Statistical methods for microarray data

Goeman, Jelle Jurjen

Appendix A

Manual of the GlobalTest package

A.1 Introduction

This document shows the functionality of the R-package *globaltest*, whose main function tests whether a given group of genes is significantly associated with a clinical variable. The demonstration in this appendix focuses on practical use of the test. To understand the idea and the mathematics behind the test, and for more details on how to interpret a test result, we refer to the papers (Goeman et al., 2005, 2004).

In recent years there has been a shift in focus from studying the effects of single genes to studying effects of multiple functionally related genes or pathways (Al-Shahrour et al., 2004; Beissbarth and Speed, 2004; Boyle et al., 2004; Mootha et al., 2003; Smid and Dorssers, 2004; Zeeberg et al., 2003; Zhang et al., 2004). Most of the current methods for studying pathways involve looking at increased proportions of differentially expressed genes in pathways of interest. These methods do not identify pathways where many genes have altered their expression in a small way. The package *globaltest* was designed to address this issue.

The *globaltest* package tests whether a group of genes is associated with a clinical variable. A group of genes can be any pre-defined set, for example based in function (KEGG, GO) or location (chromosome, cytogenetic band). The clinical variable may be a phenotypic variable or an experimental condition. It may take the form of a 0/1 group indicator, of a continuous measurement or of a survival time.

The null hypothesis to be tested is that the expression pattern of the genes in the group is not related to the clinical variable. A significant test result has three parallel interpretations.

This chapter is the manual of the R package *globaltest*, that has been published on BioConductor as: J. J. Goeman and J. Oosting (2005). Globaltest: testing association of a group of genes with a clinical variable. R package, version 3.2.0. www.bioconductor.org.

- If a pathway is significantly associated with the clinical variable, the genes in the pathway are, on average, more associated with the clinical variable than would be expected if the null hypothesis were true. One can expect a sizeable proportion of genes to be associated with the clinical variable, but these associations might not be individually significant.
- If a pathway is significantly associated with the clinical variable, samples which have similar values of the clinical variable tend also to have similar expression pattern over the pathway.
- If a pathway is significantly associated with the clinical variable, there is good potential for predicting part of the variance of the clinical variable using the genes in the pathway.

In the examples below we use data sets that are available through the BioConductor web site. All the packages necessary to repeat the examples below are available from www.bioconductor.org. We use the AML/ALL data set (Golub et al., 1999) for illustration.

```
> library(globaltest)
> library(golubEsets)
> library(hu6800)
> library(vsn)
> data(golubMerge)
> golubM <- update2MIAME(golubMerge)
> golubX <- vsn(golubM)
```

This gives us a data set `golubX`, which is of the format *exprSet*, the standard format for gene expression data in BioConductor. It has 7,129 genes for 72 samples. We used *vsn* (Huber et al., 2002) to normalize the data. Any other normalization method may be used instead. Several phenotype variables are available with `golubX`, among them “ALL.AML”, the clinical variable that interests us.

In this document we use the globaltest based on BioConductor *exprSet* input. For examples using simple vector or matrix input, see `help(globaltest)`.

A.2 Global testing of a single pathway

Suppose we are interested in testing whether AML and ALL have a different gene expression pattern for certain pathways from the KEGG database.

First we load all KEGG pathways. We will use the rest in the next section.

```
> kegg <- as.list(hu6800PATH2PROBE)
> cellcycle <- kegg[["04110"]]
```

This creates a sorted list kegg of 140 pathways, each a vector of gene names. The vector cellcycle is one of them. It corresponds to the Cell Cycle pathway, "04110" in the KEGG database, which corresponds to 94 probe sets on the hu6800 chip. Suppose we are predominantly interested in this pathway. We want to know whether this group of genes is associated with the clinical outcome AML versus ALL.

It is advisable to always first test all genes to see if the overall gene expression pattern is different for different clinical outcomes. We can do this by saying

```
> gt.all <- globaltest(golubX, "ALL.AML")
```

The first input X should be the *exprSet* object, the second input Y the name of the clinical variable in pData(X). Alternatively we can give a matrix of expressions as X and a vector as Y.

The test result is stored in a *gt.result* object, which also contains all the information needed to draw the plots.

```
> gt.all
```

```
Global Test result:
Data: 72 samples with 7129 genes; 1 pathway tested
Model: logistic

  genes tested Statistic Q Expected Q sd of Q    p-value
1  7129   7129      53.992         10  1.9035 5.1616e-35
```

We conclude that there is ample evidence that the overall gene expression profile for all 7,129 genes is associated with the clinical outcome: samples with similar AML/ALL status tend to have similar expression profiles. In cases such as this one, in which the overall expression pattern is associated with the clinical variable, we can expect most pathways (especially the larger ones) also to be associated with it.

Because golubX is an *exprSet*, we could simply give the name of the phenotype variable "AML.ALL" as our Y input. Alternatively, we can give a vector here.

The Global Test allows three different kinds of clinical variables to be tested.

- A clinical variable defining two groups, i.e. having two values (using the logistic model). For a multi-valued clinical variable, the option *levels* can be used to set which groups are to be tested against each other.

- A continuously distributed measurement (using the linear model).

- A survival time (using the Cox model). In that case Y should contain the last observation time of each individual, and an extra argument d should be supplied which contains the event indicator, which has value *event* if an event occurred.

The function `globaltest` will automatically choose an appropriate model based on Y. To override the automatic choice, use the option *model*.

Now we test the Cell Cycle pathway that interests us:

```
> gt.cc <- globaltest(golubX, "ALL.AML", cellcycle)
> gt.cc

Global Test result:
Data: 72 samples with 7129 genes; 1 pathway tested
Model: logistic

  genes tested Statistic Q Expected Q sd of Q    p-value
1    94     94      69.443     10.312  3.2901 1.0166e-18
```

We conclude that the expression pattern of the cellcycle pathway is notably different between AML and ALL samples. However, as the test on all genes was significant we can generally expect most pathways to be significant as well. To get an impression of how "special" this pathway is, one can use the function `sampling`.

```
> gt.cc <- sampling(gt.cc)
> gt.cc

Global Test result:
Data: 72 samples with 7129 genes; 1 pathway tested
Model: logistic

  genes tested Statistic Q Expected Q sd of Q    p-value comp. p
1    94     94      69.443     10.312  3.2901 1.0166e-18   0.285
```

This gives an extra output column "comparative p", which is the fraction of random genesets of the same size as the cell cycle pathway (94 genes) which have a lower p-value than cell cycle itself. In this case around 28 % of 1,000 random 'pathways' of size 94 have a lower p-value than the Cell Cycle pathway. By default 1,000 random sets are sampled; this number can be changed with the option *ndraws*.

By default the p-value of globaltest is calculated using approximate formulas which are accurate for large sample size, but may be inaccurate for very small sample size. For 72 arrays they should be accurate enough. For very small sample sizes an alternative is to use the permutation version of globaltest. This recalculates the p-value on the basis of 10,000 permutations of the clinical variable.

```
> permutations(gt.cc)
```

```
Global Test result:
Data: 72 samples with 7129 genes; 1 pathway tested
Model: logistic
Using 10000 permutations of Y

  genes tested Statistic Q Expected Q sd of Q p-value
1    94     94       69.443     10.533  3.3604       0
```

The permutation p-value is not so accurate in the lower range as it is always a multiple of one over the number of permutations and also has some sampling variance. If desired, the number of permutations can be changed with the option *nperm* to get more accurate p-values.

It is also possible to adjust the globaltest for confounders or for known risk factors. For example in the Golub Data set we may be afraid for a disturbance due to that the fact that some samples were taken from peripheral blood while others were taken from bone marrow. We can correct for this using the option *adjust*. The option *adjust* can also be used when the study design is different from the simple 'two independent samples' design of the standard global test. In a paired design, for example, put the pair-identifier (as `factor`) in *adjust*.

The user may supply one or more names of covariates in the option *adjust* or supply *adjust* as a *data.frame*. The easiest way of adjustment, however, is by using a *formula* object as input for Y, as follows:

```
> globaltest(golubX, ALL.AML ~ BM.PB, cellcycle)
```

```
Global Test result:
Data: 72 samples with 7129 genes; 1 pathway tested
Model: logistic, ALL.AML ~ BM.PB
Adjusted: 99.8 % of variance of Y remains after adjustment

  genes tested Statistic Q Expected Q sd of Q    p-value
1    94     94       69.811      10.25  3.3189 1.6258e-18
```

The test result now also gives the percentage of the variance in Y that was left after the adjustment. It is a crude measure like $1 - R^2$. If the percentage is low, the adjustment already explained most of the variance of the outcome Y and there was not much residual variance left to test the influence of the genes. To see an example, adjust for "Source" instead of "BM.PB".

The option *adjust* may again be combined with the function *sampling*, but not with *permutation*.

A.3 Multiple global testing

It is also possible to test many pathways at once. To test all KEGG pathways we proceed as follows:

```
> gt.kegg <- globaltest(golubX, "ALL.AML", kegg)
```

The result `gt.kegg` can be displayed and prints a matrix whose rows correspond to the KEGG pathways. It gives the test results for each pathway. We can also display only some of them:

```
> gt.kegg[1:10]

Global Test result:
Data: 72 samples with 7129 genes; 10 pathways tested
Model: logistic

      genes tested Statistic Q Expected Q sd of Q    p-value
00271    10     10      10.103     8.0539  5.7226 2.8564e-01
00272    11     11      51.496    16.9070 12.0600 1.6643e-02
00628     2      2      18.066    22.5560 29.5330 3.8852e-01
00330    51     51      30.768     9.2072  2.9245 6.5854e-07
00920     6      6      12.558     6.5985  4.6089 1.0505e-01
05060    13     13      35.394     8.3092  4.4675 1.1091e-04
00450    14     14      39.648     9.8767  5.3131 2.2772e-04
04010   244    244      43.726    10.2410  2.3327 1.6381e-17
00510    26     26      39.145    10.2670  4.6621 4.5571e-05
04070    82     82      32.255     7.5731  2.0705 9.8340e-13
```

```
> gt.kegg["04110"]

Global Test result:
Data: 72 samples with 7129 genes; 1 pathway tested
Model: logistic

      genes tested Statistic Q Expected Q sd of Q    p-value
04110    94     94      69.443     10.312  3.2901 1.0166e-18
```

The same options described above for the single pathway globaltest can be applied to the multiple pathway version of globaltest as well.

Two functions allow further processing to be done on the test results. The function result extracts the whole matrix of test results, while the function p.value only extracts the vector of p-values. The latter function can be used for example when a correction for multiple testing is to be done. Note however that due to the extremely high correlations between the tests for different pathways, many multiple testing procedures are inappropriate for the Global Test. See the *multtest* package for details.

We might want to sort the pathways by their p-value, and show the top five. This can be done as follows

```
> sort.gt.kegg <- sort(gt.kegg)
> sort.gt.kegg[1:5]
```

```
Global Test result:
Data: 72 samples with 7129 genes; 5 pathways tested
Model: logistic

      genes tested Statistic Q Expected Q sd of Q    p-value
04060   246    246      77.853     9.9558  2.7046 1.4526e-30
04610    82     82     112.110     8.8155  3.3998 2.0881e-29
04510   169    169      61.849     9.2011  2.3298 2.4397e-28
04020   205    205      37.144     7.6385  1.6212 2.1932e-24
00590    31     31     213.070    13.5480  6.7527 1.5357e-23
```

A.4 Diagnostic plots

There are various types of diagnostic plots available to help the user interpret the globaltest result. The plot permutations can serve as a check whether the sample size was large enough not to use the permutation version of globaltest. The geneplot visualizes the influence of individual genes on the test result. The three plots sampleplot, checkerboard and regressionplot all visualize the influence of individual samples. Of these three, sampleplot is probably the most useful.

Permutations histogram

The permutations histogram plots the values of the test statistic Q calculated for permutations of the clinical outcome in a histogram. The observed value of Q for the true values of the clinical outcome is marked with an arrow.

```
> hist(permutations(gt.cc))
```

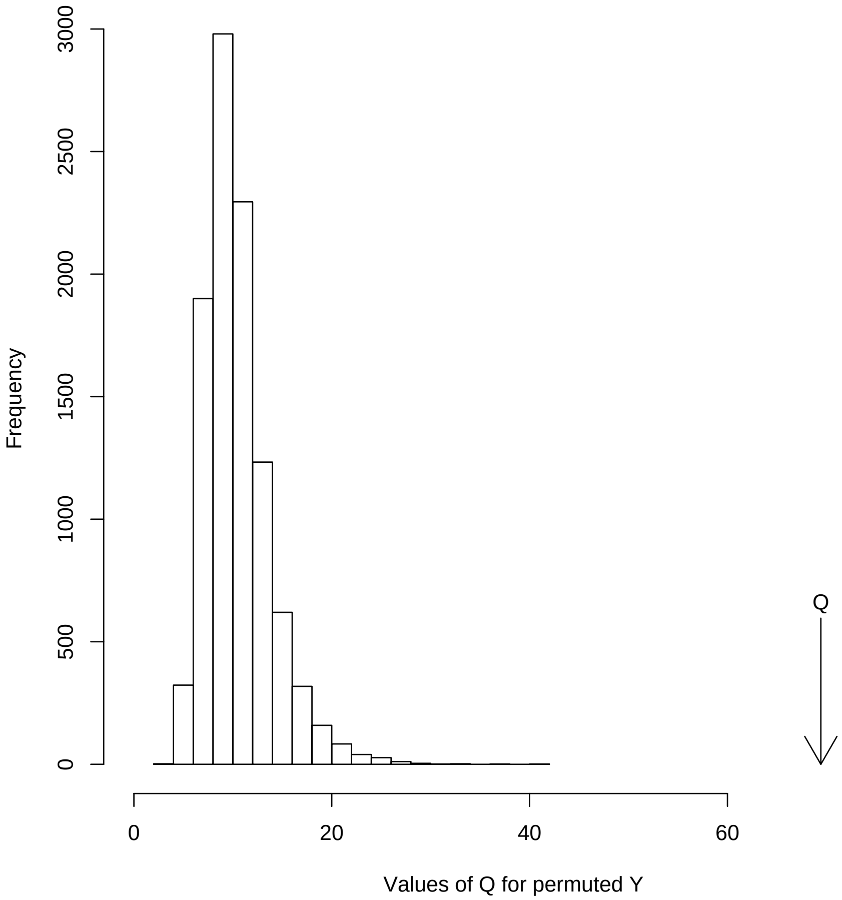

FIGURE A.1: *Histogram of values of the GlobalTest statistic Q for 10,000 permutations of the outcome variable, compared to the value of Q for the observed data.*

The output can be interpreted as a plot of the distribution of the test statistic under the null hypothesis that the pathway is not associated with the clinical variable. Strictly speaking, however, the permutation version of the Global Test is a different test with different properties (especially for survival data). It may give different p-values for small samples.

The function `permutations` may not be used when the adjusted version of `globaltest` was used.

Gene plot

The second diagnostic plot is the Gene Plot, which can be used to assess the influence of each gene on the outcome of the test. The Gene Plot gives a bar and a reference line for each gene tested. The bar indicates the influence of each gene on the test statistic.

A reference line for each bar gives the expected height of the bar under the null hypothesis that the gene is not associated with the clinical outcome (except in a survival model, where the expected height is zero). Marks indicate with how many standard deviations (under the null) the bar exceeds the reference line. Finally the bars are coloured to indicate a positive or a negative association of the gene with the clinical outcome.

The geneplot bars have two interpretations. In the first place, the bars are the Global Test statistic for the single gene pathway containing only that gene. A positive bar that is many standard deviations above the reference line therefore indicates a gene that is significantly associated with the clinical variable in Y. Secondly, the bars indicate the influence of the gene on the test result of the whole pathway (the test statistic for the group is the average of the bars for the genes). Removing a gene with a low bar (relative to the reference line) or a negative bar from the pathway will result in a lower p-value for the pathway, removing a gene with a tall positive bar will have the opposite effect.

To plot the geneplot, use any of the commands below:

```
> geneplot(gt.cc)
> geneplot(gt.kegg, "04110")
> geneplot(gt.kegg["04110"])
```

For a large number of genes the plot might become overcrowded. Use the option *genesubset* to plot only a subset of the genes, *labelsize* to resize the gene labels or *drawlabels = FALSE* to remove them. Alternatively, we can plot part of the geneplot later, as follows

```
> gp.cc <- geneplot(gt.cc)
> plot(gp.cc[1:40])
```

This allows one to look at subsets of a large pathway more closely. The return of the `geneplot` is an object of type *gt.barplot* containing the numbers and names appearing in the plot:

```
> gp.cc[1:10]
```

```
                   influence  expected        sd     z-score
U07563_cds1_at   179.10159883 26.469792 36.932786  4.13269143
```

```
X16416_at          8.49445145  7.459717 10.377423  0.09971017
U33841_at         30.67349290  6.143978  8.543243  2.87121827
U67092_at          0.06648514  4.909773  6.697347 -0.72316514
U67092_s_at        0.03786093  4.317985  5.653305 -0.75710132
X91196_s_at        3.61779645  2.516948  3.409389  0.32288738
U49844_at         73.44818990  6.217861  8.665433  7.75844974
HG4433-HT4703_at  21.67168778  9.114009 12.659503  0.99195670
X59798_at          1.09258726  7.284671  8.735287 -0.70885864
X51688_at         54.98210529 14.400044 20.094406  2.01957013
```

The option *scale* can be used to rescale the bars to have unit standard deviation.

Sample plot

The Sample Plot looks very similar to the Gene Plot and visualizes the influence of the individual samples on the test result. It has a bar and a reference line for each sample tested. The bar indicates the influence of each sample on the test statistic, similar to the `geneplot`. The direction of the bar (upward or downward) indicates evidence against or in favour of the null hypothesis. If a sample has a positive bar, its expression profile is relatively similar to that of samples which have the same value of the clinical variable and relatively unlike the profile of the samples which have a different value of the clinical variable. If the bar is negative, it is the other way around: the sample is more similar in expression profile to samples with a different clinical variable. A small p-value will therefore generally coincide with many positive bars. If there are still tall negative bars, these indicate deviating samples: removing a sample with a negative bar would result in a lower p-value.

If the null hypothesis is true the expected influence is zero. Marks on the bars indicate the standard deviation of the influence of the sample under the null hypothesis. Finally the bars are coloured to distinguish the samples. In a logistic model the colours differentiate between the original groups, in an unadjusted linear model they differentiate the values above the mean from the values below the mean of Y. In an adjusted linear or the survival model they distinguish positive from negative residuals after fitting the null model.

Again, either of the commands below gives the same output.

```
> sampleplot(gt.cc)
> sampleplot(gt.kegg, "04110")
> sampleplot(gt.kegg["04110"])
```

The options of `sampleplot` and the resulting *gt.barplot* object are be handled in the same way as described under "geneplot".

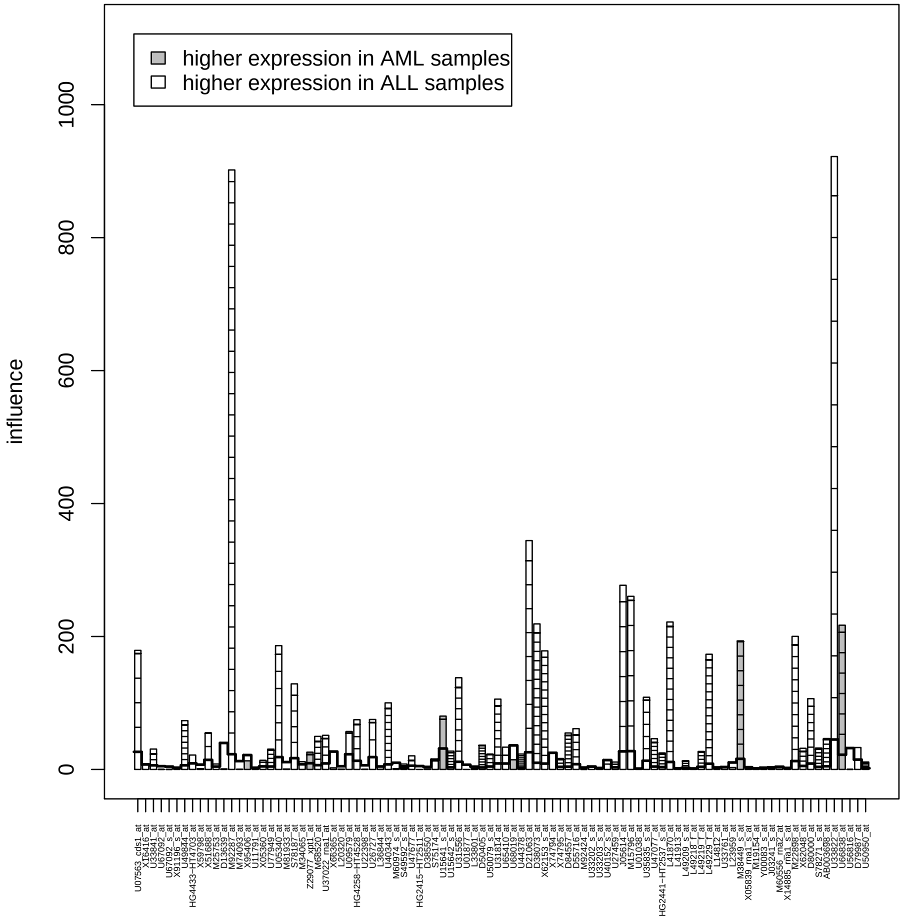

FIGURE A.2: *Gene Plot of 40 genes from the cell cycle pathway in the AML/ALL data. The height of the bar measures association of the expression of that gene with the outcome variable.*

Checkerboard plot

The fourth and fifth diagnostic plot can both also be used to assess the influence of each of the samples on the test result. The checkerboard plot visualizes the similarity between samples. It makes a square figure with the samples both on the X and on the Y-axis, so that it contains all comparisons between the samples. Samples which are relatively similar are coded white and samples which are relatively dissimilar are coded black.

For easier interpretation the samples are sorted by their clinical outcome. If

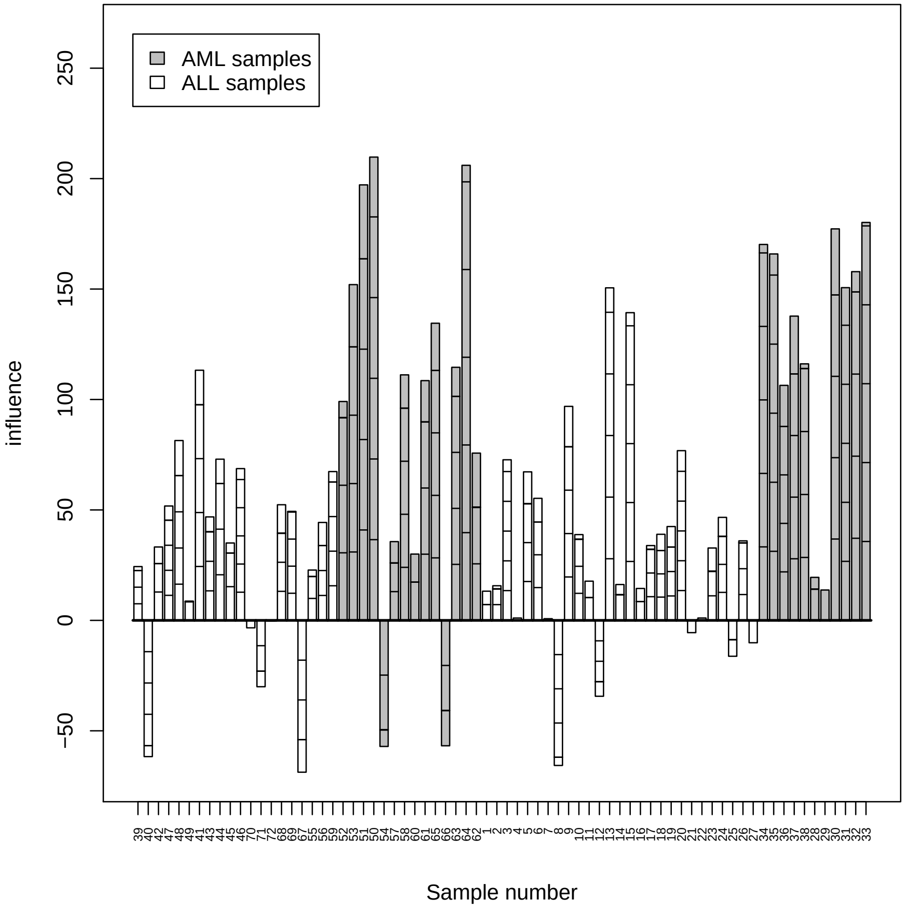

FIGURE A.3: *Sample Plot of the samples in the AML/ALL data set, based on the expression profile of the cell cycle pathway. Positive bars indicate samples whose expression profile is similar to the other samples in the same group; Negative bars indicate samples whose expression profile is similar to samples in the opposite group.*

the test was (very) significant and the clinical outcome has two values, a typical block-like structure will appear. If the clinical outcome was continuous and the test is significant, the black squares will tend to stick together around the upper left and lower right corners. By looking at these patterns some things can be learned about the structure of the data. For example, by looking at samples which deviate from the main pattern, outlying samples can be detected.

```
> checkerboard(gt.cc)
> checkerboard(gt.kegg, "04110")
```

The function `checkerboard` also has options *labelsize* and *drawlabels*. It returns a legend to link the numbers appearing in the plot if *drawlabels = FALSE* to the sample names.

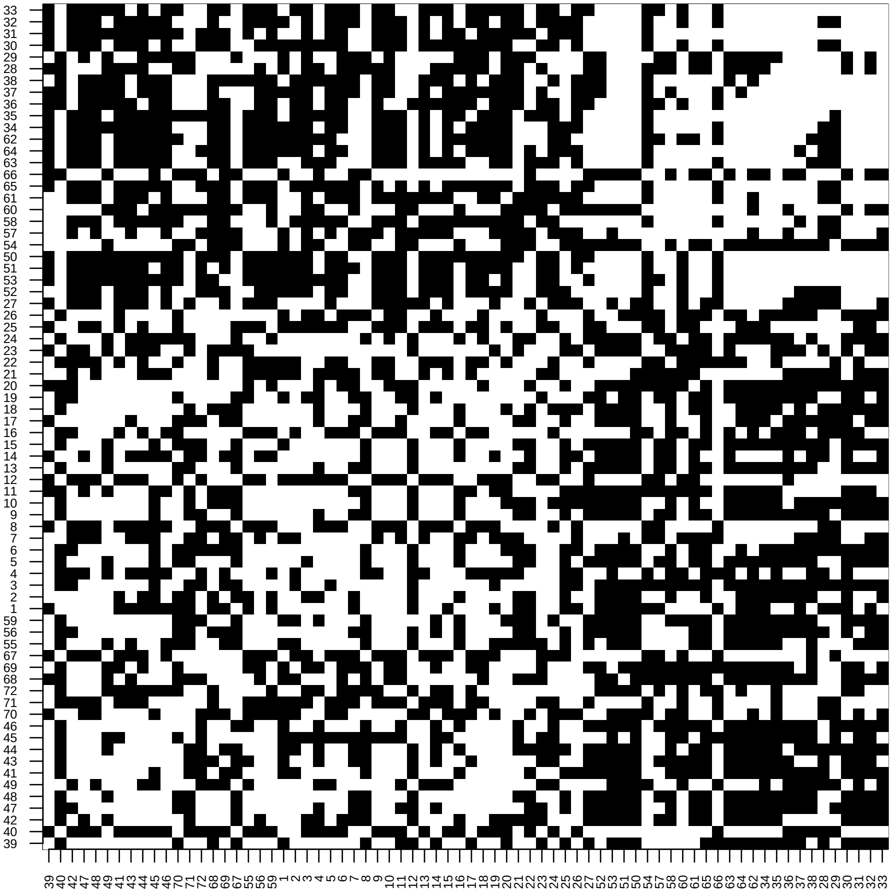

FIGURE A.4: *Checkerboard plot of the samples in the AML/ALL data set, based on the cell cycle pathway. White blocks indicate that samples have similar expression profile, black indicates dissimilar expression profile.*

Regression plot

Using the regression plot an assessment can be made of the influence of each sample on the result of the test. It is an alternative visualization of the sampleplot.

The regression plot plots all pairs of samples, just like the checkerboard plot, but now showing the covariance between their clinical outcomes on the X-axis and the covariance between their gene expression patterns on the Y-axis. The comparisons of each sample with itself have been excluded.

The test statistic of the Global Test can be seen as a regression-coefficient for this plot, so it is visualized by drawing a least squares regression line. If this regression line is steep, the test statistic has a large value (and is possibly significant).

The influence of specific samples can be assessed by drawing a second regression line through only those points in the plot, which are comparisons involving the sample of interest. For example if we are interested the sample with sample name "1", we take the points corresponding to the pairs (1,2) up to (1,72). If the regression line drawn through only these points deviates much from the general line, the sample deviates from the general pattern. This is especially the case if this line has a negative slope, which means that the sample is more similar in its gene expression pattern to the samples with a different clinical outcome than to samples with a similar clinical outcome.

If we want to test sample "1", we say:

```
> regressionplot(gt.cc, sampleid = "1")
> regressionplot(gt.kegg, "04110", sampleid = "1")
```

We can also use this plot for a group of samples, saying for example:

```
> regressionplot(gt.cc, sampleid = c("1", "2"))
```

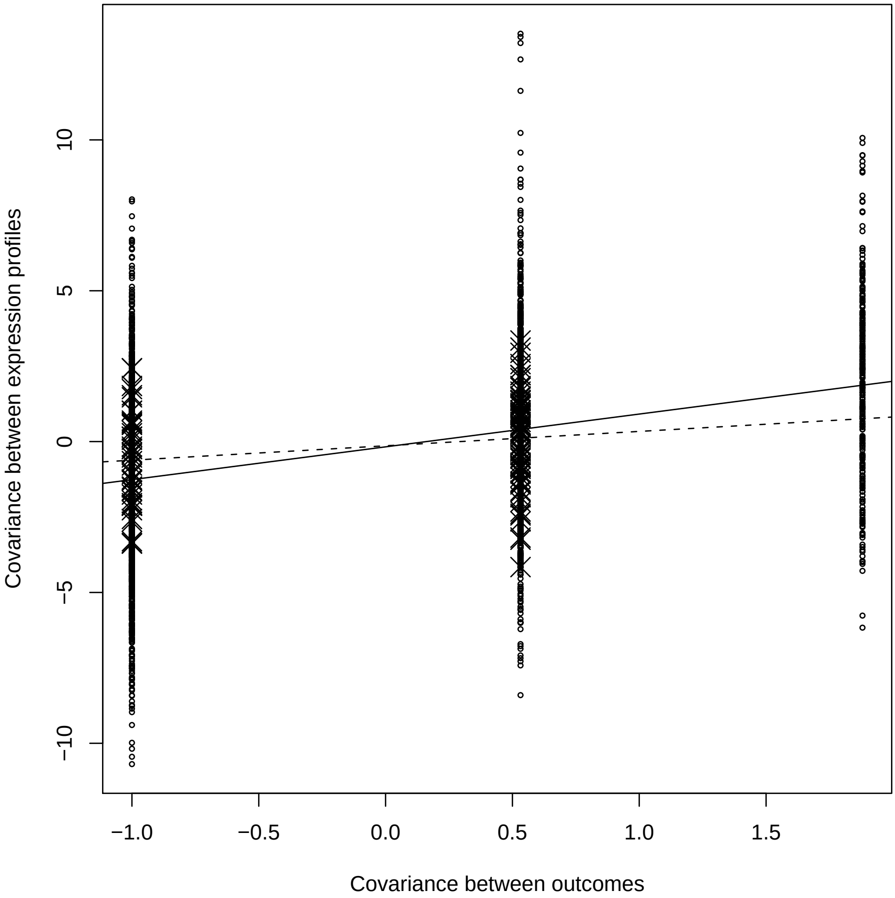

FIGURE A.5: *Regression Plot of the AML/ALL data set, based on the cell cycle pathway, showing the regression of covariance in expression profile on covariance of outcome measure. The dotted line is based only on pairs of samples involving samples "1" and "2".*