Identifying Interesting Genes with siggenes

by Holger Schwender, Andreas Krause and Katja Ickstadt

A common and important task in microarray experiments is the identification of genes whose expression values differ substantially between groups or conditions. Finding such differentially expressed genes requires methods that can deal with multiple testing problems in which thousands or even tens of thousands of hypotheses are tested simultaneously.

Usually, a statistic appropriate for testing if the expression levels are associated with a covariate of interest and the corresponding p-value are computed for each gene. Afterwards, these raw p-values are adjusted for multiplicity such that a Type I error rate is strongly controlled at a pre-specified level of significance. The classical example of such an error rate is the family-wise error rate (FWER), i.e. the probability of at least one false positive. This error rate, however, might be too conservative for a situation in which thousands of hypotheses are tested and several tens of genes should be identified. In the analysis of microarray data, another error rate has, hence, become very popular: The False Discovery Rate (FDR) which is loosely spoken the expected proportion of false positives among all rejected null hypotheses, i.e. identified genes.

There are, however, other ways to adjust for multiplicity: For example, QQ plots or the Bayesian framework can be employed for this purpose. If the observed test statistics are plotted against the values of the test statistics that would be expected under the null hypothesis most of the points will approximately lie on the diagonal. Those points that differ substantially from this line correspond to genes that are most likely differentially expressed. The Significance Analysis of Microarrays (SAM) proposed by Tusher et al. (2001) can be used to specify what “differ substantially" means. While Tusher et al. (2001) base their analysis on a moderated t statistic, Schwender et al. (2003) compare this approach with a SAM version based on Wilcoxon rank sums.

Efron et al. (2001) use an empirical Bayes analysis (EBAM) to model the distribution of the observed test statistics as a mixture of two components, one for the differentially expressed genes, and the other for the not differentially expressed genes. Following their analysis, a gene is called differentially expressed if the corresponding posterior probability is larger than 0.9.

Both SAM and EBAM are implemented in the Bioconductor package **siggenes**. In this article, we, however, will concentrate on SAM. In the following, we briefly describe the SAM procedure, its implementation in **siggenes** (for more details, see Schwender et al. (2003)), and the test statistics already available in this package. Afterwards, we show how you can write your own function for other testing situations. Finally, we will give an example of how `sam` can be applied to gene expression data.

Significance Analysis of Microarrays

In SAM, a statistic d appropriate for testing if there is an association between the expression levels and the covariate of interest is computed for each of the m genes. These observed test scores are sorted and plotted against the scores expected under the null hypothesis, where the expected test scores $\bar{d}_{(i)}, i = 1, \ldots, m$, are computed as follows: If the null distribution is known, then $\bar{d}_{(i)}$ is the $(i - 0.5)/m$ quantile of this null distribution. Otherwise, $\bar{d}_{(i)}$ is assessed by

– generating B permutations of the group labels,

– computing the m test statistics and sorting them for each of the B permutations,

– averaging over the B i^{th}-smallest scores,

where typically $B = 100$ permutations are randomly chosen (Tibshirani et al. , 2002), or if the total number of possible permutations is less than B, full permutation is performed.

Two lines parallel to the diagonal in a distance of Δ are then drawn into this plot called SAM plot (see Figure 1). Any gene showing a d value

– larger than or equal to the d value of the gene, say d_{up}, that corresponds to the left-most point on the right side of the origin that lies above the upper Δ line,

– smaller than or equal to the d value of the gene, say d_{low}, that corresponds to the right-most point on the left side of the origin that lies below the lower Δ line

is called differentially expressed. Afterwards, the FDR is estimated by

– counting how many of the mB permuted test scores are larger than or equal to d_{up} or smaller than and equal to d_{low}, and dividing this number by B,

– dividing this average by the number of identified genes,

- multiplying this ratio by the prior probability that a gene is not differentially expressed (by default, `sam` estimates this probability by the procedure of Storey and Tibshirani (2003)).

This procedure is repeated for several values of Δ, and the value of Δ is chosen that provides the best balance between the number of identified genes and the estimated FDR, i.e. that allows to simultaneously attain the two competing goals "As many genes as possible" and "As low FDR as possible" as well as possible.

The following test statistics can be called in `sam` by setting the argument '`method`' to

`d.stat`: Moderated t and F statistics. The "usual" t or F statistics are computed if the fudge factor '`s0`' is set to zero (for details, see Tusher et al. (2001)).

`wilc.stat`: Wilcoxon rank sums for one and two class analyses.

`cat.stat`: Pearson's χ^2-statistic for testing categorical data such as SNP (Single Nucleotide Polymorphism) data (Schwender, 2005).

Writing Your Own Test Score Function

It is also possible to write your own function for another testing situation and use this function in `sam`. This function must have as input the two required arguments

'`data`': A matrix or data frame containing the data. Each row of this data set should correspond to one of the m variables, i.e. genes, and each column to one of the n observations/samples.

'`cl`': A vector consisting of the class labels of the observations.

The function can also have additional optional arguments that can be called in `sam`.

The output of this function must be a list consisting of the following objects

'`d`': A numeric vector containing the test scores of the genes.

'`d.bar`': A numeric vector of length `na.exclude(d)` consisting of the sorted test scores expected under the null hypothesis.

'`p.value`': A numeric vector of the same length and order as '`d`' containing the p-values of the genes.

'`vec.false`': A numeric vector of the same length as '`d`' comprising the under the null hypothesis expected number of genes that are either larger than d_i (if $d_i \geq 0$), or smaller than d_i (if $d_i < 0$) for each gene i, $i = 1, \ldots, m$ (for more details, see Schwender et al. (2003)).

'`s`': A numeric vector containing the standard errors of the expression values.

'`s0`': A numeric value specifying the fudge factor.

'`mat.samp`': A $B \times n$ matrix containing the permuted class labels.

'`msg`': A character vector containing messages that are displayed when the `SAM` specific `S4` methods `print` and `summary` are called.

'`fold`': A numeric vector containing the fold changes of the genes. Should be set to `numeric(0)` if another analysis than a two-class analysis is performed.

Assume, e.g., that we would like to perform a SAM analysis with the "usual" t-statistic assuming equal group variances and normality. The code of a function `t.stat` for such an analysis is given by

```
t.stat <- function(data, cl){
    require(genefilter) ||
        stop("genefilter required.")
    row.out <- rowttests(data, cl)
    d <- row.out$statistic
    m <- length(na.exclude(d))
    d.bar <- qt(((1:m) - 0.5)/m, row.out$df)
    p.value <- row.out$p.value
    vec.false <- m * p.value/2
    s <- row.out$dm/d
    # dm: differences in group means
    msg <- paste("SAM Two-Class Analysis",
         "Assuming Normality\n\n")
    list(d=-d, d.bar=d.bar, p.value=p.value,
        vec.false=vec.false, s=s, s0=0,
        mat.samp=matrix(numeric(0)),
        msg=msg, fold=numeric(0))
}
```

Please note that in the output of `t.stat` '`d`' is set to `-d` since in `rowttests` the mean of group 2 is subtracted from the mean of group 1, whereas in `sam` the difference is taken the other way around.

Now `t.stat` can be used in `sam` by setting `method=t.stat`.

Example: ALL Data

As example we here employ one of the data sets used in Gentleman et al. (2005). The package **ALL** containing this data set can be downloaded by

```
> source("http://www.bioconductor.org/getBioC.R")
> getBioC("ALL")
> library(ALL)
> data(ALL)
```

As described in Scholtens and von Heydebreck (2005), we filter the genes showing low expression values or an IQR of less than 0.5, and select a subset of the samples.

```
> library(genefilter)
> subALL <- filterALL()
```

(The code of `filterALL` can be found in the Appendix 1.) This leads to an `exprSet` object containing gene expression data of 2,391 probe sets and 79 samples.

As Scholtens and von Heydebreck (2005), we would like to identify the probe sets whose expression values differ strongly between the samples for which

```
> mol.biol <- pData(subALL)$mol.biol
```

is equal to `"BCR/ABL"` and the samples for which `mol.biol=="NEG"`. Thus, `sam` is applied to this data set by specifying the required arguments '`data`' and '`cl`', where

'`data`' can either be a matrix, a data frame, or an `exprSet` or `ExpressionSet` object containing the gene expression data,

'`cl`' is a vector containing the class labels of the samples. If '`data`' is an `exprSet` object, then '`cl`' can also be a character string naming the column of `pData(data)` that contains the class labels.

So

```
> library(siggenes)
> clALL <- ifelse(mol.biol=="BCR/ABL", 0, 1)
> dataALL <- exprs(subALL)
> out1 <- sam(dataALL, clALL,
+  var.equal = TRUE, rand = 123456)
```

leads to the same results as

```
> out2 <- sam(subALL, "mol.biol",
+  var.equal = TRUE, rand = 123456)
```

where '`var.equal`' is set to `TRUE` since we here would like to assume that the group variances are equal, and '`rand`' is set to `123456` to make the results of this analysis reproducible.

By default, the number of identified genes and the estimated FDR is computed for ten values of Δ equidistantly spaced between 0.1 and $\max_i |d_{(i)} - \bar{d}_{(i)}|$. The output of our SAM analysis is thus given by

```
> out1
```

```
SAM Analysis for the Two-Class Unpaired Case
Assuming Equal Variances

   Delta   p0   False Called     FDR
1    0.1 0.63 1900.61    2075 0.57726
2    0.7 0.63  125.07     400 0.19706
3    1.4 0.63    3.86      90 0.02703
4    2.0 0.63     0.1      25 0.00252
5    2.7 0.63       0       6       0
6    3.3 0.63       0       4       0
7    4.0 0.63       0       3       0
8    4.6 0.63       0       2       0
9    5.3 0.63       0       2       0
10   5.9 0.63       0       1       0
```

where `p0` is the estimated prior probability that a gene is not differentially expressed, `False` is the number of falsely called genes (see Tusher et al. (2001)), `Called` is the number of identified genes, and `FDR` = `p0` * `False` / `Called` is the estimated FDR.

Please note that the number of falsely called genes, i.e. the under the null hypothesis expected number of genes having a test score larger than d_{up} or smaller than d_{low}, is not an estimate for the actual number of false positives.

More information, e.g., the value of the fudge factor can be obtained using `summary`. Both `summary` and `print` can also be used to generate the above table for other values of Δ. For example,

```
> print(out1, seq(1.4, 2, 0.1))
```

```
SAM Analysis for the Two-Class Unpaired Case
Assuming Equal Variances

  Delta   p0 False Called     FDR
1   1.4 0.63  3.86     90 0.02703
2   1.5 0.63  2.36     77 0.01932
3   1.6 0.63  1.45     64 0.01428
4   1.7 0.63  0.80     54 0.00934
5   1.8 0.63  0.41     45 0.00574
6   1.9 0.63  0.26     36 0.00455
7   2.0 0.63  0.10     25 0.00252
```

Let's say our goal is to identify about 100 genes and to control the FDR at a level of about 1.5%. In this case, $\Delta = 1.5$ would be a reasonable choice since little less than 100 genes are detected with an estimated FDR slightly larger than 1.5%. The SAM plot for this selection shown in Figure 1 is generated by

```
> plot(out1, 1.5, sig.col = c(3,2), pch = 16,
+  pos.stats = 2, cex = 0.6)
```

where

'`sig.col`' is a numeric value or vector specifying the color of the identified down- and up-regulated genes,

'`pos.stats`' indicates where the statistics are shown in the SAM plot,

'`cex`' specifies the relative size of the plotting symbols of the genes not identified as differentially expressed.

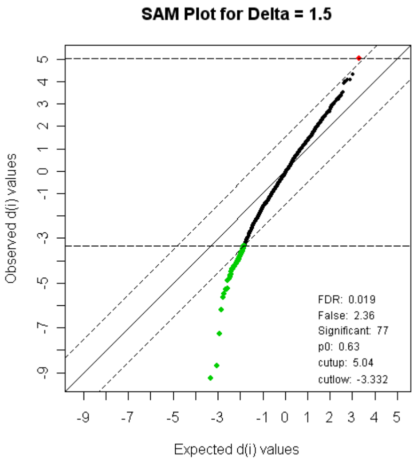

Figure 1: SAM Plot for $\Delta = 1.5$.

While the relative size of the symbols can be specified separately for the identified and the not identified genes, the symbol itself (`pch`) is the same for both types of genes. For all arguments of the `SAM` specific method `plot`, see

```
> help.sam(plot)
```

Information about the identified genes such as their *d* values, the corresponding raw p-values and the q-values (see Storey and Tibshirani (2003)) can be obtained by

```
> summary(out1,1.5)
```

An excerpt from the output of `summary` is shown in Figure 2. This information can also be stored in a csv file via `sam2excel`, or in an html file using `sam2html`. If '`data`' is an `exprSet` object or '`chipname`' is specified in `sam2html`, then the html file will also contain the gene symbols and links to public repositories such as Entrez, RefSeq and UniGene. If '`cdfname`' is specified, additionally, links to the Affymetrix webpages of the identified probe sets will be available. For example, the html file generated by

```
> sam2html(out1, 1.5, "out1.html", ll = TRUE,
+   cdfname = "HG-U95Av2")
```

is available at `http://www.statistik.uni-dortmund.de/de/content/einrichtungen/lehrstuehle/personen/holgers/out1.html`.

Finally, we would like to check if `method="t.stat"` (see previous section 1) works correctly.

```
> out3 <- sam(subALL, "mol.biol",
+   method = "t.stat")
> out3

SAM Two-Class Analysis Assuming Normality

   Delta   p0    False Called      FDR
1    0.1 0.63 1832.197   2050  0.56307
2    0.7 0.63   94.609    347  0.17177
3    1.3 0.63    3.555    100  0.02240
4    1.9 0.63    0.044     19  0.00145
5    2.5 0.63 0.000567      6 5.95e-05
6    3.1 0.63 3.33e-05      4 5.24e-06
7    3.7 0.63 2.92e-07      3 6.14e-08
8    4.4 0.63 5.73e-10      2 1.80e-10
9    5.0 0.63 5.73e-10      2 1.80e-10
10   5.6 0.63        0      0        0
```

Since in both analyses we have computed the *t* statistic assuming equal group variances, the *d* values in both analyses should be the same:

```
> tmp <- sum(round(out1@d, 8) ==
+   round(out3@d, 8))
> tmp == length(out1@d)
[1] TRUE
```

Summary

The package **siggenes** contains functions for performing both a Significance Analysis of Microarrays (SAM) and an Empirical Bayes Analysis of Microarrays (EBAM). The function `sam` provides not only a set of statistics for standard tests such as *t* and *F* test but also the possibility to employ user-written functions for other testing situations. After identifying a list of genes, not only statistics of these genes, such as their test scores and p-values, can be obtained but also links to public repositories containing biological information about these genes.

The EBAM functions are currently under revision to provide more user-friendly and less memory-consuming versions of these functions having all the features that the SAM functions already have.

Acknowledgements

The work of Katja Ickstadt and Holger Schwender has been supported by the Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 475. The work of Andreas Krause was largely carried out while he was an employee of Novartis Pharma AG, Basel, Switzerland.

Appendix

```
filterALL <- function(){
    pdat <- pData(ALL)
    subset<-intersect(grep("^B",
        as.character(pdat$BT)),
        which(pdat$mol %in% c("BCR/ABL",
        "NEG")))
    eset <- ALL[, subset]
    require(genefilter)
    f1 <- pOverA(0.25, log2(100))
```

```
SAM Analysis for the Two-Class Unpaired Case Assuming Equal Variances

 s0 = 0
 Number of permutations: 100
 MEAN number of falsely called genes is computed.

 Delta: 1.5
 cutlow: -3.332
 cutup: 5.04
 p0: 0.63
 Significant Genes: 77
 Falsely Called Genes: 2.36
 FDR: 0.0193


Genes called significant (using Delta = 1.5):

    Row d.value  stdev  rawp q.value R.fold       Name
1   134   -9.26 0.1188     0       0  0.457  1636_g_at
2  1787   -8.69 0.1327     0       0  0.442   39730_at
3   133   -7.28 0.1652     0       0  0.420    1635_at
4  1890   -6.18 0.2878     0       0  0.429   40202_at
5  1193   -5.65 0.2388     0       0  0.460   37027_at
```

Figure 2: An excerpt from the output of summary(out1, 1.5).

```
    f2 <- function(x) IQR(x) > 0.5
    selected <- genefilter(eset,
        filterfun(f1, f2))
    esetSub <- eset[selected, ]
    pdat <- pData(esetSub)
    esetSub$mol.biol <-
        as.character(esetSub$mol.biol)
    esetSub
}
```

Bibliography

B. Efron, R. Tibshirani, J. Storey and V. Tusher. Empirical Bayes Analysis of a Microarray Experiment. *Journal of the American Statistical Association*, 96, 1151–1160, 2001.

R. Gentleman, V. J. Carey, W. Huber, R. A. Irizarry and S. Dudoit, editors. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer, New York, 2005.

D. Scholtens and A. von Heydebreck. Analysis of Differential Gene Expression Studies. In: R. Gentleman, V. J. Carey, W. Huber, R. A. Irizarry, S. Dudoit, editors. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer, New York, 229–248, 2005.

H. Schwender. Modifying Microarray Analysis Methods for Categorical Data – SAM and PAM for SNPs. In: C. Weihs, W. Gaul, editors. *Classification – The Ubiquitous Challenge*. Springer, Heidelberg, 370–377, 2005.

H. Schwender, A. Krause and K. Ickstadt. Comparison of the Empirical Bayes and the Significance Analysis of Microarrays. *Technical Report*, SFB 475, University of Dortmund, Germany, 2003. URL http://www.sfb475.uni-dortmund.de/berichte/tr44-03.pdf.

J. D. Storey and R. Tibshirani. Statistical Significance for Genomewide Studies. *Proceedings of the National Academy of Sciences*, 100, 9440–9445, 2003.

R. Tibshirani, T. Hastie, B. Narsimhan, M. Eisen, G. Sherlock, P. Brown and D. Botstein. Exploratory Screening of Genes and Clusters from Microarray Experiments. *Statistica Sinica*, 12, 47–59, 2002.

V. Tusher, R. Tibshirani and G. Chu. Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *Proceedings of the National Academy of Science*, 98, 5116–5121, 2001.

Holger Schwender, Katja Ickstadt
Department of Statistics, SFB 475
University of Dortmund, Germany
holger.schwender@udo.edu,
ickstadt@statistik.uni-dortmund.de

Andreas Krause
Pharsight Corporation
Mountain View, CA, USA
akrause@pharsight.com