Global testing of differential gene expression

Manuela Hummel Reinhard Meister Ulrich Mansmann

October 26, 2021

Contents

1 Abstract

In studies about differential gene expression between different clinical diagnoses the main interest may often not be in single genes but rather in groups of genes that are associated with a pathway or have a common location in the genome. In such cases it may be better to perform a global test because the problems of multiple testing can be avoided. The approach presented here is an ANCOVA global test on phenotype effects and gene–phenotype interaction.

The test is generalized to groups of categorical variables (e.g. SNPs) and even mixed data by a likelihood ratio approach.

Testing many groups simultaneously is also possible. This, of course, causes again need for correction for multiple testing. Besides the standard approaches for correction we introduce closed and hierarchical testing procedures in which the experiment–wise error rate equals the required level of confidence of the overall test.

This document was created using R version 4.1.1 and versions 4.12.0 and 5.48.0 of the packages *GlobalAncova* and *globaltest*, respectively.

2 Major Changes to Previous Versions

Version 4.x.x

- There is a new function `gGlobalAncova` for generalized linear models: groups of tested variables can be quantitative, categorical, ordinal and even of mixed types. The new function `Plot.features` allows visualization of variable-wise contributions to the global test statistic in this general setting.
- There is a new function `gGlobalAncova.hierarchical` for hierarchical testing. Results are stored in objects of the new class *GAhier* and are accessible via corresponding methods.

Version 3.14.x

The testing of collections of functional gene sets (Gene Ontology, Broad Institute's gene sets) was adapted to new functions in package *globaltest* (version $> 5.0.0$) and can now be performed using the functions `GAGO` and `GABroad`.

Version 3.3.3

- The permutation approach is now implemented in *C* and therefore faster.
- If the number of possible phenotype permutations is smaller than the number specified in *perm* (i.e. in very small sample sizes), all possible permutations are considered for the permutation test.

Version 3.x.x

- Besides the permutation-based p-values also asymptotic p-values based on an approximation of the distribution of the test statistic are provided. Theoretical F-test p-values are no longer displayed since they are not valid in case of correlations or non-normality.

- The focus level procedure for finding interesting Gene Ontology subgraphs from Goeman and Mansmann (2007) [4] was adapted for the use with `GlobalAncova`.

- Sequential and type III decompositions of the residual sum of squares, adjustment for global covariates and pair-wise comparisons of different levels of a categorical factor are implemented. These functionalities are described in the additional vignette *GlobalAncovaDecomp.pdf*.

Version 2.x.x

- The major modification in the new version is the transfer from simple two group comparisons to a general linear model framework where arbitrary clinical variables (in especially with more groups or also continuous ones), time trends, gene–gene interactions, co–expression and so forth can be analysed.

- According to the new framework also the diagnostic plots are more flexible. The variable defining the coloring of bars can now be specified by the user, see section 7 for details.

3 Introduction

The ANCOVA global test is a test for the association between expression values and clinical entities. The test is carried out by comparison of linear models via the extra sum of squares principle. If the mean expression level for at least one gene differs between corresponding models the global null hypothesis, which is the intersection of all single gene null hypotheses, is violated. As our test is based on the sum of gene-wise reduction in sum of squares due to phenotype, all systematic differences in gene expression between phenotypes equally contribute to the power of the test.

Single genes are not, in general, the primary focus of gene expression experiments. The researcher might be more interested in relevant pathways, functional sets or genomic regions consisting of several genes. Most of the current methods for studying pathways analyse differential expression of single genes. In these methods pathways where many genes show minor changes in their expression values may not be identified. Goeman's global test and the ANCOVA global test were designed to address this issue.

Applying global tests for differential expression in pathways substantially reduces the number of tests compared to gene-wise multiple testing. The amount of correction for multiple testing decreases. Function (KEGG, GO) or location (chromosome, cytoband) could be used as grouping criteria, for example.

We want to compare our method with the global test of Goeman et al. (2004) [3]. Our function `GlobalAncova` tests whether the expectation of expression levels differs between biological entities for a given group of genes.

The GlobalAncova framework is generalized to groups of categorical variables (e.g. SNPs) and even mixed data. The test is still based on the principle of model comparison. Instead of considering differences in sums of squares we use differences in model deviances, which leads to a likelihood ratio approach.

This vignette has its focus on the practical use of the test. For more details about the mathematical background and the interpretation of results, we refer to the papers by Mansmann and Meister (2005) [8] and Hummel, Meister and Mansmann (2008) [7].

First we load the packages and data we will use.

```
> library(GlobalAncova)
> library(globaltest)
> library(golubEsets)
> library(hu6800.db)
> library(vsn)
> data(Golub_Merge)
> golubX <- justvsn(Golub_Merge)
```

This creates a dataset `golubX`, which is of the format *ExpressionSet*, the standard format for gene expression data in BioConductor. It consists of 7129 genes and 72 samples (the data are from [5]). We used *vsn* to normalize the data. Other appropriate normalization methods may be used as well. From several phenotype variables we use "ALL.AML" as the clinical diagnoses of interest. ALL and AML are two types of acute leukemia. There are 47 patients with ALL and 25 with AML.

4 Global Testing of a Single Pathway

4.1 Golub Data and Cell Cycle Pathway

Suppose we are interested in testing whether AML and ALL have different gene expression patterns for certain pathways, for example from the KEGG database. With *globaltest* we answer the question whether the expression profile has prognostic power with respect to diagnosis of AML or ALL. *GlobalAncova* asks for differences in mean expression between the two clinical groups.

4.1.1 Testing all Genes

We start by applying our test to all genes in the Golub dataset so that differences in the overall gene-expression pattern can be demonstrated.

```
> gr <- as.numeric(golubX$ALL.AML=="ALL")
> ga.all <- GlobalAncova(xx=exprs(golubX), group=gr, covars=NULL, perm=100)
```

The first input *xx* is a 7129×72 matrix that contains the expression values of all genes and samples. Missing values in the expression matrix *xx* are not allowed because otherwise gene-wise linear models could not be summarized adequately to a global group statement. If missing values occur we propose either leaving out the genes with missing values (i.e. the corresponding rows in the gene expression matrix), or imputing the data before applying *GlobalAncova*. An easy way to do the latter would be for example to calculate linear models for each gene using the available model variables (e.g. phenotype group labels). Missing values can then be estimated based on the resulting model parameters and the actual values of phenotype variables of the corresponding samples. The use of more sophisticated imputation methods [12] would be computationally

expensive and is not implemented in *GlobalAncova*. Note that we did not yet evaluate how data imputation affects *GlobalAncova* results and whether the easier imputation methods described above yield similar results as the more complex approaches. The second input *group* in the `GlobalAncova` function is a vector that defines the clinical diagnosis for the 72 patients.

Note that *GlobalAncova* is not restricted to the analysis of dichotomous phenotype groups. More complex tasks like variables with more groups or also continuous ones, time trends, gene–gene interactions and co–expression can be performed as well. Some examples will be given in section 4.2. The realization of such tasks is done by definition of two linear models that shall be compared via the extra sum of squares principle. Hence model formulas for the full model containing all parameters and the reduced model, where the terms of interest are omitted, have to be given. An alternative is to provide the formula for the full model and a character vector naming the terms of interest. Those names can be chosen by previous output of the `GlobalAncova` function. Consequently we could run the same analysis as above with two possible further function calls shown below (output is omitted). In both cases a data frame with information about all variables for each sample is required. In the case of microarray data this can be the corresponding `pData` object.

```
> GlobalAncova(xx=exprs(golubX), formula.full=~ALL.AML, formula.red=~1,
+              model.dat=pData(golubX), perm=100)
> GlobalAncova(xx=exprs(golubX), formula.full=~ALL.AML, test.terms="ALL.AMLAML",
+              model.dat=pData(golubX), perm=100)
```

To avoid alpha–inflation due to correlated data and effects of non–normality of the data tests for significance of the resulting F–ratios are performed using a permutation test approach. We apply permutation of samples which is equivalent to permuting rows of the full design matrix. Note that permutation is only conducted for such columns of the design matrix that correspond to the variables of interest. Values of additional covariates remain in the original order. This prevents us for destroying covariate effects. Still the permutation approach is not optimal since residuals may be correlated. However, this does not seem to be a severe problem. The argument *perm* defines the number of permutations, which is 10,000 for default. Here we set *perm* to just 100 or 1000 so that creating this vignette will not last too long. For getting more reliable results one should recompute the examples with more permutations.
As an alternative to the permutation approach an approximation of the F–statistic nominator according to [11] yields asymptotic p–values. Note that the approximation is not feasible for very large gene groups since the huge gene expression covariance matrix has to be estimated, which is not possible for too many genes. The default value for group size (*max.group.size*) is 2500, groups above this size are treated by the permutation approach. When using work stations with good working memory this number may be increased. The estimation of the covariance matrix is carried out with the R package *corpcor* from [13].
Whether the permutation–based or the asymptotic p–values or both should be calculated is controlled by the argument *method*.

The result of the `GlobalAncova` function is a typical ANOVA table with information about sums of squares, degrees of freedom and mean sums of squares for the effect and error term, respectively. Besides F–statistics there are given

either p–values from the permutation test or the asymptotic p–values or both. The names of all involved parameters are displayed as well as the name(s) of the tested effect(s).

```
> ga.all
```

```
$effect
[1] "group"

$ANOVA
             SSQ     DF        MS
Effect  29173.83   7129 4.0922748
Error  340387.99 499030 0.6820992

$test.result
           [,1]
F.value 5.99953
p.perm  0.00000

$terms
[1] "(Intercept)" "group"
```

From this result we conclude that the overall gene expression profile for all 7129 genes is associated with the clinical outcome. This means that samples with different AML/ALL status tend to have different expression profiles. We expect most pathways (especially the ones containing many genes) also to be associated with the phenotype groups.

If we apply Goeman's global test we get

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

```
   p-value Statistic Expected Std.dev #Cov
1 7.02e-20      7.89     1.41   0.275 7129
```

Both tests show that the data contain overwhelming evidence for differential gene expression between AML and ALL.

4.1.2 Testing the Cell Cycle Pathway

Now we ask the more specific question of whether there is evidence for differential gene expression between both diagnoses restricted to genes belonging to the cell cycle pathway. First we load all KEGG pathways.

```
> kegg <- as.list(hu6800PATH2PROBE)
```

The list `kegg` consists of 229 pathways. Each pathway is represented by a vector of gene names. We are mainly interested in the cell cycle pathway which has the identifier "04110" in the KEGG database. It corresponds to 111 probe sets on the hu6800 chip.

```
> cellcycle <- kegg[["04110"]]
```

We apply the global test to this pathway using the option *test.genes*.

```
> ga.cc <- GlobalAncova(xx=exprs(golubX), group=gr, test.genes=cellcycle,
+                       method="both", perm=1000)
> ga.cc

$effect
[1] "group"

$ANOVA
             SSQ   DF        MS
Effect  549.3037  111 4.9486817
Error  5304.0006 7770 0.6826256

$test.result
                  [,1]
F.value  7.249482e+00
p.perm   0.000000e+00
p.approx 5.591746e-12

$terms
[1] "(Intercept)" "group"
```

Also with *globaltest* we get a very small p–value

```
> gt.cc <- gt(ALL.AML, golubX, subsets=cellcycle)
> gt.cc

   p-value Statistic Expected Std.dev #Cov
1 1.24e-11      9.38     1.41   0.436  111

   p-value Statistic Expected Std.dev #Cov
1 1.24e-11      9.38     1.41   0.436  111
```

The test results clearly indicate that the expression pattern of the cell cycle pathway is different between the two clinical groups.

4.1.3 Adjusting for Covariates

Covariate information can be incorporated by specifying the *covars* option.

For example if we want to adjust for whether samples were taken from bone marrow or from peripheral blood (`BM.PB`), we can do this by

```
> ga.cc.BMPB <- GlobalAncova(xx=exprs(golubX), group=gr, covars=golubX$BM.PB,
+                            test.genes=cellcycle, method="both", perm=1000)
> ga.cc.BMPB

$effect
[1] "group"

$ANOVA
             SSQ   DF        MS
Effect  540.5233  111 4.8695790
Error  5146.4091 7659 0.6719427
```

```
$test.result
                 [,1]
F.value  7.247015e+00
p.perm   0.000000e+00
p.approx 8.795910e-12

$terms
[1] "(Intercept)" "group"       "covarsPB"
```

With the more general function call we would simply adjust the definitions of model formulas, namely *formula.full* $= \sim ALL.AML + BM.PB$ and *formula.red* $= \sim BM.PB$.

The source of the samples does not seem to have an explanatory effect on the outcome since F–statistics and p–values are very similar to the model without adjustment.

With the *globaltest* we get a similar p–value.

```
> gt.cc.BMPB <- gt(ALL.AML ~ BM.PB, golubX, subsets=cellcycle)
> gt.cc.BMPB

   p-value Statistic Expected Std.dev #Cov
1 1.18e-12      9.66     1.47   0.441  111

   p-value Statistic Expected Std.dev #Cov
1 1.18e-12      9.66     1.47   0.441  111
```

Permutation based p–values can also be obtained with Goeman's test, however only when covariates are absent.

4.2 van't Veer Data and p53–Signalling Pathway

We present another example from a study on breast cancer from van't Veer et al. (2002) [14]. This example illustrates how more complex tasks than comparing just two clinical groups can be performed with *GlobalAncova*. A subset of the data consisting of the expression values for 96 patients without *BRCA1* or *BRCA2* mutations is available with the package. The dataset (`vantVeer`) is restricted to 1113 genes associated with 9 cancer related pathways that are provided as a list named (`pathways`), too. We take one gene from the original data additionally to the expression set, namely 'AL137718'. This gene is part of the original van't Veer prognosis signature. We will later use it to demonstrate how signature genes can be related to pathways. Information about some of the originally surveyed covariates is stored in `phenodata`. The tumour suppressor protein p53 contributes as a transcription factor to cell cycle arrest and apoptosis induction. Therefore, first the p53-signalling pathway is selected as a candidate, where differential expression between relevant prognostic groups, defined by the development of distant metastases within five years, was expected.

```
> data(vantVeer)
> data(phenodata)
> data(pathways)
> metastases <- phenodata$metastases
> p53 <- pathways$p53_signalling
```

We get a significant result with the global ANCOVA.

```
> vV.1 <- GlobalAncova(xx=vantVeer, group=metastases, test.genes=p53,
+                      method="both", perm=1000)
> vV.1

$effect
[1] "group"

$ANOVA
             SSQ   DF         MS
Effect  2.893417   33 0.08767929
Error  97.424573 3102 0.03140702

$test.result
                [,1]
F.value  2.791710175
p.perm   0.014000000
p.approx 0.009093267

$terms
[1] "(Intercept)" "group"
```

4.2.1 Analysis of Various Clinical Groups

In the new version of the package also clinical variables with more than two groups can be considered. For demonstration we investigate differential expression for the three ordered levels of tumour grade.

```
> vV.3 <- GlobalAncova(xx=vantVeer, formula.full=~grade, formula.red=~1,
+           model.dat=phenodata, test.genes=p53, method="both", perm=1000)
> vV.3

$effect
[1] "grade.L" "grade.Q"

$ANOVA
             SSQ   DF         MS
Effect  3.638565   66 0.05512977
Error  96.679425 3069 0.03150193

$test.result
               [,1]
F.value  1.75004422
p.perm   0.03300000
p.approx 0.03463237

$terms
[1] "(Intercept)" "grade.L"     "grade.Q"
```

4.2.2 Gene–Gene Interaction

Now we want to go into the matter of other interesting biological questions. For example one might ask if there exists interaction between the expression of genes which the authors in [14] presented as signature for prediction of cancer recurrence and the expression of genes in a certain pathway. This question can be answered by viewing the expression values of the signature genes as linear regressors and to test their effects on the expression pattern of the pathway genes. For demonstration we pick the signature gene "AL137718", which is not part of any of the pathways, and test its effect on the p53–signalling pathway. Assume that we also want to adjust for the Estrogen receptor status. The analysis can be carried out in the following way.

```
> signature.gene <- "AL137718"
> model <- data.frame(phenodata, signature.gene=vantVeer[signature.gene, ])
> vV.4 <- GlobalAncova(xx=vantVeer, formula.full=~signature.gene + ERstatus,
+                      formula.red=~ERstatus, model.dat=model, test.genes=p53,
+                      method="both", perm=1000)
> vV.4

$effect
[1] "signature.gene"

$ANOVA
             SSQ   DF         MS
Effect  2.667014   33 0.08081859
Error  89.867452 3069 0.02928232

$test.result
               [,1]
F.value  2.75997881
p.perm   0.01200000
p.approx 0.01387833

$terms
[1] "(Intercept)"    "signature.gene" "ERstatuspos"
```

Assuming a significance level of 0.05 we get a significant effect of the signature gene on the p53–signalling pathway.

4.2.3 Co–Expression

Next we want to analyse co–expression regarding the clinical outcome of building distant metastases within five years. This can be done by simply adding the variable `metastases` to the full and reduced model, respectively. Such layout corresponds to testing the linear effect of the signature gene stratified not only by Estrogen receptor status but also by `metastases`.

```
> vV.5 <- GlobalAncova(xx=vantVeer, formula.full=~metastases + signature.gene + ERstatus,
+                      formula.red=~metastases + ERstatus, model.dat=model,
+                      test.genes=p53, method="both", perm=1000)
> vV.5
```

```
$effect
[1] "signature.gene"

$ANOVA
             SSQ   DF         MS
Effect  2.284391   33 0.06922396
Error  87.463681 3036 0.02880885

$test.result
               [,1]
F.value  2.40287099
p.perm   0.03800000
p.approx 0.02876237

$terms
[1] "(Intercept)"    "metastases"     "signature.gene"
[4] "ERstatuspos"
```

Again we get a significant result.

Supposably the most interesting question in this case concerns differential co–expression. Differential co–expression is on hand if the effect of the signature gene behaves different in both `metastases` groups. In a one dimensional context this would become manifest by different slopes of the regression lines. Hence what we have to test is the interaction between `metastases` and `signature.gene`.

```
> vV.6 <- GlobalAncova(xx=vantVeer, formula.full=~metastases * signature.gene + ERstatus,
+                      formula.red=~metastases + signature.gene + ERstatus,
+                      model.dat=model, test.genes=p53, method="both", perm=1000)
> vV.6

$effect
[1] "metastases:signature.gene"

$ANOVA
             SSQ   DF         MS
Effect  2.520643   33 0.07638311
Error  84.943038 3003 0.02828606

$test.result
               [,1]
F.value  2.70038011
p.perm   0.02500000
p.approx 0.01829782

$terms
[1] "(Intercept)"               "metastases"
[3] "signature.gene"            "ERstatuspos"
[5] "metastases:signature.gene"
```

We observe a significant differential co-expression between the chosen signature gene and the p53–signalling pathway.

With *globaltest* we can also test gene-gene interaction, also adjusted for phenotype groups. But it is not possible to test for differential co-expression or the influence of more than one signature gene on a pathway. On the other hand globaltest is able to deal with survival times as clinical outcome.

5 Testing Several Sets Simultaneously

Systems biology involves the study of mechanisms underlying complex biological processes as integrated systems of many diverse interacting components, often referred to as pathways.

We regard the possibility to investigate differential gene expression simultaneously for several of those pathways as a contribution towards understanding biological relevant relations.

The user can apply `GlobalAncova` to compute p–values for a couple of pathways with one call by specifying the *test.genes* option. The members of each pathway to be tested must belong to genes in the expression–matrix. Afterwards a suitable correction for multiple testing has to be applied. An alternative based on the closed testing approach is described later.

Suppose for example that for sake of simplicity we want to test the first four of the cancer related pathways with the van't Veer data. We proceed as follows.

```
> metastases <- phenodata$metastases
> ga.pw <- GlobalAncova(xx=vantVeer, group=metastases, test.genes=pathways[1:4],
+                       method="both", perm=1000)
> ga.pw

                           genes  F.value p.perm     p.approx
androgen_receptor_signaling    72 2.389837  0.009 3.045062e-03
apoptosis                     187 1.968467  0.013 7.694809e-04
cell_cycle_control             31 4.639853  0.001 2.958656e-05
notch_delta_signalling         34 1.497222  0.154 8.537110e-02
```

The result is a matrix whose rows correspond to the different pathways.

With the `globaltest` we get a similar matrix.

```
> gt.options(transpose = TRUE)
> gt.pw <- gt(metastases, vantVeer, subsets=pathways[1:4])
> gt.pw

                            p-value Statistic Expected Std.dev #Cov
androgen_receptor_signaling 0.008540      2.48     1.05   0.409   72
apoptosis                   0.012427      2.05     1.05   0.327  187
cell_cycle_control          0.000216      4.70     1.05   0.515   31
notch_delta_signalling      0.131999      1.57     1.05   0.496   34
```

5.1 Simultaneous Adjustment of p–values

Next we show how to extract p–values for correction for multiple testing. Note however that due to the extremely high correlations between these tests, many procedures that correct for multiple testing here are inappropriate. An appropriate way of adjusting would be for example the method of Holm (1979) [6].

An alternative to such adjustments that is not affected by correlations between tests is a closed testing procedure. For this approach you need a family of null hypotheses that is closed under intersection. Then a single hypothesis can be rejected at level α if it is rejected along with all hypotheses included in it ([9]).

For the adjustment according to Bonferroni and Holm we build a vector of the raw p–values. The function `p.adjust` provides several adjusting methods for any vector of raw p-values. For the output of function `gt` there is a specific `p.adjust` method.

```
> ga.pw.raw <- ga.pw[ ,"p.perm"]
> ga.pw.adj <- p.adjust(ga.pw.raw, "holm")
> ga.result <- data.frame(rawp=ga.pw.raw, Holm=ga.pw.adj)
> ga.result
```

```
                             rawp  Holm
androgen_receptor_signaling 0.009 0.027
apoptosis                   0.013 0.027
cell_cycle_control          0.001 0.004
notch_delta_signalling      0.154 0.154
```

```
> gt.result <- p.adjust(gt.pw)
> gt.result
```

```
                                holm  p-value Statistic Expected
androgen_receptor_signaling 0.025620 0.008540      2.48     1.05
apoptosis                   0.025620 0.012427      2.05     1.05
cell_cycle_control          0.000865 0.000216      4.70     1.05
notch_delta_signalling      0.131999 0.131999      1.57     1.05
                            Std.dev #Cov
androgen_receptor_signaling   0.409   72
apoptosis                     0.327  187
cell_cycle_control            0.515   31
notch_delta_signalling        0.496   34
```

Allowing a family–wise error rate of 0.05 all but one pathways remain significant for both methods.

5.2 Closed Testing Procedure

Closed testing procedures ([9]) offer a versatile and powerful approach to the multiple testing problem. Implementation is non–trivial, therefore, the program given in this version should be regarded as a prototype.

In order to apply the closed testing procedure we first have to create the required family of hypotheses by building all intersections between the "natural" hypotheses tested above and all intersections of those new hypotheses and so on.

The resulting family of hypotheses can be illustrated in a directed graph. If we just for the sake of illustration assume that we have only four hypotheses named "1", ... "4" then the node "1-2-3-4" for example stands for the global hypothesis that the genes of all four pathways are not differentially expressed.

Now the interesting hypothesis "1" for example can be rejected if also the hypotheses "1-2-3-4", "1-2-3", "1-2-4", "1-3-4", "2-3-4", "1-2", ..., "1-4" are rejected. These relationships are represented by the edges of the graph.

We can compute the closed testing procedure using the function

```
> ga.closed <- GlobalAncova.closed(xx=vantVeer, group=metastases,
+                                  test.genes=pathways[1:4], previous.test=ga.pw,
+                                  level=0.05, method="approx")
```

where *test.genes* is again a list of pathways. In order to shorten computing time we can provide the results of the previous application of `GlobalAncova` for the pathways of interest. The option *level* allows to manipulate the level of significance. There is again the possibility to choose between permutation and asymptotic p-values via the option *method*. Note that if you provide results of previous computation, the type of p-values has to correspond, i.e. if we now want to use *method = "approx"* in the previous test we should have used *method = "approx"* or *method = "both"* such that asymptotic p-values are available.

Also for `GlobalAncova.closed` all three different function calls as for `GlobalAncova` itself are possible.

The function `GlobalAncova.closed` provides the formed null hypotheses (this means lists of genes to be tested simultaneously), the test results for each pathway of interest and the names of significant and non significant pathways. Names for the intersections of hypotheses are built by simply coercing the names of the respective pathways. If for a pathway one single hypothesis can not be rejected there is no need to test all the remaining hypotheses. That is why in test results of non significant pathways lines are filled with NA's after a p–value $> \alpha$ occured. Here only test results for the first pathway are displayed.

```
> names(ga.closed)

[1] "new.data"        "test.results"    "significant"
[4] "not.significant"

> rownames(ga.closed$test.results[[1]])

[1] "androgen_receptor_signaling"
[2] "androgen_receptor_signaling.apoptosis"
[3] "androgen_receptor_signaling.cell_cycle_control"
[4] "androgen_receptor_signaling.notch_delta_signalling"
[5] "apoptosis.androgen_receptor_signaling.cell_cycle_control"
[6] "apoptosis.androgen_receptor_signaling.notch_delta_signalling"
[7] "cell_cycle_control.androgen_receptor_signaling.notch_delta_signalling"
[8] "cell_cycle_control.apoptosis.androgen_receptor_signaling.notch_delta_signalling"

> rownames(ga.closed$test.results[[1]]) <- NULL
> ga.closed$test.results[1]

$androgen_receptor_signaling
     genes  F.value    p.approx
[1,]    72 2.389837 0.003045062
[2,]   258 2.096100 0.000400000
```

```
[3,]   100 3.040900 0.000200000
[4,]   106 2.120700 0.002400000
[5,]   286 2.381000 0.000100000
[6,]   292 2.027300 0.000400000
[7,]   134 2.686400 0.000200000
[8,]   320 2.290200 0.000100000
```

```
> ga.closed$significant
```

```
[1] "androgen_receptor_signaling" "apoptosis"
[3] "cell_cycle_control"
```

```
> ga.closed$not.significant
```

```
[1] "notch_delta_signalling"
```

We get the same significant and non significant pathways as before.

6 Testing Public Gene Set Collections

6.1 Gene Ontology

When testing gene sets defined by the Gene Ontology it is of special interest to incorporate the hierarchical structure of the GO graph. Goeman and Mansmann (2008)[4] developed the *focus level* method, a multiple testing approach on the GO that combines the correction method of Holm (1979) [6] and the closed testing procedure from Marcus et al. (1976) [9] (also used in section 5.2). The method is originally implemented in package *globaltest*. We adapted the corresponding functions such that the procedure is available also with `GlobalAncova`. For details see the vignette of *globaltest*.

For reasons of computing time here we only apply the focus level method the subgraph of the 'cell cycle' GO term and all its descendants. To test all terms within an ontology (or several ontologies) the *id* argument can just be omitted.

```
> library(GO.db)
> descendants <- get("GO:0007049", GOBPOFFSPRING)
> gago <- GAGO(xx=exprs(golubX), formula.full=~ALL.AML, formula.red=~1,
+              model.dat=pData(golubX), id=c("GO:0007049", descendants),
+              annotation="hu6800", ontology="BP", multtest="focuslevel")
```

```
> head(gago)
```

```
                  raw.p   focuslevel                          Term
GO:0007049 2.911765e-16 5.995681e-14                     cell cycle
GO:0010564 5.222799e-15 5.995681e-14 regulation of cell cycle process
GO:0022402 9.157045e-15 5.995681e-14               cell cycle process
GO:0044770 4.617604e-15 5.995681e-14      cell cycle phase transition
GO:0044843 6.992739e-16 5.995681e-14 cell cycle G1/S phase transition
GO:0051726 1.559710e-16 5.995681e-14         regulation of cell cycle
```

All arguments for specifying the linear model used in `GlobalAncova` can be given here. Only the parameter *method* is not available because the focus level procedure does only work with the asymptotic test. Note however, that still a number of permutations can be specified (*perm*, default 10,000) since very large GO terms (with more annotated genes than defined by parameter *max.group.size*, default 2500) are tested permutation-based.

Alternative to the focus level procedure, one can choose between the methods of Holm [6], Benjamini & Hochberg [1] and Benjamini & Yekutieli [2] for multiple testing correction, using the argument *multtest*.

6.2 The Broad Gene Sets

As described in the *globaltest* vignette, also the gene set collection from the Broad Institute can be tested quite easily (c1: positional gene sets, c2: curated gene sets, c3: motif gene sets, c4: computational gene sets, c5: GO gene sets). First the file `msigdb_v.2.5.xml` containing all gene sets has to be downloaded from `http://www.broad.mit.edu/gsea/downloads.jsp#msigdb`. The function `getBroadSets` from package *GSEABase* can then be used to read the xml file into `R`. With the function `GABroad` gene sets can be selected and tested using the gene set IDs

```
> broad <- getBroadSets("your/path/to/msigdb_v.2.5.xml")
> GABroad(xx=exprs(golubX), formula.full=~ALL.AML, formula.red=~1,
+         model.dat=pData(golubX), id="chr5q33", collection=broad,
+         annotation="hu6800")
```

or all gene sets from one or several categories can be tested

```
> GABroad(xx=exprs(golubX), formula.full=~ALL.AML, formula.red=~1,
+         model.dat=pData(golubX), category=c("c1", "c3"), collection=broad,
+         annotation="hu6800", multtest="holm")
```

7 Diagnostic Plots

There are two types of diagnostic plots available supporting communication and interpretation of results of the global ANCOVA. The `Plot.genes` visualize the influence of individual genes on the test result while the `Plot.subjects` visualizes the influence of individual samples. Both functions are based on the decomposition of sums of squares.

We use again the van't Veer data constricted to the genes of the p53–signalling pathway for demonstration of the plot functions.

7.1 Gene Plot

The influence of each gene on the outcome of the test can be assessed and visualized with a diagnostic plot generated by our function `Plot.genes`. It corresponds to the function `features` in the *globaltest* package. We use the `features` function with the option *what = "w"* for displaying weighted gene-wise test statistics, which corresponds best to what is shown in `Plot.genes`. The function `Plot.genes` gives a graphical display of single gene-wise analysis

for all genes. Bars are always positive as a reduction of sum of squares is always achieved in this case. The bar height indicates the influence of the respective gene on the test statistic. The added reference line is the residual mean square error per gene and corresponds to the expected height of the bars under the null hypothesis which says that the gene is not associated with the clinical outcome. The actual and expected bar heights also correspond to the nominator and denominator of gene-wise F-statistics. Hence the ratio of the two values is a measure for the association of the respective gene with the phenotype. Bar heights for all genes can be returned by setting the option *returnValues* to `TRUE`. This helps to detect genes with most influence on the global statistics. Note however that comparisons between different gene groups can not easily be done by means of these values directly since different group sizes have an impact on global significance.

The bars can be colored according to a variable of interest with the option *Colorgroup* in order to show in which of the groups a gene has the highest expression values. The automatically chosen bar labels can be manipulated with the parameter *bar.names*.

The commands for creating gene plots in the *GlobalAncova* and the *globaltest* are as follows. Note that for the former one again three alternatives for function calls are provided, see section 4 for details.

The two approaches show almost the same results (figures 1 and 2). We prefer plotting horizontal bars rather than vertical because we think it is easier to read off the bar heights this way.

```
> Plot.genes(xx=vantVeer, group=metastases, test.genes=p53)
> gt.vV <- gt(metastases, vantVeer, subsets=p53)
> features(gt.vV, what="w")
```

Figure 1: Gene Plot for the van't Veer data with *GlobalAncova*. Shown are the genes of the p53–signalling pathway. The bar height indicates the influence of the respective gene on the test statistic. The color shows in which of the phenotype groups the gene has higher expression values. The reference line is the residual mean square error per gene.

Figure 2: Gene Plot for the van't Veer data with *globaltest*. Shown are the genes of the p53–signalling pathway. The bar height indicates the influence of the respective gene on the test statistic. The position of the fat marks gives the expected height of the bar under the null hypothesis. The other marks indicate with how many standard deviations the bar exceeds this reference.

In this case where only the influence of one variable is of interest (and therefore the easiest version of possible function calls is chosen), the same variable is assumed to be relevant for coloring. However one is free to specify another coloring. For example for the same plot we could ask which genes are higher expressed in samples with either positive or negative Estrogen receptor status, see figure 3.

```
> Plot.genes(xx=vantVeer, formula.full=~metastases, formula.red=~1,
+            model.dat=phenodata, test.genes=p53, Colorgroup="ERstatus")
```

Figure 3: Gene Plot for the van't Veer data with *GlobalAncova*. Shown are the genes of the p53–signalling pathway. The bar height indicates the influence of the respective gene on the test statistic. The color shows in which of the specified phenotype groups, in this case Estrogen receptor status, the gene has higher expression values. The reference line is the residual mean square error per gene.

7.2 Subjects Plot

The function `Plot.subjects` visualizes the influence of the individual samples on the test result and corresponds to the `subjects` of Goeman. As for the `features` plot we use the option *what = "w"* to get closest to the output of `Plot.subjects`. The function `Plot.subjects` gives information on the reduction of sum of squares per subject. Here we sum over genes. Large reduction demonstrates a good approximation of a subject's gene expressions by the corresponding group means. If an individual does not fit into the pattern of its phenotype, negative values can occur. 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. The bars are colored to distinguish samples of different clinical entities that can again be specified by the user through the option *Colorgroup*. With the option *sort* it is also possible to sort the bars with respect to the phenotype groups. Bar labels can be changed with the argument *bar.names*. Also in the subjects plot bar heights can be returned by setting the option *returnValues* to `TRUE`. That may help to detect, not only visually, samples which do not fit into their respective clinical groups.

We compare again the different approaches (figures 4 and 5):

```
> #colnames(exprs(golubX)) <- pData(golubX)[ ,1]
> Plot.subjects(xx=vantVeer, group=metastases, test.genes=p53, legendpos="bottomright")
> subjects(gt.vV, what="w")
```

Figure 4: Subjects Plot for the van't Veer data with *GlobalAncova*. The bar height indicates the influence of the respective sample on the test result. If an individual does not fit into the pattern of its phenotype, negative values can occur. Bars are colored corresponding to phenotype groups.

The function `Plot.subjects` can be invoked by the three alternative function calls (see section 4) and hence also plots corresponding to more complex testing challenges can be produced as well. To give just one example we consider again the influence of the tumour grade, which can take three possible values, on gene expression (figure 6).

```
> Plot.subjects(xx=vantVeer, formula.full=~grade, formula.red=~1,
+               model.dat=phenodata, test.genes=p53, Colorgroup="grade", legendpos="toplef
```

Figure 5: Subjects Plot for the van't Veer data with *globaltest*. The bar height indicates the influence of the respective sample on the test result. If an individual does not fit into the pattern of its phenotype, negative values can occur. The fat marks on the bars show the expected influence of the samples under the null hypothesis. The other marks indicate the standard deviation of the influence of the sample under the null hypothesis.

Figure 6: Subjects Plot for the van't Veer data with *GlobalAncova*. Tumour grade is the clinical variable of interest. The bar height indicates the influence of the respective sample on the test result. If an individual does not fit into the pattern of its phenotype, negative values can occur. Bars are colored corresponding to phenotype groups.

8 Generalized Linear Models

By moving from comparison of residual sums of squares to the comparison of model deviances, the GlobalAncova approach can be generalized to groups of non-quantitative variables. By transformation of individual test statistics to follow the same distribution, the generalized GlobalAncova can even be applied to mixed data types. Significance is assessed by a permutation approach as described in section 4. The generalized GlobalAncova test is performed by function `gGlobalAncova`. We demonstrate its use with a small simulated dataset of binary variables, where we want to test for global differences between two experimental groups (`group`).

```
> data(bindata)
> X <- as.matrix(bindata[,-1])
> gGlobalAncova(X, formula.full=~group, model.dat=bindata, perm=1000)

  p.value Statistic
1       0  194.2137
```

The syntax of the function is similar to `GlobalAncova`. The parameter to specify several sets simultaneously, however, is called *Sets* here (instead of *test.genes*).

```
> gGlobalAncova(X, formula.full=~group, model.dat=bindata,
+               Sets=list(2:5, 6:10, 19:24), perm=1000)

  p.value Statistic
1   0.001  22.80652
2   0.010  17.45448
3   0.001  25.72998
```

In analogy to `Plot.genes`, contributions of individual variables on the global test statistic can be visualized by function `Plot.features`, which shows the variable-wise differences in model deviances. A subjects plot for the generalized GlobalAncova is not yet implemented.

```
> Plot.features(X, formula.full=~group, model.dat=bindata)
```

9 Hierarchical Testing

Hierarchical testing procedures build an alternative to screening variables individually by subsequently testing groups of correlated variables within a hierarchy. For testing those groups global tests like generalized GlobalAncova can be used. We implement the hierarchical testing approach according to Meinshausen [10]. The procedure starts with testing the global null hypothesis that all variables are not associated with the design of interest, and then moves down the given hierarchy testing subclusters of variables. A subcluster is only tested if the null hypothesis corresponding to its ancestor cluster could be rejected. The p-values are adjusted for multiple testing according to cluster size $p^C_{adj} = p^C m/|C|$, where m is the total number of variables and $|C|$ is the number of variables in cluster C. By this, the family-wise error rate is simultaneously controlled over all levels of the hierarchy.

Hierarchical testing can be performed by function `gGlobalAncova.hierarchical`. A hierarchy of variables has to be given as a *dendrogram* object. It can be prespecified by domain knowledge or derived data-driven by clustering.

```
> # get a hierarchy for variables
> dend <- as.dendrogram(hclust(dist(t(X))))
> # hierarchical test
> set.seed(555)
> res <- gGlobalAncova.hierarchical(X, H=dend, formula.full=~group,
+                                   model.dat=bindata, alpha=0.05, perm=1000)

testing global hypothesis...
[1] "global p-value = 0"
hierarchical testing...
[1] "  level 2"
[1] "  level 3"
[1] "  level 4"
[1] "  level 5"
[1] "  level 6"
[1] "  level 9"
[1] "  level 12"
[1] "  level 14"
[1] "  level 17"
[1] "  level 18"
[1] "  level 19"
[1] "  level 21"
[1] "  level 22"
```

The output is an object of class *GAhier*. The most relevant results are the significant end nodes, i.e. the groups or individual variables down to which the hierarchical testing could procede while still getting significant test results.

```
> res

results of hierarchical testing procedure for 24 variables
global alpha = 0.05
```

```
significant end nodes:
           variables n.variables p.value
level22.16     true2           1   0.012
level22.19     true5           1   0.000
level21.12     true9           1   0.000
level21.15     true6           1   0.000
level19.1      true4           1   0.000
level19.3      true8           1   0.000
level18.9      true7           1   0.012
level17.7      true1           1   0.000
level17.8     true10           1   0.012
```

```
> results(res)
```

```
           variables n.variables p.value
level22.16     true2           1   0.012
level22.19     true5           1   0.000
level21.12     true9           1   0.000
level21.15     true6           1   0.000
level19.1      true4           1   0.000
level19.3      true8           1   0.000
level18.9      true7           1   0.012
level17.7      true1           1   0.000
level17.8     true10           1   0.012
```

You can also retrieve just the names of significant end nodes. In case the main interest is in individual variables, option *onlySingleton* can be set to *TRUE*. For the example this does not make a difference, since all significant end nodes are individual variables anyway.

```
> # get names of significant clusters
> sigEndnodes(res)
```

```
level22.16 level22.19 level21.12 level21.15  level19.1  level19.3
   "true2"    "true5"    "true9"    "true6"    "true4"    "true8"
 level18.9  level17.7  level17.8
   "true7"    "true1"   "true10"
```

```
> sigEndnodes(res, onlySingleton=TRUE)
```

```
level22.16 level22.19 level21.12 level21.15  level19.1  level19.3
   "true2"    "true5"    "true9"    "true6"    "true4"    "true8"
 level18.9  level17.7  level17.8
   "true7"    "true1"   "true10"
```

The dendrogram can be visualized with `Plot.hierarchy`, where the paths to significant end nodes are highlighted.

```
> # visualize results
> Plot.hierarchy(res, dend)
```

In order to reduce computational complexity for large hierarchies, a "short cut" is implemented, where the testing procedure is applied separately to K sub-hierarchies containing $m_1, \ldots, m_K$ variables. The p-values resulting within sub-hierarchies are additionally adjusted by $\tau = m/m_k, k = 1, \ldots, K$ such that final p-values are identical to the ones obtained when testing the complete hierarchy

$$p^C_{adj,k} \cdot \tau = p^C \frac{m_k}{|C|} \cdot \frac{m}{m_k} = p^C \frac{m}{|C|} = p^C_{adj}$$

```
> # starting with 3 sub-hierarchies
> set.seed(555)
> res2 <- gGlobalAncova.hierarchical(X, H=dend, K=3, formula.full=~group,
+                                    model.dat=bindata, alpha=0.05, perm=1000)

testing global hypothesis...
[1] "global p-value = 0"
hierarchical testing...
[1] "subtree 1"
[1] "  level 2"
[1] "  level 4"
[1] "  level 5"
[1] "  level 7"
[1] "subtree 2"
[1] "  level 2"
[1] "  level 6"
[1] "subtree 3"
[1] "  level 2"
[1] "  level 4"
[1] "  level 6"
[1] "  level 7"
[1] "  level 8"

> results(res2)

            variables n.variables p.value
K3_level8.5     true9           1   0.000
K3_level8.8     true6           1   0.000
K3_level7.4     true7           1   0.012
K3_level6.2     true1           1   0.000
K3_level6.3    true10           1   0.012
K2_level6.1     true4           1   0.000
K2_level6.3     true8           1   0.000
K1_level7.1     true2           1   0.012
K1_level7.4     true5           1   0.000
```

10 Acknowledgements

We thank the authors of the *globaltest* package, of which we adopted lots of functionalities. We thank Sven Knüppel who took the initiative and contributed a C-code implementation of the permutation test to our package.
This work was supported by the NGFN project 01 GR 0459, BMBF, Germany and BMBF grant 01ZX1309B, Germany.

References

[1] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B*, 57 (1):289–300, 1995.

[2] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple hypothesis testing under dependency. *The Annals of Statistics*, 29 (4):1165–1188, 2001.

[3] J. J. Goeman, F. de Kort, S. A. van de Geer, and J. C. van Houwelingen. A global test for groups of genes: testing association with a clinical outcome. *Bioinformatics*, 20 (1):93–99, 2004.

[4] J.J. Goeman and U. Mansmann. Multiple testing on the directed acyclic graph of gene ontology. *Bioinformatics*, 24 (4), 2008.

[5] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, pages 531–537, 1999.

[6] S. Holm. A simple sequentially rejective multiple test procedure. *Scand. J. Statist.*, 6:65–70, 1979.

[7] M. Hummel, R. Meister, and U. Mansmann. Globalancova: exploration and assessment of gene group effects. *Bioinformatics*, 24 (1):78–85, 2008.

[8] U. Mansmann and R. Meister. Testing differential gene expression in functional groups. *Methods Inf Med*, 44 (3), 2005.

[9] R. Marcus, E. Peritz, and K. R. Gabriel. On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63 (3):655–660, 1976.

[10] Nicolai Meinshausen. Hierarchical testing of variable importance. *Biometrika*, 95(2):265, 2008.

[11] H. Robbins and E.J.G. Pitman. Application of the method of mixtures to quadratic forms in normal variates. *The Annals of Mathematical Statistics*, 20 (4):552–560, 1949.

[12] D.B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley and Sons, Inc, New York, 1987.

[13] J. Schaefer, R. Opgen-Rhein, and K. Strimmer. *corpcor: Efficient Estimation of Covariance and (Partial) Correlation*, 2006. R package version 1.4.4.

[14] L. J. van't Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A.M. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, and S.H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.

11 Session Info

```
> sessionInfo()

R version 4.1.1 (2021-08-10)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 20.04.3 LTS

Matrix products: default
BLAS:   /home/biocbuild/bbs-3.14-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.14-bioc/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_GB              LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] grid      stats4    stats     graphics  grDevices utils
[7] datasets  methods   base

other attached packages:
 [1] GO.db_3.14.0         Rgraphviz_2.38.0     graph_1.72.0
 [4] vsn_3.62.0           hu6800.db_3.13.0     org.Hs.eg.db_3.14.0
 [7] AnnotationDbi_1.56.0 IRanges_2.28.0       S4Vectors_0.32.0
[10] golubEsets_1.35.0    Biobase_2.54.0       BiocGenerics_0.40.0
[13] GlobalAncova_4.12.0  globaltest_5.48.0    survival_3.2-13
[16] corpcor_1.6.10

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.7             lattice_0.20-45
 [3] png_0.1-7              Biostrings_2.62.0
 [5] assertthat_0.2.1       utf8_1.2.2
 [7] R6_2.5.1               GenomeInfoDb_1.30.0
 [9] RSQLite_2.2.8          httr_1.4.2
[11] ggplot2_3.3.5          pillar_1.6.4
[13] zlibbioc_1.40.0        rlang_0.4.12
[15] rstudioapi_0.13        annotate_1.72.0
[17] blob_1.2.2             Matrix_1.3-4
[19] preprocessCore_1.56.0  splines_4.1.1
[21] RCurl_1.98-1.5         bit_4.0.4
[23] munsell_0.5.0          compiler_4.1.1
[25] pkgconfig_2.0.3        tidyselect_1.1.1
[27] KEGGREST_1.34.0        tibble_3.1.5
[29] gridExtra_2.3          GenomeInfoDbData_1.2.7
[31] dendextend_1.15.1      XML_3.99-0.8
[33] fansi_0.5.0            viridisLite_0.4.0
```

```
[35] crayon_1.4.1           dplyr_1.0.7
[37] bitops_1.0-7           affy_1.72.0
[39] xtable_1.8-4           GSEABase_1.56.0
[41] gtable_0.3.0           lifecycle_1.0.1
[43] DBI_1.1.1              magrittr_2.0.1
[45] scales_1.1.1           cachem_1.0.6
[47] XVector_0.34.0         affyio_1.64.0
[49] viridis_0.6.2          limma_3.50.0
[51] ellipsis_0.3.2         vctrs_0.3.8
[53] generics_0.1.1         tools_4.1.1
[55] bit64_4.0.5            glue_1.4.2
[57] purrr_0.3.4            fastmap_1.1.0
[59] colorspace_2.0-2       BiocManager_1.30.16
[61] VGAM_1.1-5             memoise_2.0.0
```