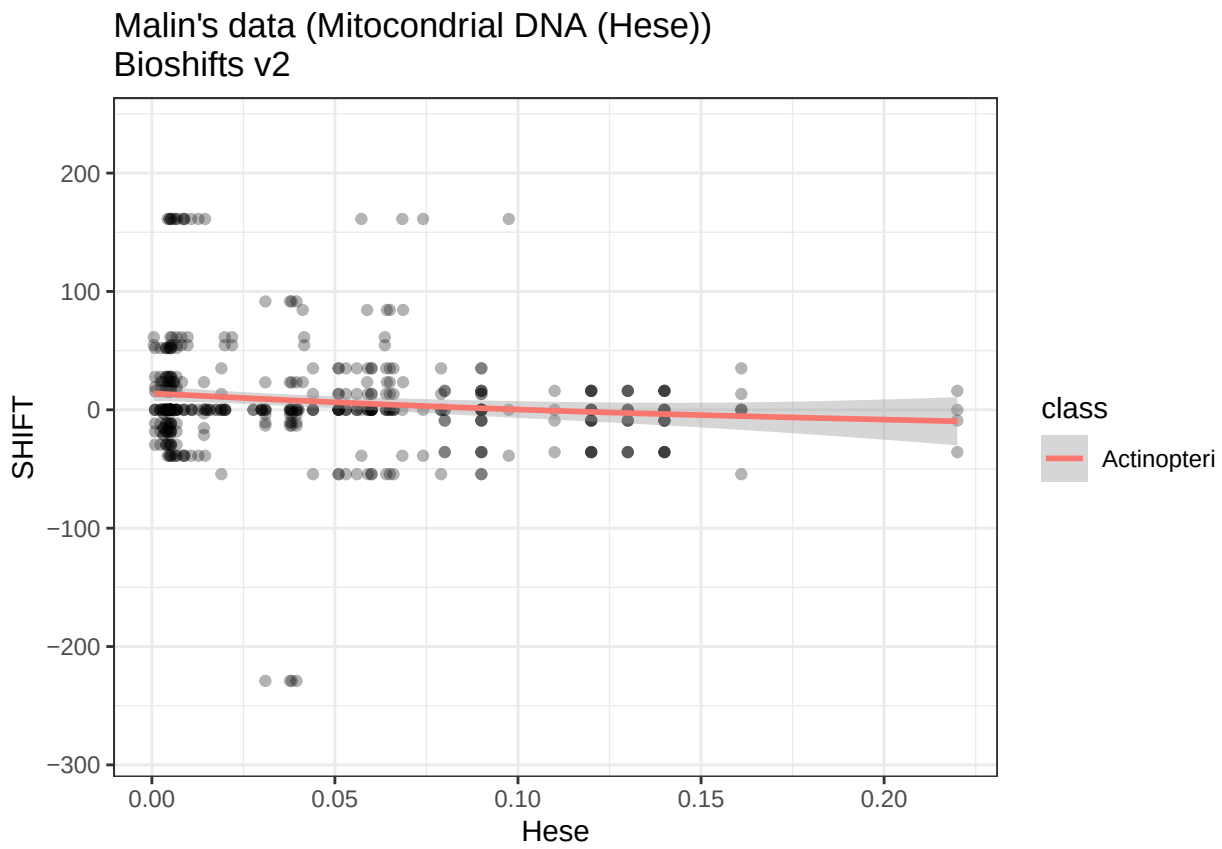



```
ggplot(bs_v2_mt, aes(x = Hese, y = SHIFT))+
    ggtitle("Malin's data (Mitocondrial DNA (Hese))\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

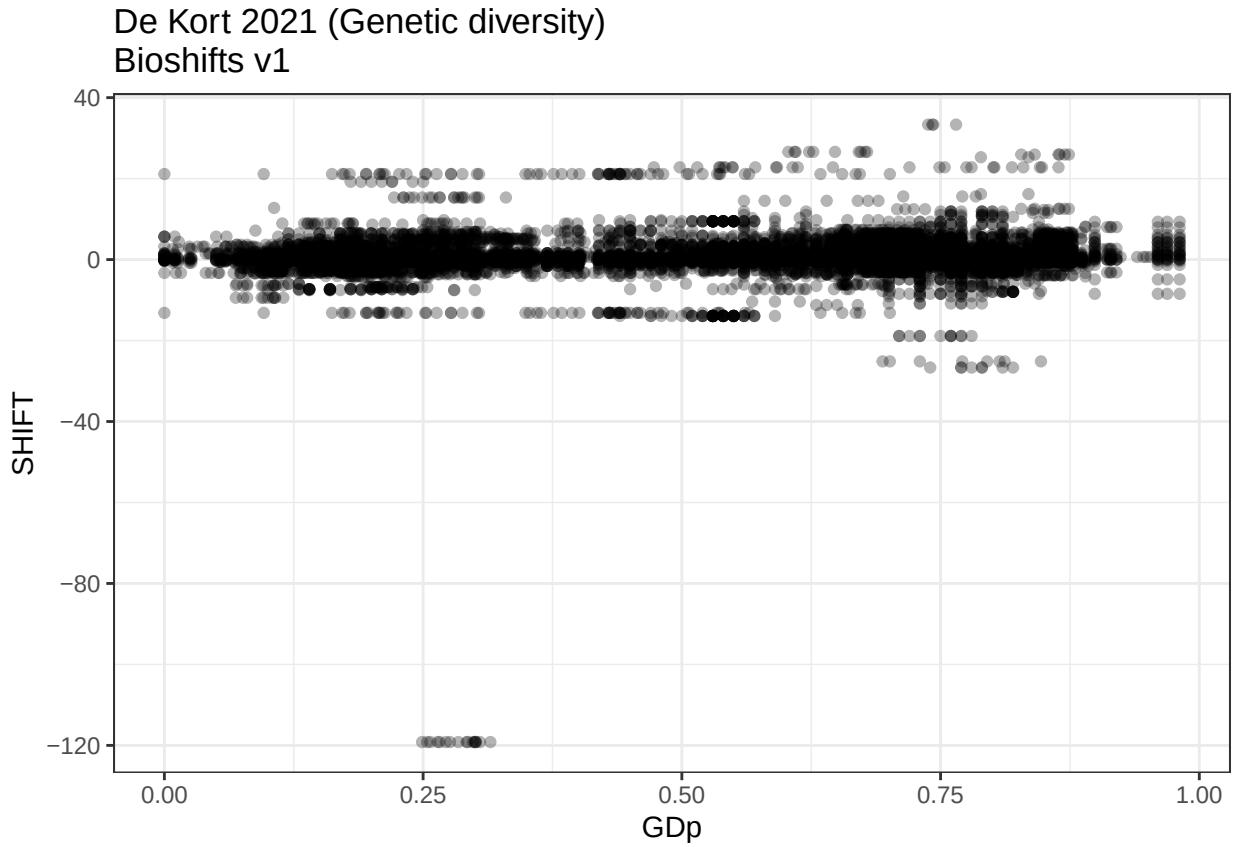
```
## Warning: Removed 1851 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1851 rows containing missing values (geom_point).
```




```
# De Kort 2021 data
bs_v1_d <- merge(bs_v1, gen_d, by.x = "sp_name_std_v1", by.y = "spp_new")
bs_v2_d <- merge(bs_v2, gen_d, by.x = "sp_name_std_v2", by.y = "spp_new", allow.cartesian=TRUE)

ggplot(bs_v1_d, aes(x = GDp, y = SHIFT))+
    ggtitle("De Kort 2021 (Genetic diversity)\nBioshifts v1")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = Class.x), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'

## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 3)
```



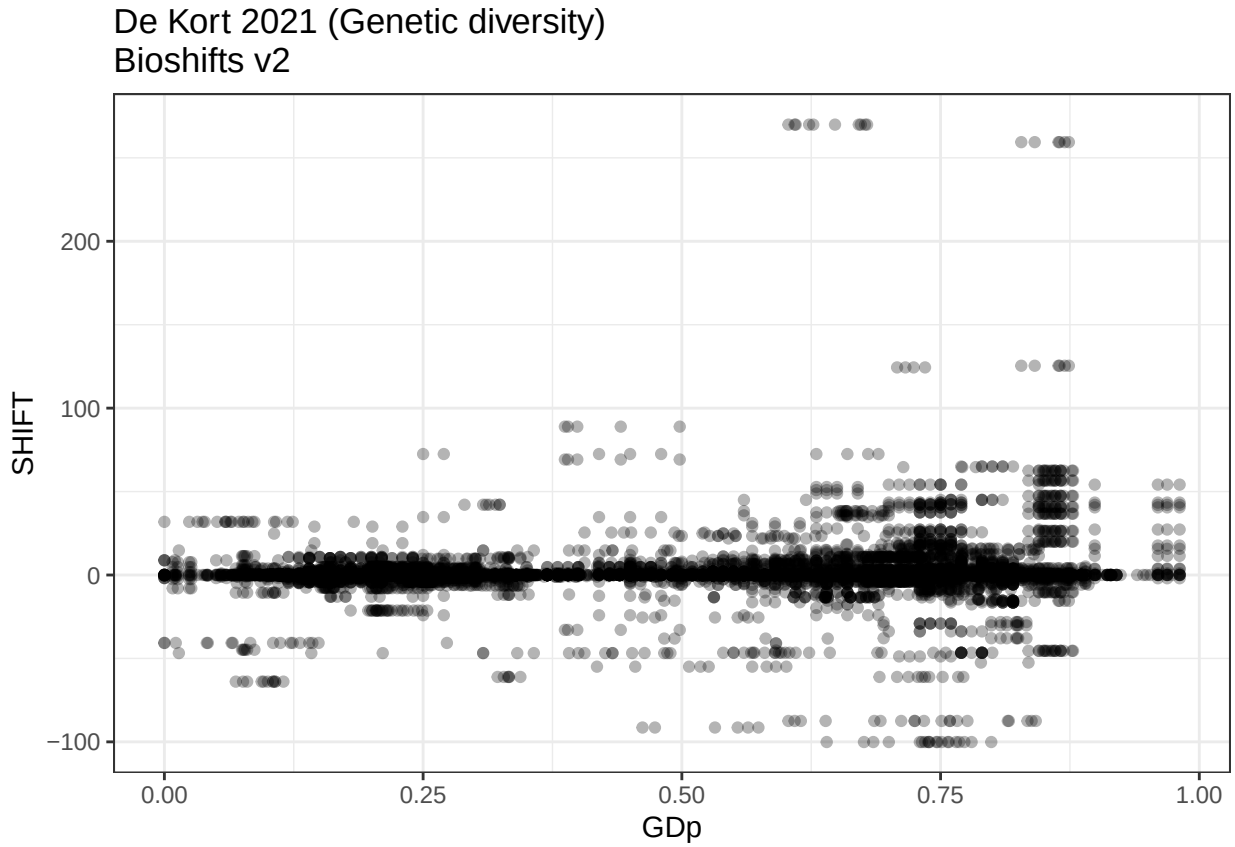

```
ggplot(bs_v2_d, aes(x = GDp, y = SHIFT))+
    ggtitle("De Kort 2021 (Genetic diversity)\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = Class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 1712 rows containing non-finite values (stat_smooth).
```

```
## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 3)
```

```
## Warning: Removed 1712 rows containing missing values (geom_point).
```




```
# De Lawrence & Fraser 2020 data
bs_v1_lf <- merge(bs_v1, gen_lf, by.x = "sp_name_std_v1", by.y = "spp_new")
bs_v1_lf$FST <- as.numeric(bs_v1_lf$FST)
```

```
## Warning: NAs introduced by coercion
```

```
bs_v2_lf <- merge(bs_v2, gen_lf, by.x = "sp_name_std_v2", by.y = "spp_new", allow.cartesian=TRUE)
bs_v2_lf$FST <- as.numeric(bs_v2_lf$FST)
```

```
## Warning: NAs introduced by coercion
```

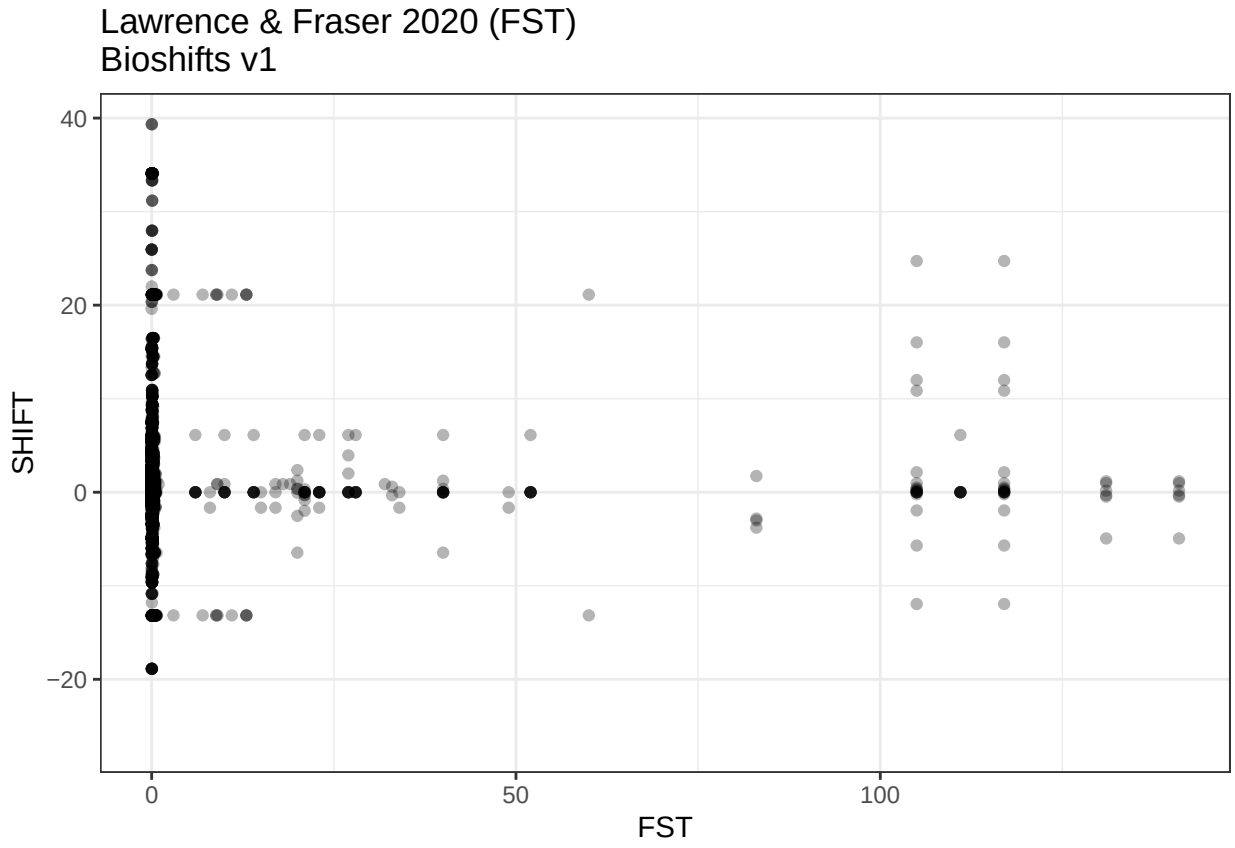
```
ggplot(bs_v1_lf, aes(x = FST, y = SHIFT))+
    ggtitle("Lawrence & Fraser 2020 (FST)\nBioshifts v1")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = Class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 1226 rows containing non-finite values (stat_smooth).
```

```
## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.
```

```
## Warning: Removed 1226 rows containing missing values (geom_point).
```

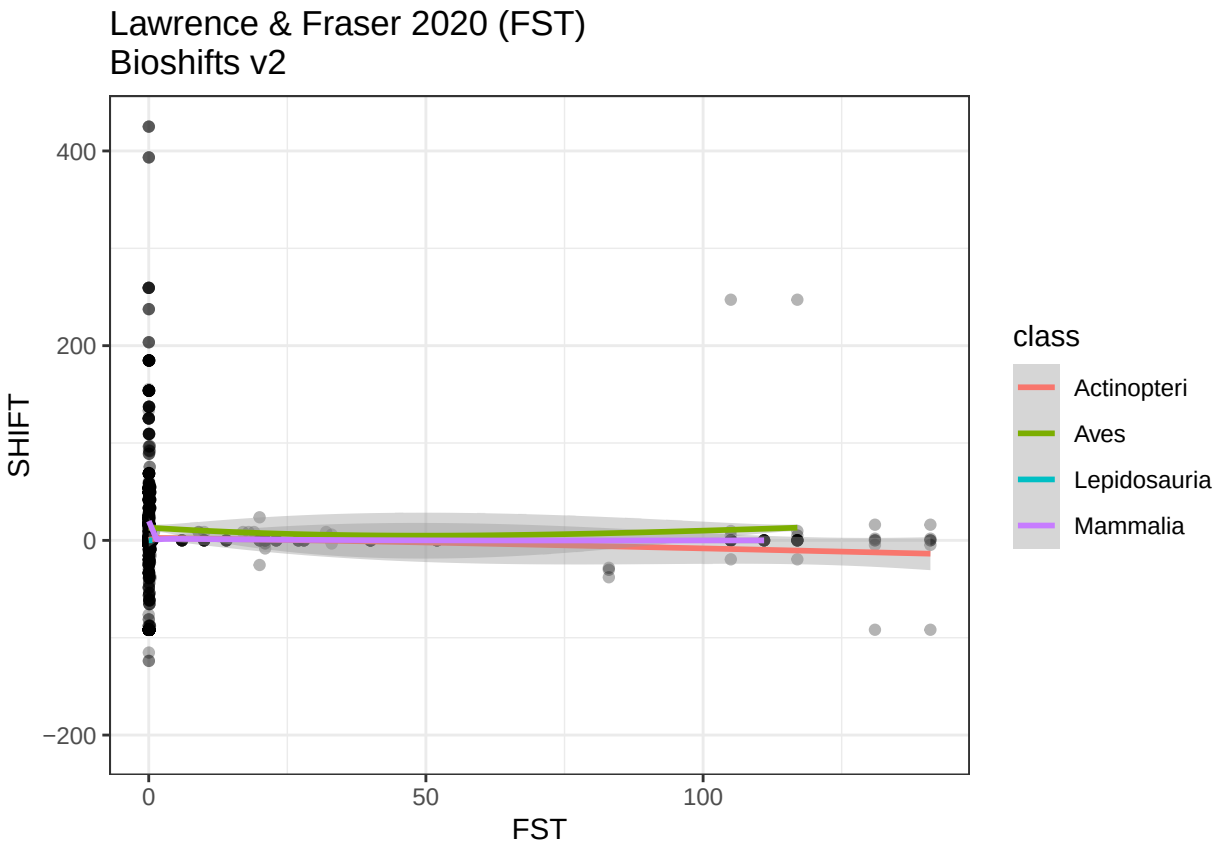



```
ggplot(bs_v2_lf, aes(x = FST, y = SHIFT))+
    ggtitle("Lawrence & Fraser 2020 (FST)\nBioshifts v2")+
    geom_point(alpha = .3)+
    theme_bw()+
    geom_smooth(aes(color = class), method = "gam")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 1409 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1409 rows containing missing values (geom_point).
```


Lawrence & Fraser 2020 (FST)
Bioshifts v2
SHIFT
400
200
0
−200
0
50
100
FST
class
Actinopteri
Aves
Lepidosauria
Mammalia