Rate of change for Kinship words

```r
library(dplyr)
library(lme4)
library(ggplot2)
```

Read in data:

```r
d = read.csv("../data/FAIR/nel-wiki-k100-alignments-merged-long.csv",
             encoding = "UTF-8",fileEncoding = "UTF-8",stringsAsFactors = F)
# Data on languages
l = read.csv("../data/FAIR_langauges_glotto_xdid.csv",stringsAsFactors = F)

# Read in kinship rate of change data
r = read.csv("../data/RaczPassmoreSheardJordan_2019/supp-data-si.csv",
             encoding = "UTF-8",fileEncoding = "UTF-8",stringsAsFactors = F)
# Edit language isos
r$language.iso = l[match(r$language,l$Language),]$iso2
r[r$language=="Norwegian",]$language.iso = "no"
r[r$language=="Ossetic",]$language.iso = "os"
# copy
rkin = r
```

Merge the data:

```r
rkin2 = left_join(d,rkin,by = c("Word_Form_l1"="word","l1"="language.iso"))
names(rkin) = paste0(names(rkin),".l2")
rkin2 = left_join(rkin2,rkin,by = c("Word_Form_l2"="word.l2","l2"="language.iso.l2"))
```

Exclude missing data:

```r
rkin2 = rkin2[!is.na(rkin2$mean.roc),]
rkin2 = rkin2[!is.na(rkin2$mean.roc.l2),]
```

Calculate difference in log frequency per million words:

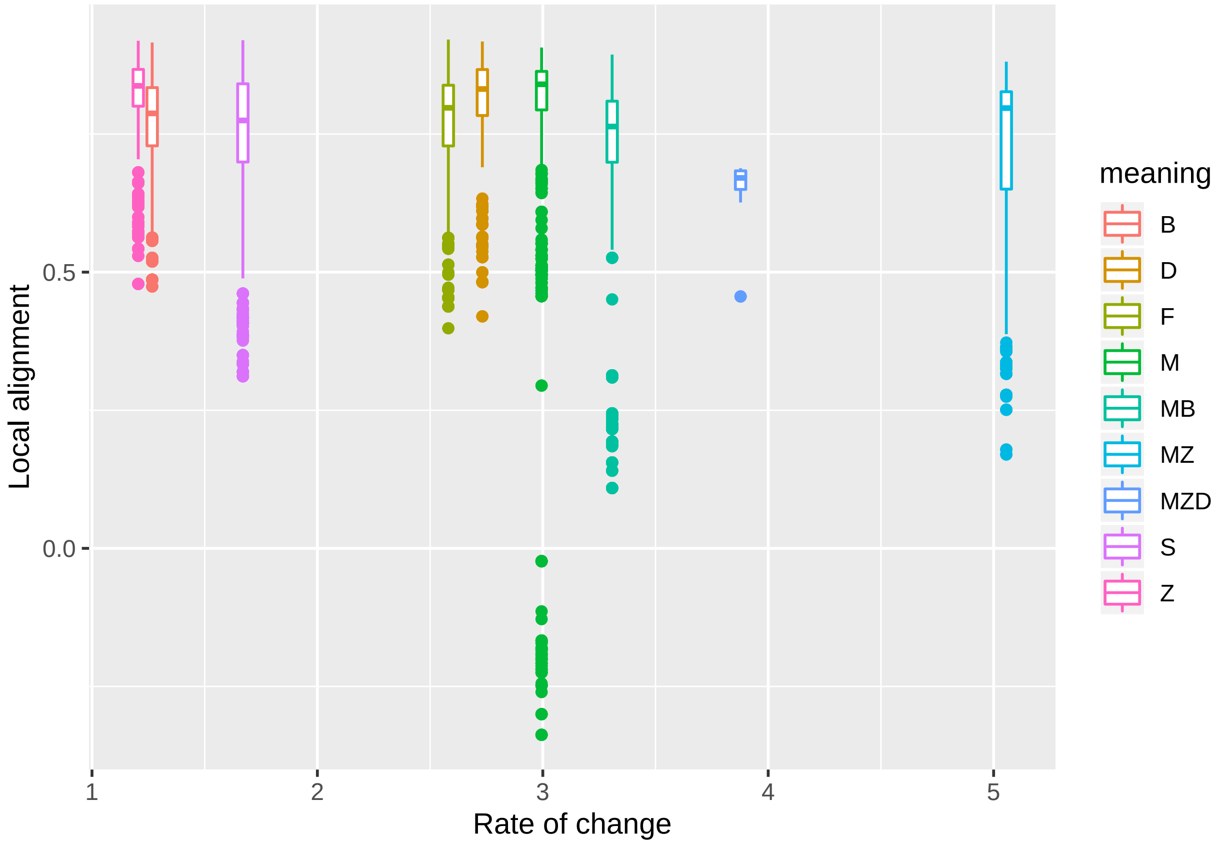
```r
rkin2$fpm.l1 = rkin2$word.count / (rkin2$corpora.size/1000000)
rkin2$fpm.l2 = rkin2$word.count.l2 / (rkin2$corpora.size.l2/1000000)
rkin2$freq_diff = log10(abs(rkin2$fpm.l1 - rkin2$fpm.l2))
```

Filter variables:

```r
rkin3 = rkin2[,c("l1","l2","meaning","Word_Form_l1","Word_Form_l2",
                 "local_alignment","Concept_ID","fpm.l1","fpm.l2",
                 "mean.roc","mean.roc.l2","lingpy.cognate","expert.cognate",
                 "glottocode","glottocode.l2","freq_diff")]
```

Plot data:

```r
ggplot(rkin3, aes(y=local_alignment,x=mean.roc,color=meaning)) +
  geom_boxplot() +xlab("Rate of change") + ylab("Local alignment")
```



Scale and center data

```
rkin3$mean.roc.scaled = scale(rkin3$mean.roc)
rkin3$local_alignment.scaled = scale(rkin3$local_alignment)
rkin3$meaning = factor(rkin3$meaning,
                       levels=c("Z","B","D","S","F","M","MZ","MB","MZD"),
                       labels = c("Sister","Brother","Daughter","Son",
                             "Father","Mother","Aunt","Uncle","Niece"))
```

Variable for language pair:

```
rkin3$langPair = apply(rkin3[,c("l1","l2")],1,function(X){
  paste(sort(X),collapse="-")
})
```

Predit local alignment by rate of change, with random intercepts for each cognate within each meaning, and random intercepts for l1 and l2. Note that we're predicting local alignment from rate of change (not the other way around). This is because rate of change is unique to a particular cognate within a particular meaning.

```
m0 = lmer(local_alignment.scaled ~
            (1|lingpy.cognate/meaning) +
            (1|langPair),
          data=rkin3,
          control = lmerControl(optimizer = "bobyqa"))
m1 = update(m0, ~.+mean.roc.scaled)

anova(m0,m1)
```

```
## refitting model(s) with ML (instead of REML)
## Data: rkin3
## Models:
## m0: local_alignment.scaled ~ (1 | lingpy.cognate/meaning) + (1 |
## m0:     langPair)
## m1: local_alignment.scaled ~ (1 | lingpy.cognate/meaning) + (1 |
## m1:     langPair) + mean.roc.scaled
##    Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0  5 20445 20482 -10217    20435
## m1  6 20436 20481 -10212    20424 10.222      1   0.001388 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(m1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: local_alignment.scaled ~ (1 | lingpy.cognate/meaning) + (1 |
##     langPair) + mean.roc.scaled
##    Data: rkin3
## Control: lmerControl(optimizer = "bobyqa")
##
## REML criterion at convergence: 20431.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -13.4490  -0.2459   0.0084   0.2600   4.0999
##
## Random effects:
##  Groups                 Name        Variance Std.Dev.
##  langPair               (Intercept) 0.7473   0.8644
##  lingpy.cognate         (Intercept) 0.2865   0.5353
##  meaning:lingpy.cognate (Intercept) 0.1086   0.3295
##  Residual                           0.2457   0.4957
## Number of obs: 13061, groups:
## langPair, 275; lingpy.cognate, 98; meaning:lingpy.cognate, 98
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)     -0.48572    0.08586  -5.657
## mean.roc.scaled -0.19683    0.06037  -3.260
##
## Correlation of Fixed Effects:
##             (Intr)
## men.rc.scld -0.102
```

Same as above, but also control for frequency difference (part of speech is the same).

```r
rkin3Freq = rkin3[!is.na(rkin3$freq_diff),]
rkin3Freq$freq_diff.scaled = scale(rkin3Freq$freq_diff)
m0.freq = lmer(local_alignment.scaled ~ freq_diff.scaled +
            (1|lingpy.cognate/meaning) +
            (1|langPair),
          data=rkin3Freq,
          control = lmerControl(optimizer = "bobyqa"))
m1.freq = update(m0.freq, ~.+mean.roc.scaled)

anova(m0.freq,m1.freq)
```

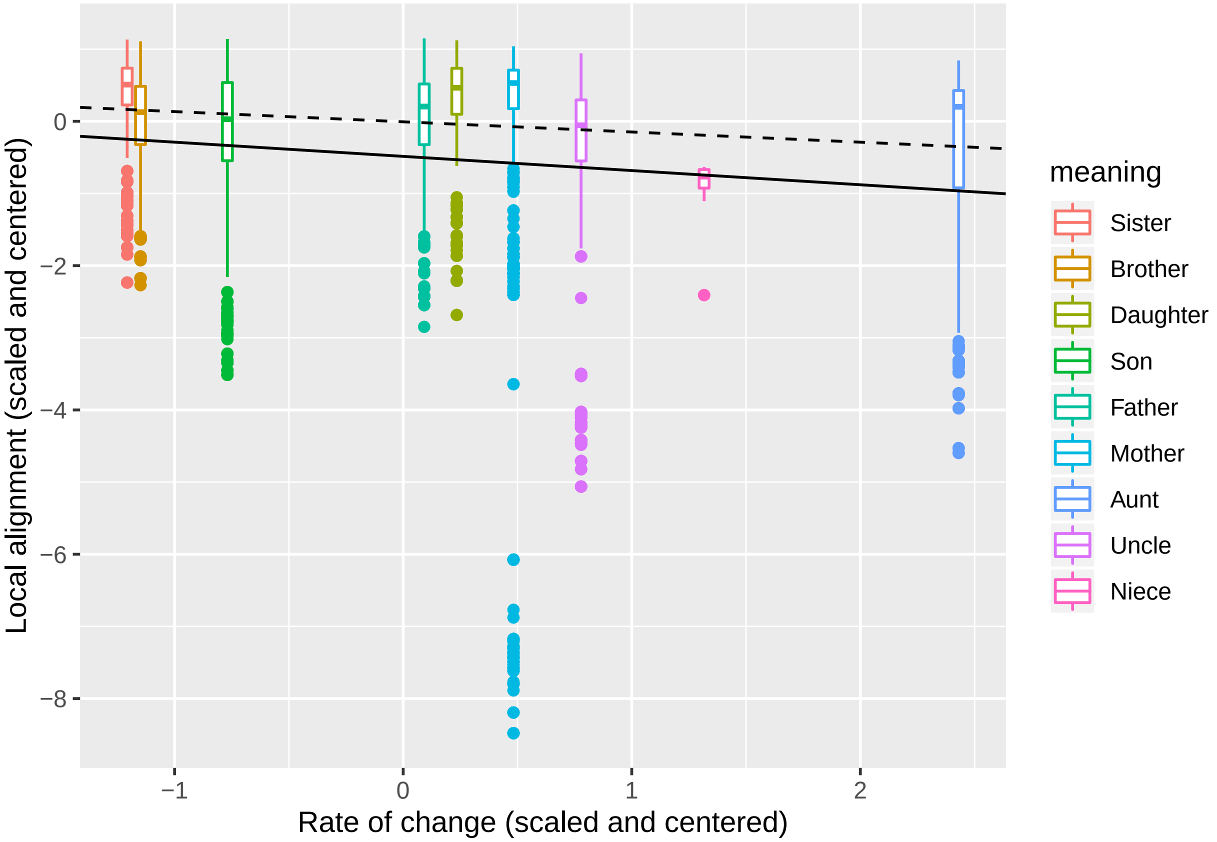
```
## refitting model(s) with ML (instead of REML)

## Data: rkin3Freq
## Models:
## m0.freq: local_alignment.scaled ~ freq_diff.scaled + (1 | lingpy.cognate/meaning) +
## m0.freq:     (1 | langPair)
## m1.freq: local_alignment.scaled ~ freq_diff.scaled + (1 | lingpy.cognate/meaning) +
## m1.freq:     (1 | langPair) + mean.roc.scaled
##         Df     AIC     BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## m0.freq  6 -3643.2 -3599.7 1827.6  -3655.2
## m1.freq  7 -3644.6 -3593.8 1829.3  -3658.6 3.3989      1    0.06524 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m1.freq)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## local_alignment.scaled ~ freq_diff.scaled + (1 | lingpy.cognate/meaning) +
##     (1 | langPair) + mean.roc.scaled
##    Data: rkin3Freq
## Control: lmerControl(optimizer = "bobyqa")
##
## REML criterion at convergence: -3641.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.1935 -0.5043  0.0438  0.5373  6.4121
##
## Random effects:
##  Groups                 Name        Variance Std.Dev.
##  langPair               (Intercept) 0.17776  0.4216
##  lingpy.cognate         (Intercept) 0.12922  0.3595
##  meaning:lingpy.cognate (Intercept) 0.44379  0.6662
##  Residual                           0.03569  0.1889
## Number of obs: 10481, groups:
## langPair, 171; lingpy.cognate, 86; meaning:lingpy.cognate, 86
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)      -0.006664   0.089906  -0.074
## freq_diff.scaled -0.004499   0.002196  -2.048
## mean.roc.scaled  -0.141280   0.076744  -1.841
##
## Correlation of Fixed Effects:
##             (Intr) frq_d.
## frq_dff.scl  0.003
## men.rc.scld -0.118  0.006
```

Plot data with model estimate (solid line), and estimate when taking frequency into account (dashed line). The rate of change is calculated for each meaning, hence the clustering. Note the plot is in scaled space.

```
ggplot(rkin3, aes(y=local_alignment.scaled,
                  x=mean.roc.scaled,
                  color=meaning)) +
  geom_boxplot() +
  geom_abline(slope=fixef(m1)["mean.roc.scaled"],
              intercept = fixef(m1)[1]) +
  geom_abline(slope=fixef(m1.freq)["mean.roc.scaled"],
              intercept = fixef(m1.freq)[1],linetype = 2) +
  xlab("Rate of change (scaled and centered)") +
  ylab("Local alignment (scaled and centered)")
```


Local alignment (scaled and centered)
Rate of change (scaled and centered)
meaning
Sister
Brother
Daughter
Son
Father
Mother
Aunt
Uncle
Niece