

# Cultural distances: controlling for history

## Introduction

We compare cultural distances between societies with linguistic distances 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

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

## 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_glottol_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("../data/FAIR/semantic_distances_FAIR.csv", stringsAsFactors = F)
```

Combine the linguistic and cultural distances

```

cult$l1.iso2 = 1[match(cult$l1, l$Language2),]$iso2
cult$l2.iso2 = 1[match(cult$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
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$family1 = 1[match(ling$l1, l$iso2),]$family
ling$area1 = 1[match(ling$l1, l$iso2),]$autotyp.area

ling$family2 = 1[match(ling$l2, l$iso2),]$family
ling$area2 = 1[match(ling$l2, l$iso2),]$autotyp.area

ling$family.group = paste(ling$family1, ling$family2)
ling$area.group = paste(ling$area1, ling$area2)

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

```

Each observation is now associated with a language family pair:

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

```

##      l1 l2      rho      family.group
## 1  be es 0.3248551 Indo-European Indo-European
## 2  bg be 0.3464910 Indo-European Indo-European
## 4  bg cs 0.4091899 Indo-European Indo-European
## 7  bg en 0.3145696 Indo-European Indo-European
## 8  bg es 0.3686999 Indo-European Indo-European
## 10 bg fr 0.3742889 Indo-European Indo-European

```

And the same is true for area:

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

```

##      l1 l2      rho      area.group
## 375 ru he 0.2888911 Inner Asia Greater Mesopotamia
## 377 ru ja 0.2721048      Inner Asia N Coast Asia
## 379 ru kk 0.2931095      Inner Asia Inner Asia
## 380 ru la 0.2125645      Inner Asia Europe
## 381 ru lt 0.3635630      Inner Asia Inner Asia
## 382 ru nl 0.3626820      Inner Asia Europe

```

## LMER models

Mixed effects model, predicting linguistic distances from cultural distances, with random intercept for family and area and random slope for cultural distance for family and area.

We compare a null model to a model with a fixed effect for cultural distance, with random intercepts for family and area, and random slopes for cultural distance by both.

```
m0 = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
m1 = lmer(
  rho.center ~ 1 +
    cult.dist.center +
    (1 + cult.dist.center | family.group) +
    (1 + cult.dist.center | area.group),
  data = ling
)
anova(m0,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)
## m1: rho.center ~ 1 + cult.dist.center + (1 + cult.dist.center | family.group) +
## m1:      (1 + cult.dist.center | area.group)
##      Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m0  8 289.75 311.05 -136.87  273.75
## m1  9 286.19 310.16 -134.09  268.19 5.5577      1      0.0184 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cultural distance is significantly correlated with linguistic distance. Here are the model estimates:

```
summary(m1)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## rho.center ~ 1 + cult.dist.center + (1 + cult.dist.center | family.group) +
##      (1 + cult.dist.center | area.group)
##      Data: ling
##
## REML criterion at convergence: 273.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.4780 -0.4697  0.1482  0.5628  2.6477
##
## Random effects:
##      Groups      Name              Variance Std.Dev. Corr
## area.group  (Intercept)          0.023659 0.15382
##              cult.dist.center 0.031171 0.17655  1.00
```

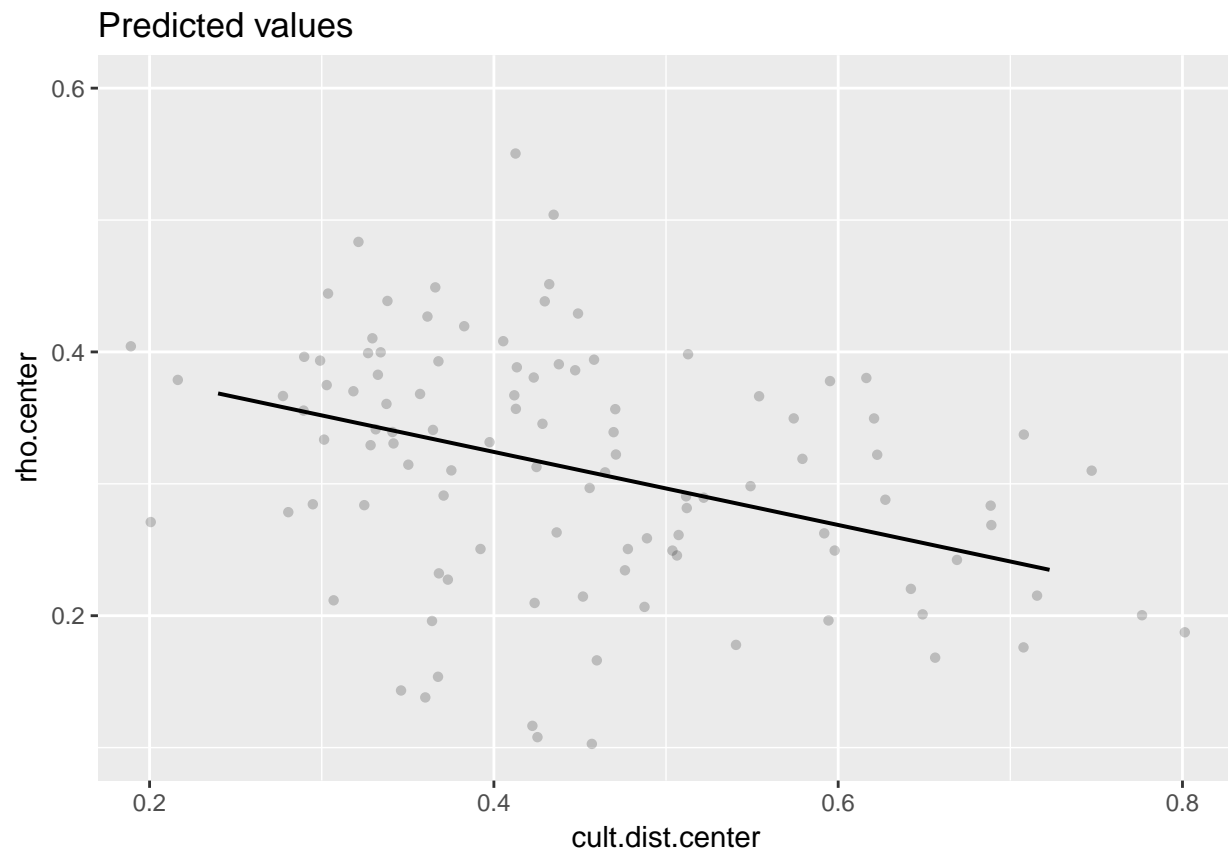
```
## family.group (Intercept)      0.031052 0.17622
## cult.dist.center 0.007347 0.08572 -1.00
## Residual 0.697762 0.83532
## Number of obs: 106, groups: area.group, 15; family.group, 14
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) -0.1149 0.1371 -0.838
## cult.dist.center -0.5013 0.1275 -3.932
##
## Correlation of Fixed Effects:
## (Intr)
## clt.dst.cnt -0.234
```

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$res.p.y = gx$plot$data$res.p.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$plot + coord_cartesian(ylim=c(0.1,0.6),
  xlim=c(0.2,0.8))
```

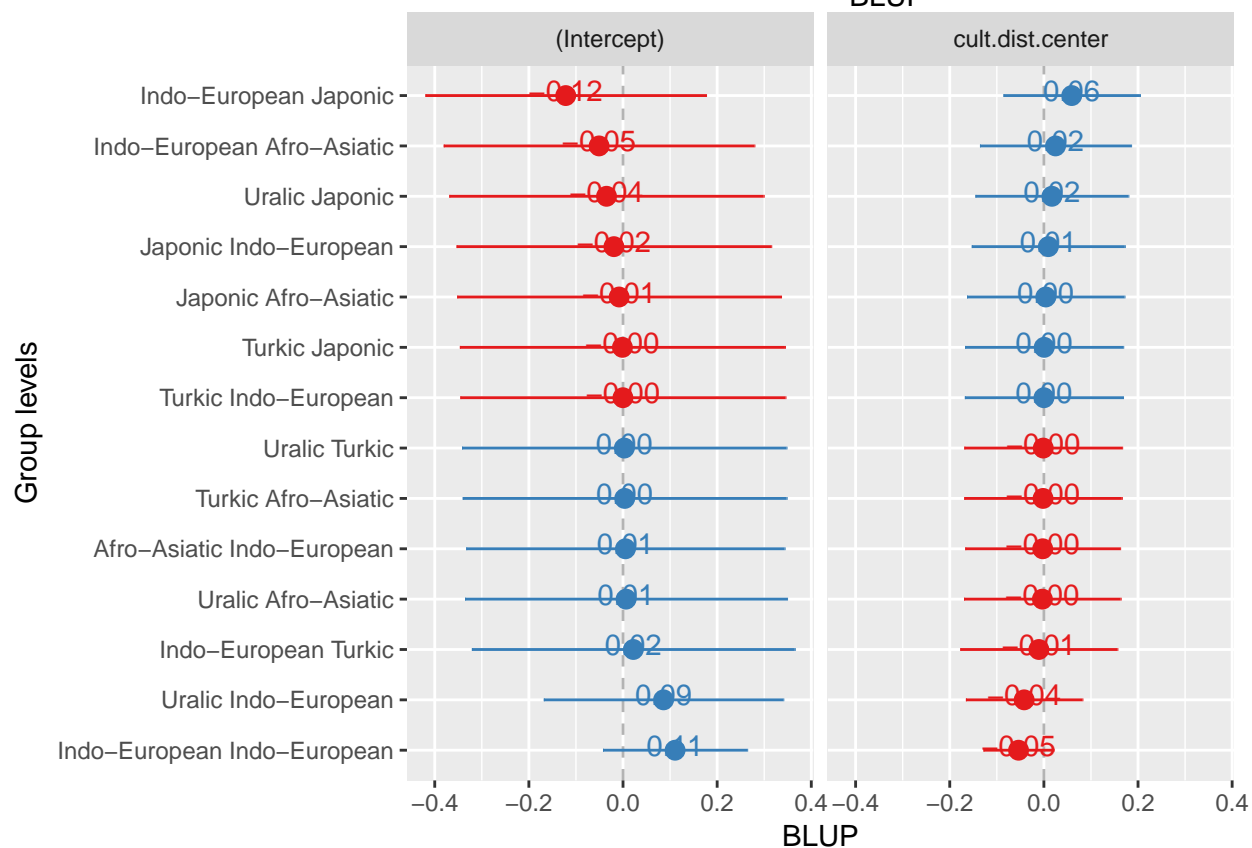
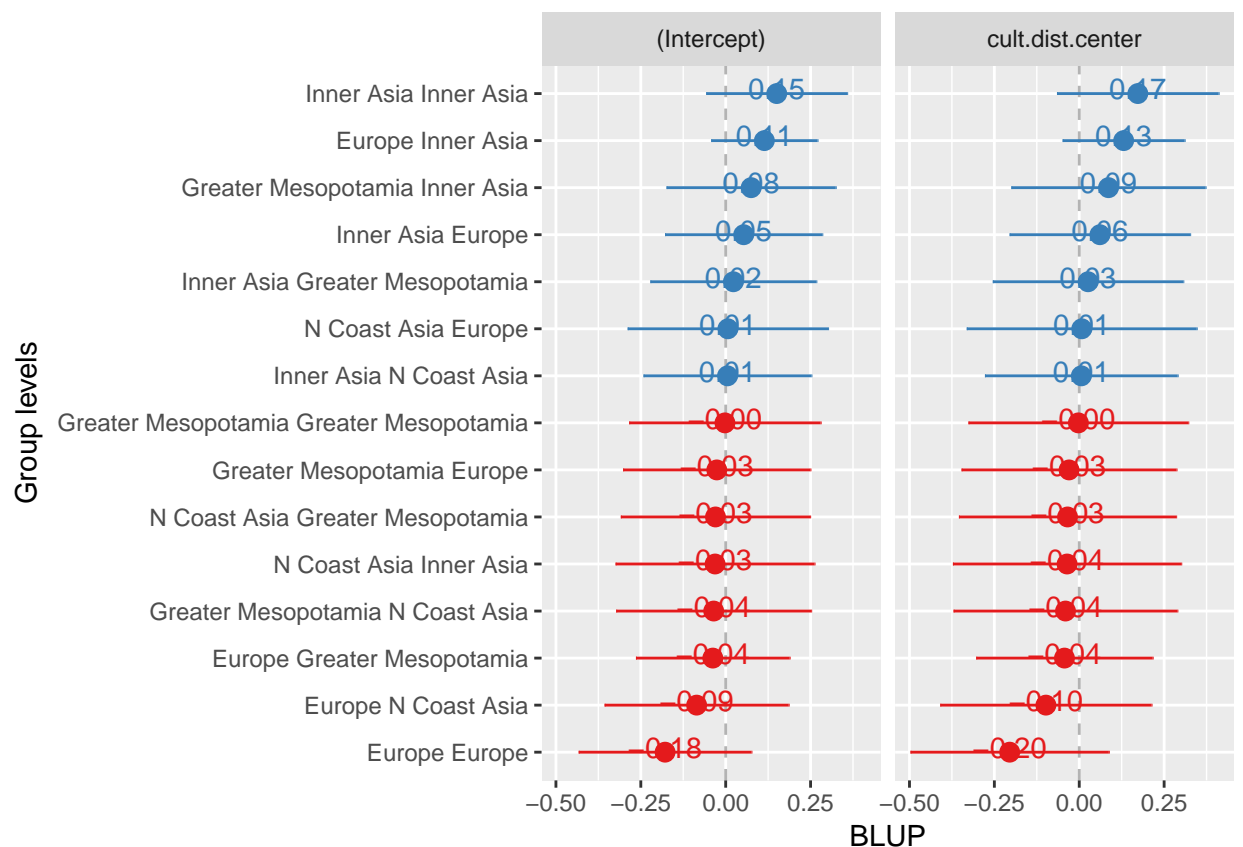


Plot the random effects. There seems to be more variation by area:

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

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

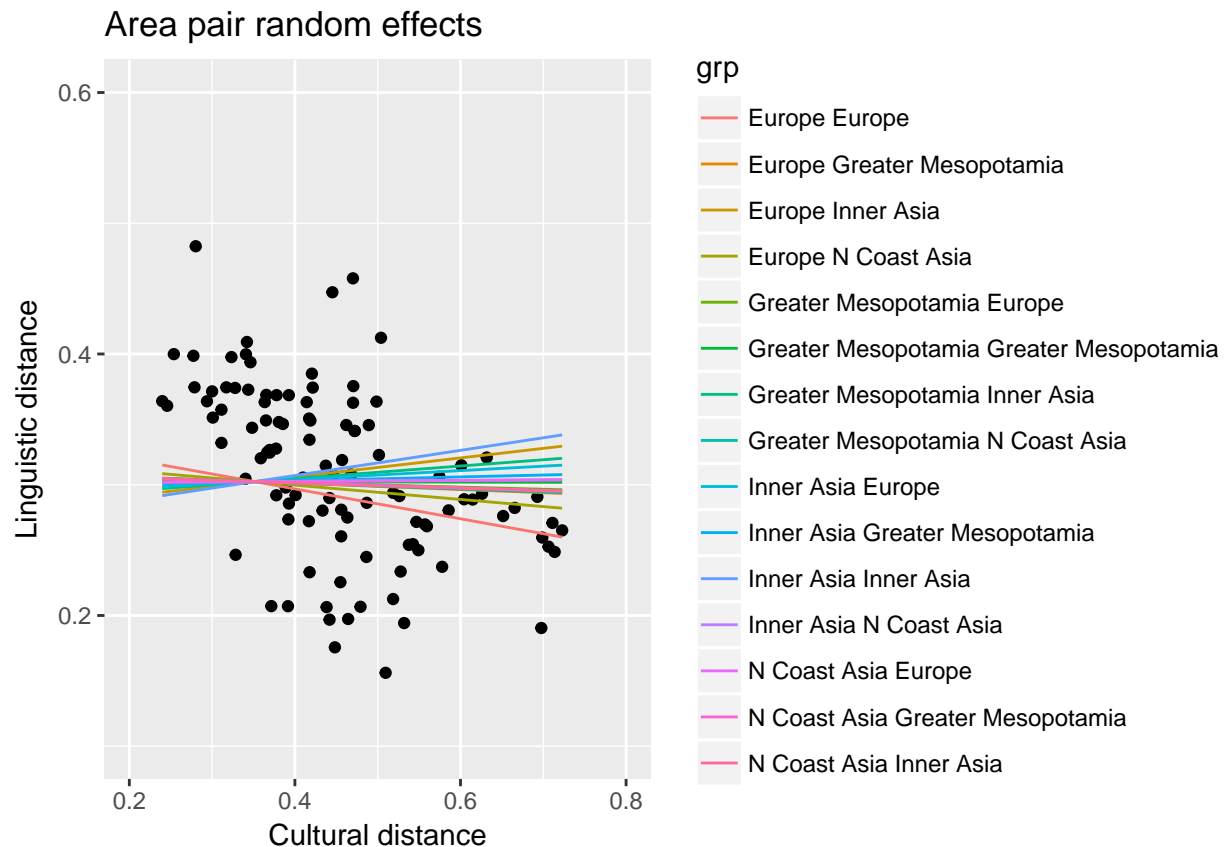
```
## Plotting random effects...
```



```

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(rho))) +
  geom_line(mapping = aes(colour=grp)) +
  xlab("Cultural distance") +
  ylab("Linguistic distance") +
  ggtitle("Area pair random effects") +
  coord_cartesian(ylim=c(0.1, 0.6),
                 xlim=c(0.2, 0.8))

```

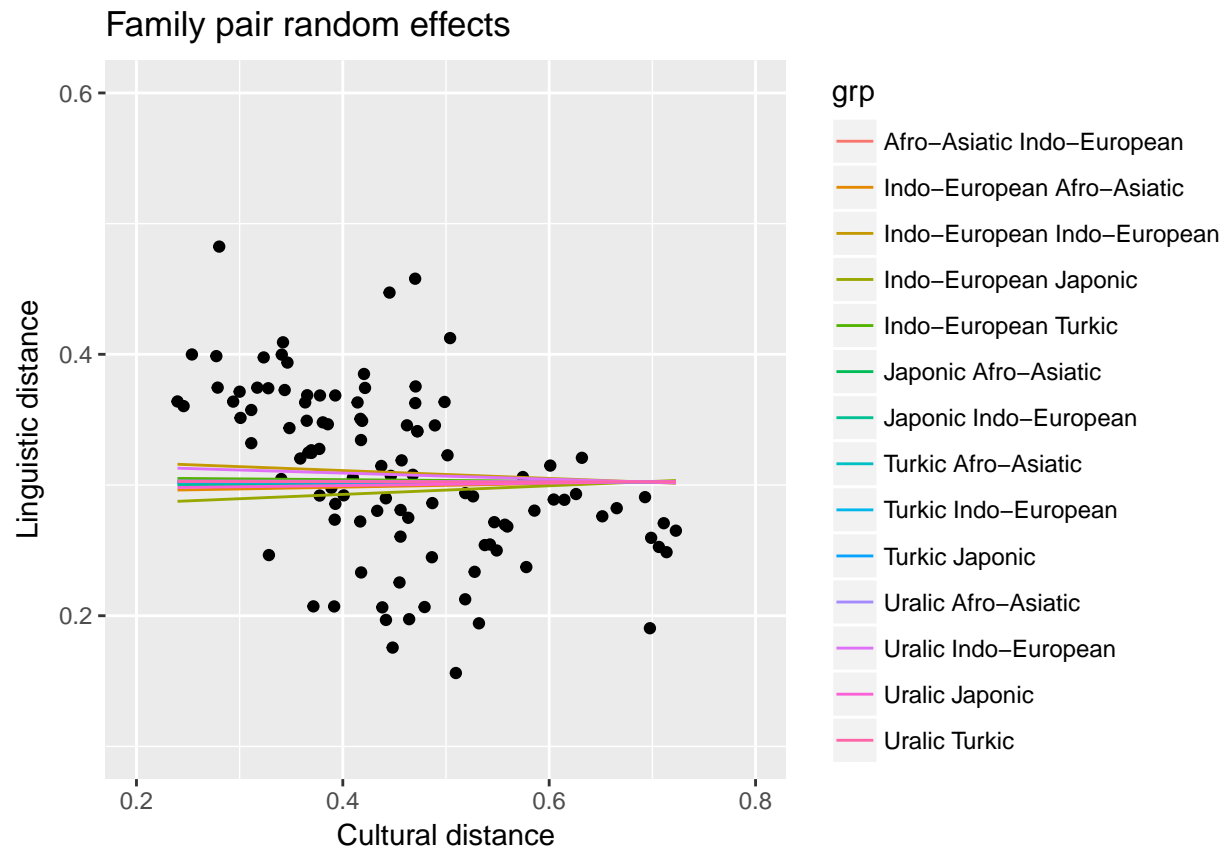


```

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(rho))) +

```

```
geom_line(mapping = aes(colour=grp)) +
  xlab("Cultural distance")+
  ylab("Linguistic distance") +
  ggtitle("Family pair random effects") +
  coord_cartesian(ylim=c(0.1,0.6),
                  xlim=c(0.2,0.8))
```





## Tests within domains

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

```
ling.dom = read.csv(
  "../results/EA_distances/All_Domains_with_ling.csv",
  stringsAsFactors = F)

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

ling.dom$family1 = l[match(ling.dom$l1, l$iso2),]$family
ling.dom$family2 = l[match(ling.dom$l2, l$iso2),]$family
ling.dom$family.group = paste(ling.dom$family1,
                              ling.dom$family2)

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

Center the data:

```
ling.dom$cult.dist.center = scale(ling.dom$cult.dist)
ling.dom$rho.center = scale(ling.dom$rho)
```

## LMER models

Test whether random slopes are warranted for family:

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

```
## mD0:      imputed_semantic_domain)
## mD1: rho.center ~ 1 + (1 | family.group) + (0 + cult.dist.center |
## mD1:      family.group) + (1 | area.group) + (1 | imputed_semantic_domain)
## mD2: rho.center ~ 1 + (1 + cult.dist.center | family.group) + (1 |
## mD2:      area.group) + (1 | imputed_semantic_domain)
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## mD0  5 2784.0 2808.6 -1387.0  2774.0
## mD1  6 2774.0 2803.6 -1381.0  2762.0 11.9989      1 0.0005323 ***
## mD2  7 2775.6 2810.1 -1380.8  2761.6  0.4134      1 0.5202648
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Random slopes for family improves the fit of the model, but adding the correlation coefficient does not.

Test the same for area:

```
mD3 = lmer(
  rho.center ~ 1 +
    (1 | family.group) +
    (0 + cult.dist.center | family.group) +
    (1 | area.group) +
    (0 + cult.dist.center | area.group) +
    (1 | imputed_semantic_domain),
  data = ling.dom)
anova(mD1,mD3)
```

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

```
## Data: ling.dom
## Models:
## mD1: rho.center ~ 1 + (1 | family.group) + (0 + cult.dist.center |
## mD1:      family.group) + (1 | area.group) + (1 | imputed_semantic_domain)
## mD3: rho.center ~ 1 + (1 | family.group) + (0 + cult.dist.center |
## mD3:      family.group) + (1 | area.group) + (0 + cult.dist.center |
## mD3:      area.group) + (1 | imputed_semantic_domain)
##      Df      AIC      BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## mD1  6 2774.0 2803.6 -1381.0  2762.0
## mD3  7 2775.8 2810.3 -1380.9  2761.8  0.1423      1      0.706
```

Random slopes for area do not improve the fit of the model.

```
mD4 = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center | family.group) +
    (1 | area.group) +
    (1 | imputed_semantic_domain),
  data = ling.dom)
```

And, in fact, area random intercepts are all set to 0:

```
ranef(mD4)$area.group
```

```
##                                     (Intercept)
## Europe Europe                                0
## Europe Greater Mesopotamia                  0
## Europe Inner Asia                           0
## Europe N Coast Asia                         0
## Greater Mesopotamia Europe                   0
```

```
## Greater Mesopotamia Greater Mesopotamia      0
## Greater Mesopotamia Inner Asia                0
## Greater Mesopotamia N Coast Asia              0
## Inner Asia Europe                             0
## Inner Asia Greater Mesopotamia               0
## Inner Asia Inner Asia                        0
## Inner Asia N Coast Asia                      0
## N Coast Asia Europe                          0
## N Coast Asia Greater Mesopotamia             0
## N Coast Asia Inner Asia                      0
```

So we'll take them out.

Now we test the main effect of cultural distance:

```
mD5 = lmer(
  rho.center ~ 1 +
    (1 + cult.dist.center || family.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 + ((1 | family.group) + (0 + cult.dist.center |
## mD5:      family.group)) + (1 + cult.dist.center | imputed_semantic_domain)
## mD6: rho.center ~ (1 | family.group) + (0 + cult.dist.center | family.group) +
## mD6:      (1 + cult.dist.center | imputed_semantic_domain) + cult.dist.center
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## mD5   7 2758.4 2792.9 -1372.2  2744.4
## mD6   8 2757.5 2796.9 -1370.7  2741.5 2.9668      1  0.08499 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Summary of the final model, with random effects plot:

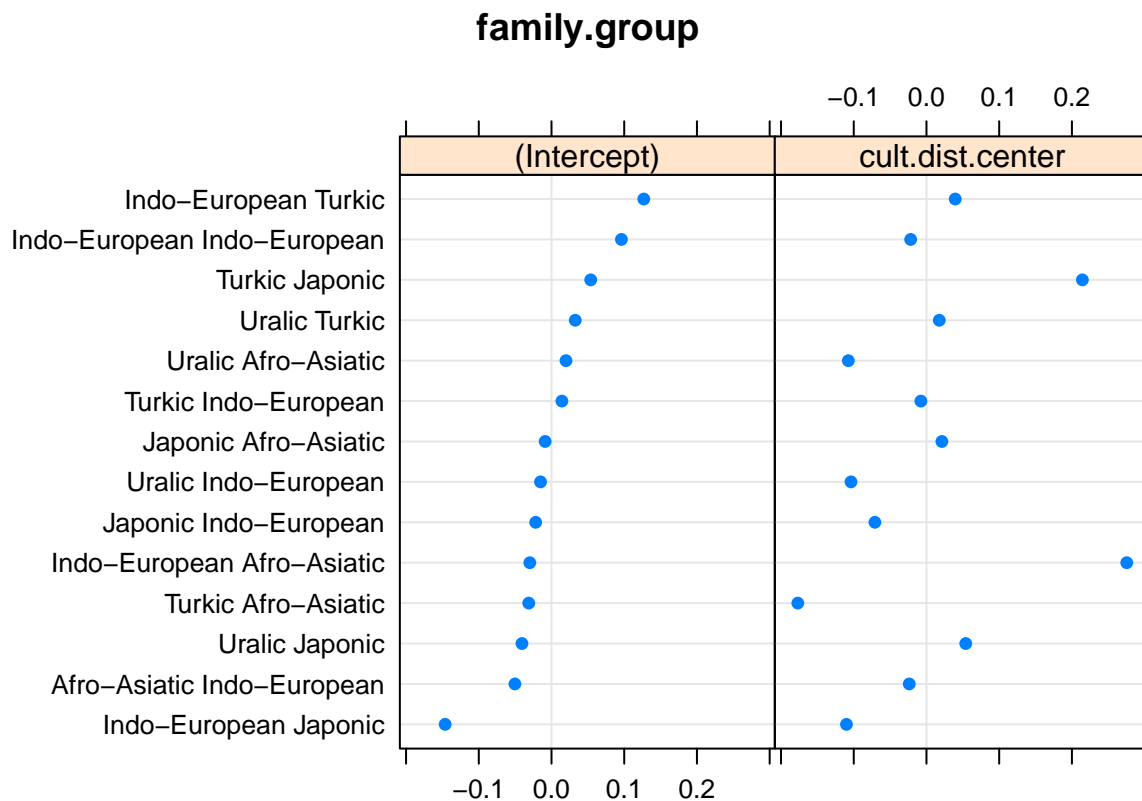
```
summary(mD6)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## rho.center ~ (1 | family.group) + (0 + cult.dist.center | family.group) +
##      (1 + cult.dist.center | imputed_semantic_domain) + cult.dist.center
##      Data: ling.dom
##
## REML criterion at convergence: 2746.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -6.6462 -0.4380  0.0606  0.4753  4.1930
##
## Random effects:
##      Groups                Name                Variance Std.Dev. Corr
## family.group              (Intercept)           0.01539  0.1240
## family.group.1            cult.dist.center 0.03347  0.1830
```

```
## imputed_semantic_domain (Intercept)      0.13163  0.3628
##                                           cult.dist.center 0.03338  0.1827  -0.36
## Residual                                0.81382  0.9021
## Number of obs: 1020, groups:
## family.group, 14; imputed_semantic_domain, 10
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   -0.10153    0.13013  -0.780
## cult.dist.center -0.16590    0.09519  -1.743
##
## Correlation of Fixed Effects:
##              (Intr)
## clt.dst.cnt -0.273
```

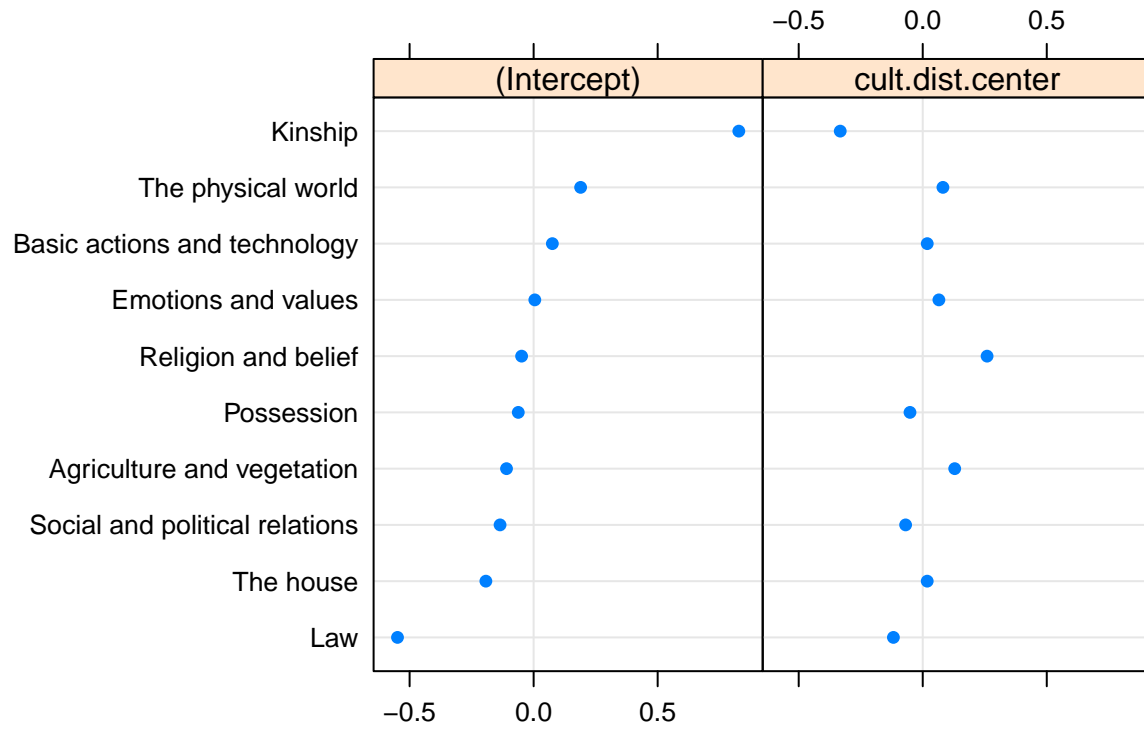
```
dotplot(ranef(mD6))
```

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



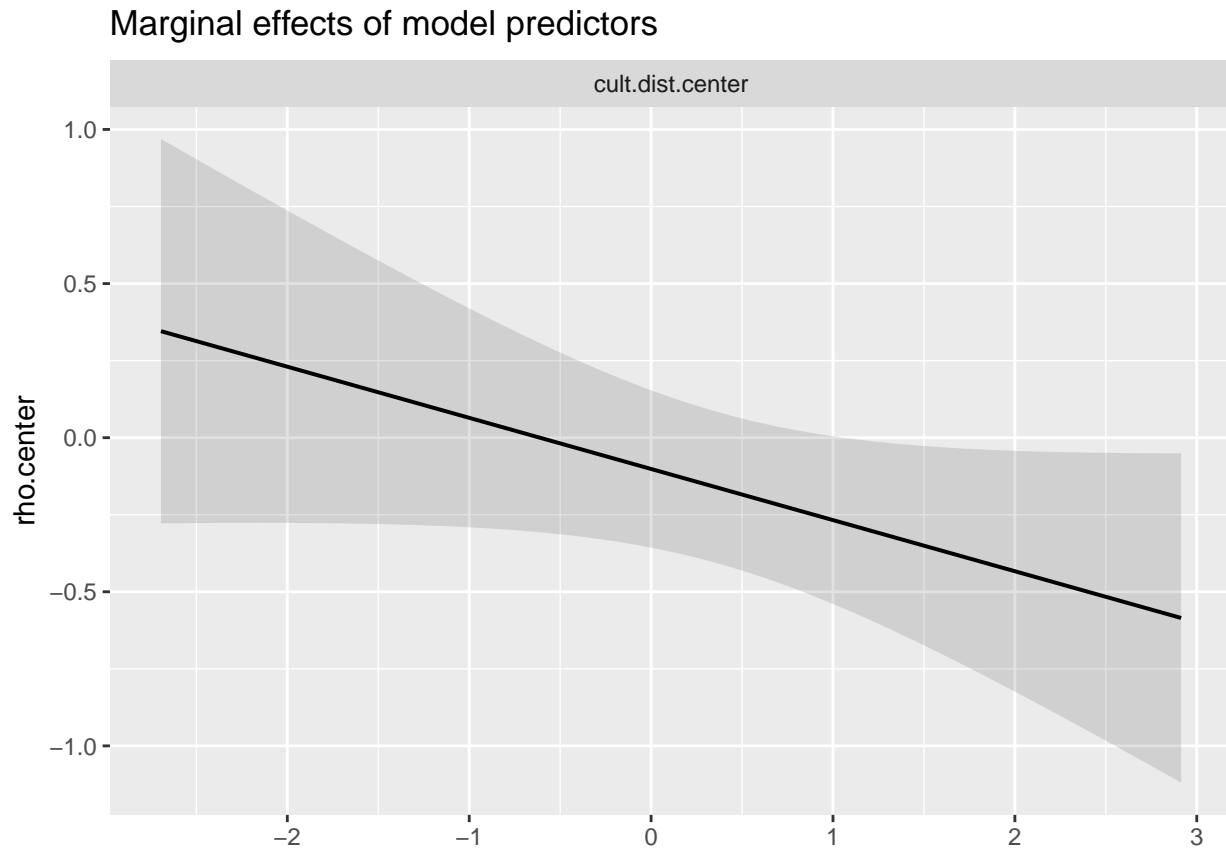
```
##
## $imputed_semantic_domain
```

## imputed\_semantic\_domain

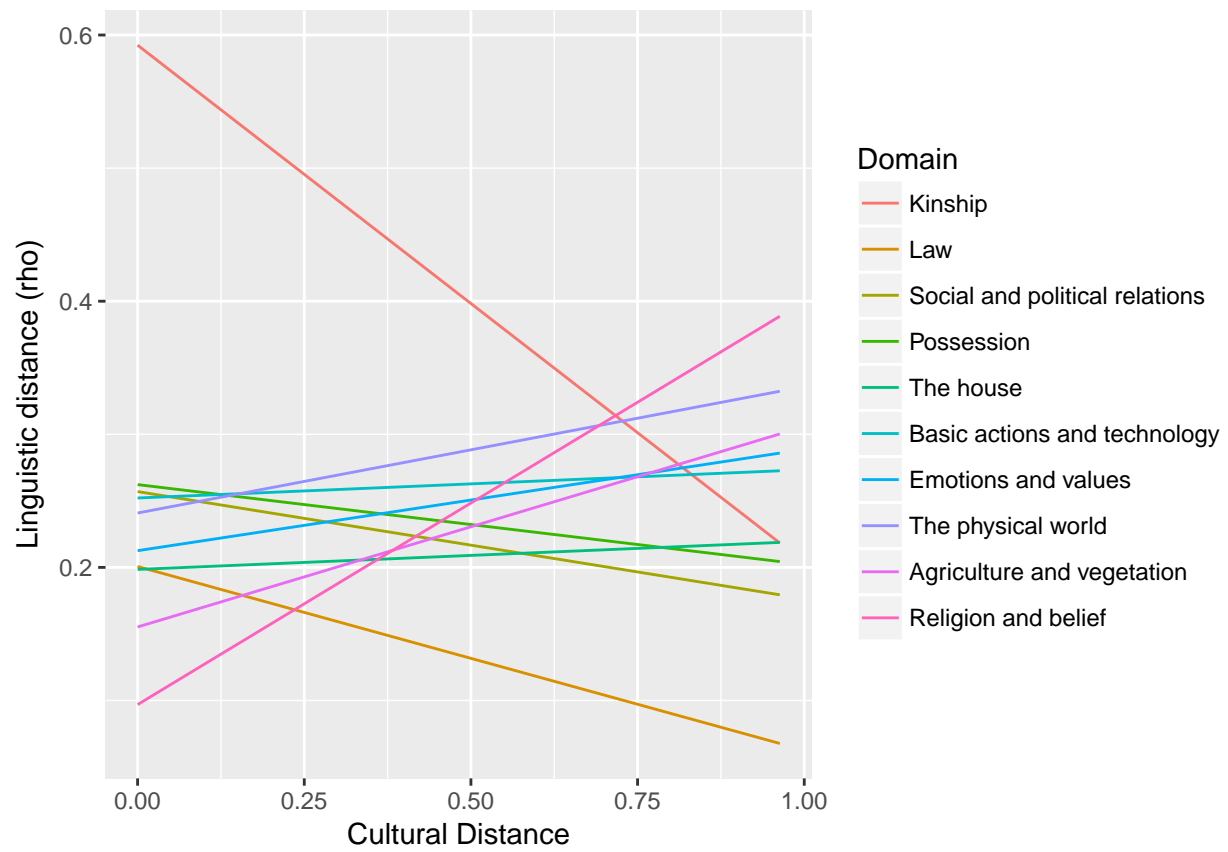


Plot the predicted relationships for each domain. The domains in the legend are sorted by the slope for cultural distance (greatest negative slope to greatest positive slope):

```
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$plot[[2]]$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 Distance") +
  ylab("Linguistic distance (rho)")
```



“Religion and belief” seems to be working differently from the other domains.

## 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 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)]
```

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

```
grph <- graph.data.frame(ling[,c("l1", "l2", "rho")], directed=FALSE)
# add value as a weight attribute
ling.m = get.adjacency(grph, attr="rho", sparse=FALSE)
rownames(ling.m) = 1[match(rownames(ling.m), l$iso2),]$Language2
colnames(ling.m) = 1[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 11 languages with data on linguistic, cultural and historical distance.

### Tests

Simple correlation without control for history:

```
ecodist::mantel(as.dist(cult.m2) ~
               as.dist(ling.m2),
               nperm = 10000)
```

```
##      mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.3361506  0.8785000  0.1216000  0.1700000 -0.6563415 -0.1146983
```

Run a mantel test comparing the linguistic distances to the cultural distances, controlling for the historical distance between languages:



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

```
##      mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%  
## -0.3146355  0.8633000  0.1368000  0.2071000 -0.6487355 -0.0319846
```

## 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.