Model Map in Phylogenetics

Hidetoshi Shimodaira

2019/01/14

Visualization of Pametric Models

Load the package *scaleboot*.

```
library(scaleboot)
```

The methods are explained in Shimodaira and Terada (2019).

Hidetoshi Shimodaira and Yoshikazu Terada. Selective Inference for Testing Trees and Edges in Phylogenetics. 2019.

Phylogenetic Analysis of 15 trees of 6 taxa

As a working example, we estimate the phylogenetic tree from the same dataset previously analyzed in Shimodaira and Hasegawa (1999), Shimodaira (2001, 2002) using the same model of evolution. The dataset consists of mitochondrial protein sequences of six mammalian species with $n = 3414$ amino acids The taxa are *Homo sapiens* (human), *Phoca vitulina* (seal), *Bos taurus* (cow), *Oryctolagus cuniculus* (rabbit), *Mus musculus* (mouse), and *Didelphis virginiana* (opossum). The software package PAML (Yang 1997) was used to calculate the site-wise log-likelihoods for the trees. The mtREV model (Adachi and Hasgawa 1996) was used for amino acid substitutions, and the site-heterogeneity was modeled by the discrete-gamma distribution (Yang 1996).

We first perform phylogentic tree selection. Look at the other RMarkdown document *phylo.Rmd* for the details. Here, we consider only 15 trees of 6 taxa, by fixing the clade (seal, cow).

We first run a phylogenetic package, such as PAML, to calculate *site-wise log-likelihood* for trees. The tree topology file is mam15.tpl, and the site-wise log-likelhiood file is mam15.mt. The mam15.mt file is converted from mam15.lnf (output from PAML) by seqmt program in CONSEL. We also run treeass in CONSEL to get mam15.ass and mam15.log from mam15.tpl. We use CONSEL only for preparing *mt* and *ass* files. All these files are found in mam15 folder.

Instead of using the program *consel* in CONSEL to compute p-values, we use scaleboot here. First, read the following two files. Then run *relltest* (internally calling *scaleboot* function) to perform multiscale bootstrap resampling.

```
### dont run
nb.rell = 100000
nb.pvclust = 10000
library(parallel)
length(cl <- makeCluster(detectCores()))
mam15.mt <- read.mt("mam15-files/mam15.mt")
mam15.ass <- read.ass("mam15-files/mam15.ass")
sa <- 9^seq(-1,1,length=13) # specify scales for multiscale bootstrap
mam15.relltest <- relltest(mam15.mt,nb=nb.rell,sa=sa,ass=mam15.ass,cluster=cl)
```

We have run the above command in *makedata.R* preveously. To get the results, simply do below, which will also load other objects.

```
data(mam15) # load mam15, mam26, mam105
ls() # look at the objects
```

```
##  [1] "mam105.ass"      "mam105.aux"      "mam105.mt"
##  [4] "mam105.relltest" "mam15.ass"       "mam15.aux"
##  [7] "mam15.mt"        "mam15.relltest"  "mam26.ass"
## [10] "mam26.aux"       "mam26.mt"
```

We have auxiliary information in *mam15.aux*. The topologies are in the order of mam15.tpl (the same order as mam15.mt). The edges are in the order of mam15.cld (extracted from mam15.log, which is the log file of treeass).

```
names(mam15.aux)
```

```
## [1] "tpl" "cld" "tax"
```

```
mam15.aux$tpl[1:3] # topologies (the first three trees, in the order of mam15.tpl file)
```

```
##                                          t1
## "((Homsa,(Phovi,Bosta)),Orycu,(Musmu,Didvi));"
##                                          t2
## "(Homsa,Orycu,((Phovi,Bosta),(Musmu,Didvi)));"
##                                          t3
## "(Homsa,((Phovi,Bosta),Orycu),(Musmu,Didvi));"
```

```
mam15.aux$cld[1:3] # edges  (the first three edges, in the order of  mam15.cld file)
```

```
##       e1       e2       e3
## "+++---" "++++--" "+--+--"
```

```
mam15.aux$tax # taxa, the order corresponds to the positions of + and - in the clade pattern.
```

```
## [1] "Homsa" "Phovi" "Bosta" "Orycu" "Musmu" "Didvi"
```

The output of relltest includes the results of trees and edges. We separate them, and also reorder the trees and edges in decreasing order of likelhiood values below. We can also specify the auxiliary information in *sbphylo*.

```
mam15 <- sbphylo(mam15.relltest, mam15.ass, treename=mam15.aux$tpl,edgename=mam15.aux$cld,taxaname=mam15.aux$tax)
```

This includes the multiscale bootstrap probability. The order can be checked as follows. T1, T2, T3, ... are sorted tree (in decreasing order of likelhiood). t1, t2, t3, ... are the original order of trees. E1, E2, E3, ... are sorted edges, and e1, e2, e3, ... are the original order of edges.

```
mam15$order.tree  # sorted tree to original tree
```

```
##  T1  T2  T3  T4  T5  T6  T7  T8  T9 T10 T11 T12 T13 T14 T15
##   1   3   2   5   6   7   4  15   8  14  13   9  11  10  12
```

```
mam15$invorder.tree  # original tree to sorted tree
```

```
##  t1  t2  t3  t4  t5  t6  t7  t8  t9 t10 t11 t12 t13 t14 t15
##   1   3   2   7   4   5   6   9  12  14  13  15  11  10   8
```

```
mam15$order.edge # sorted edge to original edge
```

```
##  E1  E2  E3  E4  E5  E6  E7  E8  E9 E10
##   2   1   4   3   5   7   6   9   8  10
```

```
mam15$invorder.edge # original edge to sorted edge
```

```
##  e1  e2  e3  e4  e5  e6  e7  e8  e9 e10
##   2   1   4   3   5   7   6   9   8  10
```

The p-values are calculated by the summary method.

```
mam15.pv <- summary(mam15)
mam15.pv$tree$value[1:5,] # p-values of the best 5 trees
```

```
##        raw        k.1        k.2       sk.1      sk.2      beta0     beta1
## T1 0.57191 0.56003732 0.74555293 0.12007465 0.3630207 -0.4058125 0.2547486
## T2 0.32259 0.30330296 0.46532132 0.60660592 0.7958229  0.3009803 0.2139440
## T3 0.03721 0.03693327 0.12835613 0.07386654 0.2043813  1.4608177 0.3266214
## T4 0.01338 0.01393287 0.08381999 0.02786575 0.1271972  1.7894990 0.4096722
## T5 0.03141 0.03148074 0.12705511 0.06296147 0.1983959  1.4999693 0.3595023
##         stat  shtest
## T1 -2.663776 0.94468
## T2  2.663776 0.80290
## T3  7.397870 0.57744
## T4 17.565482 0.17548
## T5 18.933775 0.14561
```

```
mam15.pv$edge$value[1:5,] # p-values of the best 5 edges
```

```
##        raw        k.1        k.2       sk.1      sk.2      beta0
## E1 0.93171 0.92988099 0.95599162 0.85976197 0.9030684 -1.5904294
## E2 0.58531 0.57943951 0.71832064 0.15887902 0.3374806 -0.3891599
## E3 0.32852 0.31769739 0.43463379 0.63539479 0.7740551  0.3193682
## E4 0.03737 0.03678201 0.12378829 0.07356403 0.1983152  1.4727856
## E5 0.05996 0.05961548 0.07488161 0.11923095 0.1430533  1.4991890
##         beta1
## E1 0.11552406
## E2 0.18870003
## E3 0.15477927
## E4 0.31652988
## E5 0.05882067
```

We make latex tables by the following code.

```
table2latex <- function(x) {
  rn <- rownames(x)
  cn <- colnames(x); cl <- length(cn)
  cat("\n\\begin{tabular}{",paste(rep("c",cl+1),collapse=""),"}\n",sep="")
  cat("\\hline\n")
  cat("&",paste(cn,collapse=" & "),"\\\\\n")
  for(i in seq(along=rn)) {
    cat(rn[i],"&",paste(x[i,],collapse=" & "),"\\\\\n")
  }
  cat("\\hline\n")
  cat("\\end{tabular}\n")
}
```

In the tree table below, the first two columns are computed by sbphylo: *stat* (log-likelihood difference), *shtest* (Shimodaira-Hasegawa test p-value). The other values are computed by the summary method: *k.1* (BP, bootstrap probability), *k.2* (AU, approximately unbiased p-value), *sk.2* (SI, selective inference p-value), *beta0* (β_0, signed distance), *beta1* (β_1, mean curvature), *edge* (the associated edges).

```
table2latex(mam15.pv$tree$character) # all the 15 trees
```

```
##
## \begin{tabular}{ccccccccc}
## \hline
## & stat & shtest & k.1 & k.2 & sk.2 & beta0 & beta1 & tree & edge \\
## T1 & -2.66 & 0.945 (0.001) & 0.560 (0.000) & 0.746 (0.001) & 0.363 (0.001) & -0.41 (0.00) & 0.25 (0.00) & ((Homsa,(Phovi,Bosta)),Orycu,(Musmu,Didvi)); & E1,E2 \\
## T2 &  2.66 & 0.803 (0.001) & 0.303 (0.000) & 0.465 (0.001) & 0.796 (0.001) &  0.30 (0.00) & 0.21 (0.00) & (Homsa,((Phovi,Bosta),Orycu),(Musmu,Didvi)); & E1,E3 \\
## T3 &  7.40 & 0.577 (0.002) & 0.037 (0.000) & 0.128 (0.002) & 0.204 (0.003) &  1.46 (0.01) & 0.33 (0.00) & (Homsa,Orycu,((Phovi,Bosta),(Musmu,Didvi))); & E1,E4 \\
## T4 & 17.57 & 0.175 (0.001) & 0.014 (0.000) & 0.084 (0.003) & 0.127 (0.003) &  1.79 (0.01) & 0.41 (0.01) & ((Homsa,(Phovi,Bosta)),(Orycu,Musmu),Didvi); & E2,E5 \\
## T5 & 18.93 & 0.146 (0.001) & 0.031 (0.000) & 0.127 (0.002) & 0.198 (0.003) &  1.50 (0.01) & 0.36 (0.00) & (Homsa,((Phovi,Bosta),(Orycu,Musmu)),Didvi); & E5,E6 \\
## T6 & 20.11 & 0.116 (0.001) & 0.005 (0.000) & 0.033 (0.002) & 0.052 (0.003) &  2.20 (0.02) & 0.36 (0.01) & (Homsa,(((Phovi,Bosta),Orycu),Musmu),Didvi); & E3,E6 \\
## T7 & 20.60 & 0.110 (0.001) & 0.015 (0.000) & 0.100 (0.003) & 0.149 (0.003) &  1.73 (0.01) & 0.44 (0.01) & (Homsa,(Orycu,Musmu),((Phovi,Bosta),Didvi)); & E5,E7 \\
## T8 & 22.22 & 0.074 (0.001) & 0.001 (0.000) & 0.012 (0.002) & 0.018 (0.002) &  2.71 (0.04) & 0.45 (0.02) & ((Homsa,Musmu),((Phovi,Bosta),Orycu),Didvi); & E3,E8 \\
## T9 & 25.38 & 0.033 (0.001) & 0.000 (0.000) & 0.001 (0.000) & 0.001 (0.000) &  3.71 (0.09) & 0.44 (0.04) & (((Homsa,(Phovi,Bosta)),Musmu),Orycu,Didvi); & E2,E9 \\
## T10 & 26.32 & 0.032 (0.001) & 0.002 (0.000) & 0.024 (0.002) & 0.036 (0.003) &  2.41 (0.02) & 0.43 (0.01) & ((Homsa,Musmu),Orycu,((Phovi,Bosta),Didvi)); & E7,E8 \\
## T11 & 28.86 & 0.017 (0.000) & 0.000 (0.000) & 0.005 (0.002) & 0.007 (0.002) &  3.13 (0.07) & 0.53 (0.04) & (Homsa,Orycu,(((Phovi,Bosta),Didvi),Musmu)); & E4,E7 \\
## T12 & 31.64 & 0.006 (0.000) & 0.000 (0.000) & 0.000 (0.000) & 0.001 (0.000) &  3.75 (0.10) & 0.43 (0.04) & (((Homsa,Musmu),(Phovi,Bosta)),Orycu,Didvi); & E8,E9 \\
## T13 & 31.75 & 0.006 (0.000) & 0.000 (0.000) & 0.000 (0.000) & 0.001 (0.000) &  3.95 (0.14) & 0.57 (0.07) & (Homsa,(((Phovi,Bosta),Musmu),Orycu),Didvi); & E6,E10 \\
## T14 & 34.74 & 0.002 (0.000) & 0.000 (0.000) & 0.000 (0.000) & 0.000 (0.000) &  4.57 (0.29) & 0.65 (0.09) & (Homsa,Orycu,(((Phovi,Bosta),Musmu),Didvi)); & E4,E10 \\
## T15 & 36.25 & 0.001 (0.000) & 0.000 (0.000) & 0.000 (0.000) & 0.000 (0.000) &  5.40 (0.34) & 0.39 (0.12) & ((Homsa,((Phovi,Bosta),Musmu)),Orycu,Didvi); & E9,E10 \\
## \hline
## \end{tabular}
```

```
table2latex(mam15.pv$edge$character) # all the 10 edges
```

```
##
## \begin{tabular}{cccccccc}
## \hline
## & k.1 & k.2 & sk.2 & beta0 & beta1 & edge & tree \\
## E1 & 0.930 (0.000) & 0.956 (0.001) & 0.903 (0.001) & -1.59 (0.00) & 0.12 (0.00) & ++++-- & T1,T2,T3 \\
## E2 & 0.579 (0.001) & 0.718 (0.001) & 0.337 (0.001) & -0.39 (0.00) & 0.19 (0.00) & +++--- & T1,T4,T9 \\
## E3 & 0.318 (0.000) & 0.435 (0.001) & 0.774 (0.001) &  0.32 (0.00) & 0.15 (0.00) & -+++-- & T2,T6,T8 \\
## E4 & 0.037 (0.000) & 0.124 (0.002) & 0.198 (0.002) &  1.47 (0.00) & 0.32 (0.00) & +--+-- & T3,T11,T14 \\
## E5 & 0.060 (0.000) & 0.075 (0.001) & 0.143 (0.002) &  1.50 (0.00) & 0.06 (0.00) & ---++- & T4,T5,T7 \\
## E6 & 0.038 (0.000) & 0.091 (0.002) & 0.155 (0.002) &  1.56 (0.01) & 0.22 (0.00) & -++++- & T5,T6,T13 \\
## E7 & 0.018 (0.000) & 0.069 (0.002) & 0.111 (0.003) &  1.79 (0.01) & 0.31 (0.01) & +--++- & T7,T10,T11 \\
## E8 & 0.003 (0.000) & 0.014 (0.001) & 0.023 (0.002) &  2.48 (0.02) & 0.27 (0.01) & +---+- & T8,T10,T12 \\
## E9 & 0.000 (0.000) & 0.000 (0.000) & 0.001 (0.000) &  3.69 (0.07) & 0.31 (0.03) & +++-+- & T9,T12,T15 \\
## E10 & 0.000 (0.000) & 0.000 (0.000) & 0.000 (0.000) &  4.09 (0.10) & 0.48 (0.04) & -++-+- & T13,T14,T15 \\
## \hline
## \end{tabular}
```

Visualization of the 15 trees (simple method)

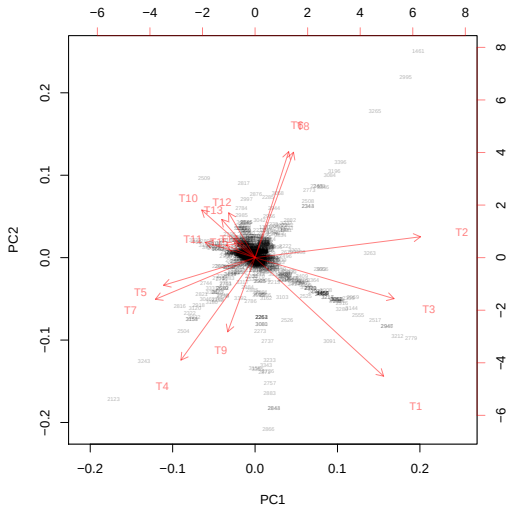
We use the following *mam15.mt* which is already used for computing the p-values for the 15 trees. It is a matrix of size $n \times m$, where $n = 3414$ is sample size and $m = 15$ is the number of bifurcating trees. We sort trees by the likelhiood value.

```
mt15 <- mam15.mt[,mam15$order.tree] # Now T1, T2, ...
colnames(mt15) <- names(mam15$order.tree)
dim(mt15)
```

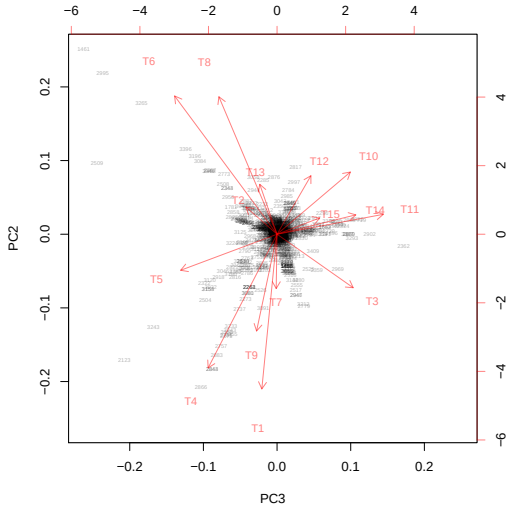
```
## [1] 3414   15
```

The matrix consists of the site-wise log-likelihood of tree i at site t as $\xi_{ti} = \log p_i(\boldsymbol{x}_t|\hat{\boldsymbol{\theta}}_i)$, $t = 1, \ldots, n$, $i = 1, \ldots, m$. The matrix is written as $(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_m)$ for the vectors $\boldsymbol{\xi}_i \in \mathbb{R}^n$, $i = 1, \ldots, m$. We compute the mean vector as $\bar{\boldsymbol{\xi}} = \sum_{i=1}^m \boldsymbol{\xi}_i/m$. Then models are represented by $\boldsymbol{a}_i := \boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}$. We apply PCA to $\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m)$ and biolot for visualization.

```
mt0 <- apply(mt15,1,mean) # using the mean vector
xx <- mt15 - mt0 # centering for rows
f <- prcomp(xx,center=FALSE,scale=FALSE)
f$x[,1] <- -f$x[,1];f$rotation[,1] <- -f$rotation[,1] # change the sign of PC1
biplot(f,cex=c(0.4,0.8),choices=c(1,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5)) ) # (PC1, PC2)
```




```
biplot(f,cex=c(0.4,0.8),choices=c(3,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5)) ) # (PC3, PC2)
```



Reconstructiong the full model X from the submodels (using 10+1 trees)

We use *mam26.mt* instead of *mam15.mt*. First we check the association mapping *mam26.ass*.

```
names(mam26.ass) # trees and edges
```

```
##  [1] "t1"  "t2"  "t3"  "t4"  "t5"  "t6"  "t7"  "t8"  "t9"  "t10" "t11"
## [12] "t12" "t13" "t14" "t15" "t0"  "ta"  "tb"  "tc"  "td"  "te"  "tf"
## [23] "tg"  "th"  "ti"  "tj"  "e1"  "e2"  "e3"  "e4"  "e5"  "e6"  "e7"
## [34] "e8"  "e9"  "e10"
```

```
attr(mam26.ass,"trees") # trees
```

```
##  [1] "t1"  "t2"  "t3"  "t4"  "t5"  "t6"  "t7"  "t8"  "t9"  "t10" "t11"
## [12] "t12" "t13" "t14" "t15" "t0"  "ta"  "tb"  "tc"  "td"  "te"  "tf"
## [23] "tg"  "th"  "ti"  "tj"
```

```
attr(mam26.ass,"edges") # edges
```

```
##  [1] "e1"  "e2"  "e3"  "e4"  "e5"  "e6"  "e7"  "e8"  "e9"  "e10"
```

t1, t2, . . . , t15 are the 15 bifurcaing trees. t0 is the star topology. ta, tb, . . . , tj are partially resolved trees. I want to know the correspondance between ta, tb, . . . , tj and e1, e2, . . . , e10.

```
(e <- attr(mam26.ass,"edges")) # edge names
```

```
##  [1] "e1"  "e2"  "e3"  "e4"  "e5"  "e6"  "e7"  "e8"  "e9"  "e10"
```

```
mam26.ass[e]  # only values > 15 are for ta, ..., tj
```

```
## $e1
## [1]  1  5  8 21
```

```
##
## $e2
## [1]  1  2  3 17
##
## $e3
## [1]  2 10 13 24
##
## $e4
## [1]  3  7 15 25
##
## $e5
## [1]  4  5  6 22
##
## $e6
## [1]  4 13 14 18
##
## $e7
## [1]  6  7 11 19
##
## $e8
## [1]  8  9 12 20
##
## $e9
## [1]  9 14 15 26
##
## $e10
## [1] 10 11 12 23
```

```
(b <- sapply(mam26.ass[e], function(a) a[a>15])) # extract values > 15
```

```
##  e1  e2  e3  e4  e5  e6  e7  e8  e9 e10
##  21  17  24  25  22  18  19  20  26  23
```

```
e2t <- names(mam26.ass)[b] # the correspondance: e1, e2, ... -> ta, tb, ...
names(e2t) <- e
e2t
```

```
##   e1   e2   e3   e4   e5   e6   e7   e8   e9  e10
## "te" "ta" "th" "ti" "tf" "tb" "tc" "td" "tj" "tg"
```

```
E2t <- e2t[mam15$order.edge ] # the correspondance: E1, E2, ... -> ta, tb, ...
names(E2t) <- names(mam15$order.edge)
E2t
```

```
##   E1   E2   E3   E4   E5   E6   E7   E8   E9  E10
## "ta" "te" "ti" "th" "tf" "tc" "tb" "tj" "td" "tg"
```

Extract the site-wise log-likelihoods for trees. Here *mt15* is the matrix for the 15 bifurcating trees ($\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{15}$). In the below, we treat the clade (cow, seal) as a leaf of the tree. *mt0* is the vector for the star topology ($\boldsymbol{\eta}_0$). *mt10* is for the matrix for 10 partially resolved trees; they correspond to the 10 internal edges ($\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{10}$). We need *E2t* for getting the correct mapping from the edges to the partially observed trees.

```
mt15<- mam26.mt[,mam15$order.tree]
(colnames(mt15) <- names(mam15$order.tree)) # Now T1, T2, ...
```

```
##  [1] "T1"  "T2"  "T3"  "T4"  "T5"  "T6"  "T7"  "T8"  "T9"  "T10" "T11"
## [12] "T12" "T13" "T14" "T15"
```

```
mt0 <- mam26.mt[,"t0"] # star topology
mt10<- mam26.mt[,E2t] # partialy resolved trees
(colnames(mt10) <- names(E2t)) # Now E1, E2, ...
```

```
##  [1] "E1"  "E2"  "E3"  "E4"  "E5"  "E6"  "E7"  "E8"  "E9"  "E10"
```

We subtract the vector of star-topology from all the other vectors of trees. $\boldsymbol{a}_i := \boldsymbol{\xi}_i - \boldsymbol{\eta}_0$, $i = 1, \ldots, 15$. $\boldsymbol{b}_i := \boldsymbol{\eta}_i - \boldsymbol{\eta}_0$, $i = 1, \ldots, 10$. Then the vector for the full model X is obtained as

$$\boldsymbol{a}_X := \boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{d},$$

where $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{10})$ and $\boldsymbol{d} = (\|\boldsymbol{b}_1\|^2, \ldots, \|\boldsymbol{b}_{10}\|^2)^T$. This is computed by the following code.

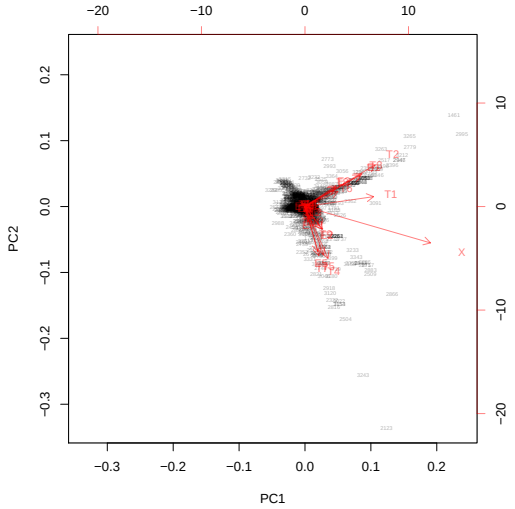
```
## Shimodaira 2001 Comm Stat A
fullmodel <- function(B,dim=NULL) {
  d <- apply(B,2,function(a) sum(a*a))
  s <- svd (B)
  if(is.null(dim)) {
    dd <- diag(1/s$d)
  } else {
    dd <- diag(c(1/s$d[1:dim],rep(0,length(s$d)-dim)))
  }
  x <- s$u %*% dd %*% t(s$v) %*% d
  colnames(x) <- "X"
  x
}
```

Then simply run below.

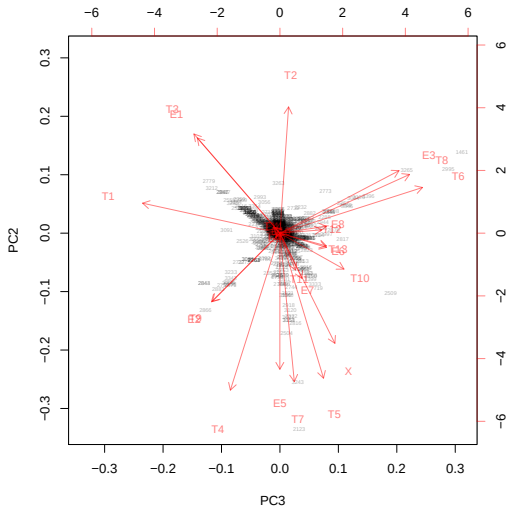
```
ax <- fullmodel(mt10 - mt0)
```

Now look at model map.

```
xx <- cbind(ax,mt15-mt0,mt10-mt0)
f <- prcomp(xx,center=FALSE,scale=FALSE)
biplot(f,cex=c(0.4,0.8),choices=c(1,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5)) ) # (PC1, PC2)
```




```
biplot(f,cex=c(0.4,0.8),choices=c(3,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5)) ) # (PC3, PC2)
```



3d plot of the model map

We need rgl packages for visualization in 3d.

```
library(rgl)
#knitr::knit_hooks$set(rgl = hook_webgl)
rgl::setupKnitr()
```

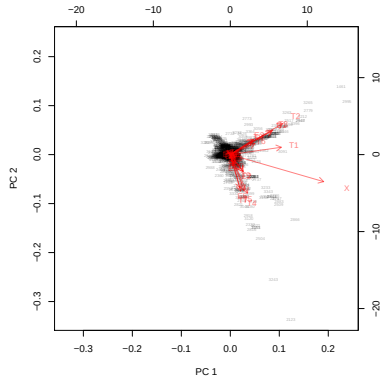
Also prepare in-house version of pca and biplot.

```
## singular value decomposition with dimnames
mysvd <- function(x) {
  s <- svd(x)
  napc <- paste("PC",seq(along=s$d))
  dimnames(s$u) <- list(dimnames(x)[[1]],napc)
  names(s$d) <- napc
  dimnames(s$v) <- list(dimnames(x)[[2]],napc)
  s
}

## principal component analysis
mypca <- function(dat) {
```

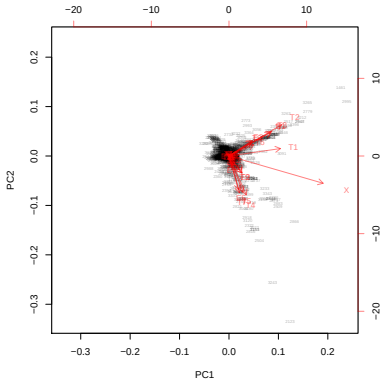
```
  s <- mysvd(dat)
  n <- dim(s$u)[1]
  m <- dim(s$v)[1]
  k <- length(s$d)
#  x <- s$u %*% diag(s$d)
#  y <- s$v %*% diag(s$d)
  x <- s$u * rep(s$d,rep(n,k))
  y <- s$v * rep(s$d,rep(m,k))
#  sdev <- s$d/sqrt(n)
#  loadings <- s$v
#  scores <- x
#  shuseibun.fuka = y
#  shuseibun.tokuten = s$u
  list(x=x,y=y,d=s$d,u=s$u,v=s$v)
}

## biplot
## (example)
## p <- mypca(scale(x)) or mypca(x)
## mybiplot(p$u,p$y) # biplot(scale=1) default
## mybiplot(p$x,p$v) # biplot(scale=0)
mybiplot <- function(x,y,choices=1:2,mag=c(1,1),
                     col.arg=c(1,2),cex.arg=c(1,1),lim.mag=1,
                     xadj.arg=c(0.5,0.5),yadj.arg=c(0.5,0.5),
                     arrow.len=0.1,xnames=NULL,ynames=NULL) {
  if(length(choices) != 2) stop("choices must be length 2")
  if(length(mag) != 2) stop("mag must be length 2")
 # x <- x[,choices] %*% diag(mag)
 # y <- y[,choices] %*% diag(1/mag)
  x <- x[,choices] * rep(mag,rep(dim(x)[1],2))
  y <- y[,choices] * rep(1/mag,rep(dim(y)[1],2))
  if(is.null(xnames)) nx <- dimnames(x)[[1]]
  else nx <- as.character(xnames)
  if(is.null(ynames)) ny <- dimnames(y)[[1]]
  else ny <- as.character(ynames)
  nd <- dimnames(x)[[2]]
  rx <- range(x)
  ry <- range(y)
  oldpar <- par(pty="s")
  a <- min(rx/ry)
  yy <- y*a
  plot(x,xlim=rx*lim.mag,ylim=rx*lim.mag,type="n",xlab=nd[1],ylab=nd[2])
  ly <- pretty(rx/a)
  ly[abs(ly) < 1e-10] <- 0
  axis(3,at = ly*a ,labels = ly)
  axis(4,at = ly*a ,labels = ly)
  text(x,nx,col=col.arg[1],cex=cex.arg[1],adj=yadj.arg)
  text(yy,ny,col=col.arg[2],cex=cex.arg[2],adj=yadj.arg)
  arrows(0,0,yy[,1]*0.8,yy[,2]*0.8,col=col.arg[2],length=arrow.len)
  par(oldpar)
  invisible(list(x=x,y=y))
}
## biplot3d
## requires(plot3d)
## (example)
## p <- mypca(scale(x)) or mypca(x)
## mybiplot3d(p$u,p$y) # biplot(scale=1) default
## mybiplot3d(p$x,p$v) # biplot(scale=0)
mybiplot3d <- function(x,y,choices=1:3,col.arg=c(1,2),alpha.arg=c(1,1),cex.arg=c(1,1),lwd=1) {
  x <- x[,choices]
  y <- y[,choices]
  xn <- rownames(x)
  if(is.null(xn)) xn <- seq(length=nrow(x))

  plot3d(rbind(x,y),type="n")

  save <- material3d("color")
  material3d(col.arg[1],alpha.arg[1])
  text3d(x,texts=xn,col=col.arg[1],cex=cex.arg[1])
  material3d(col.arg[2],alpha.arg[2])
  coords <- NULL
  for (i in 1:nrow(y)) {
    coords <- rbind(coords, rbind(c(0,0,0),y[i,]))
  }
  lines3d(coords, col=col.arg[2],cex=cex.arg[2], lwd=lwd)

  text3d(1.1*y, texts=rownames(y), col=col.arg[2],cex=cex.arg[2])
  material3d(save)
}
```

We check my in-house version pca produces the same output as the original pca.

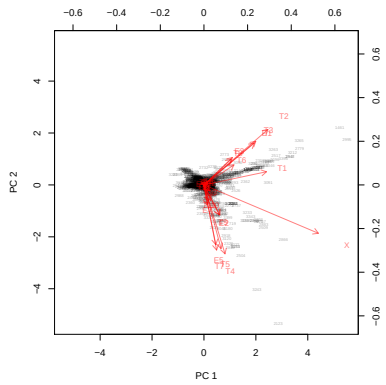
```
p <- mypca(xx) # in-house version of pca
f <- prcomp(xx,center=FALSE,scale=FALSE)  # standard pca
mybiplot(p$u,p$y, cex=c(0.4,0.8),choices=c(1,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5))) # same as default of biplot (scale=1)
```



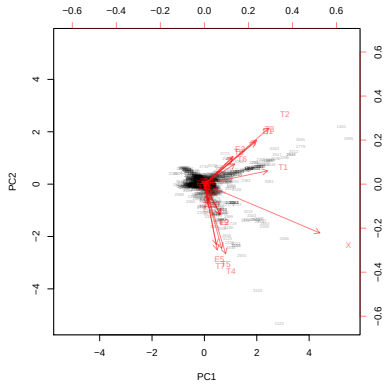

```
biplot(f, cex=c(0.4,0.8),choices=c(1,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5))) # same as above
```




```
mybiplot(p$x,p$v, cex=c(0.4,0.8),choices=c(1,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5))) # same as biplot with scale=0
```



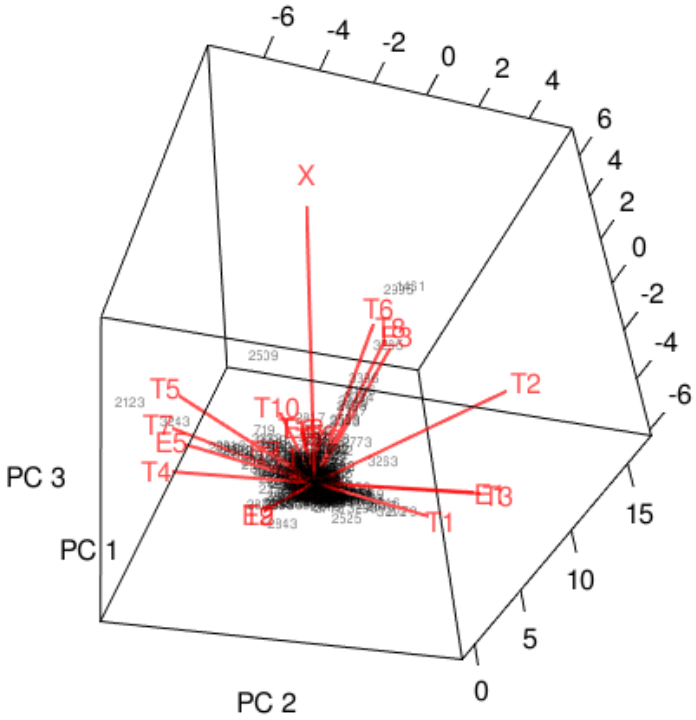

```
biplot(f, scale=0,cex=c(0.4,0.8),choices=c(1,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5))) # same as above
```




```
## 3d plotting
um=structure(c(0.352101027965546, 0.606027662754059, 0.713270783424377,
0, 0.931618392467499, -0.153592959046364, -0.329387068748474,
0, -0.0900644063949585, 0.780474007129669, -0.618666768074036,
0, 0, 0, 0, 1), .Dim = c(4L, 4L))
mybiplot3d(p$u*p$d[1],p$y,lwd=2,cex=c(0.5,1),alpha=c(0.5,0.7))
```

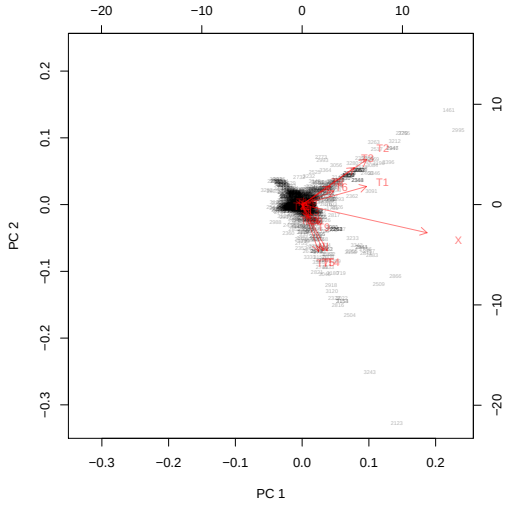
```
## NULL
```

```
#um = par3d("userMatrix"); dput(um)
par3d(userMatrix=um)
par3d(windowRect=c(0,45,600,600))
```

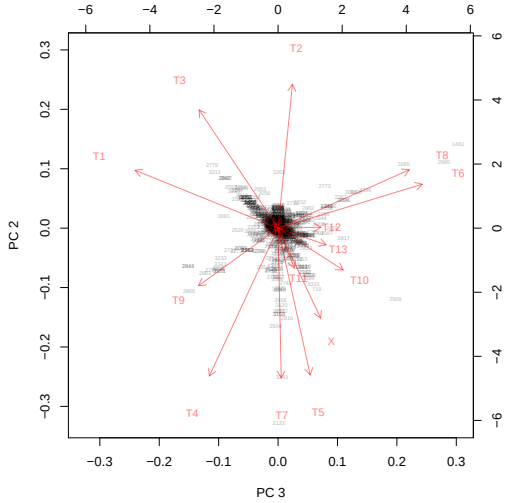


plot only 15 trees and full model

```
xx <- cbind(ax,mt15-mt0)
p <- mypca(xx) # in-house version of pca
mybiplot(p$u,p$y, cex=c(0.4,0.8),choices=c(1,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5))) # (PC1, PC2)
```



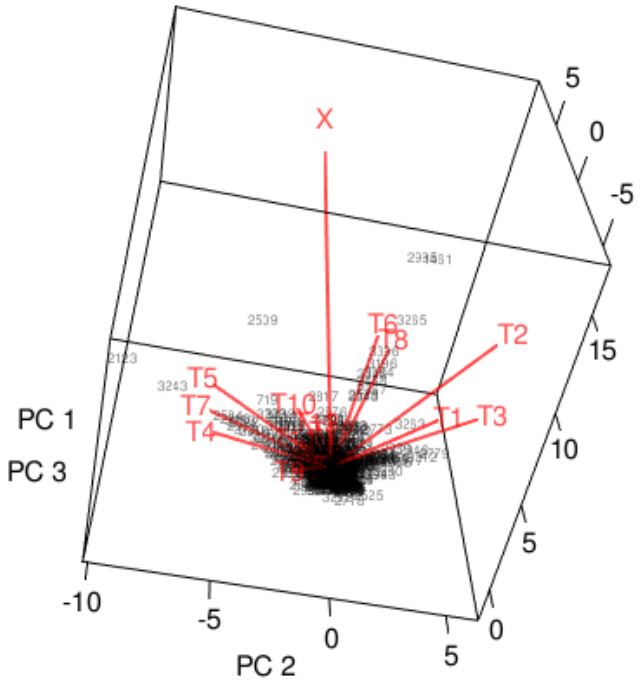

```
mybiplot(p$u,p$y, cex=c(0.4,0.8),choices=c(3,2),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5))) # (PC3, PC2)
```




```
## 3d plotting
um=structure(c(0.228986993432045, 0.835189700126648, 0.500022709369659,
0, 0.972387254238129, -0.172497615218163, -0.157184407114983,
0, -0.0450262203812599, 0.522209227085114, -0.851627767086029,
0, 0, 0, 0, 1), .Dim = c(4L, 4L))
mybiplot3d(p$u*p$d[1]*1.4,p$y,lwd=2,cex=c(0.5,1),alpha=c(0.5,0.7))
```

```
## NULL
```

```
#um = par3d("userMatrix"); dput(um)
par3d(userMatrix=um)
par3d(windowRect=c(0,45,600,600))
#rgl.postscript("20190114_map3d.pdf", fmt='pdf')
#rgl.close()
```



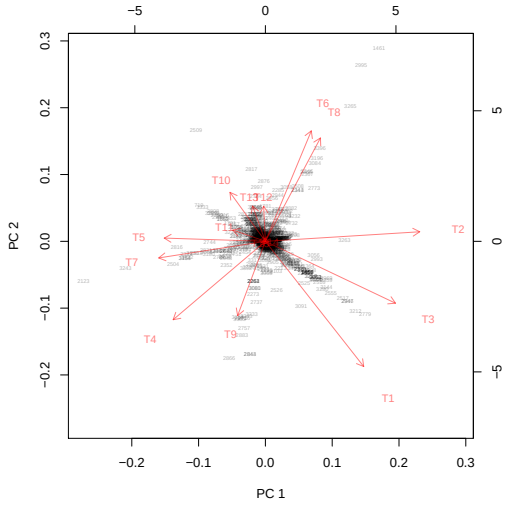
top-view of the 3d plot

The top-view of the pca in 3d is computed by projecting the points $\boldsymbol{a}_i$ to the space orthogonal to $\boldsymbol{a}_X$.

```
## projection to the orthogonal space
topview <- function(ax,A) {
  ax <- drop(ax) # strip dim
  ux <- ax/sqrt(sum(ax^2)) # unit vector
  pp <- ux %*% (t(ux) %*% A) # projection to ux
  A - pp
}
```

Now draw top-view of the previous pca3d.

```
yy <- topview(ax,mt15-mt0)
p <- mypca(yy) # in-house version of pca
mybiplot(p$u,p$y, cex=c(0.4,0.8),choices=c(1,2), mag=c(-1,1),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5))) # (PC1, PC2)
```



Reconstructiong the full model X from the submodels (using 15+1 trees)

In the previous reconstruction, we used 10+1 trees. The MLEs for the 10 partially resolved trees are usually not computed for model comparisons. Instead, here we use the 15 bifurcating trees, for which MLEs are already computed for tree selection. But we need the MLE for the star topology, anyway.

Since we know the dimension spanned by the 10 trees is 10, so specify *dim=10* below.

```
ax2 <- fullmodel(mt15 - mt0, dim=10)
```

Draw model map of the 15 trees and the full model.

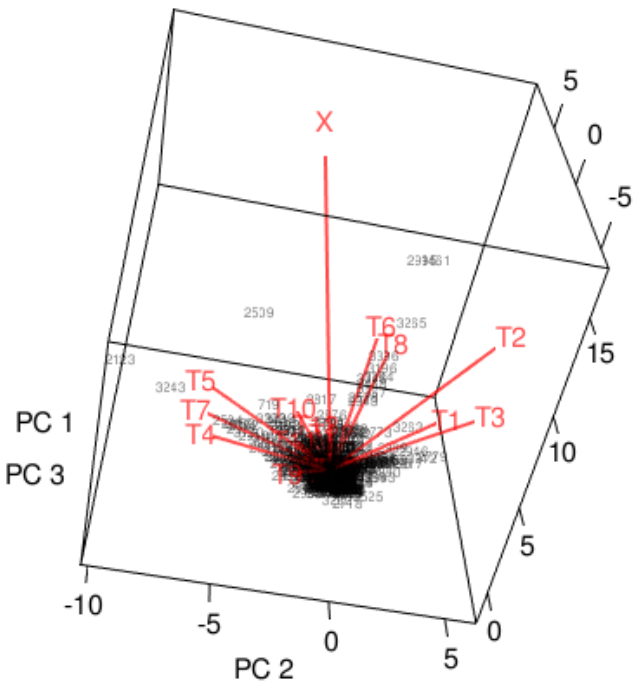
```
xx <- cbind(ax2,mt15-mt0)
p <- mypca(xx) # in-house version of pca
```

pca3d

```
## 3d plotting
um=structure(c(0.228986993432045, 0.835189700126648, 0.500022709369659,
0, 0.972387254238129, -0.172497615218163, -0.157184407114983,
0, -0.0450262203812599, 0.522209227085114, -0.851627767086029,
0, 0, 0, 0, 1), .Dim = c(4L, 4L))
mybiplot3d(p$u*p$d[1]*1.4,p$y,lwd=2,cex=c(0.5,1),alpha=c(0.5,0.7))
```

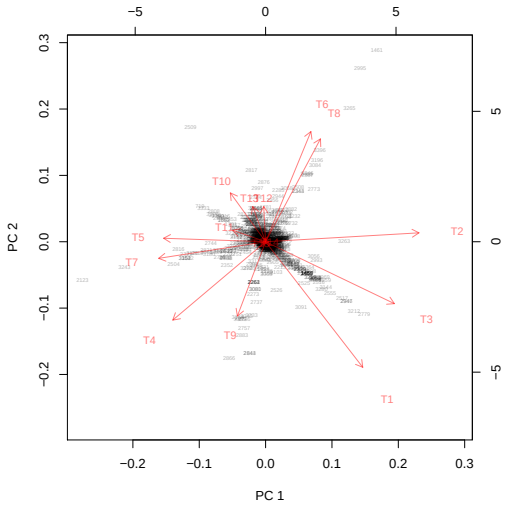
```
## NULL
```

```
#um = par3d("userMatrix"); dput(um)
par3d(userMatrix=um)
par3d(windowRect=c(0,45,600,600))
```



Now draw top-view of the previous pca3d.

```
yy <- topview(ax2,mt15-mt0)
p <- mypca(yy) # in-house version of pca
mybiplot(p$u,p$y, cex=c(0.4,0.8),choices=c(1,2), mag=c(-1,1),col=c(rgb(0,0,0,alpha=0.3),rgb(1,0,0,alpha=0.5))) # (PC1, PC2)
```



As you seen, visualization with ax2 (reconstruction by 15+1 trees) is almost identical to that of ax (reconstruction by 10+1 trees).