Computing selective inference p-values of clusters using pvclust and scaleboot

Hidetoshi Shimodaira

2019/11/17

Summary

- New versions of *pvclust* (>=2.2-0) and *scaleboot* (>=1.0-1) can compute selective inference (si) p-values as well as the conventional approximately unbiased (au) p-values and the bootstrap probability (bp).
- We recommend si values in most cases instead of au and bp values.
- Several estimating methods of p-values are implemented in the software. The default *pvclust* setting (e.g., *lung.pvclust* below) is a simple yet good enough method; this can be computed without *scaleboot*.
- More accurate, less biased versions of p-values (i.e., $k = 3$) can be computed by *scaleboot*. The setting of the wide range of scales (e.g., *lung73.k3* below) is then recommended.

Hierarchical Clustering of Lung Data

Introduction

Our method for computing selective inference p-values via multiscale bootstrap is explained in Shimodaira and Terada (2019). The theory for the selective inference behind the method is given in Terada and Shimodaira (2017). The *pvclust* package is originally described in Suzuki and Shimodaira (2006) for non-selective inference, and the multiscale bootstrap method of *scaleboot* is originally described in Shimodaira (2008).

The selective inference is also known as post-selection inference. This takes account of the fact that the clusters in the dendrogram are selected by looking at data. This contradicts the basic assumption of the conventional statistical hypothesis testing, which requires null hypotheses are selected before looking at the data. The newly proposed si values are adjusting the bias of this effect, but the conventional au values suffer from the selection bias.

- Hidetoshi Shimodaira and Yoshikazu Terada. Selective Inference for Testing Trees and Edges in Phylogenetics. Front. Ecol. Evol., 2019. https://doi.org/10.3389/fevo.2019.00174

- Yoshikazu Terada and Hidetoshi Shimodaira. Selective inference for the problem of regions via multiscale bootstrap. arXiv:1711.00949, 2017. https://arxiv.org/abs/1711.00949

- Hidetoshi Shimodaira. Testing regions with nonsmooth boundaries via multiscale bootstrap. Journal of Statistical Planning and Inference 138 (5), 1227-1241, 2008. https://doi.org/10.1016/j.jspi.2007.04.001

- Ryota Suzuki and Hidetoshi Shimodaira. Pvclust: an R package for assessing the uncertainty in hierarchical clustering. Bioinformatics 22 (12), 1540-1542, 2006. https://doi.org/10.1093/bioinformatics/btl117

pvclust and scaleboot packages

We use the following two packages here. Both packages implement the multiscale bootstrap method. Older versions of *pvclust* and *scaleboot* compute only BP and AU, but newer versions (pvclust>=2.2-0, scaleboot>=1.0-1) compute SI as well.

```
library(pvclust)  # computing p-values for clusters in hierarchical clustering
library(scaleboot)  # computing p-values for general setttings
```

Using multi-core CPU

If your pc has cpu with many cores, then we can speed up bootstrap computation.

```
### dont run
library(parallel)
length(cl <- makeCluster(detectCores()))
```

Using pvclust package

lung data

We use the sample data of microarray expression profiles. It is $n \times m$ matrix for $n = 916$ genes and $m = 73$ tumors. We compute clusters of tumors.

```
data(lung)  # in pvclust
dim(lung)
```

```
## [1] 916  73
```

run pvclust

We may run pvclust as follows. The default scale is specifed as *r=seq(.5,1.4,by=.1)* in pvlcust. It is equivalent to $\sigma^{-2} = 0.5, 0.6, \ldots, 1.4$ for multiscale bootstrap.

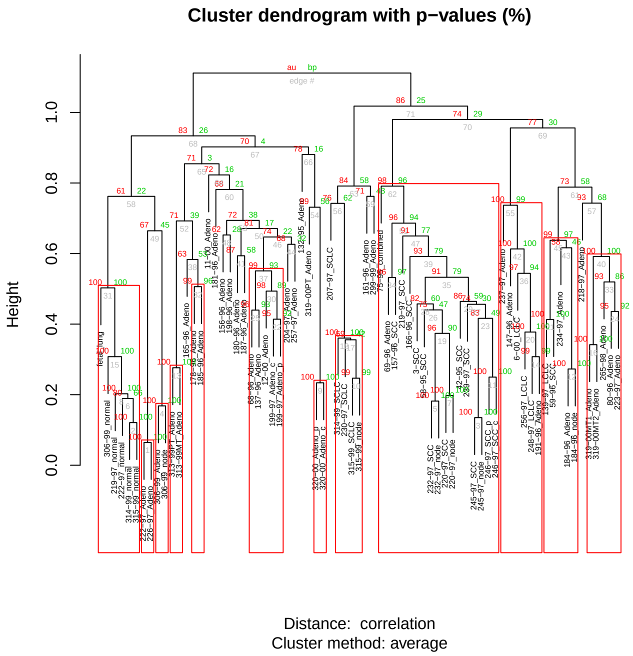
```
### dont run
lung.pvclust <- pvclust(lung, nboot=10000, parallel=cl)
```

The above code takes a long time. You can simply download the preveously performed result. The following code loads *lung.pvclust*, *lung73.pvclust*, *lung.sb*, and *lung73.sb* (see *help(lung73)*).

```
data(lung73)
```

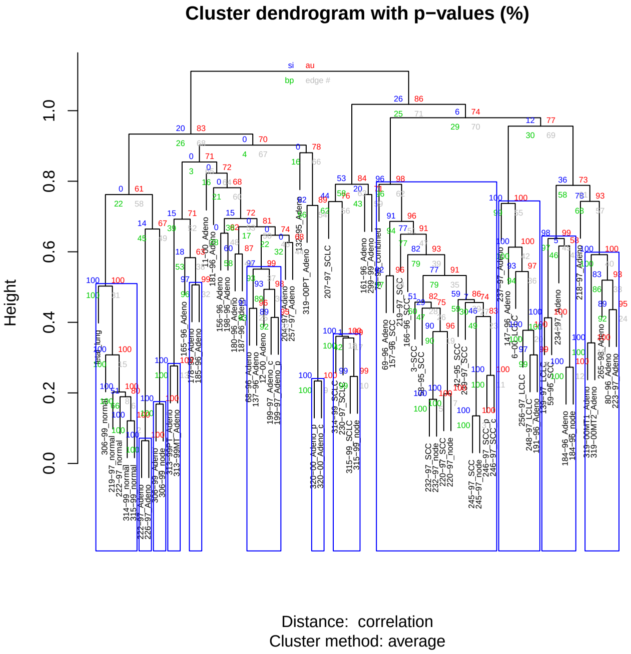
The default plot of cluster dendrogram shows two types of p-values (au, bp), and the significant clusters with au > 0.95 are shown as red boxes. au is the approximately unbiased p-value, and bp is the boostrap probability.

```
plot(lung.pvclust, cex=0.5, cex.pv=0.5)
pvrect(lung.pvclust)  # au > 0.95
```



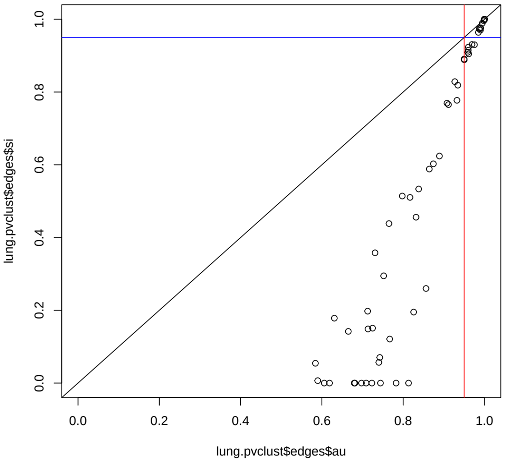
Now, we can also show apporximately unbiased p-values for selective inference, denoted as si. The cluster dendrogram with three types of p-values (si, au, bp) at each branch is shown as follows.

```
plot(lung.pvclust, print.pv = c("si", "au", "bp"), cex=0.5, cex.pv=0.5)
pvrect(lung.pvclust, pv="si") # si > 0.95
```



The lasted version of pvclust computes both si and au, but showing only au by default. In future, si will be shown by default. In this example, it happened that the red boxes and the blue boxes are showing the same clusters, but in genral, they differ. In fact, si and au values for all the clusters are different as seen below. au values are larger than si values, meaning au are easier than si to be significant (>0.95), but it tends to be false postives.

```
plot(lung.pvclust$edges$au, lung.pvclust$edges$si,xlim=c(0,1),ylim=c(0,1))
abline(0,1); abline(h=0.95, col="blue"); abline(v=0.95, col="red")
```



We only look at si values in the following sections.

Using scaleboot package

recompute p-values with the linear model compatible to pvclust

We can improve the accuracy of p-values by scaleboot package. We can recompute p-values from the pvclust result. The following result is actually included in *data(lung73)*, so we do not run it here.

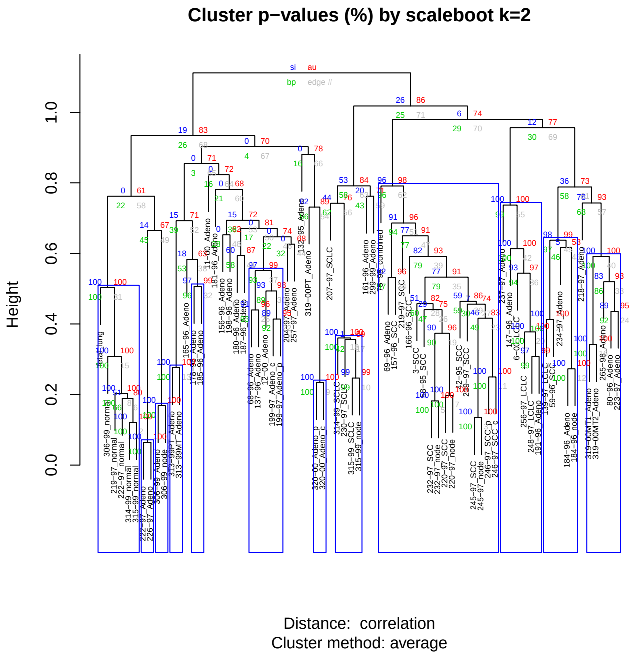
```
### dont run
lung.sb <- sbfit(lung.pvclust,cluster=cl) # model fitting
```

The above result (*lung.sb*) contains fitted models of the multiscale bootstrap probabilities. Then, p-values compatible to those of pvclust can be computed by the following code. This uses linear model (*poly.2*) for the fitting, and also uses linear model ($k = 2$) for the extrapolation to $\sigma^2 = -1$.

```
lung.poly2 <- sbpvclust(lung.pvclust, lung.sb, k=2, models = "poly.2")
```

We can see the dendrogram with the recomputed p-values as follows.

```r
plot(lung.poly2, print.pv = c("si", "au", "bp"), cex=0.5, cex.pv=0.5)
pvrect(lung.poly2, pv="si") # si > 0.95
```



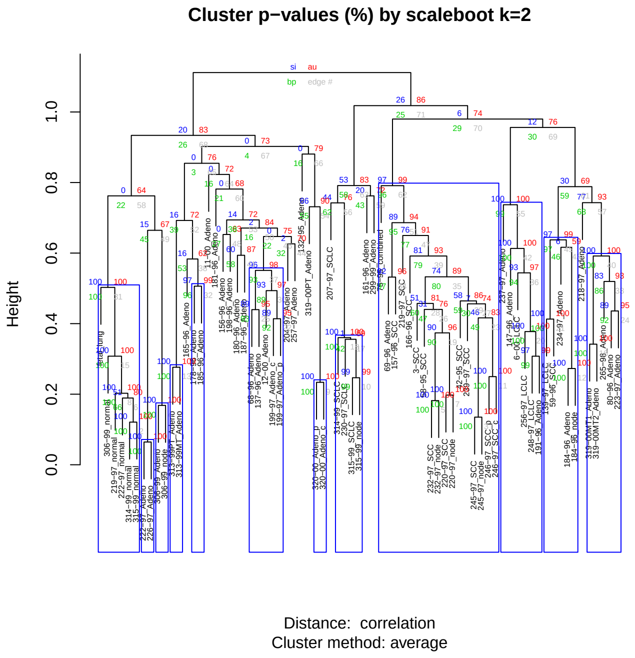
recompute p-values with general models $(k = 2)$

Now we omit the specification of models, then sbpvclust chooses good models by AIC. We again use the linear model $(k = 2)$ for the extrapolation.

```r
lung.k2 <- sbpvclust(lung.pvclust, lung.sb, k=2)
```

We can see the dendrogram with the recomputed p-values as follows.

```r
plot(lung.k2, print.pv = c("si", "au", "bp"), cex=0.5, cex.pv=0.5)
pvrect(lung.k2, pv="si") # si > 0.95
```



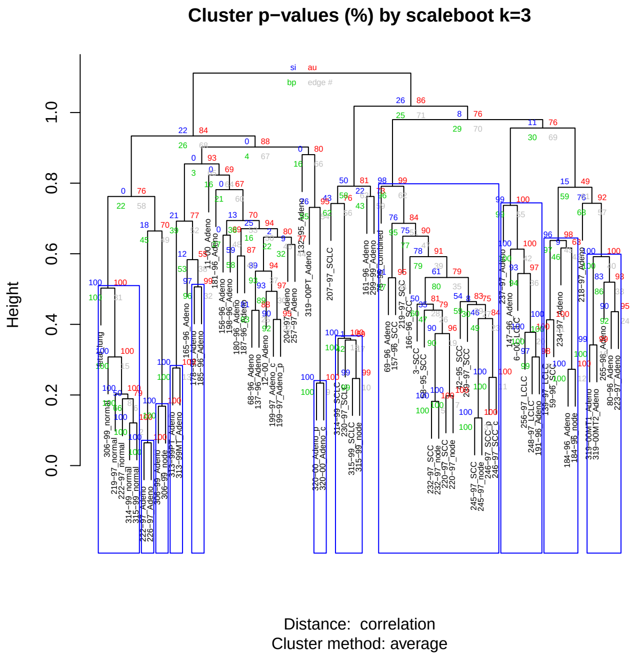
recompute p-values with general models $(k = 3)$

We use the quadratic model $(k = 3)$ for the extrapolation.

```
lung.k3 <- sbpvclust(lung.pvclust, lung.sb, k=3)
```

We can see the dendrogram with the recomputed p-values as follows.

```
plot(lung.k3, print.pv = c("si", "au", "bp"), cex=0.5, cex.pv=0.5)
pvrect(lung.k3, pv="si") # si > 0.95
```



comparing the estimation methods

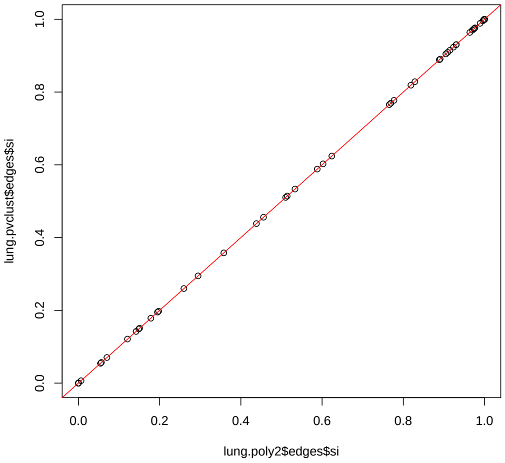
So far we have shown four versions of si values. You may wonder which one to be used.

- The default pvclust result (*lung.pvclust*)
- Recomputation via scaleboot compatible to pvclust (*lung.poly2*)
- scaleboot $k = 2$ (*lung.k2*)
- scaleboot $k = 3$ (*lung.k3*)

The first three types are trying to compute the same value ($k = 2$). Among these three, we recommend *lung.pvclust* as a simple yet good enough method. Let us see how they differ.

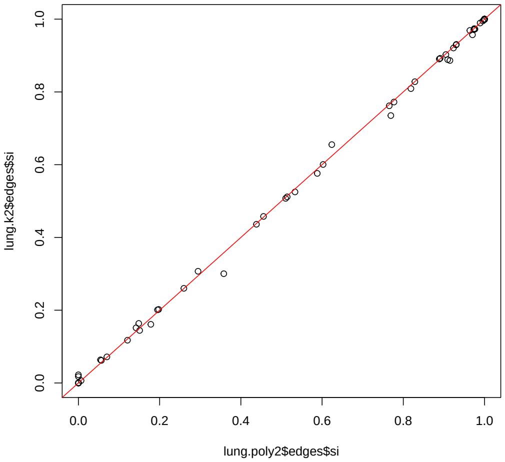
First, check if *lung.pvclust* and *lung.poly2* are actually (almost) the same.

```
plot(lung.poly2$edges$si, lung.pvclust$edges$si); abline(0,1, col="red")
```



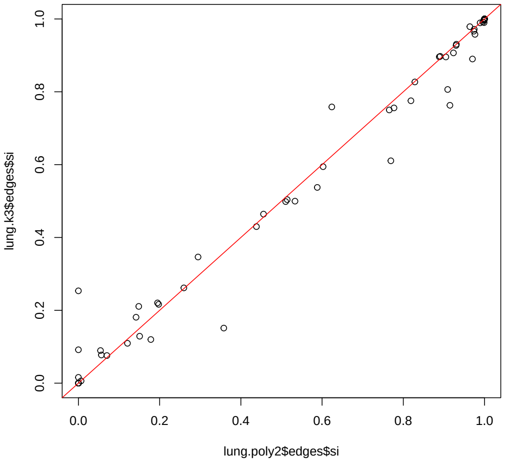
lung.k2 and *lung.poly2* are also very similar.

```
plot(lung.poly2$edges$si, lung.k2$edges$si); abline(0,1, col="red")
```



The fourth estimator *lung.k3* ($k = 3$) is different from the above three estimators.

```
plot(lung.poly2$edges$si, lung.k3$edges$si); abline(0,1, col="red")
```



$k = 3$ is expected to be more accurate (less biased) than $k = 2$, but tends to have larger variance. For reducing the variance, we recommed to use the wider range of scales as explained in the next section.

Using scaleboot with a wider range of scales

using wider range of scales instead of the pvclust default scales

The default range of scales in pvclust is rarrow. Here we try a wider range of scales: thirteen values of σ^2 are specified in log-scale from 1/9 to 9. In general, a wider range will reduce the variance of p-values. We do not run the code below, because the results are again included in *data(lung73)*.

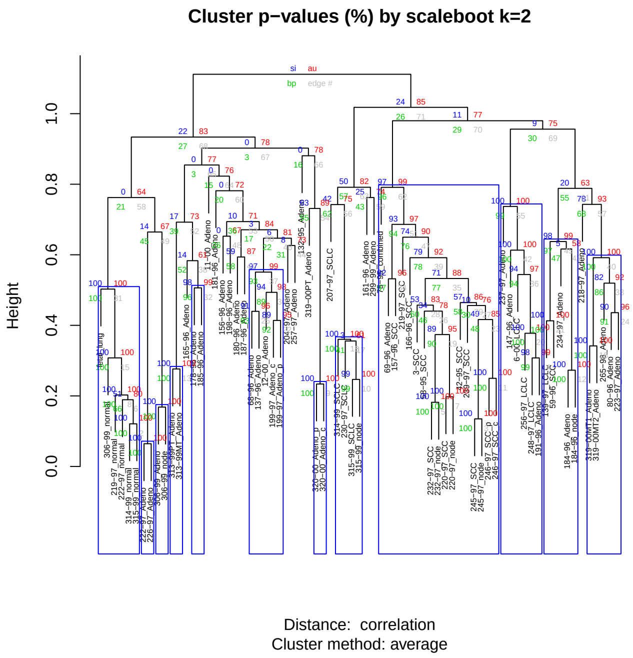
```
### dont run
sa <- 9^seq(-1,1,length=13) # wider range of scales than pvclust default
lung73.pvclust <- pvclust(lung ,r=1/sa, nboot=10000, parallel=cl)
lung73.sb <- sbfit(lung73.pvclust,cluster=cl) # model fitting
```

In the following, we compute two sets of p-values specified by the parameter k. In general, larger k increses the accuracy of p-values (less biased), and smaller k increses the stability (intuitively related to the variance). The p-values of pvclust correspond to $k = 2$.

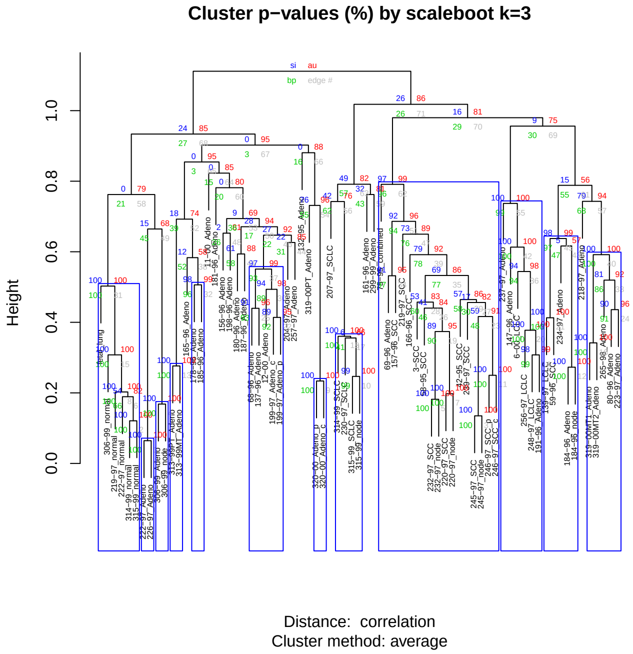
```
lung73.k2 <- sbpvclust(lung73.pvclust,lung73.sb, k=2) # k = 2
lung73.k3 <- sbpvclust(lung73.pvclust,lung73.sb, k=3) # k = 3
```

Let us see the dendrograms.

```
plot(lung73.k2, print.pv = c("si", "au", "bp"), cex=0.5, cex.pv=0.5)
pvrect(lung73.k2, pv="si") # si > 0.95
```



```
plot(lung73.k3, print.pv = c("si", "au", "bp"), cex=0.5, cex.pv=0.5)
pvrect(lung73.k3, pv="si") # si > 0.95
```



comparing estimation methods

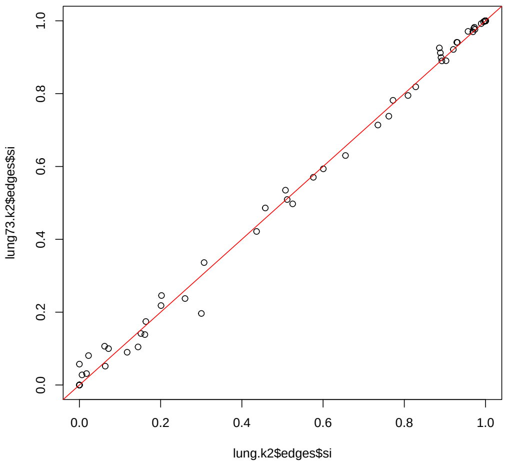
So far we have shown many dendrograms for the same dataset. We have compared the first four methods in the previous section. Now we look at the last two methods.

- The default pvclust result without scaleboot (*lung.pvclust*)
- The default pvclust scale & scaleboot compatible to pvclust (*lung.poly2*)
- The default pvclust scale & scaleboot $k = 2$ (*lung.k2*)
- The default pvclust scale & scaleboot $k = 3$ (*lung.k3*)
- The wider scale & scaleboot $k = 2$ (*lung73.k2*)
- The wider scale & scaleboot $k = 3$ (*lung73.k3*)

We recommend *lung73.k2* for $k = 2$ instead of *lnug.k2* (or *lung.pvclust*), and recommend *lung73.k3* for $k = 3$ instead of *lnug.k3*. This is because, the wider range of scales in *lung73.k2* and *lung73.k3* reduces the vairance of si values.

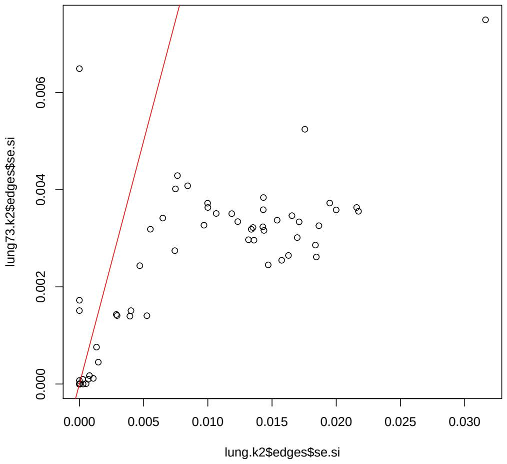
Let us see the si values first. *lung73.k2* and *lung.k2* are similar.

```
plot(lung.k2$edges$si, lung73.k2$edges$si); abline(0,1, col="red")
```



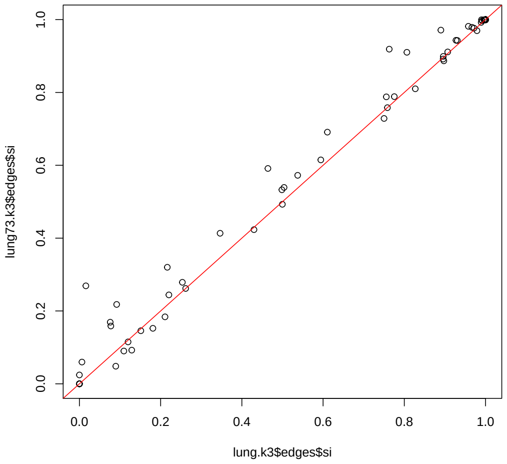
Looking at the standard errors of si values, we found that *lung73.k2* has much smaller variance than *lung.k2*. This large difference is not explained by the the difference of the number of bootstrap replicates, and it is due to the range of scales.

```
plot(lung.k2$edges$se.si, lung73.k2$edges$se.si); abline(0,1, col="red")
```



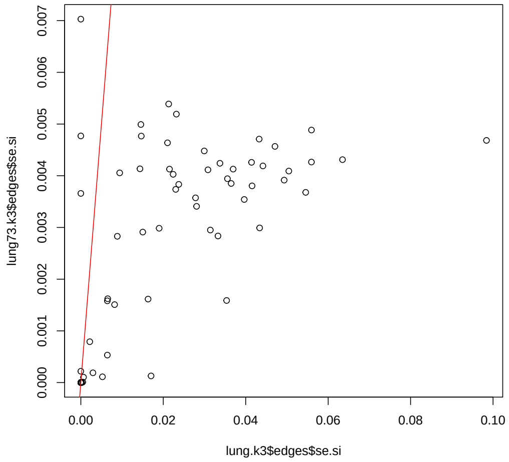
In theory, *lung73.k3* and *lung.k3* should be similar, but they are not so much. (due to the variance?)

```
plot(lung.k3$edges$si, lung73.k3$edges$si); abline(0,1, col="red")
```



Looking at the standard errors of si values, we found that *lung.k3* has large variance, and it is reduced in *lung73.k3*.

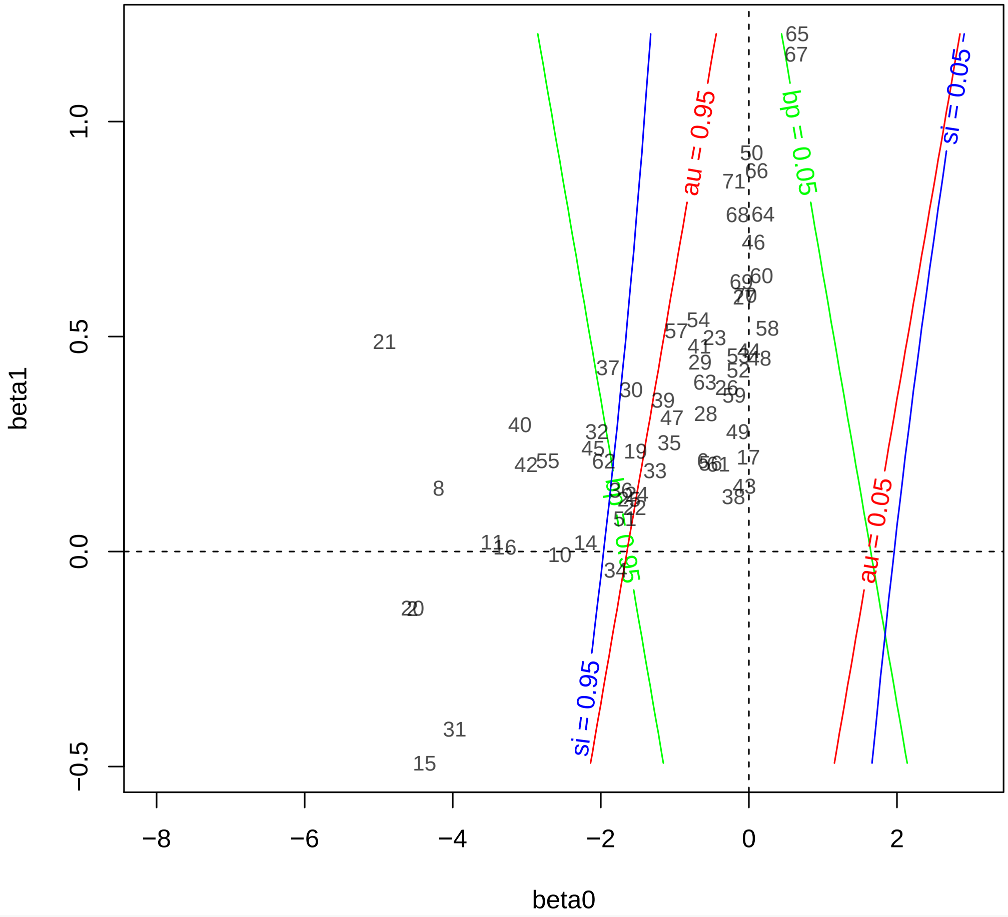
```
plot(lung.k3$edges$se.si, lung73.k3$edges$se.si); abline(0,1, col="red")
```



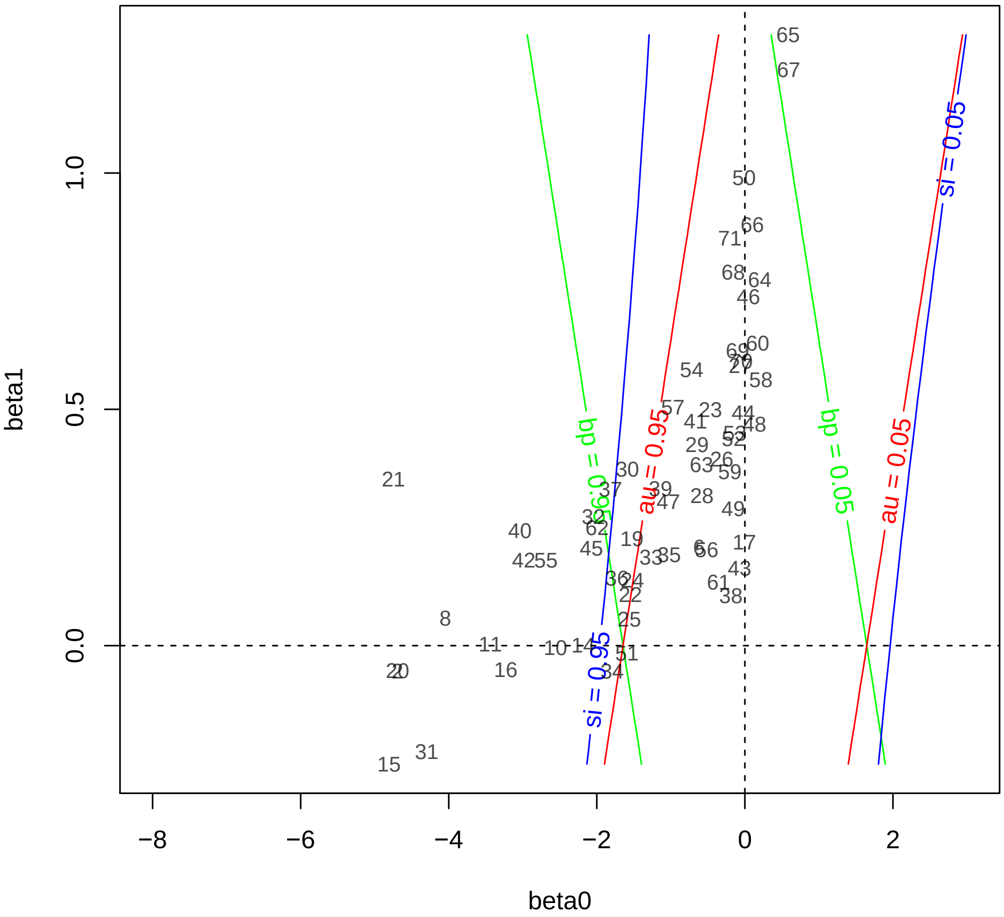
Geometric interpretations

For $k = 2$, the three types of p-values (si, au, bp) are computed from two geometric quantities β_0 (*signed distance*) and β_1 (*curvature*). Look at the estimated values of them.

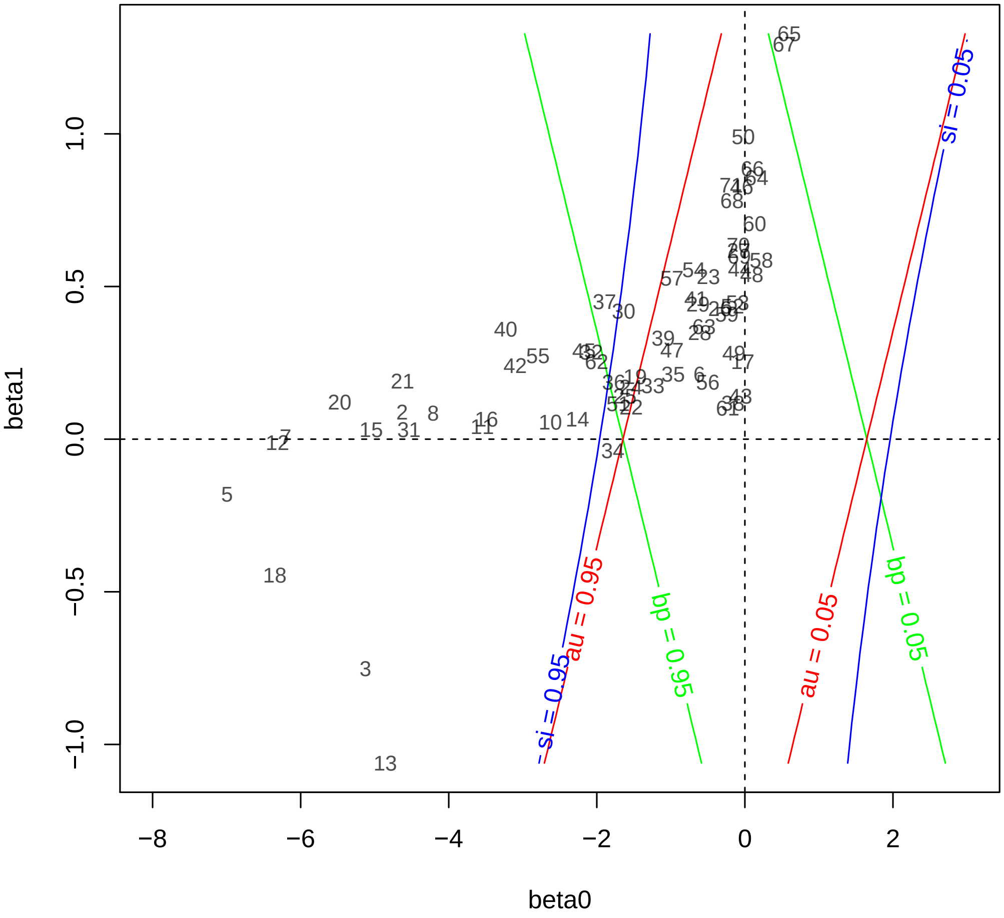
```
lung.poly2.ss <- summary(lung.sb, models = "poly.2", k=2) # compute p-values
lung.poly2.aa <- attr(lung.poly2.ss, "table") # extract table of p-values, etc
lung.poly2.beta <- lung.poly2.aa$value[,c("beta0","beta1")]
sbplotbeta(lung.poly2.beta,col=rgb(0,0,0,alpha=0.7),cex=0.8,xlim=c(-8,3))
```



```
lung.k2.ss <- summary(lung.sb, k=2) # compute p-values
lung.k2.aa <- attr(lung.k2.ss, "table") # extract table of p-values, etc
lung.k2.beta <- lung.k2.aa$value[,c("beta0","beta1")]
sbplotbeta(lung.k2.beta,col=rgb(0,0,0,alpha=0.7),cex=0.8,xlim=c(-8,3))
```



```
lung73.k2.ss <- summary(lung73.sb, k=2) # compute p-values
lung73.k2.aa <- attr(lung73.k2.ss, "table") # extract table of p-values, etc
lung73.k2.beta <- lung73.k2.aa$value[,c("beta0","beta1")]
sbplotbeta(lung73.k2.beta,col=rgb(0,0,0,alpha=0.7),cex=0.8,xlim=c(-8,3))
```



Although we expect $\beta_0 \leq 0$ and $\beta_1 \geq 0$, some clusters do not satisfy these inequalities. They might have some problem.

```
lung73.k2.aa$character[which(lung73.k2.beta[,2]<0),c("beta0","beta1")] # show beta1 < 0
```

```
##    beta0          beta1
## 3  "-5.12 (1.04)" "-0.75 (0.29)"
## 5  "-6.99 (0.74)" "-0.18 (0.18)"
## 12 "-6.31 (0.30)" "-0.01 (0.07)"
## 13 "-4.86 (1.41)" "-1.06 (0.37)"
## 18 "-6.35 (0.99)" "-0.45 (0.26)"
## 34 "-1.78 (0.02)" "-0.04 (0.01)"
```

The values in parantheses are standard errors. Those with $\beta_1 < 0$ have large standard errors, so they happened because nboot is not enough. In any case, these clusters have very small β_0 values with very high confidence levels. So we do not have to worry about it.

```
lung73.k2.aa$character[which(lung73.k2.beta[,1]>0),c("beta0","beta1")] # show beta0 > 0
```

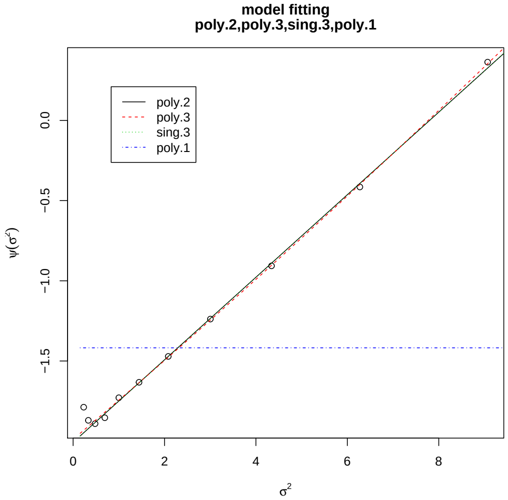
```
##    beta0          beta1
## 48 " 0.09 (0.01)" " 0.54 (0.00)"
## 58 " 0.22 (0.01)" " 0.59 (0.00)"
## 60 " 0.13 (0.00)" " 0.71 (0.00)"
## 64 " 0.16 (0.00)" " 0.86 (0.01)"
## 65 " 0.60 (0.01)" " 1.33 (0.01)"
## 66 " 0.10 (0.00)" " 0.88 (0.00)"
## 67 " 0.53 (0.01)" " 1.29 (0.01)"
```

The standard errors for $\beta_0 > 0$ are very small. They have β_0 close to origin with low confidence levels. So again, we may not need to worry about it.

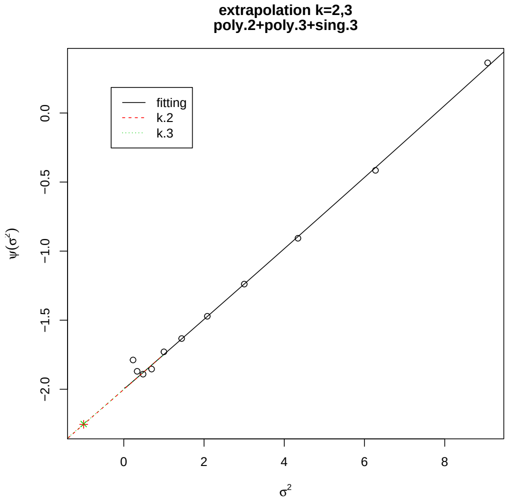
Diagnostics of model fitting

Look at details of each cluster. Model fitting for cluster id=62, say, is shown as follows.

```
plot(lung73.sb[[62]],legend="topleft") # fitting candiate models
```



```
plot(summary(lung73.sb[[62]], k=2:3),legend="topleft") # extrapolation to sigma^2 = -1
```



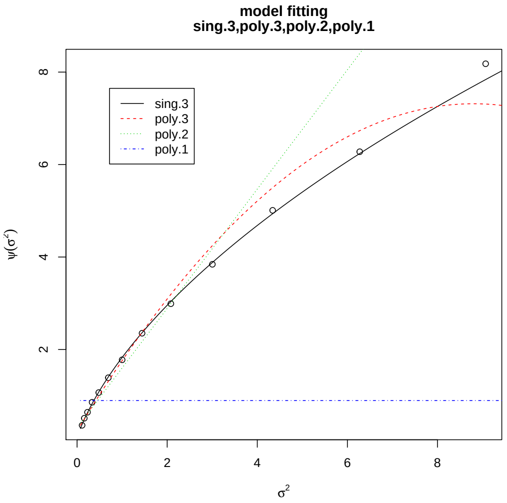
```
summary(lung73.sb[[62]], k=2:3) # p-values for k=2 and k=3
```

```
##
## Raw Bootstrap Probability (scale=1) : 95.81 (0.20)
##
## Hypothesis: alternative
##
## Corrected P-values for Models (percent,Frequentist):
##         k.2          k.3          sk.2         sk.3         beta0        beta1        aic     weight
## poly.2 98.82 (0.05) 98.82 (0.05) 97.04 (0.12) 97.04 (0.12) -2.01 (0.01)  0.26 (0.00)  -15.70 47.70
## poly.3 98.73 (0.10) 98.71 (0.12) 96.85 (0.21) 96.81 (0.24) -1.99 (0.02)  0.25 (0.01)  -15.07 34.76
## sing.3 98.82 (0.05) 98.82 (0.05) 97.04 (0.12) 97.04 (0.12) -2.01 (0.01)  0.26 (0.00)  -13.70 17.55
## poly.1 92.19 (0.11) 92.19 (0.11) 84.37 (0.21) 84.37 (0.21) -1.42 (0.01)  0.00 (0.00) 4611.41
##
## Best Model:  poly.2
##
## Corrected P-values by the Best Model and by Akaike Weights Averaging:
##          k.2          k.3          sk.2         sk.3         beta0        beta1
## best    98.82 (0.05) 98.82 (0.05) 97.04 (0.12) 97.04 (0.12) -2.01 (0.01)  0.26 (0.00)
## average 98.79 (0.07) 98.78 (0.07) 96.98 (0.15) 96.96 (0.16) -2.00 (0.02)  0.25 (0.01)
```

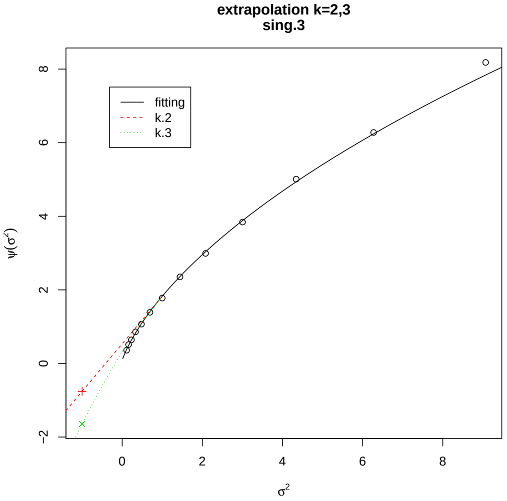
For cluster id=62 above, the quadratic model in terms of σ^2, namely, poly.3 ($\beta_0 + \beta_1\sigma^2 + \beta_2\sigma^4$) is the best model. The linear model poly.2 ($\beta_0 + \beta_1\sigma^2$) is also not bad. For computing AU and SI, we extrapolate the curve to $\sigma^2 = -1$. When $k = 1$, we use the tangent line at $\sigma^2 = 1$ for extrapolation. When $k = 2$, we use a quadratic curve for extrapolation. For cluster id=62, the difference between $k = 2$ and $k = 3$ is small. In the table, AU is indicated as k.2 and k.3, SI is indicated as sk.2 and sk.3.

Model fitting for cluster id=67 is shown as follows.

```
plot(lung73.sb[[67]],legend="topleft") # fitting candiate models
```



```
plot(summary(lung73.sb[[67]], k=2:3),legend="topleft") # extrapolation to sigma^2 = -1
```



```
summary(lung73.sb[[67]], k=2:3) # p-values for k=2 and k=3
```

```
##
## Raw Bootstrap Probability (scale=1) : 3.78 (0.19)
##
## Hypothesis: null
##
## Corrected P-values for Models (percent,Frequentist):
##        k.2          k.3          sk.2         sk.3         beta0       beta1       aic      weight
## sing.3 77.61 (0.46) 95.00 (0.18) 86.04 (0.39) 97.45 (0.09) 0.53 (0.01) 1.29 (0.01)    34.35 100.00
## poly.3 86.26 (0.29) 92.75 (0.28) 93.46 (0.16) 97.09 (0.14) 0.33 (0.00) 1.43 (0.01)   390.53
## poly.2 83.64 (0.32) 83.64 (0.32) 92.78 (0.19) 92.78 (0.19) 0.31 (0.00) 1.29 (0.01)  1319.11
## poly.1 18.53 (0.10) 18.53 (0.10) 37.05 (0.21) 37.05 (0.21) 0.90 (0.00) 0.00 (0.00) 53012.00
##
## Best Model:  sing.3
##
## Corrected P-values by the Best Model and by Akaike Weights Averaging:
##         k.2          k.3          sk.2         sk.3         beta0       beta1
## best    77.61 (0.46) 95.00 (0.18) 86.04 (0.39) 97.45 (0.09) 0.53 (0.01) 1.29 (0.01)
## average 77.61 (0.46) 95.00 (0.18) 86.04 (0.39) 97.45 (0.09) 0.53 (0.01) 1.29 (0.01)
```

For cluster id=67, the extrapolation to $\sigma^2 = 0$ with $k = 3$ is smaller than that with $k = 2$, suggesting that β_0 value with $k = 3$ would be smaller. This is also shown as estimated β_0 for poly.3 model below. (β_0 of sing.3 model is even closer to zero, which corresponds to $k \to \infty$.)

```
lung73.sb[[67]]
```

```
##
## Multiscale Bootstrap Probabilities (percent):
## 1     2     3     4     5     6     7     8     9     10    11    12    13
## 13.97 10.00  9.08  6.86  6.17  4.76  3.78  2.51  1.91  1.33  0.81  0.61  0.33
##
## Numbers of Bootstrap Replicates:
## 1     2     3     4     5     6     7     8     9     10    11    12    13
## 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
##
## Scales (Sigma Squared):
## 1      2      3      4      5      6      7 8     9     10    11    12    13
## 0.1111 0.1603 0.2311 0.3333 0.4808 0.6934 1 1.443 2.082 3.003 4.341 6.274 9.069
##
## Coefficients:
##        beta0           beta1           beta2
## sing.3 0.0837 (0.0110) 1.7437 (0.0188)  0.5166 (0.0167)
## poly.3 0.2418 (0.0048) 1.6080 (0.0143) -0.0914 (0.0024)
## poly.2 0.3103 (0.0043) 1.2899 (0.0099)
## poly.1 0.8955 (0.0039)
##
## Model Fitting:
##        rss      df pfit   aic
## sing.3    54.35 10 0.0000    34.35
## poly.3   410.53 10 0.0000   390.53
## poly.2  1341.11 11 0.0000  1319.11
## poly.1 53036.00 12 0.0000 53012.00
##
## Best Model:  sing.3
```

We expect that $\beta_0 < 0$ from the theory, the above result suggests $k = 3$ would be better than $k = 2$, although it is still positive. In general, p-values with $k = 3$ would have better accuracy than those with $k = 2$. Larger k should be better, but there is a trade-off beween the accuracy and numerical stability though, so we do not try $k = 4$ or higher.