Supplementary Information 4 for *Cultural influences on word meanings revealed throughlarge-scale semantic alignment:* Cross-cultural analyses

Bill Thompson, Seán Roberts & Gary Lupyan

This document contains the supporting information on cross-cultural analyses for *Cultural influences on word meanings revealed throughlarge-scale semantic alignment*. Most of the contents are compiled R markdown documents, showing the R code for analysing the data. The full repository can be found here:
`https://github.com/seannyD/ImputeEACulturalDifferences`
The contents are as follows.

4.1 Main cross-cultural analysis (Wikipedia data)

Predicting semantic alignment by cultural similarity

Bill Thompson, Seán Roberts & Gary Lupyan

Contents

Introduction

We compare cultural distances between socieites with semantic alignment between societies, controlling for shared history in two ways.

The first test uses mixed effects modelling. The pairing of the language family of each language (according to Glottolog, Hammarstrom et al., 2018) is used as a random effect. That means that the model can capture the likelihood that two languages from the same language family (e.g. Indo-European) will be more similar to each other than two languages from different language families. The same is done with geographic area according to Autotyp (Nichols et al., 2013), which reflect areas of known linguistic contact. The model also included a fixed effect for for the number of lexical comparisons that went into the mean semantic alignment estimates (generally, more available comparisons indicate more possible comparisons, i.e. more similar languages).

The second and thrid test controls for history using distances from a phylogenetic tree. The tree comes from Bouckaert et al. (2012), which was estimated by comapring the gain and loss of cognates in the lexicon of Indo-European languages. Patristic distances between languages are used as a measure of historical distance between languages. Patristic distance is the distance between two leaves on the tree following the shortest path (which will go through the most recent common ancestor). The branch lengths in the tree are scaled to reflect time, so the shortest distance between two leaves on the tree indicates the total amount of independent evolution between two languages. An alternative measure of historical distance was obtained from data from the Automated Similarity Judgement Program database (ASJP, Wichmann, Holman & Brown, 2018). This is a databse of basic vocabulary in a common phonetic format. We used the distances as calculated by Jäger (2018) which essentially measure the average edit distance between langauges (the number of changes to turn one vocabulary into the other), accounting for the likelihood of historical changes between sound segments. The measure is similar to calculating distances between sequences of DNA. Both of these measures of historical distance are based on the lexicon, but do not use measures of the semantic meanings of words.

The second test uses simple and partial Mantel tests (Mantel, 1967 and e.g. Smouse, Long & Sokal, 1986, Legendre, 2000; Castellano & Balletto, 2002, Goslee, 2010), using the implementation in the R package *ecodist* (Goslee & Urban, 2007). A Mantel test is a nonparemetric test that uses permutation to assess the strength of the relationship between two distance matrices. It compares the correlation between the values from two distance matrices with the correlation produced when the values of one of the matrices is permuted. This allows it to account for the dependencies between the distances. Note that the Mantel test assumes a strict distance metric, which is not necessarily the case with this data (see also e.g. Harmon & Glor, 2010), but there are few other ways to deal with continuous pairwise distances.

Mantel tests were used to test the relationship between semantic alignment and historical proximity, and between semantic alignment and geographic proximity. Geographic proximity was measured as the great circle distance between the cultural centres of each language as defined in Glottolog (Hammarstrom et al., 2018). For the analysis within domains, partial Mantel tests were used to estimate the correlation between the semantic alignment and the cultural/geographic/historical proximity while partialling out the effect of the other two proximity measures.

The third test uses multiple regression on distance matrices (Lichstein, 2007). This is a regression approach which uses distance matrices as dependent and independent variables.

The results above were robustly replicated using the filtered data and also alternative sources for semantic alignment (common crawl, see file *AnalyseCorrelation_cc.pdf*). The correlation was not robust to all tests or for data derived from the subtitles dataset (see file *AnalyseCorrelation_subs.pdf*), possibly because there were only 20 languages available to analyse.

The final section looks at relationships between sub-domains. The first section describes how the cultural similarity measure was calculated.

Calculating cultural similarity

The aim is to produce a set of distances between societies based on their cultural traits. The Ethnographic Atlas (Murdock et al., 1999) is a database of (non-linguistic) cultural traits on many societies. For each variable, societies are assigned to one category (or value). For example, the variable 'EA011' classifies a society's norms for "Transfer of residence at marriage". Each society is assigned to one of the following groups: "Wife to husband's group", "Husband to wife's group" "Couple to either group", "Nonestablishment of a common household". The D-PLACE database (https://d-place.org/, Kirby et al., 2016) links societies in the Ethnogrpahic atlas to the languages they speak (through the Glottolog ID, Hammarstrom et al., 2018). D-PLACE also provides the data in an updated format, so we use this as our primary data source.

However, there is a lot of missing data in the Ethnographic Atlas (about 25% in the whole dataset), which means that distances can't be computed easily. One approach is to impute the missing data (guess their values based on existing data). It's unlikely that any imputation method will be completely accurate, but for our purposes we don't need to be accurate, just *unbiased*. That is, the imputed values should not bias the estimates of the distances between cultures.

In this case, we use multiple imputation: calculating many possible alternative imputations and taking the mean distances over all imputations.

Imputing missing values in the Ethnographic Atlas

We use the imputation package `mice` for R (van Buuren & Groothuis-Oudshoorn, 2011). We compared various settings of the imputation method, and found that using classification and regression trees (CART) with the standard parameters produced the best results. CART works by building a decision tree: an optimal set of yes-no questions to ask about predictor variables in order to guess the value of a target variable. The tree divides the data into partitions which look similar. The algorithm works out which partition a missing data point would belong to, then samples the target variable distribution from that partition. To account for historical relationships, we included language family according to Glottolog and geographic area according to Autotyp as additional factors on which the imputation process could draw.

We ran CART multiple imputation on the Ethnographic Atlas. We excluded population size, one more variable that was coded for less than 33% of societies, and any societies that had fewer than 33% variables coded. This left 92 variables for 962 languages with 16% missing data.

We tested the imputation by taking the full Ethnographic Atlas data, creating some new missing values in random places and then re-imputing those missing values. We can then asses how accurate the imputation was for those values. Since the main analysis would only be using a small sub-set of the data, it is important to assess performance on these in particular, rather than the entire set of languages. Missing data was only inserted for languages in the main analysis of semantic alignment below. CART imputation guessed the correct value of missing data 74% of the time (average over 100 imputations). This is reasonably good, considering that most variables have between 4 and 8 possible values (median = 6). For example, this is 8.6 standard deviations better than choosing randomly (accuracy = 19%) and 5.6 standard deviations better than sampling from the known distribution of the target variable (accuracy = 37% on the same missing data). This is not good enough to use in analyses that look at individual traits, but serves our purposes to estimate overall distances between langauges.

We produced 100 imputation sets with the final settings. These were then used to create distance matrices using Gower distance between discrete traits (mean correlation between sets r = 0.94, estimates of distance vary by around 2% on average). The final distance matrix was the mean of each of the 100 distance matrices. Distances were also calculate for sub-domains of the data.

The full scripts and data can be found at https://github.com/seannyD/ImputeEACulturalDifferences. Reviewers can follow this link: https://figshare.com/s/06378bc59a771d28b1d0

Load libraries

```
library(ape)
library(ecodist)
library(lme4)
library(sjPlot)
library(ggplot2)
library(igraph)
library(lattice)
library(dplyr)
```

Parameters (using data from Northuralex and Wikipedia, k=100, unfiltered):

```
datasetName = "wikipedia-main"
lingDistancesFile = "../data/FAIR/nel-wiki-k100-alignments-by-language-pair.csv"
lingDistancesFileNK = "../data/FAIR/nel-wiki-k100-alignments-by-language-pair-without-kinship.csv"
lingDistancesByDomainFile = "../results/EA_distances/nel-wiki-k100_with_ling.csv"
# (generated by ../processing/combineCultAndLingDistances.R)
```

All domains

Load data

Read the cultural distances:

```
cult = read.csv("../results/EA_distances/CulturalDistances_Long.csv", stringsAsFactors = F)
names(cult) = c("l1","l2","cult.dist")
```

Add language family:

```
l = read.csv("../data/FAIR_langauges_glotto_xdid.csv", stringsAsFactors = F)
g = read.csv("../data/glottolog-languoid.csv/languoid.csv", stringsAsFactors = F)
l$family = g[match(l$glotto,g$id),]$family_pk
l$family = g[match(l$family,g$pk),]$name
```

Read the semantic distances

```
ling = read.csv(lingDistancesFile, stringsAsFactors = F)
```

There are very few possible comparisons for Slovenian and Northern Sami, so we'll remove these:

```
ling = ling[!(ling$l1=="se" | ling$l2 == "se"),]
ling = ling[!(ling$l1=="sl" | ling$l2 == "sl"),]
```

Combine the lingusitic and cultural distances. Note that we flip the cultural measure from a distance measure to a similarity measure.

```
cult$l1.iso2 = l[match(cult$l1,l$Language2),]$iso2
cult$l2.iso2 = l[match(cult$l2,l$Language2),]$iso2

fairisos = unique(c(ling$l1,ling$l2))
cultisos = unique(c(cult$l1.iso2, cult$l2.iso2))

cult = cult[(cult$l1.iso2 %in% fairisos) & (cult$l2.iso2 %in% fairisos),]
ling = ling[(ling$l1 %in% cultisos) & (ling$l2 %in% cultisos),]

matches = sapply(1:nrow(ling), function(i){
  which(cult$l1.iso2==ling$l1[i] & cult$l2.iso2==ling$l2[i])
})

ling$cult.dist = cult[matches,]$cult.dist
# Flip
ling$cult.dist = 1 - ling$cult.dist
# Scale
ling$cult.dist.center = scale(ling$cult.dist)
cdc.s = attr(ling$cult.dist.center,"scaled:scale")
cdc.c = attr(ling$cult.dist.center,"scaled:center")
ling$cult.dist.center = as.numeric(ling$cult.dist.center)
ling$comparison_count.center =
  scale(ling$comparison_count)

ling$family1 = l[match(ling$l1, l$iso2),]$family
ling$family2 = l[match(ling$l2, l$iso2),]$family
l[l$Language=="Arabic",]$autotyp.area= "Greater Mesopotamia"
l[l$Language=="Persian",]$autotyp.area= "Greater Mesopotamia"
ling$area1 = l[match(ling$l1, l$iso2),]$autotyp.area
```

```
ling$area2 = l[match(ling$l2, l$iso2),]$autotyp.area


fgroup = cbind(ling$family1,ling$family2)
fgroup = apply(fgroup,1,sort)
ling$family.group = apply(fgroup,2,paste,collapse=":")
agroup = cbind(ling$area1,ling$area2)
agroup = apply(agroup,1,sort)
ling$area.group = apply(agroup,2,paste,collapse=":")

ling$rho.center = scale(ling$local_alignment)
```

Each observation is now assocaited with a language family pair:

```
head(ling[,c("l1","l2","local_alignment",'family.group')])
```

```
##     l1  l2 local_alignment               family.group
## 7   ja  ab      0.01930414       Abkhaz-Adyge:Japonic
## 8   ab  zh      0.02225169  Abkhaz-Adyge:Sino-Tibetan
## 10  cv xal      0.02765860            Mongolic:Turkic
## 11 xal  ja      0.02832668           Japonic:Mongolic
## 12 xal  zh      0.02895876      Mongolic:Sino-Tibetan
## 14  bn  ab      0.03192066 Abkhaz-Adyge:Indo-European
```

And the same is true for area:

```
tail(ling[,c("l1","l2","local_alignment",'area.group')])
```

```
##      l1 l2 local_alignment            area.group
## 2522 fr es       0.3936442         Europe:Europe
## 2524 cs uk       0.4023323     Europe:Inner Asia
## 2528 cs ru       0.4082099     Europe:Inner Asia
## 2529 be ru       0.4129814 Inner Asia:Inner Asia
## 2532 uk be       0.4276664 Inner Asia:Inner Asia
## 2535 uk ru       0.5079911 Inner Asia:Inner Asia
```

Number of observations:

```
# Number of datapoints:
nrow(ling)
```

```
## [1] 731
```

```
# Number of unique languages:
length(unique(unlist(ling[,c("l1","l2")])))
```

```
## [1] 39
```

```
# Number of unique langauge families:
uniqueFamilies = unique(unlist(ling[,c("family1","family2")]))
length(uniqueFamilies)
```

```
## [1] 10
```

```
# Number of unique areas:
uniqueAreas = unique(unlist(ling[,c("area1","area2")]))
length(uniqueAreas)
```

```
## [1] 6
```

Cross-over between language famlies and areas:

```
tx = data.frame(lang= c(ling$l1,ling$l2),
           fam = c(ling$family1,ling$family2),
           area= c(ling$area1,ling$area2))
tx = tx[!duplicated(tx),]
table(tx$fam,tx$area)
```

```
##
##                 Europe Greater Mesopotamia Indic Inner Asia N Coast Asia
##   Abkhaz-Adyge       0                    1     0          0            0
##   Afro-Asiatic       0                    1     0          0            0
##   Dravidian          0                    0     3          0            0
##   Indo-European     11                    2     1          5            0
##   Japonic            0                    0     0          0            1
##   Koreanic           0                    0     0          0            1
##   Mongolic           0                    0     0          1            0
##   Sino-Tibetan       0                    0     0          0            0
##   Turkic             0                    1     0          5            0
##   Uralic             1                    0     0          4            0
##
##                 Southeast Asia
##   Abkhaz-Adyge                0
##   Afro-Asiatic                0
##   Dravidian                   0
##   Indo-European               0
##   Japonic                     0
##   Koreanic                    0
##   Mongolic                    0
##   Sino-Tibetan                1
##   Turkic                      0
##   Uralic                      0
```

LMER models

Mixed effects model, predicting semantic alignment from cultural similarity, with random intercept for family and area and random slope for cultural similarity for family and area.

We start with a null model with random intercepts for family and area, and random slopes for cultural similarity by both. We add a fixed effect of the number of comparisons made for each datapoint (number of concepts that were available to compare). Then we add a fixed effect of cultural similarity

```
m0 = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m0.5 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m1 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    cult.dist.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
an1 = anova(m0,m0.5,m1)
```

```
## refitting model(s) with ML (instead of REML)
```

```
an1
```

```
## Data: ling
## Models:
## m0: rho.center ~ 1 + (1 + cult.dist.center | family.group) + (1 +
## m0:     cult.dist.center | area.group)
## m0.5: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## m0.5:     family.group) + (1 + cult.dist.center | area.group)
## m1: rho.center ~ 1 + comparison_count.center + cult.dist.center +
## m1:     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
## m1:     area.group)
##      Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## m0    8 1654.6 1691.3 -819.30   1638.6
## m0.5  9 1293.0 1334.3 -637.50   1275.0 363.597      1  < 2.2e-16 ***
## m1   10 1278.4 1324.4 -629.22   1258.4  16.564      1  4.704e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cultural similarity is not significantly correlated with semantic alignment. Here are the model estimates:

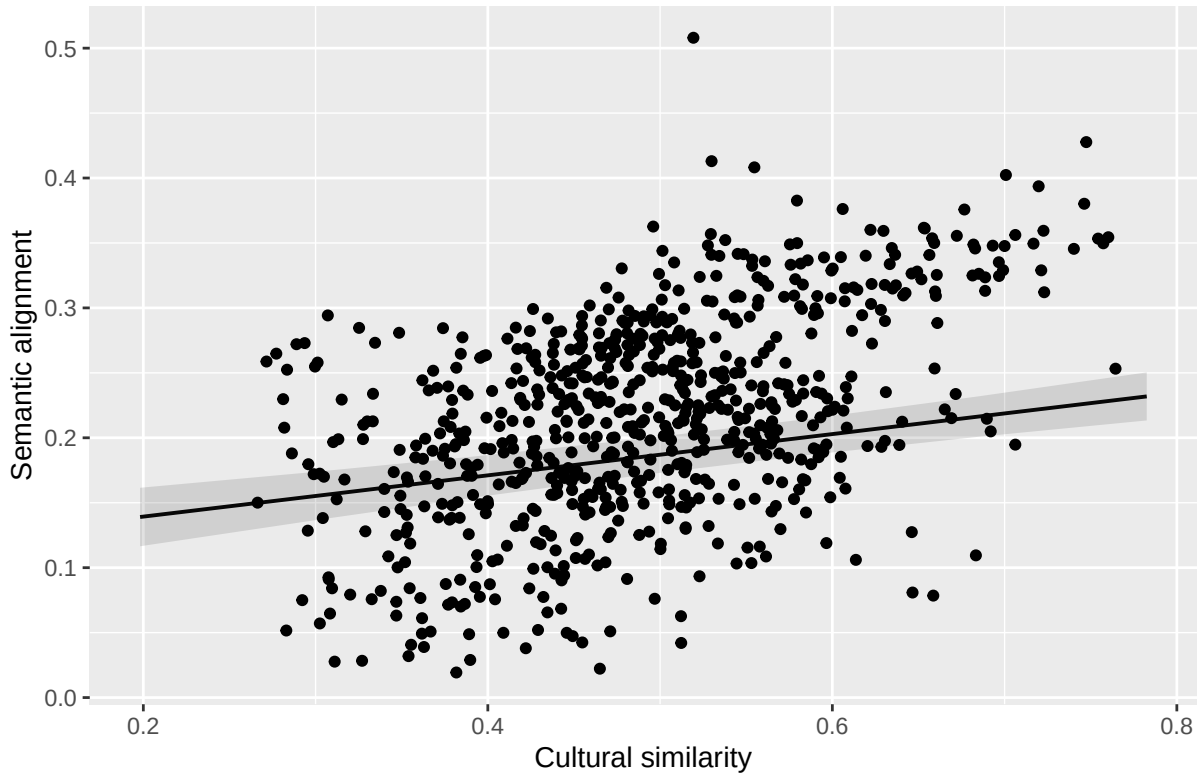
```
summary(m1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ 1 + comparison_count.center + cult.dist.center +
##     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
##     area.group)
##    Data: ling
##
## REML criterion at convergence: 1271.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6249 -0.6167  0.1103  0.6571  4.7402
##
## Random effects:
##  Groups       Name             Variance  Std.Dev. Corr
##  family.group (Intercept)      0.1612630 0.40158
##               cult.dist.center 0.0001817 0.01348  1.00
##  area.group   (Intercept)      0.0510850 0.22602
##               cult.dist.center 0.0036658 0.06055  -1.00
##  Residual                      0.2885416 0.53716
## Number of obs: 731, groups:  family.group, 48; area.group, 20
##
## Fixed effects:
##                         Estimate Std. Error t value
## (Intercept)             -0.39270    0.09073  -4.328
## comparison_count.center  0.61196    0.02688  22.770
## cult.dist.center         0.19678    0.03275   6.008
##
## Correlation of Fixed Effects:
##             (Intr) cmpr_.
## cmprsn_cnt.  0.090
## clt.dst.cnt -0.194 -0.201
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Plot the estimates, rescaling the variables back to the original units:

```
trans = function(X){
  X * attr(ling$rho.center,"scaled:scale") +
  attr(ling$rho.center,"scaled:center")
}

gx = plot_model(m1,'pred',terms='cult.dist.center')
gx$data$predicted = trans(gx$data$predicted)
gx$data$conf.low = trans(gx$data$conf.low)
gx$data$conf.high = trans(gx$data$conf.high)
gx$data$x = gx$data$x *
  cdc.s +cdc.c
gx = gx + #coord_cartesian(ylim=c(0,0.5),
          #                xlim=c(0.15,0.85)) +
  xlab("Cultural similarity") +
```

```
  ylab("Semantic alignment") +
  ggtitle("") +
  geom_point(data=ling,aes(x=cult.dist,y=local_alignment))
gx
```



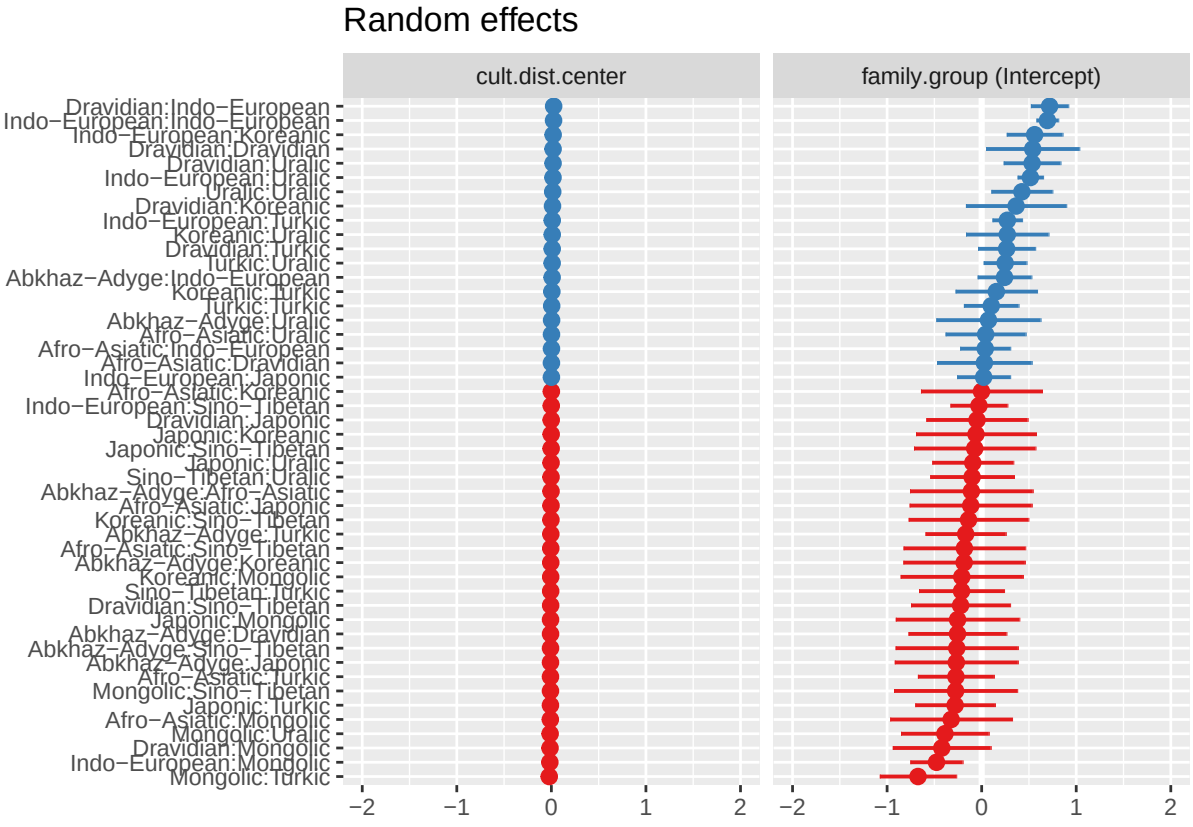

```
pdf(paste0("../results/stats/",datasetName,"/CulturalDistance_Rho_Graph.pdf"),
    height=2.5, width=2.5)
gx
dev.off()
```

```
## pdf
##   2
```

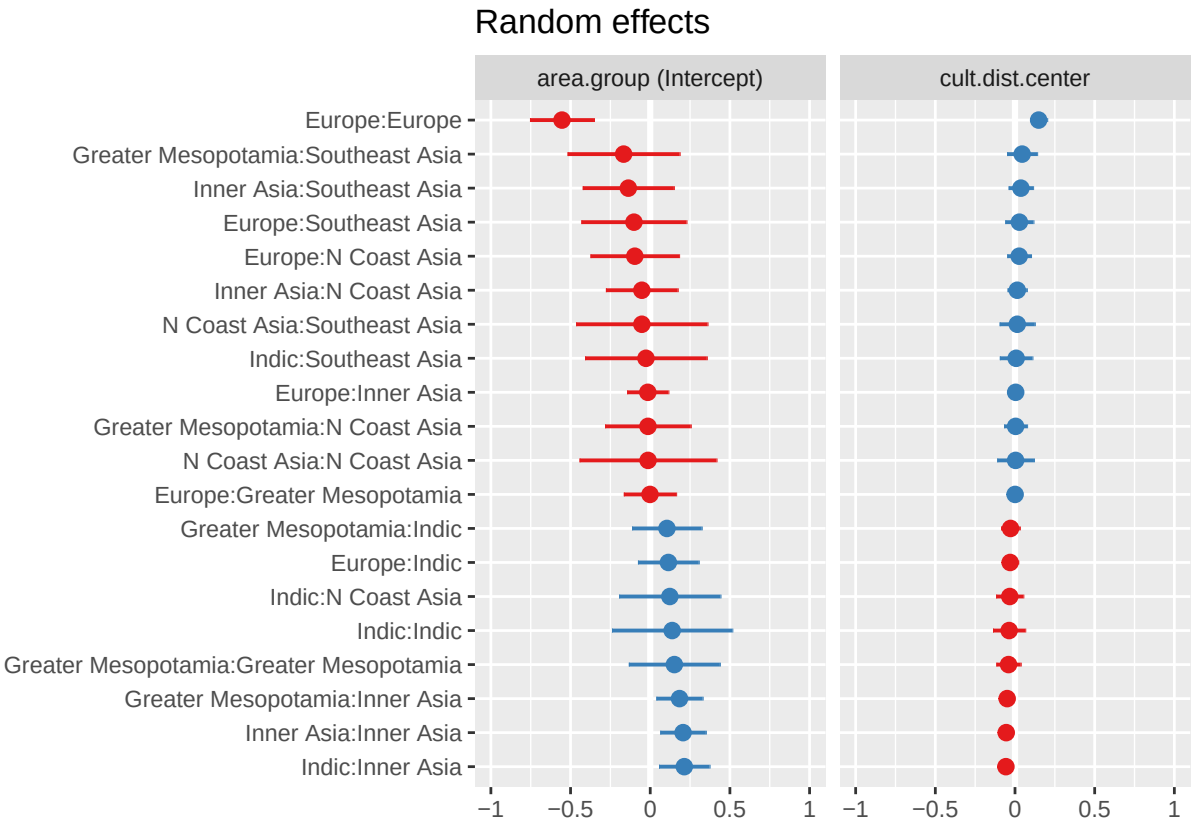
Plot the random effects:

```
plot_model(m1,'re', sort.est = "cult.dist.center")
```

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


Random effects
cult.dist.center
family.group (Intercept)
Dravidian:Indo−European
Indo−European:Indo−European
Indo−European:Koreanic
Dravidian:Dravidian
Dravidian:Uralic
Indo−European:Uralic
Uralic:Uralic
Dravidian:Koreanic
Indo−European:Turkic
Koreanic:Uralic
Dravidian:Turkic
Turkic:Uralic
Abkhaz−Adyge:Indo−European
Koreanic:Turkic
Turkic:Turkic
Abkhaz−Adyge:Uralic
Afro−Asiatic:Uralic
Afro−Asiatic:Indo−European
Afro−Asiatic:Dravidian
Indo−European:Japonic
Afro−Asiatic:Koreanic
Indo−European:Sino−Tibetan
Dravidian:Japonic
Japonic:Koreanic
Japonic:Sino−Tibetan
Japonic:Uralic
Sino−Tibetan:Uralic
Abkhaz−Adyge:Afro−Asiatic
Afro−Asiatic:Japonic
Koreanic:Sino−Tibetan
Abkhaz−Adyge:Turkic
Afro−Asiatic:Sino−Tibetan
Abkhaz−Adyge:Koreanic
Koreanic:Mongolic
Sino−Tibetan:Turkic
Dravidian:Sino−Tibetan
Japonic:Mongolic
Abkhaz−Adyge:Dravidian
Abkhaz−Adyge:Sino−Tibetan
Abkhaz−Adyge:Japonic
Afro−Asiatic:Turkic
Mongolic:Sino−Tibetan
Japonic:Turkic
Afro−Asiatic:Mongolic
Mongolic:Uralic
Dravidian:Mongolic
Indo−European:Mongolic
Mongolic:Turkic
−2
−1
0
1
2

```
##
## [[2]]
```


Random effects
area.group (Intercept)
cult.dist.center
Europe:Europe
Greater Mesopotamia:Southeast Asia
Inner Asia:Southeast Asia
Europe:Southeast Asia
Europe:N Coast Asia
Inner Asia:N Coast Asia
N Coast Asia:Southeast Asia
Indic:Southeast Asia
Europe:Inner Asia
Greater Mesopotamia:N Coast Asia
N Coast Asia:N Coast Asia
Europe:Greater Mesopotamia
Greater Mesopotamia:Indic
Europe:Indic
Indic:N Coast Asia
Indic:Indic
Greater Mesopotamia:Greater Mesopotamia
Greater Mesopotamia:Inner Asia
Inner Asia:Inner Asia
Indic:Inner Asia
−1
−0.5
0
0.5
1
−1
−0.5
0
0.5
1

Without Kinship data

The analyses below show that the strongest relationship is with Kinship. Here we run the analysis as above, but using semantic distances computed without concepts that relate to kinship. Note that the local alignment values correlate with r $>$ 0.99.

Code for constructing the data is hidden, but it is the same as above and available in the Rmd file:

Run the lmer models:

```
m0NK = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = lingNK
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m0.5NK = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = lingNK
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m1NK = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    cult.dist.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = lingNK
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
anova(m0NK,m0.5NK,m1NK)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: lingNK
## Models:
## m0NK: rho.center ~ 1 + (1 + cult.dist.center | family.group) + (1 +
## m0NK:     cult.dist.center | area.group)
## m0.5NK: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## m0.5NK:     family.group) + (1 + cult.dist.center | area.group)
## m1NK: rho.center ~ 1 + comparison_count.center + cult.dist.center +
## m1NK:     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
## m1NK:     area.group)
##        Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## m0NK    8 1654.6 1691.3 -819.30   1638.6
## m0.5NK  9 1293.0 1334.3 -637.50   1275.0 363.597      1  < 2.2e-16 ***
## m1NK   10 1278.4 1324.4 -629.22   1258.4  16.564      1  4.704e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(m1NK)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ 1 + comparison_count.center + cult.dist.center +
##     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
##     area.group)
##    Data: lingNK
##
## REML criterion at convergence: 1271.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6249 -0.6167  0.1103  0.6571  4.7402
##
## Random effects:
##  Groups       Name             Variance  Std.Dev. Corr
##  family.group (Intercept)      0.1612630 0.40158
##               cult.dist.center 0.0001817 0.01348  1.00
##  area.group   (Intercept)      0.0510850 0.22602
##               cult.dist.center 0.0036658 0.06055  -1.00
##  Residual                      0.2885416 0.53716
## Number of obs: 731, groups:  family.group, 48; area.group, 20
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)             -0.39270    0.09073  -4.328
## comparison_count.center  0.61196    0.02688  22.770
## cult.dist.center         0.19678    0.03275   6.008
##
## Correlation of Fixed Effects:
##             (Intr) cmpr_.
## cmprsn_cnt.  0.090
## clt.dst.cnt -0.194 -0.201
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

MRM

Use multiple regression on distance matrices (Lichstein, 2007) to do the same test as above. The code below uses the `igraph` package to make an undirected graph from the long format with `local_alignment` as the edge weights, then output a matrix of adjacencies.

```
# Use graph method to make distance matrix
grph <- graph.data.frame(ling[,c("l1",'l2','local_alignment')], directed=FALSE)
# add value as a weight attribute
ling.m = get.adjacency(grph, attr="local_alignment", sparse=FALSE)
rownames(ling.m) = l[match(rownames(ling.m),l$iso2),]$Language2
colnames(ling.m) = l[match(colnames(ling.m),l$iso2),]$Language2
# Same for comparison_count.center
grph <- graph.data.frame(ling[,c("l1",'l2','comparison_count')], directed=FALSE)
# add value as a weight attribute
cc.m = get.adjacency(grph, attr="comparison_count", sparse=FALSE)
rownames(cc.m) = l[match(rownames(cc.m),l$iso2),]$Language2
colnames(cc.m) = l[match(colnames(cc.m),l$iso2),]$Language2
```

Load the cultural distances as a matrix.

```
cult.m = read.csv("../results/EA_distances/CulturalDistances.csv", stringsAsFactors = F)
rownames(cult.m) = cult.m[,1]
cult.m = cult.m[,2:ncol(cult.m)]
cult.m = as.matrix(cult.m)
# Flip cultural value to distance
cult.m = 1-cult.m
mx = match(rownames(ling.m),rownames(cult.m))
cult.m = cult.m[mx,mx]
colnames(cult.m) = rownames(cult.m)
```

Make a matrix of same/different language family (1=different):

```
# Same/different matrix for language family
family.matrix = l[match(rownames(ling.m),l$Language),]$family
family.matrix = outer(family.matrix,family.matrix,"!=") *1
```

Load ASJP distances for second test:

```
asjp = readRDS("../data/ASJP/asjp17-dists_FAIR.RData")
ling.m.glotto = l[match(rownames(cult.m),l$Language2),]$glotto
ling.m.glotto = ling.m.glotto[ling.m.glotto %in% rownames(asjp)]
asjp.m = asjp[ling.m.glotto,ling.m.glotto]
asjp.lang.names = l[match(rownames(asjp.m),l$glotto),]$Language2
# Matrices for second analysis with asjp
ling.m2 = ling.m[asjp.lang.names,asjp.lang.names]
cult.m2 = cult.m[asjp.lang.names,asjp.lang.names]
cc.m2 =  cc.m[asjp.lang.names,asjp.lang.names]
```

Load the geographic distances:

```
geoDist = read.csv("../data/GeographicDistances.csv",stringsAsFactors = F)
geoDist.m = as.matrix(geoDist)
geoDist.m = geoDist.m[!is.na(geoDist.m[,1]),!is.na(geoDist.m[1,])]
# Convert to log distance in thousand km
geoDist.m = log10(geoDist.m/1000)
geoDist.m[is.infinite(geoDist.m)] = 0
```

```
colnames(geoDist.m) = gsub("\\."," ",colnames(geoDist.m))
rownames(geoDist.m) = colnames(geoDist.m)
geoDist.m1 = geoDist.m[rownames(ling.m),rownames(ling.m)]
geoDist.m2 = geoDist.m[rownames(ling.m2),rownames(ling.m2)]
```

Some language pairs do not have observed semantic alignments (10 out of 741, 1.3%). In this case, we impute the mean:

```
# For missing comparisons, impute the mean:
# (there are no zero values in the local alignment data)
ling.m[ling.m==0] = mean(ling$local_alignment)
diag(ling.m) = 0
ling.m2[ling.m2==0] = mean(ling.m2[ling.m2!=0])
diag(ling.m2) = 0
```

Center and scale values:

```
ling.m =  matrix(scale(as.vector(ling.m)),nrow=nrow(ling.m))
cc.m =  matrix(scale(as.vector(cc.m)),nrow=nrow(cc.m))
cult.m = matrix(scale(as.vector(cult.m)),nrow=nrow(cult.m))
geoDist.m1 = matrix(scale(as.vector(geoDist.m1)),nrow=nrow(geoDist.m1))

asjp.m = matrix(scale(as.vector(asjp.m)),nrow=nrow(asjp.m))
ling.m2 = matrix(scale(as.vector(ling.m2)),nrow=nrow(ling.m2))
cc.m2 =  matrix(scale(as.vector(cc.m2)),nrow=nrow(cc.m2))
cult.m2 = matrix(scale(as.vector(cult.m2)),nrow=nrow(cult.m2))
geoDist.m2 = matrix(scale(as.vector(geoDist.m2)),nrow=nrow(geoDist.m2))
```

Run the MRM model, predicting semantic alignment by cultural distance, controlling for family distance, geographic ditance, and the comparison count (number of observations). Here, the family distance between two languages is just whether they are part of the same family. Note that this does not take into account particular values for particular families, nor the random slopes within families.

```
set.seed(1282)
ecodist::MRM(as.dist(ling.m) ~
               as.dist(cult.m) +
               as.dist(family.matrix) +
               as.dist(geoDist.m1)  +
               as.dist(cc.m),nperm = 10000)
```

```
## $coef
##                          as.dist(ling.m)   pval
## Int                           0.22588260 0.0512
## as.dist(cult.m)               0.27484467 0.0114
## as.dist(family.matrix)       -0.21726450 0.1267
## as.dist(geoDist.m1)          -0.05699071 0.4481
## as.dist(cc.m)                 0.56172533 0.0001
##
## $r.squared
##        R2      pval
## 0.5698601 0.0001000
##
## $F.test
##        F   F.pval
## 243.7678   0.0001
```

Semantic alignment is significantly correlated with cultural distance.

In the result above, geographic distance is not correlated with semantic distance. Geographic distance turns out to be moderately correlated with cultural distance:

```
ecodist::MRM(as.dist(geoDist.m1) ~
               as.dist(cult.m),
             nperm = 10000)
```

```
## $coef
##                   as.dist(geoDist.m1)   pval
## Int                        -0.02111026 0.7707
## as.dist(cult.m)            -0.50170516 0.0001
##
## $r.squared
##        R2      pval
## 0.1529605 0.0001000
##
## $F.test
##        F   F.pval
## 133.4504   0.0001
```

Even when testing for non-linear geographic effects, the main result still holds:

```
ecodist::MRM(as.dist(ling.m) ~
               as.dist(cult.m) +
               as.dist(family.matrix) +
               as.dist(geoDist.m1)  +
               as.dist(geoDist.m1^2)  +
```

```
               as.dist(geoDist.m1^3)  +
               as.dist(cc.m),nperm = 10000)
```

```
## $coef
##                          as.dist(ling.m)   pval
## Int                          0.208052336 0.0865
## as.dist(cult.m)              0.272562935 0.0134
## as.dist(family.matrix)      -0.229500368 0.1162
## as.dist(geoDist.m1)         -0.018785241 0.8384
## as.dist(geoDist.m1^2)        0.021357131 0.7142
## as.dist(geoDist.m1^3)       -0.006518993 0.7250
## as.dist(cc.m)                0.563554206 0.0001
##
## $r.squared
##        R2      pval
## 0.5730241 0.0001000
##
## $F.test
##        F   F.pval
## 164.1777   0.0001
```

Below, we run the same test, but using average string distances in basic vocabulary from the ASJP (Wichmann, Holman & Brown, 2018) as controls for history. We used the distances as calculated in Jäger (2018), which used them to construct historical phylogenies.

```
ecodist::MRM(as.dist(ling.m2) ~
               as.dist(cult.m2) +
               as.dist(asjp.m) +
               as.dist(geoDist.m2) +
               as.dist(cc.m2),nperm = 10000)
```

```
## $coef
##                        as.dist(ling.m2)   pval
## Int                          0.10125375 0.0005
## as.dist(cult.m2)             0.26218752 0.0235
## as.dist(asjp.m)             -0.24422112 0.0001
## as.dist(geoDist.m2)         -0.02893256 0.7035
## as.dist(cc.m2)               0.56824024 0.0001
##
## $r.squared
##        R2      pval
## 0.5782528 0.0001000
##
## $F.test
##        F   F.pval
## 214.2326   0.0001
```

Mantel tests

Read the historical distances for Indo-European, based on the phylogenetic distances.

Data prep

The geographic distances are loaded above (from "../data/GeographicDistances.csv").

Load historical distances (Indo-European tree patristic distances):

```
hist = read.csv("../data/trees/IndoEuropean_historical_distances.csv", stringsAsFactors = F)
hist = hist[!duplicated(hist[,1]),!duplicated(hist[,1])]
rownames(hist) = hist[,1]
hist = hist[,2:ncol(hist)]
hist.m = as.matrix(hist)
colnames(hist.m) = rownames(hist.m)
hist.m = hist.m/max(hist.m)
```

Read the cultural distance as a matrix:

```
cult.m = read.csv("../results/EA_distances/CulturalDistances.csv", stringsAsFactors = F)
rownames(cult.m) = cult.m[,1]
cult.m = cult.m[,2:ncol(cult.m)]
```

Flip the cultural distance into a cultural similarity measure:

```
cult.m = 1-cult.m
```

Convert the semantic alignment to a matrix and impute the missing values with the mean. Note that in the final selection of langauges excludes any imputed values, but we perform the imputation just to be safe:

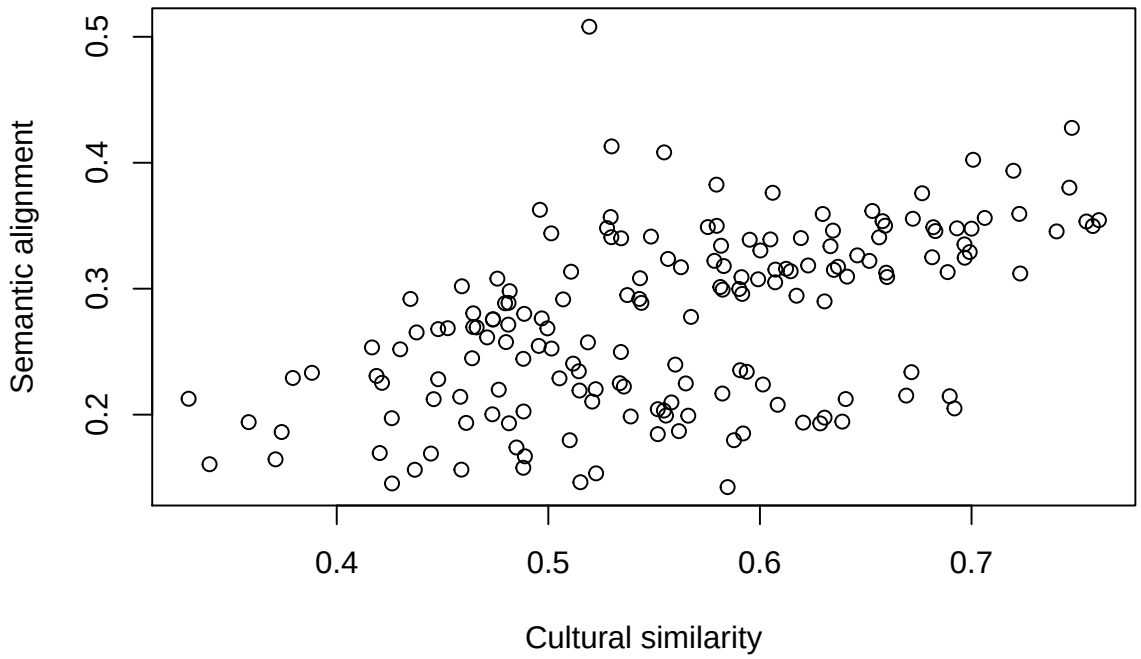
```
grph <- graph.data.frame(ling[,c("l1",'l2','local_alignment')], directed=FALSE)
# add value as a weight attribute
ling.m = get.adjacency(grph, attr="local_alignment", sparse=FALSE)
rownames(ling.m) = l[match(rownames(ling.m),l$iso2),]$Language2
colnames(ling.m) = l[match(colnames(ling.m),l$iso2),]$Language2
# For missing comparisons, impute the mean:
# (there are no zero values in the local alignment data)
ling.m[ling.m==0] = mean(ling$local_alignment)
diag(ling.m) = 0
```

Match the distance matrices

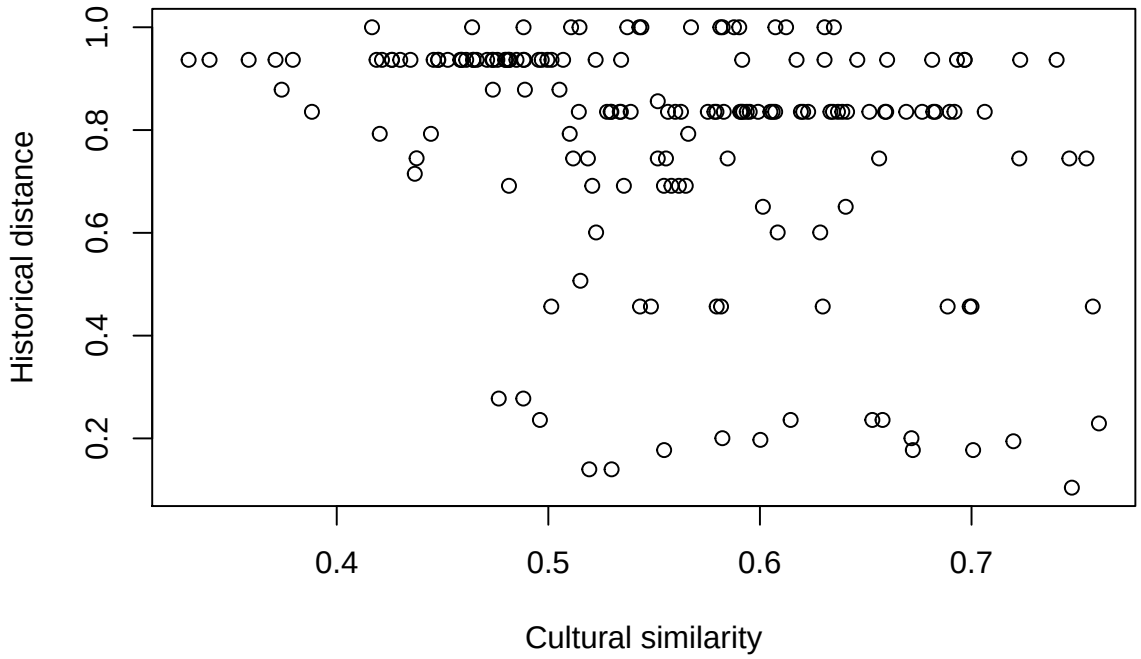
```
in.analysis = intersect(rownames(ling.m),rownames(cult.m))
in.analysis = intersect(in.analysis, rownames(hist.m))
cult.m2 = cult.m[in.analysis,in.analysis]
ling.m2 = ling.m[in.analysis,in.analysis]
hist.m2 = hist.m[in.analysis,in.analysis]
geo.m2 = geoDist.m[in.analysis,in.analysis]
```

Note that there are only 19 languages with data on lingusitic, cultural and historical distance. This is becasue the historical distances are derived from a tree of Indo-European languages (there are currently no reliable phylogentic trees constructed from cognates that span different language families). The languages in this test include: Albanian, Armenian, Belarusian, Bengali, Bulgarian, Czech, Dutch, English, French, Greek, Icelandic, Irish, Latin, Latvian, Lithuanian, Ossetian, Russian, Spanish, Ukrainian.

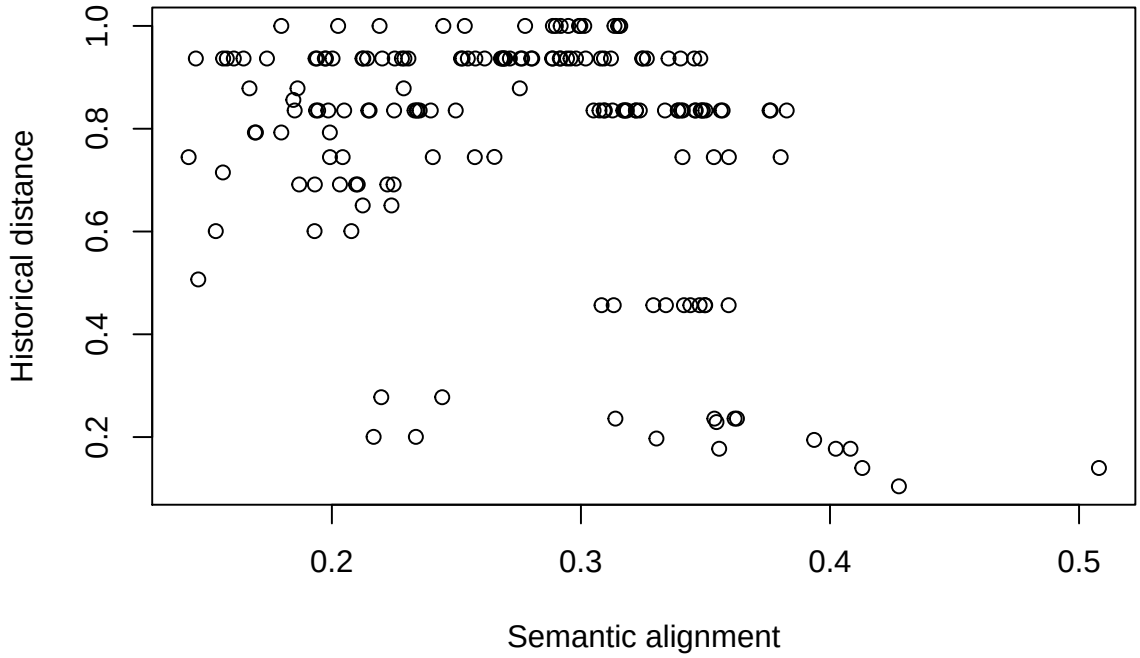
```
plot(as.dist(cult.m2),as.dist(ling.m2),
     xlab="Cultural similarity",
     ylab="Semantic alignment")
```

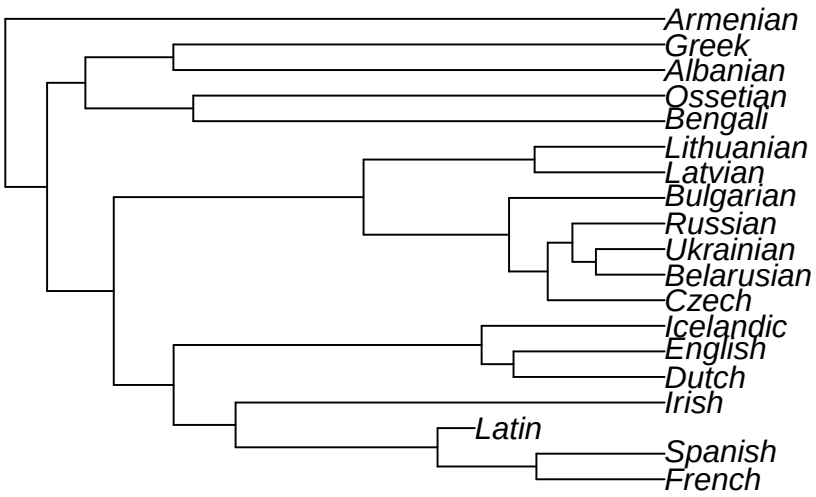



```
plot(as.dist(cult.m2),as.dist(hist.m2),
     xlab="Cultural similarity",
     ylab="Historical distance")
```




```
plot(as.dist(ling.m2),as.dist(hist.m2),
     xlab="Semantic alignment",
     ylab="Historical distance")
```


Historical distance
1.0
0.8
0.6
0.4
0.2
0.2
0.3
0.4
0.5
Semantic alignment


Armenian
Greek
Albanian
Ossetian
Bengali
Lithuanian
Latvian
Bulgarian
Russian
Ukrainian
Belarusian
Czech
Icelandic
English
Dutch
Irish
Latin
Spanish
French

Tests

The results of the test list the following measures:

- mantelr: Mantel correlation coefficient.
- pval1: one-tailed p-value (null hypothesis: r <= 0).
- pval2: one-tailed p-value (null hypothesis: r >= 0).
- pval3: two-tailed p-value (null hypothesis: r = 0).
- llim: lower confidence limit for r.
- ulim: upper confidence limit for r.

```
set.seed(1498)
```

Run tests between each pair of measures.

```
distms = list("Cultrual"= cult.m2,
              "Linguistic" = ling.m2,
              "Historical" = hist.m2,
              "Geographic" = geo.m2)
for(i in 1:3){
  for(j in (i+1):4){
    var1 = names(distms)[i]
    var2 = names(distms)[j]
    print(paste("Correlation between",
                var1,"and",var2))
    stat = ecodist::mantel(as.dist(distms[[i]]) ~
                as.dist(distms[[j]]),
                nperm = 100000)
    print(stat)
    stat = round(stat,2)
    pval = round(min(c(stat[2],stat[3])),3)
    if(pval==0){pval = "$<$ 0.001"}
    stat2 = sprintf("$r$ = %s, 95\\%% CI = [%s,%s], one-tailed $p$ = %s",
      round(stat[1],3),
      round(stat[5],3),
      round(stat[6],3),
      pval)
    stat2 = gsub("0\\.",".",stat2)
    cat(stat2,file=
          paste0("../results/stats/tex/Mantel",var1,"Vs",var2,"Distance.tex"))
  }
}
```

```
## [1] "Correlation between Cultrual and Linguistic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
##  0.5243289  0.0050000  0.9950100  0.0050300  0.3796035  0.6586819
## [1] "Correlation between Cultrual and Historical"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.3243830  0.9871000  0.0129100  0.0138900 -0.4402666 -0.2385575
## [1] "Correlation between Cultrual and Geographic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.4495398  0.9967200  0.0032900  0.0032900 -0.5754918 -0.3109193
## [1] "Correlation between Linguistic and Historical"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.3372882  0.9859600  0.0140500  0.0167300 -0.5019408 -0.1639425
## [1] "Correlation between Linguistic and Geographic"
```

```
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.2594386  0.9182700  0.0817400  0.1195200 -0.3694719 -0.1840035
## [1] "Correlation between Historical and Geographic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
##  0.4210629  0.0004100  0.9996000  0.0004100  0.3313578  0.5176683
```

Run a mantel test comparing the semantic alignment to the cultural similarity, controlling for the historical distance between languages:

```
ecodist::mantel(as.dist(ling.m2)~
                  as.dist(cult.m2) +
                  as.dist(hist.m2),
                nperm = 100000)
```

```
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
##  0.4659407  0.0100000  0.9900100  0.0107800  0.3408500  0.5938397
```

Main Test: Run a mantel test comparing the semantic alignment to the cultural similarity, controlling for the historical distance and geographic distance between languages:

```
mainMantel = ecodist::mantel(as.dist(ling.m2)~
                  as.dist(cult.m2) +
                  as.dist(hist.m2) +
                  as.dist(geo.m2),
                nperm = 100000)
mainMantel
```

```
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
##  0.4508309  0.0114200  0.9885900  0.0119500  0.2962271  0.5993660
```

```
mainMantel = round(mainMantel,2)
mainMantel2 = sprintf("$r$ = %s, 95\\%% CI = [%s,%s], one-tailed $p$ = %s",
        round(mainMantel[1],3),
        round(mainMantel[5],3),
        round(mainMantel[6],3),
        round(mainMantel[2],3)
      )
mainMantel2 = gsub("0\\.",".",mainMantel2)
cat(mainMantel2,
    file="../results/stats/tex/MantelCultrualVsLinguisticDistance_Partial.tex")
```

MRM

Perform the main test using the phylogenetic distance, but using multiple regression on distance matrices (MRM).

```
set.seed(21889)
mainMRM = ecodist::MRM(as.dist(ling.m2)~
                  as.dist(cult.m2) +
                  as.dist(hist.m2) +
                  as.dist(geo.m2), nperm=10000)
mainMRM
```

```
## $coef
##                   as.dist(ling.m2)   pval
## Int                     0.123122021 0.9147
## as.dist(cult.m2)        0.350192033 0.0108
```

```
## as.dist(hist.m2)     -0.059044640 0.1623
## as.dist(geo.m2)       0.008177519 0.8212
##
## $r.squared
##        R2      pval
## 0.3072419 0.0073000
##
## $F.test
##        F   F.pval
## 24.68846  0.00730
```

```
mainMRM2 = sprintf("$\\beta= $%s, $p=$%s",
                   round(mainMRM$coef[2,1],2),
                   round(mainMRM$coef[2,2],2))
cat(mainMRM2,
    file="../results/stats/tex/MRMCultrualVsLinguisticDistance_Partial.tex")
```

Analysis of filtered data

The analyses in this section use local alignment values based on (a) data that passes the wikipedia filter, and (b) data that passes the semantic filter.

Wikipedia filter

```
ling.filtered = read.csv(
  "../data/FAIR/nel-wiki-k100-alignments-by-language-pair_Filtered.csv",
  stringsAsFactors = F)
```

Note that the semantic alignment for the filtered and unfiltered data are essentially exactly the same, but for fewer languages:

```
ling.filtered$unfiltered.rho =
  apply(ling.filtered[,
      c("iso2_l1","iso2_l2")],1,
  function(X){
    ling[(ling$l1==X[1] & ling$l2==X[2]) |
           (ling$l1==X[2] & ling$l2==X[1]),]$local_alignment[1]
  })
cor(ling.filtered$unfiltered.rho,ling.filtered$rho,use = "complete.obs")
```

```
## [1] 0.9999918
```

Continue to build data for replication:

```
ling.filtered$area1 = l[match(ling.filtered$name_l1,l$Language),]$autotyp.area
ling.filtered$area2 = l[match(ling.filtered$name_l2,l$Language),]$autotyp.area

fgroup = cbind(ling.filtered$family1,ling.filtered$family2)
fgroup = apply(fgroup,1,sort)
ling.filtered$family.group = apply(fgroup,2,paste,collapse=":")
agroup = cbind(ling.filtered$area1,ling.filtered$area2)
agroup = apply(agroup,1,sort)
ling.filtered$area.group = apply(agroup,2,paste,collapse=":")

ling.filtered$rho.center = scale(ling.filtered$rho)
ling.filtered$comparison_count.center = scale(ling.filtered$comparison_count)

matches = sapply(1:nrow(ling.filtered), function(i){
  x = which((cult$l1==ling.filtered$name_l1[i] &
          cult$l2==ling.filtered$name_l2[i]) |
            (cult$l2==ling.filtered$name_l1[i] &
          cult$l1==ling.filtered$name_l2[i]))
  x[1]
})

ling.filtered$cult.dist = cult[matches,]$cult.dist
# flip
ling.filtered$cult.dist = 1 - ling.filtered$cult.dist
ling.filtered = ling.filtered[!is.na(ling.filtered$cult.dist),]

ling.filtered$cult.dist.center = scale(ling.filtered$cult.dist)
```

```
m0F = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling.filtered
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m0.5F = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling.filtered
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m1F = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    cult.dist.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling.filtered
)
an1F = anova(m0F,m0.5F,m1F)
```

```
## refitting model(s) with ML (instead of REML)
```

```
an1F
```

```
## Data: ling.filtered
## Models:
## m0F: rho.center ~ 1 + (1 + cult.dist.center | family.group) + (1 +
## m0F:     cult.dist.center | area.group)
## m0.5F: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## m0.5F:     family.group) + (1 + cult.dist.center | area.group)
## m1F: rho.center ~ 1 + comparison_count.center + cult.dist.center +
## m1F:     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
## m1F:     area.group)
##       Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## m0F    8 446.96 473.74 -215.48   430.96
## m0.5F  9 428.63 458.76 -205.32   410.63 20.3271      1  6.527e-06 ***
## m1F   10 425.68 459.15 -202.84   405.68  4.9519      1    0.02606 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cultural similarity is significantly correlated with semantic alignment, even in the filtered data. Here are the model estimates:

```
summary(m1F)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ 1 + comparison_count.center + cult.dist.center +
##     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
##     area.group)
```

```
##    Data: ling.filtered
##
## REML criterion at convergence: 412.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8211 -0.5105 -0.0662  0.4157  6.2698
##
## Random effects:
##  Groups       Name             Variance Std.Dev. Corr
##  family.group (Intercept)      0.17449  0.4177
##               cult.dist.center 0.07149  0.2674   -0.21
##  area.group   (Intercept)      0.64616  0.8038
##               cult.dist.center 0.02791  0.1670   1.00
##  Residual                      0.26481  0.5146
## Number of obs: 210, groups:  family.group, 31; area.group, 19
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)              -0.82996    0.21638  -3.836
## comparison_count.center   0.41070    0.08668   4.738
## cult.dist.center          0.29681    0.11189   2.653
##
## Correlation of Fixed Effects:
##             (Intr) cmpr_.
## cmprsn_cnt.  0.133
## clt.dst.cnt  0.387 -0.054
```

Semantic filter

```
ling.semFiltered = read.csv(
  "../data/FAIR/nel-wiki-k100-alignments-by-language-pair_SemanticFiltered.csv",
  stringsAsFactors = F)

ling.semFiltered$area1 = l[match(ling.semFiltered$name_l1,l$Language),]$autotyp.area
ling.semFiltered$area2 = l[match(ling.semFiltered$name_l2,l$Language),]$autotyp.area

fgroup = cbind(ling.semFiltered$family1,ling.semFiltered$family2)
fgroup = apply(fgroup,1,sort)
ling.semFiltered$family.group = apply(fgroup,2,paste,collapse=":")
agroup = cbind(ling.semFiltered$area1,ling.semFiltered$area2)
agroup = apply(agroup,1,sort)
ling.semFiltered$area.group = apply(agroup,2,paste,collapse=":")

ling.semFiltered$rho.center = scale(ling.semFiltered$rho)
ling.semFiltered$comparison_count.center = scale(ling.semFiltered$comparison_count)

matches = sapply(1:nrow(ling.semFiltered), function(i){
  x = which((cult$l1==ling.semFiltered$name_l1[i] &
          cult$l2==ling.semFiltered$name_l2[i]) |
            (cult$l2==ling.semFiltered$name_l1[i] &
          cult$l1==ling.semFiltered$name_l2[i]))
  x[1]
})

ling.semFiltered$cult.dist = cult[matches,]$cult.dist
# flip
ling.semFiltered$cult.dist = 1 - ling.semFiltered$cult.dist
ling.semFiltered = ling.semFiltered[!is.na(ling.semFiltered$cult.dist),]

ling.semFiltered$cult.dist.center = scale(ling.semFiltered$cult.dist)
```

```
m0F = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling.semFiltered
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m0.5F = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling.semFiltered
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m1F = lmer(
  rho.center ~ 1 +
```

```
    comparison_count.center +
    cult.dist.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling.semFiltered
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
an1F = anova(m0F,m0.5F,m1F)
```

```
## refitting model(s) with ML (instead of REML)
```

```
an1F
```

```
## Data: ling.semFiltered
## Models:
## m0F: rho.center ~ 1 + (1 + cult.dist.center | family.group) + (1 +
## m0F:     cult.dist.center | area.group)
## m0.5F: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## m0.5F:     family.group) + (1 + cult.dist.center | area.group)
## m1F: rho.center ~ 1 + comparison_count.center + cult.dist.center +
## m1F:     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
## m1F:     area.group)
##       Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## m0F    8 1660.3 1697.1 -822.17   1644.3
## m0.5F  9 1300.8 1342.1 -641.39   1282.8 361.576      1  < 2.2e-16 ***
## m1F   10 1286.2 1332.1 -633.09   1266.2  16.595      1  4.627e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cultural similarity is significantly correlated with semantic alignment, even in the semantic filtered data. Here are the model estimates:

```
summary(m1F)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ 1 + comparison_count.center + cult.dist.center +
##     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
##     area.group)
##    Data: ling.semFiltered
##
## REML criterion at convergence: 1279.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6257 -0.6155  0.1002  0.6528  4.7581
##
## Random effects:
##  Groups       Name             Variance  Std.Dev. Corr
##  family.group (Intercept)      0.1627947 0.40348
##               cult.dist.center 0.0001944 0.01394  1.00
##  area.group   (Intercept)      0.0515199 0.22698
##               cult.dist.center 0.0037519 0.06125  -1.00
##  Residual                      0.2916401 0.54004
## Number of obs: 731, groups:  family.group, 48; area.group, 20
##
```

```
## Fixed effects:
##                           Estimate Std. Error t value
## (Intercept)              -0.39039    0.09111  -4.285
## comparison_count.center   0.62768    0.02766  22.696
## cult.dist.center          0.19836    0.03301   6.009
##
## Correlation of Fixed Effects:
##             (Intr) cmpr_.
## cmprsn_cnt.  0.083
## clt.dst.cnt -0.192 -0.200
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Both filters

Main test on data where both the wikipedia and semantic filter are on.

```
ling.bothFiltered = read.csv(
  "../data/FAIR/nel-wiki-k100-alignments-by-language-pair_BothFiltered.csv",
  stringsAsFactors = F)

ling.bothFiltered$area1 = l[match(ling.bothFiltered$name_l1,l$Language),]$autotyp.area
ling.bothFiltered$area2 = l[match(ling.bothFiltered$name_l2,l$Language),]$autotyp.area

fgroup = cbind(ling.bothFiltered$family1,ling.bothFiltered$family2)
fgroup = apply(fgroup,1,sort)
ling.bothFiltered$family.group = apply(fgroup,2,paste,collapse=":")
agroup = cbind(ling.bothFiltered$area1,ling.bothFiltered$area2)
agroup = apply(agroup,1,sort)
ling.bothFiltered$area.group = apply(agroup,2,paste,collapse=":")

ling.bothFiltered$rho.center = scale(ling.bothFiltered$rho)
ling.bothFiltered$comparison_count.center = scale(ling.bothFiltered$comparison_count)

matches = sapply(1:nrow(ling.bothFiltered), function(i){
  x = which((cult$l1==ling.bothFiltered$name_l1[i] &
          cult$l2==ling.bothFiltered$name_l2[i]) |
            (cult$l2==ling.bothFiltered$name_l1[i] &
          cult$l1==ling.bothFiltered$name_l2[i]))
  x[1]
})

ling.bothFiltered$cult.dist = cult[matches,]$cult.dist
# flip
ling.bothFiltered$cult.dist = 1 - ling.bothFiltered$cult.dist
ling.bothFiltered = ling.bothFiltered[!is.na(ling.bothFiltered$cult.dist),]

ling.bothFiltered$cult.dist.center = scale(ling.bothFiltered$cult.dist)
```

```
m0F = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling.bothFiltered
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m0.5F = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling.bothFiltered
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m1F = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    cult.dist.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling.bothFiltered
)
an1F = anova(m0F,m0.5F,m1F)
```

```
## refitting model(s) with ML (instead of REML)
```

```
an1F
```

```
## Data: ling.bothFiltered
## Models:
## m0F: rho.center ~ 1 + (1 + cult.dist.center | family.group) + (1 +
## m0F:     cult.dist.center | area.group)
## m0.5F: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## m0.5F:     family.group) + (1 + cult.dist.center | area.group)
## m1F: rho.center ~ 1 + comparison_count.center + cult.dist.center +
## m1F:     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
## m1F:     area.group)
##       Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## m0F    8 447.24 474.02 -215.62   431.24
## m0.5F  9 429.45 459.58 -205.73   411.45 19.7903      1  8.642e-06 ***
## m1F   10 426.47 459.95 -203.24   406.47  4.9762      1     0.0257 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cultural similarity is significantly correlated with semantic alignment, even in the fully filtered data. Here are the model estimates:

```
summary(m1F)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ 1 + comparison_count.center + cult.dist.center +
##     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
##     area.group)
##    Data: ling.bothFiltered
##
## REML criterion at convergence: 413.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8275 -0.5144 -0.0659  0.4117  6.2874
##
## Random effects:
##  Groups       Name             Variance Std.Dev. Corr
##  family.group (Intercept)      0.17288  0.4158
##               cult.dist.center 0.07138  0.2672   -0.21
##  area.group   (Intercept)      0.64816  0.8051
##               cult.dist.center 0.02802  0.1674   1.00
##  Residual                      0.26612  0.5159
## Number of obs: 210, groups:  family.group, 31; area.group, 19
##
```

```
## Fixed effects:
##                           Estimate Std. Error t value
## (Intercept)               -0.8297     0.2166  -3.831
## comparison_count.center    0.4075     0.0872   4.673
## cult.dist.center           0.2976     0.1119   2.659
##
## Correlation of Fixed Effects:
##             (Intr) cmpr_.
## cmprsn_cnt.  0.135
## clt.dst.cnt  0.387 -0.053
```

Comparison between domains

The code that produce the results of this section can be found in `analysis/compareDomains.R`.

Part 1: Compare each linguistic domain to the overall cultural similarity

We fit a mixed effects model to compare the semantic alignment in a given domain to the overall cultural distance. The semantic alignment for the given domain is the dependent variable. There are random intercepts for language family and area pairs, and random slopes for overall cultural similarity by language family and by area. The `comparison_count` variable is added as a fixed effect. This null model is compared to a model with an additional fixed effect for the overall cultural similairty.

There are 21 linguistic domains with enough data. All correlations are positive and 11 are significant at the 0.05 level (adjusted for multiple comparisons).

The full results are in the file:

`../results/stats/wikipedia-main/Cor_LingAlignmentByDomains_vs_OverallCulturalSimilarity.csv`

Summary:

```
p1res
```

```
##                            Domain       Beta            p   Adjusted p sig
## 2                   Food and drink 0.29039152 3.842274e-08 8.068775e-07   *
## 6    Miscellaneous function words 0.31349670 9.672370e-08 2.031198e-06   *
## 9                        The body 0.23183657 8.711593e-07 1.829434e-05   *
## 13                        Animals 0.26483784 4.281952e-06 8.992099e-05   *
## 21                           Time 0.26708073 3.341177e-05 7.016471e-04   *
## 3      Agriculture and vegetation 0.21319270 4.954909e-05 1.040531e-03   *
## 16                   Modern world 0.15392213 2.860946e-04 6.007988e-03   *
## 14             The physical world 0.15530592 6.771587e-04 1.422033e-02   *
## 11              Spatial relations 0.11188738 1.323355e-03 2.779045e-02   *
## 20                         Kinship 0.25408132 1.332699e-03 2.798669e-02   *
## 7           Clothing and grooming 0.16478921 2.245060e-03 4.714625e-02   *
## 10               Sense perception 0.11168260 2.806430e-03 5.893504e-02
## 15 Social and political relations 0.10210872 6.603817e-03 1.386802e-01
## 1                        The house 0.10743767 1.485121e-02 3.118755e-01
## 18                        Quantity 0.13804241 1.691864e-02 3.552914e-01
## 8             Speech and language 0.11367507 3.209804e-02 6.740588e-01
## 19   Basic actions and technology 0.06996793 7.317704e-02 1.000000e+00
## 17                       Cognition 0.06413968 9.294337e-02 1.000000e+00
## 12            Emotions and values 0.06324463 9.437249e-02 1.000000e+00
## 5                       Possession 0.07833831 1.102507e-01 1.000000e+00
## 4                           Motion 0.05544251 2.537090e-01 1.000000e+00
```

Part 2: Compare each linguistic domain to the cultural similarity of each original D-PLACE domain

The method is the same as for part 1, except the cultural distance for a particular cultural domain is used instead of the overall cultural distance.

The full results are in the file:

`../results/stats/wikipedia-main/Cor_LingAlignmentByDomains_vs_DPlaceCulturalDomains.csv`

The graph below shows the mixed effects model coefficient estimate for the relationship between each linguistic domain and each cultural domain. Pink colours indicate positive correlations and blue colours indicate negative correlations. Stronger colours indicate stronger correlations. An asterisk indicates that the correlation is stronger than would be expected by chance, when adjusting the p-value for multiple comparisons.

The insert in the top left shows the distribution of Beta values.

The domains are clustered using higherarchical clustering. This is for visualisaiton and reflects similarity in the numeric relations, not history or conceptual hierarchies.

List of significant correlations (after adjusting p-value for multiple comparisons):

```
##                         Ling Domain Cult Domain      Beta   Adjusted p
## 68                         The body    Politics 0.2139638 1.292033e-03
## 155                         Kinship  Settlement 0.2785591 8.766279e-03
## 43    Miscellaneous function words  Settlement 0.2899807 7.022120e-07
## 83               Spatial relations  Settlement 0.1122428 1.431007e-03
## 59             Speech and language  Settlement 0.1915031 3.729177e-05
## 107             The physical world  Settlement 0.1321783 4.102261e-02
## 17      Agriculture and vegetation Subsistence 0.2513632 2.887202e-05
## 97                         Animals Subsistence 0.2942266 7.865336e-06
## 49           Clothing and grooming Subsistence 0.2484714 1.338589e-04
## 89             Emotions and values Subsistence 0.1524638 7.946118e-04
## 9                   Food and drink Subsistence 0.3005301 1.084517e-06
## 153                        Kinship Subsistence 0.2346825 4.160376e-03
## 41    Miscellaneous function words Subsistence 0.3353616 2.648483e-05
## 121                   Modern world Subsistence 0.1851981 7.434982e-06
## 137                       Quantity Subsistence 0.2454534 7.092645e-03
## 73                Sense perception Subsistence 0.1850017 6.280926e-04
## 113 Social and political relations Subsistence 0.1544007 3.637901e-05
## 81               Spatial relations Subsistence 0.1504141 6.897505e-04
## 57             Speech and language Subsistence 0.1991926 2.699664e-04
## 65                         The body Subsistence 0.2764446 5.291731e-08
## 105             The physical world Subsistence 0.2221189 1.157346e-04
## 161                           Time Subsistence 0.2921557 1.640226e-05
```

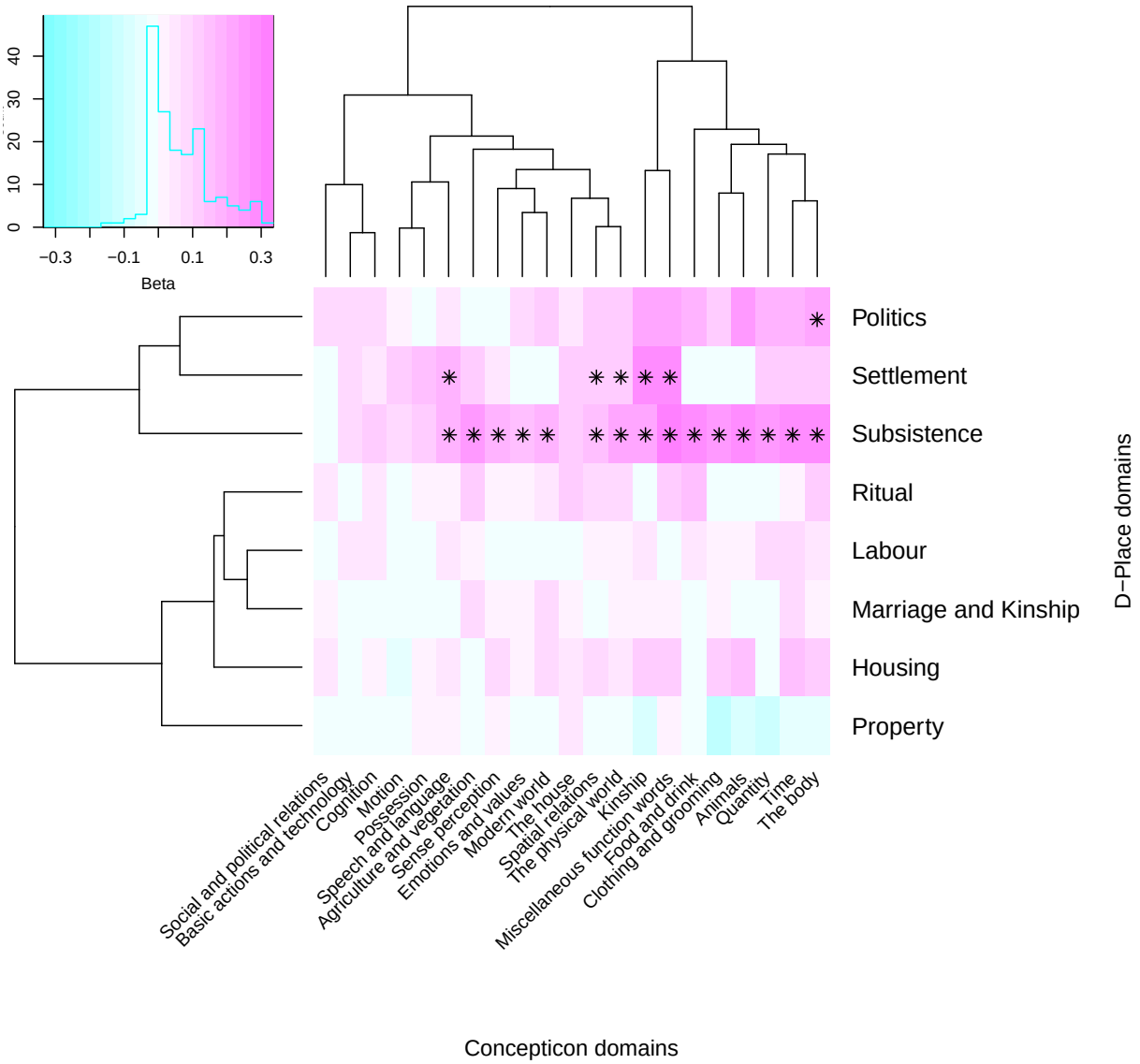

0
10
20
30
40
−0.3
−0.1
0.1
0.3
Beta
Politics
Settlement
Subsistence
Ritual
Labour
Marriage and Kinship
Housing
Property
D−Place domains
Social and political relations
Basic actions and technology
Cognition
Motion
Possession
Speech and language
Agriculture and vegetation
Sense perception
Emotions and values
Modern world
The house
Spatial relations
The physical world
Kinship
Miscellaneous function words
Food and drink
Clothing and grooming
Animals
Quantity
Time
The body
Concepticon domains

Figure 1:

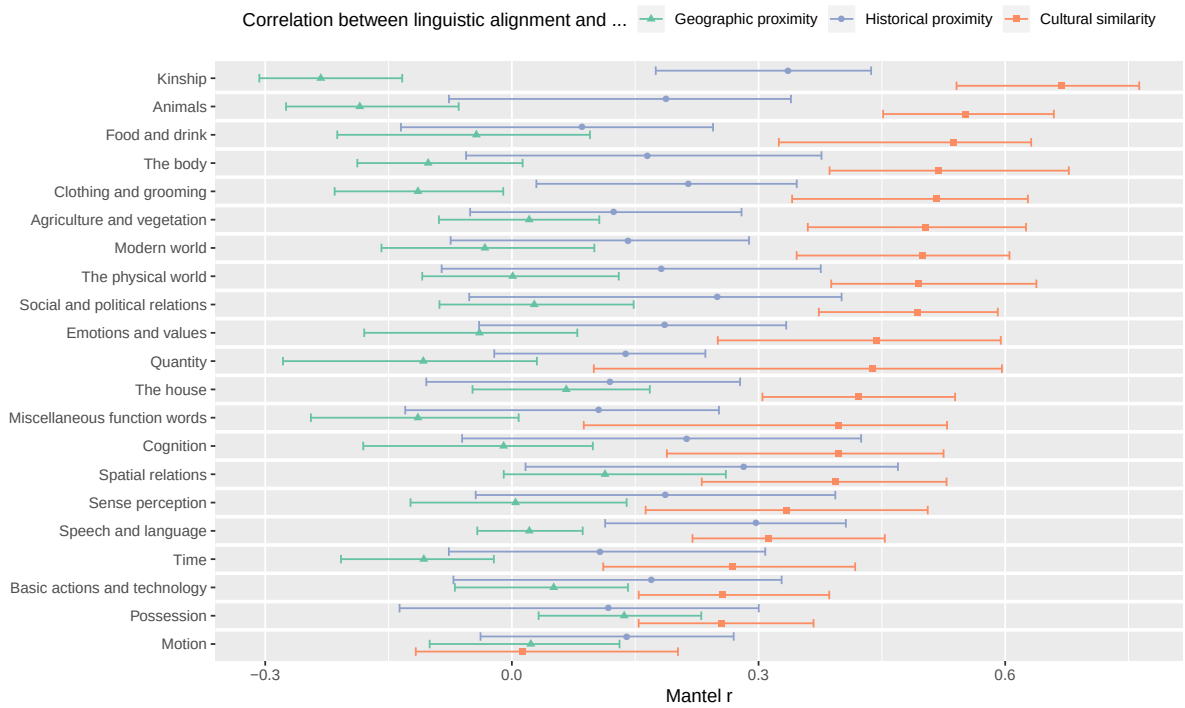


Figure 2:

Part 3: Compare each linguistic domain to the phylogenetic and geographic distance

This test compares each semantic alignment score to each of three target distances: the cultural distance, the historical distance and the geographic distance. We use a partial Mantel test (from the package `ecodist`) to estimate the strength of the relationship between the linguistic domain and the target distance, while controlling for the other two distances. The test uses 100,000 permutations.

The full results are in the file:

`Cor_LingAlignmentByDomains_vs_HistoricalAndGeographicalDistance.csv`

The graph below shows the results. Point estimates are the estimated Mantel R. The error bars show the 95% confidence intervals from the permutation test.

There appears to be a trade-off: The stronger the relationship with geographic distance, the weaker the relationship with cultural distance (r = -0.529, t = -2.72, df=19, p = 0.014). This does not hold for historical and cultural distance (r = 0.27, t = 1.22, df=19, p = 0.24).

Note that, after controlling for multiple comparisons, only 2 domains are significant:

```
##     domain comparison   mantelr     lower     upper   pval3 p.adjusted
## 37 Animals  lingVCult 0.5518312 0.4515039 0.6591382 0.00129    0.02709
## 58 Kinship  lingVCult 0.6687835 0.5407906 0.7629987 0.00012    0.00252
```

References

Bouckaert, Remco, Philippe Lemey, Michael Dunn, Simon J. Greenhill, Alexander V. Alekseyenko, Alexei J. Drummond, Russell D. Gray, Marc A. Suchard, and Quentin D. Atkinson (2012). Mapping the origins and expansion of the Indo-European language family. Science, 337(6097), 957-960.

van Buuren, S. & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1-67. URL https://www.jstatsoft.org/v45/i03/.

Castellano, S., & Balletto, E. (2002). Is the partial Mantel test inadequate?. Evolution, 56(9), 1871-1873.

Goslee, S.C. 2010. Correlation analysis of dissimilarity matrices. Plant Ecology 206(2):279-286.

Goslee, S.C. & Urban, D.L. (2007). The ecodist package for dissimilarity-based analysis of ecological data. Journal of Statistical Software 22(7):1-19.

Hammarstrom, H. , R. Forkel, M. Haspelmath (2018) clld/glottolog: Glottolog database 3.3, Jena: Max Planck Institute for the Science of Human History.

Harmon, L. J., & Glor, R. E. (2010). Poor statistical performance of the Mantel test in phylogenetic comparative analyses. Evolution: International Journal of Organic Evolution, 64(7), 2173-2178.

Jäger, G. (2018). Global-scale phylogenetic linguistic inference from lexical resources. Scientific data, 5, 180189.

Kirby, Kathryn R., Russell D. Gray, Simon J. Greenhill, Fiona M. Jordan, Stephanie Gomes-Ng, Hans-Jörg Bibiko, Damián E. Blasi, Carlos A. Botero, Claire Bowern, Carol R. Ember, Dan Leehr, Bobbi S. Low, Joe McCarter, William Divale, and Michael C. Gavin. (2016). D-PLACE: A Global Database of Cultural, Linguistic and Environmental Diversity. PLoS ONE, 11(7): e0158391.

Legendre, P. (2000). Comparison of permutation methods for the partial correlation and partial Mantel tests. Journal of Statistical Computation and Simulation, 67(1), 37-73.

Lichstein, J. W. (2007). Multiple regression on distance matrices: a multivariate spatial analysis tool. Plant Ecology, 188(2), 117-131.

Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. Cancer research, 27(2 Part 1), 209-220.

Murdock, G. P., R. Textor, H. Barry, III, D. R. White, J. P. Gray, and W. T. Divale. 1999. Ethnographic Atlas. World Cultures 10:24-136 (codebook)

Nichols, J., Witzlack-Makarevich, A. & Bickel, B. (2013), The AUTOTYP genealogy and geography database: 2013 release, http://www.spw.uzh.ch/autotyp/.

Smouse, P.E., Long, J.C. & Sokal, R.R. (1986). Multiple regression and correlation extensions of the Mantel test of matrix correspondence. Systematic Zoology 35:62 7-632.

Wichmann, Søren, Eric W. Holman, and Cecil H. Brown (eds.). 2018. The ASJP Database (version 17).

4.2 Analysis of numerals

Semantic alignments in number words

Bill Thompson, Seán Roberts & Gary Lupyan

Contents

Introduction

What predicts the semantic alignment of number words?

Although translations may be more direct for number words, there are still differences between languages regarding their semantic associations. Numerals in 16 languages were analysed using binary decision trees to find coherent clusters then a Generalised Additive Model to explain differences in alignment.

Variables:

- *l1*: Iso2 code for language 1
- *l2*: Iso2 code for language 2
- *wordform_l1*: Orthogaphic word form for language 1
- *wordform_l2*: Orthogaphic word form for language 2
- *local_alignment*: Local semantic alignment
- *global_alignment*: Global semantic alignment
- *freq_l1*: Frequency of orthographic form in l1
- *freq_l2*: Frequency of orthographic form in l1
- *neighbour_overlap*: Number of neighbours in common
- *global_density_l1*: Density
- *global_density_l2*: Density
- *local_density_l1*: Density
- *local_density_l2*: Density

- *editdistance*: Edit distance between orthographic forms
- *k*: Parameter for number of neighbours (constant at k=100).
- *n*: Number of possible comparisons
- *number*: Number concept
- *number_numeric*: Numeric value of number
- *name_l1*: Name of langauge l1
- *family_l1*: Family of language l1
- *name_l2*: Name of langauge l2
- *family_l2*: Family of language l2
- *same.family*: Compared langauges are part of same family?
- *hist.dist*: Historical distance according to phylogenetic tree.
- *l1_typology*: Numeral typology for L1 according to Calude & Verkerk (2016) (see the section on "Number line" below).
- *l2_typology*: Same as above for L2.
- *sameNumeralTypology*: Comparison of numeral typology. This is 1 if the typology is the same in L1 and L2 (i.e. both numbers are atoms or use the same composition), and zero if the typology is different.
- *seven*: True if the numeral is 7. (see below)
- *homophone*: True if the form of the word has an alternative referential class in the North Euralex database. This includes mainly homophones (unrelated meanings) but also some synonyms (words with different but related meanings). For legacy reasons, the variable is named 'homophone'.
- *freqDiff*: Difference in orthographic frequency (absolute, log scale)

The Estimated Degrees of Freedom (EDF) is an indication of how non-linear a smooth term is (higher = less linear, see e.g. Wood, 2008). In general, a curve with an EDF of around 2 will look like a quadratic curve, and an EDF of around 3 will look like a cubic curve. However, this does not have to be the case: a smooth term could have a strong linear term, and a very weak non-linear term. The EDF captures this possibility as a continuous value. The simplest way to actually assess the smooth term is to plot it.

Random effects in the GAM implementation we use are treated just like a smooth term with the identity matrix as the penalty coefficient matrix. When entering a language pair as a random (intercept) effect, coefficients are created for each pair, modelled as independent and identically distributed normal random variables. The values are defined as discrete points along a smooth function. So, just like in a mixed effects model, the predicted alignment can be adjusted by a random intercept (the coefficients), e.g. the model can represent the alignment between English and French as higher overall, and the alignment between English and Bulgarian as being slightly lower etc. Stronger differences between levels of the random effect would need be represented by more complex functions, which would be penalised (similar to how a linear mixed effect model penalises random effect coefficient estimates which deviate from a normal distribution). The EDF value for the random effects relates to the 'wiggliness' of these coefficients when plotted in a regular space. This makes the EDF difficult to interpret. A random effect where there were no differences between levels would have an EDF of 1 (a flat line), but it would also be 1 when there were consistent distances between each level. So a high EDF would indicate something like an imbalance in the distribution of coefficients, e.g. a range of values that does not fit a normal distribution. In fact, there are several language pairs with lower alignment (pairs from different language families), and few with very high alignment (possibly a ceiling effect).

See the SI of Monaghan & Roberts (2019) for further explanation of EDF and random effects applied to lingusitic data.

Monaghan & Roberts (2019) Cognitive influences in language evolution: Psycholinguistic predictors of loan word borrowing. *Cognition*, 186, 147-158.

Wood, S. N. (2008). Fast stable direct fitting and smoothness selection for generalized additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3), 495-518.

Load libraries, graphing theme

```
library(mgcv) # for gam
library(lmtest) # for model comparison
source("GAM_derivaties.R") # for derivitives plot
library(lme4)
library(tidyverse)
#library(langcog)
library(boot)
library(ggplot2)
library(lazyeval)
library(data.table)
library(MuMIn)
library(REEMtree)
library(rpart)
library(rpart.plot)
library(gridExtra)
library(grid)
library(gridBase)


myThemeBasic =
  theme_bw()+
  theme(panel.grid.minor=element_blank(),
        panel.grid.major=element_blank(),
        panel.background=element_blank())+
  theme(axis.text.x=element_text(size=13),
        axis.text.y=element_text(size=13),
        axis.title.x=element_text(size=12),
        axis.title.y=element_text(size=12))+
  theme(legend.background = element_rect(fill="transparent"))+
  theme(legend.text = element_text(size=13))+
  theme(legend.title = element_text(size=13))+
  theme(axis.title.y=element_text(vjust=0.9,size=20))+
  theme(axis.title.x=element_text(vjust=0.9,size=20))+
  theme(axis.title.y=element_text(vjust=0))+
  theme(axis.title.x=element_text(vjust=0))
```

Load data

```
PATH <- "../../data/numbers"
numbers <- read.csv(file.path(PATH,"number-alignments.csv"),
                    encoding = "UTF-8",fileEncoding = "UTF-8") %>%
    left_join(read.csv(file.path(PATH,"word_to_number.csv")),by=c("Number"="number"))

data.table::setnames(numbers,tolower(names(numbers))) #lowercase column names for consistency

language_info <- read.csv("../../data/FAIR_langauges_glotto_xdid.csv",
                          encoding = "UTF-8",fileEncoding = "UTF-8")

numbers <- left_join(numbers,select(language_info,Language,family,iso2),by=c("l1"="iso2"))
```

```
numbers <- numbers %>% rename(name_l1=Language,family_l1=family)


numbers <- left_join(numbers,select(language_info,Language,family,iso2),by=c("l2"="iso2"))

numbers <- numbers %>% rename(name_l2=Language,family_l2=family)

irreg_in_danish = c(50,60,70,80,90)
numbers <- numbers %>% mutate(
  same.family = family_l1==family_l2,
    is_danish = l1=="da" | l2=="da",
    is_single = (number_numeric < 10),
    is_decade = (number_numeric < 100 & number_numeric %% 10 ==0),
    is_hundred = (number_numeric == 100),
    is_thousand = (number_numeric == 1000),
    irreg_in_danish = (number_numeric %in% irreg_in_danish),
    lang_pair = paste(pmax(l1,l2),pmin(l1,l2),sep="-"))

histDistance = read.csv("../../data/trees/IndoEuropean_historical_distances_long.csv",
                        stringsAsFactors = F,encoding = "UTF-8",fileEncoding = "UTF-8")
numbers$hist.dist = histDistance[match(paste(numbers$name_l1,numbers$name_l2),
                            paste(histDistance$Var1,histDistance$Var2)),]$value

# Refactorise to get rid of non-existant categories
numbers$family_l1 = factor(numbers$family_l1)
numbers$family_l2 = factor(numbers$family_l2)

# Add typology data
cnv = read.csv("../../data/numbers/Calude_Verkerk_NumberData.csv",stringsAsFactors = F)
numbers <- left_join(numbers,cnv,by=c('l1','l2','number_numeric'))

numbers$isUralic = numbers$family_l1 == "Uralic" | numbers$family_l2 == "Uralic"
numbers$danish_irregular = (numbers$number_numeric %in% irreg_in_danish) &
                                     (numbers$name_l1=="Danish" | numbers$name_l2=="Danish")

# historic distance, setting Uralic languages to maximum
numbers$hist.dist2 = numbers$hist.dist
numbers$hist.dist2[is.na(numbers$hist.dist2)] = max(numbers$hist.dist,na.rm = T)
numbers$seven = numbers$number_numeric==7

#some rows got duplicated; remove
numbers <- distinct(numbers)

# Data on homophones
h = read.csv("../../data/numbers/NumberHomophones.csv",
             stringsAsFactors = F,encoding = "UTF-8",fileEncoding = "UTF-8")
h$code = paste(h$l,h$word)
numbers$homophone = (paste(numbers$l1,numbers$wordform_l1) %in% h$code) |
                    (paste(numbers$l2,numbers$wordform_l2) %in% h$code)

numbers$lang_pair.f = factor(numbers$lang_pair)

# Frequency difference (already in log scale)
```

```
numbers$freqDiff = abs(numbers$freq_l1-numbers$freq_l2)
# 78 frequency observations (3%) are missing, so impute:
freqM = bam(I(1+freqDiff)~
              #s(number_numeric) +
              s(lang_pair.f,bs='re') +
              s(editdistance) +
              s(hist.dist2),
            family = Gamma(link="identity"),
            data = numbers[!is.na(numbers$freqDiff),])
freqMPred = predict(freqM,newdata=numbers)-1
#plot(freqMPred,numbers$freqDiff)
#abline(0,1)
numbers[is.na(numbers$freqDiff),]$freqDiff =
  freqMPred[is.na(numbers$freqDiff)]
```

Group data by language:

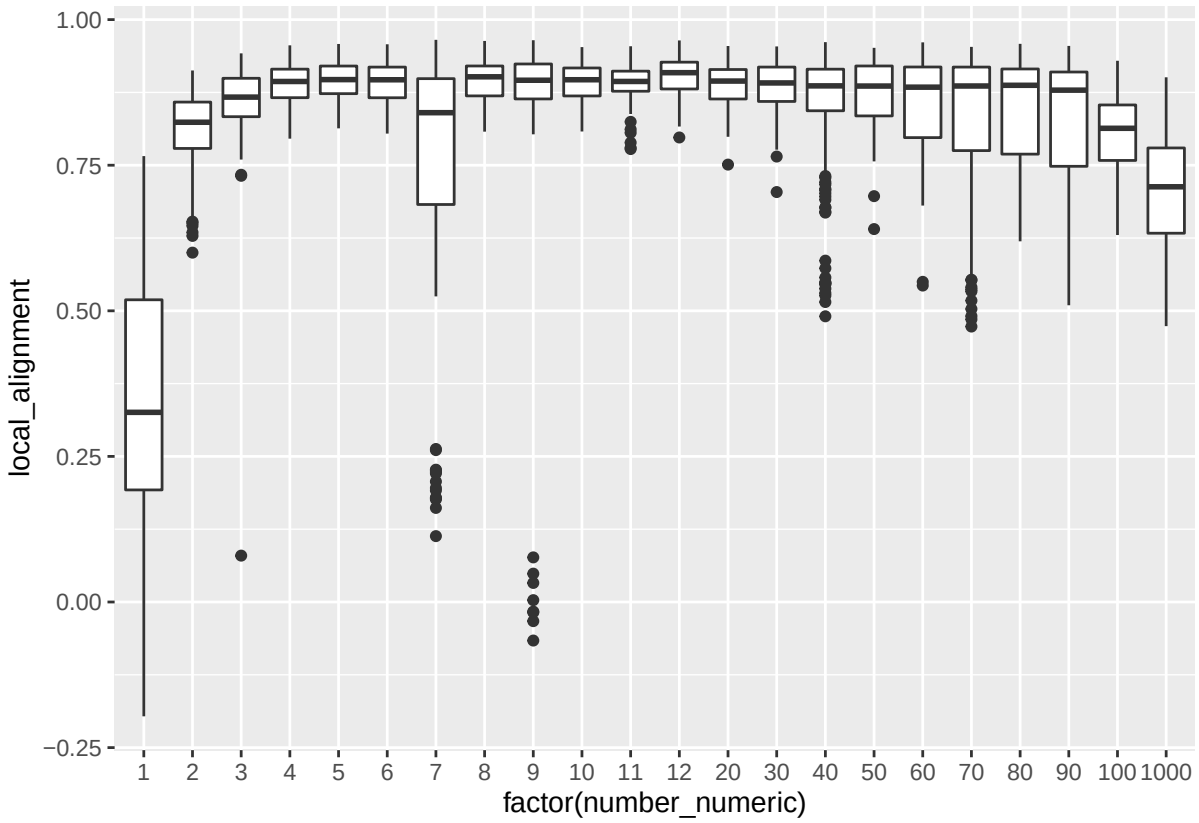
```
langAverages = data.frame()
for(l in unique(c(numbers$l1,numbers$l2))){
  dx = numbers[numbers$l1==l | numbers$l2==l,]
  langAverages = rbind(langAverages,
                  data.frame(
                   local_alignment = dx$local_alignment,
                    number_numeric = dx$number_numeric,
                    l = l,
                    l1 = dx$name_l1,
                    l2 = dx$name_l2))
}
langAverages$l = factor(langAverages$l,
     levels = names(sort(tapply(langAverages$local_alignment,langAverages$l,mean))))
```

Overview

Numeric value

Plot by numeric value:

```
ggplot(numbers,aes(x=factor(number_numeric),y=local_alignment)) + geom_boxplot()
```



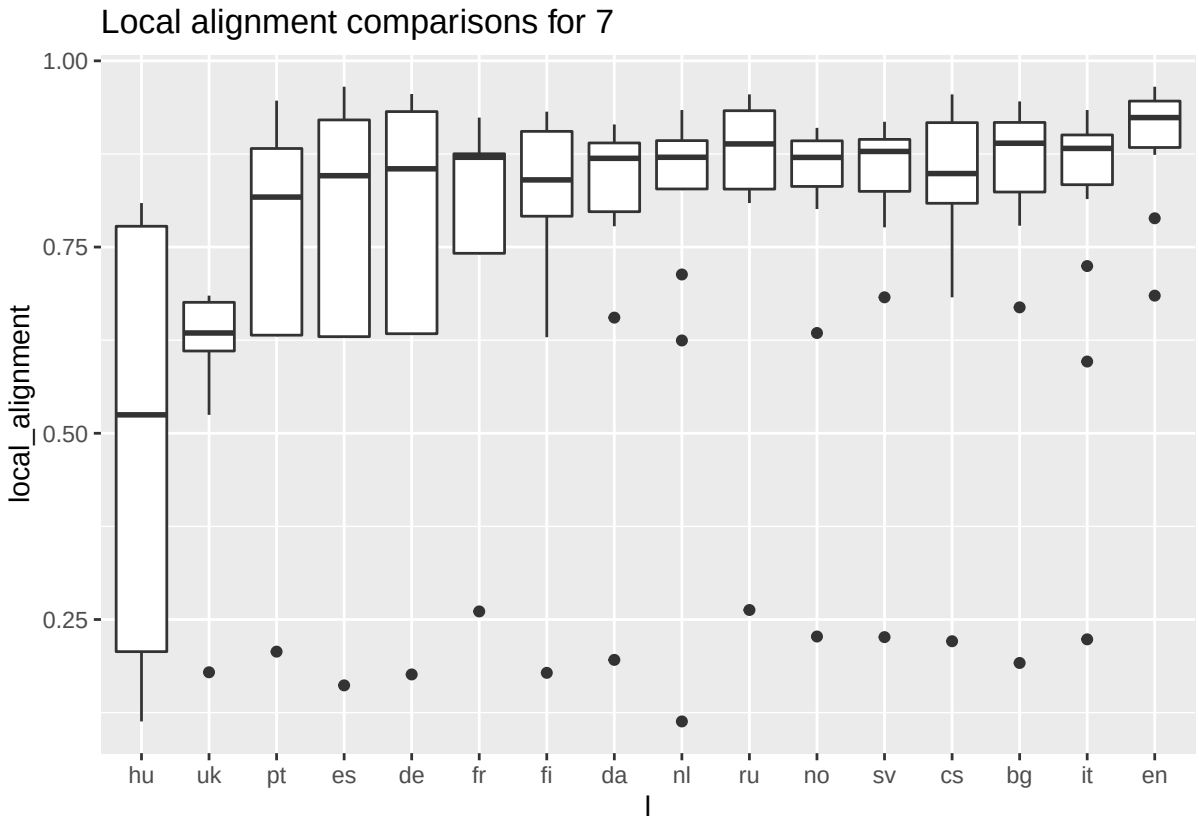
- 1 has a low numeric alignment.
- Alignment rises 2-3.
- 7 has a lower alignment.
- Drop in alignment for 100, 1000.
- Outliers for 7 and 9.
- Outliers for 40 and 70?

1 and 2 are often sources for grammaticalised indefinite/duel markers (Givón, 1981).

The slight decline from 10 to 1000 could be due to the declining frequency of occurances of these numbers (Dehaene & Mehler, 1992), which might affect convergence on meanings, but more direcly would affect the co-occurance statistics.

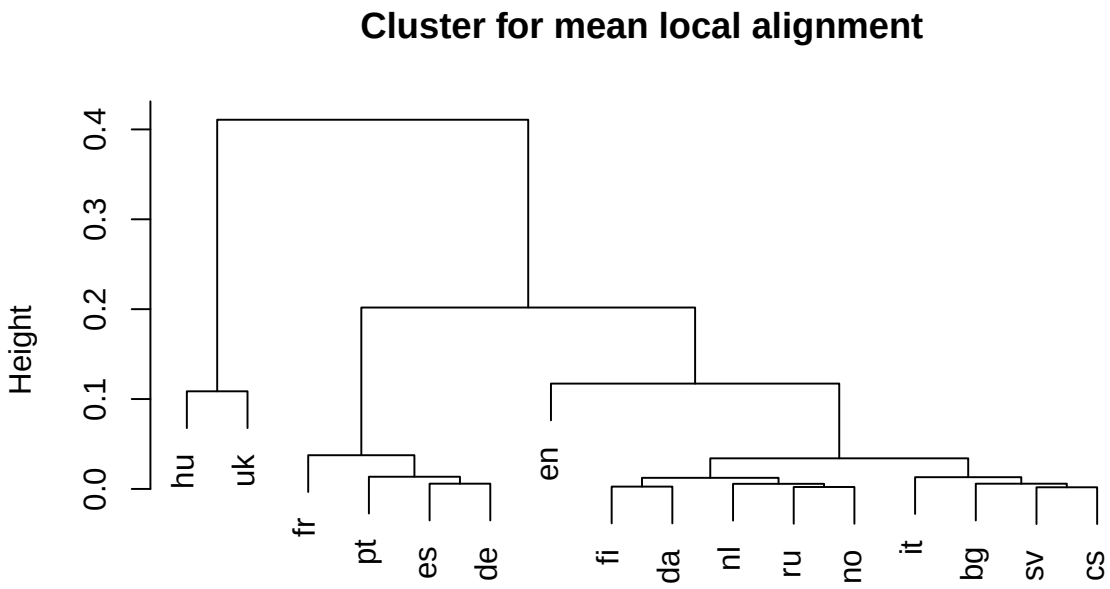
What's driving the difference with 7 and 9? 7 might be linked to there being 7 days in the week, so semantic differences in time might be reflected. Let's look at individual languages:

```
la7 = langAverages[langAverages$number_numeric==7,]
la7$l = factor(la7$l,levels=names(sort(tapply(
         la7$local_alignment,la7$l,mean))))
ggplot(la7,aes(x=l,y=local_alignment)) +
  geom_boxplot() +
  ggtitle("Local alignment comparisons for 7")
```



The plot above shows that the numeral 7 has lower alignment when it involves a comparison with Hungarian or Ukrainian. For example, all the outliers around 0.25 are comparisons with Hungarian. Also, the means are clearly different from the other languages, as shown by hierarchical clustering:

```
hc = hclust(dist(sort(tapply(la7$local_alignment,la7$l,mean))))
plot(hc,main="Cluster for mean local alignment",xlab="",sub = "")
```



This might be because Hungarian is a Uralic language, but maybe also because the Hungarian word for '7' also directly means "week". Ukrainian is also low. We note that forms for 7 and 8 are very similar in

Ukrainian (7 = sim and 8 = visim).

What are the outliers for 9? These all include comparisons to French:

```
numbers[numbers$number_numeric==9 & numbers$local_alignment<0.25,c("l1",'l2',"local_alignment")]
```

```
##      l1 l2 local_alignment
## 1    fr uk     0.003192686
## 483  fr da    -0.016457496
## 786  fr cs     0.076614888
## 903  fr bg    -0.032815537
## 1231 fr fi     0.032734709
## 1366 fr no    -0.066181331
## 1436 fr sv     0.048704798
## 2401 fr ru    -0.017702025
```

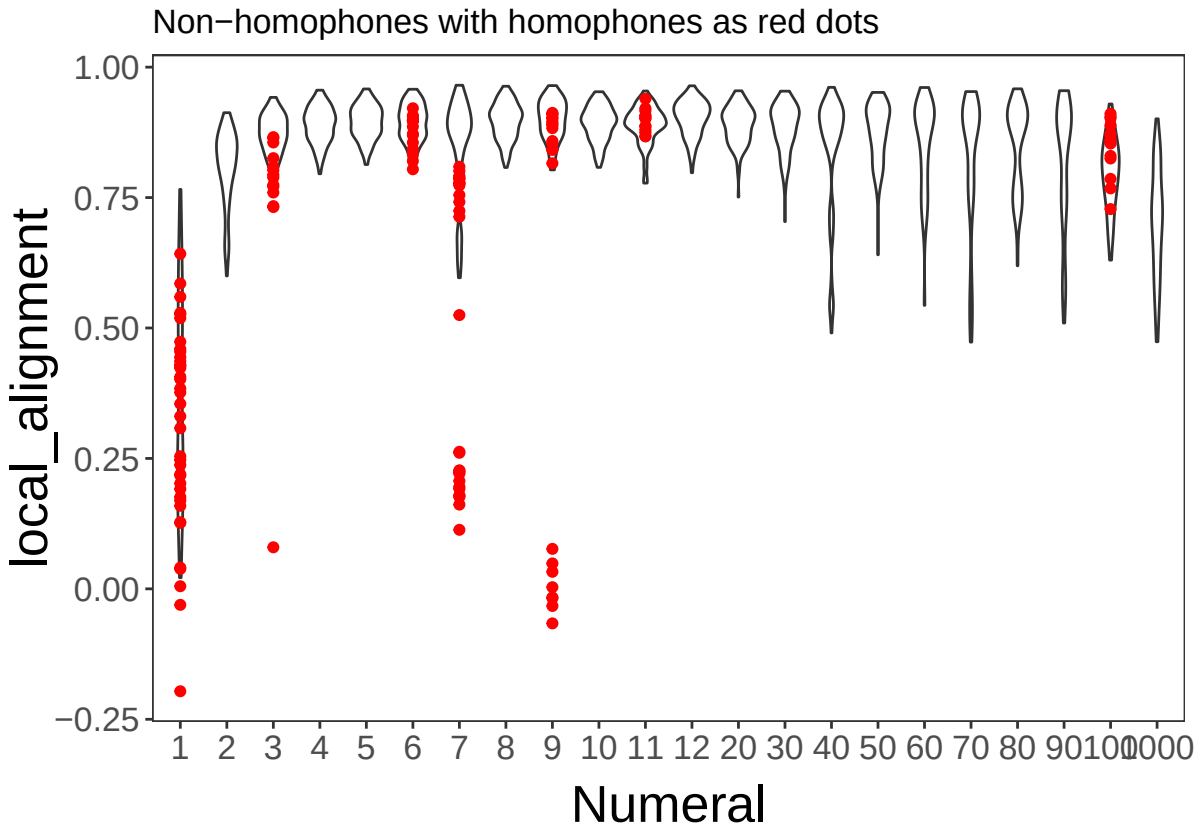
This may be because French 9 ("neuf") can mean '9' or "new". We used the North Euralex dictionary to find numbers which have alternative referential classes (see the file `NumberHomophones.csv`). Some of these will have referential classes with no related meanings (e.g. Norwegian 'tre' meaning '3' or 'tree'), and are thus homonyms. Some of these cases will be polysemous (the different meanings are related), as in the case of Hungarian 'hét' meaning '7' or 'week' (see below).

```
h[,c("l","number","otherMeanings")]
```

```
##    l number   otherMeanings
## 1 no      3 Baum::N;Holz::N
## 2 sv     11      zittern::V
## 3 fr    100         Blut::N
## 4 fr      9          neu::A
## 5 uk      1     allein::ADV
## 6 ru      1     allein::ADV
## 7 fi      6        Tanne::N
## 8 hu      7        Woche::N
```

The plot below shows the distribution of non-homophones, with homophones drawn as dots. For 7 and 9, these fall outside of the general distribution, but there are several other cases where homophones look similar to the rest of the distribution:

```
ggplot(numbers[!numbers$homophone,],
       aes(y=local_alignment,
           x=factor(number_numeric))) +
  geom_violin() +
  geom_point(data=numbers[numbers$homophone,],
             aes(y=local_alignment,
           x=factor(number_numeric)),
           colour="red") +
  myThemeBasic +
  xlab("Numeral") +
  ggtitle("Non-homophones with homophones as red dots")
```


Non−homophones with homophones as red dots
local_alignment
1.00
0.75
0.50
0.25
0.00
−0.25
1 2 3 4 5 6 7 8 9 10 11 12 20 30 40 50 60 70 80 90 100 1000
Numeral

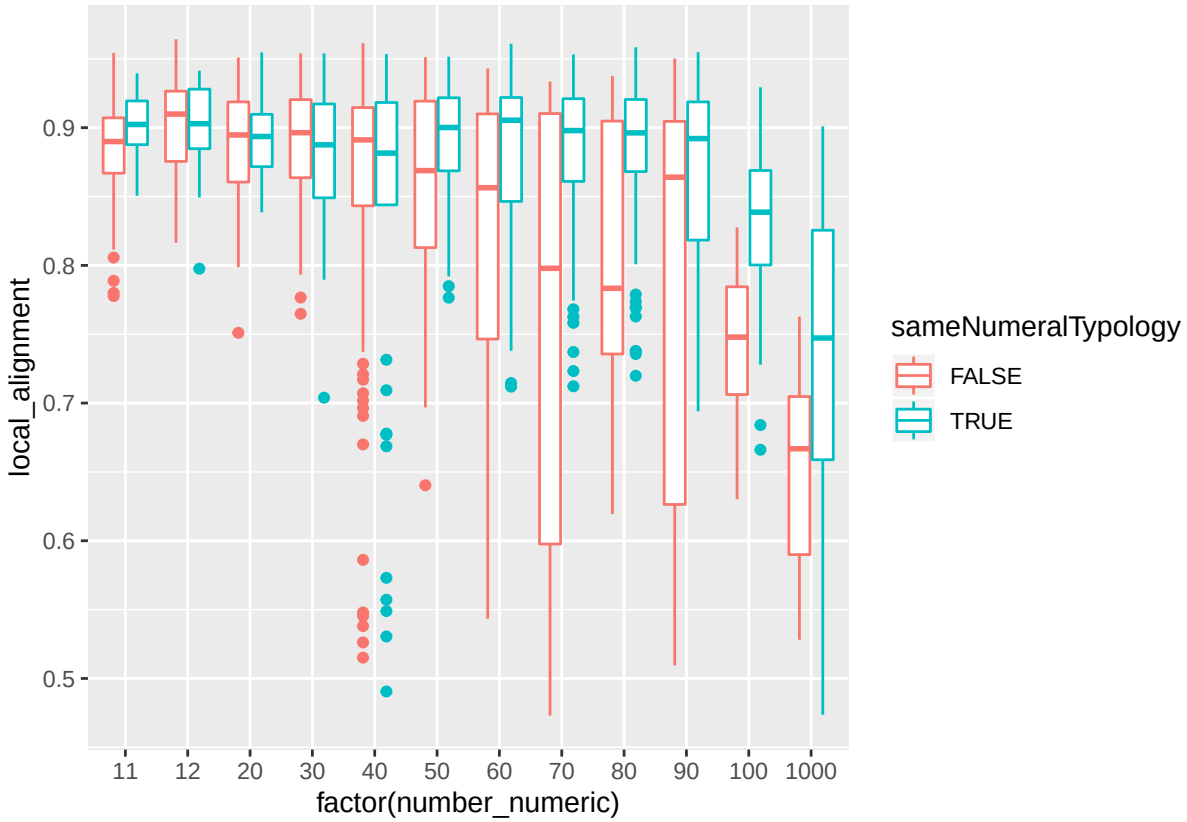
Number line

We can look at the importance of the different ways that numeral systems are compased. We use data from Calude & Verkerk (2016)'s typology which describe the composition of numerals in many Indo-European langages (what Calude & Verkerk call the 'number line'). For example, the numeral 'four' in English is an 'atom' (it is not composable into smaller parts), while 'fourteen' is composed of two atoms (four + ten). There are differences between languages in terms of whether they use an atom or a composition of two atoms. For example, English uses a unique atom to represent 12 ('twelve'), while Bulgarian uses a form that is composed as "2 + 10" (2 = dve, 10 = deset, 12 = dvanadeset). There are also differences between languages in how they compose some numerals. For example, 'eighty' in English is constructed as 8 x 10 but in French 'quatre-vingts' is 4 x 20. We used this data to identify whether two languages have the same system for forming a particular numeral.

We note that, in our sample, there are no differences between languages in the number line for numerals from 0 to ten. This is because all numbers below 10 for all languages in the sample are atoms. Therefore, the measure of numeral compostion is only explains variation for higher numerals.

It looks like there's an interaction between numeral typoogy and numeric value:

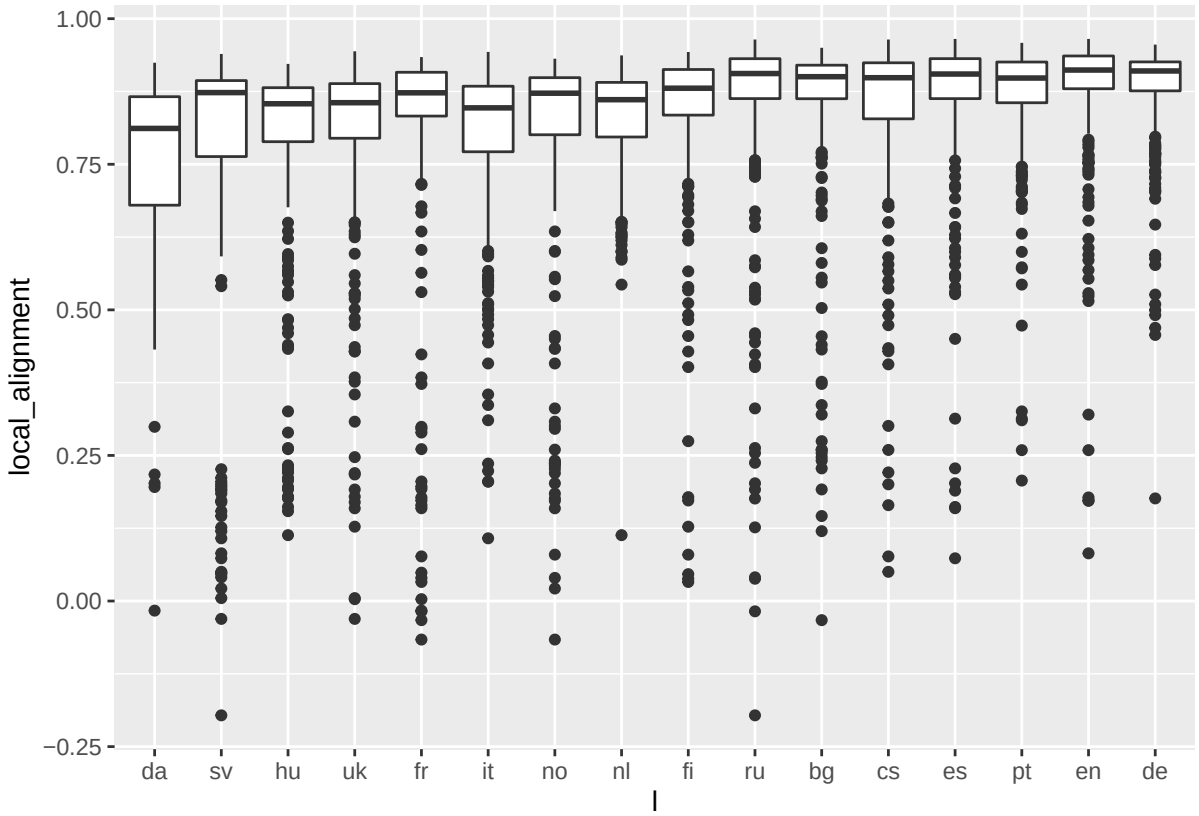
```
ggplot(numbers[numbers$number_numeric>10,],
       aes(y=local_alignment,
           x=factor(number_numeric),
           colour=sameNumeralTypology)) +
  geom_boxplot()
```



Variation by language

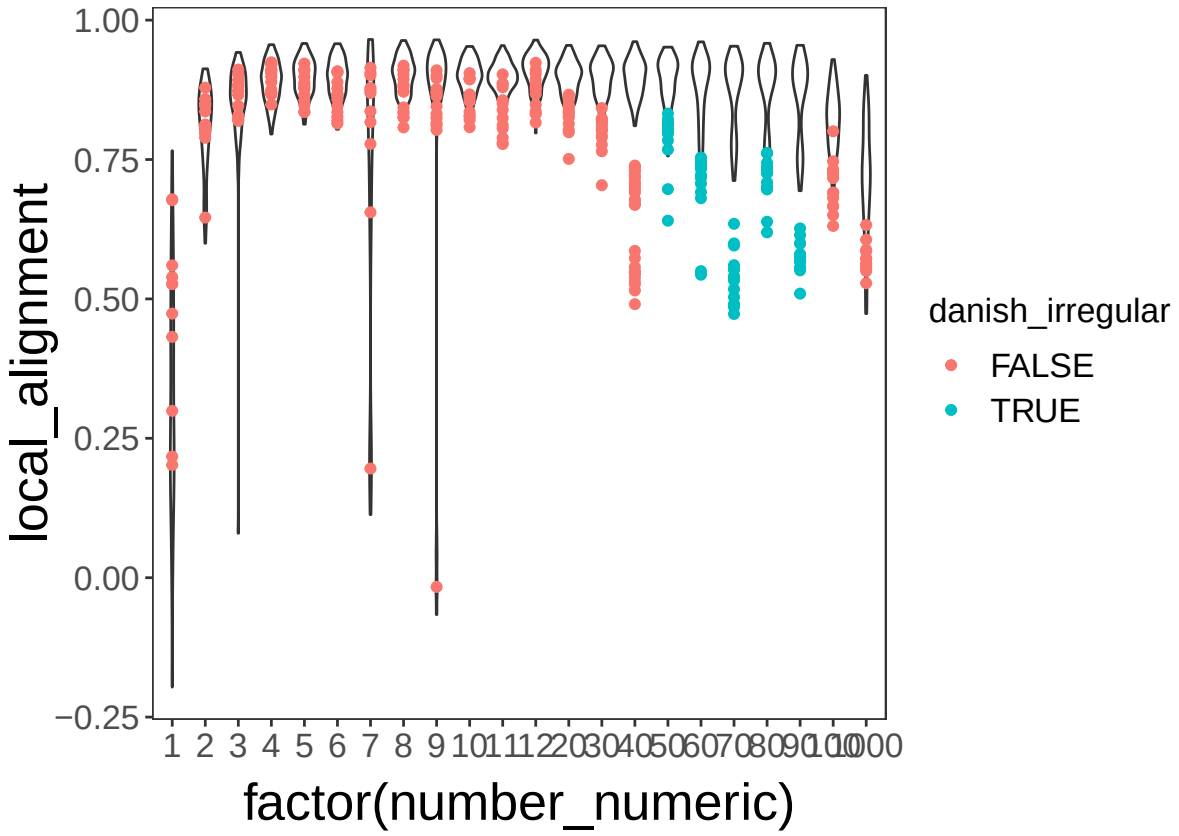
The plot below shows the overview of comparisons by language:

```
ggplot(langAverages,aes(x=l,y=local_alignment)) + geom_boxplot()
```



Danish seems to have a lower average. We note that some Danish 'crowns' are irregular (50,60,70,80,90). These pick out many outliers (dots are danish, blue dots are irregualr, violin plots are the rest of the data):

```
ggplot(numbers[!numbers$is_danish,],
       aes(x=factor(number_numeric),y=local_alignment)) +
  geom_violin() + myThemeBasic +
  geom_point(data=numbers[numbers$is_danish,],
             aes(colour=danish_irregular))
```



Formal test of effect of difference between regular and irregular numbers within Danish:

```
summary(lm(local_alignment~irreg_in_danish, data=
  numbers[numbers$is_danish,]))
```

```
##
## Call:
## lm(formula = local_alignment ~ irreg_in_danish, data = numbers[numbers$is_danish,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79901 -0.07008  0.04585  0.08937  0.16970
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.782552   0.008702  89.931  < 2e-16 ***
## irreg_in_danishTRUE -0.119891   0.018609  -6.442  4.5e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1356 on 309 degrees of freedom
## Multiple R-squared:  0.1184, Adjusted R-squared:  0.1156
## F-statistic: 41.51 on 1 and 309 DF,  p-value: 4.5e-10
```

Variation by frequency

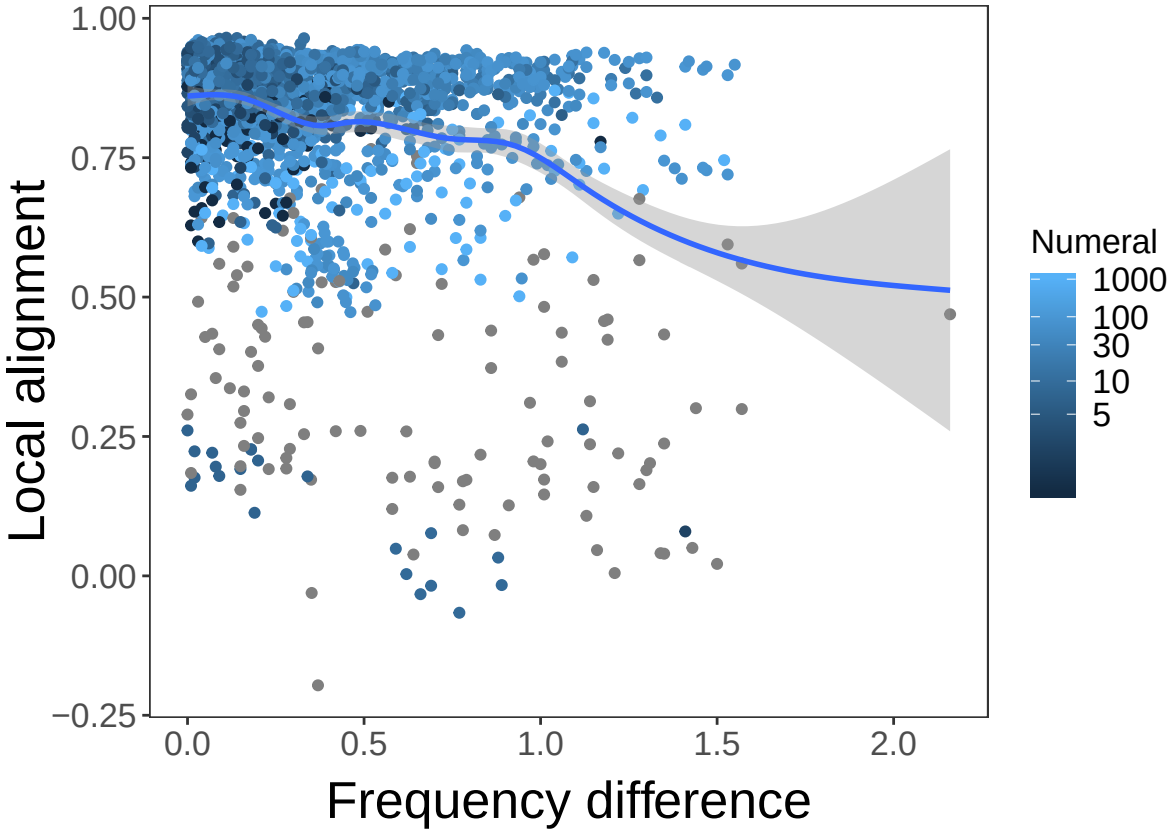
Semantic alignment by frequency (brighter colours are higher numeric values):

```
ggplot(numbers,
       aes(x=freqDiff,y=local_alignment,colour=log10(number_numeric))) +
  geom_point() +
  stat_smooth() +
  scale_colour_gradient(name = "Numeral", trans = "log",
                        breaks = log10(c(1,5,10,30,100,1000)), labels =c(1,5,10,30,100,1000)) +
  myThemeBasic + xlab("Frequency difference") +
  ylab("Local alignment")
```

```
## Warning: Transformation introduced infinite values in discrete y-axis
```

```
## Warning: Transformation introduced infinite values in discrete y-axis
```

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



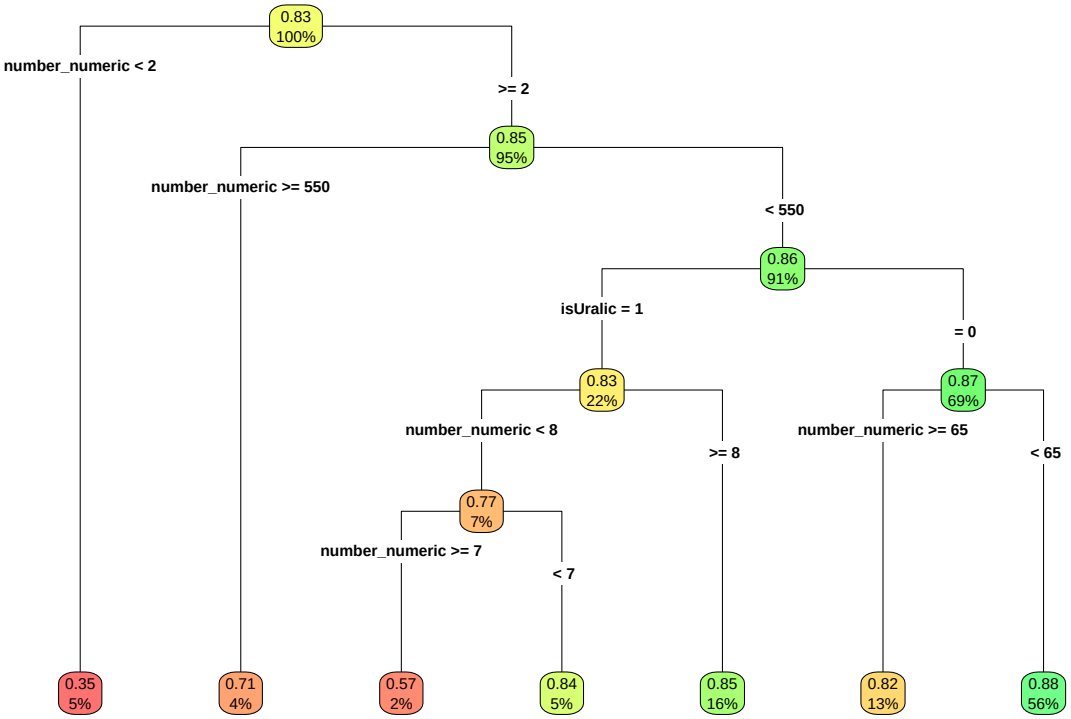
There appears to be a slight effect for larger frequency differences to be associated with lower alignment.

Decision tree

We use a decision tree to explore the data and find coherent clusters in the data. We try to predict local alignment based on various properties.

First we show that Uralic "7s" are a coherent cluster: A decision tree divides the data into categories that represent '1', '1000', and then combines divisions in the numeric value with 'isUralic' to find the cluster of Uralic 7s:

```
rt = REEMtree(local_alignment~
                number_numeric + isUralic,
              random = ~1|lang_pair,
              data=numbers)
rp.rt = tree(rt)
rp.rt$model = numbers
rpart.plot(rp.rt, type=4, branch.lty=1, clip.facs = F, box.palette="RdYlGn")
```



To help this, we include an explicit factor for seven. The final factors in the mode are:

- numeric value
- is a Uralic language
- is a Danish irregular
- is '7'
- has a homophone (an alternative referential class)
- the 'n' variable: number of comparisons possible between language pairs in the whole corpus
- whether the numbers have the same underling compositional structure

```
set.seed(1283)
rt = REEMtree(local_alignment~
                number_numeric + isUralic +
                danish_irregular + hist.dist2 +
                seven + n + sameNumeralTypology + homophone,
              random = ~1|lang_pair,
```

```
              data=numbers,
              MaxIterations=1000000)
rp.rt = tree(rt)
rp.rt$model = numbers
plot1 = rpart.plot(rp.rt, type=4, branch.lty=1, clip.facs = F, box.palette="RdYlGn")

cluster = factor(rp.rt$where,
                 labels = c("One",
                            "Hungarian 7",
                            "French 9",
                            "Small\nhomophones",
                            "Large\nhomophones",
                            "100,1000",
                            "Danish\nirregulars",
                            "Two",
                            "3-90"))

plot2 = ggplot(numbers,aes(y=local_alignment,
                    x=cluster)) +
  geom_violin() + myThemeBasic +
  xlab("") + ylab("Semantic alginment")

pdf("rDecisionTree.pdf",width=12,height=8)
  layout(t(t(c(1,2))), heights=c(2.5,1))
  par(mar=c(1,10,1,1))
  rpart.plot(rp.rt, type=4, branch.lty=2,
             clip.facs = F, box.palette="RdYlGn",
             mar=c(1,4,1,1.5),cex = 1.2,split.yshift=1)
  plot.new()
  vps <- baseViewports()
  pushViewport(vps$figure)
  vp1 <-plotViewport(c(0,0,0,0))
  print(plot2,vp = vp1)
dev.off()
```

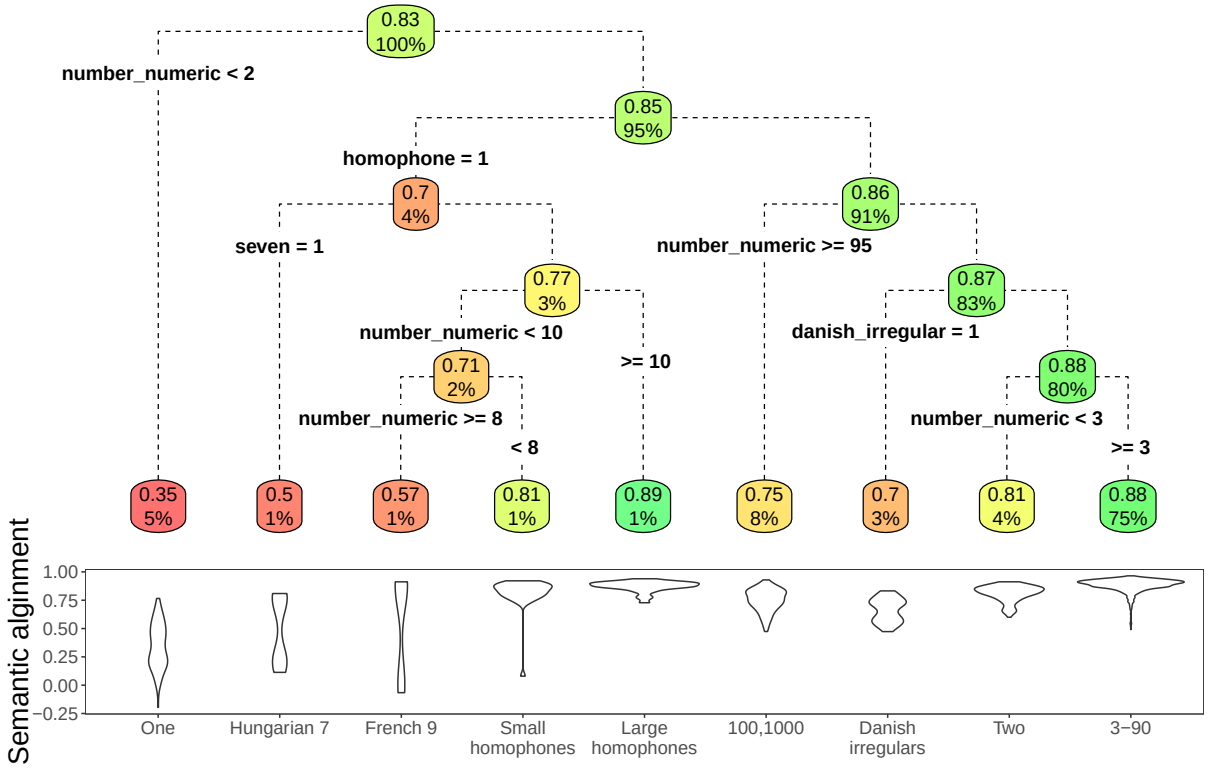


Figure 1: Decision tree predicting the semantic alignment of numerals (upper panel). Each node shows the average alignment of data under that node, and the proportion of data that is represented under that node. Each split in the tree splits the data according to the labelled criteria (e.g. the first split divides the number one from all other numbers). The lower panel shows the distribution of semantic alignment values for the data at each tip of the tree.

```
varimp = sort(rt$Tree$variable.importance)
varimp.plot = ggplot(data.frame(importance=varimp,
                  variable=factor(names(varimp),levels = names(varimp))),
       aes(y=importance,x=variable))+
  geom_col() + coord_flip()
pdf("../../results/numbers/VarImp.pdf")
varimp.plot
dev.off()
```

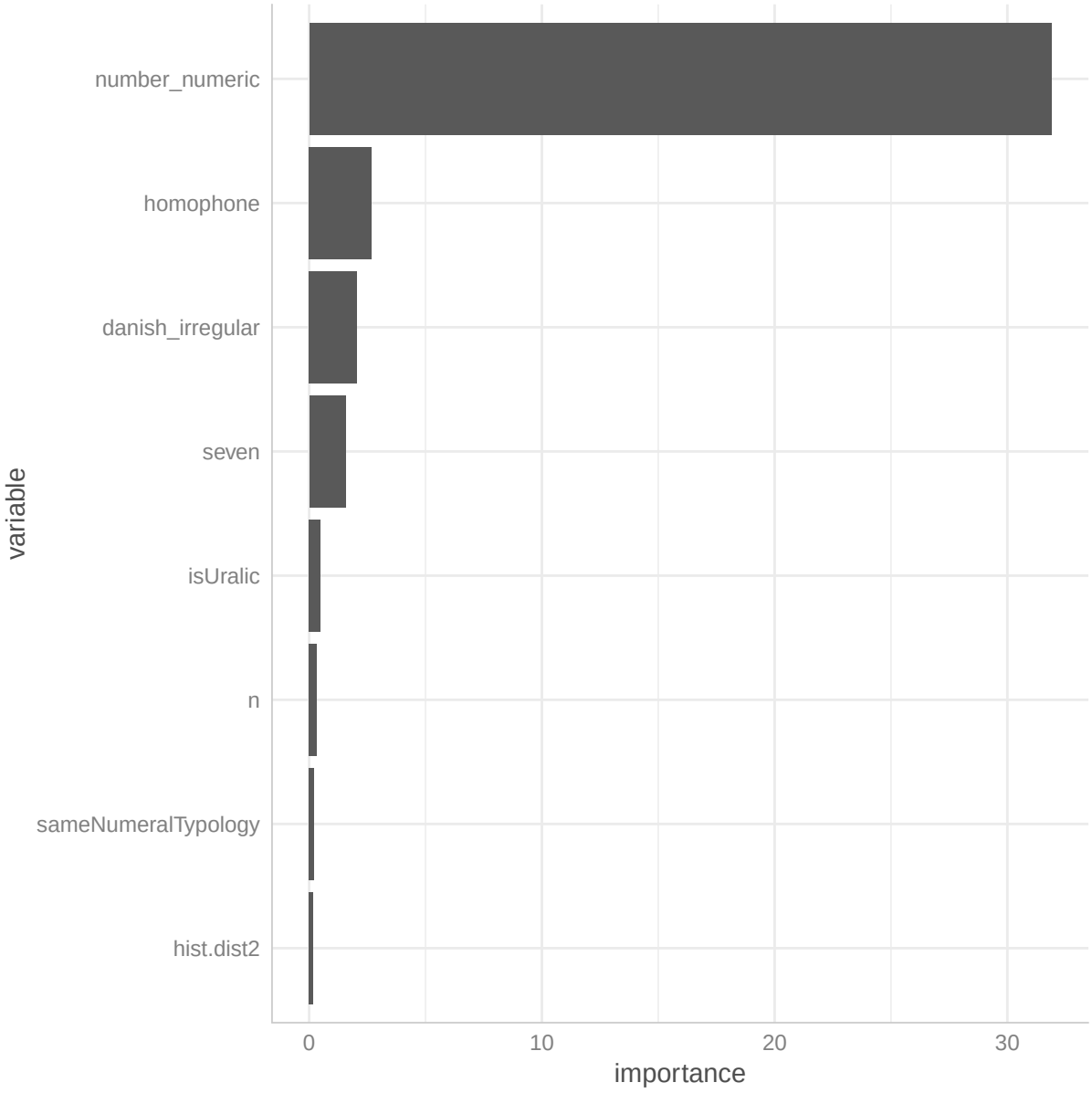


Figure 2: Variable importance measures for the decision tree above

Run a GAM

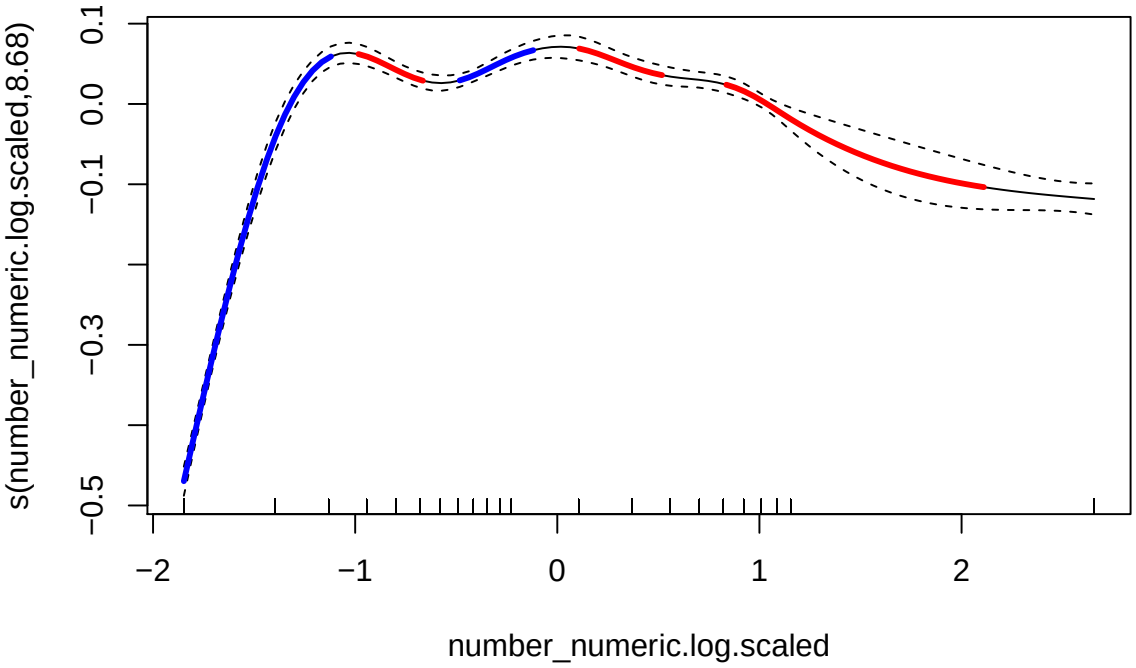
Convert lang_pair to factor and scale variables:

```
numbers$lang_pair = factor(numbers$lang_pair)

numbers$number_numeric.log = log(numbers$number_numeric)
numbers$number_numeric.log.scaled = scale(numbers$number_numeric.log)

numbers$local_alignment.center = scale(numbers$local_alignment,scale=F)

numbers$isUralic = factor(numbers$family_l1 == "Uralic" | numbers$family_l2 == "Uralic")
numbers$differentNumeralTypology = factor(!numbers$sameNumeralTypology)
numbers$seven = factor(numbers$seven)
numbers$danish_irregular = factor(numbers$danish_irregular)
numbers$homophone = factor(numbers$homophone)
```

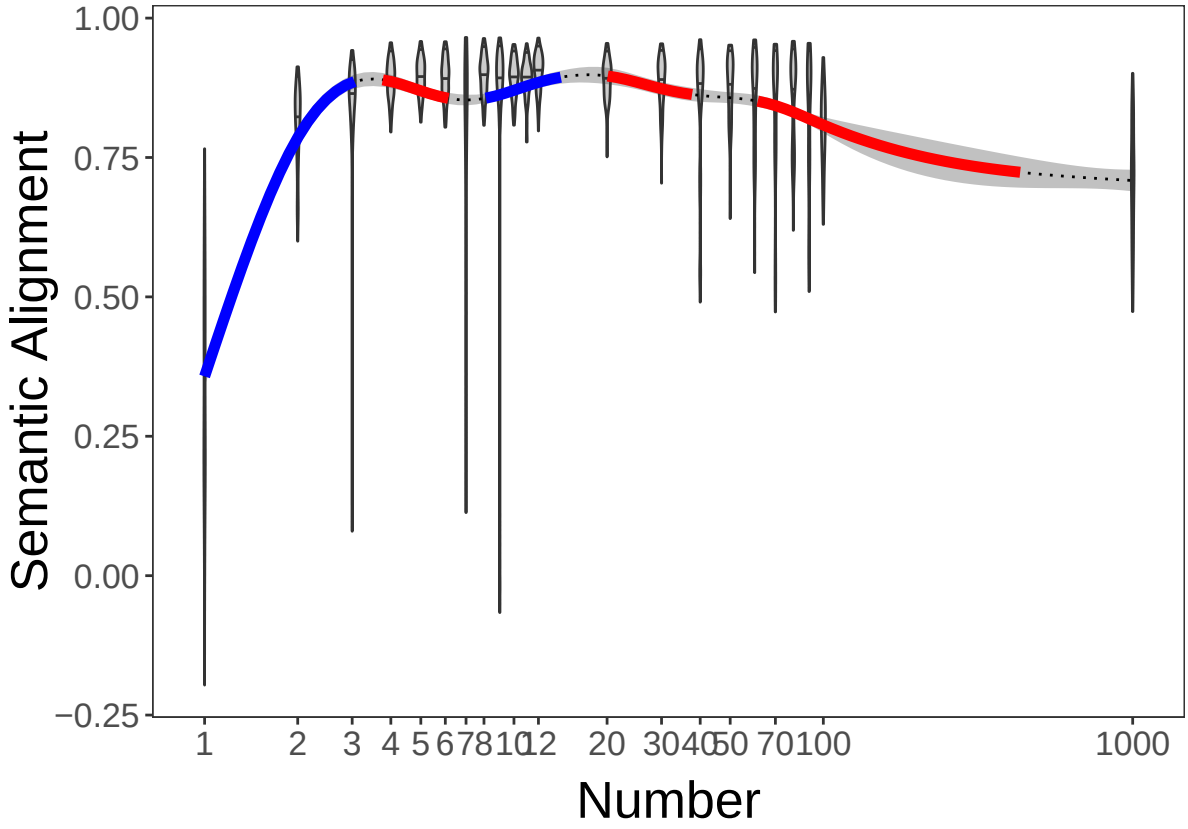
We start by looking at a simple model that has a random effect for langauge pair and a main smooth term for number. Note that since this is a non-linear model, a random effect for numeric value is very similar to a fully articulated smooth slope, so we just model numeric as a fixed effect.

```
m0 = bam(local_alignment.center ~
           s(lang_pair,bs='re') +
           s(number_numeric.log.scaled),
         data = numbers)
```

We plot the fit of the model below:




```
## Warning: Removed 61 rows containing missing values (geom_path).
## Warning: Removed 31 rows containing missing values (geom_path).
```



This shows the difference for 1, a dip for 7 (which the analyses above suggest is due to Uralic languages) and a decrease for 1000.

We now fit a full model with many other predictors:

- Number typology
- Interaction between numeric value and typology
- Historical distance (assuming Uralic is maximum distance)
- Whether the comparison is with a Uralic language
- Whether the number is 7
- Interaction between Uralic and 7
- Whether the number is a Danish irregular
- Whether the number word has a frequent alternative referential class (homophone or synonym)
- The frequency difference between the forms

```
m1 = bam(local_alignment.center~
           s(number_numeric.log.scaled, by=differentNumeralTypology) +
           s(hist.dist2) +
           s(lang_pair,bs='re') +
           isUralic*seven + danish_irregular +
           differentNumeralTypology +
           homophone +
           s(hist.dist2) +
           s(freqDiff),
         data = numbers)
```

Compare models. Adding the extra factors makes a difference.

```
lrtest(m0,m1)
```

```
## Likelihood ratio test
##
## Model 1: local_alignment.center ~ s(lang_pair, bs = "re") + s(number_numeric.log.scaled)
## Model 2: local_alignment.center ~ s(number_numeric.log.scaled, by = differentNumeralTypology) +
##     s(hist.dist2) + s(lang_pair, bs = "re") + isUralic * seven +
##     danish_irregular + differentNumeralTypology + homophone +
##     s(hist.dist2) + s(freqDiff)
##      #Df LogLik     Df  Chisq Pr(>Chisq)
## 1 109.15 2199.7
## 2 117.38 2518.2 8.2261 637.08  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## local_alignment.center ~ s(number_numeric.log.scaled, by = differentNumeralTypology) +
##     s(hist.dist2) + s(lang_pair, bs = "re") + isUralic * seven +
##     danish_irregular + differentNumeralTypology + homophone +
##     s(hist.dist2) + s(freqDiff)
##
## Parametric coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.017116   0.004897   3.495 0.000483 ***
## isUralicTRUE                 0.001801   0.009772   0.184 0.853816
## sevenTRUE                   -0.029093   0.013742  -2.117 0.034367 *
## danish_irregularTRUE        -0.144641   0.013097 -11.044  < 2e-16 ***
## differentNumeralTypologyTRUE  0.024844   0.032019   0.776 0.437883
## homophoneTRUE               -0.112477   0.008755 -12.847  < 2e-16 ***
## isUralicTRUE:sevenTRUE      -0.191216   0.020627  -9.270  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                                edf  Ref.df
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE  8.598   8.944
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE   3.986   4.514
## s(hist.dist2)                                                1.000   1.000
## s(lang_pair)                                                92.082 117.000
## s(freqDiff)                                                  1.000   1.000
##                                                                  F
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE 305.027
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE   45.087
## s(hist.dist2)                                                0.671
## s(lang_pair)                                                 3.532
## s(freqDiff)                                                 24.254
##                                                             p-value
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE  < 2e-16 ***
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE   < 2e-16 ***
```

```
## s(hist.dist2)                                                  0.413
## s(lang_pair)                                                 < 2e-16 ***
## s(freqDiff)                                                 9.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.699   Deviance explained = 71.3%
## fREML = -2301.9  Scale est. = 0.0076339  n = 2415
```

Summary of main model

The final model explains 71.33% of the deviance, compared to 62.67% in the baseline model.

Parametric effects:

- Being an Uralic language is nota strong predictor.
- The number 7 has lower alignment in general (β = -0.0291 , p = 0.0344)
- Danish irregulars have lower alignment (β = -0.145 , p = $<$ 0.001))
- Overall, there is no difference for comparisons between numbers with different numeral typologies (β = 0.0248 , p = 0.4379)
- Alignment is lower for comparisons between words were at least one has an alternative referential class (homophone or synonym) (β = -0.112 , p = $<$ 0.001)
- Comparisons with Uralic sevens are significantly lower in alignment (β = -0.191 , p = $<$ 0.001)

Smooth effects:

- The EDF for numeral is high, indicating strong non-linearities.
- There is an interaction between numeral value and numeral typology. The graph below shows that alignment is lower for comparisons between numbers with different typologies for numbers above 50. Also note that there is a slight increase in the semantic alignment for number 20 if the two languages have the same numeral typology.
- There was no effect of historical distance.
- There were significant random effects for particular language pairs.
- There was a significant linear effect of frequency difference: Alignment was lower when the frequency difference was larger.

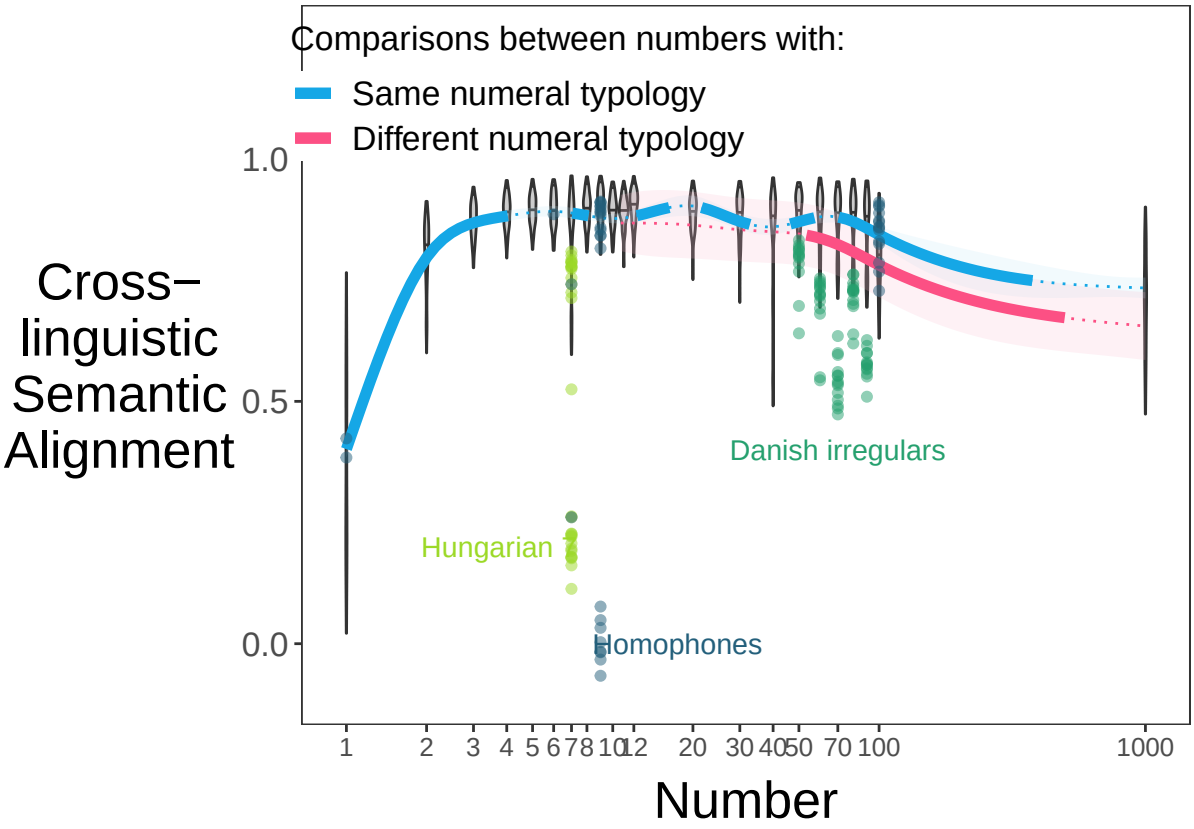
The hidden code below (see the Rmd file) detects significantly steep slopes in the GAM curve. Thick line segments indicate significant rises or decreases. We see that 1 and 2 have lower alignment, then numbers 3-20 are fairly constant. Beyond 20, there is a decrease in alignment, especially for numbers with different numeral typologies. Various outliers captured by the model are drawn on top. Note that there are no numbers below 10 that have different numeral typologies, so we have truncated the curve accordingly.

```
gamPlot
```

```
## Warning: Removed 34 rows containing missing values (geom_path).
```

```
## Warning: Removed 141 rows containing missing values (geom_path).
```

```
## Warning: Removed 109 rows containing missing values (geom_path).
```




```
pdf("../../results/numbers/FinalGamModel.pdf", width=7.5,height=4.5)
gamPlot
```

```
## Warning: Removed 34 rows containing missing values (geom_path).
## Warning: Removed 141 rows containing missing values (geom_path).
## Warning: Removed 109 rows containing missing values (geom_path).
```

```
dev.off()
```

```
## pdf
##   2
```

Controlling for linguistic history

In the model above, the effects of historical distance are minimal: the fit is lienar and not significant. This might be because there are random effects for language pairs which are taking up the variance. In the section below, we use only historical distance:

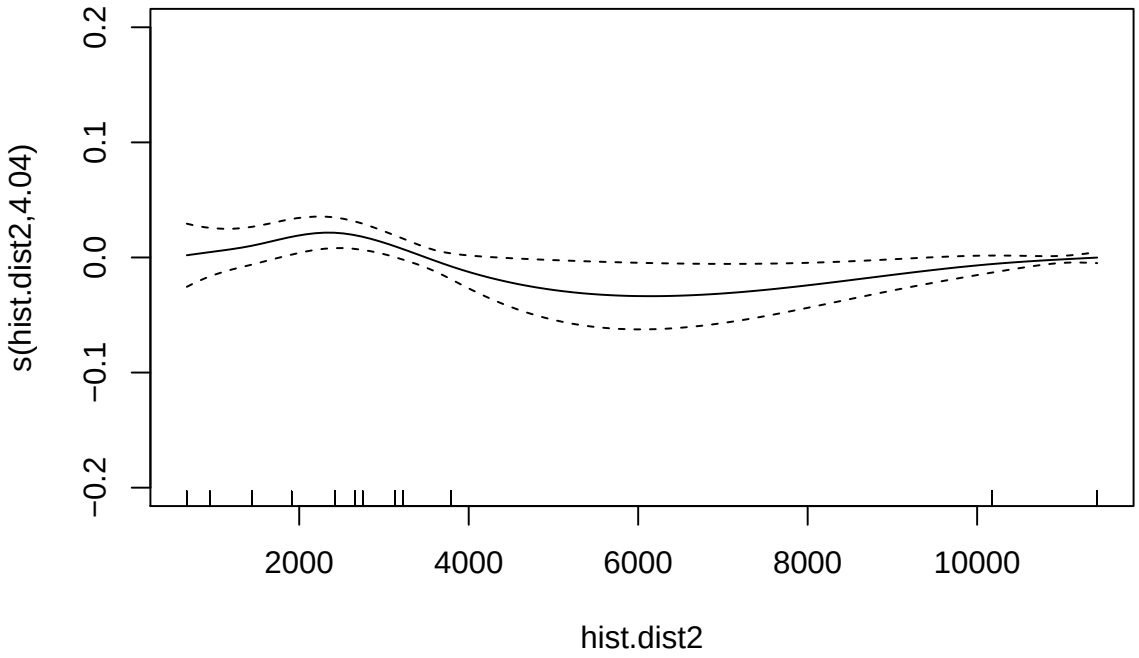
```
m1.phylo = bam(local_alignment.center~
           s(number_numeric.log.scaled, by=differentNumeralTypology) +
           s(hist.dist2) +
           seven + danish_irregular +
           homophone +
           differentNumeralTypology +
           s(freqDiff),
         data = numbers[!is.na(numbers$hist.dist),])
# Model with numeric
summary(m1.phylo)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## local_alignment.center ~ s(number_numeric.log.scaled, by = differentNumeralTypology) +
##     s(hist.dist2) + seven + danish_irregular + homophone + differentNumeralTypology +
##     s(freqDiff)
##
## Parametric coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.017078   0.002922   5.845    6e-09 ***
## sevenTRUE                   -0.036295   0.014328  -2.533   0.0114 *
## danish_irregularTRUE        -0.189497   0.013823 -13.709   <2e-16 ***
## homophoneTRUE               -0.112117   0.010836 -10.347   <2e-16 ***
## differentNumeralTypologyTRUE  0.048179   0.072356   0.666   0.5056
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                                edf Ref.df
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE 8.535  8.927
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE  4.697  5.104
## s(hist.dist2)                                              4.042  4.587
## s(freqDiff)                                                1.000  1.000
##                                                                F
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE 215.732
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE   40.902
## s(hist.dist2)                                                2.498
## s(freqDiff)                                                 62.586
##                                                            p-value
## s(number_numeric.log.scaled):differentNumeralTypologyFALSE  < 2e-16 ***
## s(number_numeric.log.scaled):differentNumeralTypologyTRUE   < 2e-16 ***
## s(hist.dist2)                                                0.0271 *
## s(freqDiff)                                                4.31e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## R-sq.(adj) =  0.674   Deviance explained = 67.8%
## fREML = -1731.5  Scale est. = 0.0081138  n = 1819
```

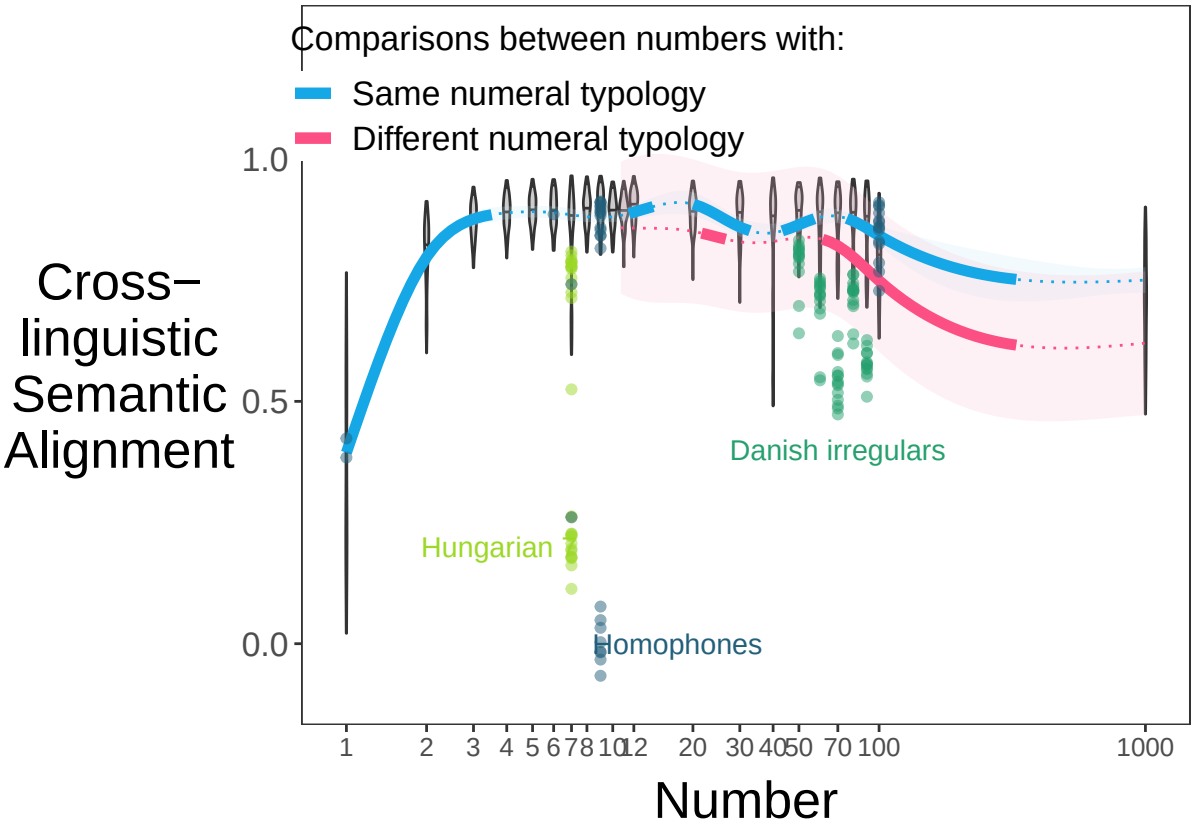
The effects are very similar. The effect of history is significant, but not large. Semantic alignment is lower for more distantly related languages:

```
plot(m1.phylo,select=3,ylim=c(-0.2,0.2))
```



Plot the main effects, which look much like the main graph above.

```
## Warning: Removed 34 rows containing missing values (geom_path).
## Warning: Removed 141 rows containing missing values (geom_path).
## Warning: Removed 119 rows containing missing values (geom_path).
```


Comparisons between numbers with:
Same numeral typology
Different numeral typology
Cross−linguistic Semantic Alignment
1.0
0.5
0.0
Danish irregulars
Hungarian
Homophones
1
2
3
4
5
6
7
8
10
12
20
30
40
50
70
100
1000
Number

Conclusion

Semantic alignment for number words is generally high, though there are some differences that can be explained (the model explained 71.33% of the deviance). 1 and 2 have lower alignment due to often being grammaticalised as indefinite or dual marker (Givon, 1981). Numbers 3-12 generally have high alignment (mean local alignment = 0.87), and higher numbers decline in alignment up to 1000. There are also language-specific differences due to how numerals are constructed (e.g. base, combination rules, see Calude & Verkerk, 2016), or for irregular forms (e.g. 50, 60, 70, 80 and 90 in Danish). Some number words have alternative associations due to homophones or polysemies (e.g. the Hungarian 7 is used directly to mean 'week', and 'neuf' in French means '9' or 'new'). The historical distance between languages did not predict much of the variation.

The differences in semantic alignment may appear either because (A) the semantic associations are different for different languages, or (B) as a side-effect of numbers appearing with different words skewing the alignment metric. Effects that support (B) could include:

- Difference in frequency. Lower frequency terms will appear in a smaller range of contexts and the semantic alignment estimates may be more stochastic.

- Alternative referential class (homophone or synonym). Different meanings will contribute different semantic relations.

However, it is more difficult to explain why different numeral typologies would lead to semantic differences, unless the way numbers are constructed affects the way people think about the numbers.

References

Calude, A. S., & Verkerk, A. (2016). The typology and diachrony of higher numerals in Indo-European: a phylogenetic comparative study. Journal of Language Evolution, 1(2), 91-108.

Dehaene, S., & Mehler, J. (1992). Cross-linguistic regularities in the frequency of number words. Cognition, 43(1), 1-29.

Givón, T. (1981). On the development of the numeral 'one'as an indefinite marker. Folia Linguistica Historica, 15(Historica vol. 2, 1), 35-54.

4.3 Neighbour net visualisation of semantic alignment

The semantic alignment between languages was converted into a semantic distance and was visualised as a Neighbour Net using Splitstree (Huson & Bryant, 2006). This was done on the data with both the language filter and the concept filter. The code for building the Neighbour Net and the resulting nexus file are here:

https://github.com/seannyD/ImputeEACulturalDifferences/blob/master/processing/makeSplitstreeNexusFile.R.

https://github.com/seannyD/ImputeEACulturalDifferences/blob/master/results/splitstree/LinguisticDistances.nex.

Neighbour Nets visualise a matrix of distances between individuals (in this case, individuals are languages). They attempt to draw a line between each pair of individuals whose length reflects the distances in the matrix. A set of individuals that were all equally distant from each other would be represented as several lines of equal length originating from a single point, with the individuals placed at the tips (a 'star' phylogeny). If distances follow a strictly diverging, binary branching tree (pure vertical transmission), then the visualisation would look like a branching tree (a phylogeny), with more closely related individuals 'clustered' together. However, many cases of cultural evolution also involve horizontal transmission, where there is transmission between individuals who have previously diverged. This leads to conflicting signals: individuals with elements inherited not just from their immediate ancestors, but from other parts of the tree. In a neighbour net, extra parallel lines are drawn along the underlying phylogeny structure. Ideally, the distance between two individuals will still be represented accurately by the shortest route between them following the lines. An individual with close ties to two other individuals will have alternative routes to reach each one. At the same time, a part of the tree that had only vertical transmission could be drawn as a phylogeny. This leads to a web-like structure that has properties of networks where there is horizontal signal but also some clustering properties of trees where there is vertical signal. For more on the use of neighbour nets in linguistics, see e.g. McMahon et al. (2007) or Skirgard et al., (2017).

Figure 4.3.1 shows the semantic distances for all languages in the sample (Delta score = 0.3844, Q-residual score = 0.0008379). Languages which are joined by short lines have higher semantic alignment (translation equivalents are closer in meaning). Languages are roughly grouped by language family, with the Indo-European group being most clear, but also Uralic, Turkic and Afro-Asiatic groups visible. The figure also reflects geographic distances, with Persian and Armenian showing up on the "Eastern" side. There is some clear conflicting signal for English between the Romance and Germanic, which reflects its mixed history (see e.g. McWhorter, 2008). This figure with languages from many different language families is somewhat misleading, since many languages that have high distances appear "together" in space (e.g. Basque and Korean), when in reality the distances along the lines are much further than for e.g. most Indo-European languages. Therefore, we also produced a Neighbour Net just for Indo-European languages.

Figure 4.3.2 shows the semantic distances for Indo-European languages visualised as a neighbour-net (Delta score = 0.3062, Q-residual score = 0.003523). The semantic distances reflect established historical relationships, as shown by the labelling of the major sub-branches according to Glottolog (Hammarstrom et al.). Again, English shows conflicting signal between Germanic and Romance. There is a clear historical signal in the Slavic languages. Also, a split between 'eastern' and 'western' languages, with Romanian and Greek being halfway between the two. Hindi, Armenian, Lithuanian and Greek are more removed from the rest historically, and that's reflected in the fact that they're placed 'together', though actually the distance from Hindi to Armenian is much larger than for Russian to Ukrainian.

Some more speculative comments can also be made. The relationship between many languages reflects geographic proximity, such as the cline from Norwegian, Danish, German and

Dutch. The proximity of Spanish and Catalan may also reflect contact and bilingualism. Romanian is an outlier in the romance languages and has a large amount of borrowed words from slavic languages (Schulte, 2009), so it's not suprising that it shows up on the edge of the Romance cluster. The proximity of Bulgarian and Greek may reflect geographical proximity and historical ties such as the Ottoman empire. The proximity of Armenian and Hindi is not predicted by linguistic family trees (e.g. Glottolog), but there is actually a history of contact, with trade leading to a historical Armenian population in Hindi-speaking places like Agra (e.g. Ferrier, 1973), and maybe more indirect borrowings through related languages (Pisowicz, 1995).

References

Huson, D.H. and Bryant, D. (2006) Application of Phylogenetic Networks in Evolutionary Studies, Mol. Biol. Evol., 23(2):254-267.

McWhorter, J. H. (2008) Our magnificent bastard tongue: The untold history of English. Penguin.

Hammarstrm, H, Forkel, R, & Haspelmath, M. (2019) Glottolog 4.0. Jena: Max Planck Institute for the Science of Human History.

McMahon, A., Heggarty, P., McMahon, R., & Maguire, W. (2007). The sound patterns of Englishes: representing phonetic similarity. English Language & Linguistics, 11(1), 113-142.

Schulte, K. (2009) Loanwords in Romanian. In M. Haspelmath & U. Tadmor, *Loanwords in the World's Languages: A Comparative Handbook*. Walter de Gruyter. p. 243.

Skirgrd, H., Roberts, S. G., & Yencken, L. (2017). Why are some languages confused for others? Investigating data from the Great Language Game. PloS one, 12(4), e0165934.

Ferrier, R. W. (1973). The Armenians and the East India Company in Persia in the seventeenth and early eighteenth centuries. The Economic History Review, 26(1), 38-62.

Pisowicz, A. (1995). How did New Persian and Arabic Words Penetrate the Midle Armenian Vocabulary? Remarks on the Material in Kostandin Erznkac ?i?s Poetry?. New Approaches to Medieval Armenian Language and Literature, 3, 95.

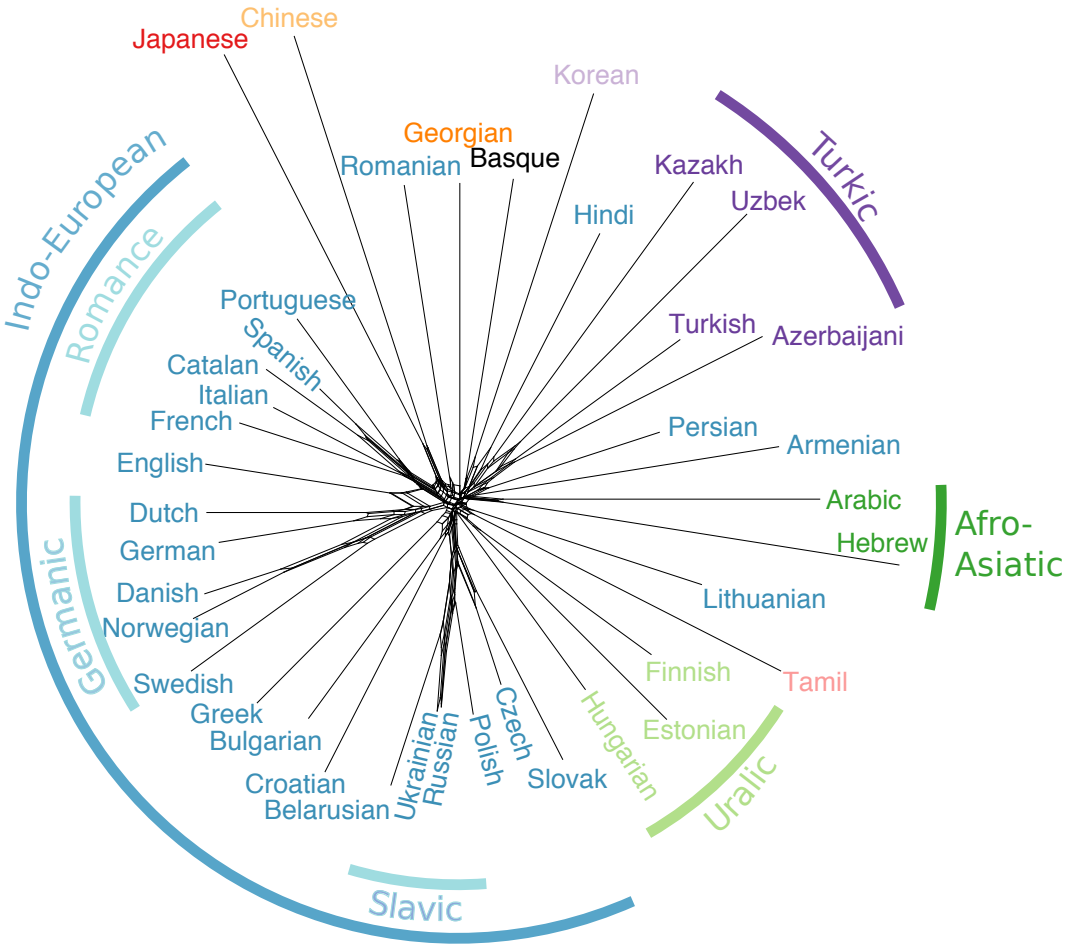


Figure 4.3.1: Neighbour Net reflecting semantic distance for all languages in the sample. Language labels are coloured by language family.

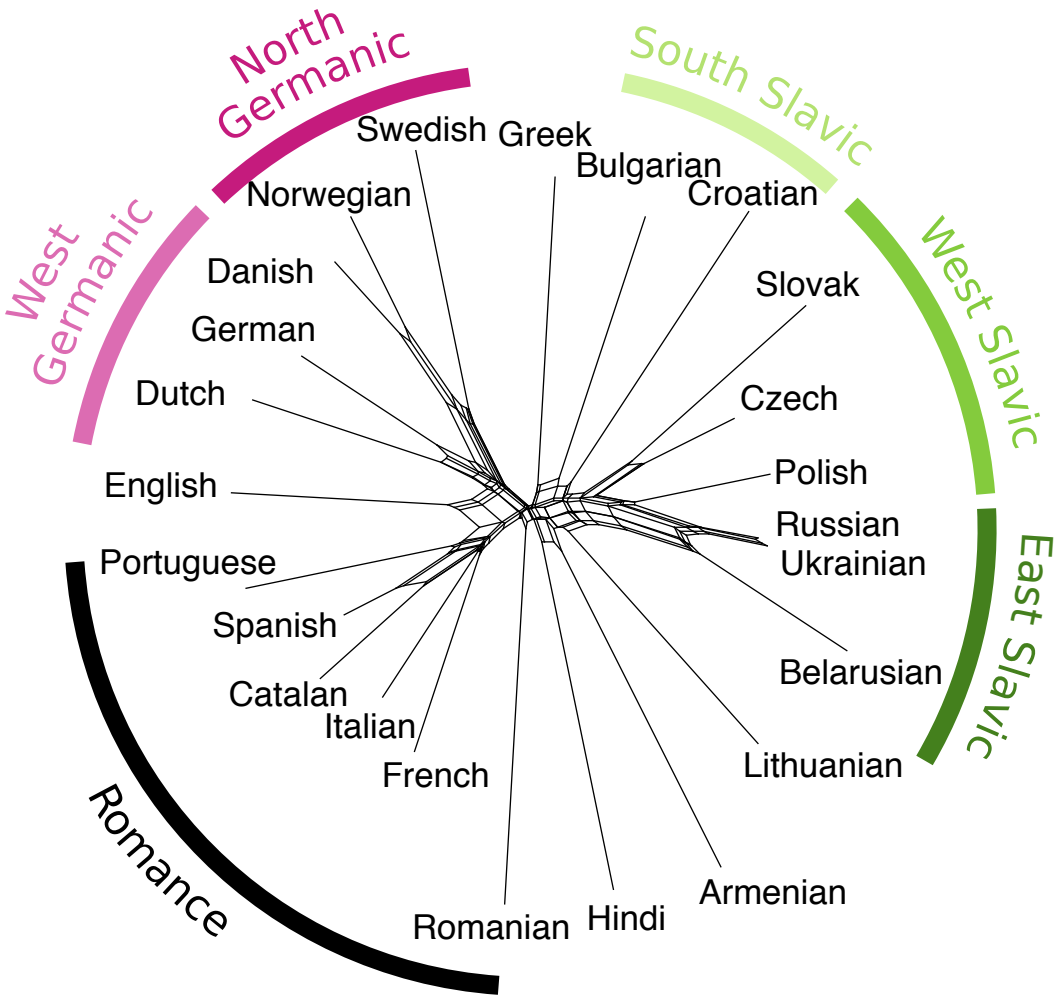


Figure 4.3.2: Neighbour Net reflecting semantic distance for Indo-European languages.

4.4 Cross-cultural analysis (Common Crawl data)

Predicting semantic alignment by cultural similarity: Common crawl data

Bill Thompson, Seán Roberts & Gary Lupyan

Contents

Introduction

This file replicates the tests for the main wikipedia data on the common crawl data.

Load libraries

```
library(ape)
library(ecodist)
library(lme4)
library(sjPlot)
library(ggplot2)
library(igraph)
library(lattice)
library(xtable)
```

Parameters (using data from Northuralex and common crawl, k=100, unfiltered):

```
datasetName = "cc"
datasetLabel = "Common Crawl"
lingDistancesFile = "../data/FAIR/nel-k100-cc-alignments-by-language-pair.csv"
lingDistancesFileNK = "../data/FAIR/nel-k100-cc-alignments-by-language-pair-without-kinsip.csv"
lingDistancesByDomainFile = "../results/EA_distances/nel-k100-cc_with_ling.csv"
# (generated by ../processing/combineCultAndLingDistances.R)
```

All domains

Load data

Read the cultural distances:

```
cult = read.csv("../results/EA_distances/CulturalDistances_Long.csv", stringsAsFactors = F)
names(cult) = c("l1","l2","cult.dist")
```

Add language family:

```
l = read.csv("../data/FAIR_langauges_glotto_xdid.csv", stringsAsFactors = F)
g = read.csv("../data/glottolog-languoid.csv/languoid.csv", stringsAsFactors = F)
l$family = g[match(l$glotto,g$id),]$family_pk
l$family = g[match(l$family,g$pk),]$name
```

Read the semantic distances

```
ling = read.csv(lingDistancesFile, stringsAsFactors = F)
```

There are very few possible comparisons for Slovenian and Northern Sami, so we'll remove these:

```
ling = ling[!(ling$l1=="se" | ling$l2 == "se"),]
ling = ling[!(ling$l1=="sl" | ling$l2 == "sl"),]
```

Combine the lingusitic and cultural distances. Note that we flip the cultural measure from a distance measure to a similarity measure.

```
cult$l1.iso2 = l[match(cult$l1,l$Language2),]$iso2
cult$l2.iso2 = l[match(cult$l2,l$Language2),]$iso2

fairisos = unique(c(ling$l1,ling$l2))
cultisos = unique(c(cult$l1.iso2, cult$l2.iso2))

cult = cult[(cult$l1.iso2 %in% fairisos) & (cult$l2.iso2 %in% fairisos),]
ling = ling[(ling$l1 %in% cultisos) & (ling$l2 %in% cultisos),]

matches = sapply(1:nrow(ling), function(i){
  which(cult$l1.iso2==ling$l1[i] & cult$l2.iso2==ling$l2[i])
})

ling$cult.dist = cult[matches,]$cult.dist
# Flip
ling$cult.dist = 1 - ling$cult.dist
# Scale
ling$cult.dist.center = scale(ling$cult.dist)
cdc.s = attr(ling$cult.dist.center,"scaled:scale")
cdc.c = attr(ling$cult.dist.center,"scaled:center")
ling$cult.dist.center = as.numeric(ling$cult.dist.center)
ling$comparison_count.center =
  scale(ling$comparison_count)

ling$family1 = l[match(ling$l1, l$iso2),]$family
ling$family2 = l[match(ling$l2, l$iso2),]$family
l[l$Language=="Arabic",]$autotyp.area= "Greater Mesopotamia"
l[l$Language=="Persian",]$autotyp.area= "Greater Mesopotamia"
ling$area1 = l[match(ling$l1, l$iso2),]$autotyp.area
```

```
ling$area2 = l[match(ling$l2, l$iso2),]$autotyp.area


fgroup = cbind(ling$family1,ling$family2)
fgroup = apply(fgroup,1,sort)
ling$family.group = apply(fgroup,2,paste,collapse=":")
agroup = cbind(ling$area1,ling$area2)
agroup = apply(agroup,1,sort)
ling$area.group = apply(agroup,2,paste,collapse=":")

ling$rho.center = scale(ling$local_alignment)
```

Each observation is now assocaited with a language family pair:

```
head(ling[,c("l1","l2","local_alignment",'family.group')])
```

```
##     l1  l2 local_alignment         family.group
## 3  myv  cv      0.05112061        Turkic:Uralic
## 5   la  cv      0.06646598 Indo-European:Turkic
## 6   cv sah      0.06823760        Turkic:Turkic
## 13  ga  cv      0.07949409 Indo-European:Turkic
## 19  cv  he      0.08650789  Afro-Asiatic:Turkic
## 20  cv  te      0.08761480     Dravidian:Turkic
```

And the same is true for area:

```
tail(ling[,c("l1","l2","local_alignment",'area.group')])
```

```
##      l1 l2 local_alignment              area.group
## 1112 uk ja       0.3804179 Inner Asia:N Coast Asia
## 1113 be ru       0.3821532   Inner Asia:Inner Asia
## 1115 uk be       0.4022741   Inner Asia:Inner Asia
## 1116 cs uk       0.4376378       Europe:Inner Asia
## 1118 cs ru       0.4581089       Europe:Inner Asia
## 1119 uk ru       0.5460480   Inner Asia:Inner Asia
```

Number of observations:

```
# Number of datapoints:
nrow(ling)
```

```
## [1] 308
```

```
# Number of unique languages:
length(unique(unlist(ling[,c("l1","l2")])))
```

```
## [1] 34
```

```
# Number of unique langauge families:
uniqueFamilies = unique(unlist(ling[,c("family1","family2")]))
length(uniqueFamilies)
```

```
## [1] 7
```

```
# Number of unique areas:
uniqueAreas = unique(unlist(ling[,c("area1","area2")]))
length(uniqueAreas)
```

```
## [1] 6
```

Cross-over between language famlies and areas:

```
tx = data.frame(lang= c(ling$l1,ling$l2),
           fam = c(ling$family1,ling$family2),
           area= c(ling$area1,ling$area2))
tx = tx[!duplicated(tx),]
table(tx$fam,tx$area)
```

```
##
##                 Europe Greater Mesopotamia Indic Inner Asia N Coast Asia
##   Afro-Asiatic       0                    1     0          0            0
##   Dravidian          0                    0     3          0            0
##   Indo-European     10                    2     1          5            0
##   Japonic            0                    0     0          0            1
##   Sino-Tibetan       0                    0     0          0            0
##   Turkic             0                    1     0          5            0
##   Uralic             1                    0     0          3            0
##
##                 Southeast Asia
##   Afro-Asiatic                0
##   Dravidian                   0
##   Indo-European               0
##   Japonic                     0
##   Sino-Tibetan                1
##   Turkic                      0
##   Uralic                      0
```

LMER models

Mixed effects model, predicting Linguistic similaritys from cultural similarity, with random intercept for family and area and random slope for cultural similarity for family and area.

We start with a null model with random intercepts for family and area, and random slopes for cultural similarity by both. We add a fixed effect of the number of comparisons made for each datapoint (number of concepts that were available to compare). Then we add a fixed effect of cultural similarity

```
m0 = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m0.5 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m1 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    cult.dist.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
an1 = anova(m0,m0.5,m1)
```

```
## refitting model(s) with ML (instead of REML)
```

```
an1
```

```
## Data: ling
## Models:
## m0: rho.center ~ 1 + (1 + cult.dist.center | family.group) + (1 +
## m0:     cult.dist.center | area.group)
## m0.5: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## m0.5:     family.group) + (1 + cult.dist.center | area.group)
## m1: rho.center ~ 1 + comparison_count.center + cult.dist.center +
## m1:     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
## m1:     area.group)
##      Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## m0    8 815.34 845.18 -399.67   799.34
## m0.5  9 797.25 830.82 -389.63   779.25 20.0908      1  7.385e-06 ***
## m1   10 794.31 831.61 -387.15   774.31  4.9442      1    0.02618 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cultural similarity is significantly correlated with Linguistic similarity. Here are the model estimates:

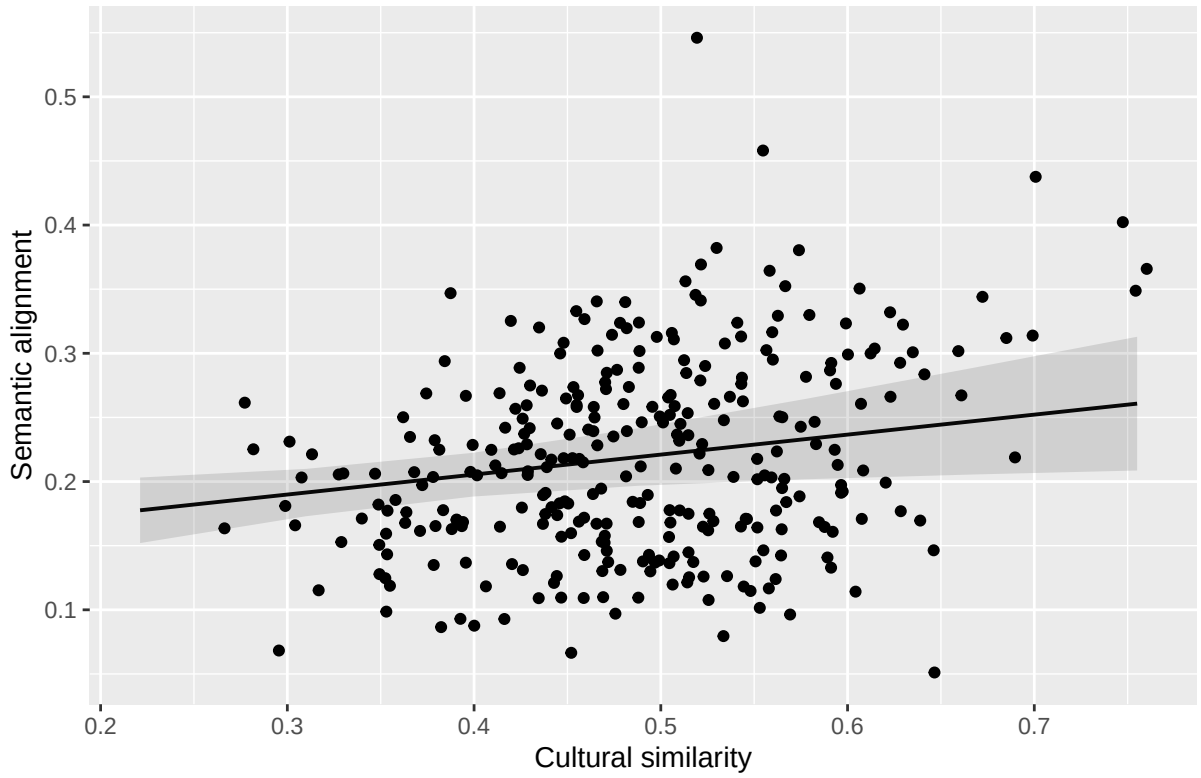
```
summary(m1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ 1 + comparison_count.center + cult.dist.center +
##     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
##     area.group)
##    Data: ling
##
## REML criterion at convergence: 784.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9661 -0.8427  0.0488  0.7422  4.3734
##
## Random effects:
##  Groups       Name             Variance Std.Dev. Corr
##  family.group (Intercept)      0.247132 0.49712
##               cult.dist.center 0.028994 0.17028  1.00
##  area.group   (Intercept)      0.059174 0.24326
##               cult.dist.center 0.006892 0.08302  1.00
##  Residual                      0.643045 0.80190
## Number of obs: 308, groups:  family.group, 23; area.group, 19
##
## Fixed effects:
##                         Estimate Std. Error t value
## (Intercept)              0.01866    0.15042   0.124
## comparison_count.center  0.27254    0.05689   4.791
## cult.dist.center         0.18217    0.07684   2.371
##
## Correlation of Fixed Effects:
##             (Intr) cmpr_.
## cmprsn_cnt.  0.098
## clt.dst.cnt  0.669 -0.030
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Plot the estimates, rescaling the variables back to the original units:

```
trans = function(X){
  X * attr(ling$rho.center,"scaled:scale") +
  attr(ling$rho.center,"scaled:center")
}

gx = plot_model(m1,'pred',terms='cult.dist.center')
gx$data$predicted = trans(gx$data$predicted)
gx$data$conf.low = trans(gx$data$conf.low)
gx$data$conf.high = trans(gx$data$conf.high)
gx$data$x = gx$data$x *
  cdc.s +cdc.c
gx = gx + #coord_cartesian(ylim=c(0,0.5),
          #                xlim=c(0.15,0.85)) +
  xlab("Cultural similarity") +
```

```
  ylab("Semantic alignment") +
  ggtitle("") +
  geom_point(data=ling,aes(x=cult.dist,y=local_alignment))
gx
```



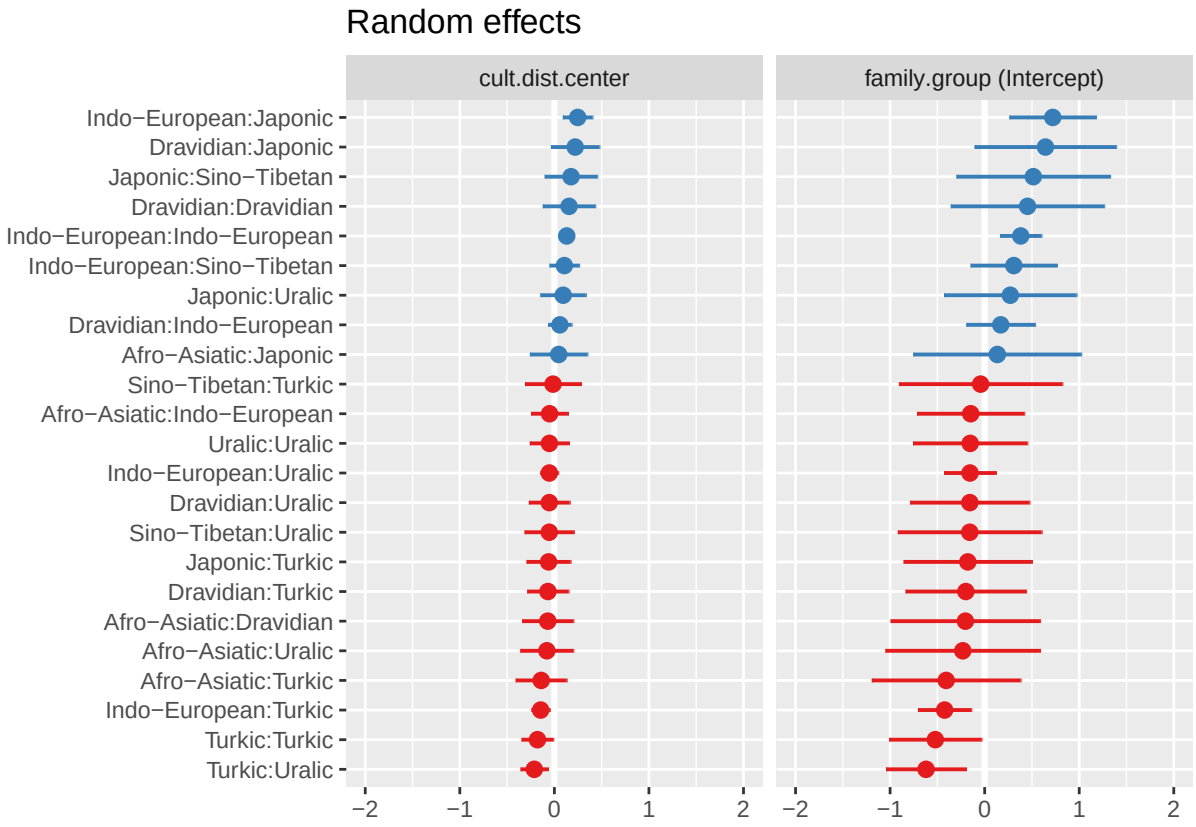

```
pdf(paste0("../results/stats/",datasetName,"/CulturalDistance_Rho_Graph.pdf"),
    height=2.5, width=2.5)
gx
dev.off()
```

```
## pdf
##   2
```

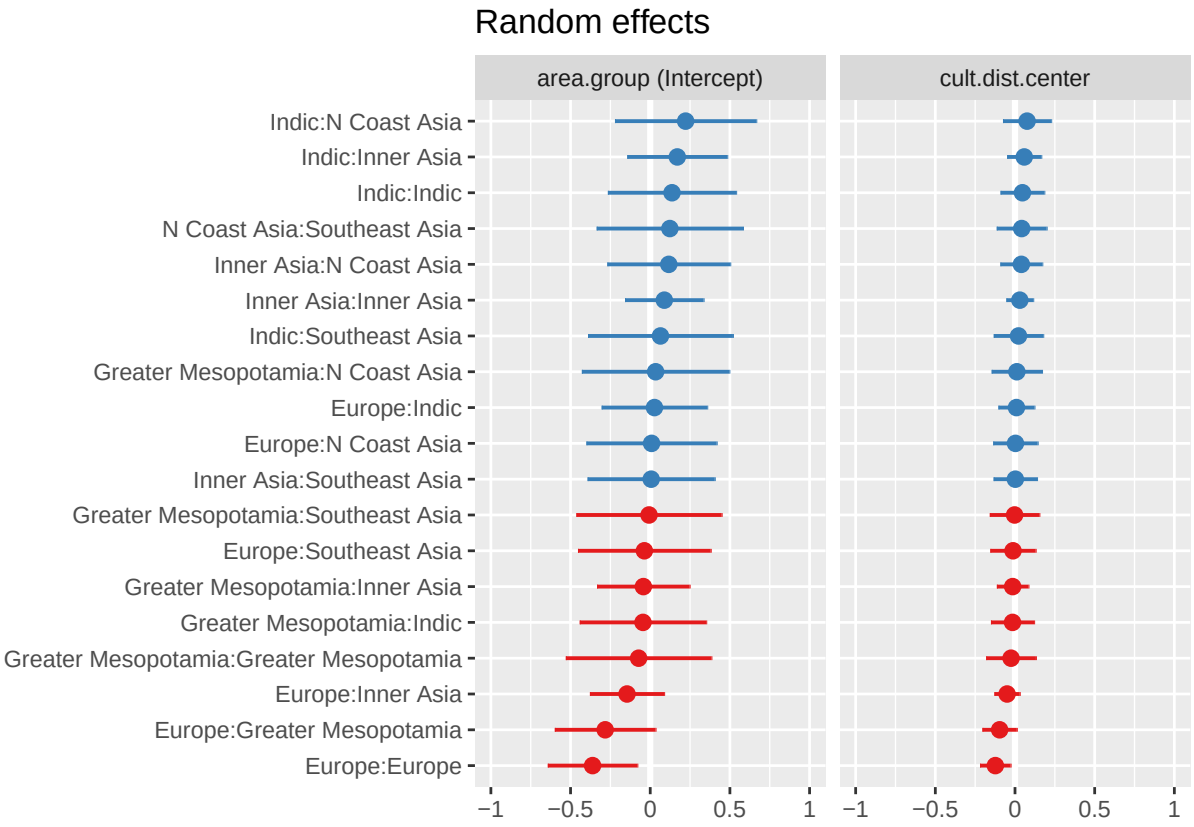
Plot the random effects:

```
plot_model(m1,'re', sort.est = "cult.dist.center")
```

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


Random effects
cult.dist.center
family.group (Intercept)
Indo−European:Japonic
Dravidian:Japonic
Japonic:Sino−Tibetan
Dravidian:Dravidian
Indo−European:Indo−European
Indo−European:Sino−Tibetan
Japonic:Uralic
Dravidian:Indo−European
Afro−Asiatic:Japonic
Sino−Tibetan:Turkic
Afro−Asiatic:Indo−European
Uralic:Uralic
Indo−European:Uralic
Dravidian:Uralic
Sino−Tibetan:Uralic
Japonic:Turkic
Dravidian:Turkic
Afro−Asiatic:Dravidian
Afro−Asiatic:Uralic
Afro−Asiatic:Turkic
Indo−European:Turkic
Turkic:Turkic
Turkic:Uralic
−2
−1
0
1
2
−2
−1
0
1
2

```
##
## [[2]]
```



MRM

Use multiple regression on distance matrices to do the same test as above. The code below loads the data into a matrix format:

```
# Use graph method to make distance matrix
grph <- graph.data.frame(ling[,c("l1",'l2','local_alignment')], directed=FALSE)
# add value as a weight attribute
ling.m = get.adjacency(grph, attr="local_alignment", sparse=FALSE)
rownames(ling.m) = l[match(rownames(ling.m),l$iso2),]$Language2
colnames(ling.m) = l[match(colnames(ling.m),l$iso2),]$Language2
# Same for comparison_count.center
grph <- graph.data.frame(ling[,c("l1",'l2','comparison_count')], directed=FALSE)
# add value as a weight attribute
cc.m = get.adjacency(grph, attr="comparison_count", sparse=FALSE)
rownames(cc.m) = l[match(rownames(cc.m),l$iso2),]$Language2
colnames(cc.m) = l[match(colnames(cc.m),l$iso2),]$Language2

cult.m = read.csv("../results/EA_distances/CulturalDistances.csv", stringsAsFactors = F)
rownames(cult.m) = cult.m[,1]
cult.m = cult.m[,2:ncol(cult.m)]
cult.m = as.matrix(cult.m)
# Flip cultural value to distance
cult.m = 1-cult.m
mx = match(rownames(ling.m),rownames(cult.m))
cult.m = cult.m[mx,mx]
```

```
colnames(cult.m) = rownames(cult.m)

# Same/different matrix for language family
family.matrix = l[match(rownames(ling.m),l$Language),]$family
family.matrix = outer(family.matrix,family.matrix,"!=") *1

# Load ASJP distances for second test
asjp = readRDS("../data/ASJP/asjp17-dists_FAIR.RData")
ling.m.glotto = l[match(rownames(cult.m),l$Language2),]$glotto
ling.m.glotto = ling.m.glotto[ling.m.glotto %in% rownames(asjp)]
asjp.m = asjp[ling.m.glotto,ling.m.glotto]
asjp.lang.names = l[match(rownames(asjp.m),l$glotto),]$Language2
# Matrices for second analysis with asjp
ling.m2 = ling.m[asjp.lang.names,asjp.lang.names]
cult.m2 = cult.m[asjp.lang.names,asjp.lang.names]
cc.m2 =  cc.m[asjp.lang.names,asjp.lang.names]

# Load the geographic distances:
geoDist = read.csv("../data/GeographicDistances.csv",stringsAsFactors = F)
geoDist.m = as.matrix(geoDist)
geoDist.m = geoDist.m[!is.na(geoDist.m[,1]),!is.na(geoDist.m[1,])]
# Convert to log distance in thousand km
geoDist.m = log10(geoDist.m/1000)
geoDist.m[is.infinite(geoDist.m)] = 0
colnames(geoDist.m) = gsub("\\."," ",colnames(geoDist.m))
rownames(geoDist.m) = colnames(geoDist.m)
geoDist.m1 = geoDist.m[rownames(ling.m),rownames(ling.m)]
geoDist.m2 = geoDist.m[rownames(ling.m2),rownames(ling.m2)]

# For missing comparisons, impute the mean:
# (there are no zero values in the local alignment data)
ling.m[ling.m==0] = mean(ling$local_alignment)
diag(ling.m) = 0
ling.m2[ling.m2==0] = mean(ling.m2[ling.m2!=0])
diag(ling.m2) = 0

# center and scale values
ling.m =  matrix(scale(as.vector(ling.m)),nrow=nrow(ling.m))
cc.m =  matrix(scale(as.vector(cc.m)),nrow=nrow(cc.m))
cult.m = matrix(scale(as.vector(cult.m)),nrow=nrow(cult.m))
geoDist.m1 = matrix(scale(as.vector(geoDist.m1)),nrow=nrow(geoDist.m1))

asjp.m = matrix(scale(as.vector(asjp.m)),nrow=nrow(asjp.m))
ling.m2 = matrix(scale(as.vector(ling.m2)),nrow=nrow(ling.m2))
cc.m2 =  matrix(scale(as.vector(cc.m2)),nrow=nrow(cc.m2))
cult.m2 = matrix(scale(as.vector(cult.m2)),nrow=nrow(cult.m2))
geoDist.m2 = matrix(scale(as.vector(geoDist.m2)),nrow=nrow(geoDist.m2))
```

Run the MRM model, predicting semantic alignment by cultural distance, controlling for family distance, geographic ditance, and the comparison count (number of observations). Here, the family distance between two languages is just whether they are part of the same family. Note that this does not take into account particular values for particular families, nor the random slopes within families.

```
set.seed(289)
MRM.fam = ecodist::MRM(as.dist(ling.m) ~
                as.dist(cult.m) +
                as.dist(family.matrix) +
                as.dist(geoDist.m1)  +
                as.dist(cc.m),nperm = 10000)

MRM.asjp = ecodist::MRM(as.dist(ling.m2) ~
                as.dist(cult.m2) +
                as.dist(asjp.m) +
                as.dist(geoDist.m2) +
                as.dist(cc.m2),nperm = 10000)

rownames(MRM.fam$coef) = c("Intercept","Cultural distance","Language family",
                           "Geographic distance","Comparison count")
colnames(MRM.fam$coef) = c("Estimate","p-value")
statMRM.fam = xtable(MRM.fam$coef, digits = 3,display=c("s",'f',"fg"),
                     caption = paste0(
        "MRM analysis predicting semantic alignment (",
        datasetLabel,"), with family control. $R^2$=",
        signif(MRM.fam$r.squared[1],3)))
print(statMRM.fam,"latex",
    file="../results/stats/tex/MRM_family_CC.tex")

rownames(MRM.asjp$coef) = c("Intercept","Cultural distance","ASJP",
                           "Geographic distance","Comparison count")
colnames(MRM.asjp$coef) = c("Estimate","p-value")
statMRM.fam = xtable(MRM.asjp$coef, digits = 3,display=c("s",'f',"fg"),
                     caption = paste0(
        "MRM analysis predicting semantic alignment (",
        datasetLabel,"), with ASJP control. $R^2$=",
        signif(MRM.asjp$r.squared[1],3)))
print(statMRM.fam,"latex",
    file="../results/stats/tex/MRM_ASJP_CC.tex")
```

Mantel tests

Read the historical distances for Indo-European, based on the phylogenetic distances.

Data prep

The geographic distances are loaded above (from "../data/GeographicDistances.csv").

Load historical distances:

```
hist = read.csv("../data/trees/IndoEuropean_historical_distances.csv", stringsAsFactors = F)
hist = hist[!duplicated(hist[,1]),!duplicated(hist[,1])]
rownames(hist) = hist[,1]
```

```
hist = hist[,2:ncol(hist)]
hist.m = as.matrix(hist)
colnames(hist.m) = rownames(hist.m)
hist.m = hist.m/max(hist.m)
```

Read the cultural distance as a matrix:

```
cult.m = read.csv("../results/EA_distances/CulturalDistances.csv", stringsAsFactors = F)
rownames(cult.m) = cult.m[,1]
cult.m = cult.m[,2:ncol(cult.m)]
```

Flip the cultural distance into a cultural similarity measure:

```
cult.m = 1-cult.m
```

Convert the linguistic similarities to a matrix. This uses `igraph` to make an undirected graph from the long format with `local_alignment` as the edge weights, then output a matrix of adjacencies.

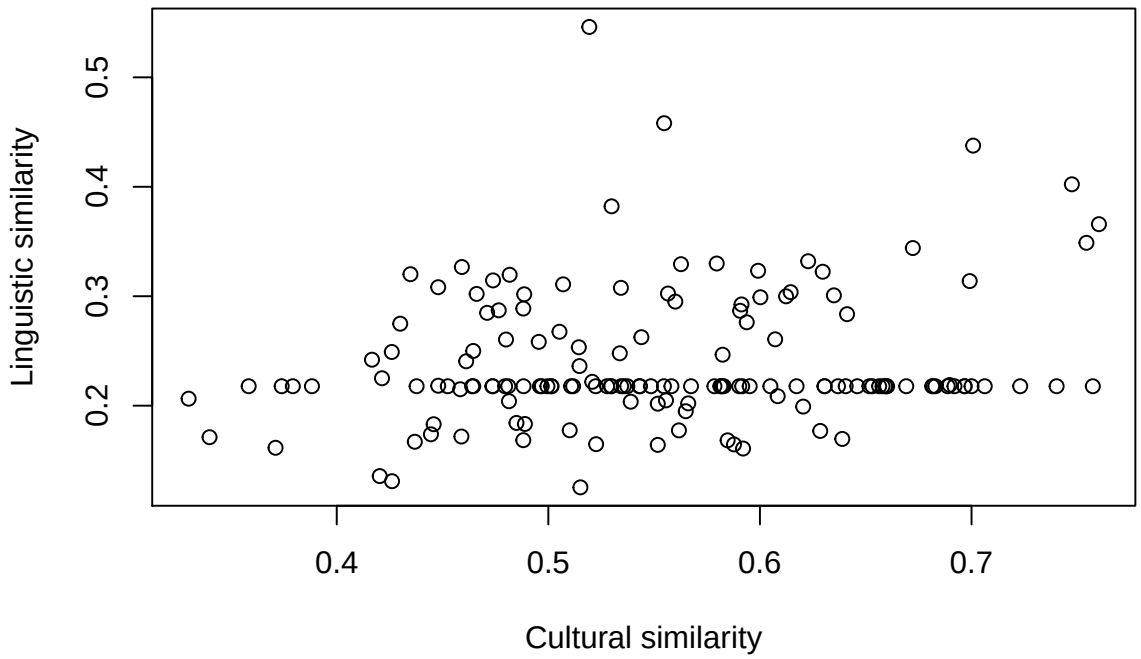
```
grph <- graph.data.frame(ling[,c("l1",'l2','local_alignment')], directed=FALSE)
# add value as a weight attribute
ling.m = get.adjacency(grph, attr="local_alignment", sparse=FALSE)
rownames(ling.m) = l[match(rownames(ling.m),l$iso2),]$Language2
colnames(ling.m) = l[match(colnames(ling.m),l$iso2),]$Language2
# For missing comparisons, impute the mean:
# (there are no zero values in the local alignment data)
ling.m[ling.m==0] = mean(ling$local_alignment)
diag(ling.m) = 0
```

Match the distance matrices

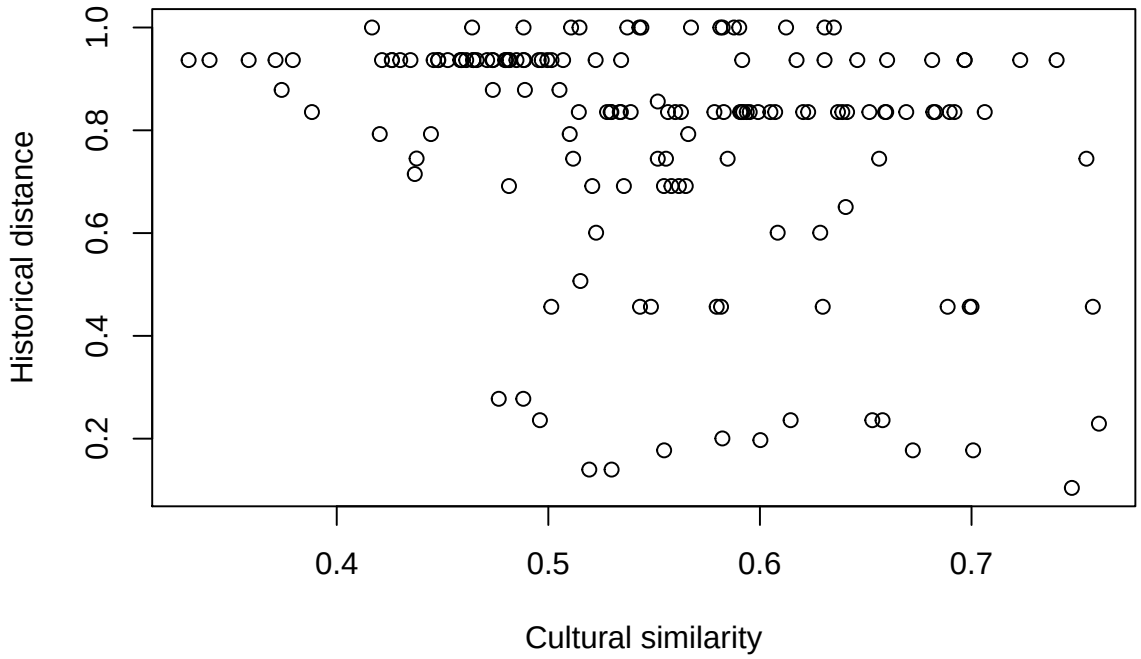
```
in.analysis = intersect(rownames(ling.m),rownames(cult.m))
in.analysis = intersect(in.analysis, rownames(hist.m))
cult.m2 = cult.m[in.analysis,in.analysis]
ling.m2 = ling.m[in.analysis,in.analysis]
hist.m2 = hist.m[in.analysis,in.analysis]
geo.m2 = geoDist.m[in.analysis,in.analysis]
```

Note that there are only 18 languages with data on lingusitic, cultural and historical distance. This is becasue the historical distances are derived from a tree of Indo-European languages (there are currently no reliable phylogentic trees constructed from cognates that span different language families). The languages in this test include: Albanian, Armenian, Belarusian, Bengali, Bulgarian, Czech, Dutch, English, French, Greek, Icelandic, Irish, Latin, Latvian, Lithuanian, Ossetian, Russian, Ukrainian.

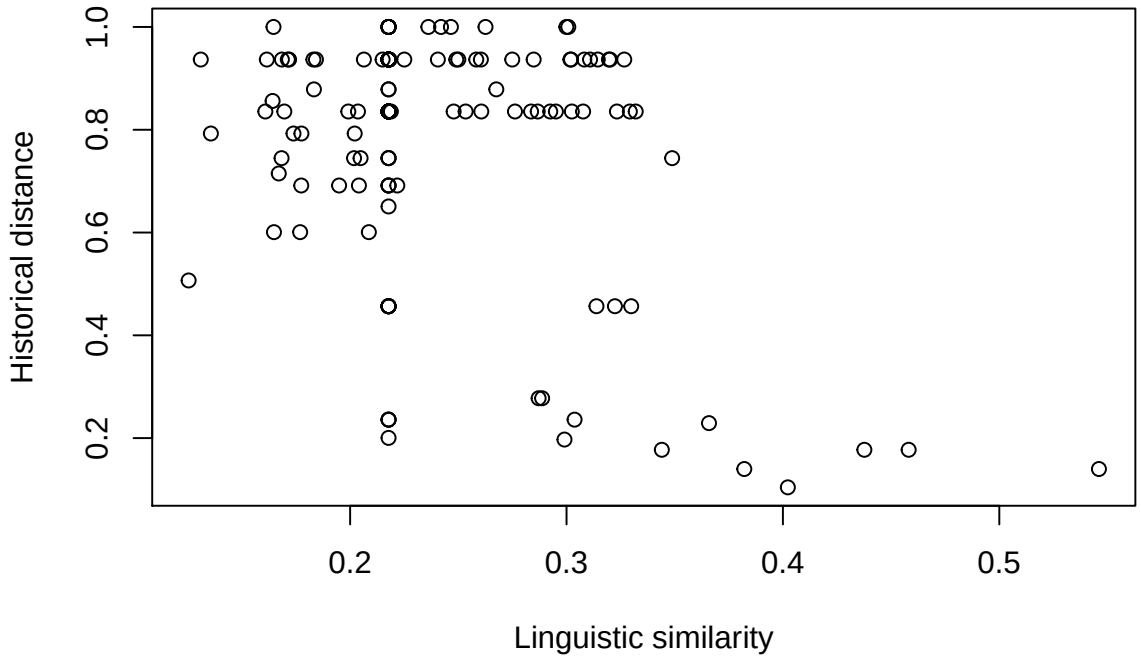
```
plot(as.dist(cult.m2),as.dist(ling.m2),
     xlab="Cultural similarity",
     ylab="Linguistic similarity")
```

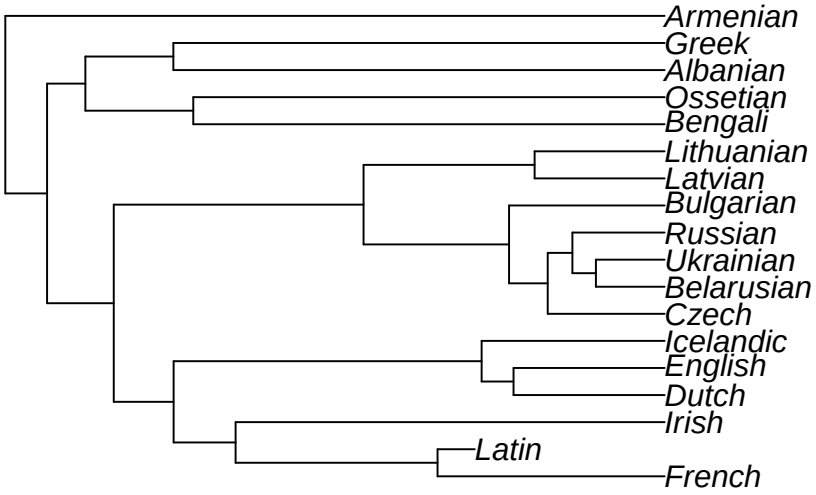



```
plot(as.dist(cult.m2),as.dist(hist.m2),
     xlab="Cultural similarity",
     ylab="Historical distance")
```




```
plot(as.dist(ling.m2),as.dist(hist.m2),
     xlab="Linguistic similarity",
     ylab="Historical distance")
```


Historical distance
1.0
0.8
0.6
0.4
0.2
0.2
0.3
0.4
0.5
Linguistic similarity


Armenian
Greek
Albanian
Ossetian
Bengali
Lithuanian
Latvian
Bulgarian
Russian
Ukrainian
Belarusian
Czech
Icelandic
English
Dutch
Irish
Latin
French

Tests

The results of the test list the following measures:

- mantelr: Mantel correlation coefficient.
- pval1: one-tailed p-value (null hypothesis: r <= 0).
- pval2: one-tailed p-value (null hypothesis: r >= 0).
- pval3: two-tailed p-value (null hypothesis: r = 0).
- llim: lower confidence limit for r.
- ulim: upper confidence limit for r.

```
set.seed(1498)
```

Run tests between each pair of measures.

```
distms = list("Cultrual"= cult.m2,
              "Linguistic" = ling.m2,
              "Historical" = hist.m2,
              "Geographic" = geo.m2)

mantelRes1 = data.frame(
  Var1 = NA, Var2 = NA, r = NA,
  llim = NA, ulim = NA, p = NA,
  stringsAsFactors = F)

for(i in 1:3){
  for(j in (i+1):4){
    var1 = names(distms)[i]
    var2 = names(distms)[j]
    print(paste("Correlation between",
                var1,"and",var2))
    stat = ecodist::mantel(as.dist(distms[[i]]) ~
                as.dist(distms[[j]]),
                nperm = 100000)
    print(stat)
    mantelRes1 = rbind(mantelRes1,
        c(var1,var2,stat[1],stat[5],stat[6],
          min(c(stat[2],stat[3]))))
    stat = round(stat,2)
    stat2 = sprintf("$r$ = %s[%s,%s], one-tailed $p$ = %s",
      stat[1],
      stat[5],
      stat[6],
      min(c(stat[2],stat[3])))
    # TODO: output stats
    #cat(stat2,file=
     #     paste0("../results/stats/tex/Mantel",var1,"Vs",var2,"Distance_CC.tex"))
  }
}
```

```
## [1] "Correlation between Cultrual and Linguistic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## 0.18028102 0.11950000 0.88051000 0.22785000 0.01309925 0.28568841
## [1] "Correlation between Cultrual and Historical"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.3148429  0.9789500  0.0210600  0.0240700 -0.4468802 -0.2035161
```

```
## [1] "Correlation between Cultrual and Geographic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.4608256  0.9970600  0.0029500  0.0029500 -0.5860424 -0.3118663
## [1] "Correlation between Linguistic and Historical"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.4040486  0.9989700  0.0010400  0.0010400 -0.5225047 -0.1936433
## [1] "Correlation between Linguistic and Geographic"
##      mantelr        pval1        pval2        pval3    llim.2.5%
## -0.083729598  0.738040000  0.261970000  0.545660000 -0.202460986
##   ulim.97.5%
## -0.005520394
## [1] "Correlation between Historical and Geographic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
##  0.4052690  0.0010100  0.9990000  0.0010100  0.3098220  0.5192832
```

```
mantelRes1= mantelRes1[2:nrow(mantelRes1),]
mantelRes1[,3:6] = apply(mantelRes1[,3:6],2,function(X){
  signif(as.numeric(X),3)
})
```

Run a mantel test comparing the Linguistic alignment to the cultural similarity, controlling for the historical distance between languages:

```
ecodist::mantel(as.dist(ling.m2)~
                  as.dist(cult.m2) +
                  as.dist(hist.m2),
                nperm = 100000)
```

```
##     mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
##  0.06112430  0.32931000  0.67070000  0.68457000 -0.09050103  0.18048090
```

Main Test: Run a mantel test comparing the Linguistic alignment to the cultural similarity, controlling for the historical distance and geographic distance between languages:

```
mainMantel = ecodist::mantel(as.dist(ling.m2)~
                  as.dist(cult.m2) +
                  as.dist(hist.m2) +
                  as.dist(geo.m2),
                nperm = 100000)
mainMantel = signif(mainMantel,3)

mantelRes1 = rbind(mantelRes1,
  c("Linguistic","Cultural **",
    mainMantel[1],mainMantel[5],mainMantel[6],
    min(mainMantel[2:3])))

mantelRes1Text = xtable(mantelRes1,
   caption =  paste0(
    "Mantel tests (",
    datasetLabel,
    "). ** = partial Mantel test, controlling for historical and geographical distance."))
print(mantelRes1Text,
    file="../results/stats/tex/Mantel_CC.tex")
```

MRM

Perform the main test, but using multiple regression on distance matrices (MRM).

Lichstein, J. W. (2007). Multiple regression on distance matrices: a multivariate spatial analysis tool. Plant Ecology, 188(2), 117-131.

```
mainMRM = ecodist::MRM(as.dist(ling.m2)~
                  as.dist(cult.m2) +
                  as.dist(hist.m2) +
                  as.dist(geo.m2), nperm=10000)
mainMRM
```

```
## $coef
##                   as.dist(ling.m2)   pval
## Int                     0.27872648 0.1786
## as.dist(cult.m2)        0.07221745 0.4538
## as.dist(hist.m2)       -0.11036698 0.0016
## as.dist(geo.m2)         0.02518882 0.3089
##
## $r.squared
##        R2      pval
## 0.1803183 0.0081000
##
## $F.test
##        F   F.pval
## 10.92596  0.00810
```

```
mainMRM2 = sprintf("$\\beta= $%s, $p=$%s",
                   round(mainMRM$coef[2,1],2),
                   round(mainMRM$coef[2,2],2))
cat(mainMRM2,
    file="../results/stats/tex/MRMCultrualVsLinguisticDistance_Partial_CC.tex")
```

4.5 Cross-cultural analysis (Subtitles data)

Predicting semantic alignment by cultural similarity: Subtitles data

Bill Thompson, Seán Roberts & Gary Lupyan

Contents

Introduction

This file replicates the tests for the main wikipedia data on the subtitles data.

Load libraries

```
library(ape)
library(ecodist)
library(lme4)
library(sjPlot)
library(ggplot2)
library(igraph)
library(lattice)
library(xtable)
```

Parameters (using data from Northuralex and common crawl, k=100, unfiltered):

```
datasetName = "subs"
datasetLabel = "Subtitles"
lingDistancesFile = "../data/FAIR/nel-k100-subs-alignments-by-language-pair.csv"
lingDistancesByDomainFile = "../results/EA_distances/nel-k100-subs_with_ling.csv"
# (generated by ../processing/combineCultAndLingDistances.R)
```

All domains

Load data

Read the cultural distances:

```
cult = read.csv("../results/EA_distances/CulturalDistances_Long.csv", stringsAsFactors = F)
names(cult) = c("l1","l2","cult.dist")
```

Add language family:

```
l = read.csv("../data/FAIR_langauges_glotto_xdid.csv", stringsAsFactors = F)
g = read.csv("../data/glottolog-languoid.csv/languoid.csv", stringsAsFactors = F)
l$family = g[match(l$glotto,g$id),]$family_pk
l$family = g[match(l$family,g$pk),]$name
```

Read the semantic distances

```
ling = read.csv(lingDistancesFile, stringsAsFactors = F)
```

There are very few possible comparisons for Slovenian and Northern Sami, so we'll remove these:

```
ling = ling[!(ling$l1=="se" | ling$l2 == "se"),]
ling = ling[!(ling$l1=="sl" | ling$l2 == "sl"),]
```

Combine the lingusitic and cultural distances. Note that we flip the cultural measure from a distance measure to a similarity measure.

```
cult$l1.iso2 = l[match(cult$l1,l$Language2),]$iso2
cult$l2.iso2 = l[match(cult$l2,l$Language2),]$iso2

fairisos = unique(c(ling$l1,ling$l2))
cultisos = unique(c(cult$l1.iso2, cult$l2.iso2))

cult = cult[(cult$l1.iso2 %in% fairisos) & (cult$l2.iso2 %in% fairisos),]
ling = ling[(ling$l1 %in% cultisos) & (ling$l2 %in% cultisos),]

matches = sapply(1:nrow(ling), function(i){
  which(cult$l1.iso2==ling$l1[i] & cult$l2.iso2==ling$l2[i])
})

ling$cult.dist = cult[matches,]$cult.dist
# Flip
ling$cult.dist = 1 - ling$cult.dist
# Scale
ling$cult.dist.center = scale(ling$cult.dist)
cdc.s = attr(ling$cult.dist.center,"scaled:scale")
cdc.c = attr(ling$cult.dist.center,"scaled:center")
ling$cult.dist.center = as.numeric(ling$cult.dist.center)
ling$comparison_count.center =
  scale(ling$comparison_count)

ling$family1 = l[match(ling$l1, l$iso2),]$family
ling$family2 = l[match(ling$l2, l$iso2),]$family
l[l$Language=="Arabic",]$autotyp.area= "Greater Mesopotamia"
l[l$Language=="Persian",]$autotyp.area= "Greater Mesopotamia"
ling$area1 = l[match(ling$l1, l$iso2),]$autotyp.area
```

```
ling$area2 = l[match(ling$l2, l$iso2),]$autotyp.area


fgroup = cbind(ling$family1,ling$family2)
fgroup = apply(fgroup,1,sort)
ling$family.group = apply(fgroup,2,paste,collapse=":")
agroup = cbind(ling$area1,ling$area2)
agroup = apply(agroup,1,sort)
ling$area.group = apply(agroup,2,paste,collapse=":")

ling$rho.center = scale(ling$local_alignment)
```

Each observation is now assocaited with a language family pair:

```
head(ling[,c("l1","l2","local_alignment",'family.group')])
```

```
##    l1 l2 local_alignment                family.group
## 18 hy sq      0.01014219 Indo-European:Indo-European
## 23 hy ko      0.01862729      Indo-European:Koreanic
## 32 hy is      0.03000878 Indo-European:Indo-European
## 48 hy ja      0.05326118       Indo-European:Japonic
## 49 et hy      0.05535150        Indo-European:Uralic
## 52 hy nl      0.05835286 Indo-European:Indo-European
```

And the same is true for area:

```
tail(ling[,c("l1","l2","local_alignment",'area.group')])
```

```
##     l1 l2 local_alignment        area.group
## 651 cs es       0.3339303     Europe:Europe
## 653 bg cs       0.3353707     Europe:Europe
## 656 el bg       0.3378855     Europe:Europe
## 659 el ru       0.3446584 Europe:Inner Asia
## 661 el cs       0.3625137     Europe:Europe
## 664 cs ru       0.3790251 Europe:Inner Asia
```

Number of observations:

```
# Number of datapoints:
nrow(ling)
```

```
## [1] 190
```

```
# Number of unique languages:
length(unique(unlist(ling[,c("l1","l2")])))
```

```
## [1] 20
```

```
# Number of unique langauge families:
uniqueFamilies = unique(unlist(ling[,c("family1","family2")]))
length(uniqueFamilies)
```

```
## [1] 6
```

```
# Number of unique areas:
uniqueAreas = unique(unlist(ling[,c("area1","area2")]))
length(uniqueAreas)
```

```
## [1] 4
```

Cross-over between language famlies and areas:

```
tx = data.frame(lang= c(ling$l1,ling$l2),
           fam = c(ling$family1,ling$family2),
           area= c(ling$area1,ling$area2))
tx = tx[!duplicated(tx),]
table(tx$fam,tx$area)
```

```
##
##                 Europe Greater Mesopotamia Inner Asia N Coast Asia
##   Afro-Asiatic       0                   1          0            0
##   Indo-European      9                   1          4            0
##   Japonic            0                   0          0            1
##   Koreanic           0                   0          0            1
##   Turkic             0                   1          0            0
##   Uralic             1                   0          1            0
```

LMER models

Mixed effects model, predicting Linguistic similaritys from cultural similarity, with random intercept for family and area and random slope for cultural similarity for family and area.

We start with a null model with random intercepts for family and area, and random slopes for cultural similarity by both. We add a fixed effect of the number of comparisons made for each datapoint (number of concepts that were available to compare). Then we add a fixed effect of cultural similarity

```
m0 = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
m0.5 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0073357
## (tol = 0.002, component 1)
```

```
m1 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    cult.dist.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
an1 = anova(m0,m0.5,m1)
```

```
## refitting model(s) with ML (instead of REML)
```

```
an1
```

```
## Data: ling
## Models:
## m0: rho.center ~ 1 + (1 + cult.dist.center | family.group) + (1 +
## m0:     cult.dist.center | area.group)
## m0.5: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## m0.5:     family.group) + (1 + cult.dist.center | area.group)
## m1: rho.center ~ 1 + comparison_count.center + cult.dist.center +
## m1:     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
## m1:     area.group)
##      Df    AIC    BIC  logLik deviance    Chisq Chi Df Pr(>Chisq)
## m0    8 409.01 434.99 -196.51   393.01
## m0.5  9 288.62 317.84 -135.31   270.62 122.3948      1     <2e-16 ***
```

```
## m1   10 289.88 322.35 -134.94   269.88   0.7417      1     0.3891
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cultural similarity is significantly correlated with Linguistic similarity. Here are the model estimates:

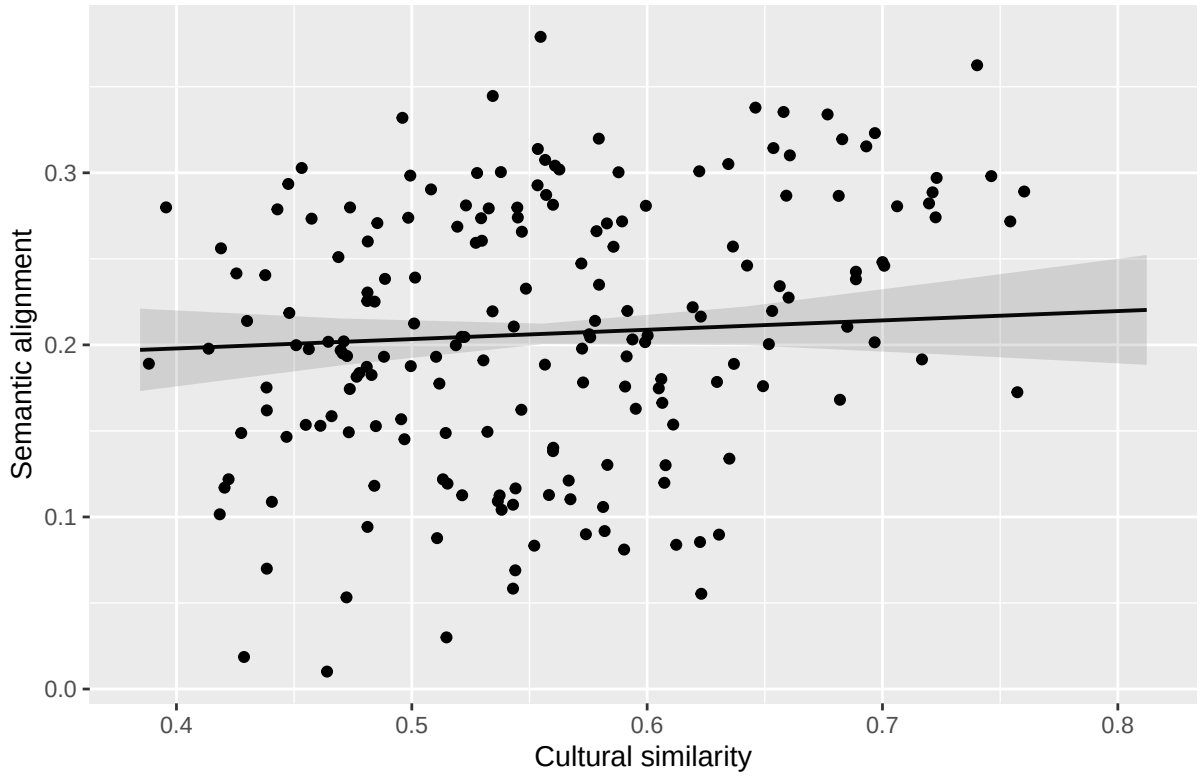
```
summary(m1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ 1 + comparison_count.center + cult.dist.center +
##     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
##     area.group)
##    Data: ling
##
## REML criterion at convergence: 282.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8968 -0.7471 -0.0734  0.6181  3.6760
##
## Random effects:
##  Groups       Name             Variance  Std.Dev. Corr
##  family.group (Intercept)      0.0002465 0.01570
##               cult.dist.center 0.0056335 0.07506  -1.00
##  area.group   (Intercept)      0.0015278 0.03909
##               cult.dist.center 0.0211421 0.14540  -1.00
##  Residual                      0.2322780 0.48195
## Number of obs: 190, groups:  family.group, 17; area.group, 10
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)              -0.01139    0.03975  -0.286
## comparison_count.center   0.81265    0.03807  21.346
## cult.dist.center          0.06060    0.07272   0.833
##
## Correlation of Fixed Effects:
##             (Intr) cmpr_.
## cmprsn_cnt.  0.013
## clt.dst.cnt -0.231 -0.136
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Plot the estimates, rescaling the variables back to the original units:

```
trans = function(X){
  X * attr(ling$rho.center,"scaled:scale") +
  attr(ling$rho.center,"scaled:center")
}

gx = plot_model(m1,'pred',terms='cult.dist.center')
gx$data$predicted = trans(gx$data$predicted)
gx$data$conf.low = trans(gx$data$conf.low)
gx$data$conf.high = trans(gx$data$conf.high)
gx$data$x = gx$data$x *
  cdc.s +cdc.c
gx = gx + #coord_cartesian(ylim=c(0,0.5),
```

```
          #                 xlim=c(0.15,0.85)) +
  xlab("Cultural similarity") +
  ylab("Semantic alignment") +
  ggtitle("") +
  geom_point(data=ling,aes(x=cult.dist,y=local_alignment))
gx
```



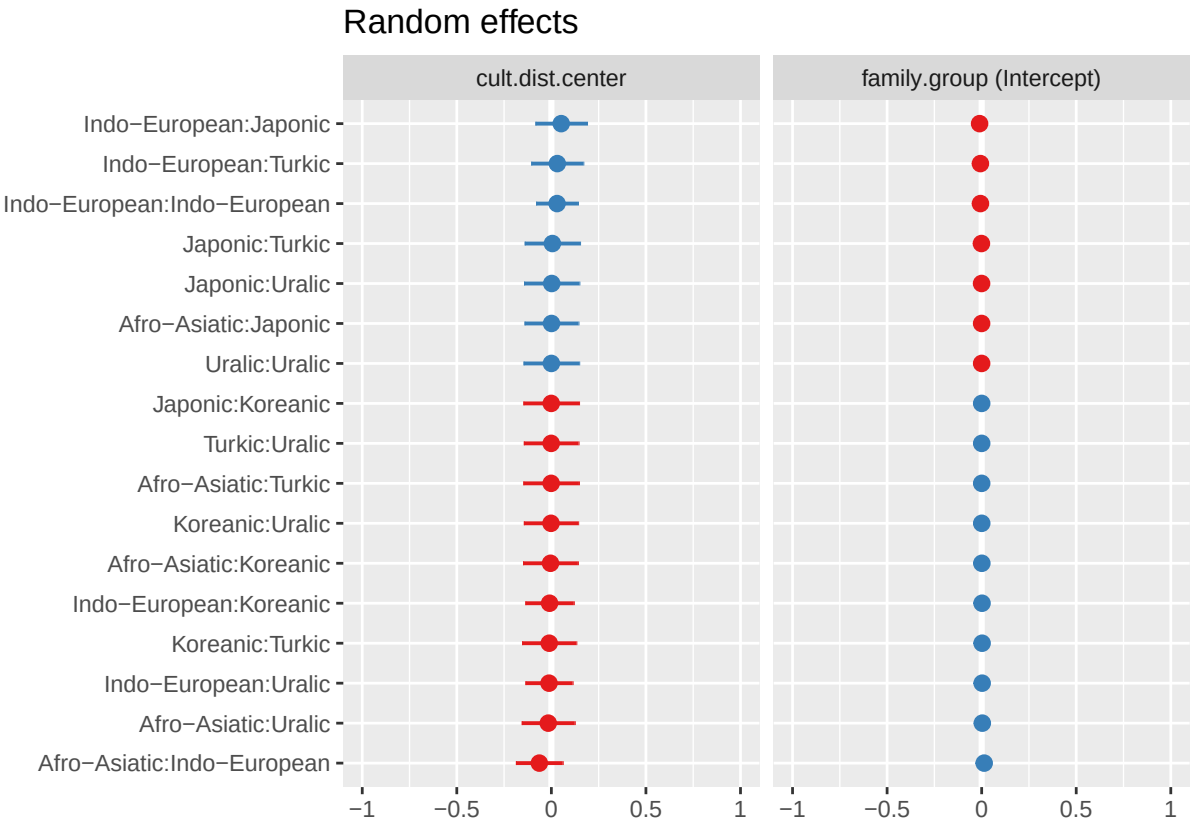

```
pdf(paste0("../results/stats/",datasetName,"/CulturalDistance_Rho_Graph.pdf"),
    height=2.5, width=2.5)
gx
dev.off()
```

```
## pdf
##   2
```

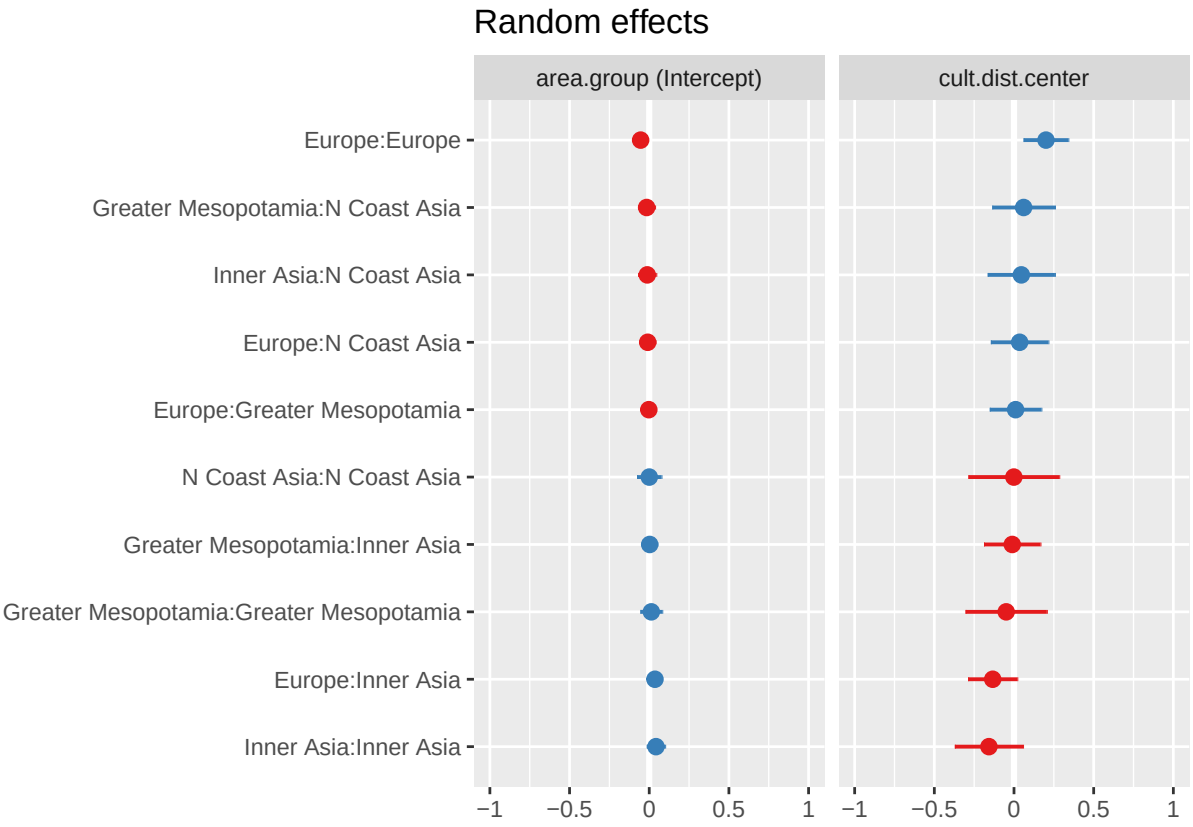
Plot the random effects:

```
plot_model(m1,'re', sort.est = "cult.dist.center")
```

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


Random effects
cult.dist.center
family.group (Intercept)
Indo−European:Japonic
Indo−European:Turkic
Indo−European:Indo−European
Japonic:Turkic
Japonic:Uralic
Afro−Asiatic:Japonic
Uralic:Uralic
Japonic:Koreanic
Turkic:Uralic
Afro−Asiatic:Turkic
Koreanic:Uralic
Afro−Asiatic:Koreanic
Indo−European:Koreanic
Koreanic:Turkic
Indo−European:Uralic
Afro−Asiatic:Uralic
Afro−Asiatic:Indo−European
−1
−0.5
0
0.5
1

```
##
## [[2]]
```



MRM

Use multiple regression on distance matrices to do the same test as above. The code below loads the data into a matrix format:

```
# Use graph method to make distance matrix
grph <- graph.data.frame(ling[,c("l1",'l2','local_alignment')], directed=FALSE)
# add value as a weight attribute
ling.m = get.adjacency(grph, attr="local_alignment", sparse=FALSE)
rownames(ling.m) = l[match(rownames(ling.m),l$iso2),]$Language2
colnames(ling.m) = l[match(colnames(ling.m),l$iso2),]$Language2
# Same for comparison_count.center
grph <- graph.data.frame(ling[,c("l1",'l2','comparison_count')], directed=FALSE)
# add value as a weight attribute
cc.m = get.adjacency(grph, attr="comparison_count", sparse=FALSE)
rownames(cc.m) = l[match(rownames(cc.m),l$iso2),]$Language2
colnames(cc.m) = l[match(colnames(cc.m),l$iso2),]$Language2

cult.m = read.csv("../results/EA_distances/CulturalDistances.csv", stringsAsFactors = F)
rownames(cult.m) = cult.m[,1]
cult.m = cult.m[,2:ncol(cult.m)]
cult.m = as.matrix(cult.m)
# Flip cultural value to distance
cult.m = 1-cult.m
mx = match(rownames(ling.m),rownames(cult.m))
cult.m = cult.m[mx,mx]
```

```r
colnames(cult.m) = rownames(cult.m)

# Same/different matrix for language family
family.matrix = l[match(rownames(ling.m),l$Language),]$family
family.matrix = outer(family.matrix,family.matrix,"!=") *1

# Load ASJP distances for second test
asjp = readRDS("../data/ASJP/asjp17-dists_FAIR.RData")
ling.m.glotto = l[match(rownames(cult.m),l$Language2),]$glotto
ling.m.glotto = ling.m.glotto[ling.m.glotto %in% rownames(asjp)]
asjp.m = asjp[ling.m.glotto,ling.m.glotto]
asjp.lang.names = l[match(rownames(asjp.m),l$glotto),]$Language2
# Matrices for second analysis with asjp
ling.m2 = ling.m[asjp.lang.names,asjp.lang.names]
cult.m2 = cult.m[asjp.lang.names,asjp.lang.names]
cc.m2 =  cc.m[asjp.lang.names,asjp.lang.names]

# Load the geographic distances:
geoDist = read.csv("../data/GeographicDistances.csv",stringsAsFactors = F)
geoDist.m = as.matrix(geoDist)
geoDist.m = geoDist.m[!is.na(geoDist.m[,1]),!is.na(geoDist.m[1,])]
# Convert to log distance in thousand km
geoDist.m = log10(geoDist.m/1000)
geoDist.m[is.infinite(geoDist.m)] = 0
colnames(geoDist.m) = gsub("\\."," ",colnames(geoDist.m))
rownames(geoDist.m) = colnames(geoDist.m)
geoDist.m1 = geoDist.m[rownames(ling.m),rownames(ling.m)]
geoDist.m2 = geoDist.m[rownames(ling.m2),rownames(ling.m2)]

# For missing comparisons, impute the mean:
ling.m[ling.m==0] = mean(ling$local_alignment)
diag(ling.m) = 0
ling.m2[ling.m2==0] = mean(ling.m2[ling.m2!=0])
diag(ling.m2) = 0

# center and scale values
ling.m =  matrix(scale(as.vector(ling.m)),nrow=nrow(ling.m))
cc.m =  matrix(scale(as.vector(cc.m)),nrow=nrow(cc.m))
cult.m = matrix(scale(as.vector(cult.m)),nrow=nrow(cult.m))
geoDist.m1 = matrix(scale(as.vector(geoDist.m1)),nrow=nrow(geoDist.m1))

asjp.m = matrix(scale(as.vector(asjp.m)),nrow=nrow(asjp.m))
ling.m2 = matrix(scale(as.vector(ling.m2)),nrow=nrow(ling.m2))
cc.m2 =  matrix(scale(as.vector(cc.m2)),nrow=nrow(cc.m2))
cult.m2 = matrix(scale(as.vector(cult.m2)),nrow=nrow(cult.m2))
geoDist.m2 = matrix(scale(as.vector(geoDist.m2)),nrow=nrow(geoDist.m2))
```

Run the MRM model, predicting semantic alignment by cultural distance, controlling for family distance, geographic ditance, and the comparison count (number of observations). Here, the family distance between two languages is just whether they are part of the same family. Note that this does not take into account particular values for particular families, nor the random slopes within families.

```
set.seed(289)
MRM.fam = ecodist::MRM(as.dist(ling.m) ~
                as.dist(cult.m) +
                as.dist(family.matrix) +
                as.dist(geoDist.m1)  +
                as.dist(cc.m),nperm = 10000)

MRM.asjp = ecodist::MRM(as.dist(ling.m2) ~
                as.dist(cult.m2) +
                as.dist(asjp.m) +
                as.dist(geoDist.m2) +
                as.dist(cc.m2),nperm = 10000)

rownames(MRM.fam$coef) = c("Intercept","Cultural distance","Language family",
                           "Geographic distance","Comparison count")
colnames(MRM.fam$coef) = c("Estimate","p-value")
statMRM.fam = xtable(MRM.fam$coef, digits = 3,display=c("s",'f',"fg"),
                     caption = paste0(
        "MRM analysis predicting semantic alignment (",
        datasetLabel,"), with family control. $R^2$=",
        signif(MRM.fam$r.squared[1],3)))
print(statMRM.fam,"latex",
    file="../results/stats/tex/MRM_family_SUBS.tex")

rownames(MRM.asjp$coef) = c("Intercept","Cultural distance","ASJP",
                           "Geographic distance","Comparison count")
colnames(MRM.asjp$coef) = c("Estimate","p-value")
statMRM.fam = xtable(MRM.asjp$coef, digits = 3,display=c("s",'f',"fg"),
                     caption = paste0(
        "MRM analysis predicting semantic alignment (",
        datasetLabel,"), with ASJP control. $R^2$=",
        signif(MRM.asjp$r.squared[1],3)))
print(statMRM.fam,"latex",
    file="../results/stats/tex/MRM_ASJP_SUBS.tex")
```

Mantel tests

Read the historical distances for Indo-European, based on the phylogenetic distances.

Data prep

The geographic distances are loaded above (from "../data/GeographicDistances.csv").

Load historical distances:

```
hist = read.csv("../data/trees/IndoEuropean_historical_distances.csv", stringsAsFactors = F)
hist = hist[!duplicated(hist[,1]),!duplicated(hist[,1])]
rownames(hist) = hist[,1]
```

```
hist = hist[,2:ncol(hist)]
hist.m = as.matrix(hist)
colnames(hist.m) = rownames(hist.m)
hist.m = hist.m/max(hist.m)
```

Read the cultural distance as a matrix:

```
cult.m = read.csv("../results/EA_distances/CulturalDistances.csv", stringsAsFactors = F)
rownames(cult.m) = cult.m[,1]
cult.m = cult.m[,2:ncol(cult.m)]
```

Flip the cultural distance into a cultural similarity measure:

```
cult.m = 1-cult.m
```

Convert the linguistic similarities to a matrix. This uses `igraph` to make an undirected graph from the long format with `local_alignment` as the edge weights, then output a matrix of adjacencies.

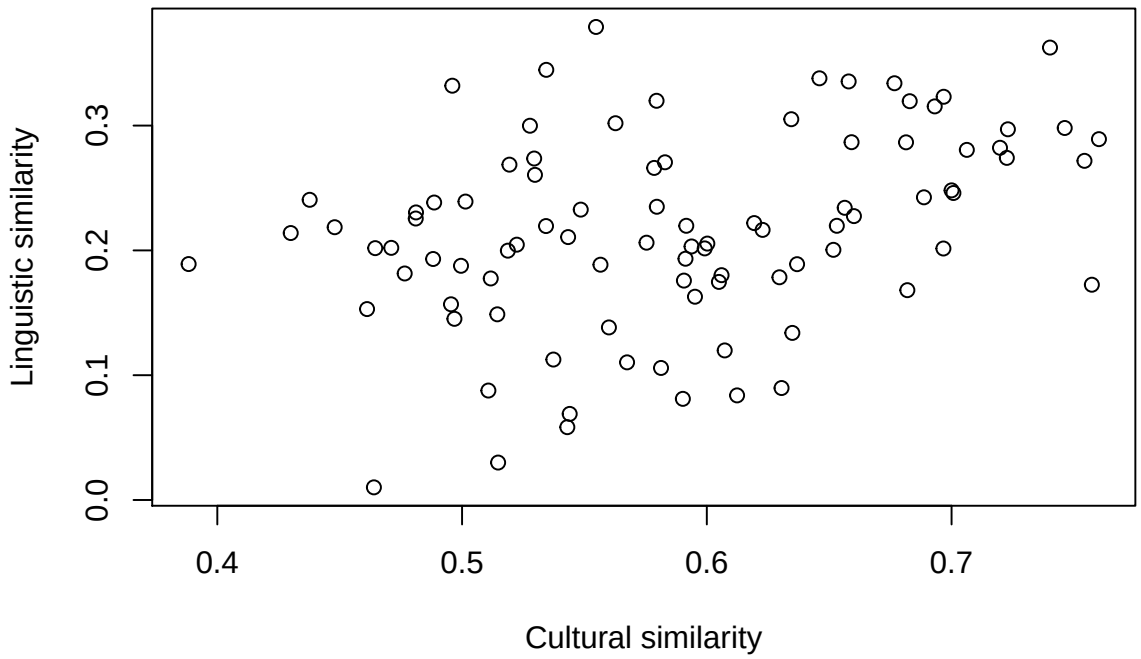
```
grph <- graph.data.frame(ling[,c("l1",'l2','local_alignment')], directed=FALSE)
# add value as a weight attribute
ling.m = get.adjacency(grph, attr="local_alignment", sparse=FALSE)
rownames(ling.m) = l[match(rownames(ling.m),l$iso2),]$Language2
colnames(ling.m) = l[match(colnames(ling.m),l$iso2),]$Language2
# For missing comparisons, impute the mean:
ling.m[ling.m==0] = mean(ling$local_alignment)
diag(ling.m) = 0
```

Match the distance matrices

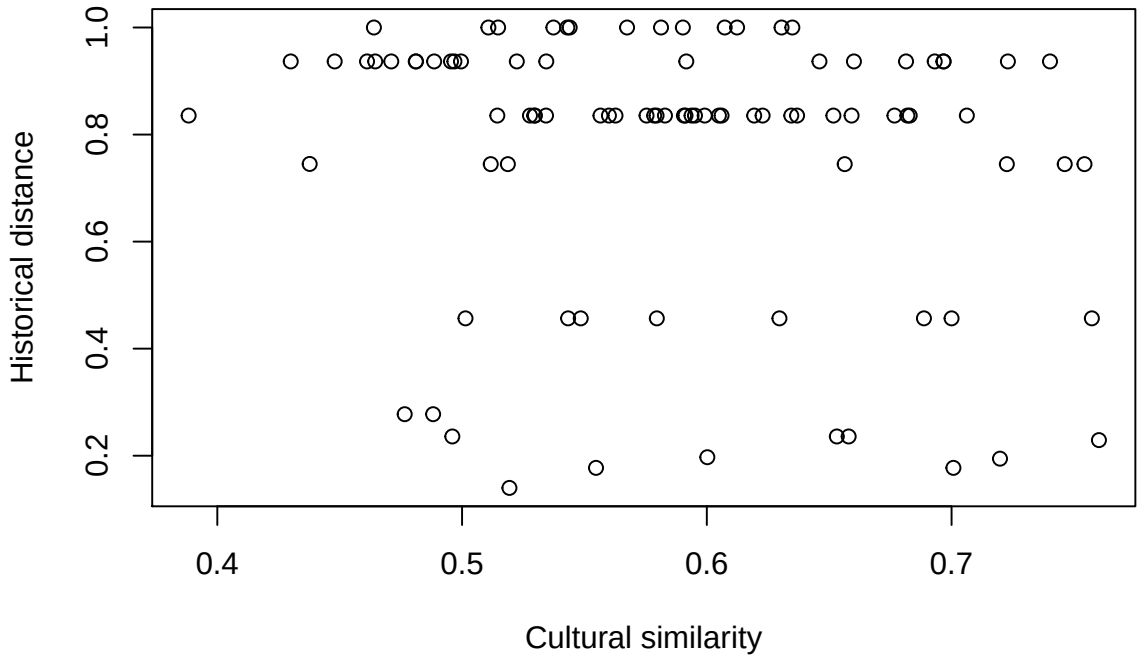
```
in.analysis = intersect(rownames(ling.m),rownames(cult.m))
in.analysis = intersect(in.analysis, rownames(hist.m))
cult.m2 = cult.m[in.analysis,in.analysis]
ling.m2 = ling.m[in.analysis,in.analysis]
hist.m2 = hist.m[in.analysis,in.analysis]
geo.m2 = geoDist.m[in.analysis,in.analysis]
```

Note that there are only 14 languages with data on lingusitic, cultural and historical distance. This is becasue the historical distances are derived from a tree of Indo-European languages (there are currently no reliable phylogentic trees constructed from cognates that span different language families). The languages in this test include: Albanian, Armenian, Bulgarian, Czech, Dutch, English, French, Greek, Icelandic, Latvian, Lithuanian, Russian, Spanish, Ukrainian.

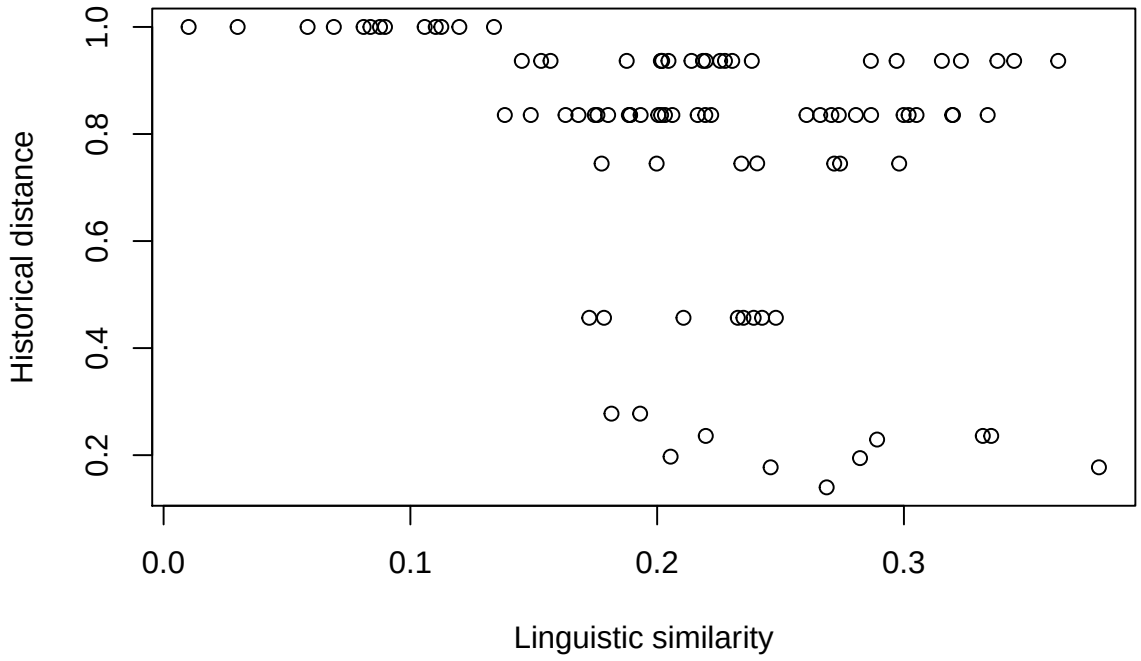
```
plot(as.dist(cult.m2),as.dist(ling.m2),
     xlab="Cultural similarity",
     ylab="Linguistic similarity")
```

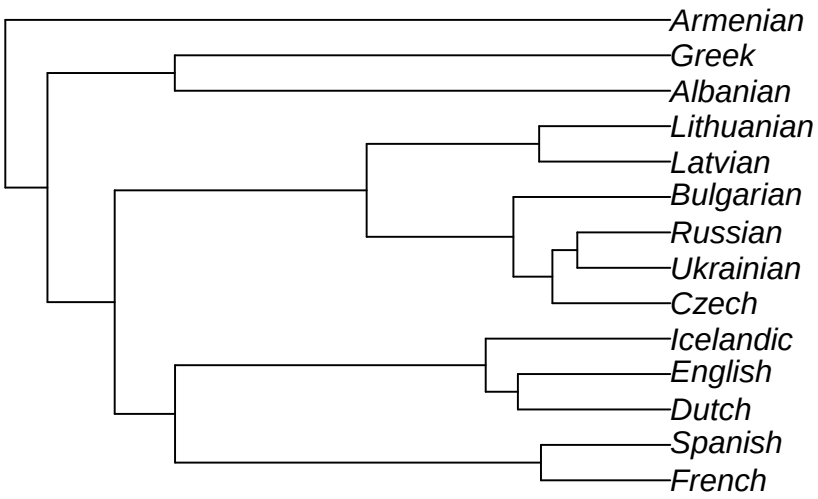



```
plot(as.dist(cult.m2),as.dist(hist.m2),
     xlab="Cultural similarity",
     ylab="Historical distance")
```




```
plot(as.dist(ling.m2),as.dist(hist.m2),
     xlab="Linguistic similarity",
     ylab="Historical distance")
```


Historical distance
1.0
0.8
0.6
0.4
0.2
0.0
0.1
0.2
0.3
Linguistic similarity


Armenian
Greek
Albanian
Lithuanian
Latvian
Bulgarian
Russian
Ukrainian
Czech
Icelandic
English
Dutch
Spanish
French

Tests

The results of the test list the following measures:

- mantelr: Mantel correlation coefficient.
- pval1: one-tailed p-value (null hypothesis: r <= 0).
- pval2: one-tailed p-value (null hypothesis: r >= 0).
- pval3: two-tailed p-value (null hypothesis: r = 0).
- llim: lower confidence limit for r.
- ulim: upper confidence limit for r.

```
set.seed(1498)
```

Run tests between each pair of measures.

```
distms = list("Cultrual"= cult.m2,
              "Linguistic" = ling.m2,
              "Historical" = hist.m2,
              "Geographic" = geo.m2)

mantelRes1 = data.frame(
  Var1 = NA, Var2 = NA, r = NA,
  llim = NA, ulim = NA, p = NA,
  stringsAsFactors = F)

for(i in 1:3){
  for(j in (i+1):4){
    var1 = names(distms)[i]
    var2 = names(distms)[j]
    print(paste("Correlation between",
                var1,"and",var2))
    stat = ecodist::mantel(as.dist(distms[[i]]) ~
                as.dist(distms[[j]]),
                nperm = 100000)
    print(stat)
    mantelRes1 = rbind(mantelRes1,
        c(var1,var2,stat[1],stat[5],stat[6],
          min(c(stat[2],stat[3]))))
    stat = round(stat,2)
    stat2 = sprintf("$r$ = %s[%s,%s], one-tailed $p$ = %s",
      stat[1],
      stat[5],
      stat[6],
      min(c(stat[2],stat[3])))
    # TODO: output stats
    #cat(stat2,file=
     #     paste0("../results/stats/tex/Mantel",var1,"Vs",var2,"Distance_SUBS.tex"))
  }
}
```

```
## [1] "Correlation between Cultrual and Linguistic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
##  0.3509606  0.0883600  0.9116500  0.1382200  0.2413895  0.5377689
## [1] "Correlation between Cultrual and Historical"
##     mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
## -0.16966425  0.84537000  0.15464000  0.30858000 -0.28569297 -0.01526378
```

```
## [1] "Correlation between Cultrual and Geographic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.3397939  0.9723900  0.0276200  0.0308800 -0.5704463 -0.1718367
## [1] "Correlation between Linguistic and Historical"
##     mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
## -0.35203243  0.96988000  0.03013000  0.03453000 -0.50321827 -0.08218648
## [1] "Correlation between Linguistic and Geographic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.2734487  0.9193900  0.0806200  0.1277600 -0.4617151 -0.0818242
## [1] "Correlation between Historical and Geographic"
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
##  0.3457543  0.0100300  0.9899800  0.0100300  0.1811727  0.5217647
```

```
mantelRes1= mantelRes1[2:nrow(mantelRes1),]
mantelRes1[,3:6] = apply(mantelRes1[,3:6],2,function(X){
  signif(as.numeric(X),3)
})
```

Run a mantel test comparing the Linguistic alignment to the cultural similarity, controlling for the historical distance between languages:

```
ecodist::mantel(as.dist(ling.m2)~
                  as.dist(cult.m2) +
                  as.dist(hist.m2),
                nperm = 100000)
```

```
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
##  0.3157282  0.1119100  0.8881000  0.1894500  0.1495754  0.5004242
```

Main Test: Run a mantel test comparing the Linguistic alignment to the cultural similarity, controlling for the historical distance and geographic distance between languages:

```
mainMantel = ecodist::mantel(as.dist(ling.m2)~
                  as.dist(cult.m2) +
                  as.dist(hist.m2) +
                  as.dist(geo.m2),
                nperm = 100000)
mainMantel = signif(mainMantel,3)

mantelRes1 = rbind(mantelRes1,
  c("Linguistic","Cultural **",
    mainMantel[1],mainMantel[5],mainMantel[6],
    min(mainMantel[2:3])))

mantelRes1Text = xtable(mantelRes1,
   caption =  paste0(
    "Mantel tests (",
    datasetLabel,
    "). ** = partial Mantel test, controlling for historical and geographical distance."))
print(mantelRes1Text,
    file="../results/stats/tex/Mantel_SUBS.tex")
```

MRM

Perform the main test, but using multiple regression on distance matrices (MRM).

```
mainMRM = ecodist::MRM(as.dist(ling.m2)~
                  as.dist(cult.m2) +
                  as.dist(hist.m2) +
                  as.dist(geo.m2), nperm=10000)
mainMRM
```

```
## $coef
##                   as.dist(ling.m2)   pval
## Int                     0.14290281 0.6955
## as.dist(cult.m2)        0.24395908 0.2398
## as.dist(hist.m2)       -0.08531160 0.1408
## as.dist(geo.m2)        -0.02225994 0.6608
##
## $r.squared
##        R2      pval
## 0.2169477 0.1326000
##
## $F.test
##        F   F.pval
## 8.034563 0.132600
```

```
mainMRM2 = sprintf("$\\beta= $%s, $p=$%s",
                   round(mainMRM$coef[2,1],2),
                   round(mainMRM$coef[2,2],2))
cat(mainMRM2,
    file="../results/stats/tex/MRMCultrualVsLinguisticDistance_Partial_SUBS.tex")
```

4.6 Summary of findings

Summary of findings

Main analyses of Wikipedia data.

See the main text for a summary of the main analyses using the Wikipedia data.

Analyses of the estimates from the Common Crawl data

Mixed effects model: The correlation between semantic alignment and cultural similarity was significant (β= 0.182, $\chi^2(1)$= 4.94, p=0.026). See the figure 1 below:

MRM results:

	Estimate	p-value
Intercept	0.306	0.031
Cultural distance	0.272	0.0373
Language family	-0.273	0.0981
Geographic distance	0.148	0.0976
Comparison count	0.110	0.138

Table 1: MRM analysis predicting semantic alignment (Common Crawl), with family control. R^2=0.0999

	Estimate	p-value
Intercept	0.190	0.0001
Cultural distance	0.285	0.0405
ASJP	-0.464	0.0001
Geographic distance	0.229	0.0103
Comparison count	0.087	0.244

Table 2: MRM analysis predicting semantic alignment (Common Crawl), with ASJP control. R^2=0.185

Mantel tests:

	Var1	Var2	r	llim	ulim	p
2	Cultrual	Linguistic	0.18	0.0131	0.286	0.12
3	Cultrual	Historical	-0.315	-0.447	-0.204	0.0211
4	Cultrual	Geographic	-0.461	-0.586	-0.312	0.00295
5	Linguistic	Historical	-0.404	-0.523	-0.194	0.00104
6	Linguistic	Geographic	-0.0837	-0.202	-0.00552	0.262
7	Historical	Geographic	0.405	0.31	0.519	0.00101
71	Linguistic	Cultural **	0.106	0.000414	0.191	0.226

Table 3: Mantel tests (Common Crawl). ** = partial Mantel test, controlling for historical and geographical distance.

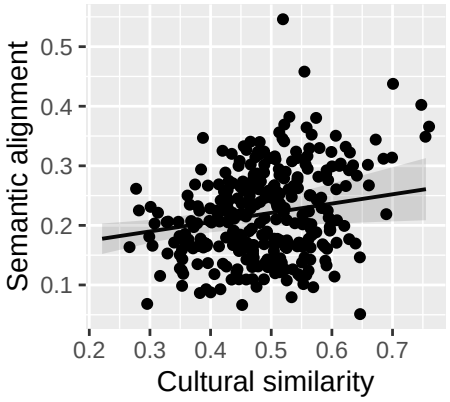


Figure 1: Semantic alignment and cultural similarity for data using the Common Crawl alignments

Analyses of the estimates from the Subtitles data

Mixed effects model: The correlation between semantic alignment and cultural similarity was not significant (β= 0.0606, $\chi^2(1)$= 0.74, p=0.39). See the figure 2 below:

MRM results:

	Estimate	p-value
Intercept	0.073	0.517
Cultural distance	0.125	0.413
Language family	-0.041	0.838
Geographic distance	-0.013	0.887
Comparison count	0.806	0.0002

Table 4: MRM analysis predicting semantic alignment (Subtitles), with family control. R^2=0.744

	Estimate	p-value
Intercept	0.083	0.297
Cultural distance	-0.023	0.884
ASJP	-0.282	0.0083
Geographic distance	0.019	0.854
Comparison count	0.831	0.0003

Table 5: MRM analysis predicting semantic alignment (Subtitles), with ASJP control. R^2=0.803

Mantel tests:

	Var1	Var2	r	llim	ulim	p
2	Cultrual	Linguistic	0.351	0.241	0.538	0.0884
3	Cultrual	Historical	-0.17	-0.286	-0.0153	0.155
4	Cultrual	Geographic	-0.34	-0.57	-0.172	0.0276
5	Linguistic	Historical	-0.352	-0.503	-0.0822	0.0301
6	Linguistic	Geographic	-0.273	-0.462	-0.0818	0.0806
7	Historical	Geographic	0.346	0.181	0.522	0.01
71	Linguistic	Cultural **	0.281	0.129	0.466	0.135

Table 6: Mantel tests (Subtitles). ** = partial Mantel test, controlling for historical and geographical distance.

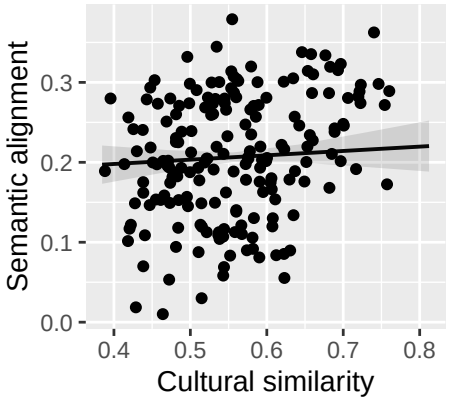


Figure 2: Semantic alignment and cultural similarity for data using the Subtitles alignments

Analysis of numerals

The analyses of numerals found:

- 1 and 2 have lower alignment due to often being grammaticalised as indefinite or dual marker (Givon, 1981).
- Numbers 3-12 generally have high alignment (mean local alignment = 0.87), and higher numbers decline in alignment up to 1000.
- There are also language-specific differences due to how numerals are constructed (e.g. base, combination rules, see Calude & Verkerk, 2016), or for irregular forms (e.g. 50, 60, 70, 80 and 90 in Danish).
- Some number words have alternative associations due to homophones (e.g. the Hungarian 7 is used directly to mean 'week', and 'neuf' in French means '9' or 'new').
- The historical distance between languages did not predict much of the variation.

See the main text for a discussion.