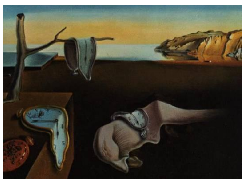

Journal of Statistical Software

November 2012, Volume 51, Issue 10. http://www.jstatsoft.org/

Meta-Statistics for Variable Selection: The R Package BioMark

Ron Wehrens
Fondazione Edmund Mach

Pietro Franceschi
Fondazione Edmund Mach

Abstract

Biomarker identification is an ever more important topic in the life sciences. With the advent of measurement methodologies based on microarrays and mass spectrometry, thousands of variables are routinely being measured on complex biological samples. Often, the question is what makes two groups of samples different. Classical hypothesis testing suffers from the multiple testing problem; however, correcting for this often leads to a lack of power. In addition, choosing α cutoff levels remains somewhat arbitrary. Also in a regression context, a model depending on few but relevant variables will be more accurate and precise, and easier to interpret biologically.

We propose an R package, **BioMark**, implementing two meta-statistics for variable selection. The first, higher criticism, presents a data-dependent selection threshold for significance, instead of a cookbook value of $\alpha = 0.05$. It is applicable in all cases where two groups are compared. The second, stability selection, is more general, and can also be applied in a regression context. This approach uses repeated subsampling of the data in order to assess the variability of the model coefficients and selects those that remain consistently important. It is shown using experimental spike-in data from the field of metabolomics that both approaches work well with real data. **BioMark** also contains functionality for simulating data with specific characteristics for algorithm development and testing.

Keywords: biomarkers, higher criticism, stability selection, spike-in data, metabolomics.

1. Introduction

In the life sciences, comparing groups of individuals to find significant differences is more and more important. The classical example is comparing patients and healthy controls using microarray data, in order to find which genes are related to the illness under study. Similar applications can be listed for many other fields, such as food science and agriculture. Apart

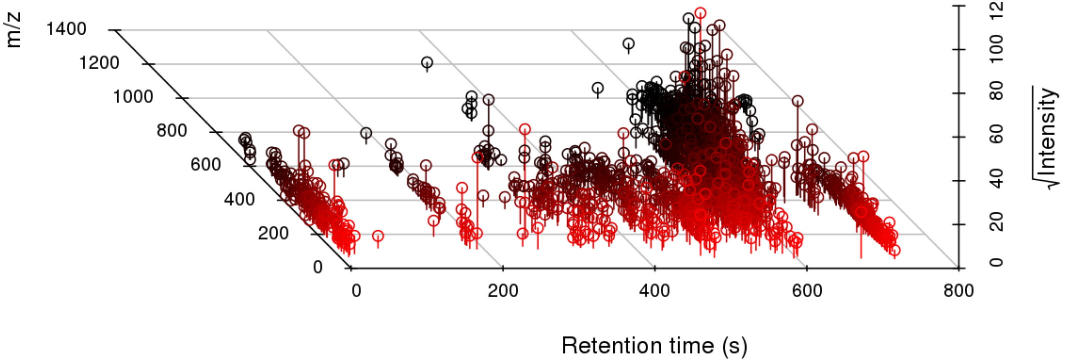

Figure 1: Visualization of apple LC-MS data: the first control sample in the apple data set, measured in positive ionization mode. In total, 1632 features have been identified. Each feature is an intensity at a specific location in the retention time - m/z plane. To avoid high-intensity features to dominate the plot completely, square roots of the intensities are plotted here.

from microarrays, many other types of data can be used, such as next generation sequencing data, measuring the abundance of RNA (and hence the expression of a gene) in the sample, and proteomics and metabolomics data, estimating levels of proteins and metabolites, respectively. In all cases the measurements are relatively complicated but present a wealth of information for each sample. In metabolomics and proteomics the number of variables can range from several hundreds to thousands, and in the case of gene expression measurements this number is one or even two orders of magnitude larger. In these cases, variables that are significantly different between the groups (in whatever way this is defined), are typically indicated with the word “biomarkers”. The same term is used in a regression context for those variables that are essential for the quality of the model – a more simple model, considering only these biomarker variables, often has the same or even improved predictive performance and is much easier to interpret biologically.

As an example from metabolomics, Figure 1 shows the result of one liquid chromatography-mass spectrometric (LC-MS) measurement of an apple sample. Every vertical line segment corresponds to a feature, defined by a retention time, a so-called m/z value (the weight-to-charge ratio of a fragment of a molecule) and an intensity. Measurements like this provide a fingerprint of the biological system under study, summarizing the abundances of all small chemical molecules active in the metabolism. The example shown in Figure 1 is one sample from a larger data set in which concentration differences have been introduced experimentally for specific compounds, and which can serve as a testbed for algorithm development, included in the **BioMark** package. We will come back to these data later.

Standard solutions for finding significant differences like the t test are often difficult to apply in practice: because of the very unfavourable sample-to-variable ratios in omics fields their power is low (see, e.g., Franceschi, Vrhovsek, Mattivi, and Wehrens 2012b), and what is left is often decreased further by multiple-testing corrections. In addition, the choice of the confidence level α is subjective in some cases this choice is used to tweak the outcome to obtain manageable numbers of “significant” differences. Popular multivariate alternatives such as partial least-squares discriminant analysis (PLSDA, Barker and Rayens 2003) require a substantial amount of tuning, which may be difficult for data with relatively few samples. Explicit variable selection methods, aiming at minimizing prediction error with only a subset of the variables, have a tendency to overtrain, which especially in cases with relatively few samples is hard to detect. In short, all of these methods – henceforth indicated as "primary" biomarker selection methods – leave something to be desired.

In recent years, examples of second-level significance testing have been published to address these issues. Here, we describe package **BioMark** for the R language (R Development Core Team 2012), implementing two of these. The first is an approach called higher criticism (Donoho and Jin 2004, 2008) for selecting optimal α cutoff levels in two-class discrimination problems; the second is stability selection (Meinshausen and Bühlmann 2010; Wehrens, Franceschi, Vrhovsek, and Mattivi 2011), for identifying variables that are consistently important upon perturbation of the data. Both have been shown to work well for real data. Other approaches that could be termed second-level selection methods are multiple-testing corrections such as the false discovery rate (Benjamini and Hochberg 1995) and the q value (Storey 2002; Storey and Tibshirani 2003), and methods focusing on differences of predefined sets of variables, rather than individual variables – these are sometimes also referred to as Enrichment Analysis methods (see, e.g., Subramanian *et al.* 2005; Efron and Tibshirani 2007).

The paper is structured as follows. First we give some background on both higher criticism and stability selection. Next, we describe the structure of the **BioMark** package, including the experimental spike-in apple data, and possibilities for data simulation for algorithm testing. Some examples will illustrate the potential of the implemented methods. The paper ends with a short history of the package and a look forward.

2. Theory

2.1. Higher criticism

Higher criticism (HC) thresholding estimates the number of non-null hypotheses by comparing p values with their expected uniform distribution under the null hypothesis (Donoho and Jin 2004, 2008). It assumes that differences are rare (there are not many) and weak (each individual difference between the groups is hard to detect). The method has only one parameter, the maximum fraction of variables to be considered as possible biomarkers. By default, it is set to 0.1, but in most cases the results are insensitive to the exact setting. The test statistic, which can be described as the "z score of the p value" (Donoho and Jin 2008), is given by

$$HC_i = \frac{\sqrt{N}(i/N - p_i)}{\sqrt{i/N(1 - i/N)}} \tag{1}$$

where p_i is the ith of the ordered p values, and N is the total number of variables. The HC threshold is defined as the position of the maximum deviation within the indicated fraction. The result can be visualized in a graph, such as the one shown in Figure 2. Normally, the maximum in the graph is reached well below the maximal number corresponding to ten percent of the data, which explains why the one parameter to be set in the HC approach is usually not very influential.

Note that Equation 1 can be used regardless of the origin of the p values: any statistical test that under the null hypothesis leads to uniformly distributed p values can be used. This

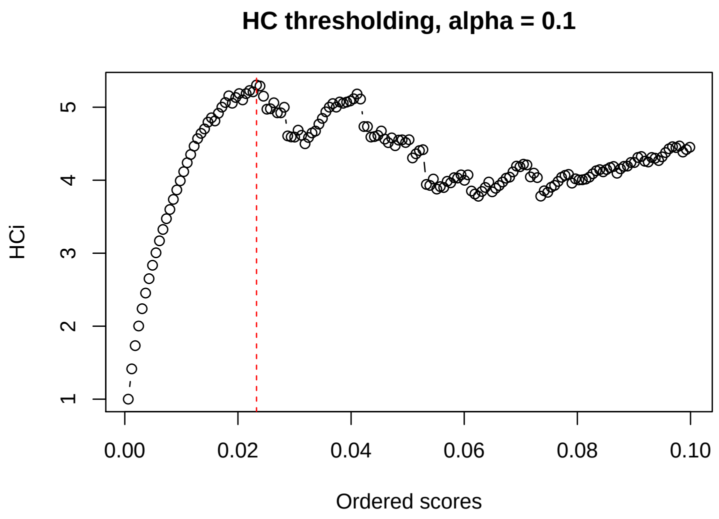

Figure 2: HC biomarker selection for the spiked apple data, positive ionization mode, Group 1 versus Controls. The x-axis shows the ordered measures of variable importance (here: p values derived from t statistics), and the y-axis shows the HC statistic. The maximum is indicated with a dashed line, and gives the number of variables to select (here: 38 out of 1632, corresponding to $\alpha \approx 0.023$).

does imply, however, that it is not possible to use p values after multiple testing correction. In cases where no p values can be obtained from theory, such as model coefficients in many multivariate regression methods, simulation can be used. In particular, one can show that simply permuting the class labels leads to null distributions from which p values can be derived, which then can be used in Equation 1 (Wehrens and Franceschi 2012). This extends the use of HC thresholding to multivariate methods like partial least squares (PLS) and the variable importance of projection (VIP) statistics, which are popular tools in some of the omics fields (Wold, Sjöström, and Eriksson 2001).

2.2. Stability selection

The central idea of stability selection is that *real* differences should be present consistently, and therefore should be found even under perturbation of the data by subsampling or bootstrapping. Two related flavours of stability selection exist. The first is based on the implicit variable selection by methods like the lasso (Meinshausen and Bühlmann 2010); these will automatically lead to a sparse model coefficient vector, in which only very few coefficients have non-zero values. Upon subsampling of the data, some coefficients will remain non-zero, others will sometimes be zero and in other cases be non-zero, and the majority will always be zero. The most convincing differences are in those variables that consistently show non-zero coefficients. The second approach, also applicable for non-sparse coefficient vectors, simply takes the top fraction, e.g., the top ten or the top ten percent, as putative biomarkers (Wehrens *et al.* 2011). Again, variables included in the top fraction by chance are expected to drop out when averaging over many perturbations.

The exact form of the perturbations is not essential, as long as the perturbation is "large enough". We have found that, e.g., leave-one-out crossvalidation is not appropriate: essentially the same model is found in every iteration, even when using random data. The strategy of Meinshausen and Bühlmann (2010) is leaving out half of the samples; a further injection of randomness is obtained by giving random weights to the variables. Wehrens *et al.* (2011) use an approach inspired by random forests (Breiman 2001): apart from leaving out a sizeable fraction of the samples (in their case 30%) they also leave out half of the variables in each iteration. This is similar to the approach by Meinshausen and Bühlmann (2010) if only binary weights are used. Both approaches again lead to very similar results, and testify of the robustness of the stability selection approach. Likewise, results using bootstrapping rather than subsampling to generate the perturbations will probably differ very little from current results.

3. The BioMark package

3.1. Test data

For algorithm development it is of essential importance to use data for which the real differences are known. Some experimental data sets have been described in the area of transcriptomics, where particular transcripts are spiked in the treatment samples – a biomarker selection method should be able to distinguish between the spiked-in and the unchanged variables. A well known example is the "Golden Spike" experiment of Choe, Boutros, Michelson, Church, and Halfon (2005); a subset of these data is available in the **st** package (Opgen-Rhein, Zuber, and Strimmer 2012) as `choe2.mat`. A common characteristic of spike-in data from microarrays is the extremely low number of replicates, making biomarker identification very difficult indeed. An example from the field of proteomics can be found in Wessels *et al.* (2012).

Package **BioMark** comes with its own set of experimental spike-in data. The data are described extensively in Franceschi, Masuero, Vrhovsek, Mattivi, and Wehrens (2012a). In short, extracts of twenty Golden Delicious apples were prepared; ten of them were designated "control" samples and analyzed by LC-MS without further manipulation, the other ten were spiked in three different ways, leading to thirty "treatment" samples. Nine chemical compounds were used for spiking, and the differences between the three sets of spiked samples are the different concentration ratios between the spike-in compounds. Concentrations were chosen to be 20% to 100% higher than the average concentration for those compounds that occur naturally in apples; for the two compounds that are not, concentrations were chosen in such a way that the feature intensities would be comparable to the other features. The forty samples were injected in both positive and negative ion mode, leading to, in total, eighty raw data files. Data processing with the Bioconductor package XCMS (Smith, Want, O'Maille, Abagyan, and Siuzdak 2006) then leads to 1632 and 995 features detected in positive and negative ion modes, respectively. The results are stored in data files `SpikePos` and `SpikeNeg`, respectively. A smaller subset of the data, described in Wehrens *et al.* (2011) and available as data file `spikedApples`, only considers the first set of spike-in concentrations. For further information on the experimental data, see Franceschi *et al.* (2012a).

As an example of how to access the experimental spike-in data, here is the code used to produce the plot in Figure 1 using the **scatterplot3d** (Ligges and Mächler 2003) package:

```
R> library("scatterplot3d")
R> library("BioMark")
R> data("SpikePos")
R> scatterplot3d(SpikePos$annotation$rt, SpikePos$annotation$mz,
+    sqrt(SpikePos$data[1,]), mar = c(4, 3, 0, 3) + 0.1, type = "h",
+    highlight.3d = TRUE, angle = 110, box = FALSE,
+    xlab = "Retention time (s)", ylab = "m/z", zlab = "sqrt(Intensity)")
```

The manual pages for `SpikePos`, `SpikeNeg` and `spikedApples` show more visualization examples.

An often-used alternative to spike-in data is to rely on simulations. For the case of two-group classification, **BioMark** contains the functions `gen.data` and `gen.data2`. Both generate a number of data sets containing two matrices, a control matrix and a treatment matrix with differences present in the first variables. Elements such as the sizes of the matrices, the numbers of variables in which real differences are introduced, and the size of the differences can be controlled by the user. As an example, consider the following:

```
R> simdata <- gen.data(ncontrol = 10, nvar = 800, group.diff = 1.5)
R> names(simdata)

[1] "X"            "Y"            "nbiomarkers"

R> dim(simdata$X)

[1]  20 800 100
```

This piece of code simulates 100 data sets (the default) of ten control samples and ten treated samples, with 800 variables. The default number of biomarkers is five. We can assess the effect of introducing differences in these variables as follows:

```
R> coldiffs <- sapply(1:100, function(i, x)
+    colMeans(x[11:20, , i]) - colMeans(x[1:10, , i]), simdata$X)
R> rowMeans(coldiffs)[1:10]

 [1]  1.47199  1.46784  1.55777  1.50566  1.54334
 [6] -0.01100 -0.02041 -0.03145 -0.01291  0.00426
```

The differences between control and treatment samples (we are looking only at the first ten here), averaged out over all simulated data sets, correspond to what we asked them to be. The default is to generate data with a diagonal covariance matrix. To obtain data with a specific covariance structure (as done in, e.g., Zuber and Strimmer 2009; Meinshausen and Bühlmann 2010; Wehrens *et al.* 2011), one can specify the `cormat` argument:

```
R> mycov <- matrix(0, 800, 800)
R> mycov[row(mycov) <= 50 & col(mycov) <= 50] <- 0.3
R> mycov[row(mycov) > 50 & col(mycov) > 50] <- 0.7
R> diag(mycov) <- 1
R> simdata2 <- gen.data(ncontrol = 10, nvar = 800, group.diff = 1.5,
+    cormat = mycov)
```

This example generates data with a blocked covariance (or correlation) structure as in Wehrens *et al.* (2011): the first fifty variables, in which the biomarkers are present, have a low correlation of 0.3, and the last 750 variables have a much higher correlation of 0.7.

Another possibility is to generate data with the covariance of the experimental control data:

```
R> simdata3 <- gen.data(ncontrol = 10, nvar = ncol(SpikePos$data),
+    group.diff = 1.5, cormat = cov(SpikePos$data[1:10, ]))
```

Such a simulation would lead to data that are more close to the experimental world. However, even when using an experimental covariance matrix, the multivariate normality assumption may be quite far from reality. To generate even more realistic data, we can simply take experimental data and multiply the designated biomarker columns with a number different from one, or add a constant difference, as is done, e.g., in Meinshausen and Bühlmann (2010) and Wehrens and Franceschi (2012). This approach is implemented in function `gen.data2`. The output has the same fields as `gen.data`, but the arguments are slightly different – an example will be shown in Section 4. These functions make it easy to assess the performance of variable selection methods over a large number of scenarios.

3.2. Package structure

Both stability selection and higher criticism are meta-statistics, further processing the results of other statistical tests or modelling methods. Several default values have been defined in the `.biom.options` object, accessible through the `biom.options()` function. To see which primary biomarker selection methods are supported, e.g., use:

```
R> biom.options()$fmethods

[1] "studentt" "shrinkt"  "pcr"     "pls"     "vip"      "lasso"
```

The first two possibilities are univariate and refer to the classical t test and the shrinkage t proposed in Zuber and Strimmer (2009), respectively. The others are multivariate methods, possibly able to benefit from explicitly modelling correlation structure. Options `pcr` and `pls` are based on principal component regression (PCR), and PLS regression, respectively – in these cases, the sizes of the regression coefficients will determine whether variables are likely to correspond to true differences or not. These functions are taken from the **pls** package (Mevik and Wehrens 2007). The next option is the variable importance of projection (VIP, Wold *et al.* 2001), basically a weighted sum of the loadings of a PLS. Again, larger values indicate higher variable importance. Finally, the `lasso` option provides access to the lasso and elastic net functionality of R package **glmnet** (Friedman, Hastie, and Tibshirani 2010), which implements not only the lasso but also the elastic net.

An overview of all possible combinations of methods and applications can be found in Table 1. The HC criterion only makes sense in a two-class discrimination setting, and can be used with all choices of `fmethod` except the lasso. Stability selection can be used for all forms of two-class discrimination, and with all multivariate methods in the case of regression.

The primary function of **BioMark** is `get.biom`, which takes a matrix of measurements and a response vector as the first two arguments. An example, using the spike-in apple data, is given by:

fmethod	Two-class discrimination	Regression
studentt	HC, stability	–
shrinkt	HC, stability	–
pcr	HC, stability	stability
pls	HC, stability	stability
vip	HC, stability	stability
lasso	stability	stability

Table 1: Variable selection methods and meta-statistics implemented in **BioMark**.

```
R> grp1.HC <- get.biom(X = SpikePos$data[1:20,], Y = SpikePos$class[1:20],
+    fmethod = c("studentt", "pls", "vip"), type = "HC")
```

For classification problems, the response vector `Y` is a factor; providing a numeric vector will be interpreted as pointing to a regression situation. The third argument indicates which modelling methods should be considered; by default this takes the value `"all"`. In this example, we only consider the Student t test. The fourth argument tells the function what statistic should be returned. It can take one of three values: `"coef"` simply returns the calculated statistics, i.e., t values, regression coefficients or VIP values. A value of `"HC"` leads to the calculation of those variables that are important according to the higher criticism approach. Likewise, defining `type = "stab"` leads to stability selection. Further arguments can be used to determine the type of scaling (default is standardization to zero mean and unit variance) and the number of components to use in the multivariate methods. Function `get.biom` returns an object of class `BMark`, for which generic functions `coef`, `summary` and `print` are available. The summary function, for example, shows how many variables are selected for each of the methods:

```
R> summary(grp1.HC)

Result of HC-based biomarker selection using 3 modelling methods.
Number of different settings for each method:
studentt      pls      vip
       1        1        1

Total number of variables in the X matrix: 1632
Number of variables selected:
studentt      pls      vip
      38       22       27
```

The selected variables are available through function `selection`:

```
R> selection(grp1.HC)

$studentt
$studentt[[1]]
 [1]  963  967  266 1034 1242 ...
```

```
$pls
$pls[[1]]
 [1]   30  266  963  967  985 ...

$vip
$vip[[1]]
 [1]   30  266  399  847  963 ...
```

The output has been edited to show only the first five selected variables in each case. As can be expected, there is overlap but the results are not completely the same.

The three possibilities for the `type` argument in the `get.biom` function lead to model coefficients, p values and stability fractions for `"coef"`, `"HC"` and `"stab"`, respectively. To extract them, the generic function `coef.BMark` is available. For example, Figure 2 was produced by providing `HCthresh` with a vector of p values from the `grp1.HC` object:

```
R> grp1.HCsel <- HCthresh(coef(grp1.HC)$studentt[[1]], plot = TRUE)
```

Finally, to assess the performance of the separate approaches, receiver operating characteristics (ROC) curves are implemented according to the suggestions in Lumley (2004). These simply plot the fraction of false positives (also called the complement of the specificity) versus the fraction of true positives (the sensitivity), where the order of inclusion is from big to small. The data for the ROC curve can be calculated by:

```
R> real.markers <- which(SpikePos$annotation$found.in.standards > 0)
R> grp1.troc <- ROC(1/coef(grp1.HC)$studentt[[1]], real.markers)
```

Since the smallest p values correspond to the largest t statistics, we can simply use the inverse of the p values as input for the ROC function, which expects the biggest values to be included first. Individual values in a ROC plot can be calculated by presenting the list of selected variables, together with the "true" markers and the number of variables to function `roc.value`. The following example calculates how well the variables selected by the combination of the t test and HC thresholding do:

```
R> grp1.HC.troc <- roc.value(selection(grp1.HC)$studentt[[1]],
+    real.markers, totalN = ncol(SpikePos$data))
```

Function `selection` is used here to extract the selected variables from the `BMark` object. Finally, then, the ROC curve can be created:

```
R> plot(grp1.troc, type = "l", main = "Student t")
R> points(grp1.HC.troc, col = "red", pch = 19, cex = 1.5)
```

The result is shown in the left plot of Figure 3. The top left corner, as usual, corresponds to the ideal situation where all biomarkers are found without any false positives. The t values selected early (those with large absolute sizes) in general correspond to true biomarkers, as can be expected. The result shows that HC in this case does an excellent job in finding an appropriate cutoff value, and puts it right at the point where the number of false positives starts to increase. This effect is even stronger in the plots for PLS and VIP in the same figure,

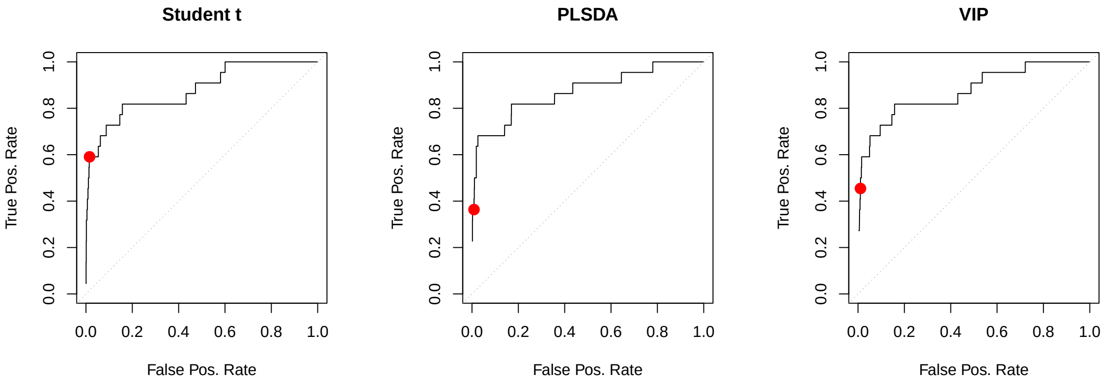

Figure 3: ROC curves for t statistics (left), PLSDA model coefficients (middle) and VIP values (right), comparing Group 1 with controls; apple data, positive ionization. In each case, the HC threshold is indicated with a red dot. For the PLS and VIP curves, two latent variables are employed (the default).

prepared by very similar code. In some practical applications one may be prepared to accept many more false positives in order to maximize the number of true positives.

It is important to realize that ROC curves such as the ones in Figure 3 can only be made when knowledge is available about what are the real biomarkers, stressing the importance of the availability of real and simulated data sets with known class differences.

3.3. Parameters

For both HC and stability selection, it has been found previously that the exact parameter values do not influence the results very much and that the methods are fairly robust (Donoho and Jin 2008; Meinshausen and Bühlmann 2010) – the default values should be appropriate in a wide range of applications. The HC approach has only two parameters: `HCalpha`, determining the maximal number of significant differences to be found, and, for multivariate discrimination methods, `nset`, which sets the number of permutations for establishing null distributions for PCLDA, PLSDA and the VIP.

For stability selection, one parameter is `min.present`, i.e., the minimal fraction of perturbed sets in which a variable needs to be selected in order to be considered a possible biomarker. This value is by default set to 0.1; setting it to higher values will lead to fewer variables being selected. Other parameters, common for all primary methods, are concerned with how the data are perturbed: `oob.fraction` defines the fraction of the samples that is left out in each subsampling, and `variable.fraction` the fraction of the variables. These are set by default to 0.3 and 0.5, respectively. Using `oob.size` rather than `oob.fraction` allows the user to define the real number of left-out samples rather than the fraction. Parameter `max.seg` defines how many subsamplings are to be provided: for very small data sets in a two-class setting, it is possible to assess all possible combinations, but in the majority of cases a random selection of `max.seg` will be returned.

The main difference within the stability selection parameters is found between the `lasso` and the other methods. The lasso itself already provides a primary selection paradigm, in that most of the coefficients will be zero for a given value of the `lambda` parameter. On the other hand, stability selection applied to t tests or regression coefficients from models like PLS will have to decide for each iteration which variables are "in" and which are "out". This is governed by `ntop`, one of the arguments in `biom.options()`, by default set to 10. That is, a particular variable will be selected by the primary method if it is among the ten biggest values. Setting `ntop` to a larger value will lead to the selection of more variables.

4. Two-class discrimination

4.1. Higher criticism

To further illustrate the application of biomarker selection in two-class discrimination, we consider a simulated data set, based on the experimental metabolomics apple data. First, we remove those variables that we know correspond to true biomarkers:

```
R> real.markers <- which(SpikePos$annotation$found.in.standards > 0)
R> full.data <- SpikePos$data[1:20, -real.markers]
```

Next, we generate a treatment class, where differences are introduced in the first ten variables

```
R> sim.data <- gen.data2(full.data, ncontrol = 10, nbiom = 10,
+    spikeI = c(1.5, 1.75, 2), nsimul = 1)
```

The `spikeI` argument is used to define the difference between the assigned biomarker variables and the non-biomarkers. In this case, the default way of spiking is used, which is a multiplication of a biomarker variable with one of the numbers from `spikeI`; if `type = "additive"` is used, these numbers will be added to the biomarker variables. Note that the non-biomarker variables and the controls will not be changed in these simulated data sets.

The HC threshold selection is then performed in the following way:

```
R> biclass.HC <- get.biom(sim.data$X[,,1], sim.data$Y, type = "HC",
+    fmethod = c("studentt", "pls", "vip"))
```

This takes some time, because for the `pls` and `vip` methods many data permutations are needed to obtain p values for the regression coefficients. Efficient calculations enable us to calculate both (related) statistics at the same time; the same goes for the numbers of latent variables. The HC threshold for the `studentt` statistic is very fast since no simulation is needed (Wehrens and Franceschi 2012).

The structure of the `biclass.HC` object is a nested list, with entries for each of the primary selection methods, as well as an `info` field containing information on the function call. The result can be inspected by using some of the functions described earlier. The selection made by the VIP, e.g., is given by:

```
R> selection(biclass.HC)$vip[[1]]

[1]  1  2  4  5  7  8 10  9
```

The simulation used as true spike-in variables the first ten: only variables three and six are missed here. The `pls` selection chooses the same variables, but the HC-Student t combination selects many more. Yet also this selection misses number three, and places number six only at place 112 – the simulated data, just like real-life data, do not always present differences in the way that is expected. However, the first eight selected variables from the Student t are the same ones selected by the two multivariate methods. One should average the results over many data sets, consisting both of real data and simulations before drawing any conclusions on the behaviour of individual methods. An initial assessment indicates that `pls` is probably the most appropriate method when the cost of false positives is high and selections need to be very efficient; if, on the other hand, it is important not to miss true positives, the Student t test can be better (Wehrens and Franceschi 2012).

4.2. Stability selection

The stability selection for the simulated data in Section 4.1 can be obtained in a perfectly analogous way. Now also the lasso is an option – let us include that in the list of primary selection methods.

```
R> biclass.stab <- get.biom(X = sim.data$X[,,1], Y = sim.data$Y,
+    fmethod = c("studentt", "pls", "vip", "lasso"), ncomp = 2:3,
+    type = "stab")
R> summary(biclass.stab)

Result of stability-based biomarker selection using 4 modelling methods.
Number of different settings for each method:
studentt      pls      vip    lasso
       1        2        2       87

Total number of variables in the X matrix: 1610
Number of variables selected:
$studentt  $pls    $vip
[1]         2       2
27         22      27

$lasso
0.4799 0.4581 0.4373 0.4174 0.3985 0.3803 0.3631 0.3466 0.3308 0.3158 0.3014
     0      3      4      5      5      5      5      5      5      5      5
...
0.0134 0.0127 0.0122 0.0116 0.0111 0.0106 0.0101 0.0096 0.0092 0.0088
    14     14     14     14     14     14     14     14     15     15
```

The (edited) output shows that the differences between the number of selected variables for each of the methods are much smaller than in the HC example: the t test now gives results that are very close to the ones from PLS and the VIP: the multivariate methods select more variables than with HC, the t test fewer. The lasso seems to select slightly fewer variables than the other methods – only in 38 out of the 87 settings chosen for the regularization method λ it selects more than ten variables. In all methods, the first eight selected variables correspond to

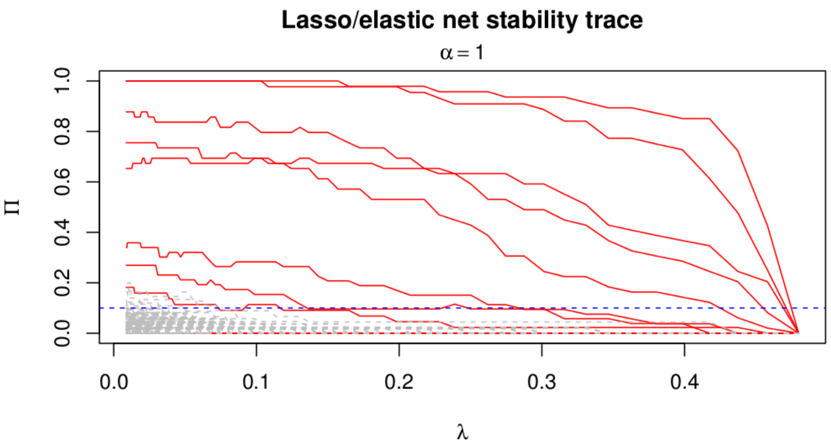

Figure 4: Lasso stability trace for simulated data: the y-axis indicates the fraction with which a certain variable is selected in the subsampling iterations. The x-axis indicates the lasso penalty. Red solid lines correspond to true biomarkers; gray dashed lines to unchanged variables.

true biomarkers, with the exception of some values for λ in the lasso, where variable number eight sometimes enters at position 10.

A convenient visualization method for the lasso result, similar to the stability plots in Meinshausen and Bühlmann (2010), can be obtained using the function `traceplot`:

```
R> traceplot(biclass.stab, lty = rep(1:2, c(10, 1600)),
+    col = rep(c("red", "gray"), c(10, 1600)))
```

The result is shown in Figure 4. Here, true biomarkers are indicated in red solid lines, and non-biomarker variables in gray dashed lines. The threshold for stability selection, here taking the default value of 0.1, is indicated with a blue dashed line. Clearly, only a few variables achieve any kind of stability in the selection, and these invariably correspond to true biomarkers. With decreasing values for λ, it can be seen that two more true biomarkers become apparent, whereas variable eight, a true biomarker, is for some λ values overshadowed by non-biomarker variables.

4.3. HC versus stability selection

Both stability selection and HC have their advantages and disadvantages – they aim at slightly different goals, too. The HC selections will by definition always lie on the ROC curve for a particular primary selection method, whereas stability selection may lead to points that are away from the curve, and can be closer to the ideal top-left corner of the plot (Meinshausen and Bühlmann 2010; Wehrens *et al.* 2011). As an example, Figure 5 shows the ROC curves of comparing Group 2 samples with controls, positive ionization mode. HC selections, shown in red, are (by definition) located on the ROC curve defined by the model coefficients. Also the PLSDA stability selection lies right on top of the ROC curve; for the Student t test and the VIP statistic, however, the stability selection shows a small advantage. Even though the

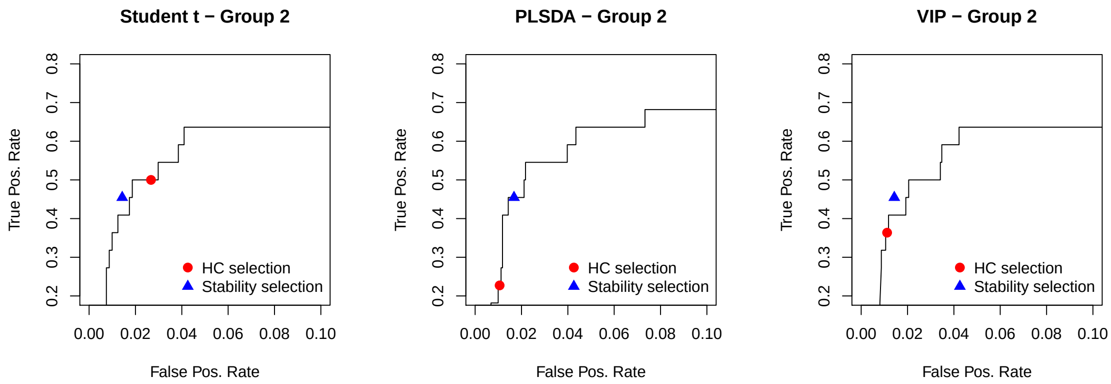

Figure 5: ROC plots for comparing Group 2 samples with controls, positive ionization. For the t test and the VIP, stability selection shows a slight improvement over the standard ROC curve.

difference may be small (the figures are showing only a small part of the complete ROC plane), it can still be relevant, in that for the same number of true positives fewer false positives are found – note that with many variables, a small distance on the x axis can still correspond to many variables. Earlier it has been found that in some cases the advantage of stability selection may be substantial (Meinshausen and Bühlmann 2010; Wehrens *et al.* 2011). In principle, it is also possible that stability selection leads to results that are less good than the ROC curve, but we have not seen examples where the difference is big.

Whereas HC selection is very fast for methods directly giving p values, like (variants of) the Student t test, multivariate methods take more time, and in these cases stability selection is faster. Typical running times for multivariate cases are a couple of minutes for HC selection and a couple of seconds for stability selection. Stability selection also is the more general method of the two: it can not only be applied in a two-class discrimination setting, but also in a regression context, as we will see in Section 5. The applicability of stability selection in discrimination, however, can be limited by the requirement that number of samples in each of the groups should be reasonably large: subsampling a group of five samples is probably not useful, and at least eight to ten samples per class are required. In the microarray and sequencing world this is still a relatively large number, but with prices dropping rapidly and the possibility for efficient multiplexing, it is expected that these numbers of replicates per class will be the rule rather than the exception within one to two years. HC selection is not directly limited by the number of replicates, although obviously better results should be expected with more information. One should, however, take into account that the method has been explicitly devised for finding differences in cases where these differences are rare and weak. The aforementioned spike-in microarray data of Choe *et al.* (2005) are therefore not very well suited for analysis by HC: the fraction of spike-ins is rather high. The HC default of $\alpha = 0.1$, also suggested by the original authors, indicates that the true differences should be present in less than ten percent of the variables.

5. Regression

Variable selection in regression is typically performed by minimizing a measure of prediction

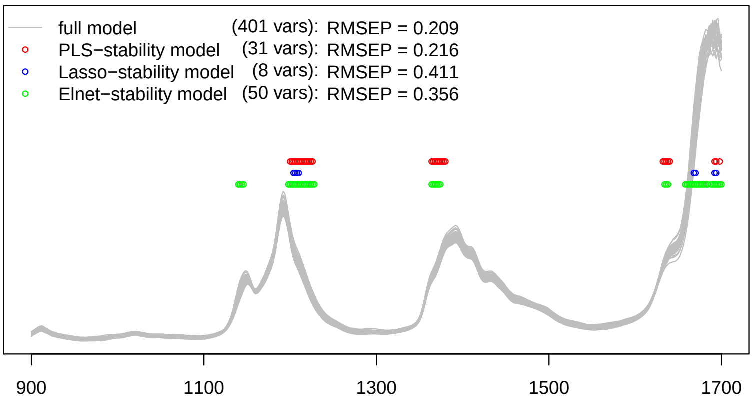

Figure 6: Stability selection in regression: the `gasoline` data from the **pls** package. Selecting only four regions, PLS stability selection achieves a comparable prediction error to the full spectrum model.

error, such as a crossvalidation error. In cases with few samples, however, this measure is usually not very reliable and one runs the risk of overfitting: this can only be evaluated by leaving out more samples and using these as a test set. Stability selection, on the other hand, does not explicitly seek to minimize prediction error, but only identifies those variables that are consistently important. As an example, consider the `gasoline` data, available in the **pls** package. These data have been used extensively in literature as test data for variable selection: the aim is to predict the octane number of the samples based on near-infrared spectral data, containing intensities at 401 wavelengths.

Stability selection with PLS (using three latent variables) and the lasso can be performed in the following steps. First, we load the data (here, the gasoline data from the **pls** package), and create indices for training and test sets, consisting of the odd and even samples, respectively. Next, we define the values for the lasso regularization parameter that we want to use, and call `get.biom`:

```
R> data("gasoline")
R> train.idx <- seq(1, 60, by = 2)
R> test.idx <- seq(2, 60, by = 2)
R> .biom.options$lasso$lambda <- exp(seq(-6, -4, length = 100))
R> gasoline.stab <- get.biom(gasoline$NIR[train.idx,],
+    gasoline$octane[train.idx], ncomp = 3, fmethod = c("pls", "lasso"),
+    type = "stab", scale.p = "none")
```

We can also calculate the results of the elastic net, here taking a (rather arbitrary) value of $\alpha = 0.5$:

```
R> .biom.options$lasso$alpha <- 0.5
R> gasoline.elnetstab <- get.biom(gasoline$NIR[train.idx,],
+    gasoline$octane[train.idx], ncomp = 3, fmethod = "lasso",
+    type = "stab", scale.p = "none")
```

The results are shown in Figure 6. The full PLS model, using 401 wavelengths, achieves the lowest error, but the stability-selected PLS model (shown with red plotting symbols) is not far behind. What is interesting is that the variables are selected in only four distinct regions, evidently reflecting physical phenomena in these areas related to the property of interest. The variables selected by the lasso model are in two of these four regions: the ones selected by the elastic net add another, fifth region, and considerably expand the region in the right of the figure. For both the lasso and the elastic net, the λ values shown have been selected by calling `cv.glmnet` on the training data and picking the λ value with the absolute smallest crossvalidation error. Prediction errors for the two lasso-based models are quite a bit larger than for the PLS models. However, it should be noted that PLS was designed to work just with data of this type: highly correlated and smooth spectra, for which a low number of latent variables is able to extract virtually all relevant information. It should also be noted that the wavelength selection here is *not* based on an explicit minimization of prediction error, the criterion in virtually all other variable selection methods.

6. Outlook

BioMark is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=BioMark` under the GPL license (version 2 or later). The first version to be released, 0.2.6, contained stability-selection functions for PLS, PCR and the t test (Wehrens *et al.* 2011). The next release, 0.3.0, added the higher-criticism functions to the repertoire (Wehrens and Franceschi 2012), while the version 0.4.0 introduced the lasso option of the stability selection. This is also the first version that – for stability selection – allows variable selection in regression problems.

Future developments will primarily focus on stability selection. One extension will be the implementation of variable weights, leading to the *randomized stability selection* as advocated by Meinshausen and Bühlmann (2010). Since the differences between the random-forest type of stability selection, currently implemented, and the weighted stability selection are not likely to be very large, the main purpose of this extension is completeness and easier comparison between the two approaches. Another extension is the possibility of incorporating user-defined tests in a simple way. At this point, this cannot be done without editing the source and recompiling the package. We hope that the package will provide a user-friendly approach to variable selection in many areas of science, and remain open to suggestions for further improvements.

References

Barker M, Rayens W (2003). "Partial Least Squares for Discrimination." *Journal of Chemometrics*, **17**, 166–173.

Benjamini Y, Hochberg Y (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society B*, **57**, 289–300.

Breiman L (2001). "Random Forests." *Machine Learning*, **45**(1), 5–32.

Choe SE, Boutros M, Michelson AM, Church GM, Halfon MS (2005). "Preferred Analysis Methods for Affymetrix GeneChips Revealed by a Wholly Defined Control Dataset." *Genome Biology*, **6**(2), R16.

Donoho D, Jin J (2004). "Higher Criticism for Detecting Sparse Heterogeneous Mixtures." *The Annals of Statistics*, **32**, 962–994.

Donoho D, Jin J (2008). "Higher Criticism Thresholding: Optimal Feature Selection When Useful Features Are Rare and Weak." *Proceedings of the National Academy of Sciences*, **105**(39), 14790–14795.

Efron B, Tibshirani R (2007). "On Testing the Significance of Sets of Genes." *The Annals of Statistics*, **1**, 107–129.

Franceschi P, Masuero D, Vrhovsek U, Mattivi F, Wehrens R (2012a). "A Benchmark Spike-In Data Set for Biomarker Identification in Metabolomics." *Journal of Chemometrics*, **26**, 16–24.

Franceschi P, Vrhovsek U, Mattivi F, Wehrens R (2012b). "Metabolic Biomarker Identification with Few Samples." In K Varmuza (ed.), *Chemometrics*. InTech Open Access Publishers. URL `http://www.intechopen.com/download/pdf/33610`.

Friedman J, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software*, **33**(1). URL `http://www.jstatsoft.org/v33/i01/`.

Ligges U, Mächler M (2003). "**scatterplot3d** – An R Package for Visualizing Multivariate Data." *Journal of Statistical Software*, **8**(11), 1–20. URL `http://www.jstatsoft.org/v11/i08/`.

Lumley T (2004). "Programmers' Niche: A Simple Class, in S3 and S4." *R News*, **4**(1), 33–36. URL `http://CRAN.R-project.org/doc/Rnews/`.

Meinshausen N, Bühlmann P (2010). "Stability Selection." *Journal of the Royal Statistical Society B*, **72**, 417–473.

Mevik BH, Wehrens R (2007). "The **pls** Package: Principal Component and Partial Least Squares Regression in R." *Journal of Statistical Software*, **18**(2). URL `http://www.jstatsoft.org/v18/i02/`.

Opgen-Rhein R, Zuber V, Strimmer K (2012). ***st**: Shrinkage t Statistic and CAT Score.* R package version 1.1.8, URL `http://CRAN.R-project.org/package=st`.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing.* Vienna, Austria. ISBN 3-900051-07-0, URL `http://www.R-project.org/`.

Smith CA, Want EJ, O'Maille G, Abagyan R, Siuzdak G (2006). "**XCMS**: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification." *Analytical Chemistry*, **78**, 779–787.

Storey JD (2002). "A Direct Approach to False Discovery Rates." *Journal of the Royal Statistical Society B*, **64**, 479–498.

Storey JD, Tibshirani R (2003). "Statistical Significance for Genome-Wide Studies." *Proceedings of the National Academy of Sciences*, **100**, 9440–9445.

Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP (2005). "Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles." *Proceedings of the National Academy of Sciences*, **102**, 15545–15550.

Wehrens R, Franceschi P (2012). "Thresholding for Biomarker Selection Using Higher Criticism." *Molecular BioSystems*, **8**, 2339–2346.

Wehrens R, Franceschi P, Vrhovsek U, Mattivi F (2011). "Stability-Based Biomarker Selection." *Analytica Chimica Acta*, **705**, 15–23.

Wessels HJCT, Bloemberg TG, van Dael M, Wehrens R, Buydens LMC, van den Heuvel LP, Gloerich J (2012). "A Comprehensive Full Factorial LC-MS/MS Proteomics Benchmark Dataset." *Proteomics*, **12**, 2276–2281.

Wold S, Sjöström M, Eriksson L (2001). "PLS-Regression: A Basic Tool of Chemometrics." *Chemometrics and Intelligent Laboratory Systems*, **58**, 109–130.

Zuber V, Strimmer K (2009). "Gene Ranking and Biomarker Discovery Under Correlation." *Bioinformatics*, **25**, 2700–2707.

Affiliation:

Ron Wehrens
Biostatistics and Data Management
Fondazione Edmund Mach
Via E. Mach 1
I-38010, San Michele all'Adige, Italy
E-mail: ron.wehrens@fmach.it
URL: http://cri.fmach.eu/BDM

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 51, Issue 10 *Submitted:* 2012-02-17
November 2012 *Accepted:* 2012-08-10