Cultural distances: Wikipedia data

Contents

Introduction

We compare cultural distances between socieites with linguistic similarities 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) is used as a random effect. That means that the model can capture the likelihood that two languages from the Indo-European language family will be more similar to each other than two languages from different language families. The same is done with geographic area according to Autotyp.

The second test controls for history using distances from a phylogenetic tree. The tree comes from Bouckaert et al. (2012). Patristic distances between languages are used as a measure of historical distance between societies in a Mantel test. Note that the Mantel test assumes a strict distance metric, which is not necessarily the case with this data, but there are few other ways to deal with continuous pairwise distances.

Load libraries

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

Parameters:

```r
datasetName = "wikipedia-main"
lingDistancesFile = "../data/FAIR/nel-wiki-k100-alignments-by-language-pair.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")
cultLangs = unique(c(cult$Var1,cult$Var2))
```

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
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] 733
```

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

```
## [1] 40
```

```
# 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:

```r
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          5            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 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
)
m0.5 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
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
)
anova(m0,m0.5,m1)
```

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

```
## 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 1658.0 1694.7 -820.97   1642.0
## m0.5  9 1298.0 1339.3 -639.98   1280.0 361.979      1  < 2.2e-16 ***
## m1   10 1285.3 1331.3 -632.65   1265.3  14.675      1  0.0001277 ***
## ---
## 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)
```

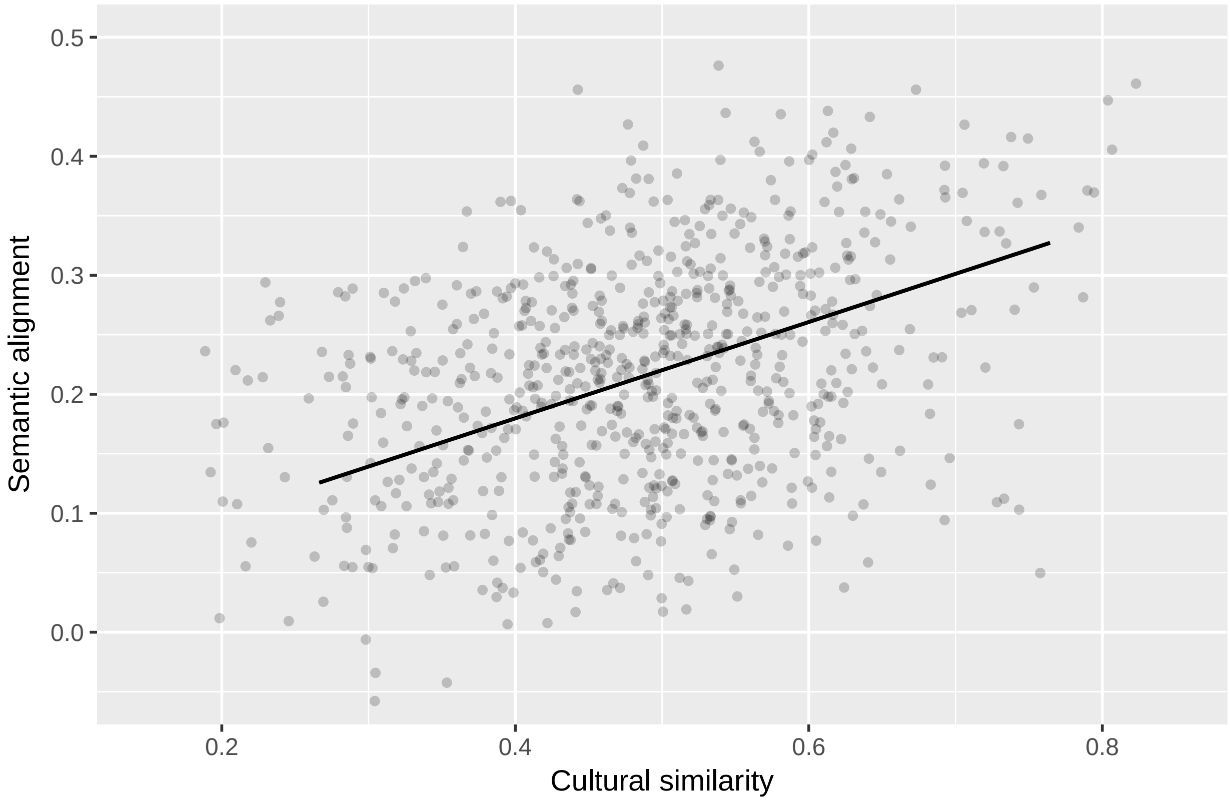
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ 1 + comparison_count.center + cult.dist.center +
##     (1 + cult.dist.center | family.group) + (1 + cult.dist.center |
##     area.group)
##    Data: ling
##
## REML criterion at convergence: 1278.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6298 -0.6127  0.0942  0.6554  4.7348
##
## Random effects:
##  Groups       Name             Variance  Std.Dev. Corr
##  family.group (Intercept)      0.1688145 0.41087
##               cult.dist.center 0.0004627 0.02151  1.00
##  area.group   (Intercept)      0.0464131 0.21544
##               cult.dist.center 0.0029874 0.05466  -1.00
##  Residual                      0.2900042 0.53852
## Number of obs: 733, groups:  family.group, 48; area.group, 20
##
## Fixed effects:
##                         Estimate Std. Error t value
## (Intercept)             -0.39344    0.09035  -4.354
## comparison_count.center  0.60900    0.02697  22.580
## cult.dist.center         0.18268    0.03233   5.651
##
## Correlation of Fixed Effects:
##             (Intr) cmpr_.
## cmprsn_cnt.  0.093
## clt.dst.cnt -0.120 -0.202
```

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

```
gx = sjp.lmer(m1,'pred','cult.dist.center',
              prnt.plot = F)
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.2
```

```
gx$plot$data$y = gx$plot$data$y *
  attr(ling$rho.center,"scaled:scale") +
  attr(ling$rho.center,"scaled:center")
gx$plot$data$resp.y = gx$plot$data$resp.y *
  attr(ling$rho.center,"scaled:scale") +
  attr(ling$rho.center,"scaled:center")
gx$plot$data$x = gx$plot$data$x *
  cdc.s +cdc.c
gx = gx$plot + coord_cartesian(ylim=c(-0.05,0.5),
                          xlim=c(0.15,0.85)) +
  xlab("Cultural similarity") +
  ylab("Semantic alignment") +
  ggtitle("")
gx
```



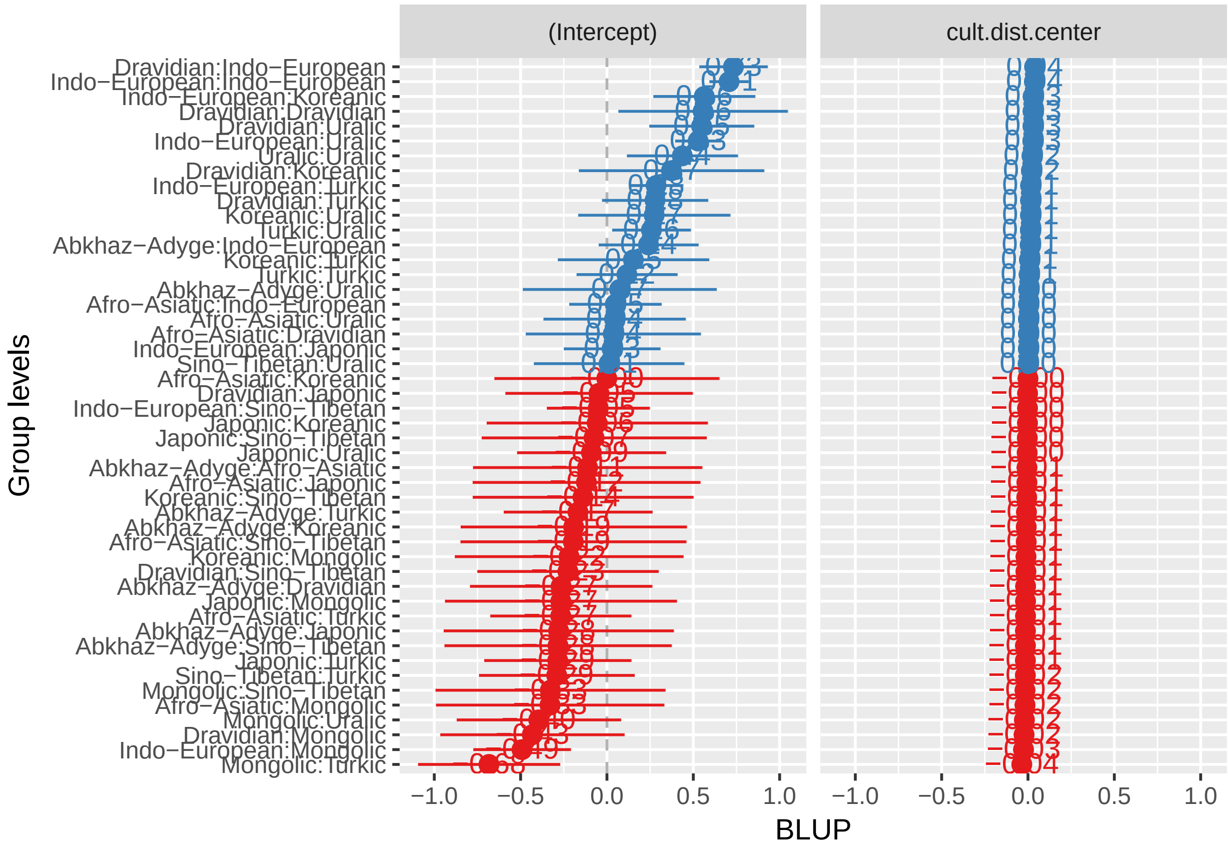
```
pdf(paste0("../results/stats/",datasetName,"/CulturalDistance_Rho_Graph.pdf"),
    height=2.5, width=2.5)
gx
dev.off()
```

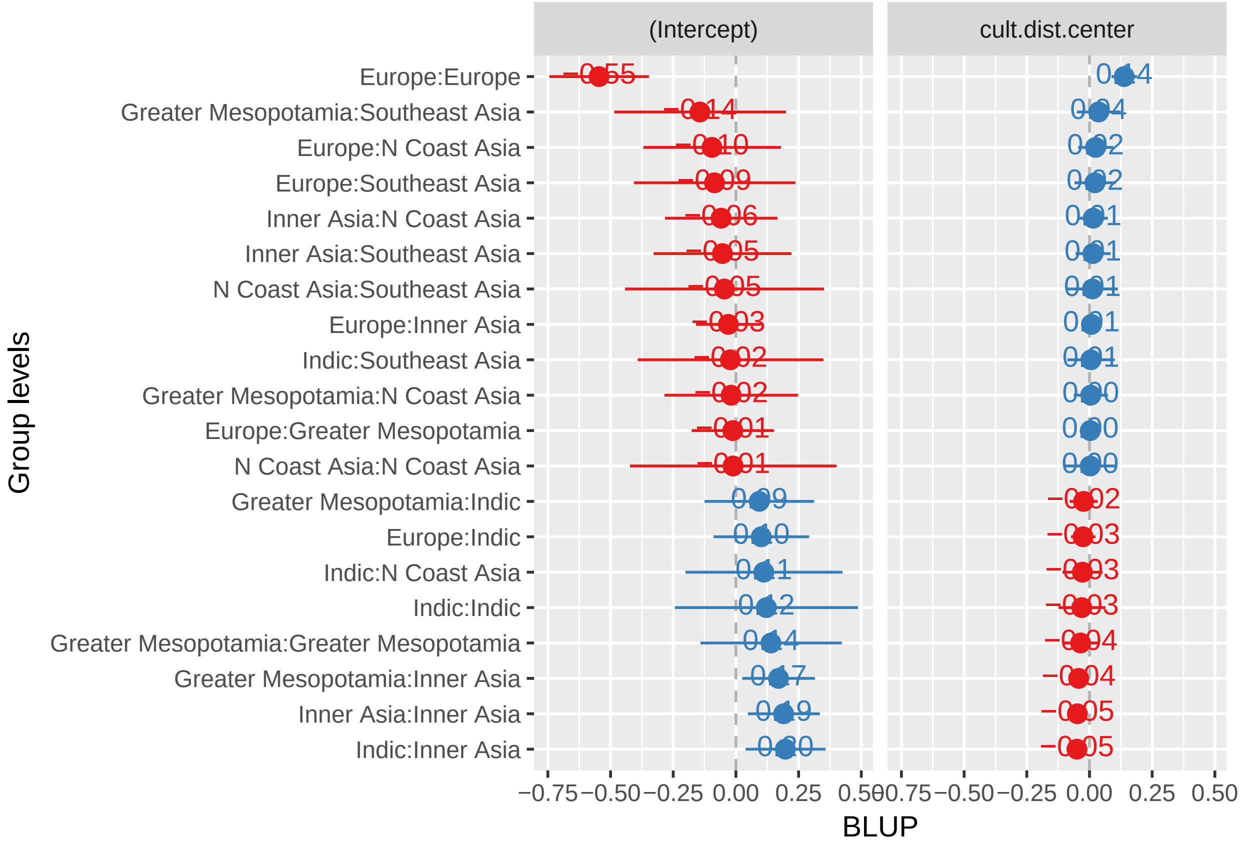
```
## pdf
##   2
```

Plot the random effects:

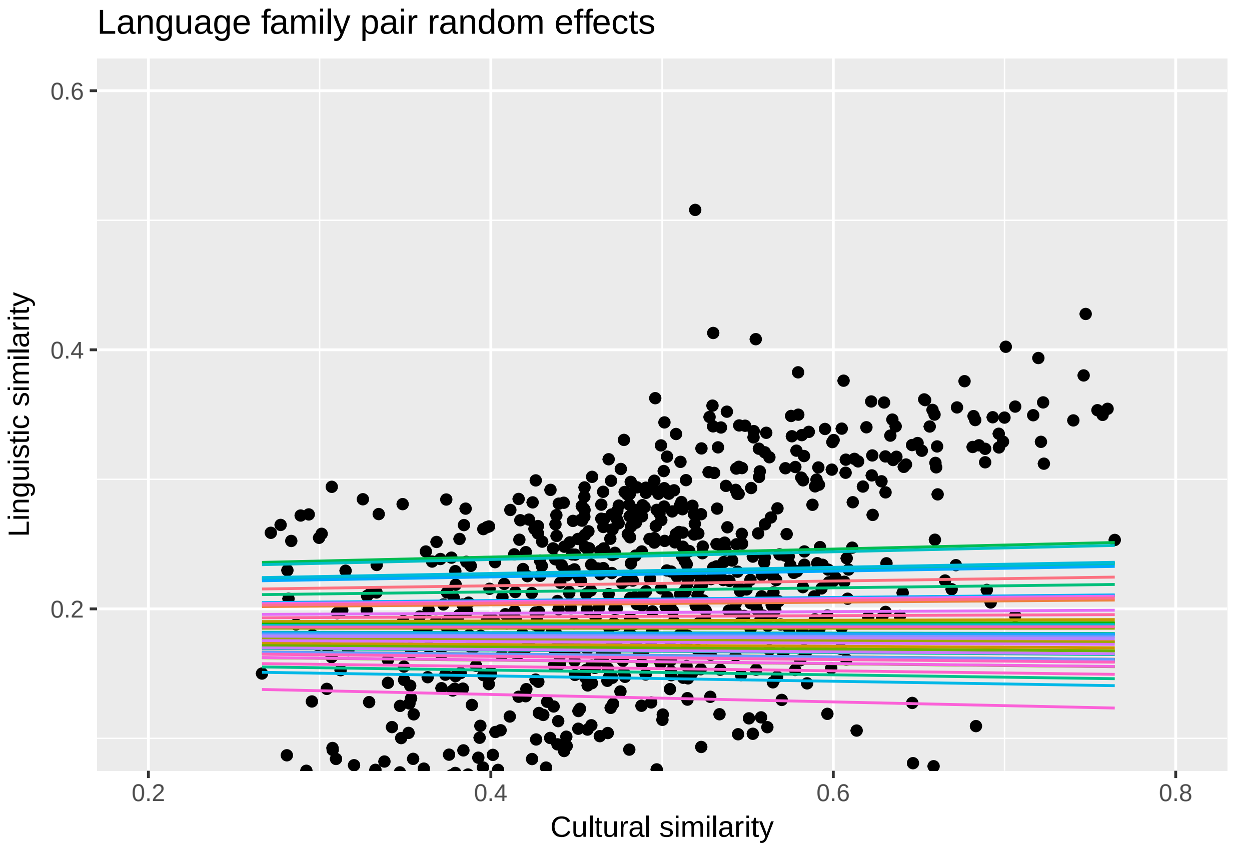
```
sjp.lmer(m1,'re', sort.est = "cult.dist.center")
```

```
## Plotting random effects...
## Plotting random effects...
```

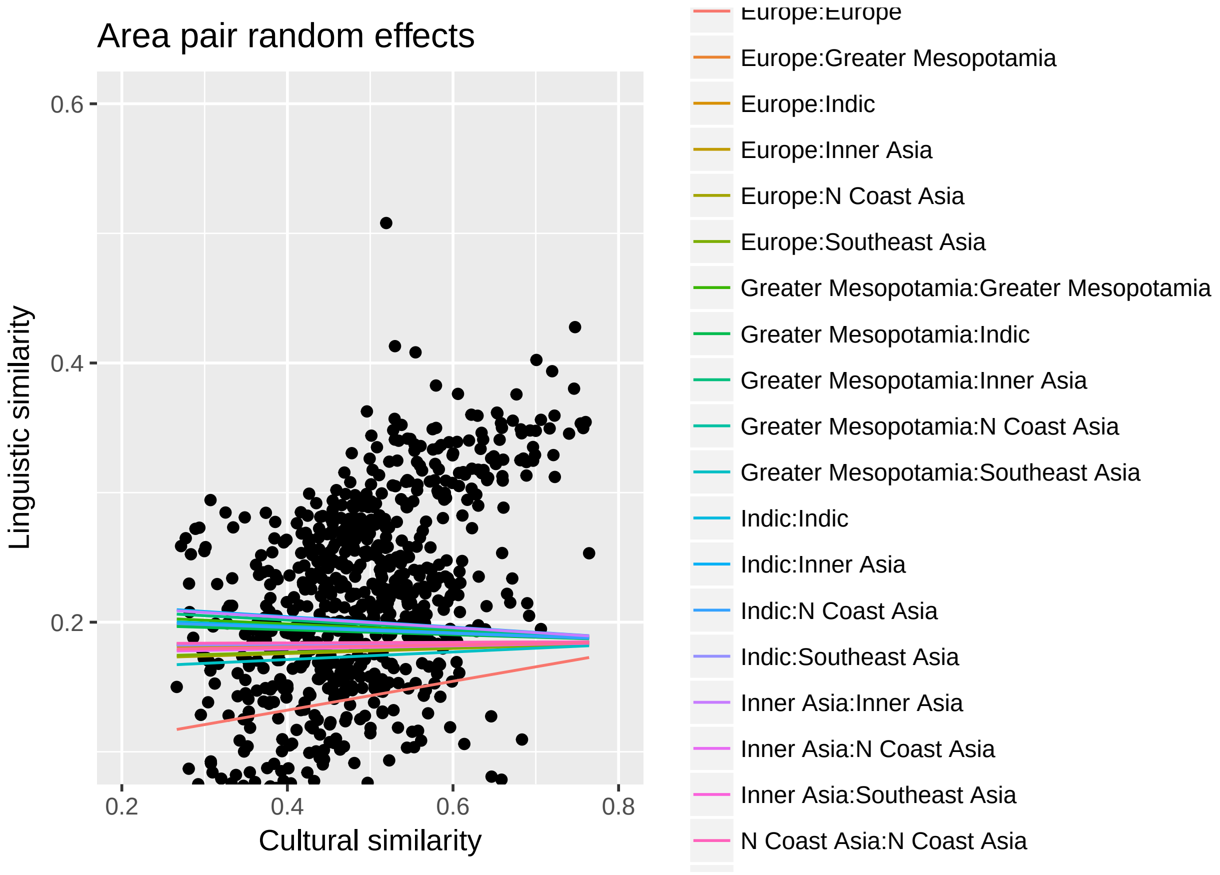

(Intercept)
cult.dist.center
Group levels
Dravidian:Indo−European
Indo−European:Indo−European
Indo−European:Koreanic
Dravidian:Dravidian
Dravidian:Uralic
Indo−European:Uralic
Uralic:Uralic
Dravidian:Koreanic
Indo−European:Turkic
Dravidian:Turkic
Koreanic:Uralic
Turkic:Uralic
Abkhaz−Adyge:Indo−European
Koreanic:Turkic
Turkic:Turkic
Abkhaz−Adyge:Uralic
Afro−Asiatic:Indo−European
Afro−Asiatic:Uralic
Afro−Asiatic:Dravidian
Indo−European:Japonic
Sino−Tibetan:Uralic
Afro−Asiatic:Koreanic
Dravidian:Japonic
Indo−European:Sino−Tibetan
Japonic:Koreanic
Japonic:Sino−Tibetan
Japonic:Uralic
Abkhaz−Adyge:Afro−Asiatic
Afro−Asiatic:Japonic
Koreanic:Sino−Tibetan
Abkhaz−Adyge:Turkic
Abkhaz−Adyge:Koreanic
Afro−Asiatic:Sino−Tibetan
Koreanic:Mongolic
Dravidian:Sino−Tibetan
Abkhaz−Adyge:Dravidian
Japonic:Mongolic
Afro−Asiatic:Turkic
Abkhaz−Adyge:Japonic
Abkhaz−Adyge:Sino−Tibetan
Japonic:Turkic
Sino−Tibetan:Turkic
Mongolic:Sino−Tibetan
Afro−Asiatic:Mongolic
Mongolic:Uralic
Dravidian:Mongolic
Indo−European:Mongolic
Mongolic:Turkic
−1.0
−0.5
0.0
0.5
1.0
−1.0
−0.5
0.0
0.5
1.0
BLUP


(Intercept)
cult.dist.center
Group levels
Europe:Europe
Greater Mesopotamia:Southeast Asia
Europe:N Coast Asia
Europe:Southeast Asia
Inner Asia:N Coast Asia
Inner Asia:Southeast Asia
N Coast Asia:Southeast Asia
Europe:Inner Asia
Indic:Southeast Asia
Greater Mesopotamia:N Coast Asia
Europe:Greater Mesopotamia
N Coast Asia:N Coast Asia
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
−0.55
−0.14
−0.10
−0.09
−0.06
−0.05
−0.05
−0.03
−0.02
−0.02
−0.01
−0.01
0.09
0.10
0.11
0.12
0.14
0.17
0.19
0.20
0.14
0.04
0.02
0.02
0.01
0.01
0.01
0.01
0.01
0.00
0.00
0.00
−0.02
−0.03
−0.03
−0.03
−0.04
−0.04
−0.05
−0.05
−0.75
−0.50
−0.25
0.00
0.25
0.50
−0.75
−0.50
−0.25
0.00
0.25
0.50
BLUP

```
px = sjp.lmer(m1,'rs.ri', prnt.plot = F)
dx = px$plot[[1]]$data
dx$x = dx$x * cdc.s + cdc.c
dx$y = dx$y *
  attr(ling$rho.center,"scaled:scale") +
  attr(ling$rho.center,"scaled:center")
ggplot(dx,aes(x,y)) +
  geom_point(data=ling,
           mapping=aes(x=as.numeric(cult.dist),
                       y=as.numeric(local_alignment))) +
  geom_line(mapping = aes(colour=grp)) +
  xlab("Cultural similarity")+
  ylab("Linguistic similarity") +
  ggtitle("Language family pair random effects") +
  coord_cartesian(ylim=c(0.1,0.6),
                 xlim=c(0.2,0.8)) +
  theme(legend.position = "none")
```



```
dx = px$plot[[2]]$data
dx$x = dx$x * cdc.s + cdc.c
dx$y = dx$y *
  attr(ling$rho.center,"scaled:scale") +
  attr(ling$rho.center,"scaled:center")
ggplot(dx,aes(x,y)) +
  geom_point(data=ling,
           mapping=aes(x=as.numeric(cult.dist),
                       y=as.numeric(local_alignment))) +
geom_line(mapping = aes(colour=grp)) +
xlab("Cultural similarity")+
ylab("Linguistic similarity") +
ggtitle("Area pair random effects") +
coord_cartesian(ylim=c(0.1,0.6),
                xlim=c(0.2,0.8))
```



Tests within domains

Load data

Load distances for specific domains and match up to language family and area:

```
ling.dom = read.csv(
  lingDistancesByDomainFile,
  stringsAsFactors = F)

ling.dom = ling.dom[!is.na(ling.dom$cult.dist),]

ling.dom = ling.dom[(ling.dom$l1 %in% cultisos) &
                      (ling.dom$l2 %in% cultisos),]
```

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

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

Match family and area data:

```
ling.dom$family1 = l[match(ling.dom$l1, l$iso2),]$family
ling.dom$family2 = l[match(ling.dom$l2, l$iso2),]$family
ling.dom$area1 = l[match(ling.dom$l1, l$iso2),]$autotyp.area
ling.dom$area2 = l[match(ling.dom$l2, l$iso2),]$autotyp.area


# Paste language family names together,
# but order shouldn't matter, so sort first
fgroup = cbind(ling.dom$family1,ling.dom$family2)
fgroup = apply(fgroup,1,sort)
ling.dom$family.group = apply(fgroup,2,paste,collapse=":")

agroup = cbind(ling.dom$area1,ling.dom$area2)
agroup = apply(agroup,1,sort)
ling.dom$area.group = apply(agroup,2,paste,collapse=":")
```

Center the data (and flip cultural distance into cultural similarity):

```
ling.dom$cult.dist = 1-ling.dom$cult.dist
ling.dom$cult.dist.center = scale(ling.dom$cult.dist)
ling.dom$rho.center = scale(ling.dom$local_alignment)
ling.dom$comparison_count.center = scale(ling.dom$comparison_count)
```

LMER models

Test whether random slopes are warraneted for family:

```
mD0 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 | family.group) +
    (1 | area.group) +
    (1 | imputed_semantic_domain),
  data = ling.dom)
mD1 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 | family.group) +
    (0 + cult.dist.center | family.group) +
    (1 | area.group) +
    (1 | imputed_semantic_domain),
  data = ling.dom)
mD2 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 | area.group) +
    (1 | imputed_semantic_domain),
  data = ling.dom)
anova(mD0,mD1,mD2)
```

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

## Data: ling.dom
## Models:
## mD0: rho.center ~ 1 + comparison_count.center + (1 | family.group) +
## mD0:     (1 | area.group) + (1 | imputed_semantic_domain)
## mD1: rho.center ~ 1 + comparison_count.center + (1 | family.group) +
## mD1:     (0 + cult.dist.center | family.group) + (1 | area.group) +
## mD1:     (1 | imputed_semantic_domain)
## mD2: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## mD2:     family.group) + (1 | area.group) + (1 | imputed_semantic_domain)
##     Df   AIC   BIC  logLik deviance    Chisq Chi Df Pr(>Chisq)
## mD0  6 12408 12448 -6198.2    12396
## mD1  7 12181 12228 -6083.6    12167 229.0385      1    < 2e-16 ***
## mD2  8 12178 12232 -6081.0    12162   5.2411      1    0.02206 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Random slopes (and the correlation coefficient) for family improves the fit of the model.

Test the same for area:

```
mD3 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group) +
    (1 | imputed_semantic_domain),
  data = ling.dom)
anova(mD2,mD3)
```

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

## Data: ling.dom
## Models:
## mD2: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## mD2:     family.group) + (1 | area.group) + (1 | imputed_semantic_domain)
## mD3: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## mD3:     family.group) + (1 + cult.dist.center | area.group) + (1 |
## mD3:     imputed_semantic_domain)
##     Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mD2  8 12178 12232 -6081.0    12162
## mD3 10 12163 12230 -6071.6    12143 18.779      2   8.36e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Random slopes for area improves the fit of the model.

Test random slopes for domain:

```
mdom1 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 +cult.dist.center| family.group) +
    (1 +cult.dist.center| area.group) +
    (1 | imputed_semantic_domain),
  data = ling.dom)
mdom2 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 +cult.dist.center| family.group) +
    (1 +cult.dist.center| area.group) +
    (1 +cult.dist.center| imputed_semantic_domain),
  data = ling.dom)
anova(mdom1,mdom2)
```

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

```
## Data: ling.dom
## Models:
## mdom1: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## mdom1:     family.group) + (1 + cult.dist.center | area.group) + (1 |
## mdom1:     imputed_semantic_domain)
## mdom2: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## mdom2:     family.group) + (1 + cult.dist.center | area.group) + (1 +
## mdom2:     cult.dist.center | imputed_semantic_domain)
##       Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mdom1 10 12163 12230 -6071.6    12143
## mdom2 12 12059 12140 -6017.7    12035 107.86      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Random slope for domains significantly improves model.

Now we test the main effect of cultural similarity:

```
mD5 = lmer(
  rho.center ~ 1 +
    comparison_count.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group) +
    (1 + cult.dist.center | imputed_semantic_domain),
  data = ling.dom)
mD6 = update(mD5, ~.+cult.dist.center)
anova(mD5,mD6)
```

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

```
## Data: ling.dom
## Models:
```

```
## mD5: rho.center ~ 1 + comparison_count.center + (1 + cult.dist.center |
## mD5:     family.group) + (1 + cult.dist.center | area.group) + (1 +
## mD5:     cult.dist.center | imputed_semantic_domain)
## mD6: rho.center ~ comparison_count.center + (1 + cult.dist.center |
## mD6:     family.group) + (1 + cult.dist.center | area.group) + (1 +
## mD6:     cult.dist.center | imputed_semantic_domain) + cult.dist.center
##     Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mD5 12 12059 12140 -6017.7    12035
## mD6 13 12061 12148 -6017.7    12035 0.0759      1     0.7829
```

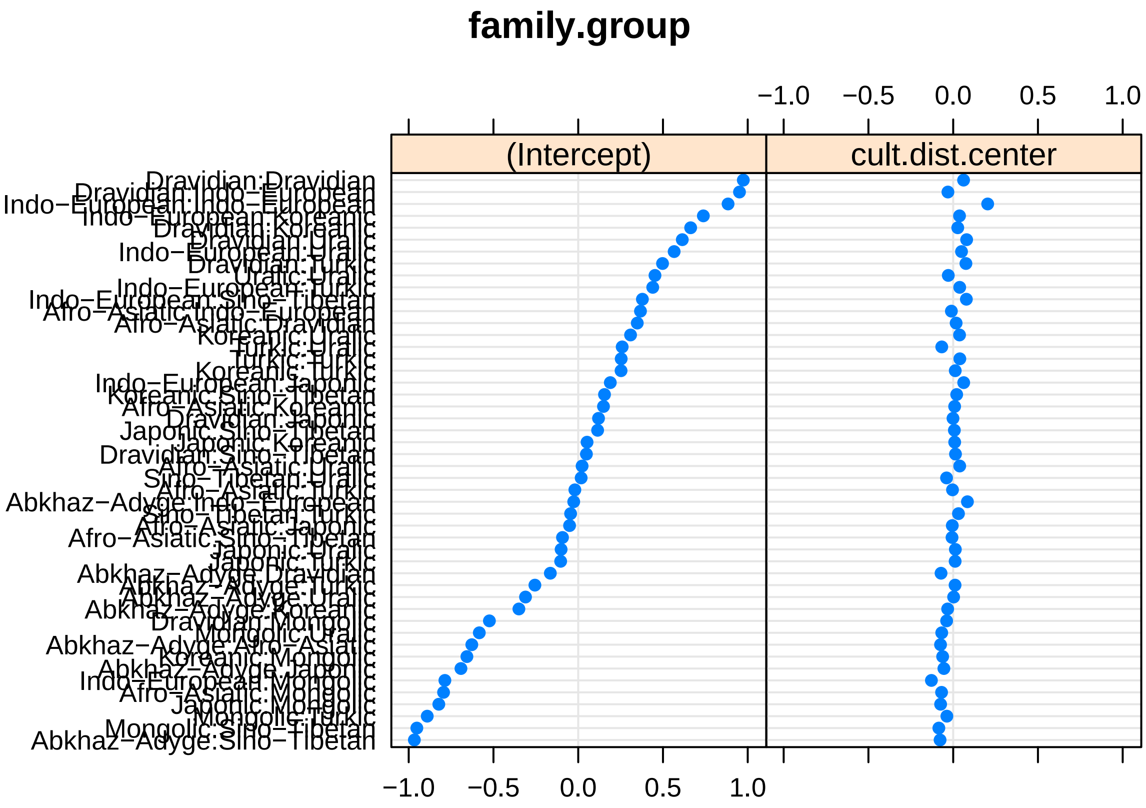
Summary of the final model, with random effects plot:

```
summary(mD6)
```

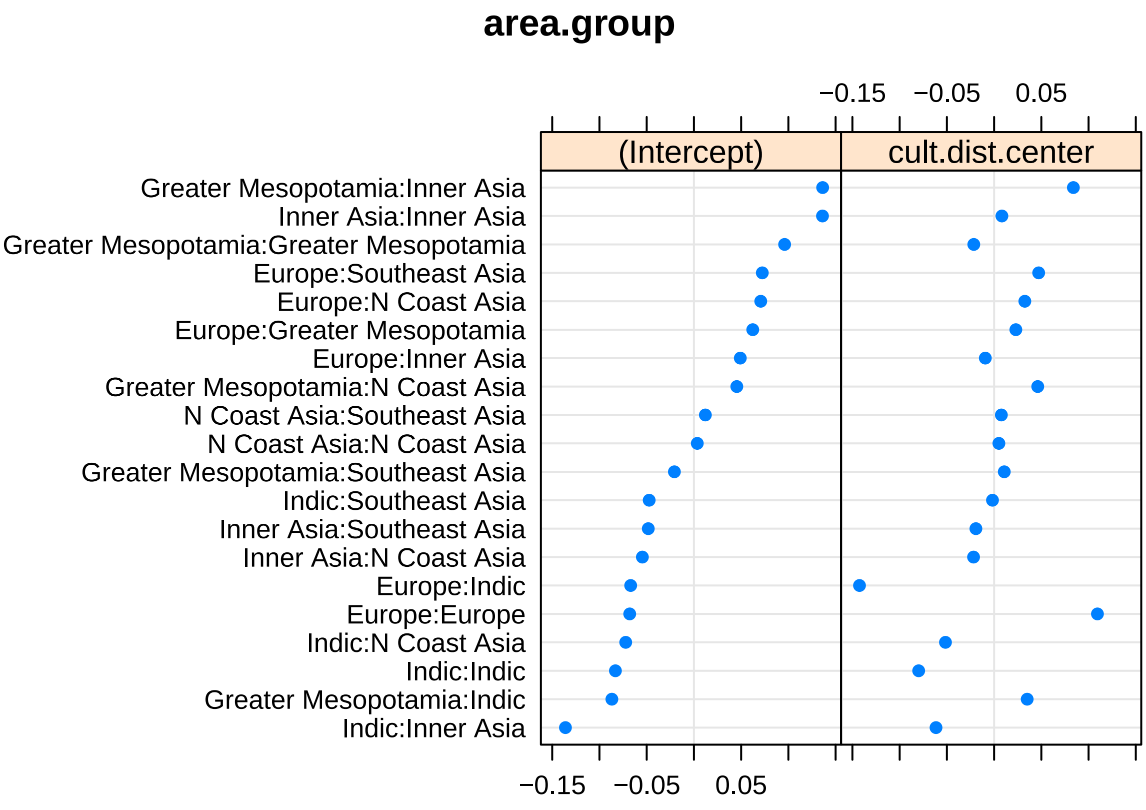
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rho.center ~ comparison_count.center + (1 + cult.dist.center |
##     family.group) + (1 + cult.dist.center | area.group) + (1 +
##     cult.dist.center | imputed_semantic_domain) + cult.dist.center
##    Data: ling.dom
##
## REML criterion at convergence: 12047.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.2076 -0.6016  0.0087  0.5935  3.8742
##
## Random effects:
##  Groups                  Name             Variance Std.Dev. Corr
##  family.group            (Intercept)      0.292589 0.54091
##                          cult.dist.center 0.007767 0.08813  0.52
##  area.group              (Intercept)      0.011297 0.10629
##                          cult.dist.center 0.006081 0.07798  0.36
##  imputed_semantic_domain (Intercept)      0.311704 0.55830
##                          cult.dist.center 0.015925 0.12619  0.82
##  Residual                                 0.435348 0.65981
## Number of obs: 5858, groups:
## family.group, 48; area.group, 20; imputed_semantic_domain, 8
##
## Fixed effects:
##                         Estimate Std. Error t value
## (Intercept)             -0.53935    0.21496  -2.509
## comparison_count.center  0.32818    0.01474  22.263
## cult.dist.center         0.01434    0.05346   0.268
##
## Correlation of Fixed Effects:
##             (Intr) cmpr_.
## cmprsn_cnt.  0.016
## clt.dst.cnt  0.698 -0.011
```

```
dotplot(ranef(mD6))
```

```
## $family.group
```

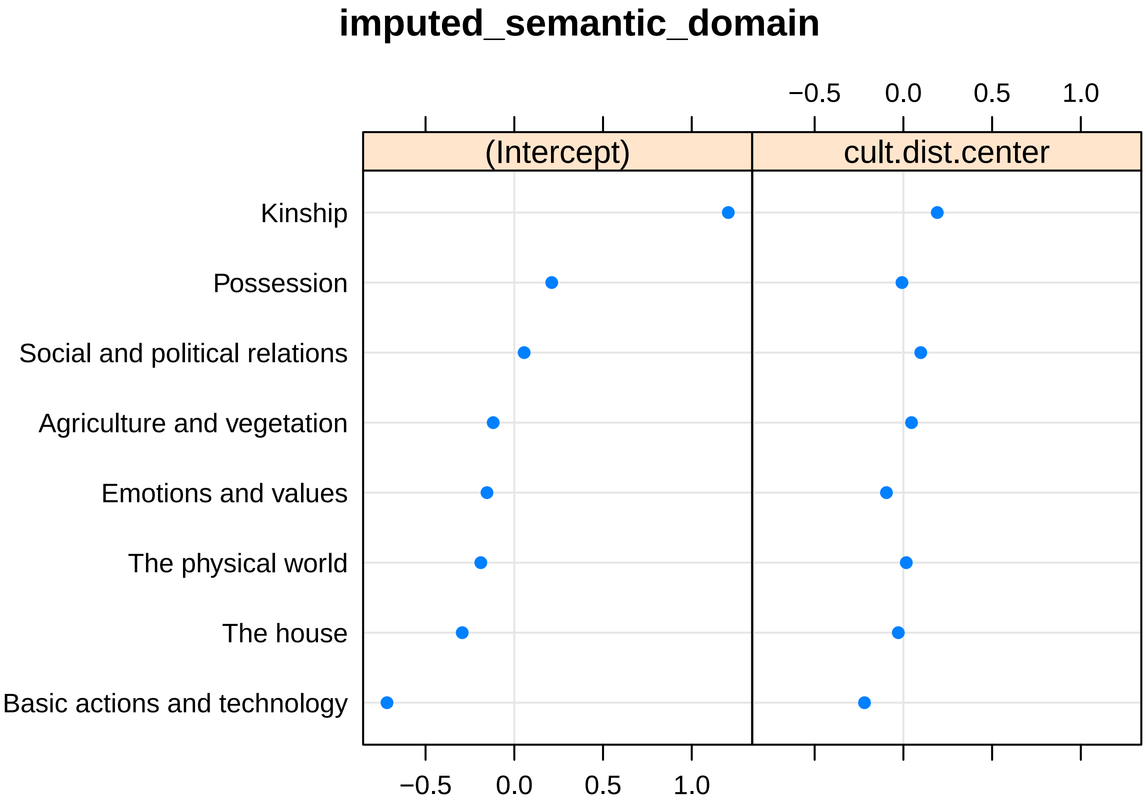

family.group
(Intercept)
cult.dist.center
−1.0
−0.5
0.0
0.5
1.0
Dravidian:Dravidian
Dravidian:Indo−European
Indo−European:Indo−European
Indo−European:Koreanic
Dravidian:Koreanic
Dravidian:Uralic
Indo−European:Uralic
Dravidian:Turkic
Uralic:Uralic
Indo−European:Turkic
Indo−European:Sino−Tibetan
Afro−Asiatic:Indo−European
Afro−Asiatic:Dravidian
Koreanic:Uralic
Turkic:Uralic
Turkic:Turkic
Koreanic:Turkic
Indo−European:Japonic
Koreanic:Sino−Tibetan
Afro−Asiatic:Koreanic
Dravidian:Japonic
Japonic:Sino−Tibetan
Japonic:Koreanic
Dravidian:Sino−Tibetan
Afro−Asiatic:Uralic
Sino−Tibetan:Uralic
Afro−Asiatic:Turkic
Abkhaz−Adyge:Indo−European
Sino−Tibetan:Turkic
Afro−Asiatic:Japonic
Afro−Asiatic:Sino−Tibetan
Japonic:Uralic
Japonic:Turkic
Abkhaz−Adyge:Dravidian
Abkhaz−Adyge:Turkic
Abkhaz−Adyge:Uralic
Abkhaz−Adyge:Koreanic
Dravidian:Mongolic
Mongolic:Uralic
Abkhaz−Adyge:Afro−Asiatic
Koreanic:Mongolic
Abkhaz−Adyge:Japonic
Indo−European:Mongolic
Afro−Asiatic:Mongolic
Japonic:Mongolic
Mongolic:Turkic
Mongolic:Sino−Tibetan
Abkhaz−Adyge:Sino−Tibetan

```
##
## $area.group
```


area.group
(Intercept)
cult.dist.center
−0.15
−0.05
0.05
Greater Mesopotamia:Inner Asia
Inner Asia:Inner Asia
Greater Mesopotamia:Greater Mesopotamia
Europe:Southeast Asia
Europe:N Coast Asia
Europe:Greater Mesopotamia
Europe:Inner Asia
Greater Mesopotamia:N Coast Asia
N Coast Asia:Southeast Asia
N Coast Asia:N Coast Asia
Greater Mesopotamia:Southeast Asia
Indic:Southeast Asia
Inner Asia:Southeast Asia
Inner Asia:N Coast Asia
Europe:Indic
Europe:Europe
Indic:N Coast Asia
Indic:Indic
Greater Mesopotamia:Indic
Indic:Inner Asia

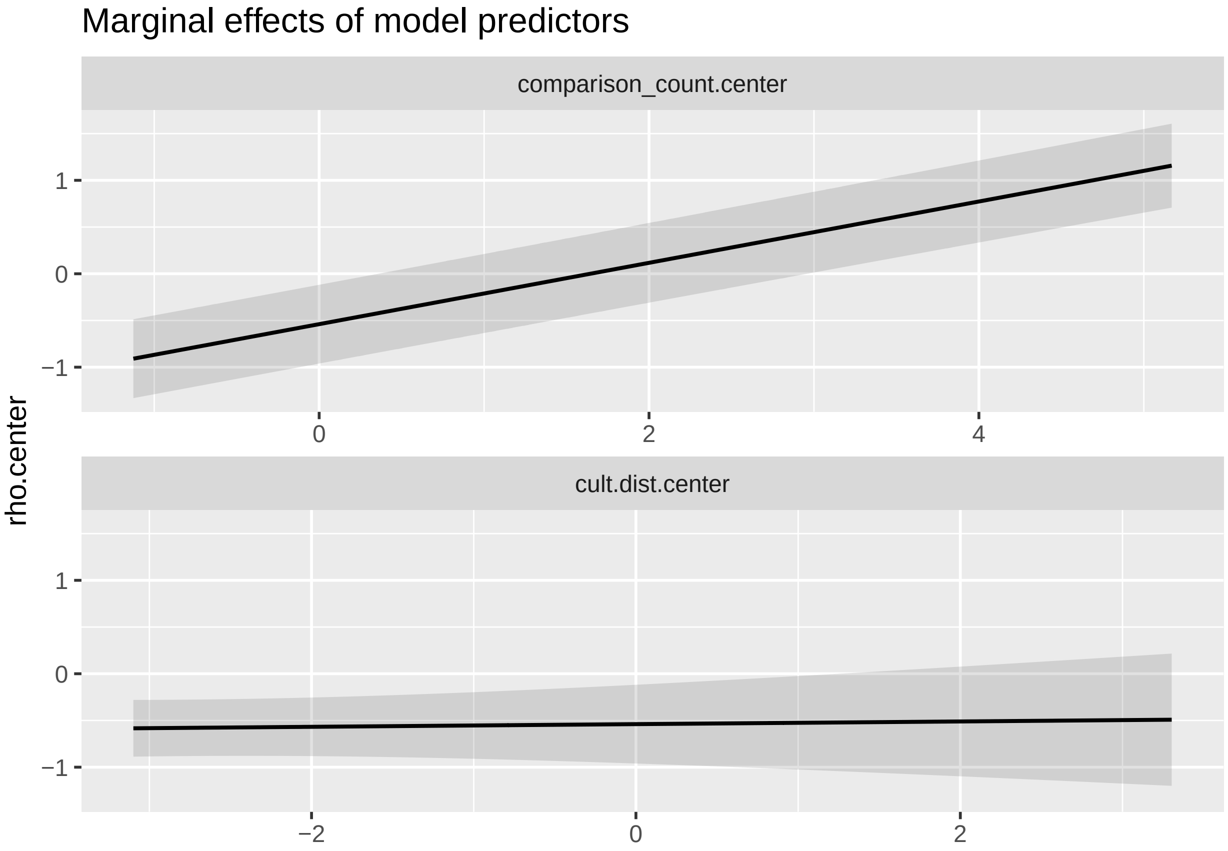
```
##
```

```
## $imputed_semantic_domain
```

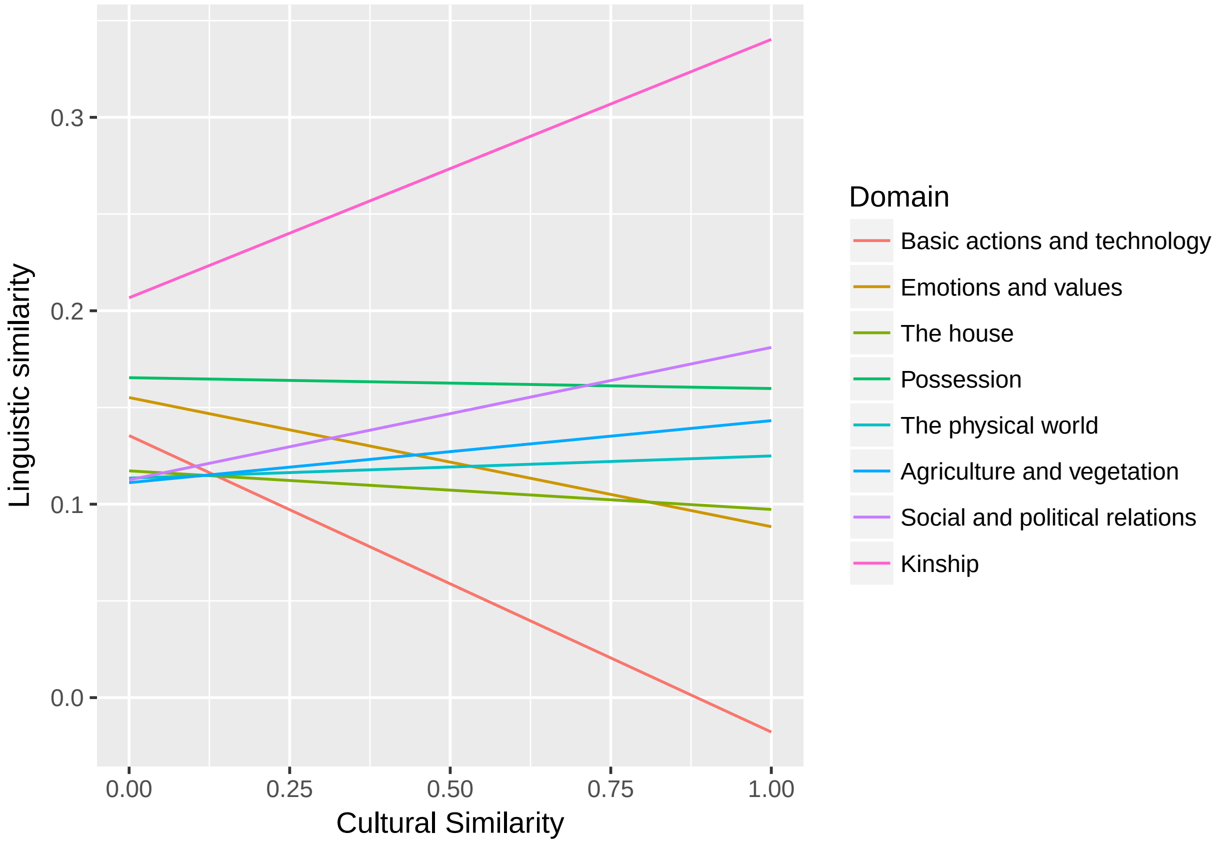


Plot the predicted relationships for each domain. The domains in the legend are sorted by the slope for cultural similarity (greatest negative slope to greatest positive slope):

```
sjp.lmer(mD6,'eff', show.ci = T)
```



```
dom.order = ranef(mD6)$imputed_semantic_domain
dom.order = rownames(dom.order[order(dom.order$cult.dist.center),])
px = sjp.lmer(mD6,'rs.ri', show.ci = T, prnt.plot = F)
pdx = px$plot[[3]]$data
pdx$Domain = factor(pdx$grp, levels = dom.order)
pdx$x = pdx$x *
  attr(ling.dom$cult.dist.center,"scaled:scale") +
  attr(ling.dom$cult.dist.center,"scaled:center")
pdx$y = pdx$y *
  attr(ling.dom$rho.center,"scaled:scale") +
  attr(ling.dom$rho.center,"scaled:center")
ggplot(pdx,
       aes(x,y,colour=Domain)) +
  geom_line() +
  xlab("Cultural Similarity") +
  ylab("Linguistic similarity")
```



Overall slope estimate for each domain:

```
ref.slopes.dom = data.frame(
  domain = rownames(ranef(mD6)$imputed_semantic_domain),
  slope = (fixef(mD6)["cult.dist.center"] +
  ranef(mD6)$imputed_semantic_domain[,2]))
ref.slopes.dom[order(ref.slopes.dom$slope),]
```

```
##                           domain        slope
## 2   Basic actions and technology -0.204840232
## 3            Emotions and values -0.081124800
## 7                      The house -0.014193186
## 5                     Possession  0.006330432
## 8             The physical world  0.030749890
## 1     Agriculture and vegetation  0.060180202
## 6 Social and political relations  0.112275430
## 4                        Kinship  0.205373193
```

Are the relevant D-place features most predictive?

Long format to line up comparisons:

```
ling.dom.wide = ling.dom[,c("l1",'l2',
            'imputed_semantic_domain',
            "local_alignment","cult.dist")]
ling.dom.wide = reshape(ling.dom.wide,
                        idvar = c("l1","l2"),
                        timevar = "imputed_semantic_domain",
                        direction = "wide")
ling.dom.wide = cbind(ling.dom.wide[,1:2],
                      ling.dom.wide[,3:ncol(ling.dom.wide)][
                        order(names(ling.dom.wide[,3:ncol(ling.dom.wide)]))
                      ])
snames = c("Agri","Tech","Emot","Kin","Poss","Soc","Hous","Wrld")
names(ling.dom.wide) = c("l1","l2",
                         paste0("L.",snames),
                         paste0("C.",snames))
```

Raw correlation between each pair of domains

```
compareAllDomains =
  cor(ling.dom.wide[,
      grepl("L\\.",names(ling.dom.wide))],
    ling.dom.wide[,
      grepl("C\\.",names(ling.dom.wide))],
    use="complete.obs")

round(compareAllDomains,2)
```

```
##        C.Agri C.Tech C.Emot C.Kin C.Poss C.Soc C.Hous C.Wrld
## L.Agri   0.47   0.39   0.42  0.41   0.33  0.48   0.34   0.52
## L.Tech   0.26   0.23   0.26  0.34   0.18  0.27   0.21   0.33
## L.Emot   0.27   0.19   0.27  0.42   0.18  0.26   0.23   0.36
## L.Kin    0.28   0.18   0.17  0.27   0.19  0.27   0.19   0.28
## L.Poss   0.44   0.39   0.42  0.52   0.34  0.48   0.38   0.53
## L.Soc    0.51   0.35   0.42  0.51   0.33  0.49   0.39   0.53
## L.Hous   0.30   0.16   0.14  0.21   0.18  0.23   0.18   0.24
## L.Wrld   0.18   0.12   0.08  0.10   0.09  0.15   0.04   0.17
```

Mantel tests

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

Data prep

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.

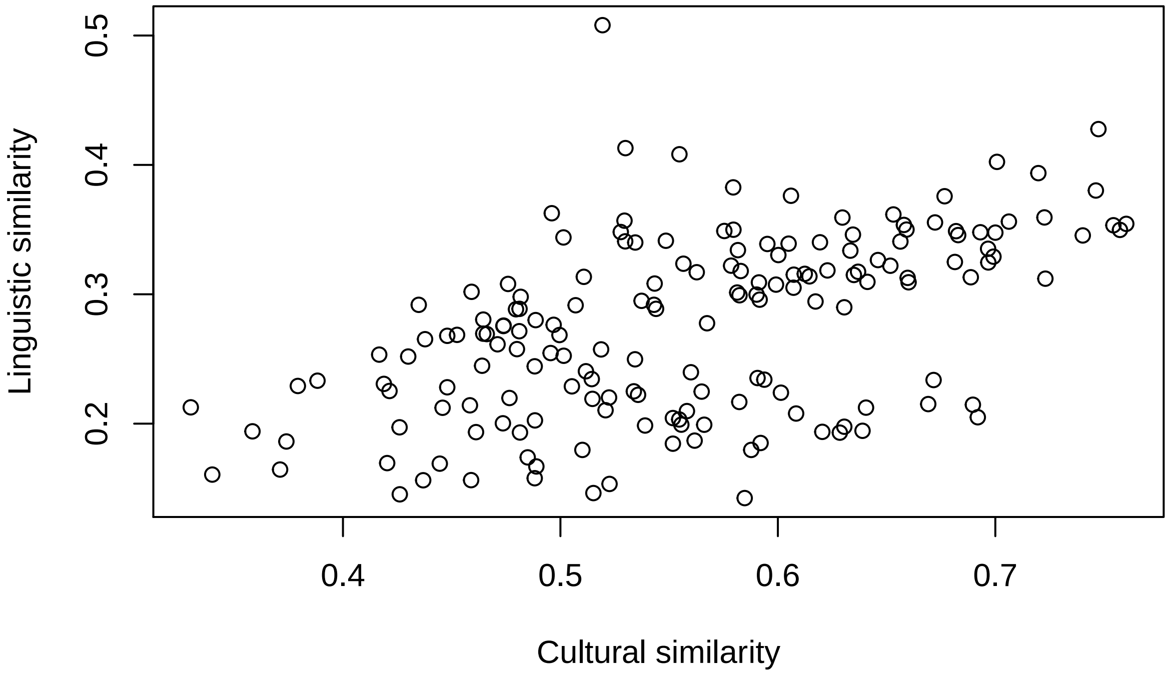
```
grph <- graph.data.frame(ling[,c("l1",'l2','local_alignment')], directed=FALSE)
# add value as a weight attribute
ling.m = get.adjacency(grph, attr="local_alignment", sparse=FALSE)
rownames(ling.m) = l[match(rownames(ling.m),l$iso2),]$Language2
colnames(ling.m) = l[match(colnames(ling.m),l$iso2),]$Language2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

Match the distance matrices

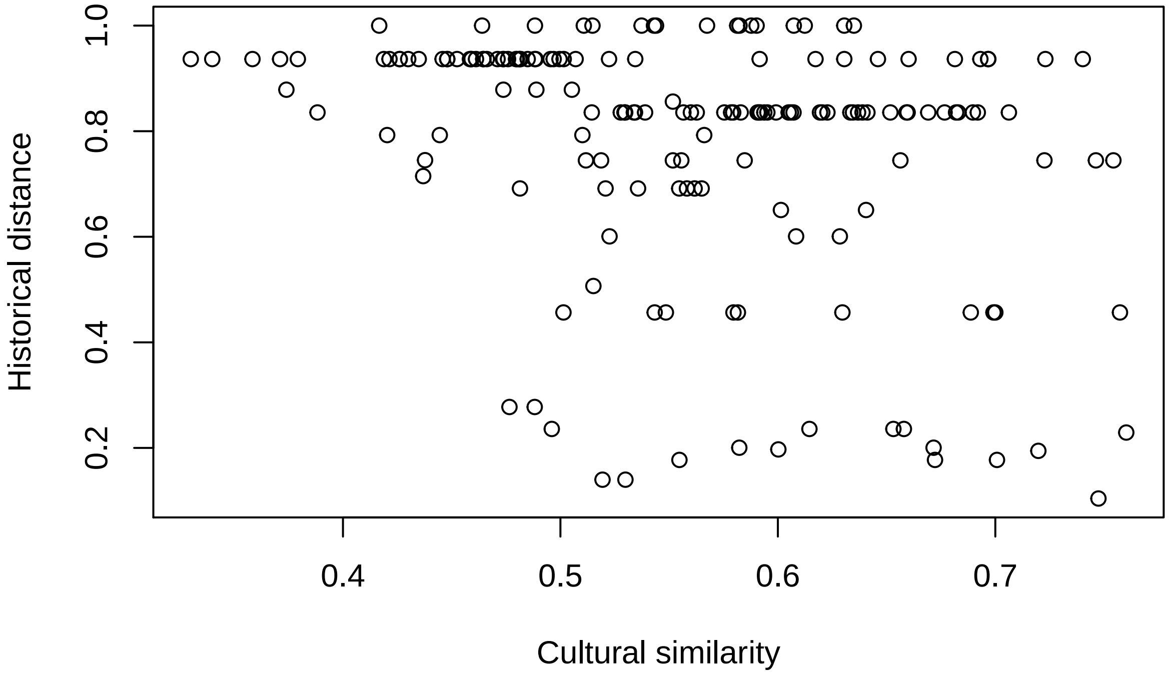
```
in.analysis = intersect(rownames(ling.m),rownames(cult.m))
in.analysis = intersect(in.analysis, rownames(hist.m))
cult.m2 = cult.m[in.analysis,in.analysis]
ling.m2 = ling.m[in.analysis,in.analysis]
hist.m2 = hist.m[in.analysis,in.analysis]
```

Note that there are only 19 languages with data on lingusitic, cultural and historical distance.

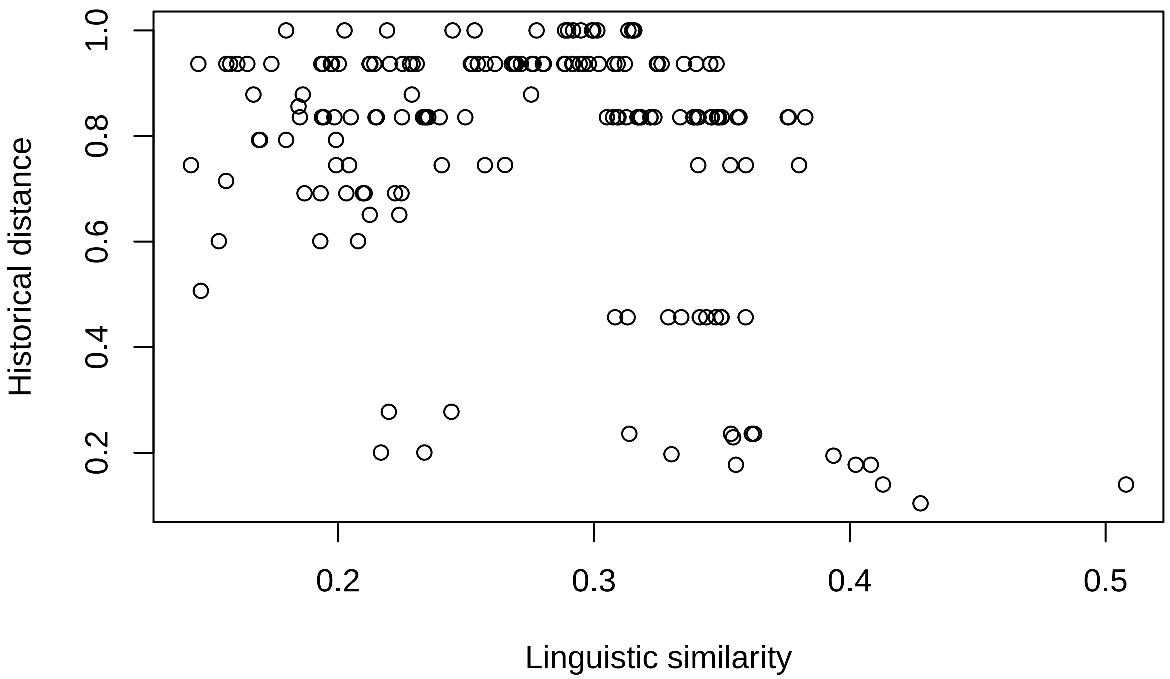
```
plot(as.dist(cult.m2),as.dist(ling.m2),
     xlab="Cultural similarity",
     ylab="Linguistic similarity")
```



```
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")
```



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)
```

Correlation between cultural and linguistic distances:

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

```
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
##  0.5243289  0.0050000  0.9950100  0.0050300  0.3796035  0.6586819
```

Correlation between cultural and historical distances:

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

```
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.3243830  0.9871000  0.0129100  0.0138900 -0.4402666 -0.2385575
```

Correlation between linguistic and historical distances:

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

```
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.3372882  0.9863300  0.0136800  0.0164400 -0.5079832 -0.1785001
```

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%
##  0.4659407  0.0105700  0.9894400  0.0115600  0.3251192  0.6192039
```

Run a mantel test comparing the linguistic similarities to the historical similarities, controlling for cultural similarity:

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

```
##     mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
## -0.20758646  0.92248000  0.07753000  0.14075000 -0.41483917  0.01594166
```

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.