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)
```

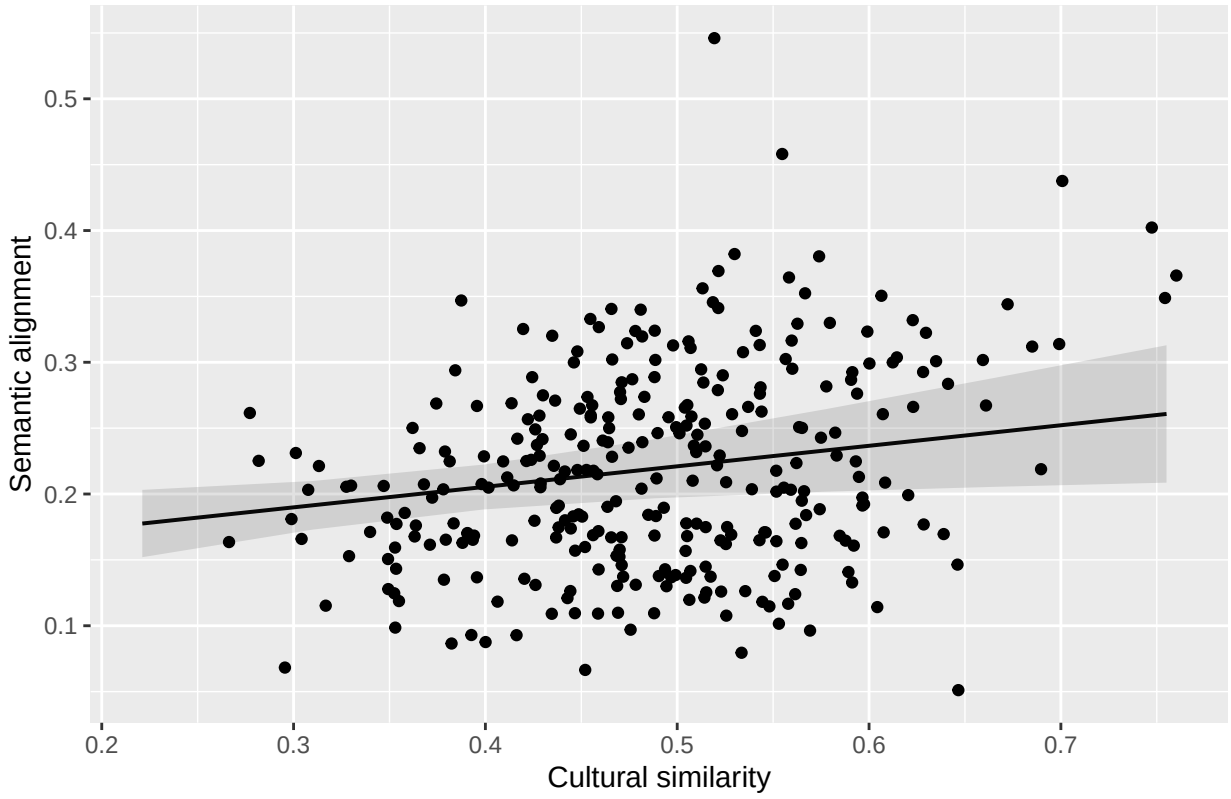
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ 1 + comparison_count.center + cult.dist.center +
##     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
##     area.group)
##    Data: ling
##
## REML criterion at convergence: 784.1
##
## Scaled residuals:
##     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
```



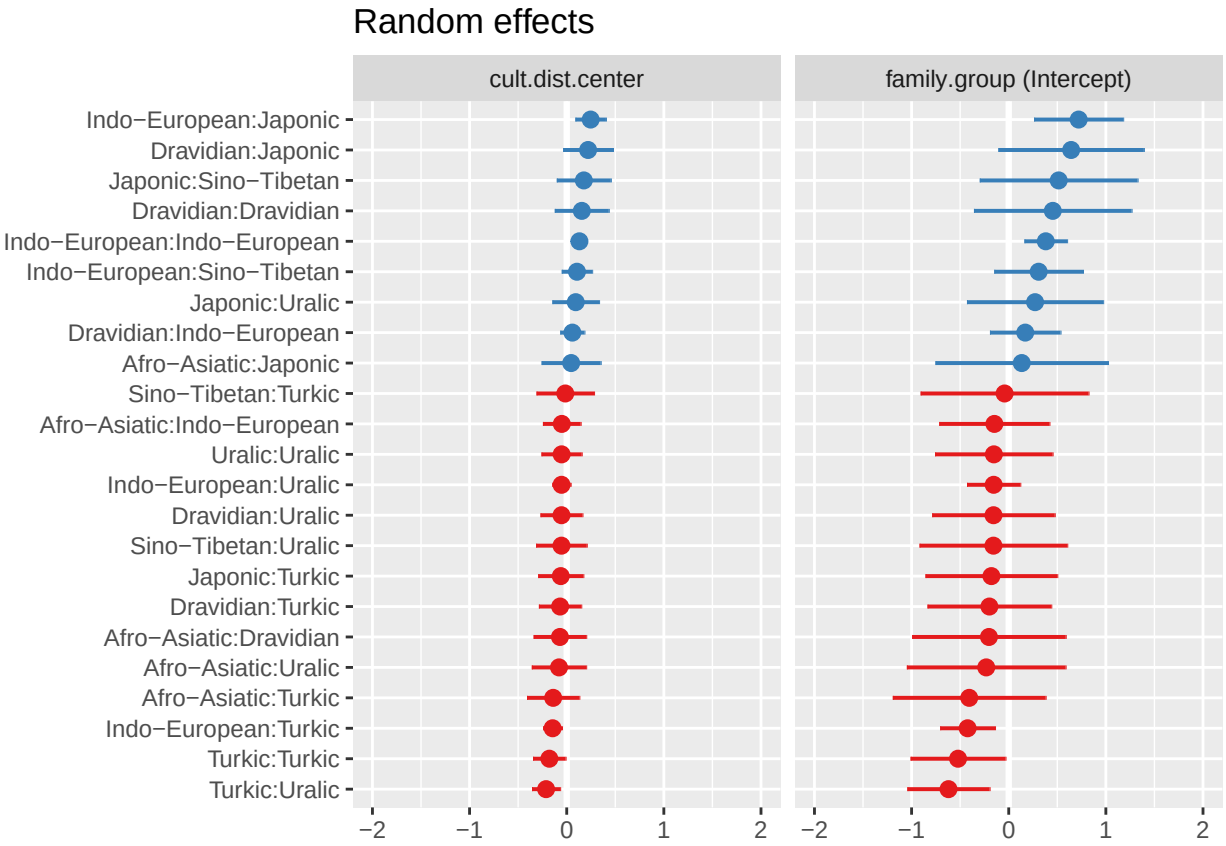

```
pdf(paste0("../results/stats/",datasetName,"/CulturalDistance_Rho_Graph.pdf"),
    height=2.5, width=2.5)
gx
dev.off()
```

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

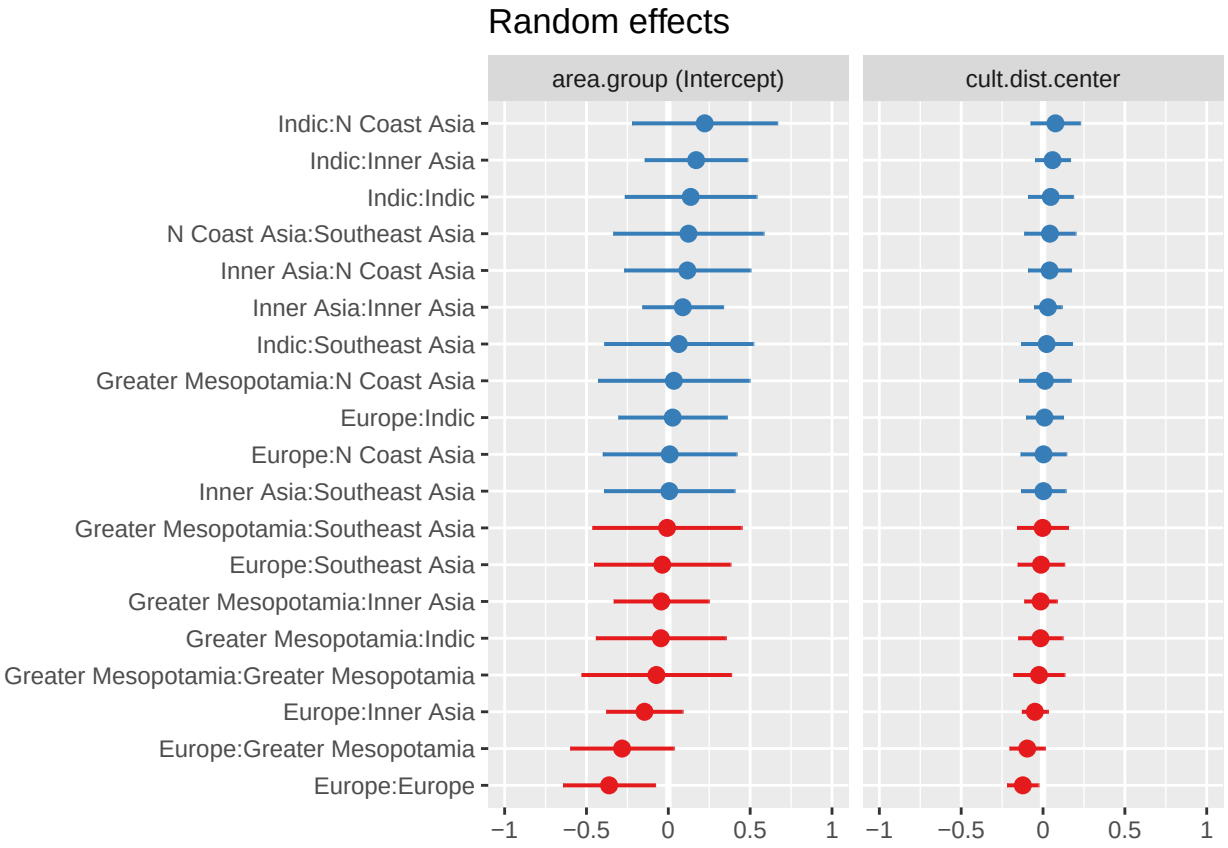
Plot the random effects:

```
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]]
```



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)]

# 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.

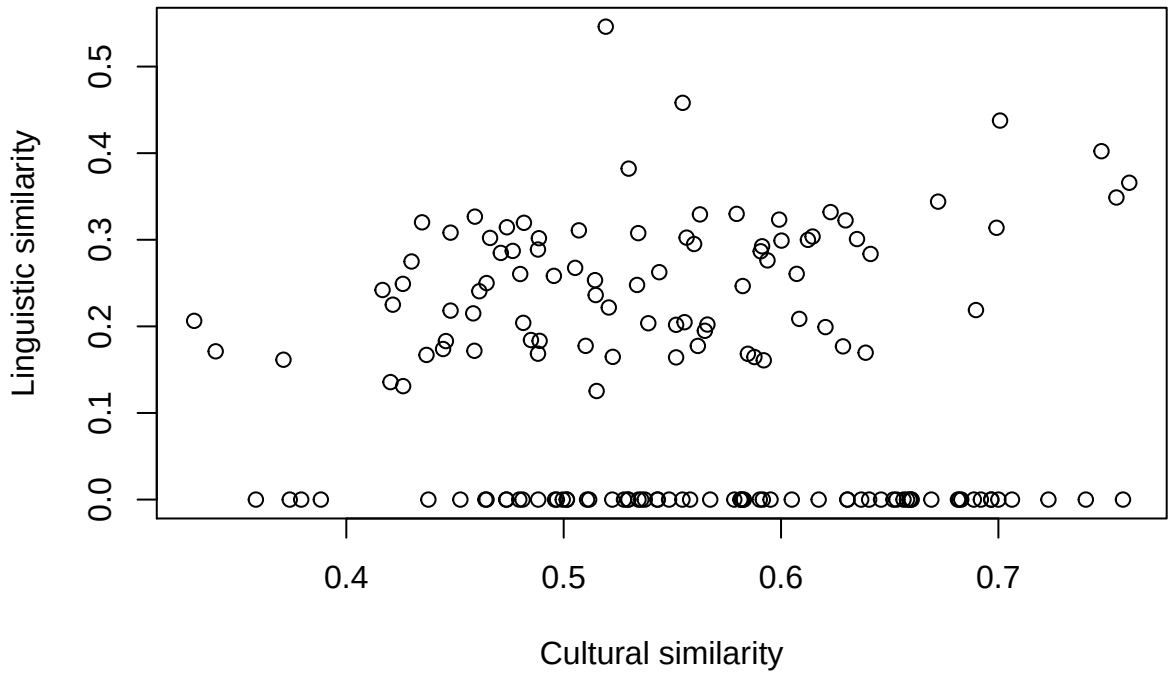
```
grph <- graph.data.frame(ling[,c("l1",'l2','local_alignment')], directed=FALSE)
# add value as a weight attribute
ling.m = get.adjacency(grph, attr="local_alignment", sparse=FALSE)
rownames(ling.m) = l[match(rownames(ling.m),l$iso2),]$Language2
colnames(ling.m) = l[match(colnames(ling.m),l$iso2),]$Language2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

Match the distance matrices

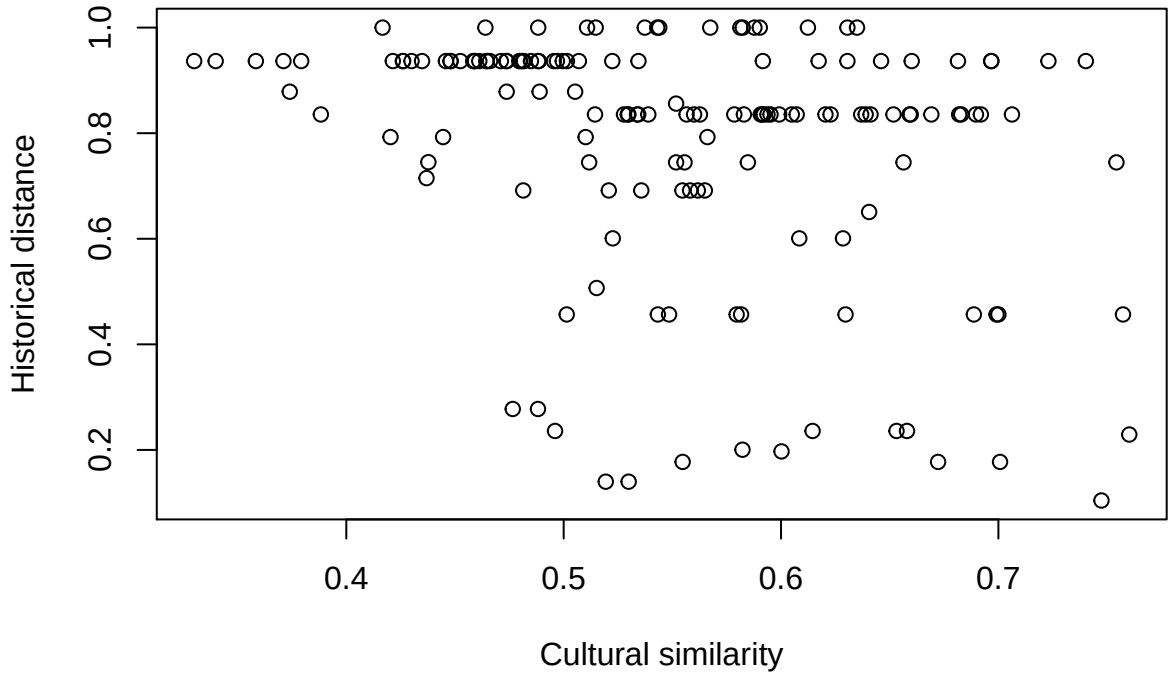
```
in.analysis = intersect(rownames(ling.m),rownames(cult.m))
in.analysis = intersect(in.analysis, rownames(hist.m))
cult.m2 = cult.m[in.analysis,in.analysis]
ling.m2 = ling.m[in.analysis,in.analysis]
hist.m2 = hist.m[in.analysis,in.analysis]
geo.m2 = geoDist.m[in.analysis,in.analysis]
```

Note that there are only 18 languages with data on lingusitic, cultural and historical distance.

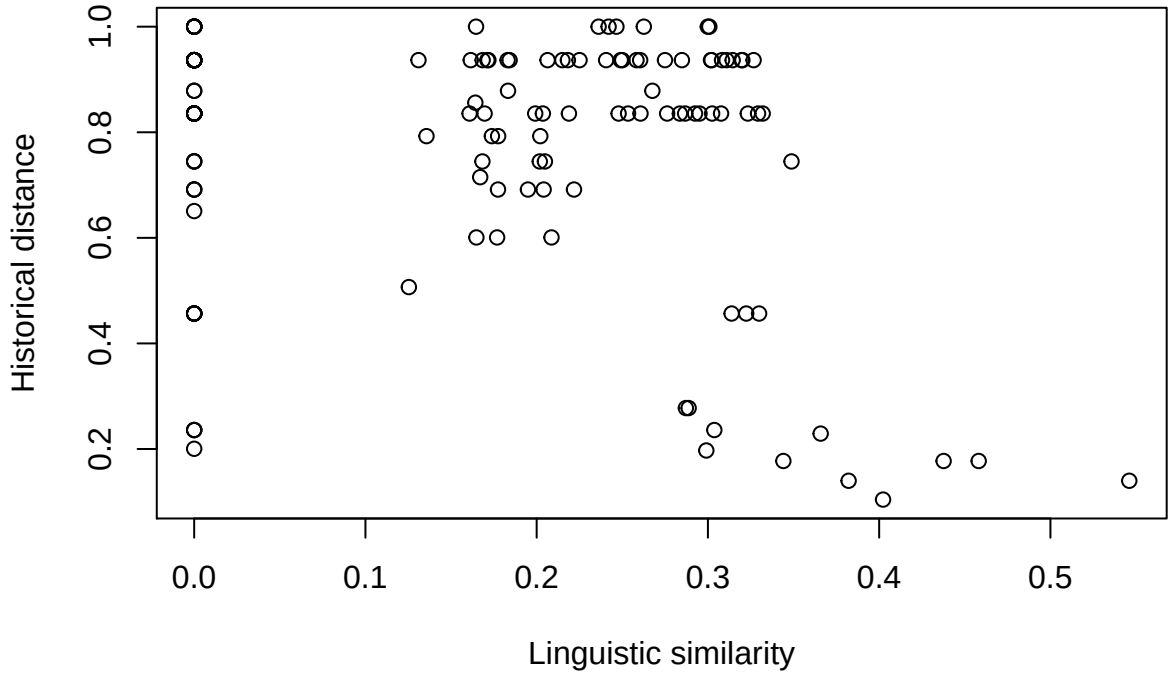
```
plot(as.dist(cult.m2),as.dist(ling.m2),
     xlab="Cultural similarity",
     ylab="Linguistic similarity")
```

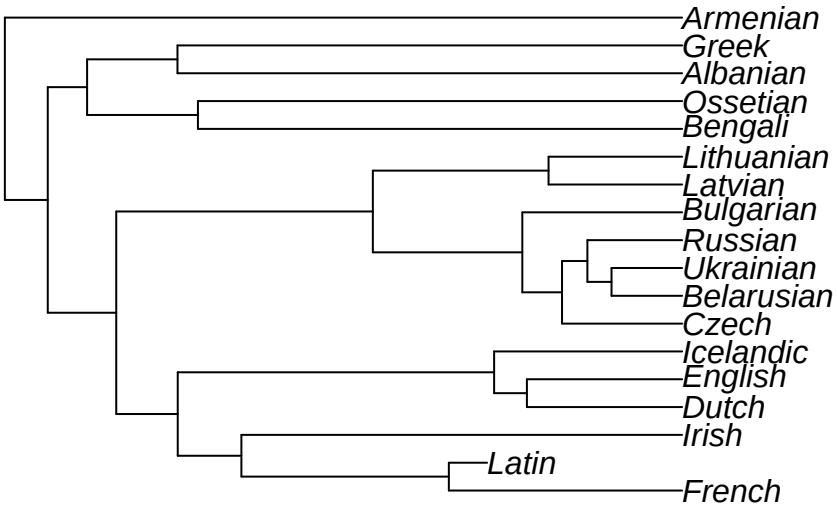



```
plot(as.dist(cult.m2),as.dist(hist.m2),
     xlab="Cultural similarity",
     ylab="Historical distance")
```




```
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
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%
## -0.0839735358  0.7232800000  0.2767300000  0.5379200000 -0.2009313168
##    ulim.97.5%
##  0.0008846432
## [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%
## -0.2496921  0.9850200  0.0149900  0.0220500 -0.3570435 -0.1142948
## [1] "Correlation between Linguistic and Geographic"
##     mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
##  0.13621101  0.12091000  0.87910000  0.25937000 -0.01342846  0.23456752
## [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%
## -0.17690227  0.91913000  0.08088000  0.17506000 -0.24784616 -0.08286161
```

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                      0.3645355 0.0352
## as.dist(cult.m2)        -0.1327268 0.5157
## as.dist(hist.m2)        -0.2254171 0.0010
## as.dist(geo.m2)          0.1030124 0.0503
##
## $r.squared
##       R2     pval
## 0.135791 0.017100
##
## $F.test
##      F F.pval
## 7.8040 0.0171
```

```
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")
```