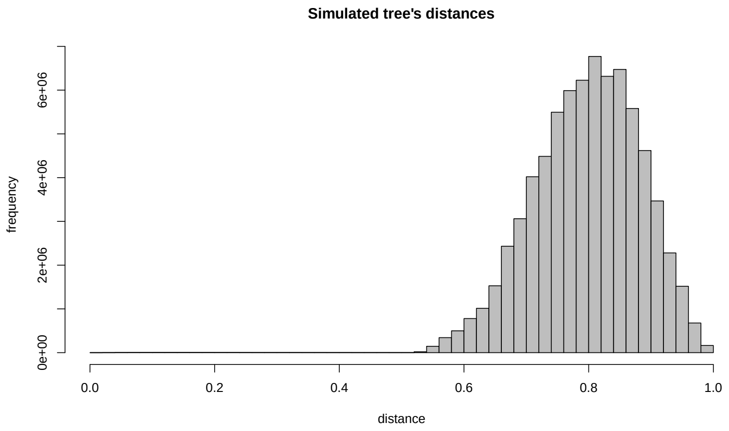
Simulated tree clustered by UCSD soft. - Feb 2016

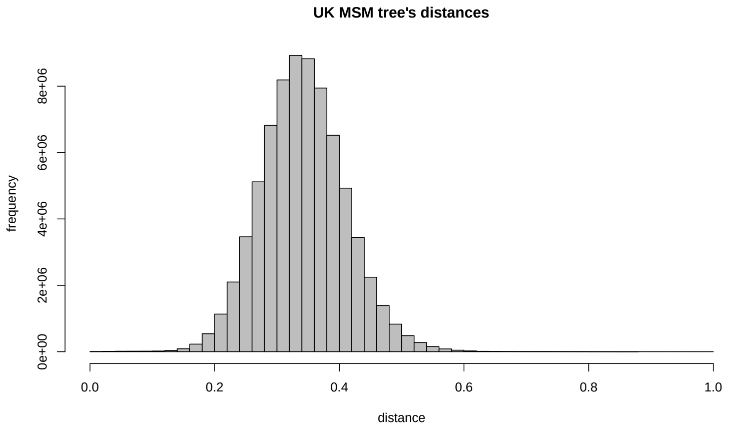
S. Le Vu

(Dated: February 25, 2016)

I. INTRO

- "time-based" distances have been extracted form simulated coalescent tree
- distances normalized from 0 to 1





II. UCSD HIVCLUSTERING

Read saved results from UCSD hivclustering

```
### only cluster members
simclus <- readRDS(file = "data/simclus.rds")[[3]]
ukclus <- readRDS(file = "data/ukclus.rds")[[3]]
```

To construct clusters, threshold were determined by quantiles of distances (0.05%, 0.1%, 1% and 10%). For simulated and UK trees

```
## read saved results of UCSD clustering
readRDS(file = "data/simclus.rds")[[1]]
```

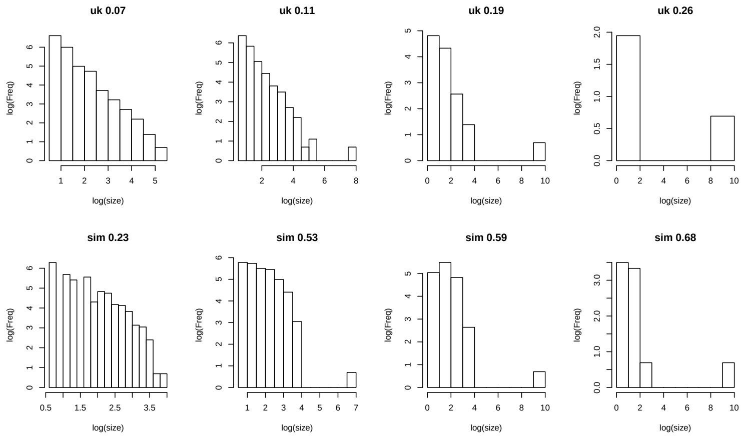
```
    0.05%      0.1%        1%       10%       25%       50% 
0.2340219 0.5334507 0.5870679 0.6843790 0.7404369 0.8025368 

readRDS(file = "data/ukclus.rds")[[1]]

     0.05%       0.1%         1%        10%        25%        50% 
0.06709549 0.10967080 0.19304638 0.25869004 0.29701520 0.34058792 
```

Number of clusters and stats for simulated and UK trees (cluster size 1 does not exist in these outputs)

```
##- Calculate size(=Freq) of each cluster across different threshold
simfreqClust <- lapply(simclus,
                       function(x) as.data.frame(table(x$ClusterID),
                       stringsAsFactors = FALSE))
ukfreqClust <- lapply(ukclus,
                      function(x) as.data.frame(table(x$ClusterID),
                      stringsAsFactors = FALSE))

##- number of different clusters by threshold
sapply(simfreqClust, function(x) dim(x)[1])

0.23 0.53 0.59 0.68 
1848 1357  529   61 

sapply(ukfreqClust, function(x) dim(x)[1])

0.07 0.11 0.19 0.26 
1490 1261  213    7 

##- cluster size
sapply(simfreqClust, function(x) summary(x$Freq))

          0.23    0.53    0.59    0.68
Min.     2.000   2.000    2.00     2.0
1st Qu.  2.000   3.000    2.00     2.0
Median   4.000   5.000    4.00     2.0
Mean     5.924   8.565   22.49   198.2
3rd Qu.  7.250  10.000    8.00     4.0
Max.    47.000 989.000 8766.00 11900.0

sapply(ukfreqClust, function(x) summary(x$Freq))

           0.07     0.11     0.19  0.26
Min.      2.000    2.000     2.00     2
1st Qu.   2.000    2.000     2.00     2
Median    3.000    3.000     2.00     2
Mean      5.117    7.427    54.67  1733
3rd Qu.   4.000    5.000     3.00     3
Max.    162.000 2235.000 10900.00 12110
```

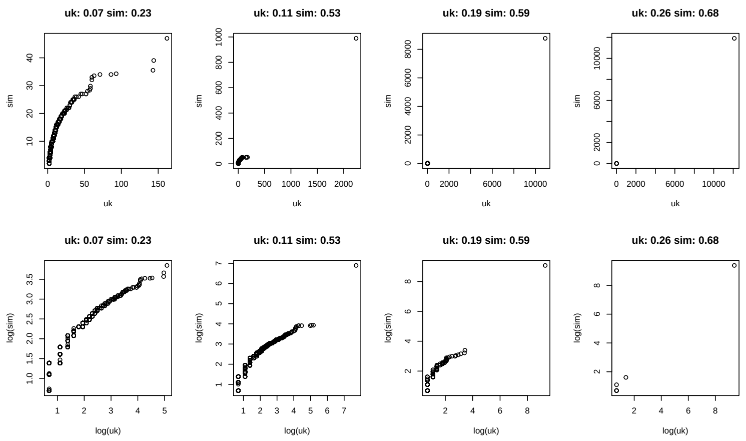
Plots of log(size) for UK and simulated clusters

```
##- distr of cluster sizes: log(x) and log(y)
## how many plots
a <- length(simfreqClust)
b <- length(ukfreqClust)

par(mfcol=c(2, max(a, b)))
for (i in 1:max(a, b)){
  h <- hist(log(ukfreqClust[[i]]$Freq), plot = F)
  h$counts <- log1p(h$counts) # log(y)
  plot(h, ylab = "log(Freq)",
       main = paste("uk", names(ukfreqClust)[i]),
       xlab = "log(size)")

  h <- hist(log(simfreqClust[[i]]$Freq), plot = F)
  h$counts <- log1p(h$counts) # log(y)
  plot(h, ylab = "log(Freq)",
          main = paste("sim", names(simfreqClust)[i]),
          xlab = "log(size)")

}
```



QQ plots UK vs simulated, untransformed and log-log

```
par(mfcol=c(2, max(a, b)))
for (i in 1:max(a, b)){
  qqplot(ukfreqClust[[i]]$Freq,
         simfreqClust[[i]]$Freq,
         main = paste("uk:", names(ukfreqClust)[i],
                      "sim:", names(simfreqClust)[i]),
         xlab = "uk", ylab = "sim")

  qqplot(log(ukfreqClust[[i]]$Freq),
         log(simfreqClust[[i]]$Freq),
         main = paste("uk:", names(ukfreqClust)[i],
                      "sim:", names(simfreqClust)[i]),
         xlab = "log(uk)", ylab = "log(sim)")

}
dev.off()

null device
          1
```



III. ASSOCIATIONS

After merging with co-variates allocated from demes states, non-clustering individuals are assigned a cluster size of 1. The proportion of individuals into clusters and stats for "size of cluster for each individuals"

```
##-proportion in or out clusters
sapply(l, function(x) round(prop.table(table(x$binclus)),2))

  0.23 0.53 0.59 0.68
0  0.1 0.04 0.02 0.01
1  0.9 0.96 0.98 0.99

##- cluster sizes (by individuals having such a size !!)
sapply(l, function(x) summary(x$size))

          0.23   0.53 0.59  0.68
Min.     1.000   1.00    1     1
1st Qu.  3.000   6.00   20 11900
Median   8.000  13.00 8766 11900
Mean     9.889  93.94 6320 11640
3rd Qu. 14.000  25.00 8766 11900
Max.    47.000 989.00 8766 11900
```

```
[1] "coucou"
```

To sort out the dependency between indivduals from same cluster

1. "downsample" to make analysis of each cluster size explained by mean of each co-variate (from here, only clusters from lower and higher threshold represented)

   ```
   ##- 1. down-sample: mean of each variable
   ## just on low and high threshold
   l <- listclus[c(1,length(listclus))]
   down <- lapply(l, function(x) aggregate(x[, 5:9], list("size" = x$size), mean))
   # str(down)
   #
   ##- linear regression
   lm_model_std = "scale(size) ~ scale(age) + scale(stage) + scale(time) + scale(risk)"
   lapply(down, function(x) summary(lm(lm_model_std, data = x)))

   $`0.23`
   ```

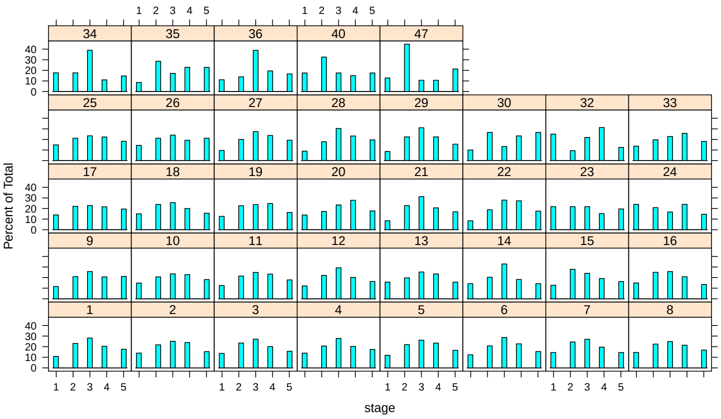
```
Call:
lm(formula = lm_model_std, data = x)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.49472 -0.73601 -0.00004  0.65872  2.36077 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.563e-16  1.698e-01   0.000    1.000
scale(age)    1.646e-01  1.753e-01   0.939    0.355
scale(stage) -1.411e-01  1.847e-01  -0.764    0.451
scale(time)  -1.287e-01  1.764e-01  -0.730    0.471
scale(risk)   9.152e-02  1.816e-01   0.504    0.618

Residual standard error: 1.033 on 32 degrees of freedom
Multiple R-squared:  0.05204,Adjusted R-squared:  -0.06645 
F-statistic: 0.4392 on 4 and 32 DF,  p-value: 0.7793


$`0.68`

Call:
lm(formula = lm_model_std, data = x)

Residuals:
       1        2        3        4        5        6        7        8        9 
-0.57885 -0.43164 -0.37925 -0.26120 -0.49761 -0.38620  0.02483 -0.04741  2.55732 
attr(,"scaled:center")
[1] 1327
attr(,"scaled:scale")
[1] 3965

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.508e-16  4.617e-01   0.000    1.000
scale(age)   2.264e-01  8.785e-01   0.258    0.809
scale(stage) 2.173e-01  9.616e-01   0.226    0.832
scale(time)  6.744e-02  5.298e-01   0.127    0.905
scale(risk)  3.001e-01  8.506e-01   0.353    0.742

Residual standard error: 1.385 on 4 degrees of freedom
Multiple R-squared:  0.04088,Adjusted R-squared:  -0.9182 
F-statistic: 0.04262 on 4 and 4 DF,  p-value: 0.9951
```

Not any significant association !!

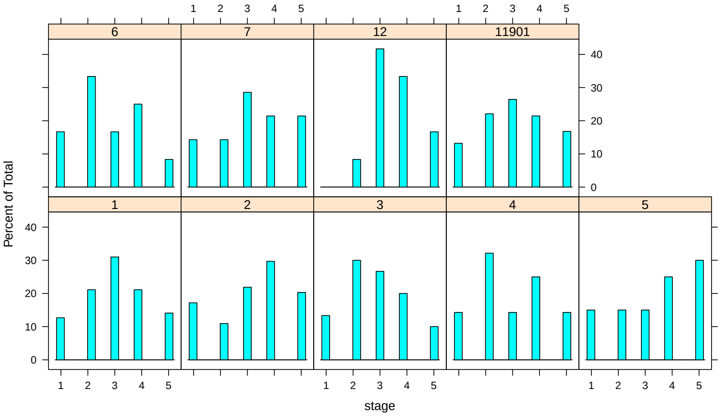
2. plot the distribution of covariates by cluster size

```r
##- 2. plots
library(lattice)
# trellis.par.set(canonical.theme(color = FALSE))
for(i in 1:length(l)){
  print(histogram(~ stage|factor(size),
            main = paste("distribution of stages by cluster sizes at threshold =", names(l[i])),
            data = l[[i]])
  )
}
```

distribution of stages by cluster sizes at threshold = 0.23



distribution of stages by cluster sizes at threshold = 0.68



What to tell ??