

# Supporting Online Material 7: statistical analyses in Witzlack-Makarevich et al. (2016) *Decomposing ...*

*Balthasar Bickel*

*January 28, 2016*

## Contents

<b>1</b>	<b>Data</b>	<b>2</b>
<b>2</b>	<b>Trees</b>	<b>4</b>
<b>3</b>	<b>Set-based models</b>	<b>7</b>
<b>4</b>	<b>Tree-based models</b>	<b>9</b>
4.1	Kiranti . . . . .	12
4.1.1	Constant evolution over the entire tree . . . . .	12
4.1.1.1	Non-Past Paradigms . . . . .	12
4.1.1.2	Past Paradigms . . . . .	13
4.1.1.3	Correlated evolution of the two paradigms . . . . .	13
4.1.2	Non-constant evolution, with possible changes in Western and Central-Eastern Kiranti	13
4.1.2.1	Non-Past Paradigms . . . . .	15
4.1.2.2	Past Paradigms . . . . .	15
4.1.2.3	Correlated evolution of the two paradigms . . . . .	16
4.2	Algonquian . . . . .	16
	<b>References</b>	<b>17</b>

## List of Figures

1	Kiranti . . . . .	5
2	Algonquian, traditional genealogy . . . . .	5
3	Algonquian, alternative genealogy . . . . .	6
4	The Western vs Central-Eastern distinction mapped on the tree as evolutionary constraints .	14

# 1 Data

The Kiranti data are sometimes split into paradigms. We treat each paradigm as an independent diachronic trial (a particular instance of morphological evolution), as explained in the main paper. Therefore, we split the data into a set of past tense and a set of nonpast tense paradigms, assuming that the two sets reflect independent histories. Where there is no difference between the two paradigms with regard to the rankings, we take the same count data for both past and non-past paradigms.

In addition, we also include sets where the two paradigms are combined so that we can fit models assuming correlations and interactions between the paradigms. For the set-based method, a suitable format for this is one in which the paradigms of a language are coded as if they were two off-shoots of that language (`kir.data.all`). For the tree-based method, we store the counts per ranking per paradigm as two variables per language (`kir.data.corr` below, after further data preparation steps).

```
kir.data <- read.csv('som4_KirantiAggregated.csv', header=T,check.names=F)

kir.data.npst <- subset(kir.data, grepl('NPST', TAM) | grepl('_IND', TAM) | grepl('^IND', TAM)) %>%
  select(-TAM)
kir.data.pst <- subset(kir.data, grepl('_PST', TAM) | grepl('_IND', TAM) | grepl('^IND', TAM)) %>%
  select(-TAM)

kir.data.all <- kir.data %>% mutate(language=paste(language, gsub('\\d*_',' ',TAM))) %>%
  select(-TAM)

alg.data <- read.csv('som3_AlgonquianAggregated.csv', header=T,check.names=F)
```

For each family, we generate a list of rankings. Each element of this list is a `data.frame` reporting the proportion in which the ranking is attested in each language. If there is a zero count, we interpret this as no evidence for the ranking. This is then reported as a .5 probability for either ranking.

For the tree-based approach we also need arcsine square root transformations in order to allow estimates outside the [0,1] interval.

```
prep.data <- function(family) {
  # 1>2
  fam12 <- family[,c("language","1 > 2","2 > 1")] %>% mutate(
    total=`2 > 1` + `1 > 2`,
    prop.tmp= `1 > 2`/total,
    prop=ifelse(is.nan(prop.tmp),.5, prop.tmp),
    propt=asin(sqrt(prop))
  ) %>% select(-prop.tmp)

  # 2>1
  fam21 <- family[,c("language","2 > 1","1 > 2")] %>% mutate(
    total=`2 > 1` + `1 > 2`,
    prop.tmp= `2 > 1`/total,
    prop=ifelse(is.nan(prop.tmp),.5, prop.tmp),
    propt=asin(sqrt(prop))
  ) %>% select(-prop.tmp)

  # 1>3
  fam13 <- family[,c("language","1 > 3","3 > 1")] %>% mutate(
    total=`3 > 1` + `1 > 3`,
    prop.tmp= `1 > 3`/total,
    prop=ifelse(is.nan(prop.tmp),.5, prop.tmp),
    propt=asin(sqrt(prop))
  ) %>% select(-prop.tmp)
```

```

    ) %>% select(-prop.tmp)
# 2>3
fam23 <- family[,c("language", "2 > 3", "3 > 2")] %>% mutate(
  total=`2 > 3` + `3 > 2`,
  prop.tmp= `2 > 3`/total,
  prop=ifelse(is.nan(prop.tmp), .5, prop.tmp),
  propt=asin(sqrt(prop))
) %>% select(-prop.tmp)

list( fam12=fam12,
      fam21=fam21,
      fam13=fam13,
      fam23=fam23
    )
}

kir.data.all.list <- prep.data(kir.data.all)
kir.data.npst.list <- prep.data(kir.data.npst)
kir.data.pst.list <- prep.data(kir.data.pst)
alg.data.list <- prep.data(alg.data)

```

We will also look into the Western and the Central-Eastern branches of Kiranti separately. For the set-based method, this requires a split of the lists; for tree-based methods we can model the impact of the Western vs. Central-Eastern distinction directly with the original data.

*Western Kiranti:*

```

kir.w.lgs <- c('Bahing', 'Kõic', 'Dumi', 'Jero', 'Wambule', 'Koyi')
kir.w.data.all <- subset(kir.data.all,
  gsub('(.*)\\s.*', '\\1', language) %in% kir.w.lgs)
kir.w.data.all.list <- prep.data(kir.w.data.all)
kir.w.data.npst <- subset(kir.data.npst,
  gsub('(.*)\\s.*', '\\1', language) %in% kir.w.lgs)
kir.w.data.npst.list <- prep.data(kir.w.data.npst)
kir.w.data.pst <- subset(kir.data.pst,
  gsub('(.*)\\s.*', '\\1', language) %in% kir.w.lgs)
kir.w.data.pst.list <- prep.data(kir.w.data.pst)

```

*Central-Eastern Kiranti:*

```

kir.ce.data.all <- subset(kir.data.all,
  !gsub('(.*)\\s.*', '\\1', language) %in% kir.w.lgs)
kir.ce.data.all.list <- prep.data(kir.ce.data.all)
kir.ce.data.npst <- subset(kir.data.npst,
  !gsub('(.*)\\s.*', '\\1', language) %in% kir.w.lgs)
kir.ce.data.npst.list <- prep.data(kir.ce.data.npst)
kir.ce.data.pst <- subset(kir.data.pst,
  !gsub('(.*)\\s.*', '\\1', language) %in% kir.w.lgs)
kir.ce.data.pst.list <- prep.data(kir.ce.data.pst)

```

The following loop creates the input for correlated evolution models. It creates a list of rankings where each ranking consists of a matrix with columns for NPST and PST proportions and rows for the languages. (We focus on the transformed proportions only.) Where there is no difference between the paradigms in a

language, this means that NPST and PST have the same values. For Dumi and Chintang, the database happens to lack the entries, creating NAs (which don't pose a problem for model fitting).

```
kir.data.corr <- lapply(kir.data.all.list, function(f) {
  f[,-c(2:5)] %>%
  separate(language, c('language', 'paradigm'), sep=" ") %>%
  spread(paradigm, propt) %>%
    mutate( # if there is no difference, it's the same values:
      NPST.IND=ifelse(is.na(NPST.IND), IND, NPST.IND),
      PST.IND=ifelse(is.na(PST.IND), IND, PST.IND)
    ) %>%
  select(-IND) -> x
  m <- as.matrix(x[,-1])
  # when adding the names we also fix the name Limbu so it will match the tip label in the tree:
  dimnames(m)[[1]] <- gsub('Limbu', 'Phedappe_Limbu', x$language)
  return(m)
})
```

## 2 Trees

*Kiranti:*

```
kir.tree <- collapse.singles(read.newick(text="(((Kulung:1)Khambu:1,(Camling:1,Puma:1,Bantawa:1)Southe
```

```
par(family=my.font, lheight=1, cex=plot.cex)
kir.tree.print <- rotate(kir.tree, node=18)
plot(kir.tree.print, no.margin=T, font=1, label.offset =.02)
node.labels(gsub("_", " \n", kir.tree.print$node.label[-1]),
  node=18:27, bg='white', cex=plot.cex-.2)
```

*Algonquian*, in two versions:

```
alg.tree <- collapse.singles(read.newick(text='((Cheyenne:1):1,(Blackfoot:1)Siksika:1,(Arapaho:1)Arapah
```

```
alg.alt.tree <- collapse.singles(read.newick(text="((Blackfoot:1)Siksika:1,((Arapaho:1)Arapahoic:1,(Pla
```

```
par(family=my.font, lheight=.2, cex=plot.cex)
plot(alg.tree, no.margin=T, font=1, label.offset =.02)
node.labels(text='\nEastern Algonquian \n',
  node=10, bg='white', cex=plot.cex-.2)
```

```
par(family=my.font, lheight=.2, cex=plot.cex)
plot(alg.alt.tree, no.margin=T, font=1, label.offset =.02)
node.labels(text=c('1 ', '2 ', '3 ', '\nEastern Algonquian \n'),
  node=10:13, bg='white', cex=plot.cex-.2)
```

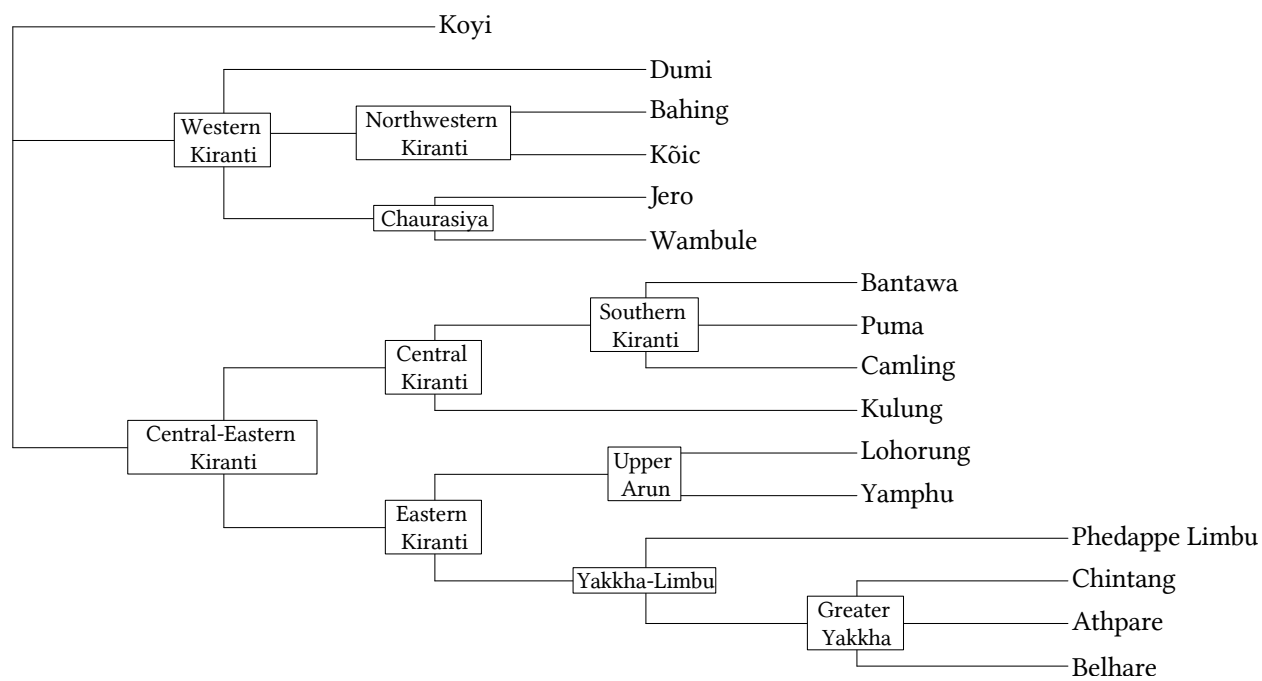


Figure 1: Kiranti

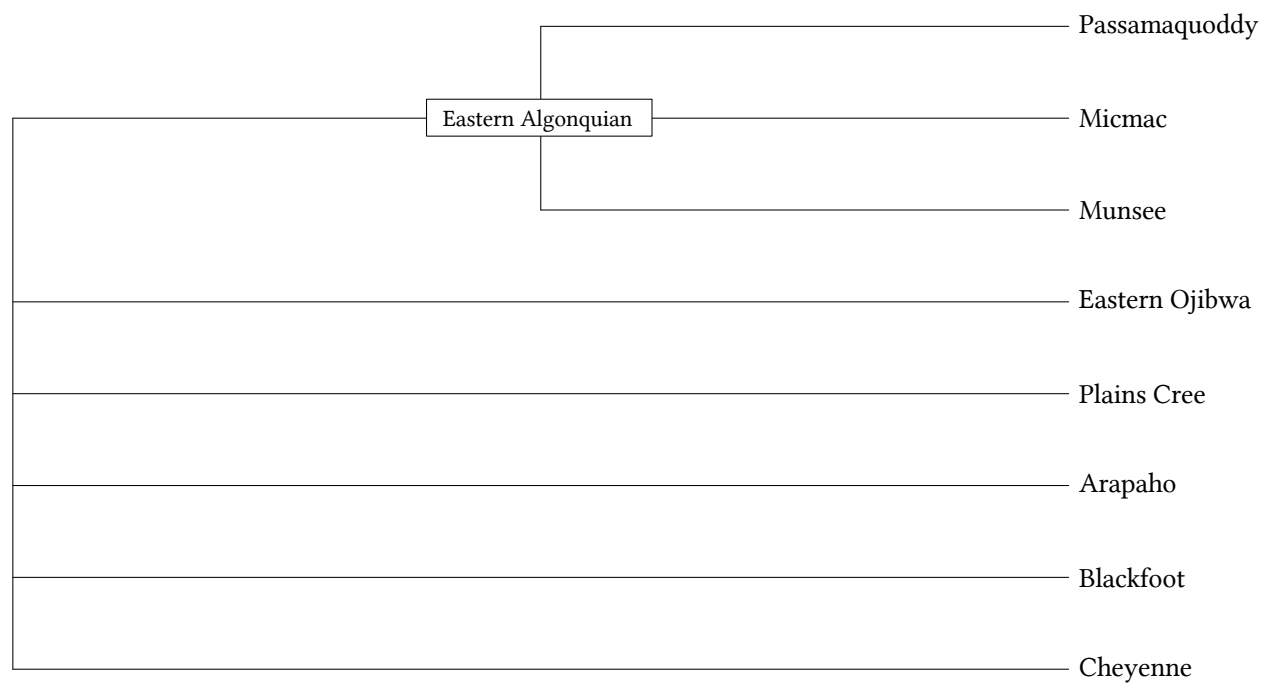


Figure 2: Algonquian, traditional genealogy

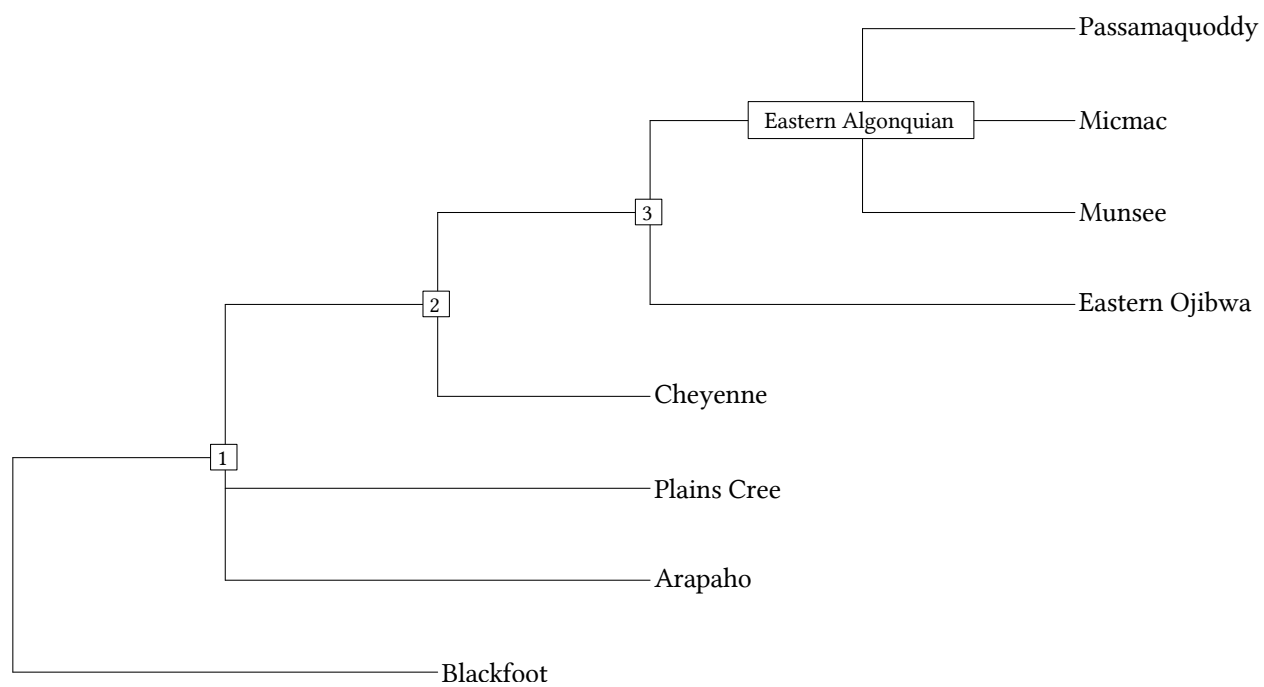


Figure 3: Algonquian, alternative genealogy

### 3 Set-based models

Here we first test for each language whether or not the count of cases favoring the ranking is greater than what would be expected under a Bernoulli process assuming equal probability ( $H_0$ ). If there is a zero count for a given ranking, we report this as  $p = 1$ :

```
binom.test.proc <- function(fam.data.list) {  
  lapply(fam.data.list, function(r) {  
    apply(r[,c(2,4)], 1, function(b) {  
      ifelse(b[2]>0,  
        binom.test(x=b[1], n=b[2], alternative='greater')$p.value,  
        1)  
    })  
  })  
}
```

From this we compute the proportion of languages (or paradigms) for which a given ranking is significantly preferred. In order to control for inflated evidence resulting from multiple testing, we correct p-values via Holm's procedure (Holm 1979). As an alternative, we also tried false discovery control (with a 10% threshold) using the `qvalue` package (Storey et al. 2015), but the results were the same.

```
prop.sig <- function(fam.list, sig, q) {  
  round(sapply(fam.list, function(r) { mean(p.adjust(r, method='holm')<sig)  
    # qs <- qvalue(r, pi0.method="bootstrap")$qvalues  
    # results <- data.frame(ps=r, qs=qs)  
    # with(results, mean(qs<q & ps<sig))  
  }  
    ), 2)  
}  
  
list.of.lists <- c(kir.data.all.list,  
  kir.w.data.all.list,  
  kir.ce.data.all.list,  
  kir.data.npst.list,  
  kir.w.data.npst.list,  
  kir.ce.data.npst.list,  
  kir.data.pst.list,  
  kir.w.data.pst.list,  
  kir.ce.data.pst.list,  
  alg.data.list)  
  
combine <- function(list.of.lists, sign, q) {  
  matrix(prop.sig(binom.test.proc(list.of.lists), sig=sign), ncol=4, byrow=T,  
    dimnames=list(c('Kiranti (all)', 'Western Kiranti (all)',  
      'Central-Eastern Kiranti (all)',  
      'Kiranti (NPST)', 'Western Kiranti (NPST)', 'Central-Eastern Kiranti (NPST)',  
      'Kiranti (PST)', 'Western Kiranti (PST)', 'Central-Eastern Kiranti (PST)',  
      'Algonquian'  
    ), c("1>2", "2>1", "1>3", "2>3")  
  ))  
}
```

```
combine(list.of.lists, sign=.05, q=.1)
```

##	1>2	2>1	1>3	2>3
## Kiranti (all)	0.00	0.00	0.09	0.13
## Western Kiranti (all)	0.22	0.00	0.44	0.00
## Central-Eastern Kiranti (all)	0.00	0.00	0.07	0.21
## Kiranti (NPST)	0.00	0.00	0.12	0.12
## Western Kiranti (NPST)	0.17	0.00	0.33	0.00
## Central-Eastern Kiranti (NPST)	0.00	0.00	0.20	0.20
## Kiranti (PST)	0.00	0.00	0.00	0.14
## Western Kiranti (PST)	0.20	0.00	0.40	0.00
## Central-Eastern Kiranti (PST)	0.00	0.00	0.11	0.22
## Algonquian	0.00	0.25	0.62	0.62



## 4 Tree-based models

We compare Ornstein-Uhlenbeck ('OU') to Brownian Motion ('BM') models using likelihood ratio tests and sample-size corrected AICs (AICc below), following Butler & King (2004). A Brownian Motion model is a special kind of Wiener process ( $W$ ) or random walk, where the values of a trait evolve randomly with diffusion intensity  $\sigma$ , or in terms of a stochastic differential equation for trait  $X$  over continuous time  $t$ :  $dX(t) = \sigma dW(t)$ . An Ornstein-Uhlenbeck model constrains this further by a drift to an optimum value  $\theta$ , towards which the process is attracted with strength  $\alpha$ :  $dX(t) = \alpha[\theta - X(t)]dt + \sigma dW(t)$  (Butler & King 2004).

The models are fitted with the `mvBM` and `mvOU` functions from package `mvMORPH` (Clavel et al. 2015). Note that `mvOU` generates both a root and an optimum ( $\theta$  in the equation) estimate in the object `theta` but when there is no root, `theta` simplifies to the optimum estimate alone. We catch this in the code below by checking the number of rows in `theta`. (The root estimates themselves are not reliable with these data and of no interest.)

To allow different trait histories in Western vs. Central-Eastern Kiranti (via the `groups` parameter in the code below), we first generate trees with a character state map produced by `phytools::make.simap` (Revell 2012). For this, we assume that transitions between the groups are equally likely ('equal rates' or 'ER' model). The mapping is stochastic, so in principle one could sample from a series of mappings that would differ in where exactly along the branches the languages group into Western vs. Central-Eastern. However, as we use the mapping only as a control for possible multiple optima, it does not matter where and when the grouping really took place. Therefore, we pick whichever mapping happens to fall out (as in Clavel et al. (2015), appendix). For illustration of a mapping, see below when we report the results from Kiranti.

Models with different histories (OU2 below) are compared to models with a single history (OU1) by likelihood ratio tests (reported as LR.OU2 below) and a sample-size corrected version of Akaike's An Information Criterion, AIC<sub>c</sub> (AICc).

### *Single-trait evolution models:*

```
tree.test.proc <- function(fam.data.list, fam.tree, groups=NULL, transformed=T) {
  do.call(rbind,
    lapply(seq_along(fam.data.list), function(i) {

      ranking <- gsub('.*(\\d)(\\d)', '\\1 > \\2', names(fam.data.list)[[i]])

      fam.data <- fam.data.list[[i]]

      mean.prop <- mean(fam.data$prop)
      sd.prop <- sd(fam.data$prop)

      if(transformed) {
        # use the arcsine-sqrt values available in "propt"
        fam.data <- setNames(fam.data$propt, fam.data$language)
      } else {
        fam.data <- setNames(fam.data$prop, fam.data$language)
      }

      bm <- mvBM(fam.tree, fam.data, model='BM1', echo=F, diagnostic=F)
      ou1 <- mvOU(fam.tree, fam.data, model='OU1', echo=F, diagnostic=F)

      if(!is.null(groups)) {
        # map the groups onto the tree
        fam.tree.maps <- make.simap(fam.tree, groups, model='ER', message=F)
        # and fit a model allowing different selective regimes per group:
```

```

ou2 <- mvOU(fam.tree.maps, fam.data, model='OUM', echo=F, diagnostic=F)
}

# Likelihood ratio test and sample-size corrected AIC:
LR.OU1 <- 2 * (ou1$LogLik - bm$LogLik)
df.OU1 <- ou1$param$nparam - bm$param$nparam
p.OU1 <- pchisq(LR.OU1, df = df.OU1, lower.tail = F)
AICc.BM <- bm$AICc
AICc.OU1 <- ou1$AICc

if(!is.null(groups)) {
  LR.OU2 <- 2 * (ou2$LogLik - ou1$LogLik)
  df.OU2 <- ou2$param$nparam - ou1$param$nparam
  p.OU2 <- pchisq(LR.OU2, df = df.OU2, lower.tail = F)
  AICc.OU2 <- ou2$AICc
}

# Model parameters:
param.est <- data.frame(
  BM.sigma= bm$sigma,
  OU1.theta = ifelse(nrow(ou1$theta)==1, # if there is no separate root estimate:
    ou1$theta,
    # else, the theta parameter is in the second row:
    ou1$theta[2,1]),
  OU1.alpha= ou1$alpha[1,1],
  OU1.sigma= ou1$sigma[1,1]
)

if(!is.null(groups)) {
  param.est <- cbind(param.est, OU2.alpha=ou2$alpha[1,1],
    OU2.sigma= ou2$sigma[1,1])
  # get the OUM parameters
  theta2 <- ou2$theta[-1,]
  names(theta2) <- paste0('OU2.theta_', names(theta2))
  # and add them as columns:
  param.est <- cbind(param.est, data.frame(t(theta2)))
}

if(transformed) {
  # transform all parameters back from the arcine sqrt transformation:
  param.est <- t(apply(param.est, 1, function(r) sin(r)^2))
}

param.est <- t(apply(param.est, 1, round, 4))

if(!is.null(groups)) {
  means <- data.frame(t(tapply(fam.data, groups, function(i) round(mean(i),2))))
  names(means) <- paste0('m_', names(means))
  sds <- data.frame(t(tapply(fam.data, groups, function(i) round(sd(i),2))))
  names(sds) <- paste0('sd_', names(sds))
  data.frame(row.names=ranking,
    means, sds,
    LR.OU1=round(LR.OU1, 2), p.OU1=round(p.OU1, 4), df.OU1=df.OU1,

```

```

        LR.OU2=round(LR.OU2, 2), p.OU2=round(p.OU2, 4), df.OU2=df.OU2,
        AICc.BM=round(AICc.BM, 2), AICc.OU1=round(AICc.OU1,2),
        AICc.OU2=round(AICc.OU2,2),
        param.est)
    } else {
colnames(param.est) <- gsub('1','', colnames(param.est))
data.frame(row.names=ranking,
            m.prop=round(mean.prop,2), sd.prop=round(sd.prop,2),
            LR.OU=round(LR.OU1, 2), p.OU=round(p.OU1, 4), df.OU=df.OU1,
            AICc.BM=round(AICc.BM, 2), AICc.OU=round(AICc.OU1,2),
            Red=round((AICc.OU1-AICc.BM),2),
            param.est)
    }
}
))
}

```

### *Correlated evolution models (Kiranti only):*

These are models that assume a correlation between the proportions in the two paradigms, yielding a larger number of parameters to estimate. Following Clavel et al. (2015), we rely on sample-sized corrected AIC comparison (AICc) exclusively here.

```

tree.corr.test.proc <- function(fam.data.list=kir.data.corr, fam.tree=kir.tree, groups=NULL) {

    kir.corr.res <- lapply(seq_along(fam.data.list), function(i) {

        fam.data <- kir.data.corr[[i]]

        bm <- mvBM(fam.tree, fam.data, model='BM1', echo=F, diagnostic=F)
        ou1 <- mvOU(fam.tree, fam.data, model='OU1', echo=F, diagnostic=F)

        if(!is.null(groups)) {
            # map the groups onto the tree
            fam.tree.maps <- make.simmmap(fam.tree, groups, model='ER', message=F)
            # and fit a model allowing different selective regimes per group:
            ou2 <- mvOU(fam.tree.maps, fam.data, model='OUM', echo=F, diagnostic=F)
        }

        # because of the complex output, we don't force a data.frame summary here
        param.est <- list(
            BM.sigma= bm$sigma,
            OU.theta = ou1$theta[2, ], # we don't need the root in row 1
            OU.alpha= ou1$alpha,
            OU.sigma= ou1$sigma
        )

        if(!is.null(groups)) {
            param.est <- c(param.est, list(OU2.alpha=ou2$alpha,
                OU2.sigma= ou2$sigma,
                OU2.theta=ou2$theta[-1,]
            ))
        }

    })
}

```

```

res <- list(AICc=data.frame(BM=round(bm$AICc,2), OU=round(ou1$AICc,2)),
           parameters=lapply(param.est, function(p) sin(p)^2),
           corr=cov2cor(stationary(ou1))
          )

if(!is.null(groups)) {
  res <- list(AICc=data.frame(BM=round(bm$AICc,2), OU1=round(ou1$AICc,2),
                             OU2=round(ou2$AICc, 2)),
             parameters=lapply(param.est, function(p) sin(p)^2),
             corr.ou1=cov2cor(stationary(ou1)),
             corr.ou2=cov2cor(stationary(ou2))
            )
}
return(res)
}
)
}

```

## 4.1 Kiranti

We first need to fix the language names in the data list so that they match what's in the tree:

```

kir.data.npst.list.fixed <- lapply(kir.data.npst.list, function(l) {
  l$language <- gsub('Limbu', 'Phedappe_Limbu', l$language)
  return(l)
})
kir.data.pst.list.fixed <- lapply(kir.data.pst.list, function(l) {
  l$language <- gsub('Limbu', 'Phedappe_Limbu', l$language)
  return(l)
})

```

### 4.1.1 Constant evolution over the entire tree

Here we assume a single selection regime over the entire tree, i.e. we assume that the parameters of Ornstein-Uhlenbeck model are the same throughout and paradigms evolve towards the optimum value during the history of the entire family.

```

(kir.npst.res.trees <- tree.test.proc(kir.data.npst.list.fixed, kir.tree))

```

#### 4.1.1.1 Non-Past Paradigms

	##	m.prop	sd.prop	LR.OU	p.OU	df.OU	AICc.BM	AICc.OU	Red BM.sigma	OU.theta	OU.alpha	OU.sigma
## 1 > 2	0.61	0.30	4.04	0.1326	2	26.37	29.04	2.67	0.0098	0.7854	0.6254	0.1005
## 2 > 1	0.39	0.30	4.04	0.1326	2	26.37	29.04	2.67	0.0098	0.2146	0.6254	0.1005
## 1 > 3	0.76	0.27	11.60	0.0030	2	28.85	23.97	-4.89	0.0133	0.9136	0.7809	0.2477
## 2 > 3	0.85	0.28	7.23	0.0269	2	28.17	27.65	-0.52	0.0122	0.9977	0.6782	0.0946

**4.1.1.2 Past Paradigms** For testing these paradigms we need to drop Dumi and Chintang because we happen to lack the data in the database:

```
kir.tree.reduced <- drop.tip(kir.tree, tip=c('Dumi', 'Chintang'))
(kir.pst.res.trees <- tree.test.proc(kir.data.pst.list.fixed, kir.tree.reduced))
```

```
##      m.prop sd.prop LR.OU  p.OU df.OU AICc.BM AICc.OU  Red BM.sigma OU.theta OU.alpha OU.sigma
## 1 > 2  0.58   0.30  3.74 0.1540    2  23.83  27.45  3.61  0.0092  0.7194  0.6232  0.0967
## 2 > 1  0.42   0.30  3.74 0.1540    2  23.83  27.45  3.61  0.0092  0.2806  0.6232  0.0967
## 1 > 3  0.75   0.27  7.42 0.0244    2  20.82  20.75 -0.07  0.0060  0.9367  0.8940  0.0679
## 2 > 3  0.82   0.31  1.13 0.5685    2  20.70  26.93  6.22  0.0059  0.9942  0.1647  0.0255
```

```
kir.corr.res <- tree.corr.test.proc()
```

**4.1.1.3 Correlated evolution of the two paradigms** *AIC<sub>c</sub> comparison:*

```
kir.corr.res.aiccs <- do.call(rbind, sapply(kir.corr.res, "[", 1))
rownames(kir.corr.res.aiccs) <- gsub('.*(\\d)(\\d)', '\\1 > \\2', names(kir.data.corr))
kir.corr.res.aiccs
```

```
##      BM    OU
## 1 > 2  9.64 47.31
## 2 > 1  9.64 47.31
## 1 > 3 22.31 53.36
## 2 > 3 -9.59 24.97
```

**4.1.2 Non-constant evolution, with possible changes in Western and Central-Eastern Kiranti**

For this we map the Western vs Central-Eastern group distinction to the tree, as explained above.

```
n.west <- length(kir.w.lgs)
groups <- c(rep('Central-Eastern', length(kir.tree$tip.label)-n.west), rep('Western', n.west))
names(groups) <- kir.tree$tip.label
```

Here is how the mapping is done, and Fig. 4 below is an illustration of the result:

```
kir.tree.maps <- make.simmap(kir.tree, groups, model='ER')
```

```
## make.simmap is sampling character histories conditioned on the transition matrix
##
## Q =
##      Central-Eastern  Western
## Central-Eastern -0.03906933  0.03906933
## Western         0.03906933 -0.03906933
## (estimated using likelihood);
## and (mean) root node prior probabilities
## pi =
## Central-Eastern  Western
##      0.5         0.5
```

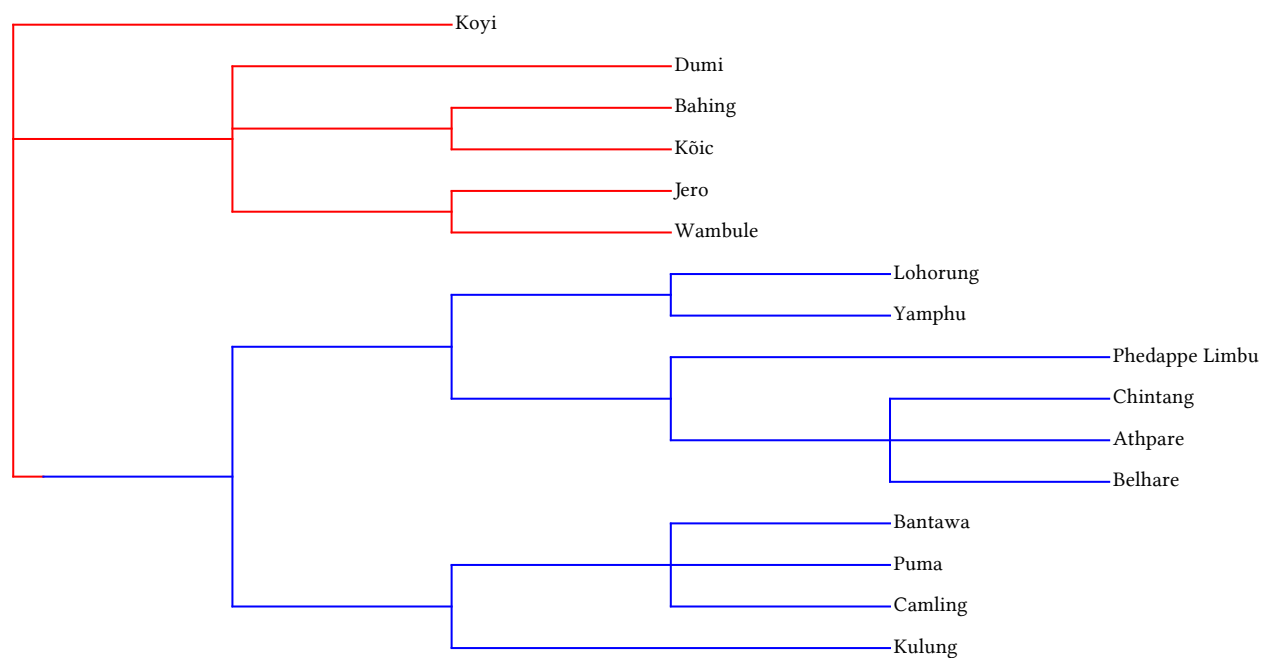


Figure 4: The Western vs Central-Eastern distinction mapped on the tree as evolutionary constraints

```
kir.npst.res.trees.m <- tree.test.proc(kir.data.npst.list.fixed, kir.tree, groups=groups)
```

#### 4.1.2.1 Non-Past Paradigms *Descriptives:*

##	m_Central.Eastern	m_Western	sd_Central.Eastern	sd_Western
## 1 > 2	1.08	0.74	0.51	0.20
## 2 > 1	0.49	0.83	0.51	0.20
## 1 > 3	1.28	1.04	0.38	0.45
## 2 > 3	1.50	1.03	0.23	0.60

*LR tests and AIC<sub>c</sub> comparison:* LR.OU1 and p.OU1 report a comparison with BM; LR.OU2 and pOU2 report a comparison with OU1 (i.e. whether a model with flexible histories (OU2) fits better than one with a single history (OU1))

##	LR.OU1	p.OU1	df.OU1	LR.OU2	p.OU2	df.OU2	AICc.BM	AICc.OU1	AICc.OU2
## 1 > 2	4.04	0.1326	2	1.38	0.2393	1	26.37	29.04	32.02
## 2 > 1	4.04	0.1326	2	1.38	0.2393	1	26.37	29.04	32.02
## 1 > 3	11.60	0.0030	2	0.18	0.6754	1	28.85	23.97	28.15
## 2 > 3	7.23	0.0269	2	2.13	0.1441	1	28.17	27.65	29.88

*Model parameters:*

##	BM.sigma	OU1.theta	OU1.alpha	OU1.sigma	OU2.alpha	OU2.sigma
## 1 > 2	0.0098	0.7854	0.6253	0.1005	0.3610	0.6463
## 2 > 1	0.0098	0.2146	0.6254	0.1005	0.0068	0.8046
## 1 > 3	0.0133	0.9136	0.7809	0.2477	0.9337	0.8026
## 2 > 3	0.0122	0.9977	0.6782	0.0946	0.2066	0.9976

##	OU2.theta_Central.Eastern	OU2.theta_Western
## 1 > 2	0.7781	0.4477
## 2 > 1	0.2219	0.5524
## 1 > 3	0.9174	0.8245
## 2 > 3	0.9949	0.7469

```
groups.reduced <- groups[!names(groups) %in% c('Dumi', 'Chintang')]
kir.pst.res.trees.m <- tree.test.proc(kir.data.pst.list.fixed, kir.tree.reduced, groups=groups.reduced)
```

#### 4.1.2.2 Past Paradigms *Descriptives:*

##	m_Central.Eastern	m_Western	sd_Central.Eastern	sd_Western
## 1 > 2	1.03	0.68	0.51	0.14
## 2 > 1	0.54	0.89	0.51	0.14
## 1 > 3	1.29	0.93	0.35	0.41
## 2 > 3	1.48	0.92	0.29	0.60

*LR tests and AIC<sub>c</sub> comparison:* LR.OU1 and p.OU1 report a comparison with BM; LR.OU2 and pOU2 report a comparison with OU1 (i.e. whether a model with flexible histories (OU2) fits better than one with a single history (OU1))

##		LR.OU1	p.OU1	df.OU1	LR.OU2	p.OU2	df.OU2	AICc.BM	AICc.OU1	AICc.OU2
##	1 > 2	3.74	0.1540	2	1.51	0.2198	1	23.83	27.45	31.00
##	2 > 1	3.74	0.1540	2	1.51	0.2198	1	23.83	27.45	31.00
##	1 > 3	7.42	0.0244	2	0.23	0.6333	1	20.82	20.75	25.58
##	2 > 3	1.13	0.5685	2	2.05	0.1519	1	20.70	26.93	29.93

*Model parameters:*

##		BM.sigma	OU1.theta	OU1.alpha	OU1.sigma	OU2.alpha	OU2.sigma
##	1 > 2	0.0092	0.7194	0.6232	0.0967	0.7370	0.6091
##	2 > 1	0.0092	0.2806	0.6232	0.0967	0.7184	0.6153
##	1 > 3	0.0060	0.9367	0.8940	0.0679	0.3557	0.2484
##	2 > 3	0.0059	0.9942	0.1647	0.0255	0.4222	0.0432

##		OU1.alpha	OU1.sigma	OU2.alpha	OU2.sigma	OU2.theta_Central	OU2.theta_Eastern	OU2.theta_Western
##	1 > 2	0.6232	0.0967	0.7370	0.6091		0.7349	0.3705
##	2 > 1	0.6232	0.0967	0.7184	0.6153		0.2651	0.6295
##	1 > 3	0.8940	0.0679	0.3557	0.2484		0.9223	0.7742
##	2 > 3	0.1647	0.0255	0.4222	0.0432		0.9803	0.4266

**4.1.2.3 Correlated evolution of the two paradigms** *AIC<sub>c</sub> comparison:* OU1 is a model with a single history, OU2 one with flexible histories:

```
kir.corr.res.m <- tree.corr.test.proc(groups=groups)
kir.corr.res.aiccs.m <- do.call(rbind, sapply(kir.corr.res.m, "[", 1))
rownames(kir.corr.res.aiccs.m) <- gsub('.*(\\d)(\\d)', '\\1 > \\2', names(kir.data.corr))
kir.corr.res.aiccs.m
```

##		BM	OU1	OU2
##	1 > 2	9.64	47.31	108.61
##	2 > 1	9.64	47.31	108.62
##	1 > 3	22.31	53.36	116.44
##	2 > 3	-9.59	24.97	85.27

## 4.2 Algonquian

We need to fix the language names in the tree so that they match what's in the data list:

```
alg.tree$tip.label <- with(alg.tree, ifelse(tip.label %in% 'Plains_Cree', 'Cree (Plains)',
                                           ifelse(tip.label %in% 'Eastern_Ojibwa', 'Ojibwa (Eastern)', tip.label)))
alg.alt.tree$tip.label <- with(alg.alt.tree, ifelse(tip.label %in% 'Plains_Cree', 'Cree (Plains)',
                                                    ifelse(tip.label %in% 'Eastern_Ojibwa', 'Ojibwa (Eastern)', tip.label)))
```

Flat tree (as in Figure 2)



```
(alg.res.trees <- tree.test.proc(alg.data.list, alg.tree))
```

```
##      m.prop sd.prop LR.OU  p.OU df.OU AICc.BM AICc.OU  Red BM.sigma OU.theta OU.alpha OU.sigma
## 1 > 2  0.23   0.26  3.62 0.0571    1  15.34  17.32 1.98  0.0095  0.1788  0.6564  0.9943
## 2 > 1  0.77   0.26  3.62 0.0571    1  15.34  17.32 1.98  0.0095  0.8212  0.4915  0.9986
## 1 > 3  0.82   0.14  0.00 0.9990    1   5.04  10.64 5.60  0.0007  0.8950  0.0000  0.0007
## 2 > 3  0.79   0.19  0.00 1.0000    1   8.44  14.04 5.60  0.0017  0.8621  0.0000  0.0017
```

Alternative, more structured tree (as in Figure 3)

```
(alg.alt.res.trees <- tree.test.proc(alg.data.list, alg.alt.tree))
```

```
##      m.prop sd.prop LR.OU  p.OU df.OU AICc.BM AICc.OU  Red BM.sigma OU.theta OU.alpha OU.sigma
## 1 > 2  0.23   0.26  5.51 0.0635    2  17.23  26.65 9.42  0.0092  0.1759  0.8269  0.9042
## 2 > 1  0.77   0.26  5.51 0.0635    2  17.23  26.65 9.42  0.0092  0.8241  0.8260  0.9040
## 1 > 3  0.82   0.14  3.16 0.2061    2   6.31  18.08 11.77 0.0006  0.8289  0.8292  0.3615
## 2 > 3  0.79   0.19  3.47 0.1760    2   9.66  21.12 11.46 0.0014  0.8788  0.8282  0.6553
```

## References

- Butler, Marguerite A. & Aaron A. King. 2004. Phylogenetic comparative analysis: A modeling approach for adaptive evolution. *The American Naturalist* 164. 683–695.
- Clavel, Julien, Gilles Escarguel & Gildas Merceron. 2015. mvMORPH: An R package for fitting multivariate evolutionary models to morphometric data. *Methods in Ecology and Evolution* 6. 1311–1319.
- Holm, Sture. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6. 65–70.
- Revell, Liam J. 2012. phytools: An R package for phylogenetic comparative biology (and other things). *Methods in Ecology and Evolution* 3. 217–223.
- Storey, John D., Andrew J. Bass, Alan Dabney & David Robinson. 2015. qvalue: Q-value estimation for false discovery rate control. ms.