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

Hidetoshi Shimodaira

2019/01/14

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).

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

Yoshikazu Terada and Hidetoshi Shimodaira. Selective inference for the problem of regions via multiscale bootstrap. arXiv:1711.00949, 2017.

Hidetoshi Shimodaira. Testing regions with nonsmooth boundaries via multiscale bootstrap. Journal of Statistical Planning and Inference 138 (5), 1227-1241, 2008.

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

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>=2.1-0, scaleboot>=1.0-0) 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.pv <- pvclust(lung, nboot=10000, parallel=cl)
```

using preveously computed result

We have run pvclust in *makedata.R* in *scaleboot*. Instead of the default scale of *pvclust*, we have used a wider range of the scale: thirteen values of σ^2 are specified in log-scale from 1/9 to 9.

```
nb.pvclust = 10000
sa <- 9^seq(-1,1,length=13) # wider range of scales than pvclust default
lung73.pvclust <- pvclust(lung,r=1/sa,nboot=nb.pvclust,parallel=cl) # took 30 mins with my laptop pc
```

You can get the result as follows.

```
data(lung73)  # in scaleboot
lung73.pvclust # pvclust object
```

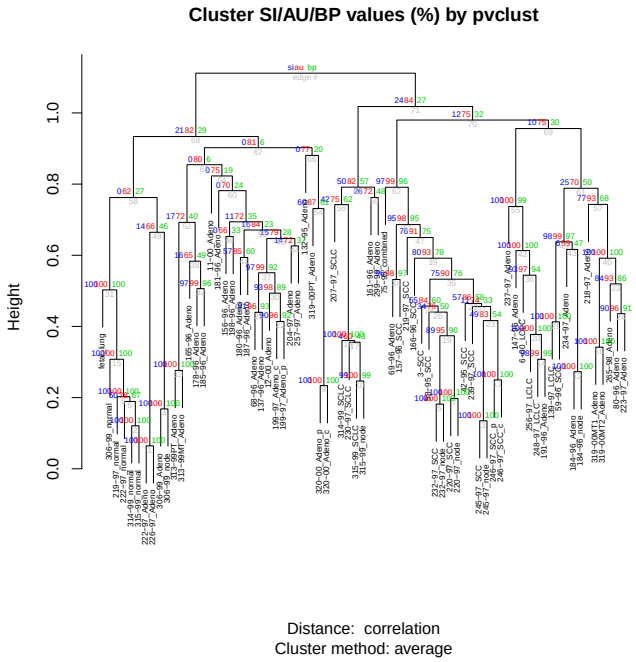
```
##
## Cluster method: average
## Distance      : correlation
##
## Estimates on edges:
##
##       si    au    bp se.si se.au se.bp       v      c  pchi
## 1  1.000 1.000 1.000 0.000 0.000 0.000   0.000  0.000 0.000
## 2  1.000 1.000 1.000 0.000 0.000 0.000  -4.658  0.104 0.898
## 3  1.000 1.000 1.000 0.000 0.000 0.000  -7.195 -0.170 0.593
## 4  1.000 1.000 1.000 0.000 0.000 0.000   0.000  0.000 0.000
## 5  1.000 1.000 1.000 0.000 0.000 0.000  -6.974 -0.073 1.000
## 6  0.500 0.785 0.670 0.002 0.002 0.001  -0.614  0.174 0.000
## 7  1.000 1.000 1.000 0.000 0.000 0.000  -6.510  0.114 0.208
## 8  1.000 1.000 1.000 0.000 0.000 0.000  -4.126  0.074 0.235
## 9  1.000 1.000 1.000 0.000 0.000 0.000   0.000  0.000 0.000
## 10 0.991 0.996 0.994 0.000 0.000 0.000  -2.592  0.050 0.960
## 11 1.000 1.000 1.000 0.000 0.000 0.000  -3.573  0.050 0.862
## 12 1.000 1.000 1.000 0.000 0.000 0.000  -6.487  0.004 0.962
## 13 1.000 1.000 1.000 0.000 0.000 0.000  -7.506 -0.086 0.731
## 14 0.980 0.991 0.986 0.000 0.001 0.001  -2.278  0.071 0.004
## 15 1.000 1.000 1.000 0.000 0.000 0.000  -5.379  0.090 0.632
## 16 1.000 1.000 1.000 0.000 0.000 0.000  -3.466  0.058 0.028
## 17 0.043 0.602 0.435 0.001 0.002 0.001  -0.046  0.211 0.000
## 18 1.000 1.000 1.000 0.000 0.000 0.000 -10.196  0.212 1.000
## 19 0.891 0.954 0.900 0.001 0.001 0.001  -1.485  0.201 0.131
## 20 1.000 1.000 1.000 0.000 0.000 0.000  -5.397  0.106 0.953
## 21 1.000 1.000 1.000 0.000 0.000 0.000  -4.400  0.154 0.006
## 22 0.901 0.956 0.924 0.001 0.001 0.001  -1.570  0.134 0.000
## 23 0.486 0.828 0.536 0.002 0.001 0.001  -0.518  0.429 0.000
## 24 0.898 0.956 0.914 0.001 0.001 0.001  -1.537  0.169 0.033
## 25 0.913 0.962 0.929 0.000 0.001 0.001  -1.620  0.154 0.132
## 26 0.343 0.763 0.501 0.002 0.002 0.001  -0.360  0.356 0.000
## 27 0.111 0.738 0.330 0.002 0.002 0.001  -0.098  0.538 0.000
## 28 0.545 0.837 0.602 0.002 0.001 0.001  -0.620  0.361 0.173
## 29 0.572 0.858 0.580 0.002 0.001 0.001  -0.636  0.435 0.019
## 30 0.930 0.976 0.886 0.000 0.001 0.001  -1.587  0.382 0.082
## 31 1.000 1.000 1.000 0.000 0.000 0.000  -5.011  0.175 0.576
## 32 0.974 0.990 0.962 0.000 0.000 0.001  -2.045  0.275 0.124
## 33 0.841 0.935 0.856 0.001 0.001 0.001  -1.287  0.226 0.000
## 34 0.955 0.979 0.969 0.000 0.001 0.001  -1.953  0.084 0.000
## 35 0.747 0.903 0.759 0.001 0.001 0.001  -1.001  0.300 0.000
## 36 0.931 0.970 0.940 0.000 0.001 0.001  -1.719  0.160 0.080
## 37 0.970 0.990 0.924 0.000 0.000 0.001  -1.883  0.447 0.003
## 38 0.161 0.648 0.490 0.001 0.002 0.001  -0.177  0.203 0.000
## 39 0.805 0.930 0.776 0.001 0.001 0.001  -1.116  0.356 0.000
## 40 1.000 1.000 0.998 0.000 0.000 0.000  -3.298  0.384 0.000
## 41 0.571 0.852 0.597 0.002 0.001 0.001  -0.646  0.400 0.000
## 42 0.999 0.999 0.997 0.000 0.000 0.000  -3.022  0.224 0.296
## 43 0.061 0.588 0.466 0.001 0.002 0.001  -0.068  0.153 0.609
## 44 0.135 0.719 0.371 0.002 0.002 0.001  -0.126  0.454 0.000
## 45 0.983 0.993 0.970 0.000 0.000 0.001  -2.177  0.300 0.007
## 46 0.155 0.795 0.284 0.002 0.002 0.001  -0.126  0.696 0.000
## 47 0.757 0.910 0.751 0.001 0.001 0.001  -1.010  0.331 0.000
## 48 0.000 0.662 0.330 0.000 0.002 0.001   0.010  0.429 0.000
## 49 0.139 0.657 0.456 0.001 0.002 0.001  -0.147  0.258 0.000
## 50 0.163 0.837 0.230 0.003 0.002 0.001  -0.122  0.862 0.000
## 51 0.949 0.979 0.948 0.000 0.001 0.001  -1.826  0.199 0.000
## 52 0.174 0.720 0.402 0.002 0.002 0.001  -0.167  0.416 0.019
## 53 0.109 0.719 0.352 0.002 0.002 0.001  -0.099  0.480 0.000
## 54 0.605 0.865 0.613 0.002 0.001 0.001  -0.695  0.408 0.000
## 55 0.998 0.999 0.995 0.000 0.000 0.000  -2.873  0.311 0.029
## 56 0.423 0.754 0.624 0.002 0.002 0.001  -0.501  0.186 0.000
## 57 0.772 0.929 0.683 0.001 0.001 0.001  -0.974  0.498 0.000
## 58 0.000 0.624 0.274 0.000 0.002 0.001   0.143  0.458 0.000
## 59 0.256 0.723 0.476 0.002 0.002 0.001  -0.266  0.327 0.000
## 60 0.000 0.699 0.237 0.000 0.002 0.001   0.096  0.619 0.000
## 61 0.249 0.704 0.498 0.002 0.002 0.001  -0.267  0.270 0.000
```

```
## 62 0.970 0.988 0.960 0.000 0.001 0.001  -2.004  0.251 0.006
## 63 0.502 0.824 0.569 0.002 0.001 0.001  -0.553  0.378 0.000
## 64 0.000 0.746 0.192 0.000 0.002 0.001   0.106  0.767 0.000
## 65 0.000 0.802 0.056 0.000 0.003 0.001   0.371  1.219 0.000
## 66 0.000 0.771 0.203 0.000 0.002 0.001   0.046  0.787 0.000
## 67 0.000 0.810 0.063 0.000 0.003 0.001   0.325  1.202 0.000
## 68 0.213 0.818 0.287 0.003 0.002 0.001  -0.173  0.735 0.000
## 69 0.096 0.753 0.301 0.002 0.002 0.001  -0.081  0.602 0.597
## 70 0.125 0.754 0.318 0.002 0.002 0.001  -0.108  0.580 0.000
## 71 0.236 0.837 0.271 0.004 0.002 0.001  -0.187  0.797 0.000
## 72 1.000 1.000 1.000 0.000 0.000 0.000   0.000  0.000 0.000
```

dendrogram with p-values

Showing the dendrogram is easy. Each edge has three types of *p*-values: SI, AU, BP. Higher values indicate larger confidence in the clusters. **Warning: Do not use the following *p*-values as explained below.**

```
plot(lung73.pvclust, # Warning:  dont use the p-values of this plot!!!
     cex=0.5, cex.pv=0.5, offset=c(0.6,0.1,0.1,0.1))
```




```
# plot(lung.pv,  # use this plot!!!
#     cex=0.5, cex.pv=0.5, offset=c(0.6,0.1,0.1,0.1))
```

In the above dendrogram, we have used the wider range of scales. But it is not appropriate for pvclust actually. The wider range of scales is good for *scaleboot*, but the narrow range of scales is appropriate for *pvclust*.

Next, we recalcualte the *p*-values using *scaleboot*.

Using scaleboot package

Recalculating the p-values

We have run *pvclust* for calculating p-values by multiscale bootstrap. However, we use only the multiscale bootstrap probabilities from the output of *pvclust* below. First, we fit scaling law models to the the multiscale bootstrap probabilities.

```
### dont run
lung73.sb <- sbfit(lung73.pvclust,cluster=cl) # took a second with my laptop pc
```

Again, we have run it in *makedata.R*, and here we use the result in *data(lung73)*.

```
lung73.sb  # class is "scalebootv"
```

```
##
## Multiscale Bootstrap Probabilities (percent):
##    1   2   3   4   5   6   7   8   9   10  11  12  13
## 1  100 100 100 100 100 100 100 100 100 100 100 100 100
```

```
## 2  100 100 100 100 100 100 100 100 100  99  98  94  89
## 3  100 100 100 100 100 100 100 100 100 100 100 100 100
## 4  100 100 100 100 100 100 100 100 100 100 100 100 100
## 5  100 100 100 100 100 100 100 100 100 100 100 100  99
## 6   97  93  89  83  77  71  66  61  55  51  46  44  39
## 7  100 100 100 100 100 100 100 100 100 100 100  99  96
## 8  100 100 100 100 100 100 100 100 100 100  99  97  93  88
## 9  100 100 100 100 100 100 100 100 100 100 100 100 100 100
## 10 100 100 100 100 100 100 100  99  98  96  92  87  82  76
## 11 100 100 100 100 100 100 100 100 100  99  98  95  90  85
## 12 100 100 100 100 100 100 100 100 100 100 100 100 100  98
## 13 100 100 100 100 100 100 100 100 100 100 100 100 100 100
## 14 100 100 100 100 100 100  99  99  97  93  89  83  77  70
## 15 100 100 100 100 100 100 100 100 100 100 100  99  97  94
## 16 100 100 100 100 100 100 100 100 100 100  99  97  94  89  83
## 17  55  52  51  49  45  43  41  39  37  35  34  31  30
## 18 100 100 100 100 100 100 100 100 100 100 100 100 100 100
## 19 100 100 100  99  98  95  90  84  76  69  62  54  46
## 20 100 100 100 100 100 100 100 100 100 100 100  99  97  93
## 21 100 100 100 100 100 100 100 100 100 100  99  96  92  84
## 22 100 100 100  99  98  96  92  88  82  76  69  61  53
## 23  94  90  83  74  65  56  48  42  36  30  25  22  18
## 24 100 100 100  99  98  96  92  86  79  73  65  56  50
## 25 100 100 100 100  99  96  93  88  82  75  67  60  53
## 26  86  80  72  65  58  51  46  42  37  33  29  25  20
## 27  56  53  49  45  40  35  29  25  21  17  14  11   9
## 28  96  92  87  81  74  66  60  54  47  40  32  25  18
## 29  96  92  87  80  74  66  59  50  42  34  26  20  14
## 30 100 100 100  99  98  95  89  80  71  59  48  37  28
## 31 100 100 100 100 100 100 100 100 100  99  98  94  87
## 32 100 100 100 100 100  99  96  92  85  76  65  54  44
## 33 100 100  99  98  95  91  86  81  73  66  57  48  37
## 34 100 100 100 100  99  98  97  94  90  85  80  72  63
## 35  99  98  97  94  90  86  79  72  64  55  45  34  24
## 36 100 100 100 100  99  97  94  89  83  75  68  61  54
## 37 100 100 100 100 100  99  97  93  86  74  63  48  36  24
## 38  64  62  60  58  57  56  53  51  47  43  37  31  24
## 39 100  99  98  96  91  86  79  71  61  52  41  32  23
## 40 100 100 100 100 100 100 100 100  99  96  90  78  64  47
## 41  97  93  88  83  75  66  57  49  41  35  28  24  20
## 42 100 100 100 100 100 100 100 100  99  96  92  84  75  62
## 43  55  54  53  52  50  48  46  45  44  41  39  36  32
## 44  66  59  51  45  40  36  31  27  25  22  18  17  15
## 45 100 100 100 100 100 100  99  97  93  87  78  66  54  42
## 46  64  56  47  39  32  26  21  17  14  11   9   7   5
## 47 100  98  97  94  90  83  78  70  60  51  41  32  24
## 48  53  46  39  35  32  30  26  25  23  21  19  17  15
## 49  64  61  58  55  52  49  45  41  38  35  32  28  25
## 50  61  53  43  36  28  22  16  13   9   7   5   3   3
## 51 100 100 100 100  99  97  95  91  86  79  69  59  47
## 52  64  60  57  52  49  43  39  35  31  27  22  17  12
## 53  56  51  48  45  44  41  38  33  29  22  16  11   7
## 54  99  96  92  85  76  65  55  47  40  34  30  26  22
## 55 100 100 100 100 100 100  99  98  94  87  78  64  50
## 56  92  88  84  79  73  68  62  57  53  47  44  40  35
## 57 100  99  97  93  86  77  68  58  46  37  28  20  14
## 58  36  32  29  27  26  24  22  20  19  17  15  14  12
## 59  79  72  67  60  54  48  43  40  35  32  28  26  23
## 60  37  33  30  26  26  23  21  18  16  12   9   7   5
## 61  67  67  66  66  64  62  58  56  48  41  32  22  14
## 62 100 100 100 100 100 100  98  96  91  85  77  68  56  46
## 63  92  88  84  78  72  65  57  51  44  37  29  23  16
## 64  35  30  26  22  21  19  16  14  11   7   5   4   2
## 65  11   9   7   6   5   4   3   2   2   1   1   1   0
## 66  39  36  32  28  24  20  16  12  10   7   5   4   3
## 67  14  11   9   8   5   5   3   3   2   1   1   1   0
## 68  61  57  52  46  39  33  26  21  16  12   8   5   2
## 69  52  49  45  42  38  34  29  26  21  16  11   7   4
## 70  58  53  48  43  38  34  29  25  21  17  13   9   6
## 71  61  57  52  46  39  32  25  20  14  10   6   3   2
## 72 100 100 100 100 100 100 100 100 100 100 100 100 100
##
## 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
##
## AIC values of Model Fitting:
```

```
##    poly.1   poly.2  poly.3 sing.3
## 1      NA       NA      NA     NA
## 2   66.43   -15.26  -13.79 -13.53
## 3   -7.00   -12.05  -13.23 -11.05
## 4    0.65   -13.23  -12.18 -11.58
## 5   -8.95    -7.52   -6.99  -5.63
## 6 4209.99    51.22   -4.98 -13.71
## 7    6.17   -10.56  -11.63  -8.56
## 8   46.31   -11.91  -12.98 -11.16
## 9      NA       NA      NA     NA
## 10  65.47   -18.68  -16.96 -16.68
## 11  19.83   -17.39  -15.76 -15.39
## 12 -15.50   -13.54  -12.95 -11.55
## 13  -9.72   -11.39  -13.00 -10.18
## 14 236.56     0.23   -2.71   2.23
## 15  13.26   -14.34  -12.45 -12.34
## 16  52.26    -6.68   -9.95  -4.68
## 17 6526.92  194.26    9.60  -6.71
## 18  -7.73    -5.83  -11.71  -4.03
## 19 3638.71   -8.36   -9.61 -10.33
## 20  25.88   -15.70  -14.29 -13.70
## 21 268.87    -5.83   -9.70  -3.83
## 22 1483.80   10.58  -14.53  12.58
## 23 24161.90  821.64 111.02  -7.48
## 24 2466.77   -2.07   -1.14  -0.07
## 25 1902.08   -7.20   -8.96  -5.20
## 26 17283.86  450.15 131.65  19.00
## 27 32246.59  428.54  17.07  28.28
## 28 16773.03   -6.46   -9.93  -4.46
## 29 23249.87    1.24   -0.50   1.57
## 30 13284.96   -4.56   -9.62  -8.25
## 31  212.41   -14.21  -13.39 -12.21
## 32 5258.68    -8.09   -6.12  -6.09
## 33 5346.05   109.12   -7.01 111.12
## 34  528.97    89.11   -6.62  91.11
## 35 11403.24  562.40 195.53  564.40
## 36 1951.36    -6.63   -8.14  -7.15
## 37 16172.61    3.16    4.97   5.16
## 38 6454.35   584.67  79.11  586.67
## 39 14324.83   93.00  46.97  95.00
## 40 5940.13    13.57   8.27  15.57
## 41 20568.70  319.49   3.46  23.50
## 42 1814.50   -12.17  -13.08 -10.17
## 43 3399.61   -12.81  -14.52 -10.81
## 44 26372.96 1272.91 246.37   0.19
## 45 5939.42    -0.11   -6.11   1.89
## 46 47879.69 2015.14 430.53   -0.18
## 47 13036.64  148.21  75.35  150.21
## 48 23817.00 1520.60 498.58  50.18
## 49 9244.42    13.76   -9.69  -6.86
## 50 59032.06 1805.82 264.45   -4.10
## 51 3181.42   157.94  23.69  159.94
## 52 21449.88    1.13   -1.42  -2.02
## 53 26503.96  121.67  36.78  123.67
## 54 22318.18 1184.50 181.80   3.01
## 55 4431.13    -5.67   -8.90  -3.67
## 56 4866.24    20.35  21.72  22.06
## 57 27489.82   49.95   -4.26   0.64
## 58 25005.42 1354.17 345.18  24.86
## 59 14951.60  464.89  43.20   -7.05
## 60 36218.90  533.40 188.59  85.38
## 61 12778.75 2487.72 555.33 2489.72
## 62 4407.26    1.24 -10.51   3.24
## 63 18481.82   64.06  50.54  66.06
## 64 45749.58  756.82 252.11  94.98
## 65 50476.15 1284.74 448.77  28.75
## 66 48975.67  893.58  50.95   -7.85
## 67 52893.32 1307.79 462.20  48.38
## 68 47659.81  156.62  20.46  15.00
## 69 36130.65  -12.04 -14.29 -14.20
## 70 35203.01  205.80  33.16   -3.83
## 71 51888.09   94.21  14.10  22.44
## 72       NA      NA     NA     NA
```

Then p-values are calculated by *summary* method from the *scalebootv* object. For cluster id=62, for example, we can compute p-values as follows. We specify k (default is 3) for computing p-values.

```
summary(lung73.sb[62], k=2) # compute p-values (k=2)
```

```
##
## Corrected P-values by Akaike Weights Averaging (percent,Frequentist):
```

```
##    raw          k.2          sk.2         beta0        beta1       hypothesis  model  weight
## 62 95.67 (0.20) 98.49 (0.11) 96.36 (0.23) -1.95 (0.02) 0.21 (0.01) alternative poly.3 99.62
```

We can use any functions of *scaleboot* package like above. However, we use a specially designed function for working jointly with *pvclsut* as follows.

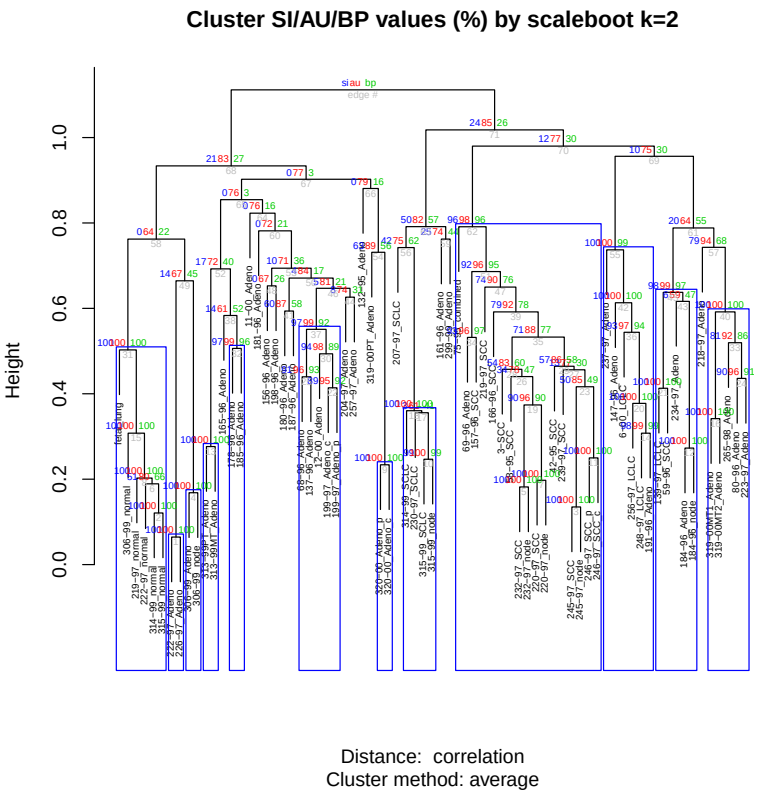
```
lung73.k2 <- sbpvclust(lung73.pvclust,lung73.sb, k=2) # compute p-values (k=2)
lung73.k2
```

```
##
## Cluster method: average
## Distance      : correlation
##
## Estimates on edges:
##
##       si    au    bp se.si se.au se.bp      v      c pchi
## 1  1.000 1.000 1.000 0.000 0.000 0.000     NA     NA    0
## 2  1.000 1.000 1.000 0.000 0.000 0.000 -4.720  0.124    0
## 3  1.000 1.000 1.000 0.000 0.000 0.000 -5.502 -0.593    0
## 4  1.000 1.000 1.000 0.002 0.001 0.000 -4.785 -0.891    0
## 5  1.000 1.000 1.000 0.000 0.000 0.000 -7.118 -0.101    0
## 6  0.510 0.796 0.658 0.003 0.002 0.002 -0.617  0.210    0
## 7  1.000 1.000 1.000 0.000 0.000 0.000 -5.935 -0.050    0
## 8  1.000 1.000 1.000 0.000 0.000 0.000 -4.275  0.126    0
## 9  1.000 1.000 1.000 0.000 0.000 0.000     NA     NA    0
## 10 0.991 0.996 0.994 0.001 0.000 0.000 -2.589  0.048    0
## 11 1.000 1.000 1.000 0.000 0.000 0.000 -3.563  0.045    0
## 12 1.000 1.000 1.000 0.000 0.000 0.000 -6.290 -0.042    0
## 13 1.000 1.000 1.000 0.000 0.000 0.000 -5.851 -0.500    0
## 14 0.977 0.989 0.986 0.002 0.001 0.001 -2.246  0.049    0
## 15 1.000 1.000 1.000 0.000 0.000 0.000 -5.349  0.082    0
## 16 0.999 1.000 1.000 0.000 0.000 0.000 -3.344  0.006    0
## 17 0.032 0.615 0.411 0.003 0.002 0.002 -0.034  0.259    0
## 18 1.000 1.000 1.000 0.001 0.001 0.000 -5.240 -0.907    0
## 19 0.896 0.957 0.901 0.004 0.002 0.001 -1.500  0.214    0
## 20 1.000 1.000 1.000 0.000 0.000 0.000 -5.326  0.086    0
## 21 1.000 1.000 1.000 0.000 0.000 0.000 -4.168  0.067    0
## 22 0.886 0.947 0.924 0.004 0.002 0.001 -1.524  0.093    0
## 23 0.496 0.851 0.487 0.003 0.001 0.002 -0.504  0.536    0
## 24 0.898 0.956 0.914 0.003 0.001 0.001 -1.535  0.167    0
## 25 0.910 0.960 0.929 0.003 0.002 0.001 -1.610  0.144    0
## 26 0.343 0.781 0.465 0.003 0.002 0.002 -0.345  0.432    0
## 27 0.130 0.768 0.304 0.003 0.002 0.002 -0.109  0.622    0
## 28 0.542 0.834 0.604 0.003 0.002 0.002 -0.617  0.353    0
## 29 0.574 0.860 0.578 0.003 0.002 0.002 -0.638  0.440    0
## 30 0.935 0.978 0.886 0.003 0.001 0.001 -1.609  0.401    0
## 31 1.000 1.000 1.000 0.000 0.000 0.000 -4.944  0.153    0
## 32 0.974 0.990 0.962 0.001 0.001 0.001 -2.045  0.275    0
## 33 0.811 0.918 0.858 0.004 0.002 0.001 -1.230  0.159    0
## 34 0.929 0.964 0.967 0.004 0.002 0.001 -1.818 -0.017    0
## 35 0.709 0.878 0.774 0.004 0.002 0.001 -0.959  0.207    0
## 36 0.935 0.972 0.941 0.003 0.001 0.001 -1.737  0.173    0
## 37 0.970 0.990 0.925 0.001 0.000 0.001 -1.885  0.447    0
## 38 0.140 0.610 0.518 0.003 0.002 0.002 -0.162  0.117    0
## 39 0.790 0.921 0.781 0.004 0.002 0.002 -1.095  0.320    0
## 40 0.999 1.000 0.998 0.000 0.000 0.000 -3.213  0.341    0
## 41 0.605 0.875 0.578 0.003 0.002 0.002 -0.673  0.477    0
## 42 0.998 0.999 0.997 0.000 0.000 0.000 -2.989  0.207    0
## 43 0.060 0.585 0.468 0.002 0.002 0.001 -0.067  0.149    0
## 44 0.081 0.737 0.310 0.005 0.002 0.002 -0.069  0.564    0
## 45 0.979 0.992 0.969 0.002 0.001 0.001 -2.135  0.272    0
## 46 0.048 0.807 0.213 0.007 0.002 0.002 -0.035  0.832    0
## 47 0.739 0.899 0.758 0.004 0.002 0.002 -0.987  0.288    0
## 48 0.000 0.672 0.264 0.000 0.003 0.002  0.092  0.539    0
## 49 0.144 0.665 0.450 0.003 0.002 0.002 -0.150  0.277    0
## 50 0.037 0.845 0.167 0.007 0.002 0.002 -0.024  0.991    0
## 51 0.923 0.965 0.946 0.004 0.002 0.001 -1.708  0.098    0
## 52 0.175 0.723 0.399 0.003 0.002 0.002 -0.168  0.424    0
## 53 0.102 0.705 0.363 0.003 0.002 0.002 -0.095  0.445    0
## 54 0.634 0.893 0.559 0.003 0.001 0.002 -0.697  0.548    0
## 55 0.998 0.999 0.994 0.000 0.000 0.000 -2.819  0.284    0
## 56 0.423 0.754 0.623 0.003 0.002 0.001 -0.501  0.187    0
## 57 0.788 0.937 0.677 0.003 0.001 0.002 -0.995  0.536    0
## 58 0.000 0.640 0.216 0.000 0.003 0.002  0.214  0.572    0
## 59 0.249 0.741 0.439 0.003 0.002 0.002 -0.247  0.401    0
## 60 0.000 0.717 0.206 0.000 0.002 0.002  0.123  0.699    0
## 61 0.203 0.635 0.553 0.003 0.002 0.002 -0.239  0.106    0
## 62 0.964 0.985 0.959 0.002 0.001 0.001 -1.953  0.215    0
## 63 0.495 0.819 0.574 0.003 0.002 0.002 -0.549  0.362    0
## 64 0.000 0.759 0.156 0.000 0.002 0.002  0.154  0.858    0
## 65 0.000 0.760 0.028 0.000 0.005 0.001  0.600  1.306    0
## 66 0.000 0.786 0.161 0.000 0.002 0.002  0.099  0.891    0
```

```
## 67 0.000 0.774 0.034 0.000 0.005 0.001  0.536  1.287    0
## 68 0.205 0.826 0.270 0.004 0.002 0.002 -0.162  0.777    0
## 69 0.096 0.755 0.299 0.003 0.002 0.002 -0.081  0.608    0
## 70 0.116 0.767 0.296 0.004 0.002 0.002 -0.096  0.633    0
## 71 0.242 0.847 0.260 0.004 0.002 0.002 -0.189  0.833    0
## 72 1.000 1.000 1.000 0.000 0.000 0.000     NA     NA    0
```

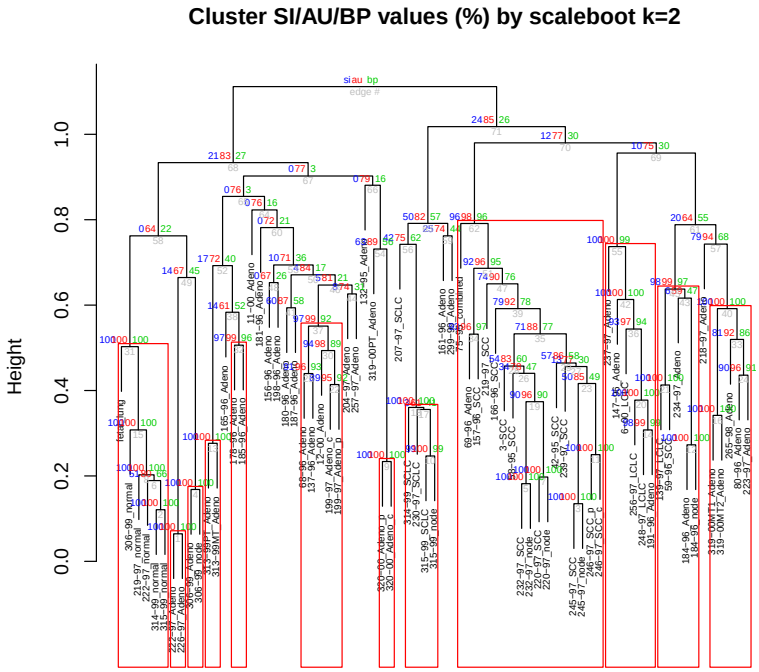
This is *pvclust* object, and you can draw the dendrogram from it. We draw rectangles for clusters with SI > 0.95. SI is the selective inference p-value which is appropriate for clusters, because SI adjusts the selection bias, whereas AU does not.

```
plot(lung73.k2, # p-values should be close to plot(lung.pv)
     cex=0.5, cex.pv=0.5, offset=c(0.6,0.1,0.1,0.1))
pvrect(lung73.k2, pv="si") # draw rectangle for clusters si>0.95
```



In the older versions of *pvclust* and *scaleboot* can only compute AU instead of SI. AU is not appropriate for clusters found by the dedrogoram you looked at. AU is appropriate only for clusters which you were interested in before looking at the dendrogram. The red rectangles are clusters with AU$>$0.95, which were computed in older version of the program. Fortunately, the results did not change so much in this data.

```
plot(lung73.k2, # p-values should be close to plot(lung.pv)
     cex=0.5, cex.pv=0.5, offset=c(0.6,0.1,0.1,0.1))
pvrect(lung73.k2, pv="au") # draw rectangle for clusters au>0.95
```

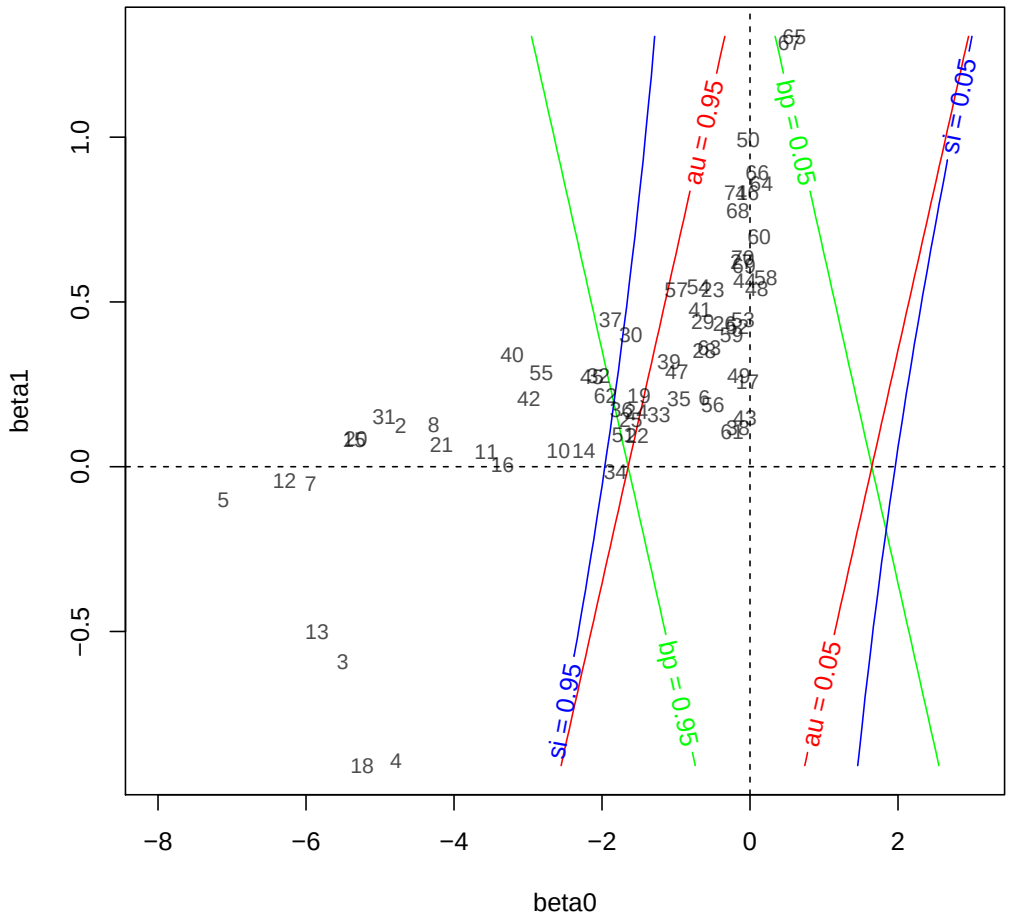


Distance: correlation
Cluster method: average

Diagnostics of beta0 and beta1

The three types of p-values (SI, AU, BP) are computed from two geometric quantities β_0 and β_1. Look at the estimated values of them.

```
lung73.ss <- summary(lung73.sb, k=2) # compute p-values
lung73.aa <- attr(lung73.ss, "table") # extract table of p-values, etc
lung73.beta <- lung73.aa$value[,c("beta0","beta1")]
sbplotbeta(lung73.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.aa$character[which(lung73.beta[,2]<0),c("beta0","beta1")] # show beta1 < 0
```

```
##    beta0          beta1
## 3  "-5.50 (0.96)" "-0.59 (0.26)"
## 4  "-4.79 (1.19)" "-0.89 (0.31)"
## 5  "-7.12 (0.62)" "-0.10 (0.15)"
## 7  "-5.94 (0.46)" "-0.05 (0.12)"
## 12 "-6.29 (0.33)" "-0.04 (0.08)"
## 13 "-5.85 (0.93)" "-0.50 (0.25)"
## 18 "-5.24 (1.30)" "-0.91 (0.35)"
## 34 "-1.82 (0.02)" "-0.02 (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.aa$character[which(lung73.beta[,1]>0),c("beta0","beta1")] # show beta0 > 0
```

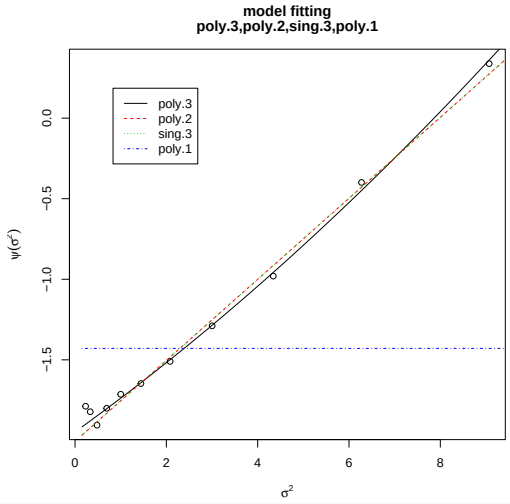
```
##    beta0          beta1
## 48 " 0.09 (0.01)" " 0.54 (0.00)"
## 58 " 0.21 (0.01)" " 0.57 (0.00)"
## 60 " 0.12 (0.00)" " 0.70 (0.00)"
## 64 " 0.15 (0.00)" " 0.86 (0.01)"
## 65 " 0.60 (0.01)" " 1.31 (0.01)"
## 66 " 0.10 (0.00)" " 0.89 (0.01)"
## 67 " 0.54 (0.01)" " 1.29 (0.01)"
```

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.

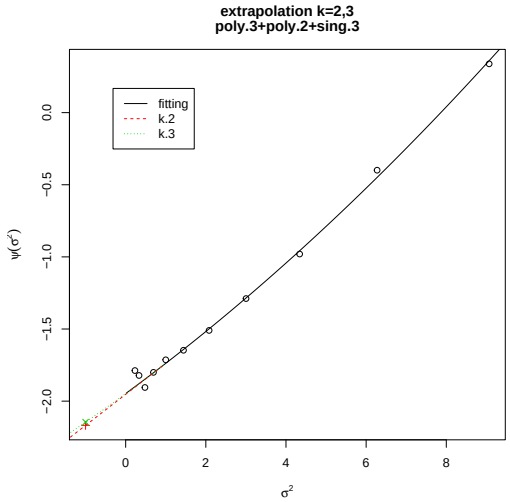
Diagnostics of model fitting

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
```



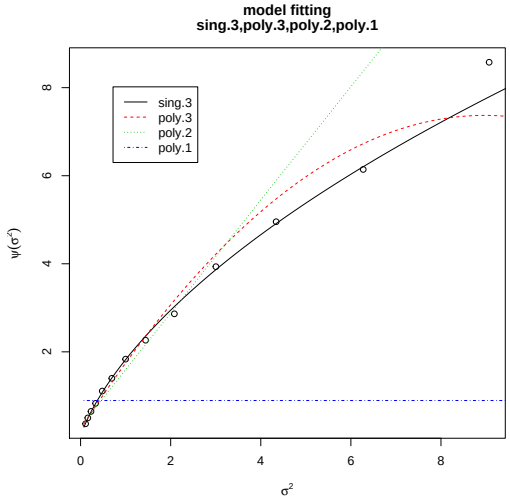

```
summary(lung73.sb[[62]], k=2:3) # p-values for k=2 and k=3
```

```
##
## Raw Bootstrap Probability (scale=1) : 95.67 (0.20)
##
## Hypothesis: alternative
##
## Corrected P-values for Models (percent,Frequentist):
##        k.2          k.3          sk.2         sk.3         beta0        beta1        aic     weight
## poly.3 98.49 (0.11) 98.40 (0.14) 96.36 (0.23) 96.21 (0.28) -1.95 (0.02)  0.21 (0.01)  -10.51 99.62
## poly.2 98.80 (0.05) 98.80 (0.05) 97.01 (0.12) 97.01 (0.12) -2.01 (0.01)  0.25 (0.00)    1.24  0.28
## sing.3 98.80 (0.05) 98.80 (0.05) 97.01 (0.12) 97.01 (0.12) -2.01 (0.01)  0.25 (0.00)    3.24  0.10
## poly.1 92.35 (0.10) 92.35 (0.10) 84.69 (0.21) 84.69 (0.21) -1.43 (0.01)  0.00 (0.00) 4407.26
##
## Best Model:  poly.3
##
## Corrected P-values by the Best Model and by Akaike Weights Averaging:
##         k.2          k.3          sk.2         sk.3         beta0        beta1
## best    98.49 (0.11) 98.40 (0.14) 96.36 (0.23) 96.21 (0.28) -1.95 (0.02)  0.21 (0.01)
## average 98.49 (0.11) 98.40 (0.14) 96.36 (0.23) 96.21 (0.27) -1.95 (0.02)  0.21 (0.01)
```

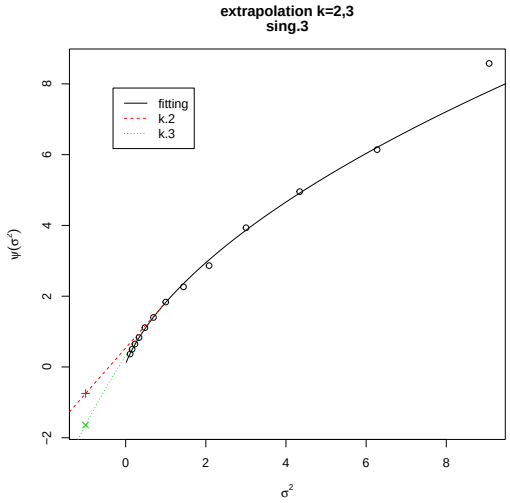
For cluster id=62 above, the quadratic model in terms of σ^2, namely, poly.3 ($\beta_0 + \beta_1\sigma^2 + \beta_2\sigma^4$) is the best model. The linear model poly.2 ($\beta_0 + \beta_1\sigma^2$) is also not bad. For computing AU and SI, we extrapolate the curve to $\sigma^2 = -1$. When $k = 1$, we 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) : 3.33 (0.18)
##
## Hypothesis: null
##
## Corrected P-values for Models (percent,Frequentist):
##        k.2          k.3          sk.2         sk.3         beta0       beta1       aic      weight
## sing.3 77.37 (0.46) 94.94 (0.18) 85.87 (0.40) 97.42 (0.09) 0.54 (0.01) 1.29 (0.01)    48.38 100.00
## poly.3 85.93 (0.29) 92.39 (0.29) 93.31 (0.17) 96.92 (0.14) 0.33 (0.00) 1.41 (0.01)   462.20
## poly.2 83.50 (0.32) 83.50 (0.32) 92.71 (0.19) 92.71 (0.19) 0.31 (0.00) 1.28 (0.01)  1307.79
## poly.1 18.54 (0.10) 18.54 (0.10) 37.08 (0.21) 37.08 (0.21) 0.89 (0.00) 0.00 (0.00) 52893.32
##
## 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    77.37 (0.46) 94.94 (0.18) 85.87 (0.40) 97.42 (0.09) 0.54 (0.01) 1.29 (0.01)
## average 77.37 (0.46) 94.94 (0.18) 85.87 (0.40) 97.42 (0.09) 0.54 (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
```

```
## 13.61 10.61  8.91  7.51  5.48  4.65  3.33  2.97  2.36  1.16  0.87  0.71  0.22
##
## 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 0.0824 (0.0112) 1.7413 (0.0189)  0.5214 (0.0169)
## poly.3 0.2456 (0.0047) 1.5887 (0.0142) -0.0886 (0.0024)
## poly.2 0.3110 (0.0043) 1.2850 (0.0098)
## poly.1 0.8949 (0.0039)
##
## Model Fitting:
##        rss      df pfit   aic
## sing.3    68.38 10 0.0000    48.38
## poly.3   482.20 10 0.0000   462.20
## poly.2  1329.79 11 0.0000  1307.79
## poly.1 52917.32 12 0.0000 52893.32
##
## 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.

Using k=3 for p-values

Now try $k = 3$ for computing p-values.

```
lung73.k3 <- sbpvclust(lung73.pvclust,lung73.sb, k=3) # compute p-values (k=3)
lung73.k3
```

```
##
## Cluster method: average
## Distance      : correlation
##
## Estimates on edges:
##
##       si    au    bp se.si se.au se.bp      v      c pchi
## 1  1.000 1.000 1.000 0.000 0.000 0.000     NA     NA    0
## 2  1.000 1.000 1.000 0.000 0.000 0.000 -4.720  0.124    0
## 3  1.000 1.000 1.000 0.001 0.001 0.000 -5.502 -0.593    0
## 4  1.000 1.000 1.000 0.003 0.003 0.000 -4.785 -0.891    0
## 5  1.000 1.000 1.000 0.000 0.000 0.000 -7.118 -0.101    0
## 6  0.536 0.817 0.658 0.005 0.004 0.002 -0.617  0.210    0
## 7  1.000 1.000 1.000 0.000 0.000 0.000 -5.935 -0.050    0
## 8  1.000 1.000 1.000 0.000 0.000 0.000 -4.275  0.126    0
## 9  1.000 1.000 1.000 0.000 0.000 0.000     NA     NA    0
## 10 0.991 0.996 0.994 0.001 0.000 0.000 -2.589  0.048    0
## 11 1.000 1.000 1.000 0.000 0.000 0.000 -3.563  0.045    0
## 12 1.000 1.000 1.000 0.000 0.000 0.000 -6.290 -0.042    0
## 13 1.000 1.000 1.000 0.000 0.000 0.000 -5.851 -0.500    0
## 14 0.977 0.989 0.986 0.002 0.001 0.001 -2.246  0.049    0
## 15 1.000 1.000 1.000 0.000 0.000 0.000 -5.349  0.082    0
## 16 0.999 1.000 1.000 0.000 0.000 0.000 -3.344  0.006    0
## 17 0.067 0.663 0.411 0.003 0.004 0.002 -0.034  0.259    0
## 18 0.999 0.999 1.000 0.005 0.005 0.000 -5.240 -0.907    0
## 19 0.898 0.958 0.901 0.004 0.002 0.001 -1.500  0.214    0
## 20 1.000 1.000 1.000 0.000 0.000 0.000 -5.326  0.086    0
## 21 1.000 1.000 1.000 0.000 0.000 0.000 -4.168  0.067    0
## 22 0.881 0.944 0.924 0.005 0.003 0.001 -1.524  0.093    0
## 23 0.597 0.910 0.487 0.004 0.002 0.002 -0.504  0.536    0
## 24 0.897 0.955 0.914 0.003 0.002 0.001 -1.535  0.167    0
## 25 0.909 0.960 0.929 0.004 0.002 0.001 -1.610  0.144    0
## 26 0.417 0.838 0.465 0.004 0.003 0.002 -0.345  0.432    0
## 27 0.162 0.795 0.304 0.004 0.003 0.002 -0.109  0.622    0
## 28 0.539 0.832 0.604 0.004 0.003 0.002 -0.617  0.353    0
## 29 0.576 0.861 0.578 0.004 0.002 0.002 -0.638  0.440    0
## 30 0.937 0.979 0.886 0.003 0.001 0.001 -1.609  0.401    0
## 31 1.000 1.000 1.000 0.000 0.000 0.000 -4.944  0.153    0
## 32 0.974 0.990 0.962 0.001 0.001 0.001 -2.045  0.275    0
## 33 0.800 0.910 0.858 0.005 0.003 0.001 -1.230  0.159    0
## 34 0.922 0.959 0.967 0.005 0.003 0.001 -1.818 -0.017    0
## 35 0.685 0.860 0.774 0.005 0.003 0.001 -0.959  0.207    0
## 36 0.936 0.973 0.941 0.004 0.002 0.001 -1.737  0.173    0
```

```
## 37 0.970 0.990 0.925 0.001 0.000 0.001 -1.885  0.447    0
## 38 0.116 0.577 0.518 0.003 0.004 0.002 -0.162  0.117    0
## 39 0.783 0.917 0.781 0.005 0.002 0.002 -1.095  0.320    0
## 40 0.999 1.000 0.998 0.000 0.000 0.000 -3.213  0.341    0
## 41 0.628 0.889 0.578 0.004 0.002 0.002 -0.673  0.477    0
## 42 0.998 0.999 0.997 0.000 0.000 0.000 -2.989  0.207    0
## 43 0.059 0.584 0.468 0.003 0.003 0.001 -0.067  0.149    0
## 44 0.216 0.845 0.310 0.004 0.003 0.002 -0.069  0.564    0
## 45 0.979 0.992 0.969 0.002 0.001 0.001 -2.135  0.272    0
## 46 0.265 0.921 0.213 0.004 0.002 0.002 -0.035  0.832    0
## 47 0.729 0.892 0.758 0.005 0.003 0.002 -0.987  0.288    0
## 48 0.025 0.813 0.264 0.005 0.003 0.002  0.092  0.539    0
## 49 0.150 0.673 0.450 0.003 0.004 0.002 -0.150  0.277    0
## 50 0.277 0.944 0.167 0.005 0.002 0.002 -0.024  0.991    0
## 51 0.915 0.959 0.946 0.004 0.003 0.001 -1.708  0.098    0
## 52 0.179 0.727 0.399 0.004 0.003 0.002 -0.168  0.424    0
## 53 0.089 0.690 0.363 0.004 0.004 0.002 -0.095  0.445    0
## 54 0.751 0.951 0.559 0.004 0.002 0.002 -0.697  0.548    0
## 55 0.997 0.999 0.994 0.000 0.000 0.000 -2.819  0.284    0
## 56 0.424 0.755 0.623 0.003 0.003 0.001 -0.501  0.187    0
## 57 0.796 0.942 0.677 0.004 0.002 0.002 -0.995  0.536    0
## 58 0.000 0.782 0.216 0.000 0.003 0.002  0.214  0.572    0
## 59 0.320 0.803 0.439 0.004 0.004 0.002 -0.247  0.401    0
## 60 0.000 0.792 0.206 0.000 0.004 0.002  0.123  0.699    0
## 61 0.152 0.567 0.553 0.003 0.004 0.002 -0.239  0.106    0
## 62 0.962 0.984 0.959 0.003 0.001 0.001 -1.953  0.215    0
## 63 0.490 0.815 0.574 0.004 0.003 0.002 -0.549  0.362    0
## 64 0.000 0.852 0.156 0.000 0.004 0.002  0.154  0.858    0
## 65 0.000 0.950 0.028 0.000 0.002 0.001  0.600  1.306    0
## 66 0.000 0.879 0.161 0.000 0.003 0.002  0.099  0.891    0
## 67 0.000 0.949 0.034 0.000 0.002 0.001  0.536  1.287    0
## 68 0.250 0.855 0.270 0.005 0.003 0.002 -0.162  0.777    0
## 69 0.100 0.758 0.299 0.004 0.003 0.002 -0.081  0.608    0
## 70 0.164 0.807 0.296 0.004 0.004 0.002 -0.096  0.633    0
## 71 0.264 0.860 0.260 0.005 0.003 0.002 -0.189  0.833    0
## 72 1.000 1.000 1.000 0.000 0.000 0.000     NA     NA    0
```

Redraw the dendrogram.

```
plot(lung73.k3, # p-values should be close to plot(lung.pv)
     cex=0.5, cex.pv=0.5, offset=c(0.6,0.1,0.1,0.1))
pvrect(lung73.k2, pv="si") # draw rectangle for clusters si>0.95
```