Supporting Information: Exploring deep-time relationships between cultural and genetic evolution in Northeast Asia

December 20, 2018

Contents

List of Tables

List of Figures

S1 Packages and functions

S1.1 Packages

For the analysis we use the following three main packages:

- `ade4` provides tools for multivariate data analysis
- `adespatial` provides tools for multiscale spatial analysis of multivariate data
- `vegan` provides tools for ordination methods and diversity analysis
- `FactoMinor` provides tools for dimensionality reduction of mixed data (categorical and continous)

```
# Data analysis (main)
library(ade4)
library(vegan)
library(adespatial)
library(FactoMineR)

# Data analysis (additional)
library(missMDA)
library(ape)

# Data handling and manipulation
library(dplyr)
library(dendextend)
library(broom)
library(reshape2)
library(plyr)

# Plotting and knitting
library(knitr)
library(kableExtra)
library(ggplot2)
library(ggpolypath)
library(cowplot)
library(ggridges)
library(RColorBrewer)

# Spatial Analysis and mapping
library(sp)
library(spdep)
library(rgdal)
library(rgeos)
library(mapproj)

# Wrapper for running  SplitsTree from whithin R
#install_github('IVS-UZH/RSplitsTree')
library(RSplitsTree)
```

S1.2 Functions

In this section we document all custom-defined functions for performing the redundancy analysis and for data preprocessing.

S1.2.1 Redundancy Analysis

```
# Distance-based Moran's Eigenvector (dbMEM) Analysis
random_points_to_dbmem <- function(r_points, print_count=FALSE){
#' This function computes dbMEMs for each random point sample in r_points
#' @param r_points: the random points (SpatialPointsDataFrame)
#' @param print_count: Print the number of processed samples when iterating over r_points?
#' @return a list comprising the dbMEMs for each sample

  if (class(r_points)[1] != "SpatialPointsDataFrame") {
    stop("please provide a SpatialPointsDataFrame")
  }
  n_sample <- unique(r_points$sample_id)
  #n_sample <- max(r_points$sample_id)

  epsilon <- 0.1
  geo_pco  <- list()

  for (j in n_sample) {

      # Get all points of sample j
      points <- r_points[r_points$sample_id == j,]

      # Compute distances between all points in the sample
      mat <- spDists(points, points)

      # Compute the mst, find its longest edge and use as a threshold
      mst_1 <- spantree(mat)
      mst_le <- max(mst_1$dist)

      # Add a small epsilon (for numerical stability)
      thresh <- mst_le + epsilon

      # Find all nearest neighbors within the distance threshold
      nb <- dnearneigh(points, 0, thresh)

      # Normalize the data
      spwt <- lapply(nbdists(nb, points), function(x) 1 - (x/(4 * thresh))^2)

      # Compute weighted neighbor list
      lw <- nb2listw(nb, style = "B", glist = spwt, zero.policy = TRUE)

      # Compute MEMs with a corresponding positive autocorrelation
      res <- as.data.frame(scores.listw(lw, MEM.autocor = "positive"))

      rownames(res) <- points$nam_label
      colnames(res) <- paste("geo_pco_", seq(1,ncol(res)), sep="")
      res <- res[order(rownames(res)), drop = FALSE, ]
      geo_pco[[paste('sample_', j, sep="")]] <- res

      if (print_count) {
        if (j%%1000 == 0) {
          print(paste(j, " samples processed"))}}}
```

```r
  return (geo_pco)}

# Redundancy Analysis
rda_wrapper <- function (response, explanatory, random_geo_pco=NULL,
                         n_perm=100, indi_lang=F, print_count=FALSE) {
#' This function performs a spatially constrained RDA
#' @param response: the repsonse variable
#' @param explanatory: the explanatory variable
#' @param random_geo_pco dbMEMs of random spatial point patterns
#' @param n_perm number of permutations per random geo sample
#' @param ind_lang when TRUE sample only languages from different families
#' @param print_count: print the number of processed samples when iterating over r_points?
#' @return a list comprising the rda results for each sample

  if (is.null(rownames(response)) & is.null(rownames(explanatory))) {
    stop("Row names of the response and the explanatory variable must be defined!")
  }

  if (any(rownames(response) != rownames(explanatory))) {
    stop("Row names of the response and the
         explanatory variable must be identical and match in order!")
  }

  ex_name <- sub('\\_.*', '', colnames(explanatory)[1])
  re_name <- sub('\\_.*', '', colnames(response)[1])

  rda_results <- list()
  progress = 0

  for (sample in names(geo_mem)) {
    progress = progress + 1
    print(progress)

    constraint <- random_geo_pco[[sample]]

    if (indi_lang){

    #  Sample one language from each language family with two members
    sample_lgs <- c(single_lgs, sample(uralic,1), sample(chkkat, 1), sample(tungus, 1))
    explanatory_sample = explanatory[rownames(explanatory) %in% sample_lgs, , drop=F]
    response_sample = response[rownames(response) %in% sample_lgs, , drop=F]
    constraint_sample = constraint[rownames(constraint) %in% sample_lgs, ,drop=F]

    rda <- rda(X = explanatory_sample, Y = response_sample,
               Z = constraint_sample)

    # Compute the (adjusted) explained variance
    r2_res = RsquareAdj2_part(rda)

    r2_semi <- r2_res$r_squared_a
    r2_partial <- r2_res$r_squared_b
    r2_adj_semi <- r2_res$adj_r_squared_a
    r2_adj_partial <- r2_res$adj_r_squared_b
```

```
    # Perform permutations
    perm <- permute_rda(n_perm, explanatory_sample, response_sample, constraint_sample)}

    else {
      rda <- rda(X = explanatory, Y = response, Z = constraint)

      # Compute the (adjusted) explained variance
      r2_res = RsquareAdj2_part(rda)

      r2_semi <- r2_res$r_squared_a
      r2_partial <- r2_res$r_squared_b
      r2_adj_semi <- r2_res$adj_r_squared_a
      r2_adj_partial <- r2_res$adj_r_squared_b

      # Perform permutations
      perm <- permute_rda(n_perm, explanatory, response, constraint)}

    rda_results[[sample]] <- list(r2_semi=r2_semi, r2_adj_semi=r2_adj_semi,
                                  r2_partial=r2_partial, r2_adj_partial=r2_adj_partial,
                                  perm_r2_semi=perm$r2_semi,
                                  perm_r2_adj_semi=perm$r2_adj_semi,
                                  perm_r2_partial=perm$r2_partial,
                                  perm_r2_adj_partial=perm$r2_adj_partial,
                                  explanatory=ex_name, response=re_name, geo=TRUE)}
   if (print_count){
    if (i%%100 == 0) {
      print(paste(i, " samples processed"))}}
 return(rda_results)}

# Permute RDA
permute_rda <- function(n_perm, explanatory, response, constraint=NULL) {
#' This function runs n_perm RDAs with permuted data
#' @param n_perm: the number of permutations
#' @param response: the repsonse variable
#' @param explanatory: the explanatory variable
#' @param constraint the constraint
#' @return a list comprising the RDA results for each permutation

  if (!is.numeric(n_perm)){stop("The number of permutations must
                                be .. well... a number.")}
  permutation_results <- data.frame(r2_adj=rep(NA, n_perm))

  for (i in 1:n_perm){
    # permute the response
    permutation_order <- sample(1:nrow(response))
    response <- response[permutation_order, , drop=F]

    # Geo-constraint?
    if (is.null(constraint)) {rda <- rda(X = explanatory, Y = response)}
    else {rda <- rda(X = explanatory, Y = response, Z = constraint)}

     # Compute the (adjusted) explained variance
    r2_res = RsquareAdj2_part(rda)
```

```
    r2_semi <- r2_res$r_squared_a
    r2_partial <- r2_res$r_squared_b
    r2_adj_semi <- r2_res$adj_r_squared_a
    r2_adj_partial <- r2_res$adj_r_squared_b

    permutation_results[i, c("r2_semi")] <- r2_semi
    permutation_results[i, c("r2_partial")] <- r2_partial
    permutation_results[i, c("r2_adj_semi")] <- r2_adj_semi
    permutation_results[i, c("r2_adj_partial")] <- r2_adj_partial}

  return(permutation_results)}

# Compute adjusted R-squared
RsquareAdj2_part <- function (x) {

  #' This function computes the (adjusted) R-squared of an RDA model using either
  #' - vegan's semipartial method
  #' - CONOCO's partial method
  #' code adapted from:
  #' https://davidzeleny.net/blog/2016/09/08/adjusted-r2-in-partial-constrained-
  #' ordination-the-difference-between-r-vegan-and-canoco-5/
  #' @param x: RDA result
  #' @return a list comprising the R-squared / adjusted R-squared

  m <- x$CCA$qrank
  n <- nrow(x$CCA$u)
  R2_a <- x$CCA$tot.chi/x$tot.chi
  R2p_a <- x$pCCA$tot.chi/x$tot.chi
  p_a <- x$pCCA$rank
  radj_a <- RsquareAdj(R2_a + R2p_a, n, m + p_a) - RsquareAdj(R2p_a, n, p_a)

  R2_b <- x$CCA$tot.chi/(x$tot.chi - x$pCCA$tot.chi)
  p_b <- x$pCCA$rank
  radj_b <- 1 - (1 - R2_b)*(n - p_b - 1)/(n - m - p_b - 1)

  if (any(na <- m >= n - 1)) radj_b[na] <- NA

  return (list(r_squared_a = R2_a, adj_r_squared_a = radj_a, r_squared_b = R2_b,
               adj_r_squared_b = radj_b))}
```

S1.2.2 Helper functions for data preprocessing

```
trim_data <- function(data.list, trim.to=siberia_metadata_all, extra.coverage=.8) {
 #' This function trims the linguistic input data and only keeps variables with
 #' data points for languages in the study area and with non-constant values

 #' @param data.list the data to be trimmed
 #' @param trim.to contains the ids of those languages that are retained after trimming
 #' @param extra.coverage the percentage of covered data
 #' @return the trimmed data
```

```
  lgs <- lapply(data.list, function(l) {
      l$UULID <- ifelse(l$isocode %in% c('bxm','bxr'),
                '[i-bua][a-1095][g-buri1258]',
               paste(l$UULID))
      subset(l, UULID %in% trim.to$UULID)
     })
  vars <- lgs[sapply(lgs, function(l) {
       length(l$UULID)==length(unique(l$UULID)) &
       length(unique(l[,1]))>1 &
         length(unique(l$UULID)) >= floor(extra.coverage*length(trim.to$UULID))
       })]
  return(lapply(vars, function(l) l[,c(1,3)]))}


compute_coverage <- function(data.list, gg=siberia_metadata) {
 #' This function computes the coverage of all languages
 #' @param data.list the input data
 #' @param gg a data.frame comprising the languages for which the coverage is computed
 #' @return the input data with the coverage added as a separate column

  x <- sapply(gg$UULID, function(l) {
          coverage <- round(mean(sapply(data.list, function(v) { l %in% v$UULID })),2)*100
        })
  df <- data.frame(UULID=names(x), Coverage=x)
  gg$Language <- rownames(gg)
  df.g <- merge(df, gg)
  return(df.g)}


flatten <- function(data.list, gg=siberia_metadata) {
 #' This function flattens the nested linguistic data
 #' @param data.list: the input data
 #' @param gg: a data.frame comprising the languages for which the data are flattened
 #' @return the flattened data

  df.list <- lapply(seq_along(data.list), function(v) {
    df <- data.list[[v]]
    var.name <- gsub('(.*\\$)', '\\2', names(data.list)[v])
    names(df)[1] <- var.name
    return(dplyr::select(df, UULID, dplyr::everything()))
    })
  df.flat <- Reduce(function(x,y) dplyr::full_join(x, y, by='UULID'), df.list)
  rownames(df.flat) <- sapply(df.flat$UULID, function(x) rownames(gg[gg$UULID %in% x,]))
  return(df.flat)}


print_dist <- function(d, caption) {
  #' This function prints the distance matrices in a table
  #' @param d: the input distance matrix
  #' @param caption: the caption of the table
  d.m <- as.matrix(sort_dist_mat(d))
  d.m[upper.tri(d.m, diag=T)] <- NA
  colnames(d.m) <- abbreviate(colnames(d.m), minlength=8)
```

```
  rownames(d.m) <- abbreviate(rownames(d.m), minlength=8)
  options(knitr.kable.NA = '')
  d.m <- d.m[2:nrow(d.m),1:ncol(d.m)-1]

  kable(d.m, digits=3, format = 'latex', caption=caption) %>%
    kable_styling(latex_options = c("scale_down", "hold_position")) %>%
                  column_spec(1, border_left=T) %>%
                  column_spec(ncol(d.m)+1, border_right=T)}


rda_to_z_val <- function(rda_result, r2_type) {
  #' This function first flattens the RDA results and then computes the z-value for the
  #' difference between observed and permuted R-squared
  #' @param rda_results: the results from the RDA analysis
  #' @param r_2_type: the R-squared type that should be retained (r2_semi, r2_adj_semi,
  #'                  r2_partial, r2_adj_partial)
  #' @return a list comprising the flattened values and the z-standardized values

  perm_type = paste("perm_", r2_type, sep="")
  rda_distributions <- lapply(rda_result, function(x)
  unlist(sapply(x, function(y)
    y[[r2_type]] - mean(y[[perm_type]]))))

  rda_df <- ldply(rda_distributions, data.frame)
  colnames(rda_df) <- c("Association", "Values")

  rda_zs <- lapply(rda_result, function(x)
  unlist(sapply(x, function(y)
    (y[[r2_type]] - mean(y[[perm_type]]))/sd(y[[perm_type]]))))
  rda_df_z <- ldply(rda_zs, data.frame)

  colnames(rda_df_z) <- c("Association", "z")
  return(list(val=rda_df, z_val=rda_df_z))}


add_proportion_rda <- function(rda_flat, sig_diff){
  #' This function adds the proportions of significant locations
  #' to the flattened RDA results
  #' @param rda_flat: the flattened RDA results
  #' @param sig_diff: the threshold value for significance

  # Which proportion of points is larger than sig_diff?
  proportion <- sapply(unique(rda_flat$z_val$Association), function(x) {

    sum(rda_flat$z_val$z[rda_flat$z_val$Association==x] >= sig_diff)/
      sum(rda_flat$z_val$Association==x) })

  # Add labels (for plotting)
  assoc_labels <- as.vector(sapply(unique(rda_flat$z_val$Association), function (s) {
    s_split <- strsplit(s[1], "_")[[1]]
    s_label <- s_split
    label = paste(s_split[1], "\u2192", s_split[2])
  }, simplify = TRUE))
```

```
  prop_df <- data.frame(Association=assoc_labels, Proportion=proportion)

  mean_associations <- ddply(rda_flat$val, "Association", function(x) mean(x$Values))
  rda_flat$val$Association <- factor(rda_flat$val$Association,
                        levels=mean_associations$Association[order(mean_associations$V1)])

  rda_flat$val$sig_loc <-  sapply(rda_flat$val$Association,
                                  function(x) proportion[as.character(x)])

  return(list(proportion = prop_df, rda_df=rda_flat))}
```

S2 Data

S2.1 Lexical data

We load Levenshtein distances derived from the ASJP data (Wichmann et al. 2015):

```
lex_dist <- sort_dist_mat(as.dist(
  read.csv('data/lexicon/ASJP12PopDist.csv', header=T, row.names = 1)))

attr(lex_dist, 'Labels')[10] <- 'West Greenlandic'
```

S2.2 Genetics and Music

We read the distance matrices for genetics and music. Table S1 shows how these data are matched to unique identifiers in the language data. The UULID concatenates IDs from the ISO 639.3 standard ("i-"), from the AUTOTYP (Bickel et al. 2017) database ("a-"), and from the GLOTTOLOG (Hammarström, Forkel, and Haspelmath 2017) catalogue ("g-").

```
# Read the gentic data
genetics <- read.csv("data/genetics/SNPs13PopDist.csv",sep=",",
                     header=TRUE, row.names=1)

# Read the music data
music <- read.csv("data/music/MusicAll13PopDist.csv",sep=",",
                  header=TRUE, row.names=1)
```

Table S1: Music and genome-wide SNP data, with sample sizes, references and match to languages.

Population	Music	SNPs	SNP Sources	Language	UULID
Korean	30	6	Lazaridis et al. (2014)	Korean	[i-kor][a-141][g-kore1280]
Japanese (Mainland)	30	45	Abecasis et al. (2012)	Japanese	[i-jpn][a-118][g-nucl1643]
Ainu (Hokkaido)	30	36	Jinam et al. (2012)	Ainu	[i-ain][a-12][g-ainu1240]
Koryak	30	27	Rasmussen et al. (2010)	Koryak	[i-kpy][a-1808][g-kory1246]

Population	Music	SNPs	SNP Sources	Language	UULID
Chukchi	14	20	Lazaridis et al. (2014)	Chukchi	[i-ckt][a-56][g-chuk1273]
Yakut	8	21	Lazaridis et al. (2014)	Yakut	[i-sah][a-2662][g-yaku1245]
Even	17	18	Lazaridis et al. (2014), Fedorova et al. (2013)	Even	[i-eve][a-738][g-even1260]
Yukagir	12	9	Lazaridis et al. (2014)	Yukagir (Tundra)	[i-yux][a-2797][g-sout2750]
Evenk	28	16	Rasmussen et al. (2010)	Evenki	[i-evn][a-527][g-even1259]
Buryat	30	19	Rasmussen et al. (2010)	Buriat	[i-bua][a-1095][g-buri1258]
West Greenlandic (Inuit)	8	10	Rasmussen et al. (2010)	West Greenlandic	[i-kal][a-511][g-kala1399]
Selkup	12	20	Rasmussen et al. (2010), Lazaridis et al. (2014)	Selkup	[i-sel][a-2393][g-selk1253]
Nganasan	15	15	Rasmussen et al. (2010)	Naganasan	[i-nio][a-2172][g-ngan1291]

S2.3 Grammar and Phonology

Our data comprise numerical and categorical variables for grammar and phonology, distance matrices for genetics and music, and the geographical locations of peoples in Northern Asia and Greenland in the form of language polygons.

Data on grammar and phonology are aggregated from the following sources:

- AUTOTYP (Bickel et al. 2017)
- WALS (Dryer and Haspelmath 2013), enriched by recodings (Bickel and Zakharko 2018)
- ANU Phonotactics database (Donohue et al. 2013)
- PHOIBLE (Moran, McCloy, and Wright 2014)

The geographical polygon locations of the languages are taken from the Ethnologue (Simons and Fennig 2018); as there is no polygon for Ainu, we drew one ourselves.

The split into phonology vs. grammar is based on the broad definition in `categorization_of_variables.csv` which includes phonotactic and morphophonological data under phonology.[1] We extract a subset from the data comprising languages from thirteen different sites. Note that in AUTOTYP and WALS, Buriat is represented by ISO code `bua` (Buriat in general), while in PHOIBLE it is represented by ISO code `bxr` and in the ANU data by `bxm`. We map all of them below to `bua`.

```
# Define single languages and family-related languages
single_lgs <- c("Ainu", "Buryat", "Japanese", "Korean", "West Greenlandic",
                "Yakut", "Yukagir")
uralic <- c("Selkup", "Nganasan")
chkkat <- c("Chukchi", "Koryak")
```

[1] Available at https://www.geo.uzh.ch/microsite/MusicGenesLanguages

```
tungus <- c("Even", "Evenki")

# Read categorization of variables
var_categories <- read.csv("data/typology/categorization_of_variables.csv",
                           stringsAsFactors = F)


# Read the grammar, phonology and typlogy data (meta data)
typology.list <- readRDS("data/typology/typology.list.RDS")
phonological_vars <- unlist(as.character(sapply(typology.list, function(x) {
        var_categories[var_categories$Variable == x[1, "variable.ID"],
                       "Broad_Phon_Definition_Binary"] == "Phonology" }) ))
phonology_list <- typology.list[phonological_vars == "TRUE"]
grammar_vars <- unlist(as.character(sapply(typology.list, function(x) {
        var_categories[var_categories$Variable == x[1, "variable.ID"],
                       "Broad_Phon_Definition_Binary"] == "Grammar" })))
grammar_list <- typology.list[grammar_vars == "TRUE"]
typology_coverage_df <- readRDS("data/typology/typology.coverage.RDS")

# Define subset
siberia_sample <- c(
 "[i-ain][a-12][g-ainu1240]",    # Ainu
 "[i-bua][a-1095][g-buri1258]",  # Buriat
 "[i-bxm][a-][g-mong1330]",      # Buriat (Mongolia)
 "[i-bxr][a-][g-russ1264]",      # Buriat (Russia)
 "[i-ckt][a-56][g-chuk1273]",    # Chukchi
 "[i-eve][a-738][g-even1260]",   # Even
 "[i-evn][a-527][g-even1259]",   # Evenki
 "[i-kal][a-511][g-kala1399]",   # West Greenlandic
 "[i-jpn][a-118][g-nucl1643]",   # Japanese
 "[i-kor][a-141][g-kore1280]",   # Korean
 "[i-kpy][a-1808][g-kory1246]",  # Koryak
 "[i-nio][a-2172][g-ngan1291]",  # Nganasan
 "[i-sel][a-2393][g-selk1253]",  # Selkup
 "[i-sah][a-2662][g-yaku1245]",  # Yakut
 "[i-ykg][a-423][g-nort2745]")   # Yukagir (Tundra)

# Extract meta data for the above subset
siberia_metadata_all <- subset(typology_coverage_df, UULID %in% siberia_sample)
siberia_metadata <- subset(siberia_metadata_all, !isocode %in% c('bxm','bxr'))

counts <- xtabs(~autotyp.Stock, siberia_metadata, drop.unused.levels = T)
rownames(siberia_metadata) <- with(siberia_metadata,
                                   ifelse(autotyp.Stock %in% names(counts[counts>1]),
                                          paste(autotyp.Stock, autotyp.Language, sep="/"),
                                          paste(autotyp.Language)))

rownames(siberia_metadata) <- gsub('Yukagir/','', rownames(siberia_metadata))
```

We only use variables with one data point per language, and only variables with non-constant values (which otherwise can't deliver a distance signal). At the same time, we also remap `bxr` and `bxm` to `bua` (cf. above). We trim the linguistic data accordingly.

```r
# Trim the data
siberia_grammar_list <- trim_data(grammar_list,extra.coverage = 0.8)
siberia_phonology_list <- trim_data(phonology_list,extra.coverage = 0.8)
```

The following lists the phonological variables we captured. For full definitions and descriptions of the variables, see the source databases listed above.

```
## - WALS$FrontRndV.Presence
## - WALS$GlottalizedC.Presence
## - WALS$Laterals.Presence
## - WALS$Uvulars.Presence
## - WALS$VoicingC.Presence
## - WALS$11A Front Rounded Vowels
## - WALS$19A Presence of Uncommon Consonants
## - WALS$1A Consonant Inventories
## - WALS$2A Vowel Quality Inventories
## - WALS$3A Consonant-Vowel Ratio
## - WALS$4A Voicing in Plosives and Fricatives
## - WALS$6A Uvular Consonants
## - WALS$7A Glottalized Consonants
## - WALS$8A Lateral Consonants
## - PHOIBLE$syllabic_count
## - PHOIBLE$short_count
## - PHOIBLE$long_count
## - PHOIBLE$consonantal_count
## - PHOIBLE$sonorant_count
## - PHOIBLE$continuant_count
## - PHOIBLE$delayedRelease_count
## - PHOIBLE$approximant_count
## - PHOIBLE$trill_count
## - PHOIBLE$nasal_count
## - PHOIBLE$lateral_count
## - PHOIBLE$labial_count
## - PHOIBLE$round_count
## - PHOIBLE$labiodental_count
## - PHOIBLE$coronal_count
## - PHOIBLE$anterior_count
## - PHOIBLE$distributed_count
## - PHOIBLE$strident_count
## - PHOIBLE$dorsal_count
## - PHOIBLE$high_count
## - PHOIBLE$low_count
## - PHOIBLE$front_count
## - PHOIBLE$back_count
## - PHOIBLE$tense_count
## - PHOIBLE$retractedTongueRoot_count
## - PHOIBLE$periodicGlottalSource_count
## - PHOIBLE$spreadGlottis_count
## - PHOIBLE$constrictedGlottis_count
## - PHOIBLE$vowels_count
## - PHOIBLE$vowels.syllabic.consonants_count
## - PHOIBLE$long.vowels_count
## - PHOIBLE$glides_count
## - PHOIBLE$liquids_count
```

```
## - PHOIBLE$nasals_count
## - PHOIBLE$fricatives_count
## - PHOIBLE$affricates_count
## - PHOIBLE$stops_count
## - PHOIBLE$stops.affricates_count
## - PHOIBLE$liquids.glides_count
## - PHOIBLE$liquids.glides.nasals_count
## - PHOIBLE$creaky.breathy_count
## - PHOIBLE$aspirated.fricatives_count
## - PHOIBLE$voiceless.stops_count
## - PHOIBLE$velar.nasals_count
## - PHOIBLE$short_presence
## - PHOIBLE$long_presence
## - PHOIBLE$trill_presence
## - PHOIBLE$lateral_presence
## - PHOIBLE$labiodental_presence
## - PHOIBLE$distributed_presence
## - PHOIBLE$back_presence
## - PHOIBLE$retractedTongueRoot_presence
## - PHOIBLE$spreadGlottis_presence
## - PHOIBLE$constrictedGlottis_presence
## - PHOIBLE$long.vowels_presence
## - PHOIBLE$liquids_presence
## - PHOIBLE$affricates_presence
## - PHOIBLE$creaky.breathy_presence
## - PHOIBLE$aspirated.fricatives_presence
## - PHOIBLE$velar.nasals_presence
## - ANU$CVC language
## - ANU$CCVC language
## - ANU$Onset second C is preferentially glide
## - ANU$Onset second C = y
## - ANU$Onset second C = w
## - ANU$Palatals/Affricates (ts, c etc.) allowed as coda
## - ANU$Glottal stops in onsets?
## - ANU$Glottal stops in codas?
## - ANU$Velar nasals in onsets?
## - ANU$Velar nasals in codas?
```

And these the grammar variables:

```
## - autotyp$NP_per_language$AdjAttrAgr.Presence
## - autotyp$NP_per_language$AdjAttrConstr.Presence
## - autotyp$NP_per_language$AdjAttrGvt.Presence
## - autotyp$NP_per_language$AdjAttrMarking.overt.Presence
## - autotyp$NP_per_language$NPAgr.Presence
## - autotyp$NP_per_language$NPConstr.Presence
## - autotyp$NP_per_language$NPGvt.Presence
## - autotyp$NP_per_language$NPMarking.overt.Presence
## - WALS$AdjNounOrder
## - WALS$CaseMarking.Presence
## - WALS$CaseMarkingTypes.v1
## - WALS$InflAffixPositions
## - WALS$NegationType
## - WALS$SOOrder
## - WALS$VInitialOrder
```

Table S2: Language data coverage for all thirteen sites.

Language	Grammar (%)	Phonology (%)
Buryat	100	83
Chukchi	100	100
Even	100	100
Evenki	100	100
Japanese	100	100
West Greenlandic	100	100
Korean	100	100
Koryak	100	100
Nganasan	100	100
Yakut	100	100
Yukagir	100	100
Ainu	95	100
Selkup	90	100

```
## - WALS$VerbMedialOrder
## - WALS$112A Negative Morphemes
## - WALS$26A Prefixing vs. Suffixing in Inflectional Morphology
## - WALS$51A Position of Case Affixes
## - WALS$69A Position of Tense-Aspect Affixes
## - WALS$87A Order of Adjective and Noun
```

For each site we compute the coverage, i.e. the percentage of available variables per site.

```
# Compute the coverage for each variable
grammar_coverage <- compute_coverage(siberia_grammar_list) %>%
  dplyr::select(Language, Coverage)
phonology_coverage <- compute_coverage(siberia_phonology_list) %>%
  dplyr::select(Language, Coverage)
```

We simplify and standardize the language names and visualize the coverage in a table. Finally, we flatten the nested linguistic data and convert them to data frames.

```
# Flatten the data
grammar <- flatten(siberia_grammar_list)
phonology <- flatten(siberia_phonology_list)
```

S2.4 Geographic locations

We import the language polygons and 15,000 samples of point locations taken randomly from these. The random point samples were generated in PostGIS with the function `ST_GeneratePoints` For further details see the SQL code `generate_random_point_samples.sql` at https://www.geo.uzh.ch/microsite/MusicGenesLanguages/.

```
# Fetch the language polygons
geo_polygons <- readRDS("data/geo/geo_polygons.RDS")

# Fetch random spatial points in the polygons
geo_random_points <- readRDS("data/geo/geo_random_points.RDS")
```

Since the data are gathered from different sources, the names used for the thirteen sites differ. We standardise

all names.

```
# list_A <- list(genetics, music, grammar, phonology)
# differing_names <- lapply(list_A, function(y)
#   lapply(list_A, function (x) setdiff(rownames(y), rownames(x))))

# We update all non-matching names using the names in geo_random_points as a template
# Genetics
colnames(genetics)[colnames(genetics) == 'westGreenland'] <- 'West Greenlandic'
rownames(genetics)[rownames(genetics) == 'westGreenland'] <- 'West Greenlandic'
colnames(genetics)[colnames(genetics) == 'Evenk'] <- 'Evenki'
rownames(genetics)[rownames(genetics) == 'Evenk'] <- 'Evenki'

# Music
colnames(music)[colnames(music) == 'WestGreenland'] <- 'West Greenlandic'
rownames(music)[rownames(music) == 'WestGreenland'] <- 'West Greenlandic'
colnames(music)[colnames(music) == 'Nganasa'] <- 'Nganasan'
rownames(music)[rownames(music) == 'Nganasa'] <- 'Nganasan'
colnames(music)[colnames(music) == 'Evenk'] <- 'Evenki'
rownames(music)[rownames(music) == 'Evenk'] <- 'Evenki'

# Grammar
rownames(grammar)[rownames(grammar) == 'Chukchi-Kamchatkan/Chukchi'] <-'Chukchi'
rownames(grammar)[rownames(grammar) == 'Tungusic/Evenki'] <- 'Evenki'
rownames(grammar)[rownames(grammar) == 'Greenlandic Eskimo (West)'] <-'West Greenlandic'
rownames(grammar)[rownames(grammar) == 'Uralic/Selkup'] <- 'Selkup'
rownames(grammar)[rownames(grammar) == 'Yukagir (Tundra)'] <- 'Yukagir'
rownames(grammar)[rownames(grammar) == 'Tungusic/Even'] <- 'Even'
rownames(grammar)[rownames(grammar) == 'Buriat'] <- 'Buryat'
rownames(grammar)[rownames(grammar) == 'Uralic/Nganasan'] <- 'Nganasan'
rownames(grammar)[rownames(grammar) == 'Chukchi-Kamchatkan/Koryak'] <- 'Koryak'

# Phonology
rownames(phonology)[rownames(phonology) == 'Chukchi-Kamchatkan/Chukchi'] <-'Chukchi'
rownames(phonology)[rownames(phonology) == 'Tungusic/Evenki'] <- 'Evenki'
rownames(phonology)[rownames(phonology) == 'Greenlandic Eskimo (West)'] <-'West Greenlandic'
rownames(phonology)[rownames(phonology) == 'Uralic/Selkup'] <- 'Selkup'
rownames(phonology)[rownames(phonology) == 'Yukagir (Tundra)'] <- 'Yukagir'
rownames(phonology)[rownames(phonology) == 'Tungusic/Even'] <- 'Even'
rownames(phonology)[rownames(phonology) == 'Buriat'] <- 'Buryat'
rownames(phonology)[rownames(phonology) == 'Uralic/Nganasan'] <- 'Nganasan'
rownames(phonology)[rownames(phonology) == 'Chukchi-Kamchatkan/Koryak'] <- 'Koryak'
```

S3 Dimensionality reduction

S3.1 Factorial analysis of mixed data (FAMD) of Grammar and Phonology

In view of the fact that the grammar and phonology data are partly numerical and partly categorical, we use a balanced mix of PCA and MCA (Lê, Josse, and Husson 2008). Empty values are imputed using the methods developed by Josse and Husson (2016).

```
# Impute empty values
grammar_imputed <- imputeFAMD(grammar[,-1])
```

```
phonology_imputed <- imputeFAMD(phonology[,-1])

# Perform FAMD
grammar_famd <- FAMD(grammar[,-1],
                     tab.comp=grammar_imputed$tab.disj,
                     ncp=10,
                     graph=F)

phonology_famd <- FAMD(phonology[,-1],
                       ncp=10,
                       tab.comp=phonology_imputed$tab.disj,
                       graph=F)
```

We rescale the dimensions obtained through FAMD in relation to the explained variance:

```
for(i in 1:ncol(phonology_famd$ind$coord)) {
  phonology_famd$ind$coord[,i] <-
    scale(phonology_famd$ind$coord[,i])*
    phonology_famd$eig[i,"percentage of variance"]}

for(i in 1:ncol(grammar_famd$ind$coord)) {
  grammar_famd$ind$coord[,i]<-
    scale(grammar_famd$ind$coord[,i])*
    grammar_famd$eig[i,"percentage of variance"]}
```

S3.2 Principal Coordinates Analysis (PCoA) of Music and Genes

We perform a principal coordinate analysis (PCoA) on the distance matrices for genetics and music. Similar to a PCA, a PCoA produces a set of orthogonal axes whose importance is measured by eigenvalues (Dray, Legendre, and Peres-Neto 2006). However, in contrast to the PCA, non-Euclidean distance matrices can be used. We correct for negative eigenvalues using the Cailliez procedure.

```
# Convert the matrices into dist objects
genetics_dist <- as.dist(genetics, diag = FALSE, upper = FALSE)
music_dist <- as.dist(music, diag = FALSE, upper = FALSE)


# Perform PCoA
genetics_pcoa <- pcoa(genetics_dist, correction = "cailliez")
music_pcoa <- pcoa(music_dist, correction = "cailliez")
```

We rescale the PCoA components in relation to the explained variance.

```
for(i in 1:ncol(genetics_pcoa$vectors)) {
genetics_pcoa$vectors[,i] <- scale(genetics_pcoa$vectors[,i])*
  genetics_pcoa$values$Rel_corr_eig[i]}

for(i in 1:ncol(music_pcoa$vectors)) {
music_pcoa$vectors[,i] <- scale(music_pcoa$vectors[,i])*
  music_pcoa$values$Rel_corr_eig[i]}
```

S3.3 Distance-based Moran's Eigenvector Map Analysis (dbMEM) of the spatial locations

We take 1,000 random point locations from the language polygons (as represented in Figure S1, corresponding to Figure 1 in the main text) and compute the spherical distance between them. Then we perform a distance-based Moran's eigenvector map analysis (`dbMEM`) where we decompose the spatial structure of each of the resulting 1,000 distance matrices (Borcard and Legendre 2002). Similar to a PCoA, dbMEM reveals the principal coordinates of the spatial locations from which the distance matrix was generated. However, in contrast to PCoA, dbMEM is primarily concerned with the interaction between spatial neighbours. Thus, only distances below a certain threshold feed directly into constructing the principal coordinates, whereas distances above the threshold are "truncated" (i.e. they are set to four times the threshold value). In the dbMEM, we use the length of the longest edge in the minimum spanning tree as a truncation threshold. Moreover, we only return those eigenfunctions that correspond to positive autocorrelation.

```
# Some random points are not complete. We remove them.
incomplete_samples <- as.data.frame(geo_random_points) %>%
  dplyr::group_by(sample_id) %>%
  dplyr::summarise (n=n()) %>%
  dplyr::filter(n!=13)

incomplete_samples <- as.vector(incomplete_samples$sample_id)
geo_random_points <- geo_random_points[!geo_random_points$sample_id %in% incomplete_samples, ]

# We take 1000 samples from the random points and compute dbMEMs for each sample
n_geo_samples = 1000
choose_p <- sample(unique(geo_random_points$sample_id), n_geo_samples, replace=F)
geo_mem <- random_points_to_dbmem(geo_random_points[geo_random_points$sample_id
                                                    %in% choose_p, ])
```

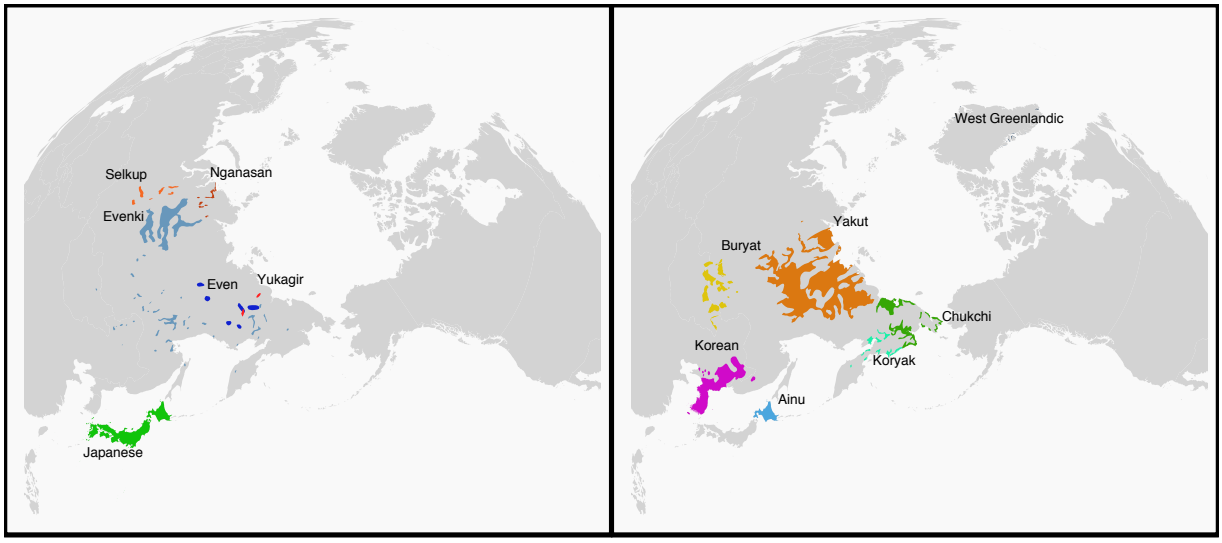

Figure S1: Geographical locations of the thirteen languages. Language polygons are plotted on two panels because they partly overlap in spatial distributions. Similarly-colored pairs of languages belong to the same family: Even and Evenki belong to the Tungusic family, Selkup and Nganasan to the Uralic family, and Koryak and Chukchi to the Chukotko-Kamchatkan family.

S3.4 Visualizing the explained variance

We visualize the results of the PCA and PCoA in a scree plot. The figures below show the fraction of total variance in the data as explained by each PC/PCo in decreasing order. We extract the eigenvalues from the PCos/PCs and visualize the explained variance.

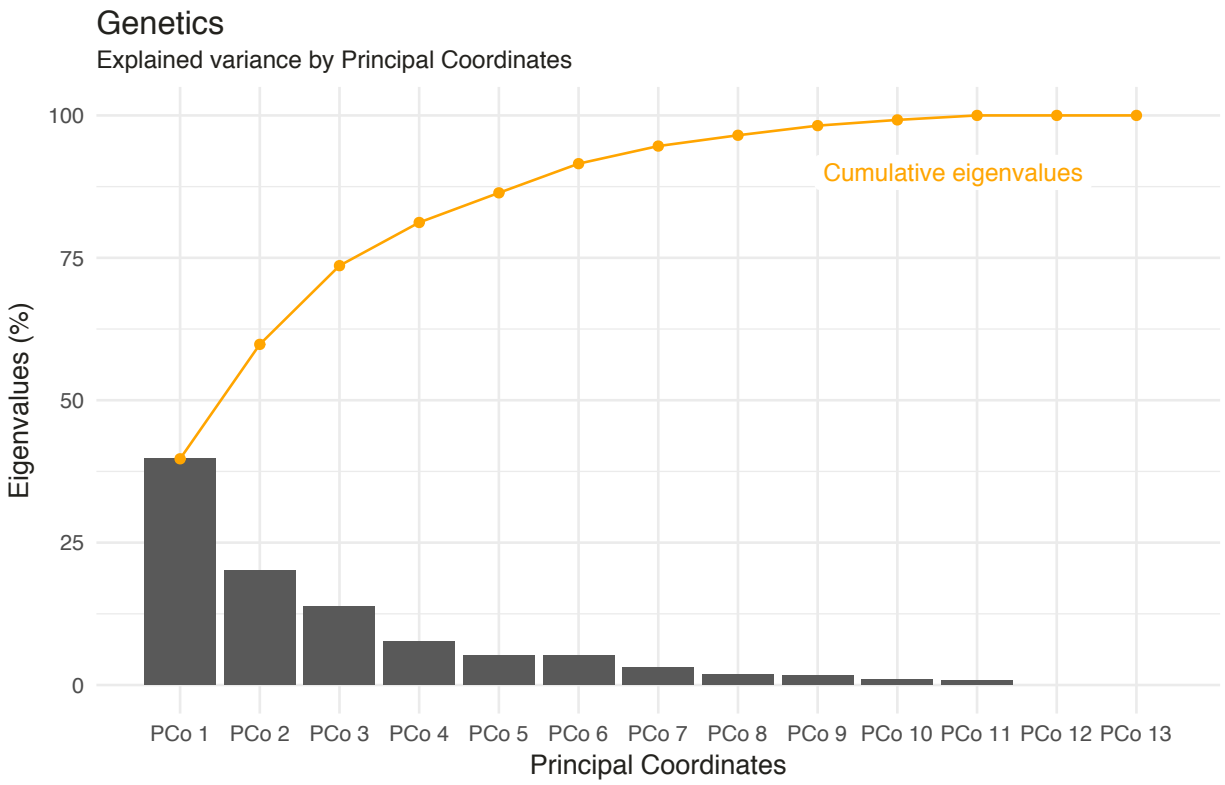

Figure S2: Scree plot of explained variance for Genetics

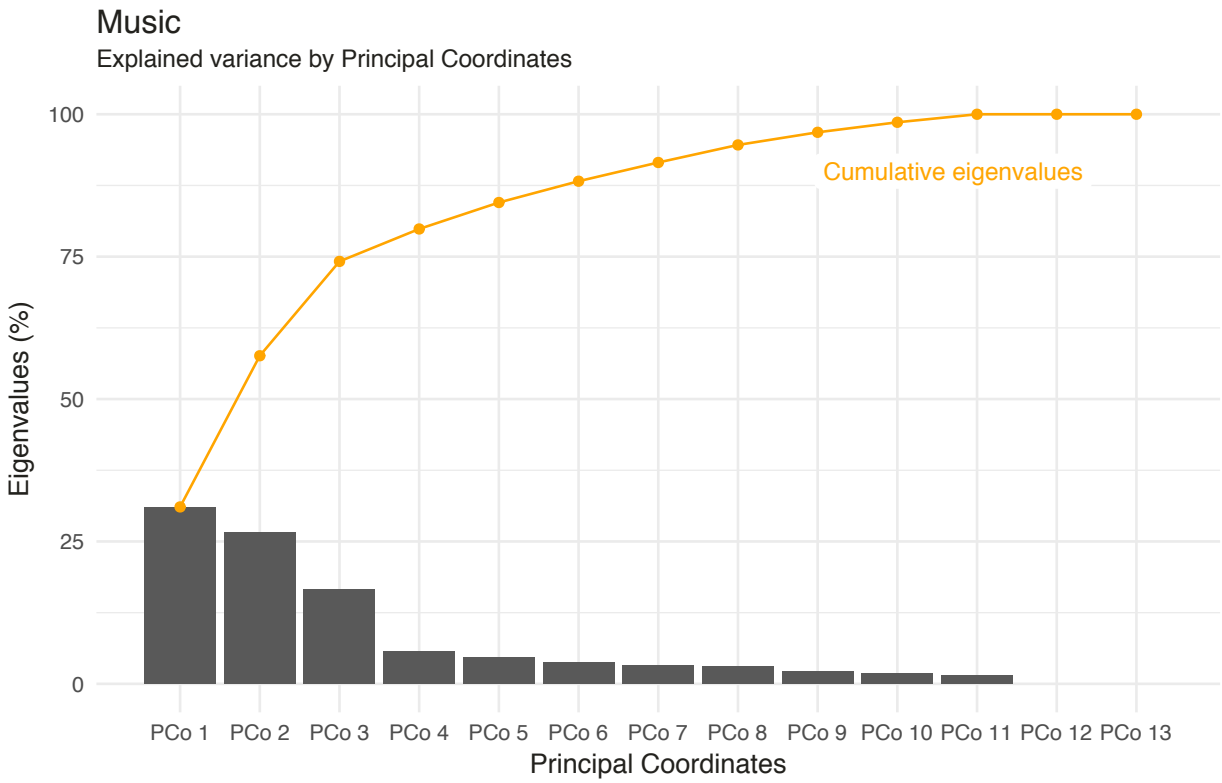

Figure S3: Scree plot of explained variance for Music

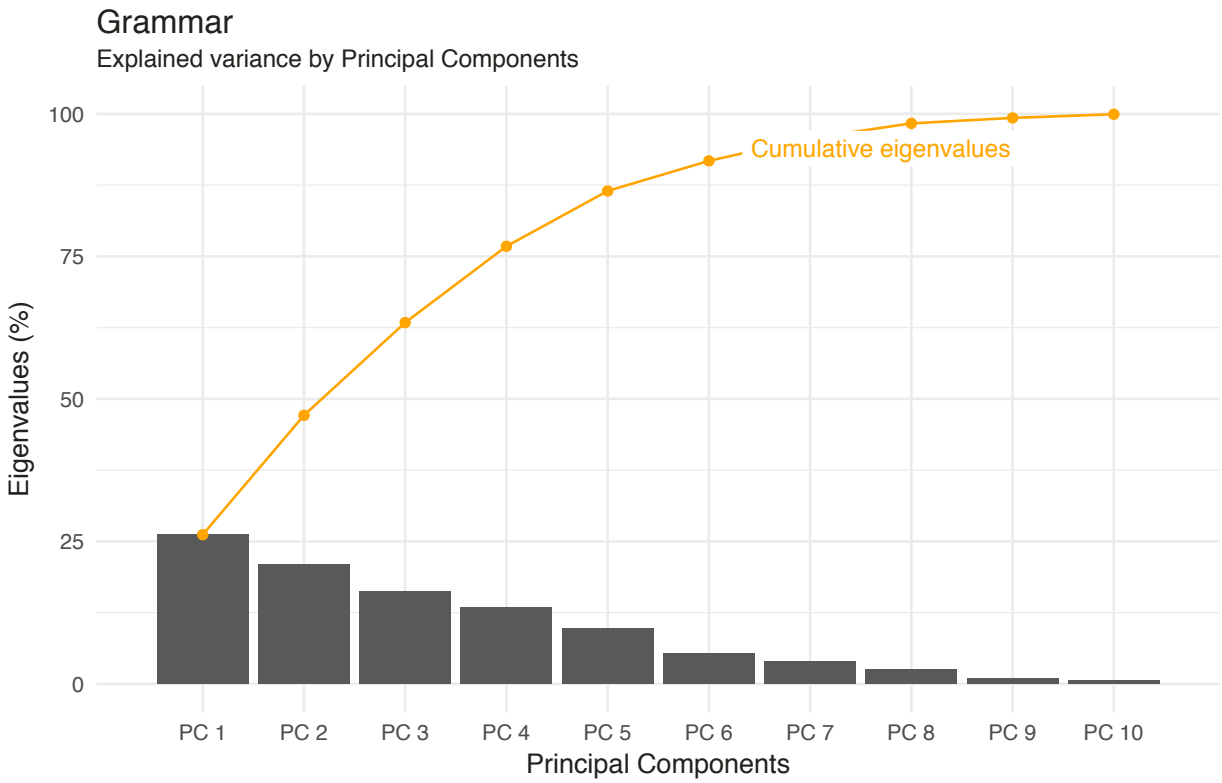

Figure S4: Scree plot of explained variance for Gramamr

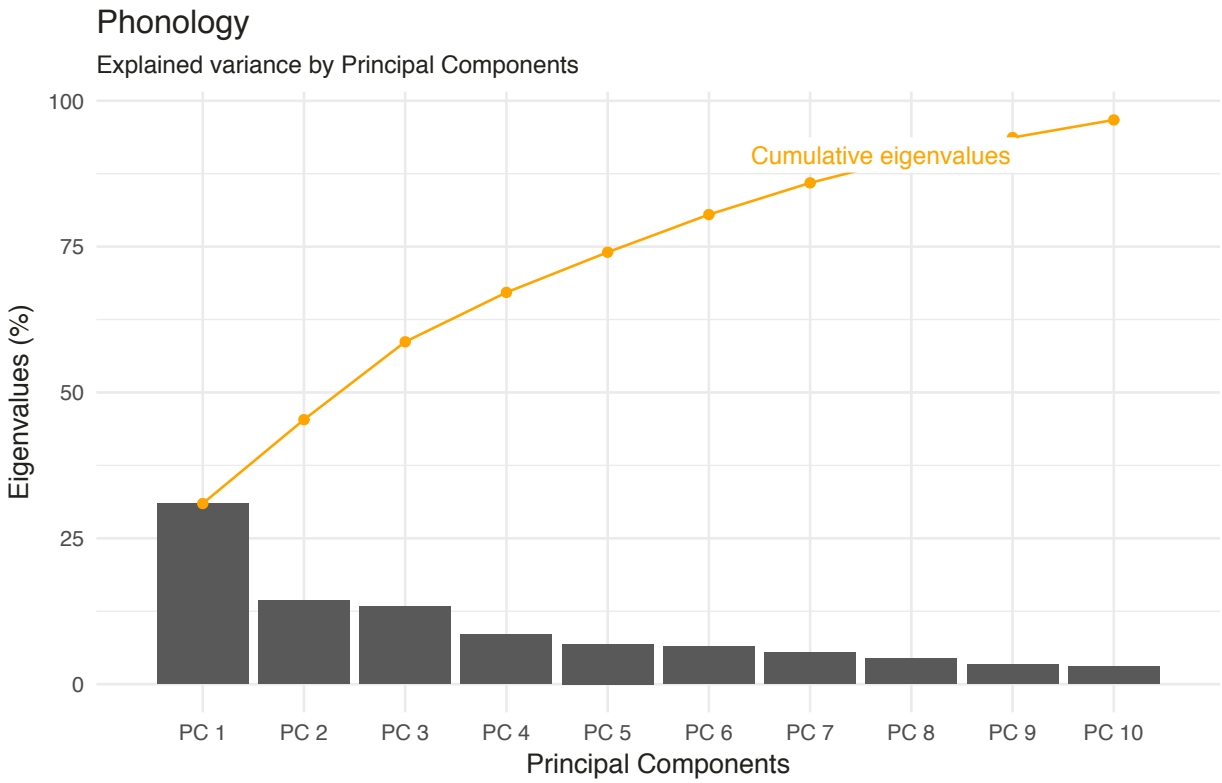

Figure S5: Scree plot of explained variance for Phonology

S3.5 Heatmaps of PCs and PCos

We extract the principal components/coordinates for all factors: genetics, grammar, music, phonology. We normalize the PCs/PCos to a range from 0 to 1 and then plot a heatmap for each factor.

```
# Extract the PCs and the PCos from the PCA and PCoA results
genetics_pco <- genetics_pcoa$vectors
music_pco <- music_pcoa$vectors
grammar_pc <- grammar_famd$ind$coord
phonology_pc <- phonology_famd$ind$coord

# Change the column names of all PCs and PCoAs
colnames(genetics_pco) <- paste("genetics_pco_", seq(1,ncol(genetics_pco)), sep="")
colnames(music_pco) <- paste("music_pco_", seq(1,ncol(music_pco)), sep="")
colnames(grammar_pc) <- paste("grammar_pc_", seq(1,ncol(grammar_pc)), sep="")
colnames(phonology_pc) <- paste("phonology_pc_", seq(1,ncol(phonology_pc)), sep="")
```

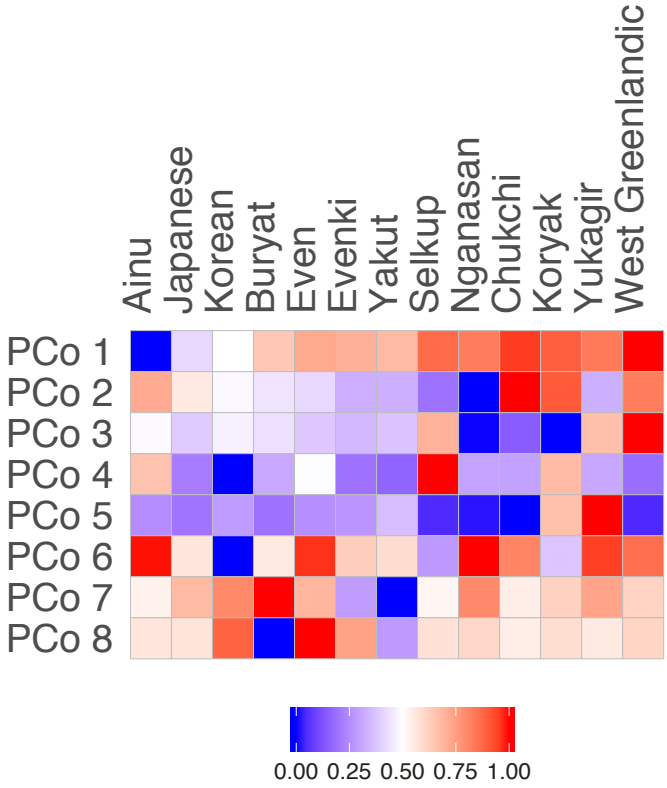

Figure S6: Heat plot of the first four PCos (normalized) of Genetics

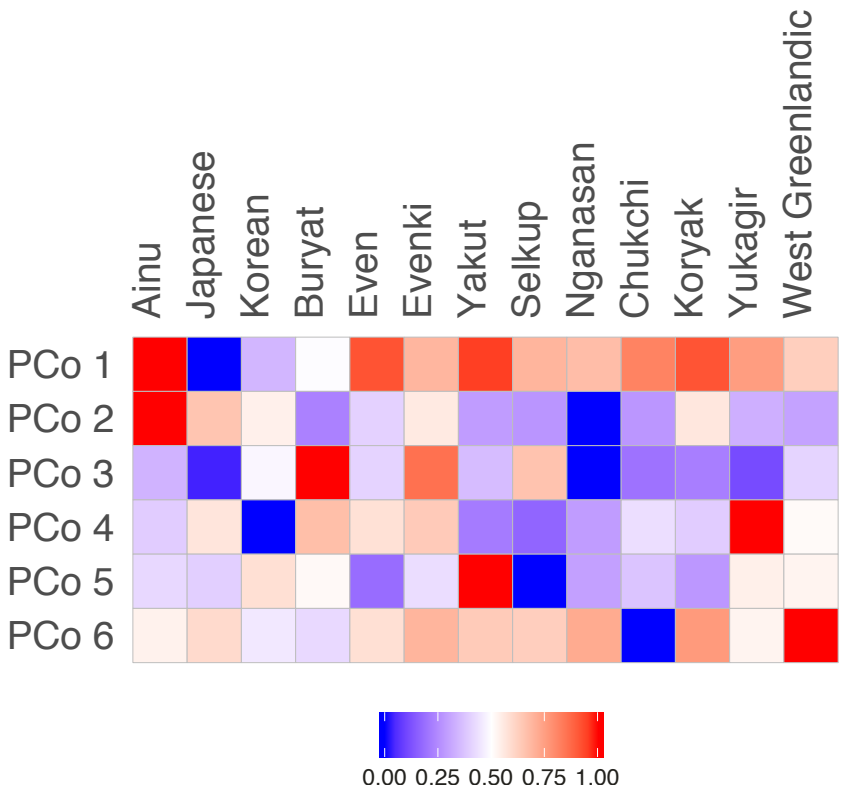

Figure S7: Heat plot of the first five PCos (normalized) of Music

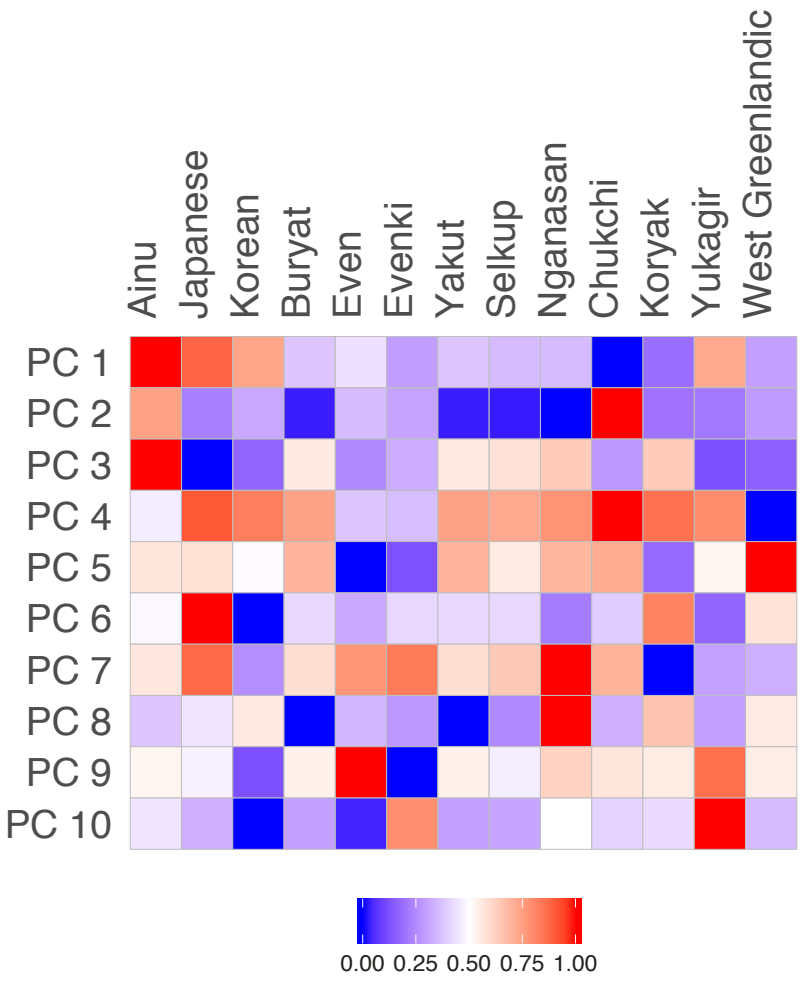

Figure S8: Heat plot of the first six PCs (normalized) of Grammar

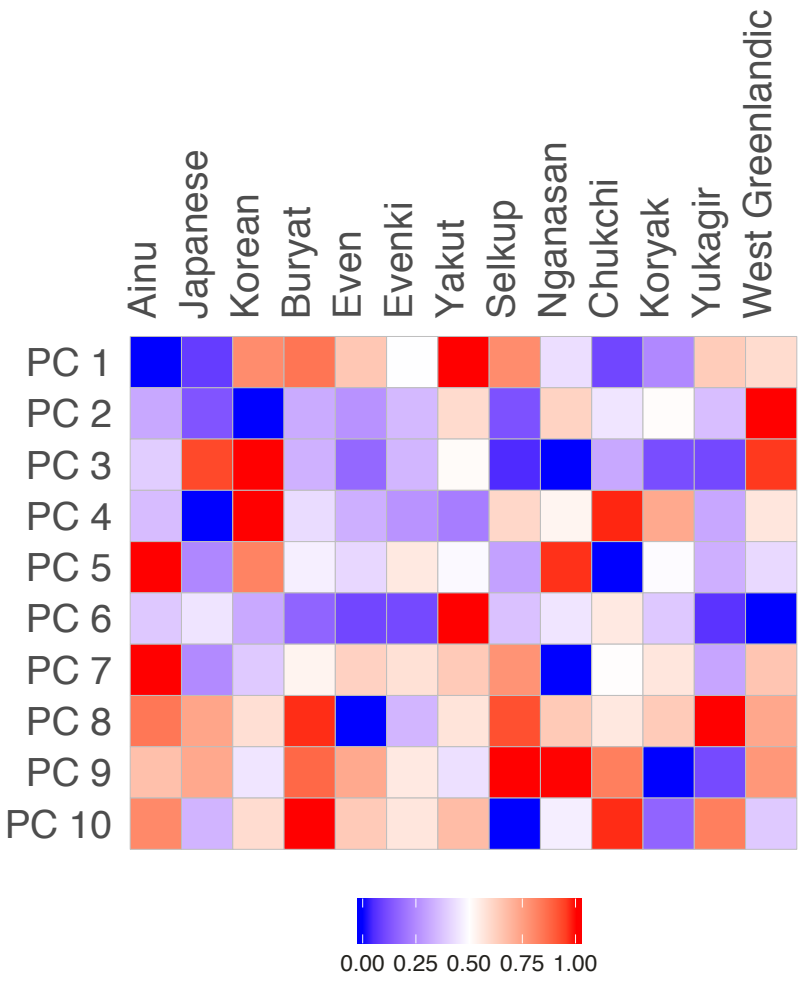

Figure S9: Heat plot of the first six PCs (normalized) of Phonology

S4 Distance visualization (NeighborNets)

Lexical distances are visualized directly. For nonlexical data, we compute Euclidean distances from the dimensionality-reduced data.

Table S3: Lexical distances

	Ainu	Buryat	Chukchi	Even	Evenk	Japanese	Korean	Koryak	Selkup	WstGrnln	Yakut
Buryat	97.88										
Chukchi	100.93	97.66									
Even	97.54	92.58	95.09								
Evenk	98.80	91.57	97.57	60.09							
Japanese	96.70	101.78	99.93	100.36	100.23						
Korean	99.63	99.79	97.88	98.81	94.90	100.54					
Koryak	100.93	96.32	58.09	95.27	98.52	101.85	98.06				
Selkup	98.49	102.29	100.30	97.37	97.66	98.79	98.91	98.57			
WstGrnln	101.45	102.17	100.12	97.65	97.10	100.77	99.75	100.06	99.09		
Yakut	100.91	95.27	101.29	100.09	100.15	104.37	101.13	101.32	98.68	102.77	
Yukagir	102.36	101.08	98.46	100.05	100.27	96.72	98.75	97.08	97.59	100.87	95.61

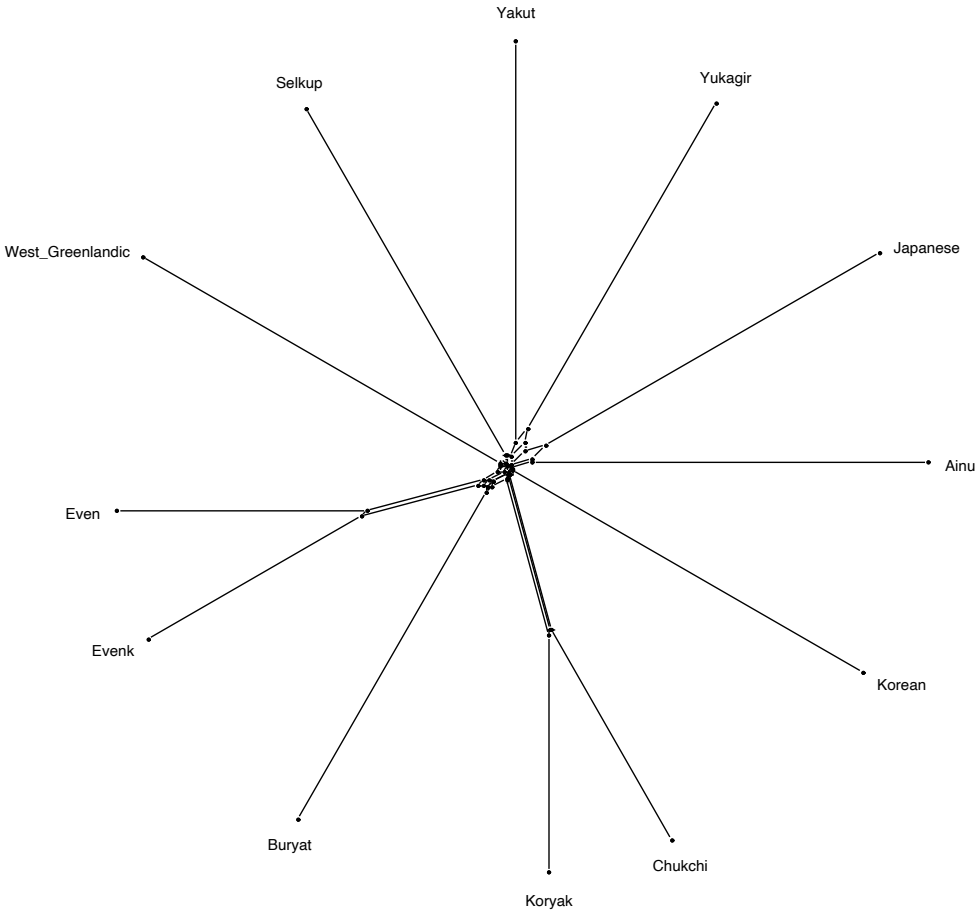

Figure S10: Lexical (ASJP) distances

Table S4: Genetic distances

	Ainu	Buryat	Chukchi	Even	Evenki	Japanese	Korean	Koryak	Nganasan	Selkup	WstGrnln	Yakut
Buryat	0.982											
Chukchi	1.434	0.620										
Even	1.089	0.169	0.548									
Evenki	1.089	0.171	0.606	0.134								
Japanese	0.642	0.354	0.860	0.474	0.461							
Korean	0.795	0.264	0.795	0.396	0.356	0.181						
Koryak	1.353	0.552	0.222	0.481	0.542	0.784	0.723					
Nganasan	1.347	0.471	0.720	0.386	0.351	0.751	0.663	0.659				
Selkup	1.346	0.449	0.670	0.364	0.394	0.766	0.661	0.620	0.445			
WstGrnln	1.506	0.659	0.450	0.595	0.648	0.934	0.836	0.561	0.800	0.558		
Yakut	1.053	0.168	0.636	0.176	0.068	0.430	0.329	0.569	0.393	0.425	0.673	
Yukagir	1.276	0.357	0.591	0.273	0.295	0.674	0.564	0.546	0.441	0.305	0.489	0.32

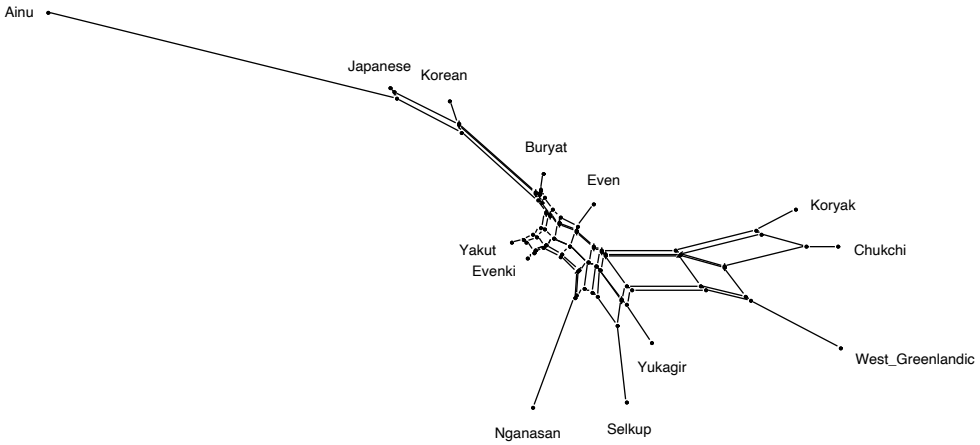

Figure S11: Genetic distances

Table S5: Music distances

	Ainu	Buryat	Chukchi	Even	Evenki	Japanese	Korean	Koryak	Nganasan	Selkup	WstGrnln	Yakut
Buryat	1.067											
Chukchi	0.812	0.579										
Even	0.650	0.607	0.237									
Evenki	0.657	0.422	0.507	0.394								
Japanese	1.193	0.891	1.009	1.080	0.900							
Korean	0.895	0.492	0.635	0.677	0.464	0.489						
Koryak	0.483	0.735	0.352	0.207	0.434	1.034	0.666					
Nganasan	1.146	0.641	0.372	0.560	0.761	1.028	0.739	0.672				
Selkup	0.874	0.333	0.320	0.332	0.349	0.948	0.510	0.470	0.468			
WstGrnln	0.863	0.379	0.295	0.355	0.379	0.826	0.436	0.445	0.402	0.213		
Yakut	0.773	0.639	0.241	0.223	0.512	1.154	0.740	0.341	0.499	0.385	0.384	
Yukagir	0.790	0.582	0.187	0.281	0.487	0.918	0.594	0.347	0.405	0.379	0.255	0.326

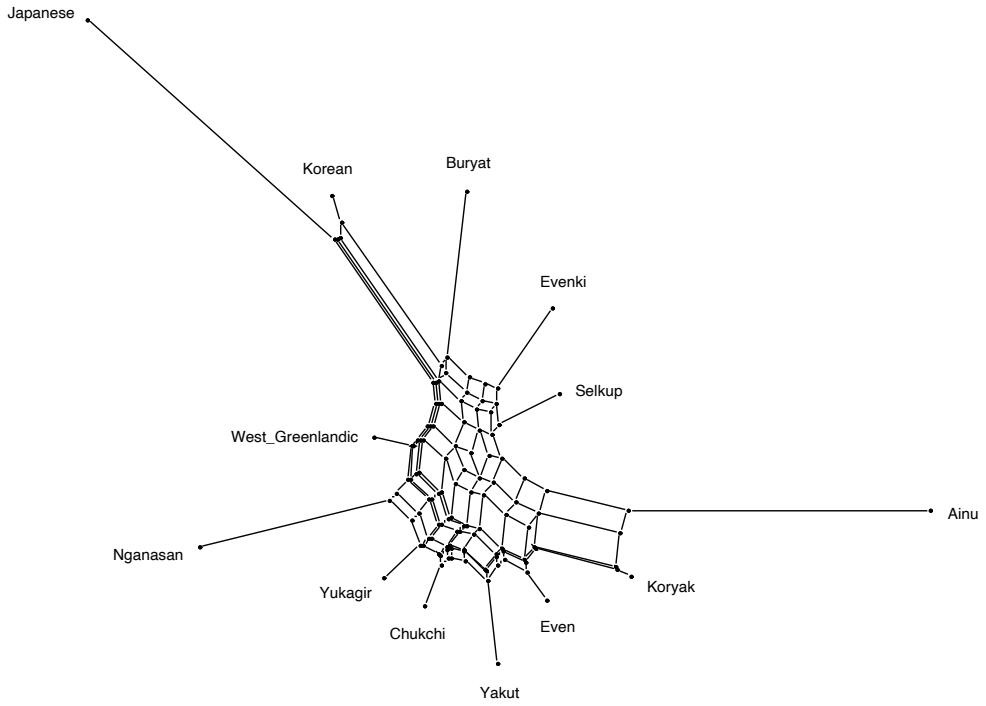

Figure S12: Music distances

Table S6: Grammar distances

	Ainu	Buryat	Chukchi	Even	Evenki	Japanese	Korean	Koryak	Nganasan	Selkup	WstGrnln	Yakut
Buryat	81.938											
Chukchi	105.394	80.476										
Even	76.006	42.796	72.558									
Evenki	83.883	37.314	67.703	16.119								
Japanese	73.789	59.836	99.902	56.485	66.184							
Korean	65.703	46.522	84.511	40.392	50.336	28.816						
Koryak	90.254	31.855	67.829	44.215	35.347	75.693	60.764					
Nganasan	84.245	13.441	82.380	47.237	41.213	65.224	51.388	33.593				
Selkup	82.919	6.183	80.227	40.686	34.415	62.122	48.249	27.935	11.871			
WstGrnln	91.142	48.505	76.418	43.015	37.660	71.927	60.151	58.889	53.682	49.467		
Yakut	81.938	0.000	80.476	42.796	37.314	59.836	46.522	31.855	13.441	6.183	48.505	
Yukagir	70.834	43.200	88.477	40.515	49.621	25.549	10.140	60.228	49.049	45.307	57.717	43.2

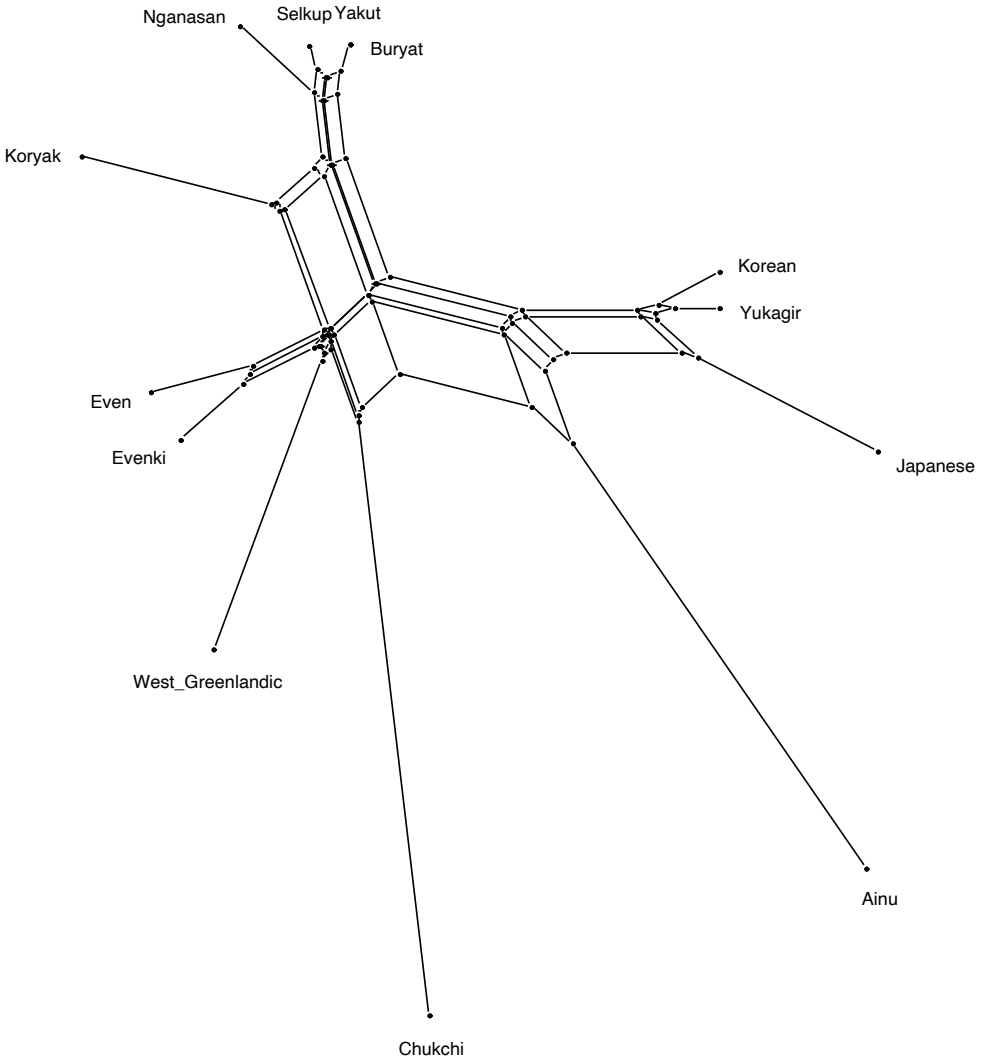

Figure S13: Grammar distances (scaled)

Table S7: Phonology distances

	Ainu	Buryat	Chukchi	Even	Evenki	Japanese	Korean	Koryak	Nganasan	Selkup	WstGrnln	Yakut
Buryat	83.269											
Chukchi	35.093	75.481										
Even	66.705	26.282	59.691									
Evenki	51.217	35.828	47.501	17.831								
Japanese	36.352	79.779	42.402	65.157	50.469							
Korean	85.044	37.856	78.678	45.054	48.572	76.406						
Koryak	35.069	61.962	24.821	45.521	32.485	47.321	70.136					
Nganasan	52.493	49.202	45.957	37.905	29.319	60.815	64.454	28.793				
Selkup	80.852	21.502	72.144	26.125	37.724	79.301	42.573	58.836	50.660			
WstGrnln	74.627	51.837	65.357	52.763	45.631	72.464	61.988	55.689	50.231	63.784		
Yakut	99.986	32.177	91.516	47.354	55.255	95.393	52.201	77.806	63.314	43.611	55.289	
Yukagir	66.633	24.819	58.545	19.952	21.568	65.137	49.337	42.638	33.907	28.432	50.335	47.953

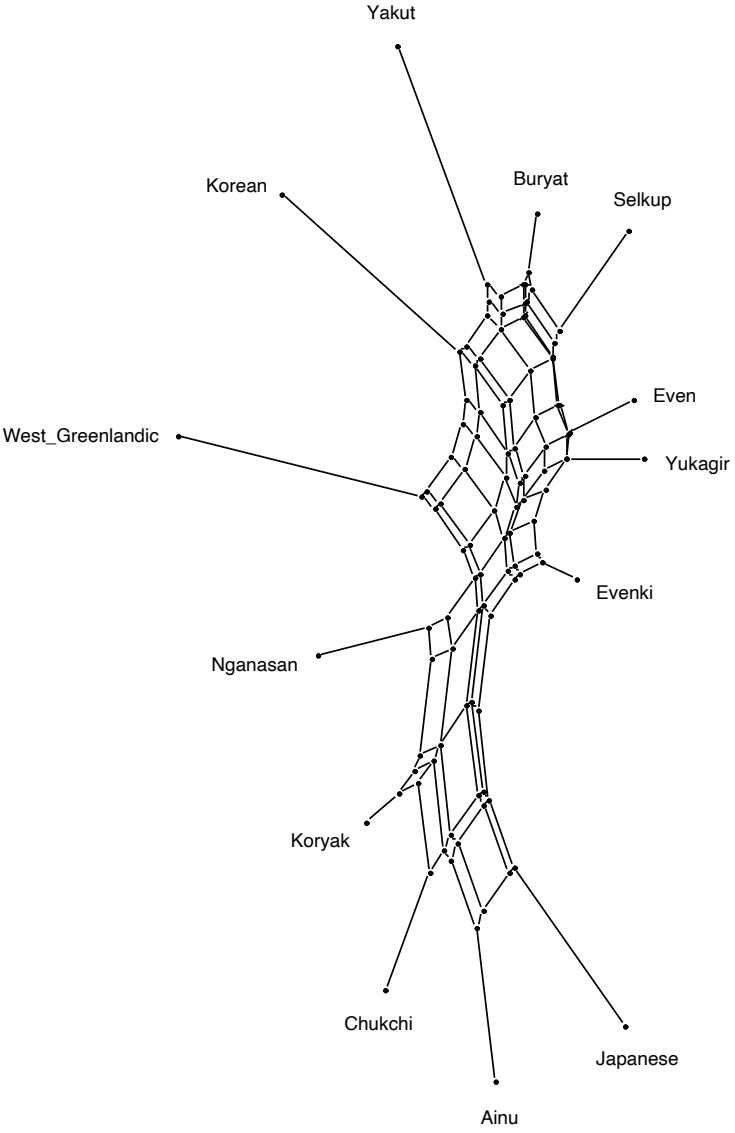

Figure S14: Phonology distances (scaled)

S5 Redundancy Analysis (RDA)

Redundancy Analysis (RDA) extracts the variation in a set of explanatory variables that can be explained by a set of response variables (Legendre and Legendre 2012). In our case the response and explanatory variables comprise the principle components (coordinates) of a factor, e.g. the explanatory variable may comprise the PCs of grammar and the response those of phonology. RDA carries out a multiresponse multiple linear regression, i.e. regression of multiple response variables on multiple explanatory variables (Van Den Wollenberg 1977).

S5.1 Partial RDA

We collect the pairs of all four matrices where each of them once functions as a predictor and once as a response:

```
factor_names <- c("genetics", "music", "grammar","phonology")

# All possible combinations for which we perfom RDA
all_comb <- expand.grid(factor_names, factor_names)
comb <- all_comb[!all_comb$Var1 == all_comb$Var2, ]
comb <- as.data.frame(t(comb), stringsAsFactors=FALSE)
```

For all pairs we perform a partial RDA where we explore the influence of space on the observed relationships. Partial RDA removes the effects of one or more explanatory variables on the response prior to an ordinary RDA (Borcard, Legendre, and Drapeau 1992). In addition to space, we also control for the possible influence of phylogenetic inheritance in the three cases where we have data from the same language families (Even and Evenki from the Tungusic, Selkup and Nganasan from the Uralic, and Koryak and Chukchi from the Chukotko-Kamchatkan family). Moreover, we assess to what extent our analysis is sensitive to the number of principal components/coordinates for each factor.

Specifically, we control for

- **the influence of space:**

  Since most languages in our sample occupy relatively large territories [Fig. S1], choosing a central point location might yield a misleading picture of the possible spatial interactions. Instead, we randomly sample 1,000 spatial locations (SL) from the language polygons and perform a partial RDA for each. With each sample we remove the influence for one possible scenario of spatial neighborhood and thereby also control for spatial autocorrelation.

- **the influence of phylogenetic inheritance:**

  For each sample we randomly pick only one language from the three language families with two members.

- **the influence of the number of PCs/PCos:**

  Sub-sampling reduces the number of languages that can be used as observations in the partial RDA. Consequently, using all principal components/coordinates would yield an overdetermined model. Hence, we only retain those PCs/PCos per factor which account for at least k% of the explained variance. We run the RDA with three different setups ($k = 10\%$, $k = 15\%$ and $k = 20\%$) and compare the results.

For each setup, we compare the distribution of the adjusted R^2 across the 1,000 spatial locations against a distribution for adjusted R^2 values based on random permutations ($N = 100$), i.e. samples for which the rows of the explanatory variable were randomly permuted for each run of the partial RDA. For illustration we report in Section S5.2 the detailed results for $k = 15\%$ and in Section S5.3 we summary the results for all setups.

Prior to performing the RDA, however, we match the order of sites for each factor and combine all factors in a list.

```
# Collect all factors in a list, order alphabetically
# (geo_mem is already in alphabetical order)

n_perm = 100

# Setup k = 10%
# Retain all k PCs/PCos which account for at least 10% of the explained variance
var_th <- 0.1

genetics_pco_rel <- genetics_pco[, which(genetics_pcoa$values$Rel_corr_eig >= var_th)]
grammar_pc_rel <- grammar_pc[, which(grammar_famd$eig[, 2]>=var_th*100)]
phonology_pc_rel <- phonology_pc[, which(phonology_famd$eig[, 2]>=var_th*100)]
music_pco_rel <- music_pco[, which(music_pcoa$values$Rel_corr_eig >= var_th)]


factors_1 = list(genetics = genetics_pco_rel[order(rownames(genetics_pco_rel)), ],
                 music = music_pco_rel[order(rownames(music_pco_rel)), ],
                 grammar = grammar_pc_rel[order(rownames(grammar_pc_rel)), ],
                 phonology = phonology_pc_rel[order(rownames(phonology_pc_rel)), ],
                 geo = geo_mem)

rda_sp_1 <- lapply(comb, function (x) {
  rda_wrapper(factors_1[x[1]][[1]], factors_1[x[2]][[1]], factors_1$geo,
              indi_lang=T, n_perm=n_perm)})
```

```
# Setup k = 15%
# Retain all k PCs/Pcos which account for at least 15% of the explained variance
var_th <- 0.15

genetics_pco_rel <- genetics_pco[, which(genetics_pcoa$values$Rel_corr_eig >= var_th)]
grammar_pc_rel <- grammar_pc[, which(grammar_famd$eig[, 2]>=var_th*100)]
phonology_pc_rel <- phonology_pc[, which(phonology_famd$eig[, 2]>=var_th*100), drop=F]
music_pco_rel <- music_pco[, which(music_pcoa$values$Rel_corr_eig >= var_th)]

factors_15 = list(genetics = genetics_pco_rel[order(rownames(genetics_pco_rel)), ],
                  music = music_pco_rel[order(rownames(music_pco_rel)), ],
                  grammar = grammar_pc_rel[order(rownames(grammar_pc_rel)), ],
                  phonology = phonology_pc_rel[order(rownames(phonology_pc_rel)), ,drop=F],
                  geo = geo_mem)

rda_sp_15 <- lapply(comb, function (x) {
  rda_wrapper(factors_15[x[1]][[1]], factors_15[x[2]][[1]], factors_15$geo, indi_lang=T,
              n_perm=n_perm)})
```

```
# Setup k = 20%
# Retain all k PCs/Pcos which account for at least 20% of the explained variance
var_th <- 0.2

genetics_pco_rel <- genetics_pco[, which(genetics_pcoa$values$Rel_corr_eig >= var_th)]
grammar_pc_rel <- grammar_pc[, which(grammar_famd$eig[, 2]>=var_th*100)]
phonology_pc_rel <- phonology_pc[, which(phonology_famd$eig[, 2]>=var_th*100), drop=F]
music_pco_rel <- music_pco[, which(music_pcoa$values$Rel_corr_eig >= var_th)]
```

```
factors_2 = list(genetics = genetics_pco_rel[order(rownames(genetics_pco_rel)), ],
                 music = music_pco_rel[order(rownames(music_pco_rel)), ],
                 grammar = grammar_pc_rel[order(rownames(grammar_pc_rel)), ],
                 phonology = phonology_pc_rel[order(rownames(phonology_pc_rel)), , drop=F],
                 geo = geo_mem)

rda_sp_2 <- lapply(comb, function (x) {
  rda_wrapper(factors_2[x[1]][[1]], factors_2[x[2]][[1]], factors_2$geo, indi_lang=T,
              n_perm=n_perm)})
```

```
# Rename the list entries
names(rda_sp_1) <- sapply(rda_sp_1, function (q){
  return (paste(q[[1]]$explanatory, q[[1]]$response, sep="_"))})
names(rda_sp_15) <- sapply(rda_sp_15, function (q){
  return (paste(q[[1]]$explanatory, q[[1]]$response, sep="_"))})
names(rda_sp_2) <- sapply(rda_sp_2, function (q){
  return (paste(q[[1]]$explanatory, q[[1]]$response, sep="_"))})
```

S5.2 Density plots

For setup $k = 15\%$ we generate density plots of the differences of observed and permuted adjusted R^2 values for each association.

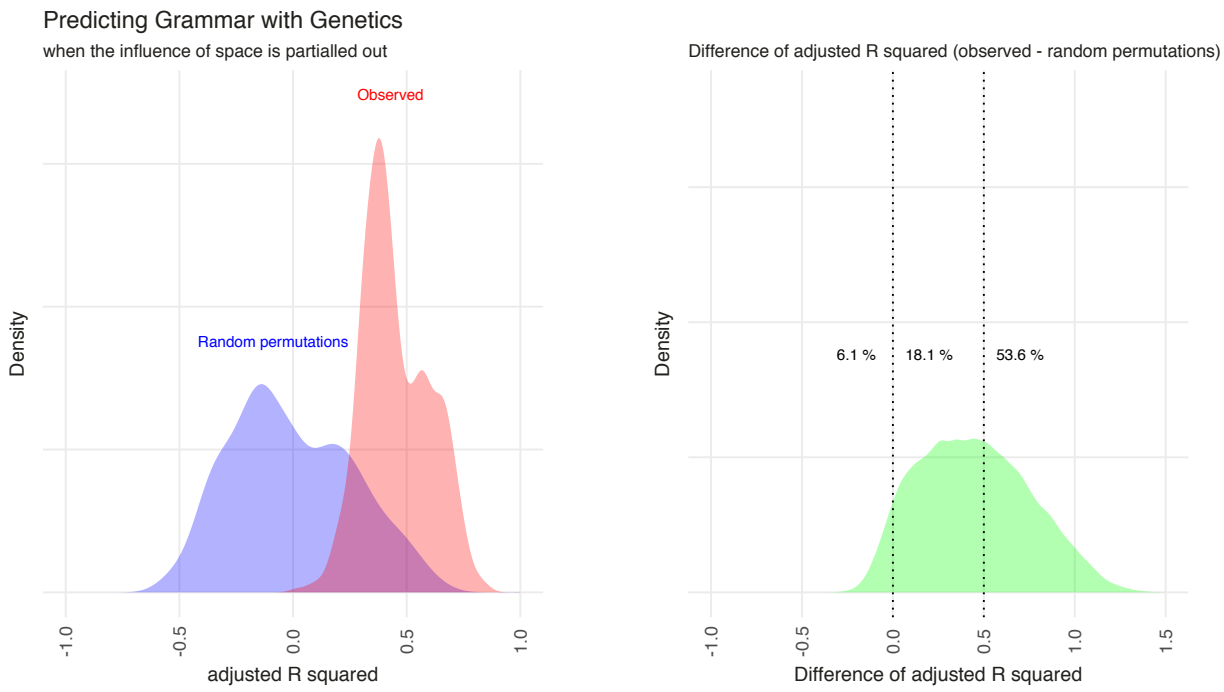

Figure S15: Partial RDA of Genetics (explanatory variable) and Grammar (response)

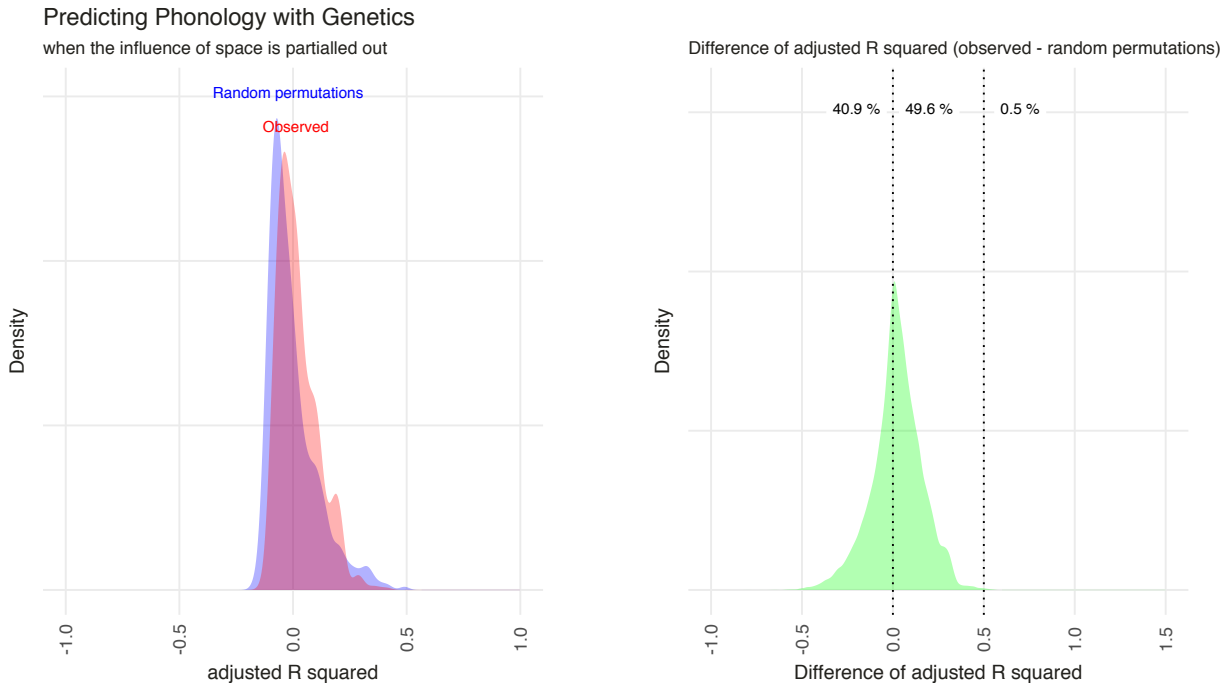

Figure S16: Partial RDA of Genetics (explanatory variable) and Phonology (response)

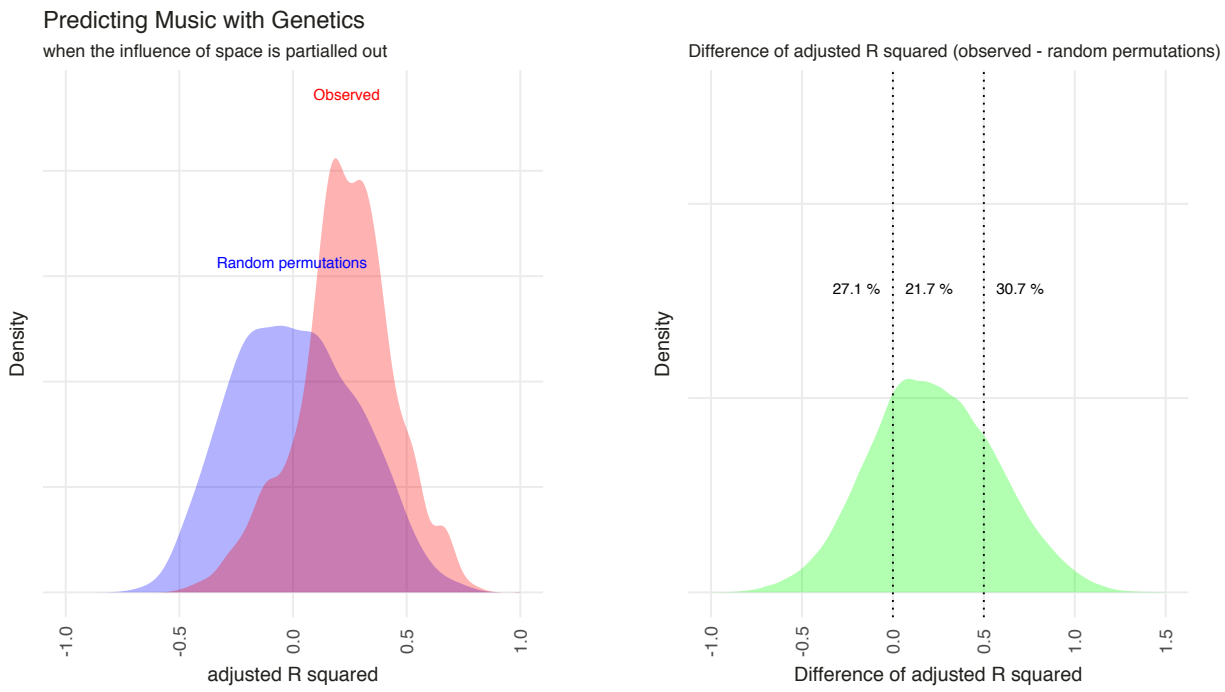

Figure S17: Partial RDA of Genetics (explanatory variable) and Music (response)

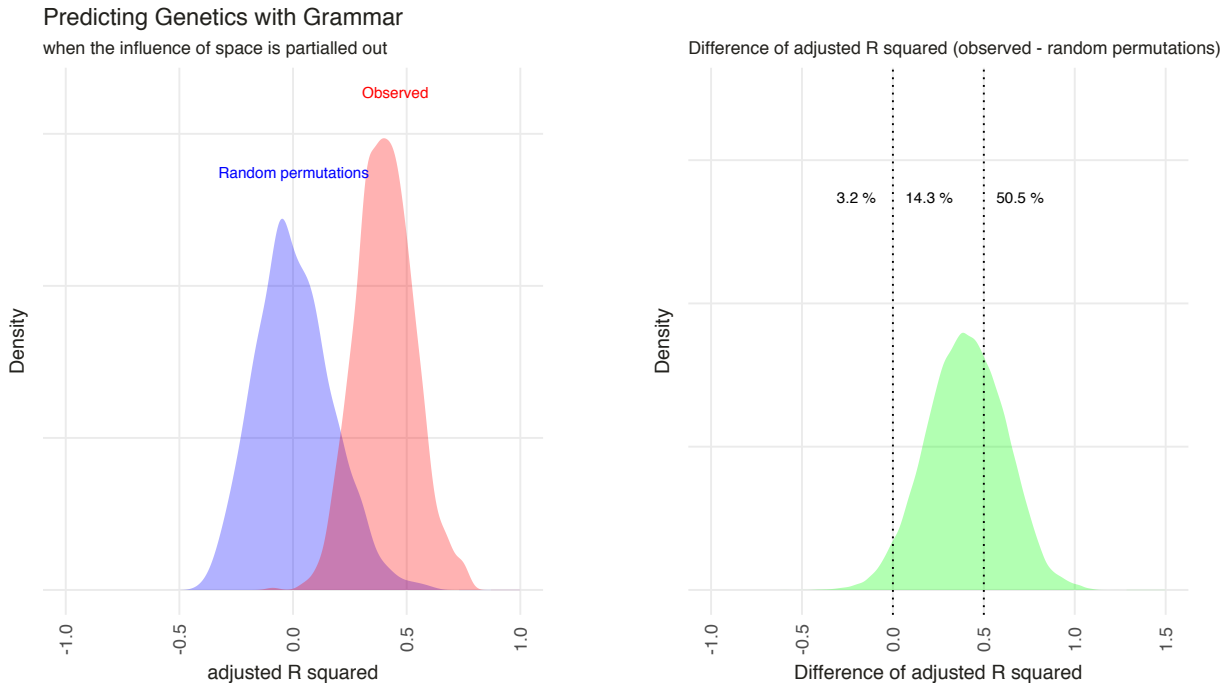

Figure S18: Partial RDA of Grammar (explanatory variable) and Genetics (response)

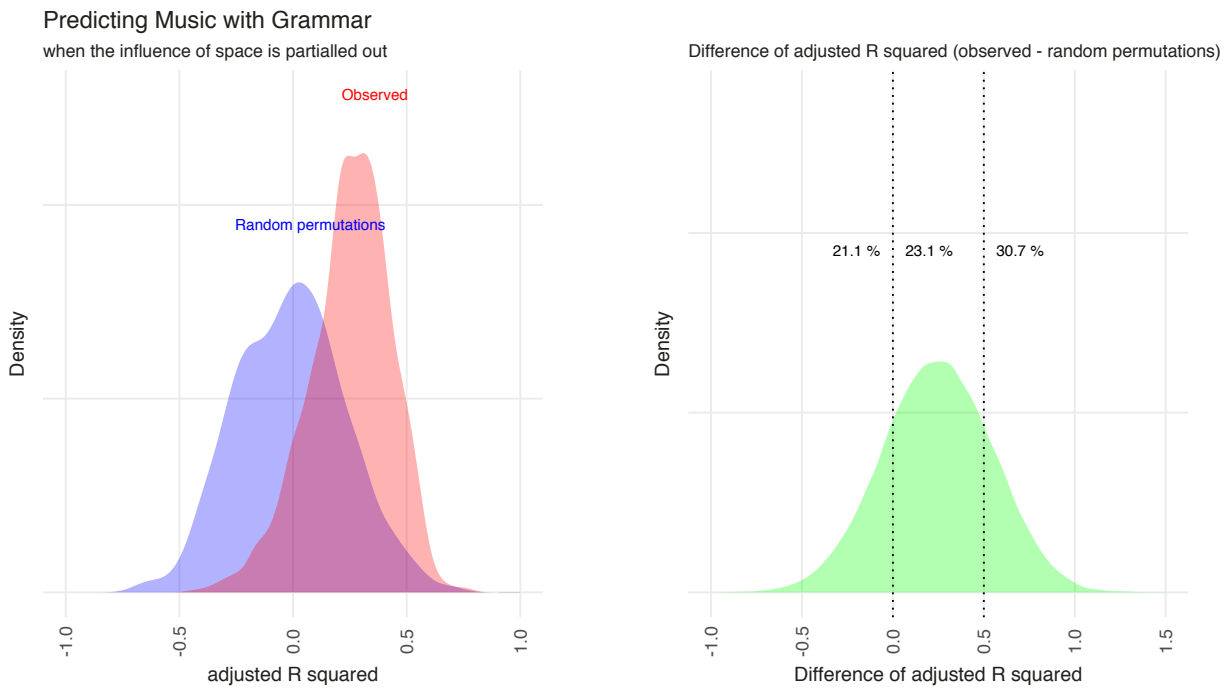

Figure S19: Partial RDA of Grammar (explanatory variable) and Music (response)

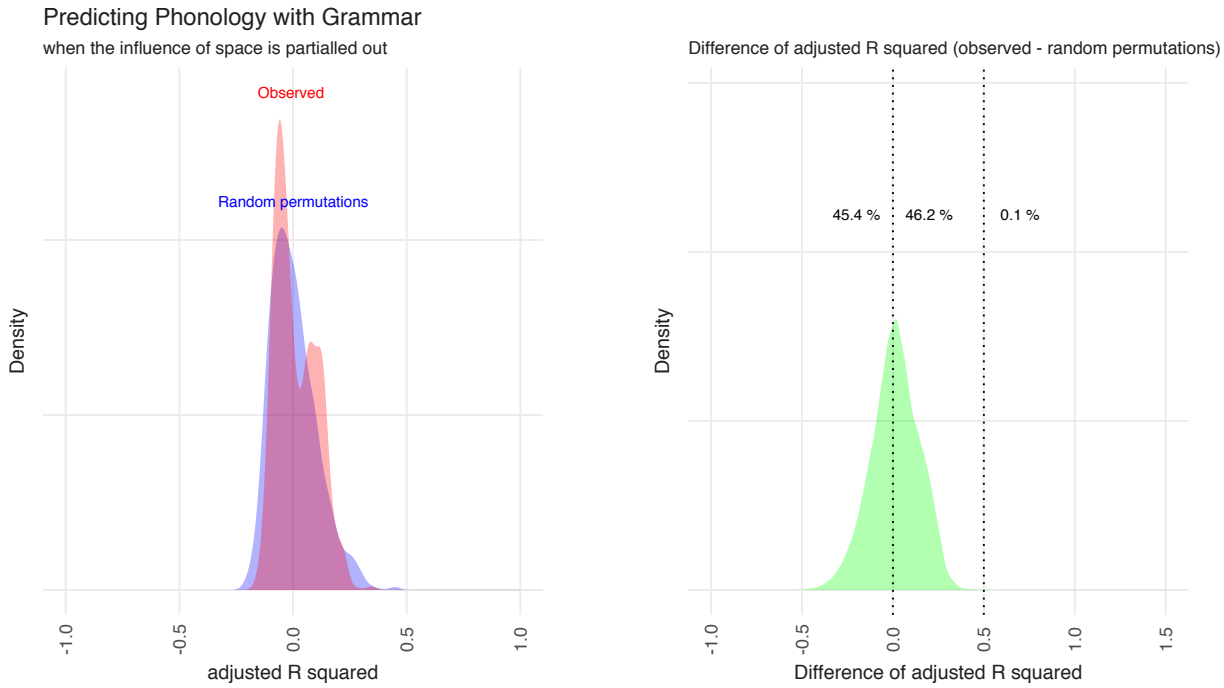

Figure S20: Partial RDA of Grammar (explanatory variable) and Phonology (response)

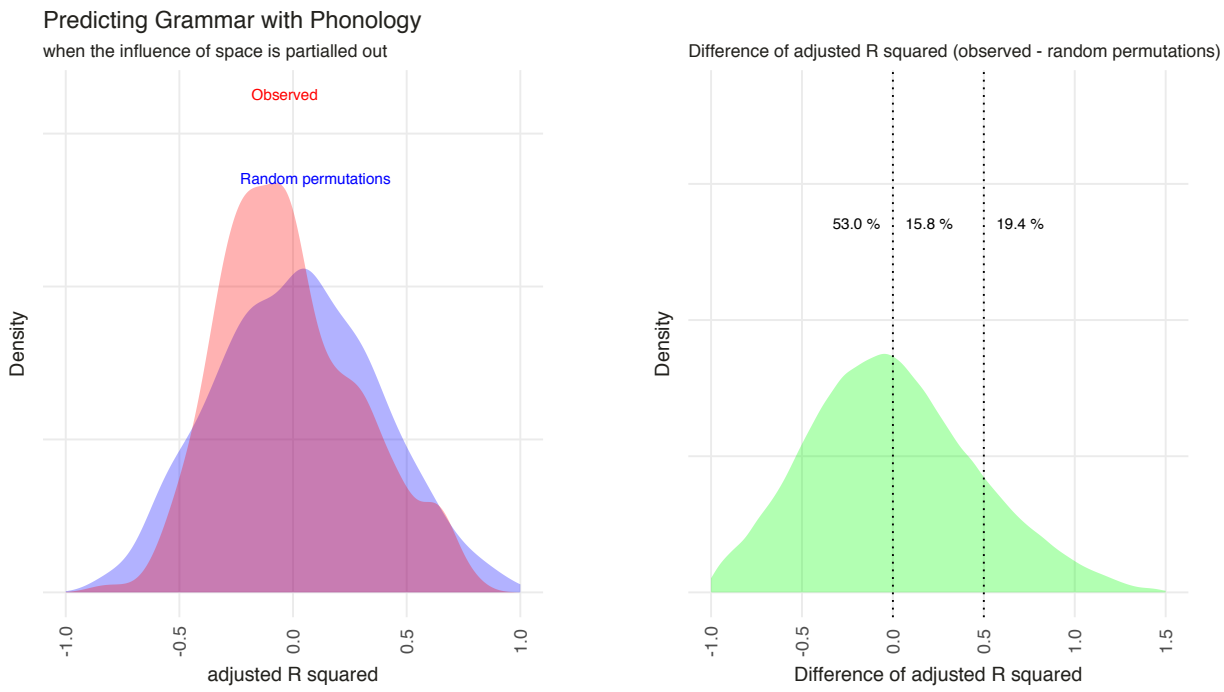

Figure S21: Partial RDA of Phonology (explanatory variable) and Grammar (response)

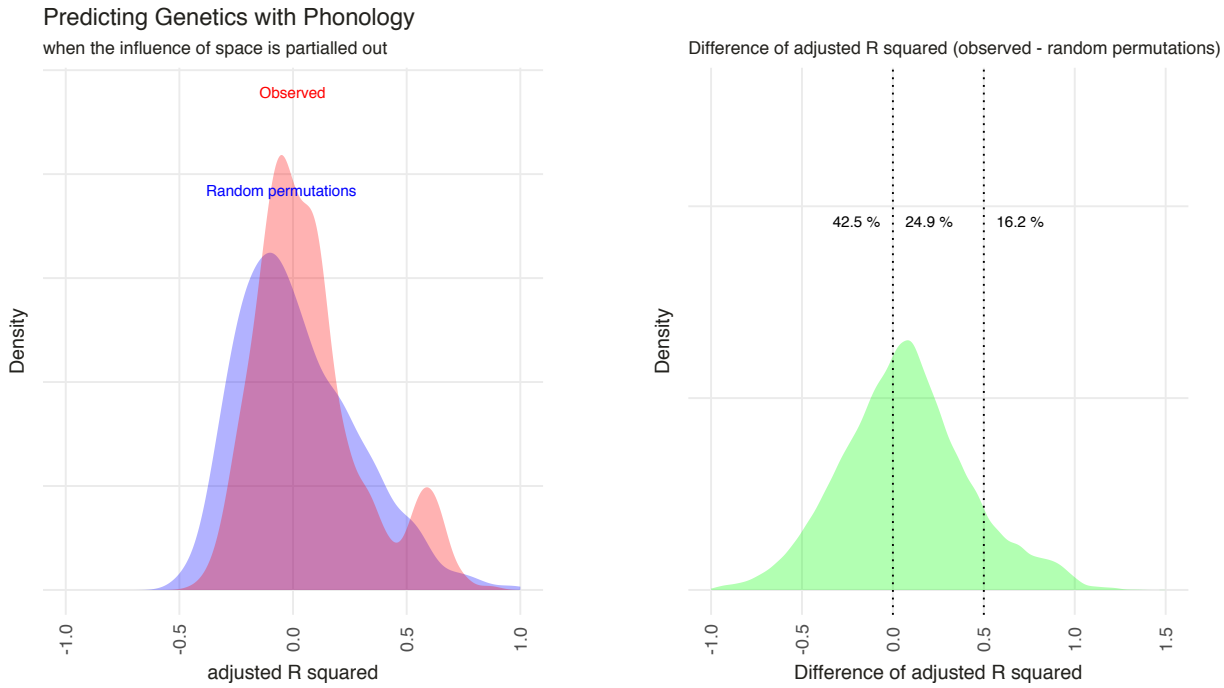

Figure S22: Partial RDA of Phonology (explanatory variable) and Genetics (response)

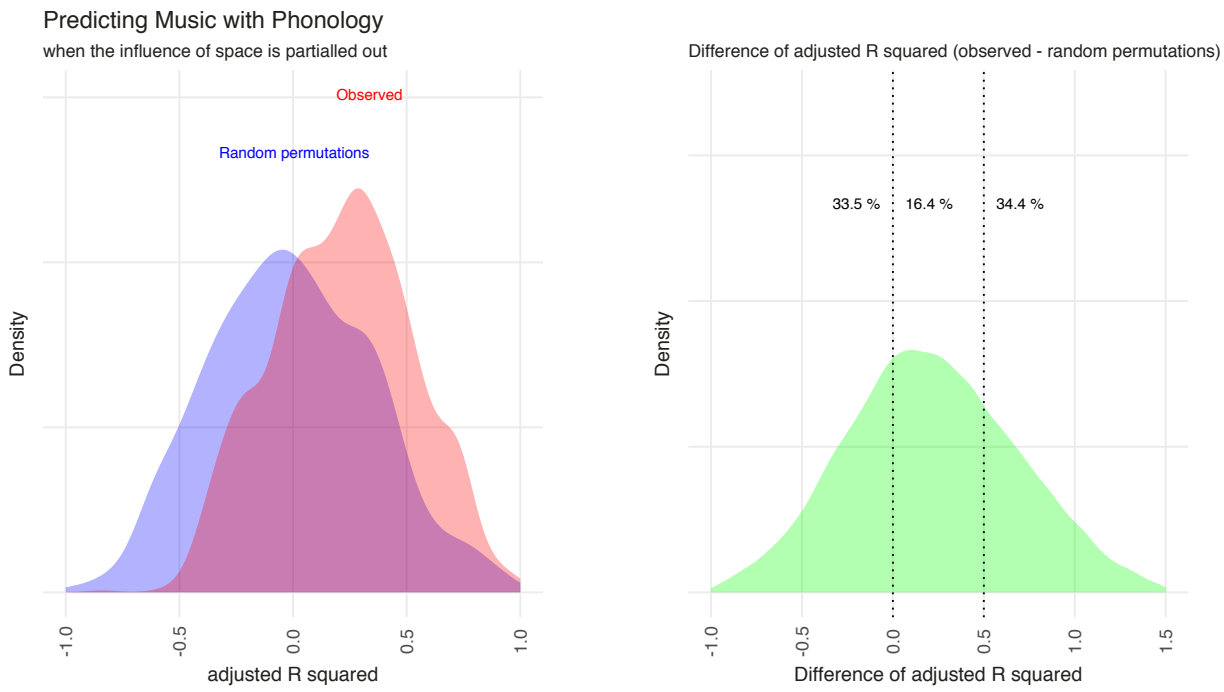

Figure S23: Partial RDA of Phonology (explanatory variable) and Music (response)

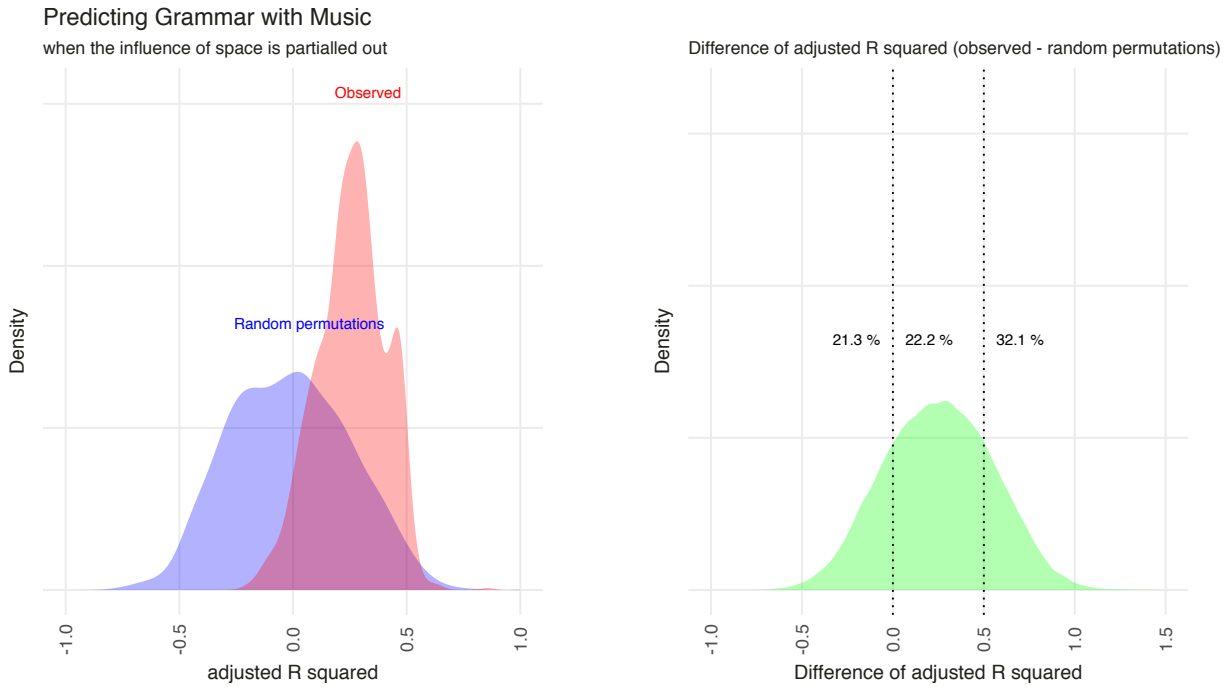

Figure S24: Partial RDA of Music (explanatory variable) and Grammar (response)

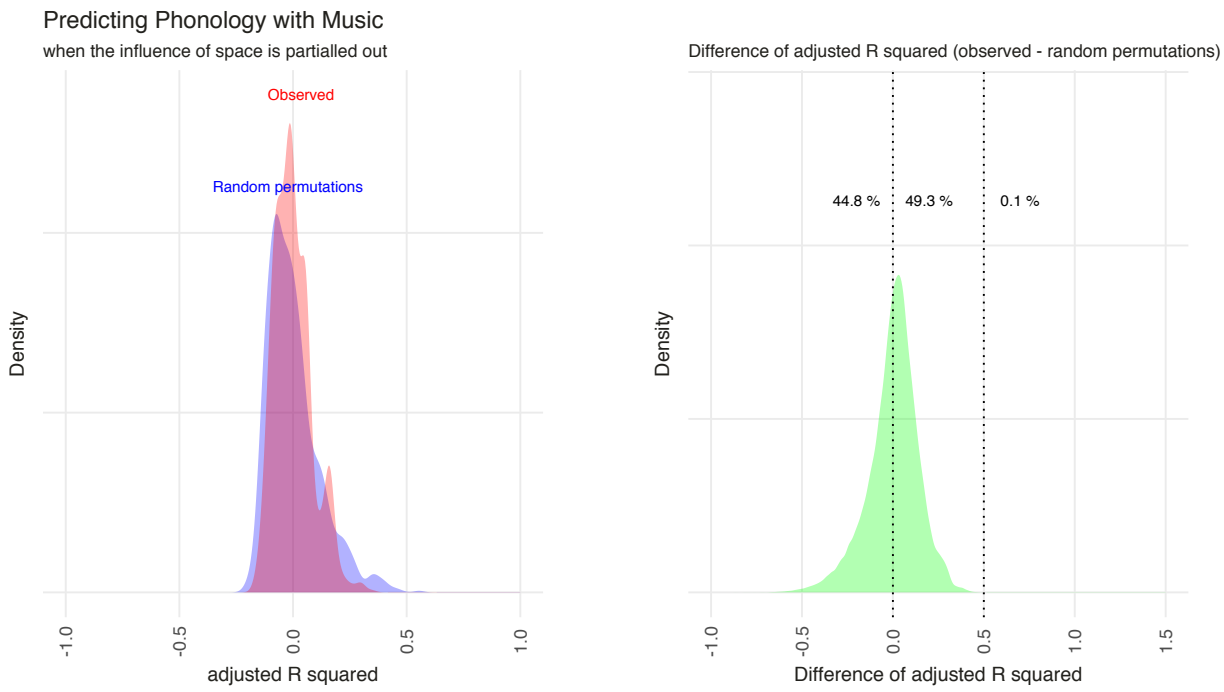

Figure S25: Partial RDA of Music (explanatory variable) and Phonology (response)

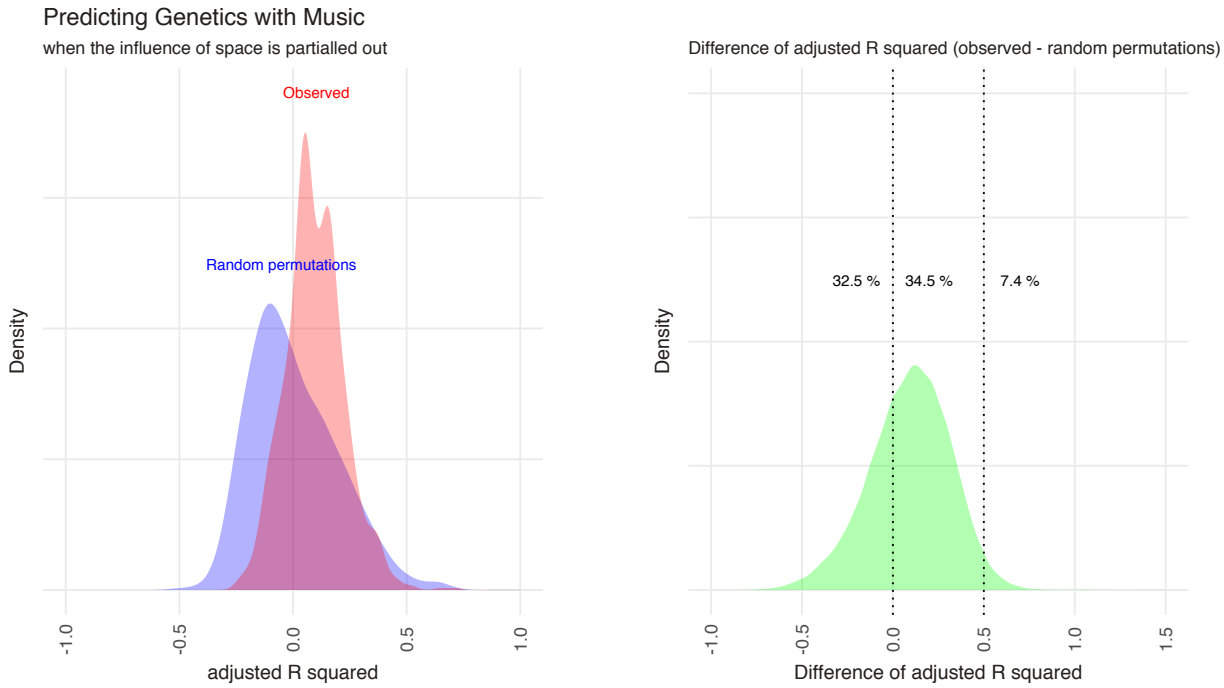

Figure S26: Partial RDA of Music (explanatory variable) and Genetics (response)

S5.3 Comparison

The difference between the empirical and the permuted adjusted R^2 needs to be statistically evaluated in order to determine whether the distribution of permuted adjusted R^2 is likely to produce values equal or larger than the empirical one. To do so, we asses the z-scores of each random spatial location, defined as $z = \Delta R^2 / \text{sd}(R^2_{perm})$, where $\Delta R^2 = R^2_{empirical} - E[R^2_{permuted}]$ and R^2 refers to the adjusted R^2. The expected values and standard deviations are estimated on the sample of all $1,000$ randomly sampled spatial location.

In order to assess how robust the results are across the 1,000 spatial locations that we sampled from the language polygons, we also report the proportion of locations where $z \geq 1.644854$, which corresponds to a significance level $p \leq 0.05$. Results for all k are reported in Figures S27-S28.

```
rda_sp_1_flat <- rda_to_z_val(rda_sp_1, r2_type="r2_adj_semi")
rda_sp_15_flat <- rda_to_z_val(rda_sp_15, r2_type="r2_adj_semi")
rda_sp_2_flat <- rda_to_z_val(rda_sp_2, r2_type="r2_adj_semi")
```

```
sig_diff = 1.644854
proportion_sp_1 <- add_proportion_rda(rda_sp_1_flat, sig_diff = sig_diff)
proportion_sp_15 <- add_proportion_rda(rda_sp_15_flat, sig_diff = sig_diff)
proportion_sp_2 <- add_proportion_rda(rda_sp_2_flat, sig_diff = sig_diff)
```

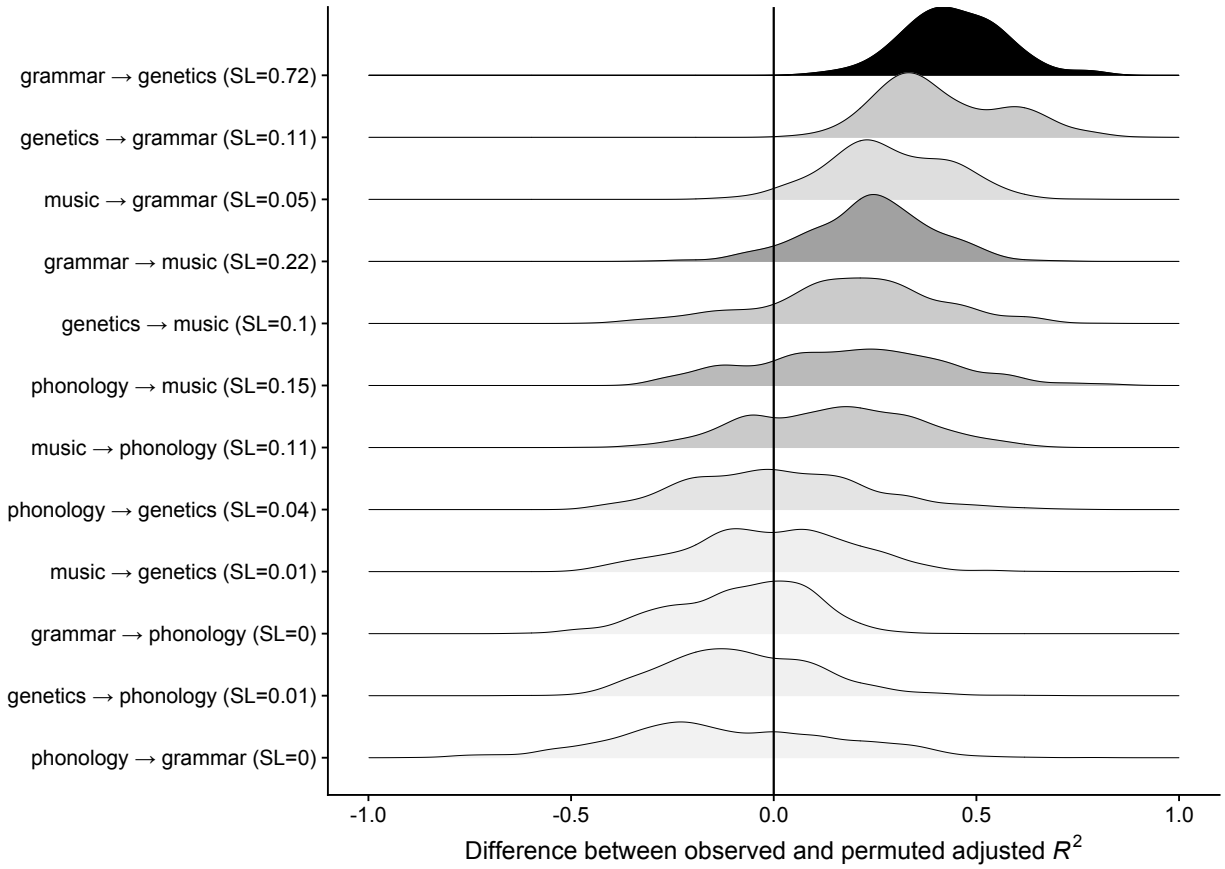

Figure S27: Densities of the difference between observed and permuted adjusted R^2 values in the partial RDA. All input components contribute at least 10% to the explained variance. Numbers between brackets (and grey shading) correspond to the proportion of spatial locations (SL) for which the difference between observed and permuted adjusted R^2 is larger than 0 with $p \leq .05$

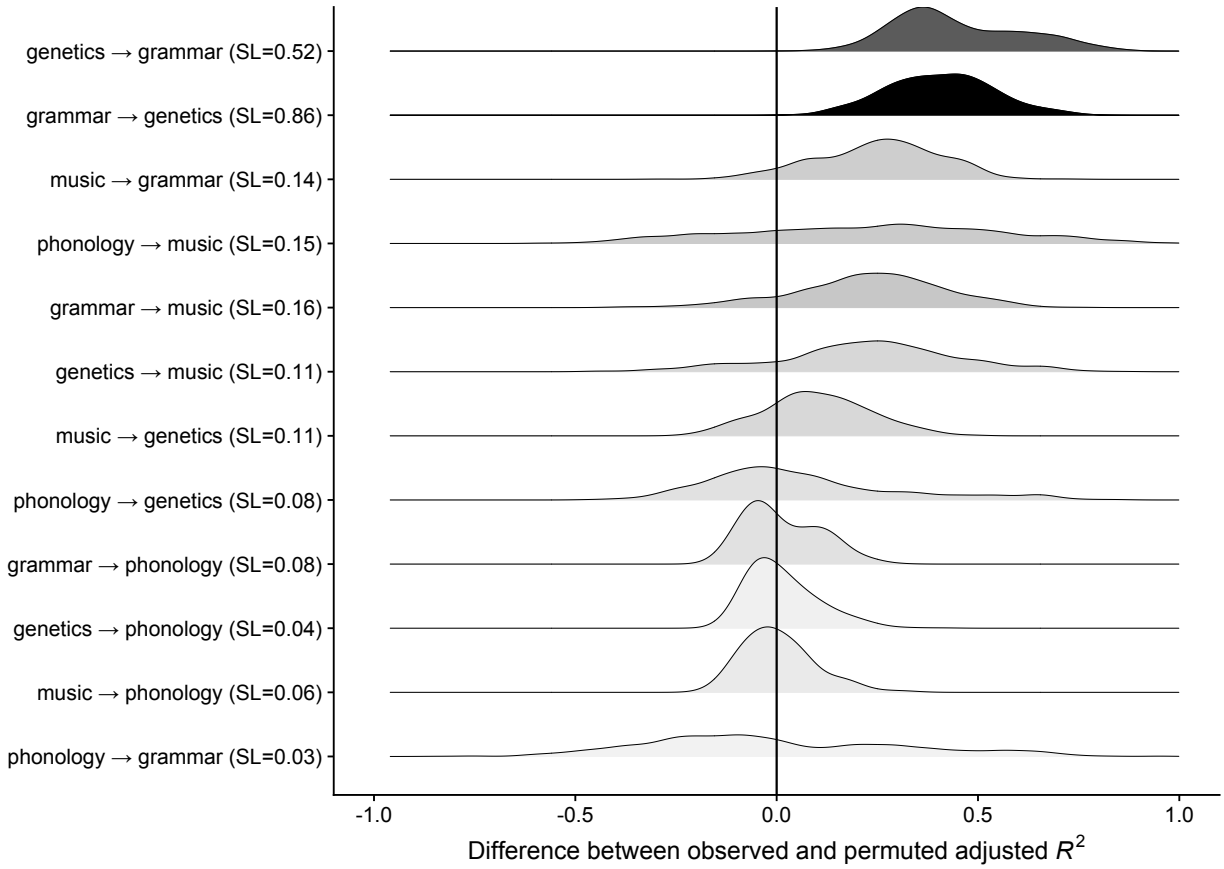

Figure S28: Densities of the difference between observed and permuted adjusted R^2 values in the partial RDA. All input variables contribute at least 15% to the explained variance. Numbers between brackets (and grey shading) correspond to the proportion of spatial locations (SL) for which the difference between observed and permuted adjusted R^2 is larger than 0 with $p \leq .05$

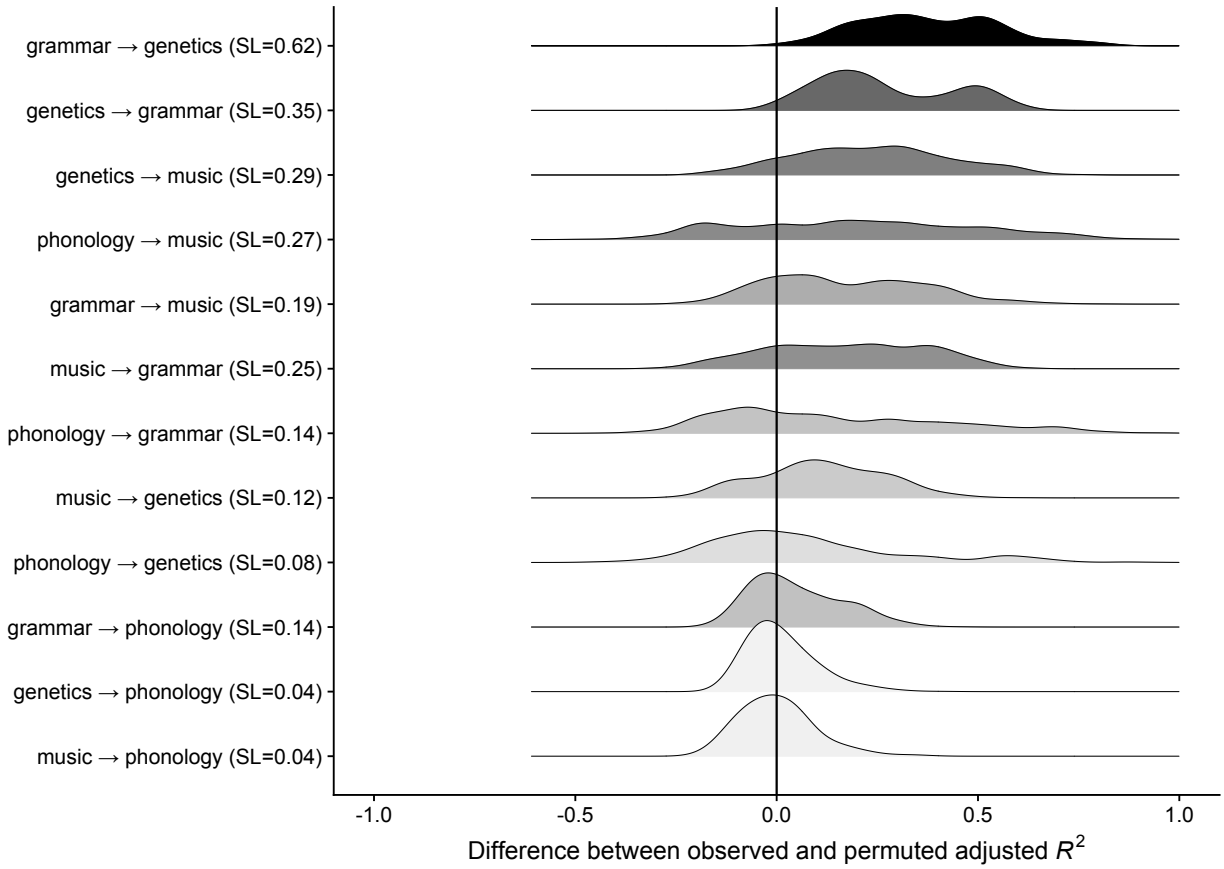

Figure S29: Densities of the difference between observed and permuted adjusted R^2 values in the partial RDA. All input variables contribute at least 20% to the explained variance. Numbers between brackets (and grey shading) correspond to the proportion of spatial locations (SL) for which the difference between observed and permuted adjusted R^2 is larger than 0 with $p \leq 0.05$

S5.4 Locations with low adjusted R^2

Spatial neighborhood is unlikely to explain the observed relationship between factors if the adjusted R^2 for all spatial locatons is well above zero, if it does not overlap with the adjusted R^2 under random permutations, and if it yields a z-score above conventional values of significance. However, it might still be the case that the correlations we find are an artefact by spatial proximity. This is the case if the locations with particularly low adjusted R^2 (i.e. the lower tail of observed distributions in Figures S15-S26) cluster together. To rule this out, we plot those locations for which the adjusted R^2 is in the .05 percentile. Figures S30-S31 suggest that low-R^2 locations are randomly distributed in the polygons and do not mass up in one region alone.

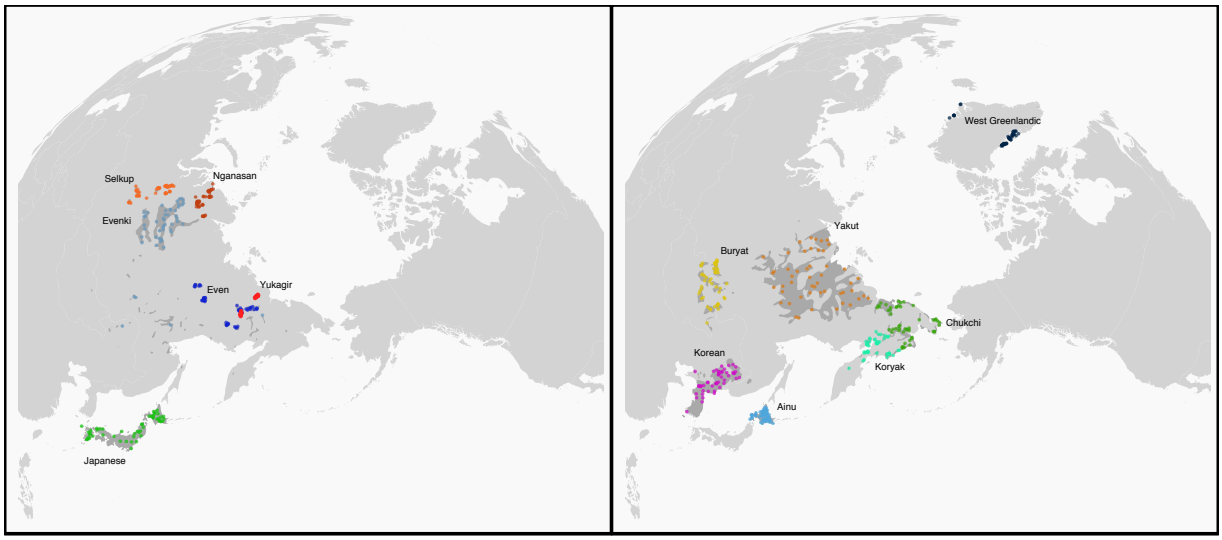

Figure S30: Location samples used for removing the influence of space in the partial RDA between Genetics (explanatory variable) and Grammar (response) with a low adjusted R^2

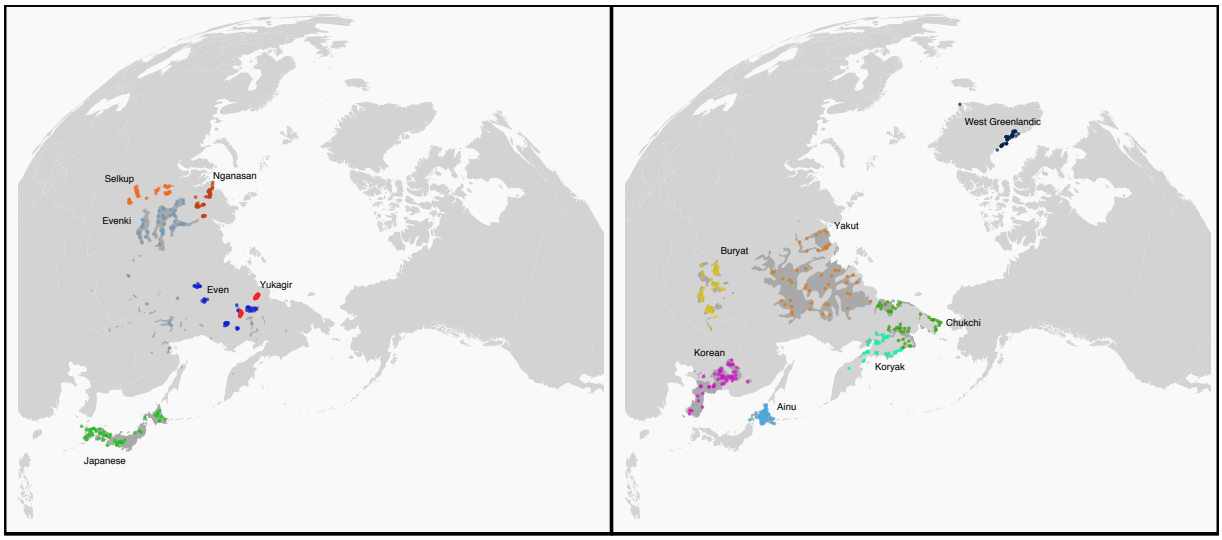

Figure S31: Point samples used for removing the influence of space in the partial RDA between Grammar (explanatory variable) and Genetics (response) with a low adjusted R^2

References

Abecasis, Goncalo R, Adam Auton, Lisa D Brooks, Mark a DePristo, Richard M Durbin, Robert E Handsaker, Hyun Min Kang, Gabor T Marth, and Gil a McVean. 2012. "An integrated map of genetic variation from 1,092 human genomes." *Nature* 491 (7422):56–65. https://doi.org/10.1038/nature11632.

Bickel, Balthasar, Johanna Nichols, Taras Zakharko, Alena Witzlack-Makarevich, Kristine Hildebrandt, Michael Rießler, Lennard Bierkandt, Fernando Zúñiga, and John B Lowe. 2017. *The AUTOTYP Typological Databases, Version 0.1.0*. GitHub [https://github.com/autotyp/autotyp-data/tree/0.1.0].

Bickel, Balthasar, and Taras Zakharko. 2018. *Recoding of Wals Online*. GitHub repository [https://github.com/IVS-UZH/WALS-recodings].

Borcard, Daniel, and Pierre Legendre. 2002. "All-Scale Spatial Analysis of Ecological Data by Means of Principal Coordinates of Neighbour Matrices." *Ecological Modelling* 153 (1-2). Elsevier:51–68.

Borcard, Daniel, Pierre Legendre, and Pierre Drapeau. 1992. "Partialling Out the Spatial Component of Ecological Variation." *Ecology* 73 (3). Wiley Online Library:1045–55.

Donohue, Mark, Rebecca Hetherington, James McElvenny, and Virginia Dawson. 2013. *World Phonotactics Database*. Canberra: Department of Linguistics, The Australian National University [http://phonotactics.anu.edu.au].

Dray, Stéphane, Pierre Legendre, and Pedro R Peres-Neto. 2006. "Spatial Modelling: A Comprehensive Framework for Principal Coordinate Analysis of Neighbour Matrices (Pcnm)." *Ecological Modelling* 196 (3-4). Elsevier:483–93.

Dryer, Matthew S., and Martin Haspelmath, eds. 2013. *WALS Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology [http://wals.info/].

Fedorova, Sardana A, Maere Reidla, Ene Metspalu, Mait Metspalu, Siiri Rootsi, Kristiina Tambets, Natalya Trofimova, et al. 2013. "Autosomal and uniparental portraits of the native populations of Sakha (Yakutia): implications for the peopling of Northeast Eurasia." *BMC Evolutionary Biology* 13 (January):127. https://doi.org/10.1186/1471-2148-13-127.

Hammarström, Harald, Robert Forkel, and Martin Haspelmath. 2017. *Glottolog 3.0*. Jena: Jena: Max Planck Institute for the Science of Human History.

Jinam, Timothy, Nao Nishida, Momoki Hirai, Shoji Kawamura, Hiroki Oota, Kazuo Umetsu, Ryosuke Kimura, et al. 2012. "The history of human populations in the Japanese Archipelago inferred from genome-wide SNP data with a special reference to the Ainu and the Ryukyuan populations." *Journal of Human Genetics* 57 (12):787–95. https://doi.org/10.1038/jhg.2012.114.

Josse, Julie, and François Husson. 2016. "missMDA: A Package for Handling Missing Values in Multivariate Data Analysis." *Journal of Statistical Software* 70 (1):1–31. https://doi.org/10.18637/jss.v070.i01.

Lazaridis, Iosif, Nick Patterson, Alissa Mittnik, Gabriel Renaud, Swapan Mallick, Karola Kirsanow, Peter H. Sudmant, et al. 2014. "Ancient human genomes suggest three ancestral populations for present-day Europeans." *Nature* 513 (7518):409–13. https://doi.org/10.1038/nature13673.

Legendre, Pierre, and Loic FJ Legendre. 2012. *Numerical Ecology*. Vol. 24. Elsevier.

Lê, Sébastien, Julie Josse, and François Husson. 2008. "FactoMineR: An R Package for Multivariate Analysis." *Journal of Statistical Software* 25:1–18.

Moran, Steven, Daniel McCloy, and Richard Wright, eds. 2014. *PHOIBLE Online*. Munich: Max Planck Digital Library [http://phoible.org/].

Rasmussen, Morten, Yingrui Li, Stinus Lindgreen, Jakob Skou Pedersen, Anders Albrechtsen, Ida Moltke, Mait Metspalu, et al. 2010. "Ancient human genome sequence of an extinct Palaeo-Eskimo." *Nature* 463 (7282):757–62. https://doi.org/10.1038/nature08835.

Simons, Gary F., and Charles D. Fennig. 2018. “Ethnologue: Languages of the World, Twenty-First Edition.” Dallas, Texas: SIL International. http://www.ethnologue.com.

Van Den Wollenberg, Arnold L. 1977. “Redundancy Analysis an Alternative for Canonical Correlation Analysis.” *Psychometrika* 42 (2). Springer:207–19.

Wichmann, Søren, André Müller, Annkathrin Wett, Viveka Velupillai, Julia Bischoffberger, Cecil H. Brown, Eric W. Holman, et al. 2015. *The ASJP Database (Version 16).* Repository at http://asjp.clld.org.