R codes for Yang et al. The global loss of floristic uniqueness

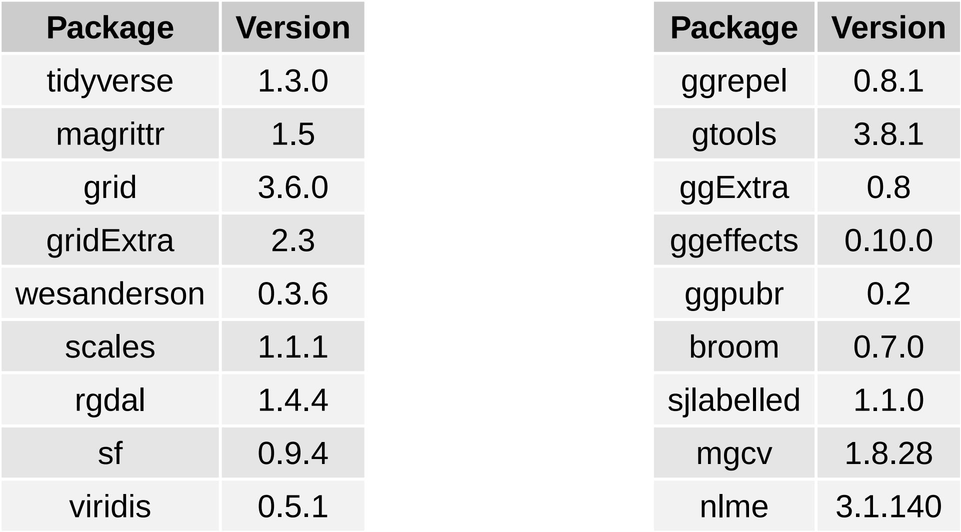
Qiang Yang

Introduction

This document includes the R codes used for the main analysis in *Yang et al. The global loss of floristic uniqueness.* The document is structured according to the order of the figures. The statistical analysis is also included in the section named by the relevant figure. Specifically, the analysis on how the extent of homogenization between regions are associated with geographic distance, climatic dissimilarity and administrative relation is in section Fig. 3. Fitting of decay of similarity between regions with increasing geographic distance is also in section Fig. 3. The analysis on the relationship between homogenization and human activities is in section Fig. 4. GAM fitting is in section Figs. S5-S6. To save space, some figure-styling codes that are insignificant in presenting the results are not shown. To reduce the size of the document, Fig. 3A,B,D,E, the map in Fig.4 and Fig. S9 are not shown. All the other figures are produced during the knitting of the original Rmarkdown document. These figures could be reproduced with the relevant R codes and data. For more questions about the code and data, contact **Qiang Yang** (qiang.yang@uni-konstanz.de).

Required packages

```
Packages_ <- c(
  "tidyverse", "magrittr", "grid", "gridExtra", "wesanderson",
  "scales", "rgdal", "sf", "viridis", "ggrepel", "gtools",
  "ggExtra", "ggeffects", "ggpubr", "broom", "sjlabelled",
  "mgcv", "nlme"
)
invisible(lapply(Packages_, library, character.only = TRUE))
```



Package	Version
tidyverse	1.3.0
magrittr	1.5
grid	3.6.0
gridExtra	2.3
wesanderson	0.3.6
scales	1.1.1
rgdal	1.4.4
sf	0.9.4
viridis	0.5.1

Package	Version
ggrepel	0.8.1
gtools	3.8.1
ggExtra	0.8
ggeffects	0.10.0
ggpubr	0.2
broom	0.7.0
sjlabelled	1.1.0
mgcv	1.8.28
nlme	3.1.140

Fig. 1

define a function for making the hex plot

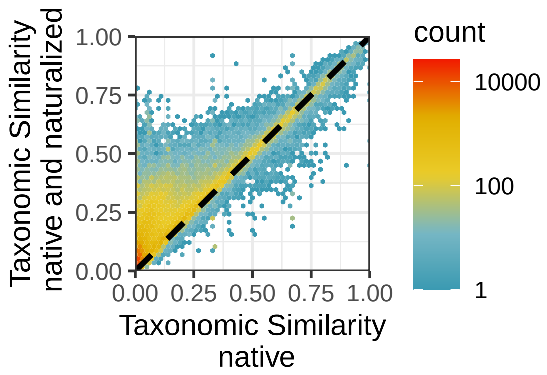
```r
pal <- wes_palette("Zissou1", 30, type = "continuous") # define color gradients
f.hex <- function(dat) {
  dat %>%
    ggplot(aes(x = native, y = native.and.naturalized)) +
    geom_hex(bins = 50, aes(fill = stat(count))) +
    scale_fill_gradientn(colours = pal, trans = "log10") +
    geom_abline(
      slope = 1, intercept = 0, color = "black", size = 1,
      linetype = "dashed"
    ) +
    coord_fixed(ratio = 1, xlim = c(0, 1), ylim = c(0, 1)) +
    scale_x_continuous(expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0)) +
    theme_bw()
}
```

define a function for making the histogram

```r
f.hist <- function(dat) {
  dat %>%
    ggplot(aes(x = homo.value)) +
    geom_histogram(bins = 20, colour = "darkgray", fill = NA, size = 0.6) +
    theme_classic() +
    geom_vline(xintercept = 0, color = "red", size = .6) +
    scale_y_continuous(expand = c(0, 0))
}
```

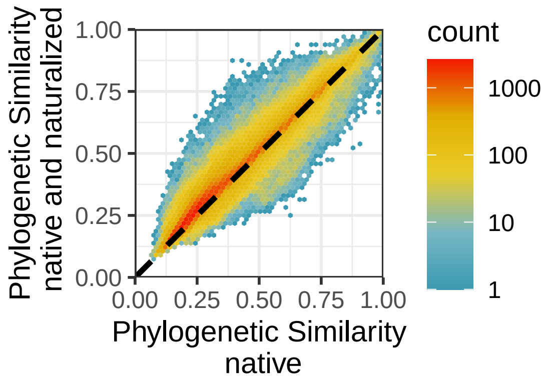
the hex plot in Fig. 1A

```r
load(file = "df.tidy.pairwise.similarity.and.homogenization.Rdata")
select <- dplyr::select
df.tidy.pairwise.dat.regional.level <-
  df.tidy.pairwise.similarity.and.homogenization
df.tidy.pairwise.dat.regional.level %>%
  select(beta.sim.native.identity, beta.sim.native.and.naturalized.identity) %>%
  rename(
    native = beta.sim.native.identity,
    native.and.naturalized = beta.sim.native.and.naturalized.identity
  ) %>%
  f.hex() +
  scale_fill_gradientn(
    colours = pal, trans = "log10",
    breaks = c(1, 100, 10000)
  ) +
  xlab("Taxonomic Similarity\n native") +
  ylab("Taxonomic Similarity\n native and naturalized")
```



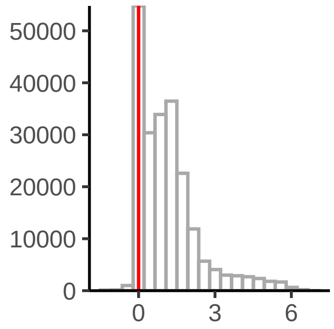
the hex plot in Fig. 1B

```r
df.tidy.pairwise.dat.regional.level %>%
  select(
    beta.sim.native.phylogeny,
    beta.sim.native.and.naturalized.phylogeny
  ) %>%
  rename(
    native = beta.sim.native.phylogeny,
    native.and.naturalized = beta.sim.native.and.naturalized.phylogeny
  ) %>%
  f.hex() +
  scale_fill_gradientn(
    colours = pal, trans = "log10",
    breaks = c(1, 10, 100, 1000)
  ) +
  xlab("Phylogenetic Similarity\n native") +
  ylab("Phylogenetic Similarity\n native and naturalized")
```



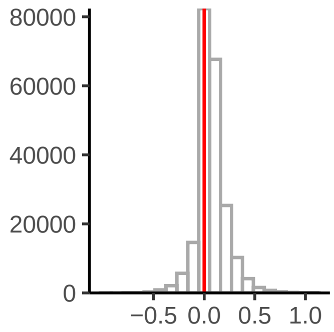
the histogram in Fig. 1A

```
dat <- df.tidy.pairwise.dat.regional.level %>%
  select(homo_by.naturalized_delta.0.001_id) %>%
  rename(homo.value = homo_by.naturalized_delta.0.001_id)
f.hist(dat) +
  scale_x_continuous(breaks = c(0, 3, 6)) +
  xlab("") +
  ylab("")
```



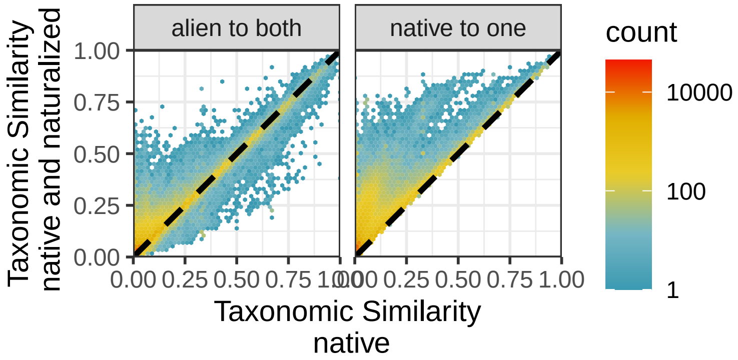
the histogram in Fig. 1B

```
dat <- df.tidy.pairwise.dat.regional.level %>%
  dplyr::select(homo_by.naturalized_delta.0.001_phylo) %>%
  rename(homo.value = homo_by.naturalized_delta.0.001_phylo)
f.hist(dat) +
  scale_x_continuous(breaks = c(-0.5, 0, 0.5, 1)) +
  xlab("") +
  ylab("")
```



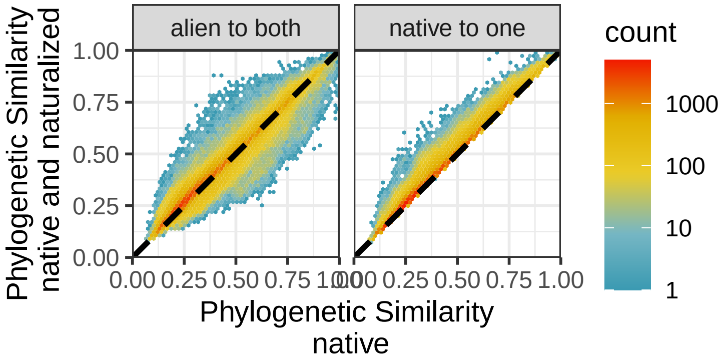
the hex plot in Fig. 1C

```
df.plot.dat.identity <- df.tidy.pairwise.dat.regional.level %>%
  select(beta.sim.native.identity:beta.sim.native.and.outside.species.identity) %>%
  mutate(row.id = 1:nrow(df.tidy.pairwise.dat.regional.level)) %>%
  rename(
    `native to one` = beta.sim.native.and.exchange.identity,
    `alien to both` = beta.sim.native.and.outside.species.identity
  ) %>%
  gather(key = "group", value = "Similarity", -row.id, -beta.sim.native.identity) %>%
  filter(group %in% c("native to one", "alien to both"))
df.plot.dat.identity %>%
  ggplot(aes(x = beta.sim.native.identity, y = Similarity)) +
  facet_wrap(~group, nrow = 1) +
  geom_hex(bins = 50, aes(fill = stat(count))) +
  scale_fill_gradientn(colours = pal, trans = "log10", breaks = c(1, 100, 10000)) +
  geom_abline(slope = 1, intercept = 0, size = 1, linetype = "dashed") +
  coord_fixed(ratio = 1, xlim = c(0, 1), ylim = c(0, 1)) +
  theme_bw() +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  xlab("Taxonomic Similarity\n native") +
  ylab("Taxonomic Similarity\n native and naturalized")
```



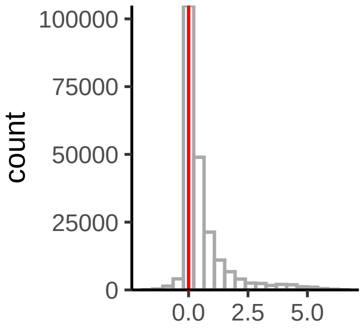
the hex plot in Fig. 1D

```
df.plot.dat.phylogeny <- df.tidy.pairwise.dat.regional.level %>%
  select(beta.sim.native.phylogeny:beta.sim.native.and.outside.species.phylogeny) %>%
  mutate(row.id = 1:nrow(df.tidy.pairwise.dat.regional.level)) %>%
  rename(
    `native to one` = beta.sim.native.and.exchange.phylogeny,
    `alien to both` = beta.sim.native.and.outside.species.phylogeny
  ) %>%
  gather(key = "group", value = "Similarity", -row.id, -beta.sim.native.phylogeny) %>%
  filter(group %in% c("native to one", "alien to both"))
df.plot.dat.phylogeny %>%
  ggplot(aes(x = beta.sim.native.phylogeny, y = Similarity)) +
  facet_wrap(~group, nrow = 1) +
  geom_hex(bins = 50, aes(fill = stat(count))) +
  scale_fill_gradientn(colours = pal, trans = "log10", breaks = c(1, 10, 100, 1000)) +
  geom_abline(slope = 1, intercept = 0, size = 1, linetype = "dashed") +
  coord_fixed(ratio = 1, xlim = c(0, 1), ylim = c(0, 1)) +
  theme_bw() +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  xlab("Phylogenetic Similarity\n native") +
  ylab("Phylogenetic Similarity\n native and naturalized")
```



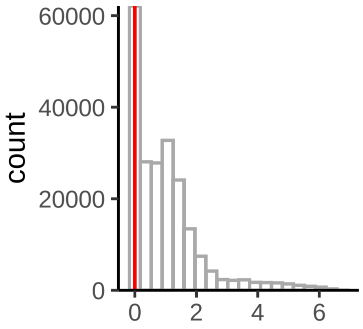
the histogram in Fig. 1C (left panel)

```
dat.taxonomic.outside <- df.tidy.pairwise.dat.regional.level %>%
  dplyr::select(homo_by.outside.species_delta.0.001_id) %>%
  rename(homo.value = homo_by.outside.species_delta.0.001_id)
f.hist(dat = dat.taxonomic.outside) +
  scale_x_continuous(breaks = c(0, 2.5, 5)) +
  xlab("")
```



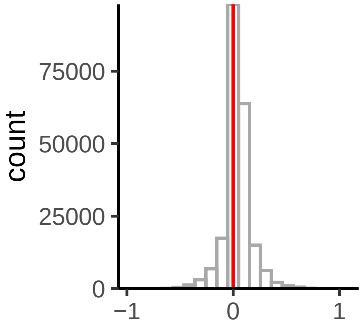
the histogram in Fig. 1C (right panel)

```
dat.taxonomic.within <- df.tidy.pairwise.dat.regional.level %>%
  dplyr::select(homo_by.exchange_delta.0.001_id) %>%
  rename(homo.value = homo_by.exchange_delta.0.001_id)
f.hist(dat = dat.taxonomic.within) +
  scale_x_continuous(breaks = c(0, 2, 4, 6)) +
  xlab("")
```



the histogram in Fig. 1D (left panel)

```
dat.phylogenetic.outside <- df.tidy.pairwise.dat.regional.level %>%
  dplyr::select(homo_by.outside.species_delta.0.001_phylo) %>%
  rename(homo.value = homo_by.outside.species_delta.0.001_phylo)
f.hist(dat = dat.phylogenetic.outside) +
  scale_x_continuous(breaks = c(-1, 0, 1)) +
  xlab("")
```



the histogram in Fig. 1D (right panel)

```
dat.phylogenetic.within <- df.tidy.pairwise.dat.regional.level %>%
  dplyr::select(homo_by.exchange_delta.0.001_phylo) %>%
  rename(homo.value = homo_by.exchange_delta.0.001_phylo)
f.hist(dat = dat.phylogenetic.within) +
  scale_x_continuous(breaks = c(0, 0.5, 1)) +
  xlab("")
```

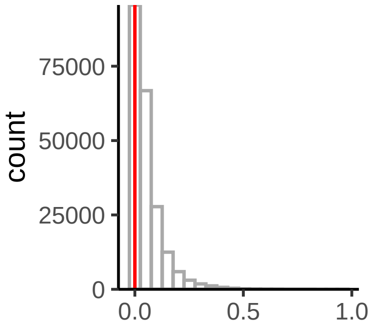


Fig. 2

Fig. 2A

```
load(file = "sp.centroids.combined.regions.Rdata")
rownamess <- rownames(sp.centroids.combined.regions@coords)
rownamess <- as.numeric(rownamess)
df.region.centroid_a <- data.frame(
  new_region_id_a = rownamess,
  lon_a = sp.centroids.combined.regions@coords[, 1],
  lat_a = sp.centroids.combined.regions@coords[, 2]
)
df.region.centroid_b <- data.frame(
  new_region_id_b = rownamess,
  lon_b = sp.centroids.combined.regions@coords[, 1],
  lat_b = sp.centroids.combined.regions@coords[, 2]
)
df <- df.tidy.pairwise.dat.regional.level %>%
  select(
    new_region_id_a, new_region_id_b,
    homo_by.naturalized_delta.0.001_id,
    homo_by.naturalized_delta.0.001_phylo
  )
df2 <- df
colnames(df2) <- c("new_region_id_b", "new_region_id_a", colnames(df)[3:4])
df <- bind_rows(df, df2)
# grouping the regions into three climatic zones ####
df <- df %>%
  left_join(df.region.centroid_a) %>%
  left_join(df.region.centroid_b)
breaks <- c(-Inf, -23.4367, 23.4367, Inf)
labels <- c("SNT", "T", "NNT")
df %<>%
  mutate(region_a_group = cut(lat_a, breaks = breaks, labels = labels)) %>%
  mutate(region_b_group = cut(lat_b, breaks = breaks, labels = labels))
df %<>% rename(
  identity = homo_by.naturalized_delta.0.001_id,
  phylogeny = homo_by.naturalized_delta.0.001_phylo
)
df %<>% gather(
  key = "quantification.approach", value = "value",
  identity, phylogeny
)
df %<>% mutate(group = paste(region_a_group, region_b_group, sep = " & "))
df %<>% anti_join(df %>% filter(group %in% c("T & NNT", "SNT & NNT", "SNT & T")))
df %<>% mutate(group = fct_relevel(
  group,
  rev(c(
    "NNT & NNT", "T & T", "SNT & SNT",
    "NNT & T", "NNT & SNT", "T & SNT"
  ))
))
df %>%
  filter(quantification.approach == "identity") %>%
  ggplot(aes(x = group, y = value)) +
  geom_boxplot(outlier.shape = NA, width = 0.5) +
  scale_color_brewer(palette = "Dark2") +
  theme_classic() +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 20, face = "bold", hjust = -0.3),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 12, color = "black"),
    plot.margin = margin(0, 0, 0, 0.7, "cm")
  ) +
  coord_flip(ylim = c(-1.2, 5)) +
  ylab("Extent of taxonomic\n homogenization") +
  xlab("") +
  geom_hline(yintercept = 0, linetype = "dashed")
```

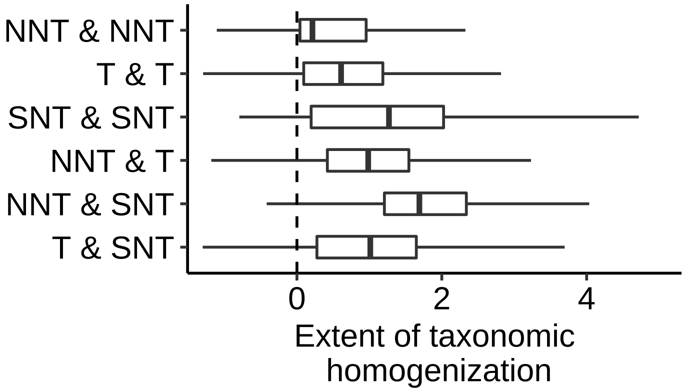


Fig. 2B

```
df %>%
  filter(quantification.approach == "phylogeny") %>%
  ggplot(aes(x = group, y = value)) +
  geom_boxplot(outlier.shape = NA, width = 0.5) +
  scale_color_brewer(palette = "Dark2") +
  theme_classic() +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 20, face = "bold", hjust = -0.3),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 12, color = "black")
  ) +
  coord_flip(ylim = c(-0.3, 0.6)) +
  ylab("Extent of phylogenetic\n homogenization") +
  xlab("") +
  geom_hline(yintercept = 0, linetype = "dashed")
```

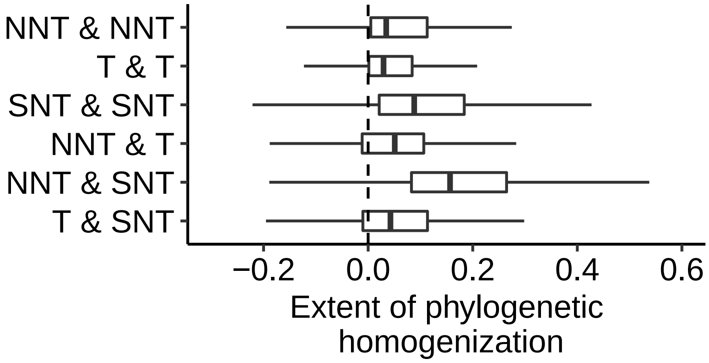

NNT & NNT
T & T
SNT & SNT
NNT & T
NNT & SNT
T & SNT
−0.2
0.0
0.2
0.4
0.6
Extent of phylogenetic homogenization

Fig. 3

Fig. 3A,B,D,E include many data points, making them very slow to open. I do not show them in the document, but the readers could reproduce them using the codes in this section if they are interested.

fitting the decay of similarity with geographic distance and climatic dissimilarity with GLMs

```
load(file = "df.regional.coredata.Rdata")
GLM.f <- function(response, predictor, formula_, newdata_) {
  df <- df.regional %>%
    dplyr::select(one_of(c(response, predictor))) %>%
    na.omit()
  colnames(df) <- c("y", "x")
  fit.obs <- glm(formula = formula_, family = binomial(link = "log"), data = df)
  predicted.obs <- predict(fit.obs, newdata = newdata_, type = "response")
  out <- newdata_ %>% mutate(predicted.obs = predicted.obs)
  return(out)
}
mod.native.tax.sim.to.geo.d <-
  GLM.f(
    response = "beta.sim.native.identity",
    predictor = "geographic_distance",
    formula_ = y ~ x + 0,
    newdata_ = data.frame(x = seq(1, 20001, length.out = 500))
  )
mod.natura.tax.sim.to.geo.d <-
  GLM.f(
    response = "beta.sim.native.and.naturalized.identity",
    predictor = "geographic_distance",
    formula_ = y ~ x + 0,
    newdata_ = data.frame(x = seq(1, 20001, length.out = 500))
  )
mod.native.phy.sim.to.geo.d <-
  GLM.f(
    response = "beta.sim.native.phylogeny",
    predictor = "geographic_distance",
    formula_ = y ~ x + 0,
    newdata_ = data.frame(x = seq(1, 20001, length.out = 500))
  )
mod.natura.phy.sim.to.geo.d <-
  GLM.f(
    response = "beta.sim.native.and.naturalized.phylogeny",
    predictor = "geographic_distance",
    formula_ = y ~ x + 0,
    newdata_ = data.frame(x = seq(1, 20001, length.out = 500))
  )
mod.native.tax.sim.to.cli.d <-
  GLM.f(
    response = "beta.sim.native.identity",
    predictor = "environmental_distance",
    formula_ = y ~ x,
    newdata_ = data.frame(x = seq(0, 15, length.out = 500))
  )
```

```
mod.natura.tax.sim.to.cli.d <-
  GLM.f(
    response = "beta.sim.native.and.naturalized.identity",
    predictor = "environmental_distance",
    formula_ = y ~ x,
    newdata_ = data.frame(x = seq(0, 15, length.out = 500))
  )
mod.native.phy.sim.to.cli.d <-
  GLM.f(
    response = "beta.sim.native.phylogeny",
    predictor = "environmental_distance",
    formula_ = y ~ x,
    newdata_ = data.frame(x = seq(0, 15, length.out = 500))
  )
mod.natura.phy.sim.to.cli.d <-
  GLM.f(
    response = "beta.sim.native.and.naturalized.phylogeny",
    predictor = "environmental_distance",
    formula_ = y ~ x,
    newdata_ = data.frame(x = seq(0, 15, length.out = 500))
  )
```

Fig. 3A

```
df.plot <- df.regional %>%
  select(
    beta.sim.native.identity,
    beta.sim.native.and.naturalized.identity,
    geographic_distance
  )
colnames(df.plot) <- c("native", "native and naturalized", "x")
df.plot %<>% gather(key = "key", value = "value", -x)
df.plot %<>% mutate(key = fct_relevel(key, c("native", "native and naturalized")))
p.taxonomic.similarity.vs.geographic.distance <- df.plot %>%
  ggplot(aes(x = x, y = value)) +
  geom_point(aes(color = key), alpha = 0.02, pch = 1, size = 0.2) +
  theme_bw() +
  theme(
    legend.position = "top",
    legend.title = element_blank()
  ) +
  scale_color_manual(values = c("black", "#AA00FF")) +
  ylab("Taxonomic similarity") +
  xlab("Geographic distance (km)") +
  ggtitle("A") +
  scale_x_continuous(breaks = seq(0, 15000, 5000))
# add fitting curves
fit.dat <- mod.native.tax.sim.to.geo.d %>%
  mutate(group = "native") %>%
  bind_rows(mod.natura.tax.sim.to.geo.d %>%
    mutate(group = "native and naturalized")) %>%
  set_colnames(c("geographic_distance", ".fitted", "group"))
```

```
p.taxonomic.similarity.vs.geographic.distance +
  geom_line(
    data = fit.dat,
    aes(x = geographic_distance, y = .fitted, color = group), size = 2
  ) +
  scale_color_manual(values = c("black", "#AA00FF"))
```

Fig. 3B

```
df.plot <- df.regional %>%
  select(
    beta.sim.native.identity,
    beta.sim.native.and.naturalized.identity,
    environmental_distance
  )
colnames(df.plot) <- c("native", "native and naturalized", "x")
df.plot %<>% gather(key = "key", value = "value", -x)
df.plot %<>% mutate(key = fct_relevel(key, c("native", "native and naturalized")))
p.taxonomic.similarity.vs.climatic.dissimilarity <- df.plot %>%
  ggplot(aes(x = x, y = value)) +
  geom_point(aes(color = key), alpha = 0.02, pch = 1, size = 0.2) +
  theme_bw() +
  theme(
    legend.position = "top",
    legend.title = element_blank()
  ) +
  scale_color_manual(values = c("black", "#AA00FF")) +
  ylab("Taxonomic similarity") +
  xlab("Climatic dissimilarity") +
  ggtitle("B") +
  scale_x_continuous(breaks = c(0, 4, 8, 12))
# add fitting curves
fit.dat <- mod.native.tax.sim.to.cli.d %>%
  mutate(group = "native") %>%
  bind_rows(mod.natura.tax.sim.to.cli.d %>%
    mutate(group = "native and naturalized")) %>%
  set_colnames(c("climatic_dissimilarity", ".fitted", "group"))
p.taxonomic.similarity.vs.climatic.dissimilarity +
  geom_line(
    data = fit.dat,
    aes(x = climatic_dissimilarity, y = .fitted, color = group), size = 2
  ) +
  scale_color_manual(values = c("black", "#AA00FF"))
```

Fig. 3D

```
df.plot <- df.regional %>%
  select(
    beta.sim.native.phylogeny,
    beta.sim.native.and.naturalized.phylogeny,
    geographic_distance
  )
colnames(df.plot) <- c("native", "native and naturalized", "x")
df.plot %<>% gather(key = "key", value = "value", -x)
df.plot %<>% mutate(key = fct_relevel(key, c("native", "native and naturalized")))
p.phylogenetic.similarity.vs.geographic.distance <- df.plot %>%
  ggplot(aes(x = x, y = value)) +
  geom_point(aes(color = key), alpha = 0.02, pch = 1, size = 0.2) +
  theme_bw() +
  theme(
    legend.position = "top",
    legend.title = element_blank()
  ) +
  scale_color_manual(values = c("black", "#AA00FF")) +
  ylab("phylogenetic similarity") +
  xlab("Geographic distance (km)") +
  ggtitle("D") +
  scale_x_continuous(breaks = seq(0, 15000, 5000))
# add fitting curves
fit.dat <- mod.native.phy.sim.to.geo.d %>%
  mutate(group = "native") %>%
  bind_rows(mod.natura.phy.sim.to.geo.d %>%
    mutate(group = "native and naturalized")) %>%
  set_colnames(c("geographic_distance", ".fitted", "group"))
p.phylogenetic.similarity.vs.geographic.distance +
  geom_line(
    data = fit.dat,
    aes(x = geographic_distance, y = .fitted, color = group), size = 2
  ) +
  scale_color_manual(values = c("black", "#AA00FF"))
```

Fig. 3E

```
df.plot <- df.regional %>%
  select(
    beta.sim.native.phylogeny,
    beta.sim.native.and.naturalized.phylogeny,
    environmental_distance
  )
colnames(df.plot) <- c("native", "native and naturalized", "x")
df.plot %<>% gather(key = "key", value = "value", -x)
df.plot %<>% mutate(key = fct_relevel(key, c("native", "native and naturalized")))
p.phylogenetic.similarity.vs.climatic.dissimilarity <- df.plot %>%
  ggplot(aes(x = x, y = value)) +
  geom_point(aes(color = key), alpha = 0.02, pch = 1, size = 0.2) +
  theme_bw() +
  theme(
    legend.position = "top",
    legend.title = element_blank()
  ) +
  scale_color_manual(values = c("black", "#AA00FF")) +
  ylab("phylogenetic similarity") +
  xlab("Climatic dissimilarity") +
  ggtitle("E") +
  scale_x_continuous(breaks = c(0, 4, 8, 12))
# add fitting curves
fit.dat <- mod.native.phy.sim.to.cli.d %>%
  mutate(group = "native") %>%
  bind_rows(mod.natura.phy.sim.to.cli.d %>%
    mutate(group = "native and naturalized")) %>%
  set_colnames(c("climatic_dissimilarity", ".fitted", "group"))
p.phylogenetic.similarity.vs.climatic.dissimilarity +
  geom_line(
    data = fit.dat,
    aes(x = climatic_dissimilarity, y = .fitted, color = group), size = 2
  ) +
  scale_color_manual(values = c("black", "#AA00FF"))
```

Analysis of the relationship between homogenization and geographic distance, climatic dissimilarity and administrative relationship

```
load(file = "df.regional.coredata.Rdata")
df.regional %<>% select(homo_by.naturalized_delta.0.001_id:administrative_relation)
df.regional$administrative_relation[df.regional$administrative_relation
== "others"] <- "dependency"
df.regional$administrative_relation[df.regional$administrative_relation
== "no administrative relation"] <- "no relation"
df.regional %<>% na.omit()
df.regional %<>%
  mutate(administrative_relation = fct_relevel(
    administrative_relation,
    c("no relation", "dependency", "same country")
  ))
MRM.taxonomic <- lm(homo_by.naturalized_delta.0.001_id ~
geographic_distance * environmental_distance * administrative_relation,
data = df.regional
)
MRM.phylogenetic <- lm(homo_by.naturalized_delta.0.001_phylo ~
geographic_distance * environmental_distance * administrative_relation,
data = df.regional
)
load(file = "new.data.explanatory.variable.Rdata")
# make predictions
df.fitted.taxonomic <- new.data %>%
  mutate(.fitted = predict(MRM.taxonomic, newdata = new.data))
df.fitted.phylogenetic <- new.data %>%
  mutate(.fitted = predict(MRM.phylogenetic, newdata = new.data))
```

Fig. 3C

```
df.fitted.taxonomic %>%
  ggplot(aes(x = `geographic_distance`, y = `.fitted`)) +
  geom_line(size = 1, aes(
    linetype = `administrative_relation`,
    color = as.factor(`environmental_distance`)
  )) +
  scale_color_manual(
    values = rev(c("darkred", "red", "orange", "deepskyblue", "blue"))
  ) +
  scale_linetype_manual(values = c("solid", "dotdash", "dotted")) +
  theme_bw() +
  theme(legend.title = element_blank()) +
  scale_x_continuous(breaks = seq(0, 20000, 5000))
```

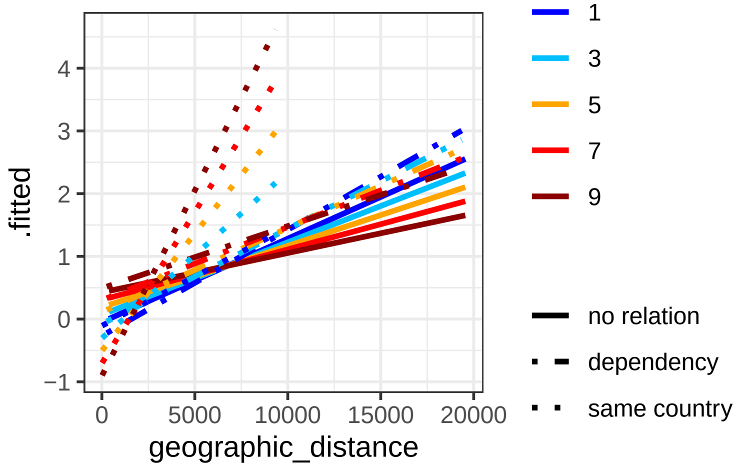


Fig. 3F

```
df.fitted.phylogenetic %>%
  ggplot(aes(x = `geographic_distance`, y = `.fitted`)) +
  geom_line(size = 1, aes(
    linetype = `administrative_relation`,
    color = as.factor(`environmental_distance`)
  )) +
  scale_color_manual(
    values = rev(c("darkred", "red", "orange", "deepskyblue", "blue"))
  ) +
  scale_linetype_manual(values = c("solid", "dotdash", "dotted")) +
  theme_bw() +
  theme(legend.title = element_blank()) +
  scale_x_continuous(breaks = seq(0, 20000, 5000))
```

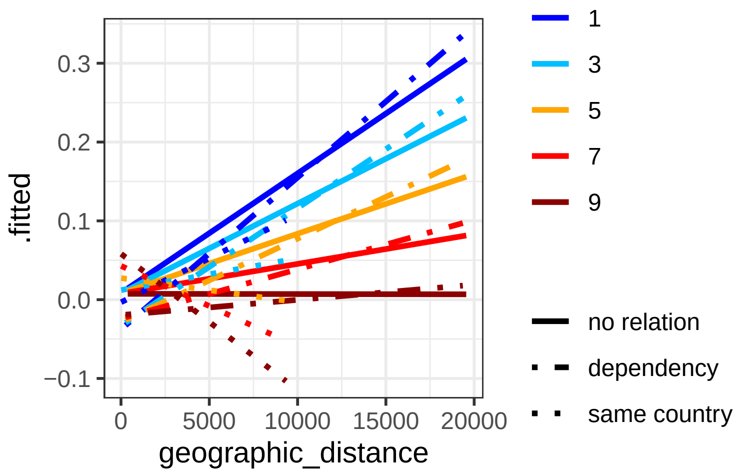

0.3
0.2
0.1
0.0
−0.1
.fitted
0
5000
10000
15000
20000
geographic_distance
1
3
5
7
9
no relation
dependency
same country

Fig. 4

the map in Fig. 4A

```r
load(file = "wrld.sf.Rdata")
load(file = "wrld.regional.level.Rdata")
load(file = "small.island.centroids.Rdata")
colors1 <- rev(magma(7)) # color gradients for taxonomic homogenization
colors2 <- c("green", rev(magma(6)))
scale_manual_id <- list(
  scale_fill_manual(values = colors1, drop = FALSE),
  scale_color_manual(values = colors1, drop = FALSE)
)
wrld.regional.level %<>%
  mutate(mean.homo.id.group = paste("<", mean.homo.id.group))
wrld.regional.level %<>%
  mutate(mean.homo.phylo.group = paste("<", mean.homo.phylo.group))
small.island.centroids %<>%
  mutate(mean.homo.id.group = paste("<", mean.homo.id.group))
small.island.centroids %<>%
  mutate(mean.homo.phylo.group = paste("<", mean.homo.phylo.group))
ggplot() +
  geom_sf(data = wrld.sf, fill = "gray", color = "gray") +
  geom_sf(
    data = wrld.regional.level %>% filter(island == 0),
    aes(fill = mean.homo.id.group), color = NA
  ) +
  geom_sf(
    data = wrld.regional.level %>% filter(island == 1),
    aes(fill = mean.homo.id.group), color = NA
  ) +
  geom_sf(data = small.island.centroids, aes(color = mean.homo.id.group), shape = 1) +
  coord_sf(crs = "+proj=wag4 +lon_0=0 +datum=WGS84 +units=m +no_defs") +
  theme_minimal() +
  theme(
    legend.position = "right",
    legend.title = element_blank()
  ) +
  scale_manual_id
```

the map in Fig. 4B

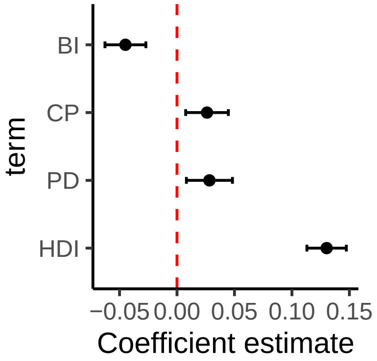
```r
scale_manual_phylo <- list(
  scale_fill_manual(values = colors2, drop = FALSE),
  scale_color_manual(values = colors2, drop = FALSE)
)
ggplot() +
  geom_sf(data = wrld.sf, fill = "gray", color = "gray") +
  geom_sf(
    data = wrld.regional.level %>% filter(island == 0),
    aes(fill = mean.homo.phylo.group), color = NA
  ) +
  geom_sf(
    data = wrld.regional.level %>% filter(island == 1),
    aes(fill = mean.homo.phylo.group), color = NA
  ) +
  geom_sf(
    data = small.island.centroids,
    aes(color = mean.homo.phylo.group), shape = 1
  ) +
  coord_sf(crs = "+proj=wag4 +lon_0=0 +datum=WGS84 +units=m +no_defs") +
  theme_minimal() +
  theme(
    legend.position = "right",
    legend.title = element_blank()
  ) +
  scale_manual_phylo
```

regress the homogenization to human activities

```
load(file = "df.average.Rdata")
load(file = "df.region.social.economic.Rdata")
df <- df.average %>% left_join(df.region.social.economic, by = c("new_region_id"))
df %<>% dplyr::select(
  new_region_id,
  mean.homo.id,
  mean.homo.phylo,
  population.density_median.value,
  hdi.1990_median.value,
  cropland.value_median.value,
  biodiversity.intactness_median.value
)
df %<>% mutate(population.density_median.value = log(population.density_median.value + 1))
colnames(df) <- c(
  "new_region_id", "mean.homo.id", "mean.homo.phylo",
  "PD", "HDI", "CP", "BI"
)
df %<>% na.omit()
df %<>% mutate_at(c("BI", "CP", "PD", "HDI"), scale)
library(spdep)
load(file = "sp.combined.regions.Rdata")
sp.combined.regions %<>% subset(new_region_id %in% df$new_region_id)
sp.combined.regions@data %<>% left_join(df)
# define the full model
lm.orig.taxonomic <- lm(
  formula = mean.homo.id ~ BI + CP + PD + HDI,
  data = sp.combined.regions@data
)
lm.orig.phylogenetic <- lm(
  formula = mean.homo.id ~ BI + CP + PD + HDI,
  data = sp.combined.regions@data
)
# add an autocovariate to the model
autocov_taxonomic <- autocov_dist(lm.orig.taxonomic$residuals,
  xy = coordinates(sp.combined.regions),
  nbs = 100,
  type = "inverse.squared", zero.policy = TRUE,
  style = "S", longlat = NULL
)
autocov_phylogenetic <- autocov_dist(lm.orig.phylogenetic$residuals,
  xy = coordinates(sp.combined.regions),
  nbs = 100,
  type = "inverse.squared", zero.policy = TRUE,
  style = "S", longlat = NULL
)
# add autocovariate to the data
sp.combined.regions@data %<>%
  mutate(
    autocov_taxonomic = autocov_taxonomic,
    autocov_phylogenetic = autocov_phylogenetic
  )
# re-do the regression
df.mod.taxonomic <- lm(mean.homo.id ~ BI + CP + PD + HDI + autocov_taxonomic,
  data = sp.combined.regions@data
)
df.mod.phylogenetic <- lm(mean.homo.phylo ~ BI + CP + PD + HDI + autocov_phylogenetic,
  data = sp.combined.regions@data
)
df.mod.coefficient.taxonomic <- tidy(df.mod.taxonomic)
df.mod.coefficient.phylogenetic <- tidy(df.mod.phylogenetic)
```

the points-error.bar plot in Fig. 4A

```
df.mod.coefficient.taxonomic %>%
  filter(term %in% c("PD", "HDI", "CP", "BI")) %>%
  mutate(term = fct_relevel(term, c("HDI", "PD", "CP", "BI"))) %>%
  ggplot(aes(x = term, y = estimate)) +
  geom_hline(yintercept = 0, col = "red", linetype = "dashed", size = 0.5) +
  geom_point() +
  geom_errorbar(aes(
    ymin = estimate - 1.96 * std.error,
    ymax = estimate + 1.96 * std.error
  ), width = .1) +
  coord_flip() +
  theme_classic() +
  scale_y_continuous(breaks = c(-0.05, 0, 0.05, 0.10, 0.15, 0.20)) +
  ylab("Coefficient estimate")
```



the points-error.bar plot in Fig. 4B

```
df.mod.coefficient.phylogenetic %>%
  filter(term %in% c("PD", "HDI", "CP", "BI")) %>%
  mutate(term = fct_relevel(term, c("HDI", "PD", "CP", "BI"))) %>%
  ggplot(aes(x = term, y = estimate)) +
  geom_hline(yintercept = 0, col = "red", linetype = "dashed", size = 0.5) +
  geom_point() +
  geom_errorbar(aes(
    ymin = estimate - 1.96 * std.error,
    ymax = estimate + 1.96 * std.error
  ), width = .1) +
  coord_flip() +
  theme_classic() +
  scale_y_continuous(breaks = c(-0.01, 0, 0.01, 0.02, 0.03)) +
  ylab("Coefficient estimate")
```

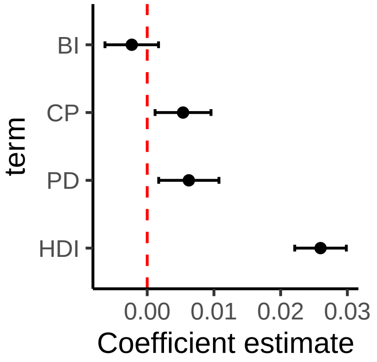


Fig. S2

```r
load("df.tidy.pairwise.similarity.and.homogenization.Rdata")
df <- df.tidy.pairwise.similarity.and.homogenization
percentage.change.taxonomic <-
  (df$beta.sim.native.and.naturalized.identity / df$beta.sim.native.identity - 1) * 100
percentage.change.phylogenetic <-
  (df$beta.sim.native.and.naturalized.phylogeny / df$beta.sim.native.phylogeny - 1) * 100
percentage.change.taxonomic[is.na(percentage.change.taxonomic)] <- Inf
percentage.change.taxonomic[percentage.change.taxonomic >= 3000] <- 3000
p1 <- data.frame(x = percentage.change.taxonomic) %>%
  ggplot(aes(x = x)) +
  geom_histogram(color = "black", fill = "white") +
  theme_classic() +
  geom_vline(
    xintercept = median(percentage.change.taxonomic),
    color = "red", size = 1, linetype = "dashed"
  ) +
  xlab("Percent change in taxonomic similarity") +
  ylab("Count") +
  scale_x_continuous(
    breaks = c(0, 500, 1000, 1500, 2000, 2970),
    labels = c("0", "500%", "1,000%", "1,500%", "2,000%", "Inf")
  ) +
  ggtitle("A")
p2 <- data.frame(x = percentage.change.phylogenetic) %>%
  ggplot(aes(x = x)) +
  geom_histogram(color = "black", fill = "white", bins = 50) +
  theme_classic() +
  geom_vline(
    xintercept = median(percentage.change.phylogenetic),
    color = "red", size = 1, linetype = "dashed"
  ) +
  xlab("Percent change in phylogenetic similarity") +
  ylab("Count") +
  scale_x_continuous(
    breaks = c(-50, 0, 50, 100),
    labels = c("-50%", "0%", "50%", "100%")
  ) +
  coord_cartesian(xlim = c(-75, 100)) +
  ggtitle("B")
grid.arrange(p1, p2, ncol = 1)
```

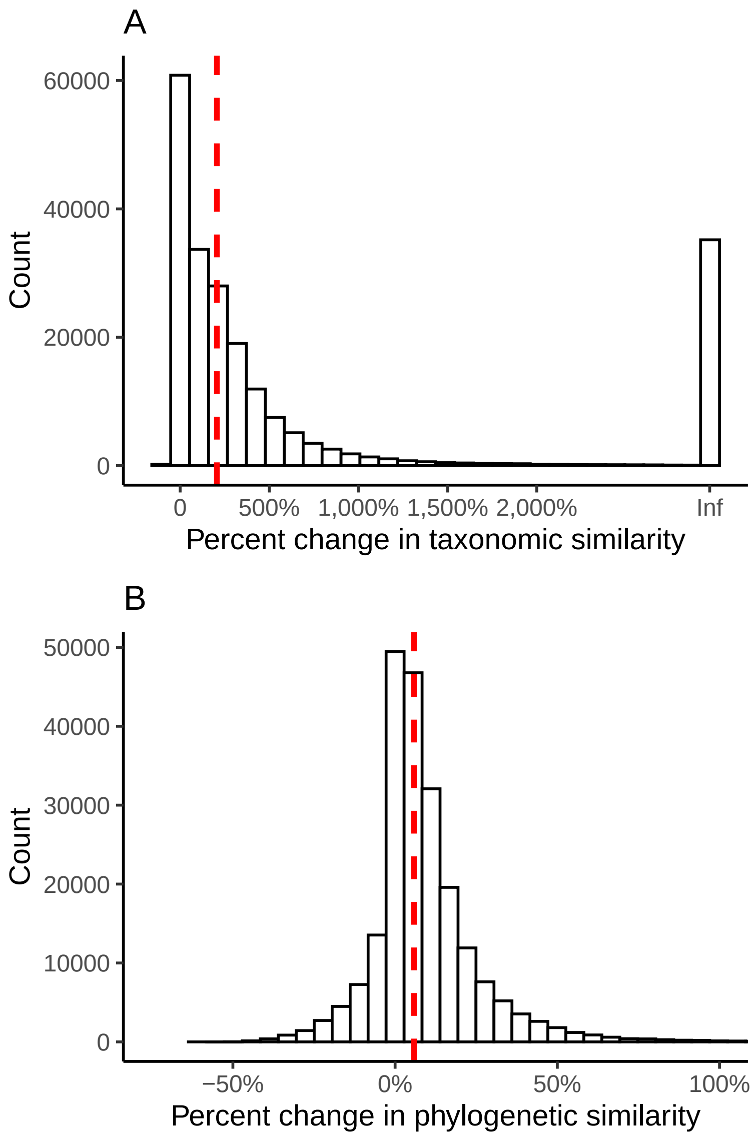
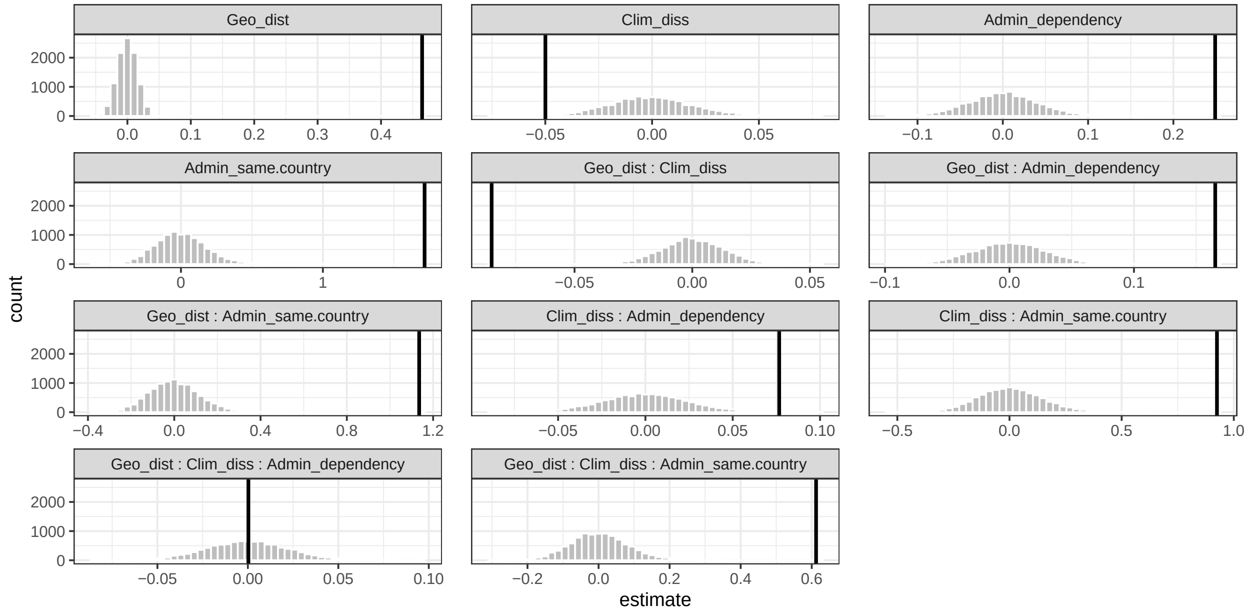

A
60000
40000
20000
0
Count
0
500%
1,000%
1,500%
2,000%
Inf
Percent change in taxonomic similarity
B
50000
40000
30000
20000
10000
0
Count
−50%
0%
50%
100%
Percent change in phylogenetic similarity

Fig. S3-S4

The permutation results were saved as df.MRM.permutation.Rdata.

```r
load(file = "df.MRM.permutation.Rdata")
df.taxonomic.coefficient.estimates <- df.MRM.permutation %>%
  dplyr::select(perm.id, perm.note, MRM.L.perm.taxonomic.tidy) %>%
  unnest(cols = MRM.L.perm.taxonomic.tidy) %>%
  filter(term != "(Intercept)")
df.term <- df.taxonomic.coefficient.estimates %>%
  filter(perm.id == 1) %>%
  dplyr::select(term) %>%
  rename(old.term = term) %>%
  mutate(term = c(
    "Geo_dist",
    "Clim_diss",
    "Admin_dependency",
    "Admin_same.country",
    "Geo_dist : Clim_diss",
    "Geo_dist : Admin_dependency",
    "Geo_dist : Admin_same.country",
    "Clim_diss : Admin_dependency",
    "Clim_diss : Admin_same.country",
    "Geo_dist : Clim_diss : Admin_dependency",
    "Geo_dist : Clim_diss : Admin_same.country"
  ))
df.taxonomic.coefficient.estimates %<>%
  rename(old.term = term) %>%
  left_join(df.term) %>%
  dplyr::select(-old.term)
df.taxonomic.coefficient.estimates.obs <- df.taxonomic.coefficient.estimates %>%
  filter(perm.note == "obs") %>% # observed estimates
  dplyr::select(perm.id, perm.note, term, estimate) %>%
  rename(estimate.obs = estimate)

df.taxonomic.coefficient.estimates %<>%
  dplyr::select(estimate, term) %>%
  mutate(term = fct_relevel(term, filter(df.taxonomic.coefficient.estimates, perm.id == 1)$term))
df.taxonomic.coefficient.estimates %>%
  ggplot(aes(x = estimate)) +
  geom_histogram(bins = 50, fill = "gray", color = "white") +
  geom_vline(
    data = df.taxonomic.coefficient.estimates.obs %>%
      mutate(term = fct_relevel(term,df.taxonomic.coefficient.estimates.obs$term)),
    aes(xintercept = estimate.obs), size = 1
  ) +
  facet_wrap(~term, scales = "free_x", ncol = 3) +
  theme_bw() +
  theme(
    panel.spacing.x = unit(6, "mm"),
    legend.position = "none",
    strip.text = element_text(size = 9)
  )
```



```
df.phylogenetic.coefficient.estimates <- df.MRM.permutation %>%
  dplyr::select(perm.id, perm.note, MRM.L.perm.phylogenetic.tidy) %>%
  unnest(cols = MRM.L.perm.phylogenetic.tidy) %>%
  filter(term != "(Intercept)")
df.term <- df.phylogenetic.coefficient.estimates %>%
  filter(perm.id == 1) %>%
  dplyr::select(term) %>%
  rename(old.term = term) %>%
  mutate(term = c(
    "Geo_dist",
    "Clim_diss",
    "Admin_dependency",
    "Admin_same.country",
    "Geo_dist : Clim_diss",
    "Geo_dist : Admin_dependency",
    "Geo_dist : Admin_same.country",
    "Clim_diss : Admin_dependency",
    "Clim_diss : Admin_same.country",
    "Geo_dist : Clim_diss : Admin_dependency",
    "Geo_dist : Clim_diss : Admin_same.country"
  ))
df.phylogenetic.coefficient.estimates %<>%
  rename(old.term = term) %>%
  left_join(df.term) %>%
  dplyr::select(-old.term)
df.phylogenetic.coefficient.estimates.obs <- df.phylogenetic.coefficient.estimates %>%
  filter(perm.note == "obs") %>% # observed estimates
  dplyr::select(perm.id, perm.note, term, estimate) %>%
  rename(estimate.obs = estimate)

df.phylogenetic.coefficient.estimates %<>%
  dplyr::select(estimate, term) %>%
  mutate(term = fct_relevel(term, filter(df.phylogenetic.coefficient.estimates, perm.id == 1)$term))
df.phylogenetic.coefficient.estimates %>%
  ggplot(aes(x = estimate)) +
  geom_histogram(bins = 50, fill = "gray", color = "white") +
  geom_vline(
    data = df.phylogenetic.coefficient.estimates.obs %>%
      mutate(term = fct_relevel(term,df.phylogenetic.coefficient.estimates.obs$term)),
    aes(xintercept = estimate.obs), size = 1
  ) +
  facet_wrap(~term, scales = "free_x", ncol = 3) +
  theme_bw() +
  theme(
    panel.spacing.x = unit(6, "mm"),
    legend.position = "none",
    strip.text = element_text(size = 9)
  )
```

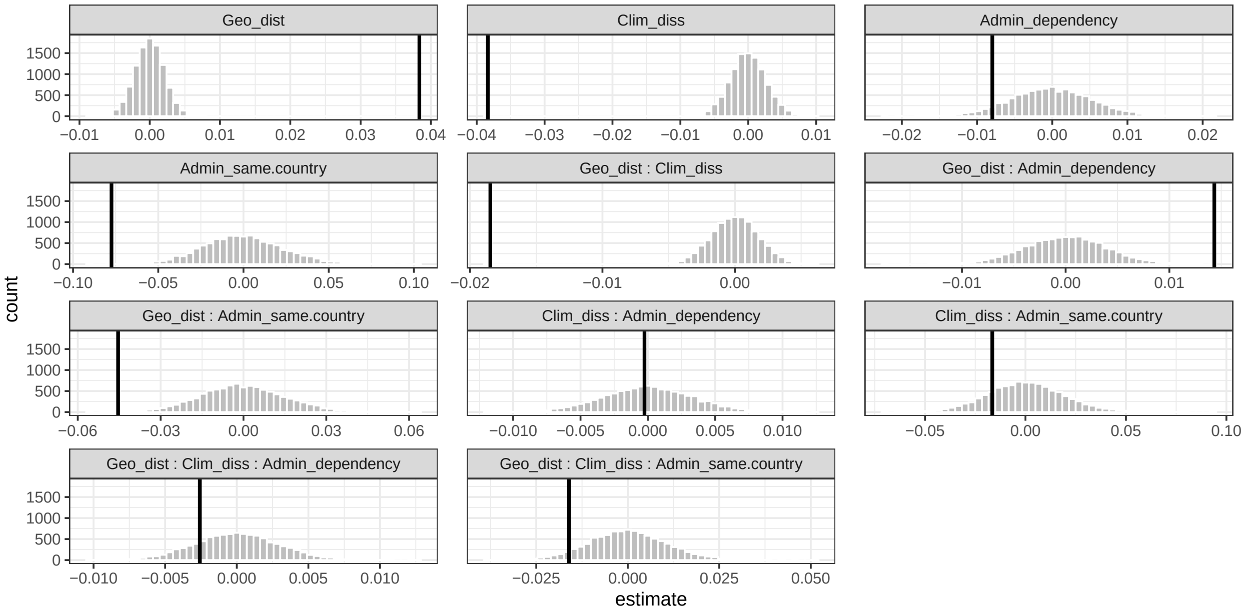


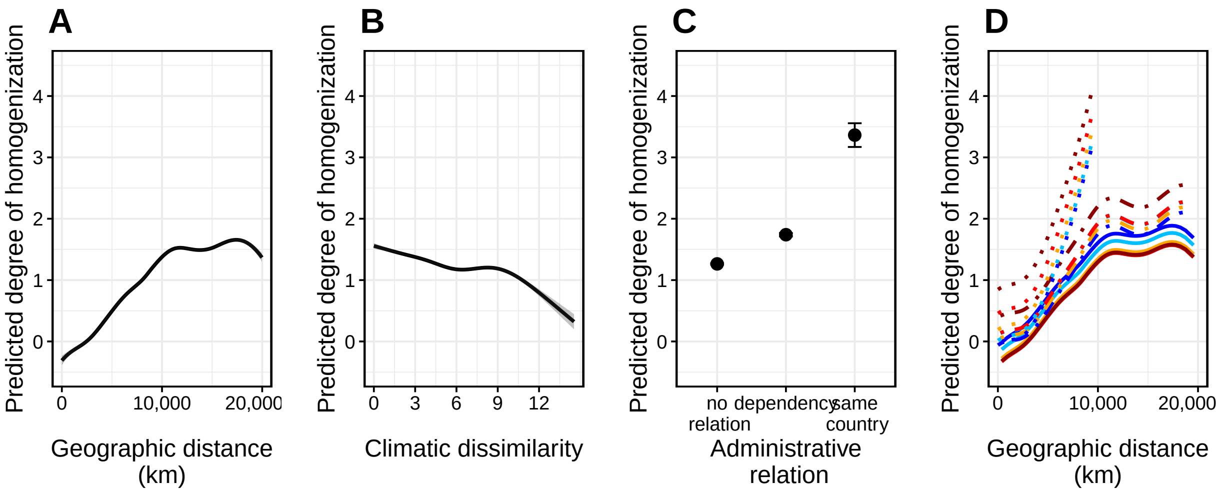
Fig. S5-S6

Perform GAM

```r
load(file = "df.regional.coredata.Rdata")
# tidy data
df.regional %<>%
  select(homo_by.naturalized_delta.0.001_id:administrative_relation)
df.regional$administrative_relation[df.regional$administrative_relation
== "others"] <- "dependency"
df.regional$administrative_relation[df.regional$administrative_relation
== "no administrative relation"] <- "no relation"
df.regional %<>% na.omit()
df.regional %<>%
  mutate(administrative_relation = fct_relevel(
    administrative_relation,
    c("no relation", "dependency", "same country")
  ))
# define a gam function
func.gam <- function(Res = "homo_by.naturalized_delta.0.001_id", k.value = 5) {
  df <- df.regional %>%
    select(one_of(
      Res, "geographic_distance",
      "environmental_distance", "administrative_relation"
    ))
  colnames(df) <- c(
    "Response", "geographic_distance",
    "environmental_distance", "administrative_relation"
  )
  gam.mod <- mgcv::gam(Response ~ administrative_relation +
    s(geographic_distance, k = k.value) +
    s(environmental_distance, k = k.value) +
    s(geographic_distance, environmental_distance,
      k = k.value
    ) +
    s(geographic_distance,
      by = administrative_relation,
      k = k.value
    ) +
    s(environmental_distance,
      by = administrative_relation,
      k = k.value
    ),
  data = df, method = "ML"
  )
  return(gam.mod)
}
# run gams
# set.seed(1234)
# tic() # about two hours to finish, I just load the saved results
# gam.taxonomic.5 <- func.gam(Res = "homo_by.naturalized_delta.0.001_id",
# k.value = 5)
# gam.taxonomic.6 <- func.gam(Res = "homo_by.naturalized_delta.0.001_id",
# k.value = 6)
# gam.taxonomic.7 <- func.gam(Res = "homo_by.naturalized_delta.0.001_id",
# k.value = 7)
# gam.phylogenetic.5 <- func.gam(
# Res = "homo_by.naturalized_delta.0.001_phylo", k.value = 5)
# gam.phylogenetic.6 <- func.gam(
# Res = "homo_by.naturalized_delta.0.001_phylo", k.value = 6)
# gam.phylogenetic.7 <- func.gam(
# Res = "homo_by.naturalized_delta.0.001_phylo", k.value = 7)
# toc()
# gams.LL <- list(gam.taxonomic.5, gam.taxonomic.6, gam.taxonomic.7,
#                gam.phylogenetic.5, gam.phylogenetic.6, gam.phylogenetic.7)
# names(gams.LL) <- c("gam.taxonomic.5", "gam.taxonomic.6", "gam.taxonomic.7",
# "gam.phylogenetic.5", "gam.phylogenetic.6", "gam.phylogenetic.7")
# save(gams.LL, file = paste0(data.path, "/gams.LL.Rdata"))
load("gams.LL.Rdata")
# define a theme for plot
mytheme1 <- theme_bw() +
  theme(
    plot.title = element_text(size = 20, face = "bold", hjust = -0.02),
    axis.text = element_text(color = "black", size = 11),
    axis.title = element_text(color = "black", size = 14),
    axis.line = element_blank(), axis.ticks = element_line(color = "black"),
    panel.border = element_rect(
      fill = NULL, colour = NULL, size = 1,
      linetype = "solid", color = "black",
      inherit.blank = FALSE
    )
  )
# define the range of the predictors
geo_range <- seq(0, max(df.regional$geographic_distance),
  length.out = 100
) %>%
  round()
geo_range_predictor <- paste(geo_range, collapse = ", ")
geo_range_predictor <- paste0(
  "geographic_distance [",
  geo_range_predictor, "]"
)
env_range <- seq(0, max(df.regional$environmental_distance),
  length.out = 100
) %>%
  round(digits = 2)
env_range_predictor <- paste(env_range, collapse = ", ")
env_range_predictor <- paste0(
  "environmental_distance [",
  env_range_predictor, "]"
)
load(file = "new.data.explanatory.variable.Rdata")
# define a function for figure
gam.figure.func <- function(gam.mod = gams.LL$gam.taxonomic.5,
                            yrange = c(-0.5, 4.5),
                            ymin.legend1 = 2.5, ymax.legend1 = 4.5,
                            ymin.legend2 = -Inf, ymax.legend2 = 1) {
  df.partial.geographic_distance <-
    ggpredict(gam.mod, geo_range_predictor) %>%
    as_tibble() %>%
    dplyr::select(x, predicted, conf.low, conf.high)
  p.partial.geographic_distance <- df.partial.geographic_distance %>%
    ggplot(aes(x = x, y = predicted)) +
    geom_line(size = 1) +
    geom_ribbon(aes(ymax = conf.high, ymin = conf.low), alpha = 0.3) +
    mytheme1 +
    xlab("Geographic distance\n(km)") +
    ylab("\nPredicted degree of homogenization") +
    scale_x_continuous(
      breaks = seq(0, 20000, 10000),
      labels = c("0\n ", "10,000\n ", "20,000   \n")
    ) +
    ggtitle("A") +
    coord_cartesian(ylim = yrange)
  df.partial.environmental_distance <-
    ggpredict(gam.mod, env_range_predictor) %>%
    as_tibble() %>%
    dplyr::select(x, predicted, conf.low, conf.high)
  p.partial.environmental_distance <- df.partial.environmental_distance %>%
    ggplot(aes(x = x, y = predicted)) +
    geom_line(size = 1) +
    geom_ribbon(aes(ymax = conf.high, ymin = conf.low), alpha = 0.3) +
    mytheme1 +
    xlab("Climatic dissimilarity\n") +
    ylab("\nPredicted degree of homogenization") +
    scale_x_continuous(
      breaks = c(0, 3, 6, 9, 12),
      labels = c("0\n ", "3\n ", "6\n", "9\n", "12\n")
    ) +
    ggtitle("B") +
    coord_cartesian(ylim = yrange)
  df.partial.administrative_relation <-
    ggpredict(gam.mod, "administrative_relation") %>%
    as_tibble()
  df.partial.administrative_relation %<>%
    mutate(xlabel = get_labels(x)) %>% as_tibble()
  df.partial.administrative_relation %<>%
    mutate(
      xlabel =
        fct_relevel(
          xlabel,
          c("no relation", "dependency", "same country")
        )
    )
  p.partial.administrative_relation <- df.partial.administrative_relation %>%
    ggplot(aes(x = xlabel, y = predicted)) +
    geom_point(size = 3) +
    geom_errorbar(aes(ymin = conf.low, ymax = conf.high), width = 0.2) +
    mytheme1 +
    xlab("Administrative\n relation") +
    ylab("\nPredicted degree of homogenization") +
    scale_x_discrete(
      breaks = c("no relation", "dependency", "same country"),
      labels = c("no\n relation", "dependency", "same\n country")
    ) +
    ggtitle("C") +
    coord_cartesian(ylim = yrange)
  df.predicted <- new.data %>%
    mutate(predicted = predict(object = gam.mod, newdata = new.data))
  df.predicted %<>%
    mutate(environmental_distance = as.factor(environmental_distance))
  p.predicted <- df.predicted %>%
    ggplot(aes(x = geographic_distance, y = predicted)) +
    geom_line(size = 1, aes(
      linetype = `administrative_relation`,
      color = `environmental_distance`
    )) +
    scale_color_manual(
      values = rev(c("darkred", "red", "orange", "deepskyblue", "blue"))
    ) +
    scale_linetype_manual(values = c("solid", "dotdash", "dotted")) +
    mytheme1 +
    scale_x_continuous(
      breaks = seq(0, 20000, 10000),
      labels = c("0\n", "10,000\n", "20,000    \n")
    ) +
    xlab("Geographic distance\n(km)") +
    ylab("\nPredicted degree of homogenization") +
    ggtitle("D") +
    coord_cartesian(ylim = yrange, xlim = c(0, 20000)) +
    theme(legend.position = "none")
  figure.LL <- list(
    p.partial.geographic_distance,
    p.partial.environmental_distance,
    p.partial.administrative_relation,
    p.predicted
  )
  return(figure.LL)
}
```

Fig. S5

```
figures <- gam.figure.func(gam.mod = gams.LL$gam.taxonomic.6,
                           yrange = c(-0.5, 4.5), ymin.legend1 = 2.5,
                           ymax.legend1 = 4.5, ymin.legend2 = -Inf,
                           ymax.legend2 = 1)
grid.arrange(figures[[1]], figures[[2]],
             figures[[3]], figures[[4]], nrow = 1)
```



make Fig. S6

```
figures <- gam.figure.func(gam.mod = gams.LL$gam.phylogenetic.6,
                           yrange = c(-0.1, 0.28), ymin.legend1 = 0.13,
                           ymax.legend1 = 0.26, ymin.legend2 = -Inf,
                           ymax.legend2 = 0)
grid.arrange(figures[[1]], figures[[2]],
             figures[[3]], figures[[4]], nrow = 1)
```

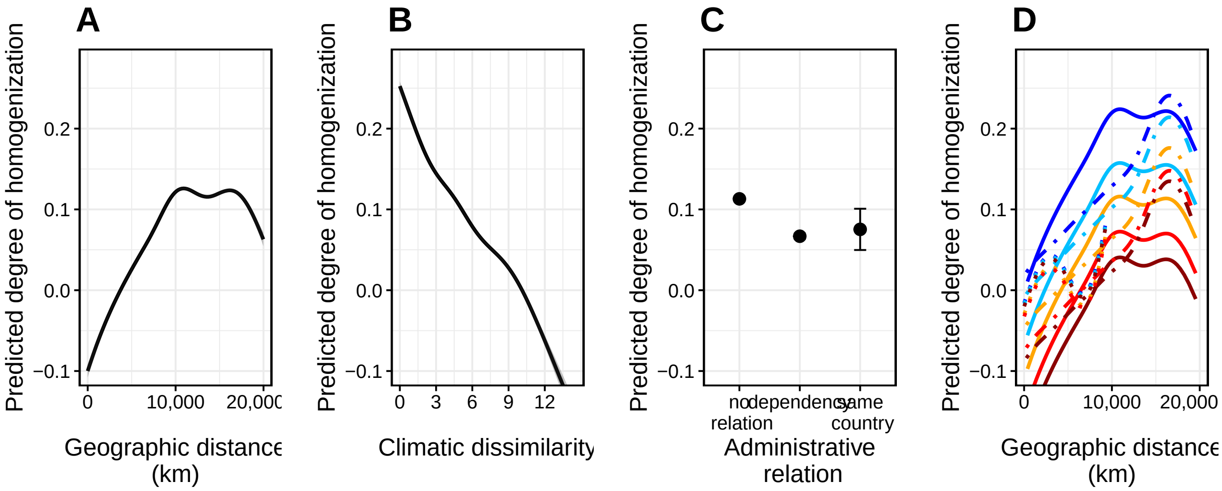


Fig. S7

```r
load(file = "df_island.Rdata")
load(file = "df.average.Rdata")
# add island information to df.average
# excluding the region Chile, new_region_id 647
df.average %<>%
  filter(new_region_id != 647) %>%
  left_join(df_island %>% dplyr::select(new_region_id, island) %>% distinct())
# make the graph
df <- df.average %>%
  dplyr::select(new_region_id, mean.homo.id, mean.homo.phylo, island) %>%
  rename(taxonomic = mean.homo.id, phylogenetic = mean.homo.phylo) %>%
  gather(key = "Group", value = "value", taxonomic, phylogenetic)
df %<>% mutate(Group = fct_relevel(Group, c("taxonomic", "phylogenetic")))
df %<>% mutate(region = ifelse(island == 1, "island", "mainland region"))
p <- ggboxplot(df,
  x = "region", y = "value",
  color = "region", palette = "jco",
  add = "jitter", outlier.shape = NA, add.params = list(shape = 1),
  facet.by = "Group", short.panel.labs = FALSE
)
p <- p + stat_compare_means(label = "p.format")
p <- facet(p, facet.by = "Group", scales = "free_y")
p + ylab("Mean degree of homogenization") + xlab("") +
  theme(
    strip.text = element_text(size = 12),
    legend.position = "none"
  ) +
  xlab("")
```

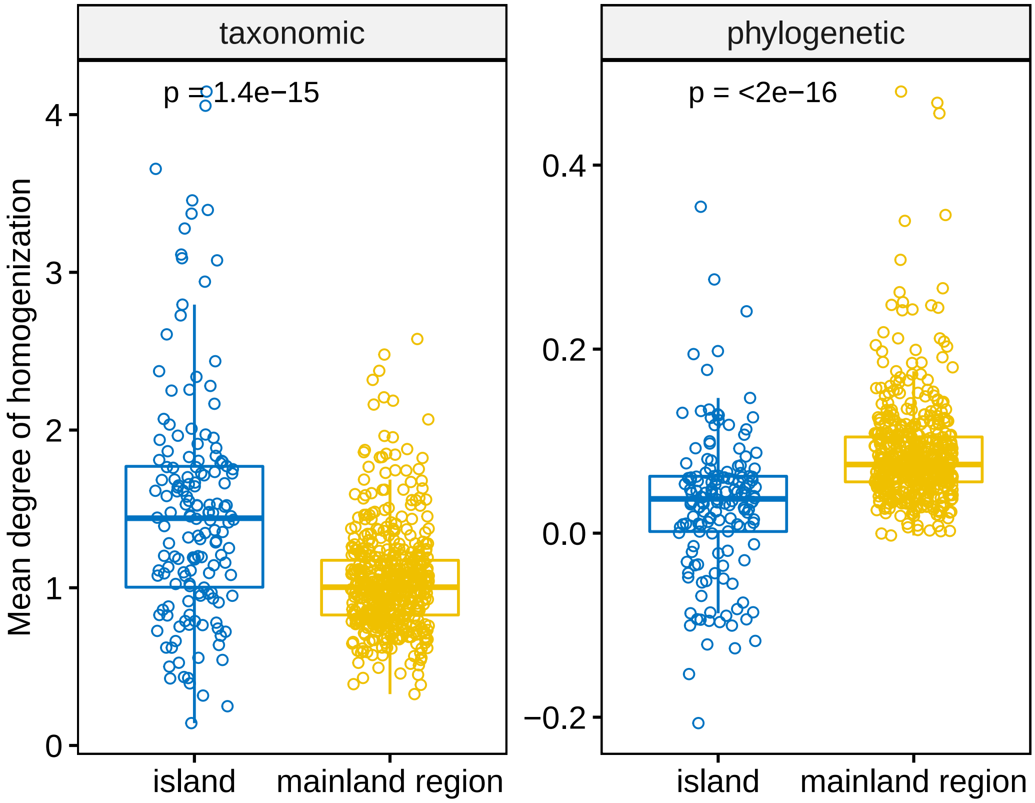
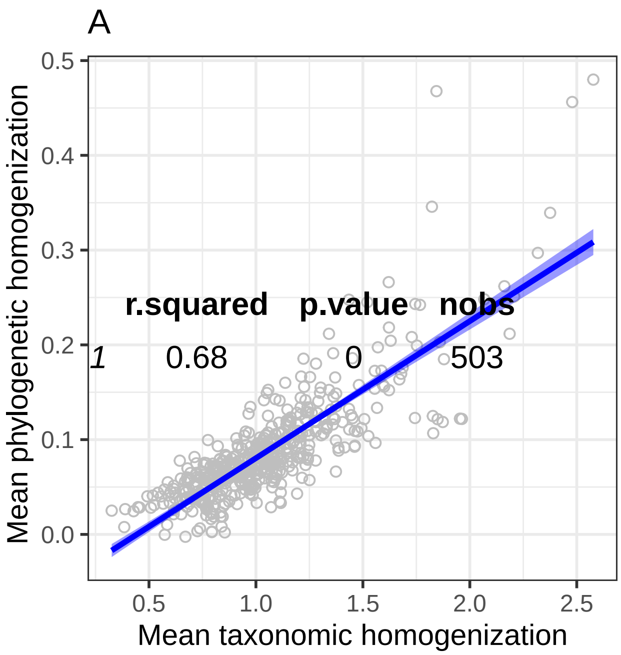

taxonomic
phylogenetic
p = 1.4e−15
p = <2e−16
Mean degree of homogenization
0
1
2
3
4
−0.2
0.0
0.2
0.4
island
mainland region
island
mainland region

Fig. S8

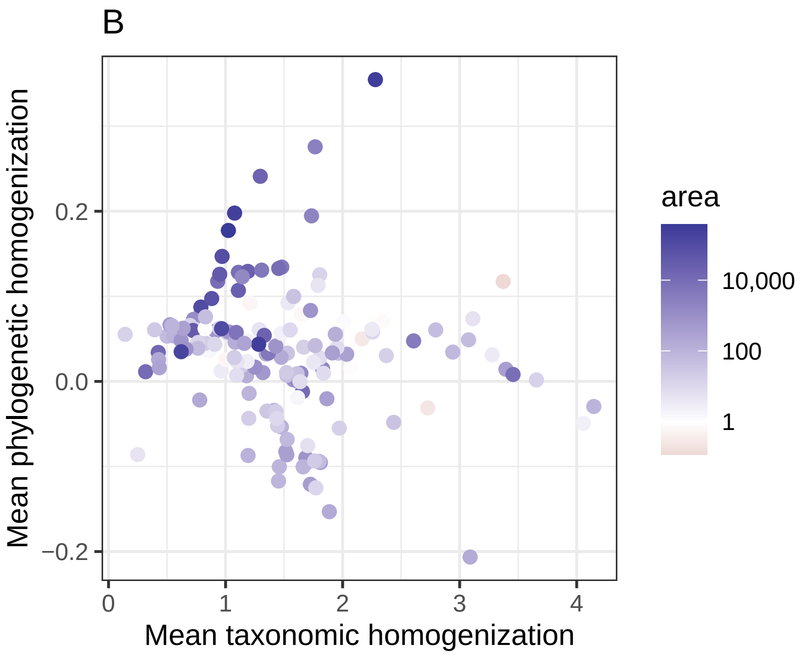
Fig. S8A

```r
load("df.average.Rdata")
load("df_island.Rdata")
# add area data
df.average %<>% left_join(df_island) %>% filter(new_region_id != 647)
# add up areas if there are more codes belonging to one new_region_id
df.average %<>%
  group_by(new_region_id, island, mean.homo.id, mean.homo.phylo) %>%
  summarise(GeodAREA = sum(GeodAREA)) %>%
  ungroup()
df_homo <- df.average
# correlation coefficient for mainland
df.mainland <- df_homo %>%
  filter(island == 0) %>%
  dplyr::select(mean.homo.id, mean.homo.phylo)
lm.mainland <- lm(mean.homo.phylo ~ mean.homo.id, data = df.mainland)
mod.performance <- lm.mainland %>%
  glance() %>%
  select(r.squared, p.value, nobs)
df_homo %>%
  filter(island == 0) %>%
  ggplot(., aes(x = mean.homo.id, y = mean.homo.phylo)) +
  geom_point(shape = 1, color = "gray") +
  geom_smooth(method = "lm", color = "blue", fill = "blue") +
  theme_bw() +
  xlab("Mean taxonomic homogenization") +
  ylab("\n Mean phylogenetic homogenization") +
  ggtitle("A")
mod.performance %>%
  round(digits = 2) %>%
  grid.table(theme = ttheme_minimal())
```



the scatter plot in Fig. S8B

```
df.island <- df_homo %>% filter(island == 1)
df.island %<>% mutate(log10_area = log10(GeodAREA))
ggplot() +
  geom_point(
    data = df.island,
    aes(x = mean.homo.id, y = mean.homo.phylo, color = log10_area), cex = 2
  ) +
  theme_bw() +
  scale_color_gradient2(
    name = "area",
    breaks = c(0, 2, 4, 6),
    labels = c(1, 100, "10,000", "1,000,000")
  ) +
  xlab("Mean taxonomic homogenization") +
  ylab("\nMean phylogenetic homogenization") +
  ggtitle("B")
```



the effect size plot in Fig. S8B

```
df.island.2 <- df.island %>% mutate_at(c("mean.homo.id", "log10_area"), scale)
mod.island <- lm(mean.homo.phylo ~ mean.homo.id + log10_area + mean.homo.id:log10_area,
  data = df.island.2
)
df_mod_island <- summary(mod.island)$coefficients
df_mod_island %<>%
  as.data.frame() %>%
  mutate(
    explainatory.variable =
      c("Intercept", "taxonomic", " log10(area)", "log10(area):taxonomic")
  ) %>%
  filter(explainatory.variable != "Intercept") %>%
  mutate(
    explainatory.variable =
      fct_relevel(
        explainatory.variable,
        c(" log10(area)", "taxonomic", "log10(area):taxonomic")
      )
  ) %>%
  arrange(explainatory.variable) %>%
  mutate(LEVEL = c(3, 2, 1))
ggplot(data = df_mod_island, aes(x = Estimate, y = LEVEL)) +
  geom_point() +
  geom_vline(xintercept = 0, col = "red", size = 0.5) +
  annotate("text",
    size = 4, x = df_mod_island$Estimate, y = df_mod_island$LEVEL + 0.45,
    label = df_mod_island$explainatory.variable
  ) +
  geom_errorbar(aes(
    xmin = Estimate - 1.96 * `Std. Error`,
    xmax = Estimate + 1.96 * `Std. Error`
  ), width = 0.2) +
  theme_classic() +
  xlab("Model estimates (±1.96*SE)") +
  ylim(c(0.7, 4)) +
  xlim(-0.052, 0.06) +
  theme(
    axis.title.y = element_blank(),
    axis.line.y = element_blank(),
    axis.text.y = element_blank()
  )
```

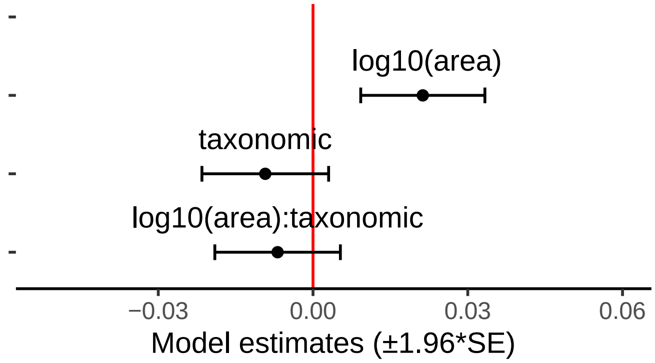


Fig. S9

```r
load(file = "wrld.rgional.level.similarity.Rdata")
load(file = "small.island.centroids.similarity.Rdata")
load(file = "wrld.sf.Rdata")
wrld.regional.level %<>%
  mutate(mean.sim.id.group = paste("<", mean.sim.id.group))
wrld.regional.level %<>%
  mutate(mean.sim.phylo.group = paste("<", mean.sim.phylo.group))
small.island.centroids %<>%
  mutate(mean.sim.id.group = paste("<", mean.sim.id.group))
small.island.centroids %<>%
  mutate(mean.sim.phylo.group = paste("<", mean.sim.phylo.group))
colors1 <- rev(magma(7))
colors2 <- colors1
scale_manual_id <- list(
  scale_fill_manual(values = colors1, drop = FALSE),
  scale_color_manual(values = colors1, drop = FALSE)
)
p.mean.id <-
  ggplot() +
  geom_sf(data = wrld.sf, fill = "gray", color = "gray") +
  geom_sf(data = wrld.regional.level %>% filter(island == 0),
          aes(fill = mean.sim.id.group), color = NA) +
  geom_sf(data = wrld.regional.level %>% filter(island == 1),
          aes(fill = mean.sim.id.group), color = NA) +
  geom_sf(data = small.island.centroids,
          aes(color = mean.sim.id.group), shape = 1) +
  coord_sf(crs = "+proj=wag4 +lon_0=0 +datum=WGS84 +units=m +no_defs") +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    legend.position = "right",
    plot.title = element_text(face = "bold")
  ) +
  scale_manual_id +
  labs(title = "a")
scale_manual_phylo <- list(
  scale_fill_manual(values = colors2, drop = FALSE),
  scale_color_manual(values = colors2, drop = FALSE)
)
p.mean.phylo <-
  ggplot() +
  geom_sf(data = wrld.sf, fill = "gray", color = "gray") +
  geom_sf(data = wrld.regional.level %>% filter(island == 0),
          aes(fill = mean.sim.phylo.group), color = NA) +
  geom_sf(data = wrld.regional.level %>% filter(island == 1),
          aes(fill = mean.sim.phylo.group), color = NA) +
  geom_sf(data = small.island.centroids,
          aes(color = mean.sim.phylo.group), shape = 1) +
  coord_sf(crs = "+proj=wag4 +lon_0=0 +datum=WGS84 +units=m +no_defs") +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    legend.position = "right",
    plot.title = element_text(face = "bold")
  ) +
  scale_manual_phylo +
  labs(title = "b")
grid.arrange(p.mean.id, p.mean.phylo, nrow = 2)
```

Fig. S10

```r
load(file = "df.average.similarity.Rdata")
df <- df.average %>%
  dplyr::select(new_region_id, mean.sim.id, mean.sim.phylo, island) %>%
  rename(taxonomic = mean.sim.id, phylogenetic = mean.sim.phylo) %>%
  gather(key = "Group", value = "value", taxonomic, phylogenetic)
df %<>% mutate(Group = fct_relevel(Group, c("taxonomic", "phylogenetic")))
df %<>% mutate(region = ifelse(island == 1, "island", "mainland region"))
p <- ggboxplot(df,
  x = "region", y = "value",
  color = "region", palette = "jco",
  add = "jitter", outlier.shape = NA, add.params = list(shape = 1),
  facet.by = "Group", short.panel.labs = FALSE
)
# Use only p.format as label. Remove method name.
p <- p + stat_compare_means(label = "p.format")
p <- facet(p, facet.by = "Group", scales = "free_y")
p + ylab("Mean native similarity") + xlab("") +
  theme(
    strip.text = element_text(size = 12),
    legend.position = "none"
  )
```

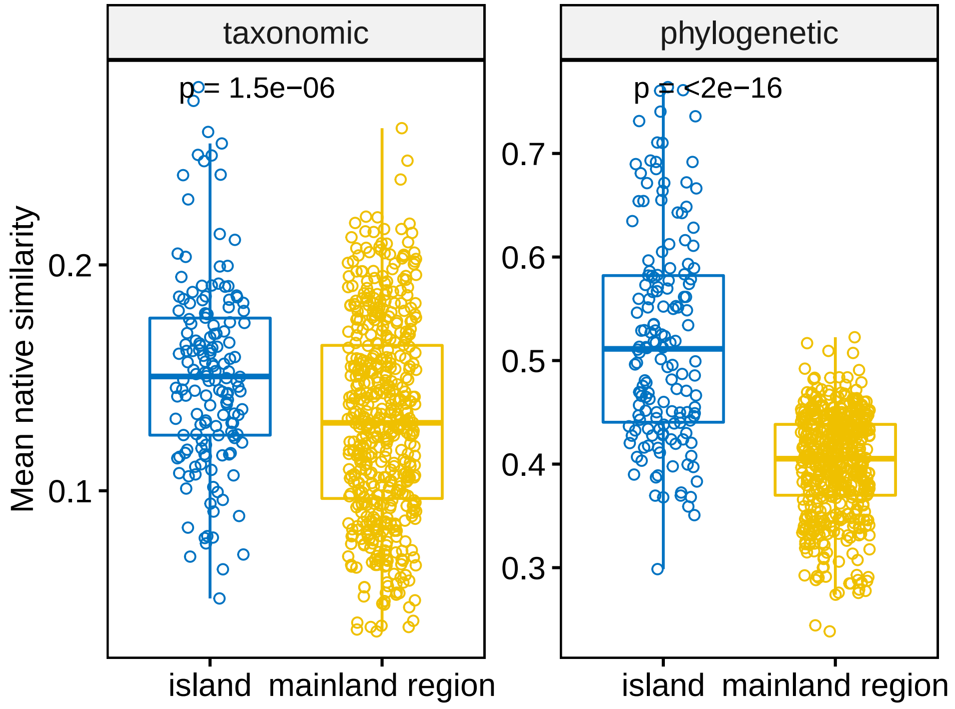


Fig. S11