

**R software tutorial:  
Random Forest Clustering Applied to Renal Cell Carcinoma  
Steve Horvath and Tao Shi**

Correspondence: shorvath@mednet.ucla.edu  
Department of Human Genetics and Biostatistics  
University of California, Los Angeles, CA 90095-7088, USA.

In this R software tutorial we describe some of the results underlying the following article.

- *Shi T, Seligson D, Belldegrun AS, Palotie A, Horvath S. (2005) Tumor classification by tissue microarray profiling: random forest clustering applied to renal cell carcinoma. Mod Pathol. 2005 Apr;18(4):547-57*

**Additional References**

General intro to random forest

- Breiman L. Random forests. *Machine Learning* 2001;45(1):5-32.
- L. Breiman and Adele Cutler's random forests: <http://stat-www.berkeley.edu/users/breiman/RandomForests/>

The following article describes theoretical studies of RF clustering.

- Tao Shi and Steve Horvath (2006) Unsupervised Learning with Random Forest Predictors. *Journal of Computational and Graphical Statistics*. Volume 15, Number 1, March 2006, pp. 118-138(21)

The following reference describes the R implementation of random forests

- Liaw A. and Wiener M. Classification and Regression by randomForest. *R News*, 2(3):18-22, December 2002.

The tutorial and data can be found at the following webpage.

<http://www.genetics.ucla.edu/labs/horvath/kidneypaper/RCC.htm>

The following webpage contains additional, theoretical material

<http://www.genetics.ucla.edu/labs/horvath/RFclustering/RFclustering.htm>

```
## The following tutorial shows how to carry out RF clustering
## using the freely available software R
## (http://cran.r-project.org/). Before running it, you need to install
## the randomForest library, which is a contributed package in R.
## RF clustering takes 3 parameters:
## 1) number of features sampled at each split
## 2) number of forests
## 3) number of trees in each forest.
## They are implemented in the function RFdist as the options: mtry1, no.rep,
## and no.tree, respectively.
## The file FunctionsRFclustering.txt also contains other relevant functions
## such as "Rand" for the Rand index.
```

```
#####R Software and Inputting the Data#####
## 1) To install the R software, go to http://www.R-project.org
## 2) After installing R, you need to install two additional R packages: randomForest and Hmisc
## Open R and go to menu "Packages\Install package(s) from CRAN", then
## choose randomForest. R will automatically
## install the package. When asked "Delete downloaded files (y/N)? ", answer "y".
## 3) Download the zip file containing:
## a) R function file: "FunctionsRFclustering.txt", which contains several
## R functions needed for RF clustering and results assessment
## b) A test data file: "testData.csv"
## c) MDS coordinate file: "cmd1.csv"
## d) The tutorial file: "RFclusteringTutorial.txt"
## 4) Unzip all the files into the same directory.
## 5) Open the R software by double clicking its icon.
## 6) Open the tutorial file "RFclusteringTutorialRenalCancer.doc" in
## Microsoft Word or an alternative text editor.
## 7) Copy and paste the R commands from the tutorial into the R session.
## Comments are preceded by "#" and are automatically ignored by R.
```

```
# set the working directory (where the data are located)
setwd("C:/Documents and Settings/shorvath/My
Documents/ADAG/TaoShi/RFclustering/TutorialRCC366")
```

```
## load the library and ignore the warning message
source("FunctionsRFclustering.txt")
```

```

# Here we read in the data from all 366 patients

dat1 = read.table("testData_366.csv", sep=",", header=T, row.names=1)
## This is the input file for RF clustering algorithm
## the first 8 columns contain the tumor marker data. Rows correspond to the patients
##i.e. the objects that will be clustered.
datRF = dat1[,1:8]

names(datRF)
[1] "Marker1" "Marker2" "Marker3" "Marker4" "Marker5" "Marker6" "Marker7"
"Marker8"

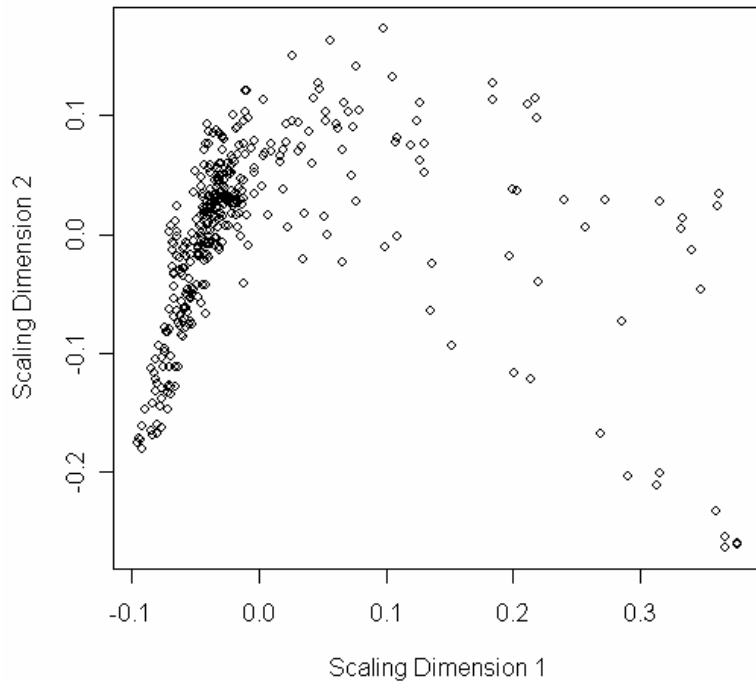
## Calculating RF distance between samples based on the 8 marker measurements
## This will take quite long time depends how
## many tree and repetitions you choose
## We suggest to use relatively large number of forests
## with large number of trees
distRF = RFdist(datRF, mtry1=3, 4000, 100, addcl1=T,addcl2=F,imp=T, oob.prox1=T)

## Classical multidimensional scaling based on RF distance
# we use 2 scaling dimensions.
cmd1 = cmdscale(as.dist(distRF$c11),2)

## Due to the randomness of the RF clustering algorithm, you may get slightly different results
## To save time, we provide the coordinates of cmd1 in the file "cmd1_366.csv",
## so we just read it in directly
cmd1 = as.matrix(read.csv("cmd1_366.csv", header=T, row.names=1))
# The above line should be removed when you run the code on a new data set.

```

```
plot(cmd1,xlab="Scaling Dimension 1",ylab="Scaling Dimension 2")
```



We also make use of an alternative partitioning around medoid function "pamNew" pamNew corrects the clustering membership assignment by taking account of the silhouette strength which is standard output of pam. Specifically, the clustering membership of an observation with a negative silhouette strength is reassigned to its neighboring cluster.

```
#In the following, we will use PAM clustering on the 2 scaling coordinates  
## PAM clustering based on the scaling coordinates  
RFclusterLabel = pamNew(cmd1, 2)
```

Comment for statisticians: This last step requires an explanation since it is rather unusual. A standard RF clustering analysis would use pam clustering in conjunction with the original RF dissimilarity without using scaling coordinates. Using the scaling dimensions to process the RF dissimilarity amounts to using a reduced form of the RF dissimilarity. We refer to the resulting dissimilarity as “processed” RF dissimilarity.

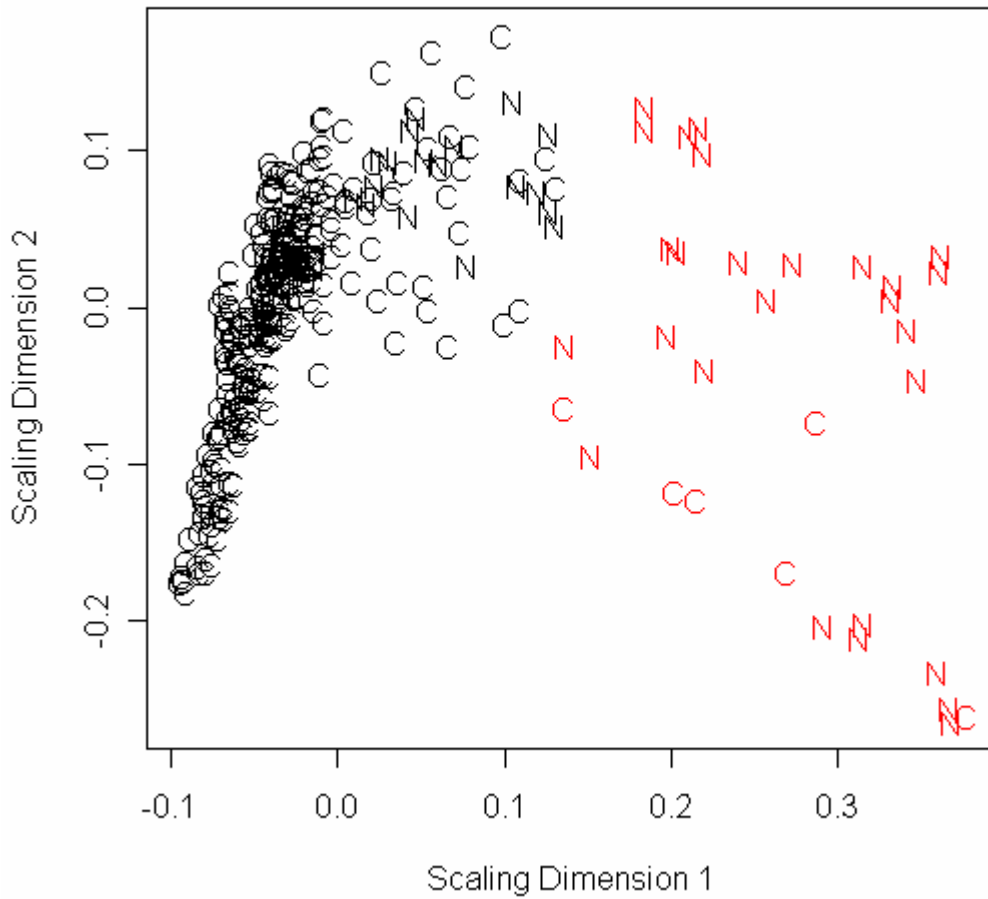
In other contexts, several authors have advocated the use principal components in clustering analysis in order to amplify the signal. However, this is a controversial step. Many statisticians would argue that the unprocessed, original RF dissimilarity contains a more meaningful signal. As discussed below, we use the unprocessed RF dissimilarity in our latest work. Here we report the “old” analysis to enhance reproducibility.

Actually, the main reason why we used the processed RF dissimilarity is that we wanted to visualize the cluster assignment in the corresponding 2 dimensional scaling plot.

```
# See the following plot.
```

**## Figure 1.A in Shi, Seligson et al (2005)**

```
# Classical MDS plot  
# Clear cell patients are labeled by "C" and non-clear cell patients are labelled by "N"  
# Patients are colored by their RF clustering memberships  
plot(cmd1, type="n", xlab="Scaling Dimension 1", ylab="Scaling Dimension 2")  
text(cmd1, label=ifelse(dat1$ClearCell==1, "C", "N"), col= RFclusterLabel)
```

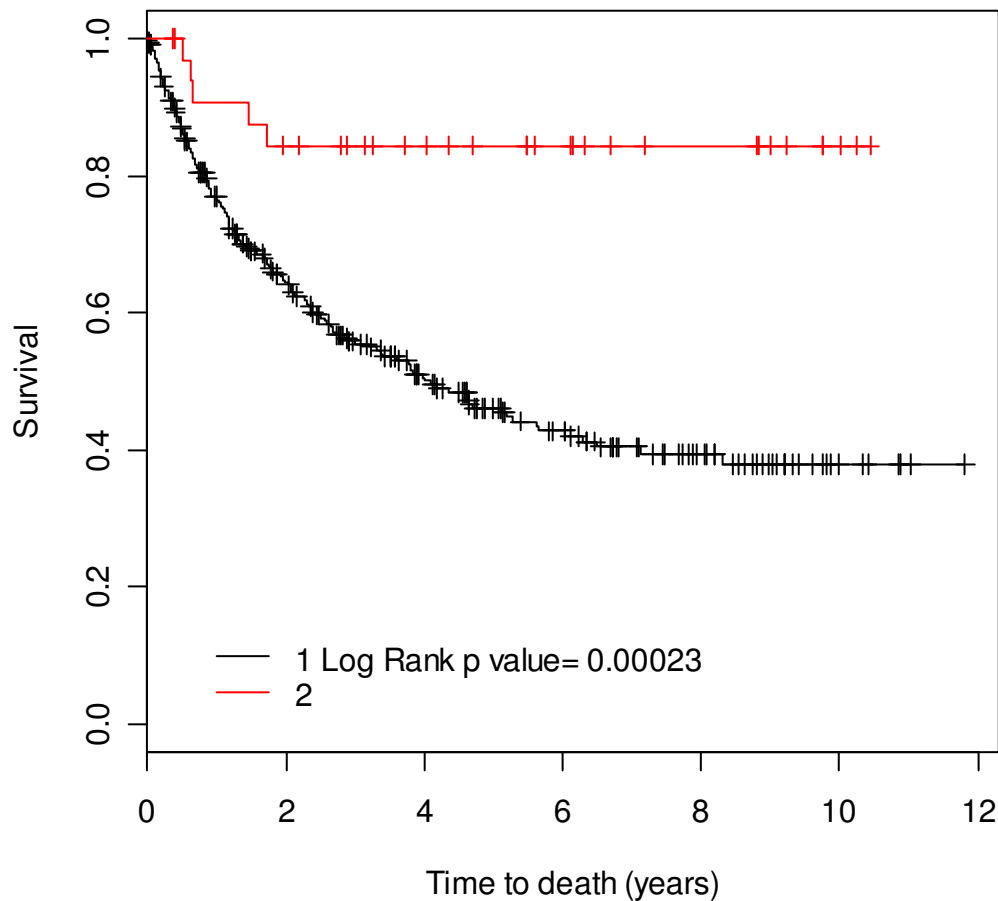


```

## The following corresponds to Figure 1.B
## Now we check survival difference between the two RF clusters
## Now we use Kaplan Meier plots to compare survival distributions.
## time denotes the survival or follow up time
## event denotes the death indicator variable
## We also report the Log-Rank test p value
fit.diff = survdiff(Surv(time, event) ~ factor(RFclusterLabel), data=dat1)
chisq2 = signif(1-pchisq(fit.diff$chisq,length(levels(factor(RFclusterLabel)))-1), 3)
fit1 = survfit(Surv(time, event)~RFclusterLabel, data=dat1, conf.type="log-log")
plot(fit1, conf.int=F,col=c(1:3), xlab="Time to death (years)",
ylab="Survival", main=c("K-M curves"), legend.text=c(paste("1 Log Rank p value=", chisq2), 2));

```

### K-M curves



Comment: the p-value is less significant than that reported in our article. Main reason: in the article, we used a slightly different clustering procedure (different from pamNew). In this tutorial, we strive to be consistent with other analyses and hence use pamNew. If you want the details of the original clustering analysis, please contact [shorvath@mednet.ucla.edu](mailto:shorvath@mednet.ucla.edu)

```
# For each patient, we define its clear cell status (cancer type)
ClearCellStatus= ifelse(dat1$ClearCell==1, "Clear", "Non-Clear")
```

```
## cluster 1 is enriched with clear cell patients and cluster 2 is enriched with
##non-clear cell patients (Table Supp1)
chisq.test(print(table(RFclusterLabel, ClearCellStatus)))
```

```
          ClearCellStatus
RFclusterLabel Clear Non-Clear
1             309      23
2              7      27
```

Pearson's Chi-squared test with Yates' continuity correction

```
data: print(table(RFclusterLabel, ClearCellStatus))
X-squared = 131.3041, df = 1, p-value < 2.2e-16
```

# Comment: note that the RF cluster label disagrees with the clear cell status on 23+7=30 patients.

```
#What if we used original pam clustering for the RF dissimilarity (instead of pamNEW)?
RFclusterLabelOriginalPAM = pam(cmd1, 2)$clustering
```

```
table(RFclusterLabelOriginalPAM, ClearCellStatus )
```

```
          ClearCellStatus
RFclusterLabelOriginalPAM Clear Non-Clear
1             307      19
2              9      31
```

# This leads to 28 misclassifications.

```
## Next, we compare RF clustering results with clustering results
## based on a processed version of the Euclidean distance.
## Specifically, to arrive at an unbiased comparison with
## the RF clustering method, we also use the cmdscale processing
## Euclidean distance
EuclidclusterLabel = pamNew(cmdscale(dist(datRF),2),2)
```

```
table(EuclidclusterLabel, ClearCellStatus )
```

EuclidclusterLabel	ClearCellStatus	
	Clear	Non-Clear
1	278	5
2	38	45

# Comment: note that the Euclidean cluster label disagrees with the clear cell status on 38+5=43 patients. Here the RF dissimilarity seems to do better.

#Note that in the above analysis, we used processed RF dissimilarity measures (i.e. we used multi-dimensional scaling”). In the following we briefly report the finding of using the original, unprocessed Euclidean distance.

```
UnprocessedEuclidclusterLabel= pamNew(dist(datRF),2)
```

```
table(UnprocessedEuclidclusterLabel, ClearCellStatus )
```

UnprocessedEuclidclusterLabel	Clear		Non-Clear
	1	283	
2	33	43	

#Now there are 40 misassignments. Thus, processing does not make a big difference for the Euclidean distance. If anything, it makes the clustering result worse.

# What would happen if we used the original PAM clustering procedure (instead of pamNEW)?

```
UnprocessedEuclidclusterLabelOriginalPAM= pam(dist(datRF),2)$clustering
```

```
table(UnprocessedEuclidclusterLabelOriginalPAM, ClearCellStatus )
```

UnprocessedEuclidclusterLabelOriginalPAM	ClearCellStatus	
	Clear	Non-Clear
1	272	4
2	44	46

#Now there are 44+4=48 misclassifications.

```

##=====##
##                                     ##
## Now we restrict the analysis to the 307 Clear Cell patients ##
##                                     ##
##=====##

## read in the data set. 307 clear cell patients
dat1 = read.table("testData_307.csv", sep=",", header=T, row.names=1)

## This is the input file for RF clustering algorithm
datRF = dat1[,1:8]

## Calculating RF distance between samples based on the 8 marker measurements
## This will take quite long time depends how many tree and repetitions you choose
## We suggest to use relatively large number of forests with large number of trees
distRF = RFdist(datRF, mtry1=3, 4000, 100, addcl1=T,addcl2=F,imp=T, oob.prox1=T)

## Multidimensional scaling based on RF distance
cmd1 = cmdscale(as.dist(distRF$c11),2)

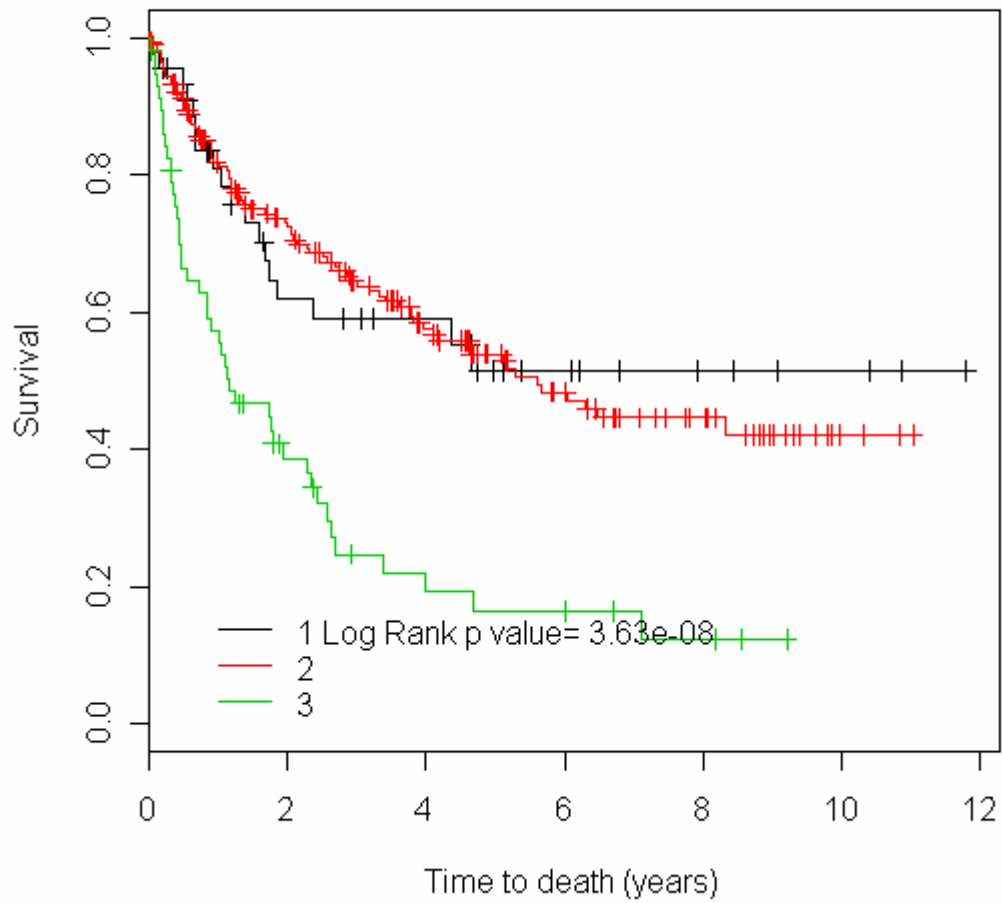
## To save time, I've provided the coordinates of cmd1 in the file "cmd1.csv",
##so we just read it in directly
cmd1 = as.matrix(read.csv("cmd1_307.csv", header=T, row.names=1))

## PAM clustering based on the scaling coordinates
RFclusterLabel = pamNew(cmd1, 3)

## check survival difference
## variables "time" and "event" in dat1 are survival time and censoring indicator, respectively
## Log-Rank p value = 3.63e-08!
fit.diff = survdiff(Surv(time, event) ~ factor(RFclusterLabel), data=dat1)
chisq2 = signif(1-pchisq(fit.diff$chisq,length(levels(factor(RFclusterLabel)))-1), 3)
fit1 = survfit(Surv(time, event)~RFclusterLabel, data=dat1, conf.type="log-log")
plot(fit1, conf.int=F,col=c(1:3), xlab="Time to death (years)",
ylab="Survival", main=c("K-M curves"), legend.text=c(paste("1 Log Rank p value=", chisq2),
2,3));

```

## K-M curves



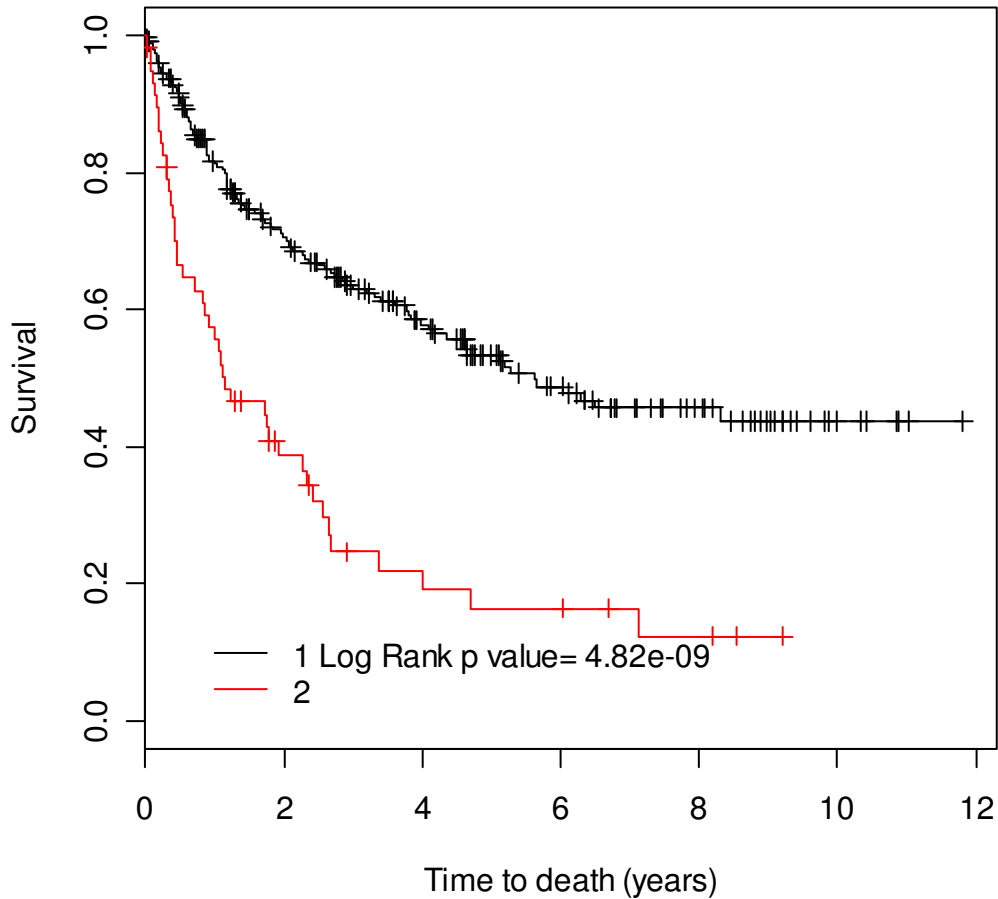
## Note that clusters 1(black) and 2(red) have similar survival distributions.  
## Therefore, we combine the two clusters and create a new cluster label  
label2 = 1+(RFclusterLabel>2)

```

## Check survival difference between the resulting RF cluster labels.
## Log-Rank p value = 4.82e-09! (Fig 2.B)
fit.diff = survdiff(Surv(time, event) ~ factor(label2), data=dat1)
chisq2 = signif(1-pchisq(fit.diff$chisq,length(levels(factor(label2))))-1), 3)
fit1 = survfit(Surv(time, event)~label2, data=dat1, conf.type="log-log")
plot(fit1, conf.int=F,col=c(1:3), xlab="Time to death (years)",
ylab="Survival", main=c("K-M curves"), legend.text=c(paste("1 Log Rank p value=", chisq2), 2));

```

### K-M curves



```

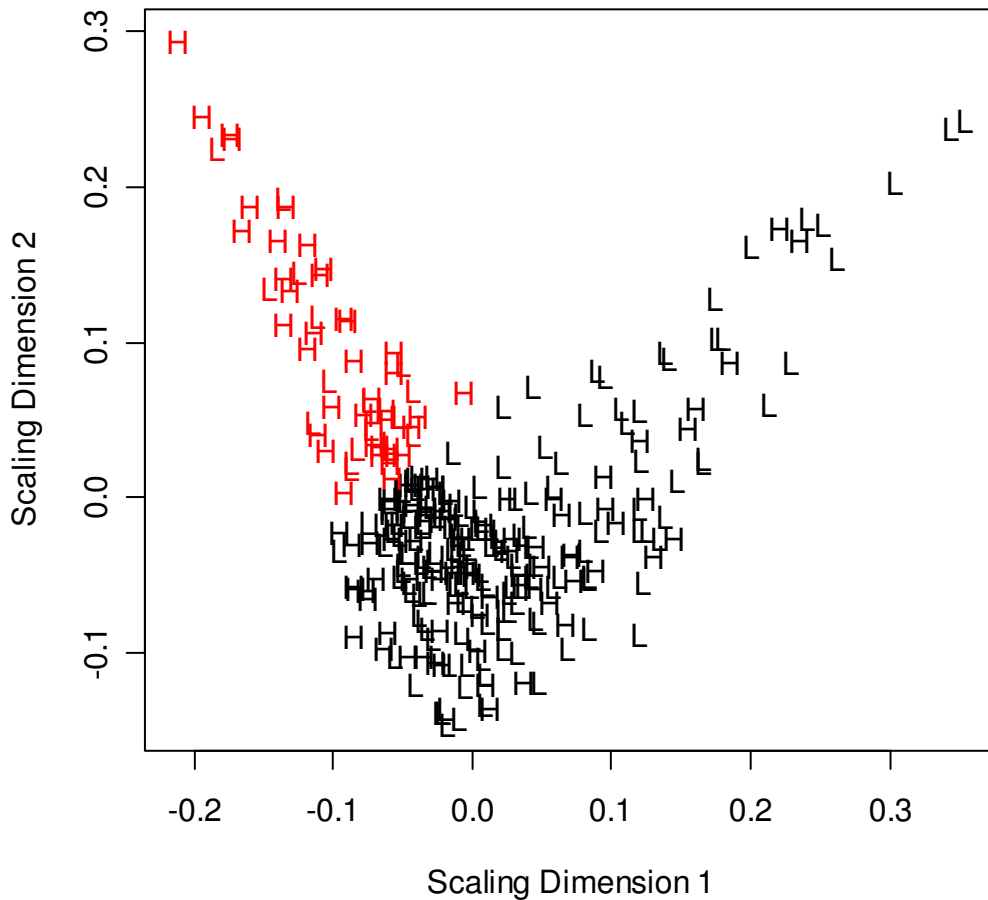
## cluster 1 is enriched with low-grade patients (grade<3) and cluster 2 is enriched
##with high-grade patients
chisq.test(print(table(label2, dat1$grade>2)))
label2 FALSE TRUE
  1    159    83
  2     20    39

Pearson's Chi-squared test with Yates' continuity correction

data: print(table(label2, dat1$grade > 2))
X-squared = 18.6086, df = 1, p-value = 1.605e-05

## MDS plot, The patients are labeled by 'H' for high-grade patients or 'L' for low-grade patients
## and colored by their RF clustering memberships
plot(cmd1,xlab="Scaling Dimension 1" ,ylab="Scaling Dimension 2", type="n")
text(cmd1, label=ifelse(dat1$grade>2, "H", "L"), col=label2)

```



```
## Comparing RF clustering results with clustering results based on Euclidean distance
EuclidclusterLabel = pamNew(cmdscale(dist(datRF),2),2) ## Euclidean distance
```

```
## Notice RF clustering gives lowest number of missclassified pateints
```

```
table(label2, ifelse(dat1$grade>2, "High", "Low"))
label2 High Low
1      83 159
2      39  20
```

```
table(EuclidclusterLabel, ifelse(dat1$grade>2, "High", "Low"))
EuclidclusterLabel High Low
1      83 142
2      39  37
```

```
## check survival difference using Euclidean distance based clusters
```

```
## here we use 3 clusters
```

```
label4 = pamNew(datRF, 3)
```

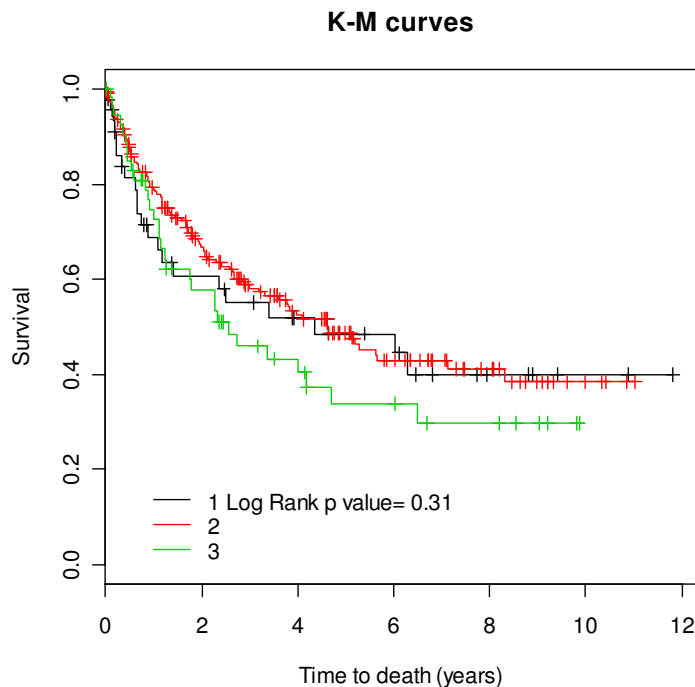
```
## check survival difference
```

```
fit.diff = survdiff(Surv(time, event) ~ factor(label4), data=dat1)
```

```
chisq2 = signif(1-pchisq(fit.diff$chisq,length(levels(factor(label4))))-1), 3)
```

```
fit1 = survfit(Surv(time, event)~label4, data=dat1, conf.type="log-log")
```

```
plot(fit1, conf.int=F,col=c(1:3), xlab="Time to death (years)",
ylab="Survival", main=c("K-M curves"), legend.text=c(paste("1 Log Rank p value=", chisq2),
2,3));
```



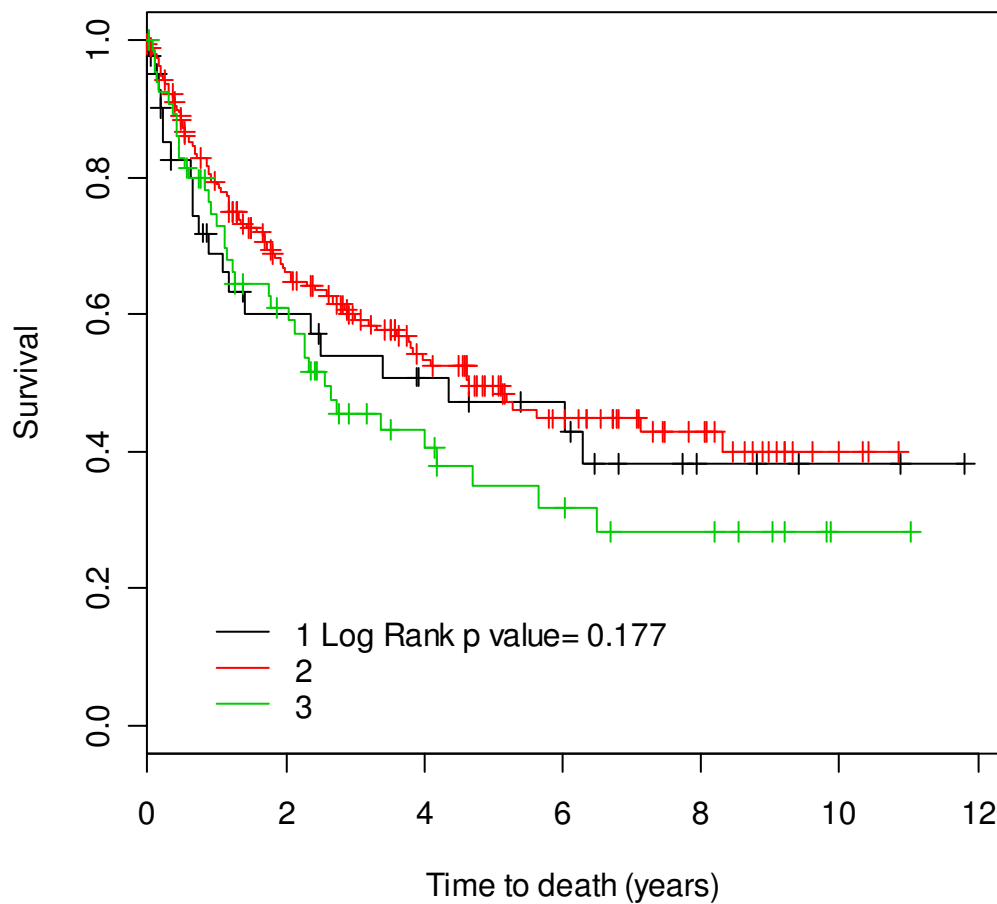
```
## Thus when using the Euclidean distance on the original data, the resulting KM curves are not significantly different.
```

```

## Now we use standard PAM clustering in conjunction with the Euclidean distance
label5 = pam(dist(datRF), 3)$clustering
## check survival difference
fit.diff = survdiff(Surv(time, event) ~ factor(label5), data=dat1)
chisq2 = signif(1-pchisq(fit.diff$chisq,length(levels(factor(label5)))-1), 3)
fit1 = survfit(Surv(time, event)~label5, data=dat1, conf.type="log-log")
plot(fit1, conf.int=F,col=c(1:3), xlab="Time to death (years)",
ylab="Survival", main=c("K-M curves"), legend.text=c(paste("1 Log Rank p value=", chisq2),
2,3));

```

### K-M curves



## The resulting KM curves based on the Euclidean distance are not significantly different.

## Discussion

Cluster analysis will always remain somewhat of an art form. Myriads of clustering procedures have been developed. Different from the case of supervised learning methods, it is difficult to gather empirical evidence that one procedure is superior over another. While our analysis provides evidence that the RF dissimilarity is superior to the Euclidean distance in this application, it is reassuring that the 2 dissimilarities provide similar results on many other (unreported) data sets.

The difference between the RF dissimilarity and the Euclidean distance are highlighted in Shi and Horvath (2006). In this theory paper, we present a slightly different and more thorough RF analysis of the 307 clear cell data. Main difference: we report the RF cluster analysis using the *unprocessed*, original RF dissimilarity. A corresponding tutorial can be found here <http://www.genetics.ucla.edu/labs/horvath/RFclustering/RFclustering.htm>

We are always glad to hear from successes and failures of the RF dissimilarity. Please email edits and suggestions to [shorvath@mednet.ucla.edu](mailto:shorvath@mednet.ucla.edu)