tensorICA - Tensorial Independent Component Analysis for tensor-valued multi-omic data - R package

Han Jing, Andrew E. Teschendorff, Joni Virta
jinghan@picb.ac.cn
andrew@picb.ac.cn
joni.virta@aalto.fi

January 9, 2019

1. Introduction

This is a demo for using the `tensorICA` package in R. The `tensorICA` package is designed for fast decomposition analysis of tensor-valued multi-omic data. It contains several functions for decomposing multi-omic tensor-valued data implementing two tensorial ICA methods and several utility functions for visualizing the relationship between component against phenotype to help with feature selection. Tensorial ICA aims to infer from a data-tensor, statistically independent sources of data variation, which should better correspond to underlying biological factors. Indeed, since biological sources of data variation are generally non-Gaussian and often sparse, the statistical independence assumption implicit in the ICA formalism can help improve the deconvolution of complex mixtures and thus better identify the true sources of data variation. In the package, we here include two tensorial methods call JADE and FOBI, which are the abbreviation of tensorial joint approximate diagonalization of high-order eigenmatrices and tensorial fourth-order blind identication, respectively. These two methods have been elaborated in `tensorBSS` package.

2. Get started with tensorICA package

Tensorial ICA works by decomposing a data tensor into a source tensor and mixing matrices. The key property of tICA is that the independent components in source tensor are as statistically independent from each other as possible. Statistical independence is a stronger criterion than linear decorrelation and allows improved inference of sparse sources of data variation. A prior tensorial PCA is requested as a whitening step to reduce the noise. Positive kurtosis can be used to rank independent components to select the most sparse factors. The largest absolute weights within each independent component can be used for feature selection, while the corresponding component in the mixing matrices informs about the pattern of variation of this component across data types and samples, respectively.

2.1 Load tensorICA package and example data

We expect the tensorial ICA methods can capture the variation correlated to the phenotype. However, evaluating methods on real data objectively is challenging due to the difficulty of defining a goldstandard set of true positive associations. Fortunately, a meta-analysis of several smoking EWAS in blood has demonstrated that smoking-associated differentially methylated CpGs are highly reproducible, defining a gold-standard set of 62 smoking-associated CpGs. In addition, it has been showed that all 62 smoking-associated CpGs are associated with smoking exposure effectively if DNA methylation is measured in buccal cell. So, here we use a small tensor-valued DNA methylation dataset meatured on blood and buccal tissues as an example dataset to test methods in terms of their ability to identify these 62 smoking-associated CpGs with their corresponding smoking phenotype information. This tensor dataset consists of two matched HumanMethylation450 BeadChip data matrices meatured on blood and buccal. In each tissue layer matrix, the data is

defined over the same CpG sites (columns) and the same individuals (rows). Because there are two distinct samples (blood and buccal) per individual, most of the variation is genetic. Hence, to reduce this background genetic variation, the CpG sites contained in the tensor dataset are obtained by combining 1000 non-smoking associated CpGs with the 62 smoking associated CpGs. We expect tensorICA can capture the components which correlated to smoking phenotype with large weight on the 62 smoking associated CpGs.
The input example data `buccalbloodtensor` is stored in the package. We can load it with following commands.

```
> library(tensorICA);
> data(buccalbloodtensor);
```

It is a list that contains all the data we are going to use the this demo. `buccalbloodtensor` is a list containing 4 elements. Thereinto, the test tensor-valued DNA methylation dataset is stored in `buccalbloodtensor\$data`. This tensor is builded by 2 layers of dataset matrix of different tissue type. The 11 matched phenotypes of the same 152 samples are stored in `buccalbloodtensor\$pheno.l`. Among all the phenotype, Smoking is the average smoking pack numbers per year of each sample and SmokingStatus is the sample's smoking status, which 0 stands for nonsmokers, 1 stands for ex-smokers and 2 stands for current smokers. A important property corresponding to the input phenotype is whether the phenotypes are categorical, which is stored in a boolean logical vector `buccalbloodtensor\$pheno.i`. Lastly, `buccalbloodtensor\$testDMCs` is the 1062 test CpG names. The first 1000 CpGs are the randomly picked non-smoking associated CpGs and the last 62 CpGs are the smoking associated CpGs.

2.2 Estimate subspace dimension carrying significant variation

Before the decomposition, we need to estimate subspace dimension carrying significant variation in the first step of dimension reduction. Function `EstDim` is based on Random Matrix Theory(RMT) algorithm to do dimension reduction and output subspace size which can be discribed as the number of significant component in further decompsition. Here in the output list, `dim` is a vector containing subspace size of each tissue type matrix in dimension reduction, in another words, which is the number of significant components of each tissue type matrix and `dJ` is the number of significant components of joint variation across all tissue type matrices. These two parameters will be used in the next decomposing step.

```
> require(isva);
> Dim.l <- EstDim(buccalbloodtensor$data);
> dim <- Dim.l$dim;
> dJ <- Dim.l$dJ;
```

2.3 Perform tensorial ICA

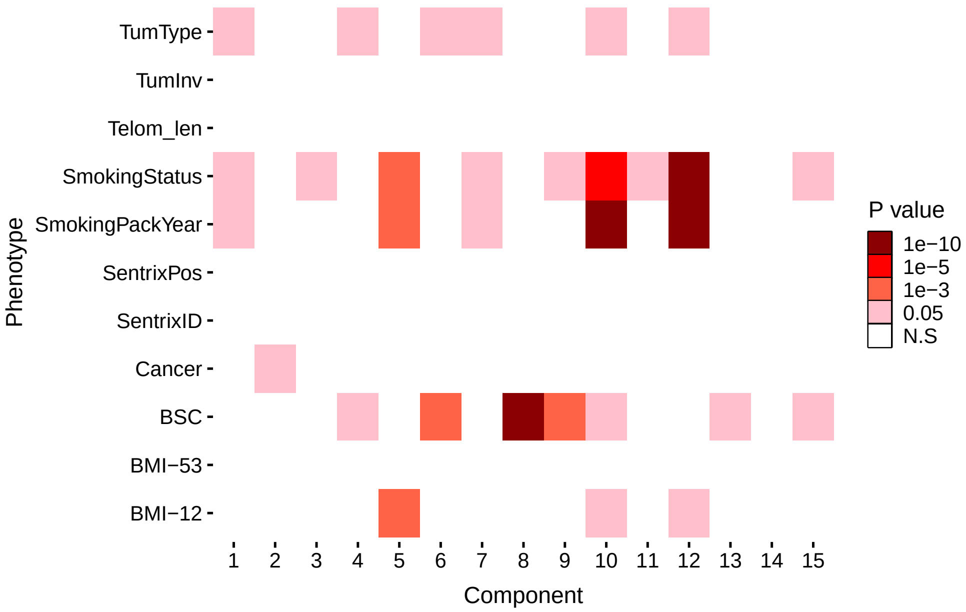
The next step is to then decompose the buccal-blood tensor dataset using tensorial ICA to infer the sources of data variation carrying significant variation. We observe that the function `DoTICA` uses tensorial PCA (TPCA) as a preliminary step to do the dimensional reduction. The 'method' parameter controls which ICA method we use, with 'JADE' and 'FOBI' the two options available. Here we choose tFOBI method. The output object `tica.o` contain respective rotation matrices `U` from TPCA step, unmixing matrices `W` and source matrices `S` which are of the same size as input matrices containing the principal components.

```
> tica.o <- DoTICA(Data = buccalbloodtensor$data, dim = dim, method = "FOBI");
```

2.4 Correlate inferred components against phenotype

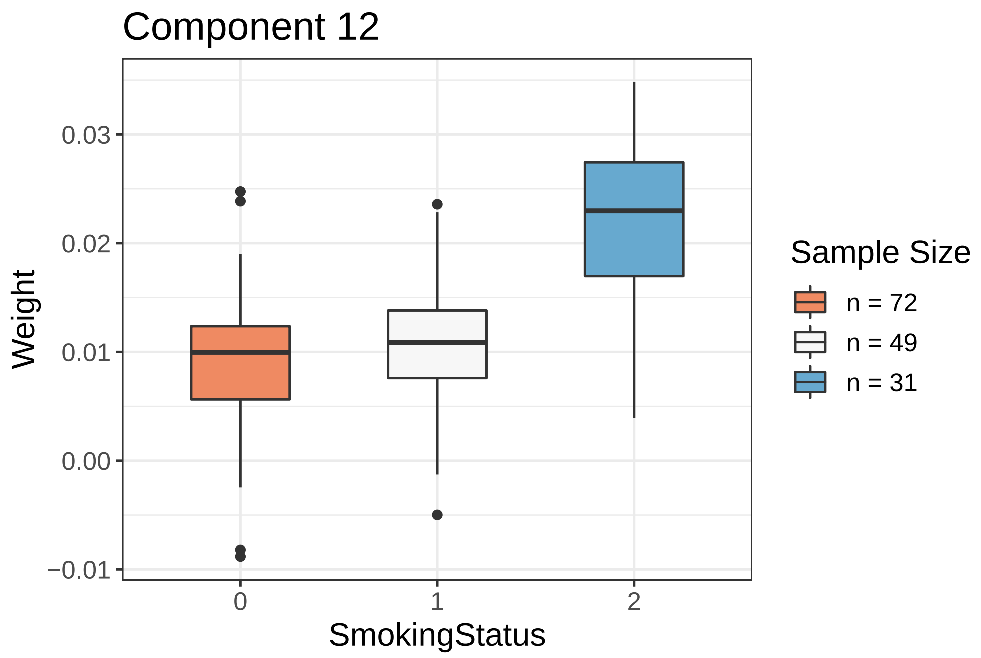
Now that we have estimated all the significant independent sources, a typical next step is to correlate the inferred components against associated phenotypes. To help assess which components are correlated with which phenotypes, the function cor_phenotype provides a correlation P-value heatmap between phenotype and the components. The P-values derive from a linear regression model in the case of continuous phenotypes, or from a Kruskal-Wallis/ANOVA test in the case of categorical phenotypes. We can also use the function cor_phenotype to visualize the mixing matrix component weights against phenotype. In the package, the associated phenotype buccalbloodtensor$pheno.l is a list of 11 matched phenotypes of the 152 samples. Among them, SmokingPackYear is the average smoking pack numbers per year of each sample and SmokingStatus is the sample's smoking status (0 stands for nonsmokers, 1 stands for ex-smokers and 2 stands for current smokers). We note that the components in the P-value heatmap are not ranked by variance, since the components derived from ICA-like methods do not rely on variance for inference but on higher order statistical moments (notably Kurtosis). Selection of components by variance is only used in the prewhitening step, as implemented using TPCA.

```
> phenotype.p <- cor_phenotype(tica.o = tica.o, phenotype = buccalbloodtensor$pheno.l,
+                                phenotype.is.categorical = buccalbloodtensor$pheno.i);
> phenotype.p$pv.p;
```

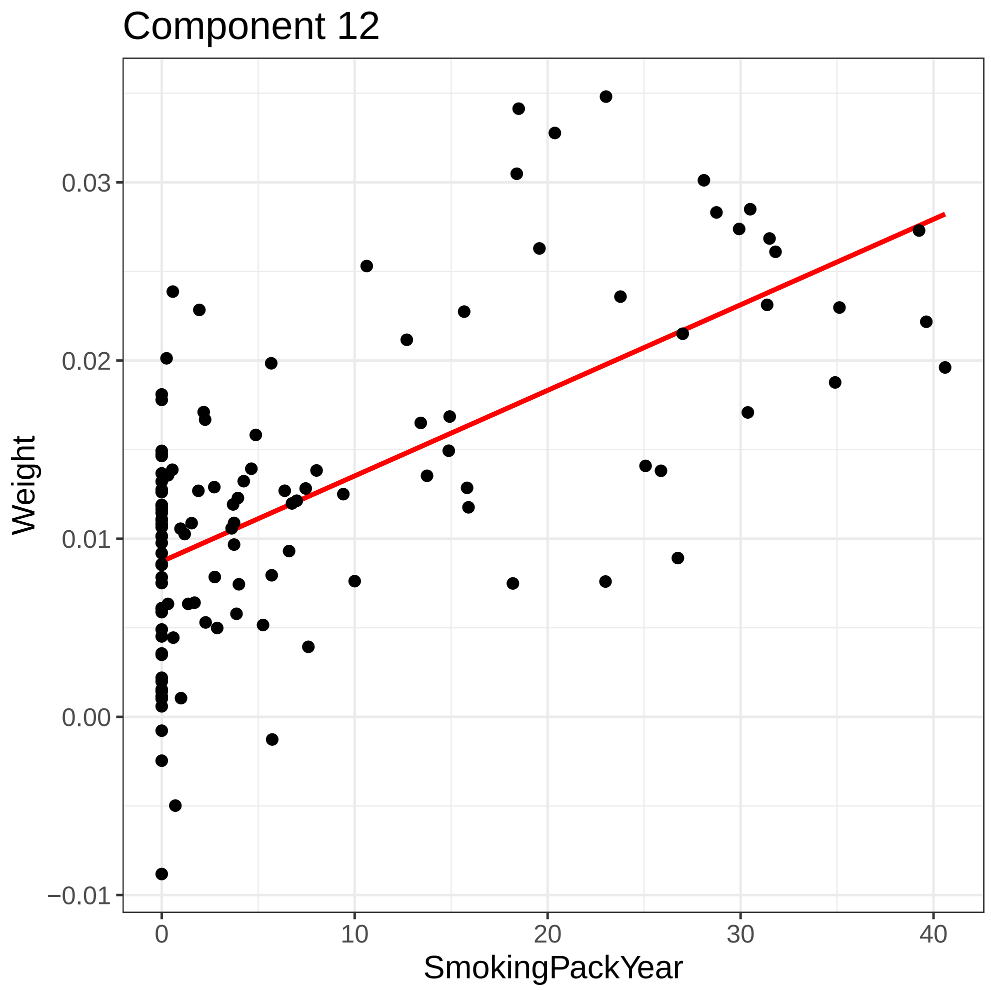


From the P-value heatmap, we can now clearly see that component 10 and component 12 are significantly correlated with smoking status and smoking pack years. Correspondingly, by plotting the weights we can observe for instance a strong correlation between component 12 and smoking related phenotypes.

```
> phenotype.p12 <- cor_phenotype(tica.o = tica.o, phenotype = buccalbloodtensor$pheno.l,
+                                  phenotype.is.categorical = buccalbloodtensor$pheno.i,
+                                  component = 12);
> phenotype.p12$compheno.pl$SmokingStatus;
```

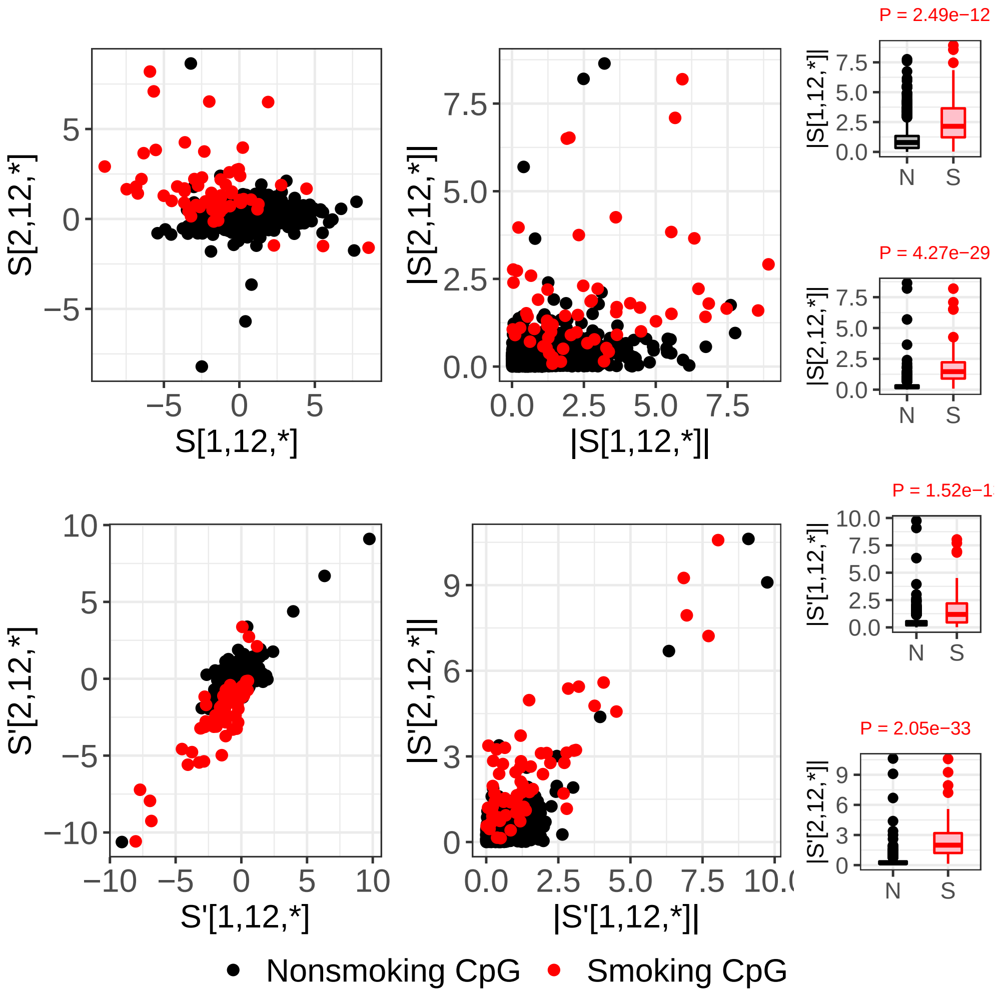


```
> phenotype.p12$compheno.pl$SmokingPackYear
```



2.5 Weights in feature space

The independent sources are defined over features, and the weights in these components inform us about which features are important in driving those components. Thus, for instance, we can inspect which features are driving component 12, which as shown above is associated with smoking. We note that for the sources there is a pair of independent components associated with component-12, since there are two tissue-types. Thus, a scatterplot of these pairs of components is appropriate, which we can do in the independent component space or in the rotated basis (labeled by S') where the two dimensions correspond directly to the two tissue-types (as opposed to some linear combination). We can also plot either the weights or absolute weights. The package provides a function to generate all of these plots. In this particular case, the red labeled features correspond to the gold-standard smoking-associated CpGs and these should have larger absolute weights in component 12 compared to non-smoking CpGs. This is confirmed with a Wilcoxon rank sum test between smoking associated CpG (S) and non-smoking CpGs (N). The result shows significant differences in all cases.



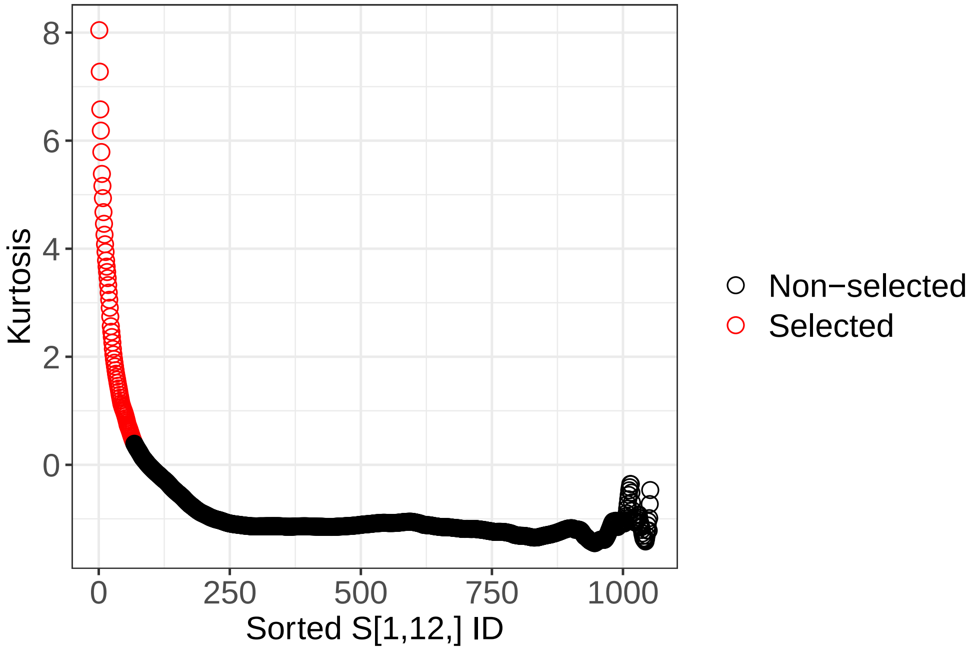
2.6 Feature selection

More generally, we may not know a-priori which features are important or we may not have a feature importance label. Typically, the task is to select features from the infered components, for which we provide the function feature_selection. The package provide two ways of selecting the number of top weighted CpGs. First, we can estimate the number of driver features by assessing how much each feature contributes to the kurtosis of the component. This is done in an iterative sequential manner where top-ranked features are subsequently removed until the resulting kurtosis falls below some prespecfied threshold. The threshold is determined by a given upper kurtosis quantile as given by a Gaussian null distribution. This is the parameter 'CLkurt' in the function feature_selection. Alternatively, we can simply specify the number of top-ranked features to select with the parameter 'topN'. The function returns the index list 'feature.n$pred.idx', which is list of the index of selected features in the two tissues. Here we only show the features selected in the first tissue as an example. In this case, the function selects the top 67 CpGs.

```
> feature.k <- feature_selection(tica.o = tica.o, component = 12, CLkurt = 0.95);
> buccalbloodtensor$testDMCs[feature.k$pred.idx[[1]]];
```

```
 [1] "cg12876356" "cg11207515" "cg25987452" "cg19945931" "cg23916896"
 [6] "cg23161492" "cg13751956" "cg23079012" "cg18316974" "cg03636183"
[11] "cg16733643" "cg10500026" "cg23576855" "cg05575921" "cg12803068"
[16] "cg01940273" "cg08757924" "cg09469111" "cg01351337" "cg19670431"
[21] "cg02157475" "cg21242417" "cg03991871" "cg08880327" "cg23520688"
[26] "cg16505233" "cg06178322" "cg04621997" "cg00710180" "cg25648203"
[31] "cg24090911" "cg00969573" "cg09417849" "cg11577329" "cg26806511"
[36] "cg17863681" "cg26729913" "cg00078857" "cg26703534" "cg20457894"
[41] "cg26192556" "cg09095364" "cg02378784" "cg22352709" "cg07720851"
[46] "cg09509179" "cg26963277" "cg23771366" "cg00719224" "cg25921609"
[51] "cg01615339" "cg11660018" "cg12409982" "cg09935388" "cg23114183"
[56] "cg10606240" "cg24083631" "cg25575628" "cg21103074" "cg00699461"
[61] "cg21121843" "cg01249187" "cg07756483" "cg12806681" "cg16701266"
[66] "cg17248924" "cg19859270"
```

```
> feature.k$k.p[[1]];
```



```
> feature.n <- feature_selection(tica.o = tica.o, component = 12, topN = 62);
> buccalbloodtensor$testDMCs[feature.n$pred.idx[[1]]];
```

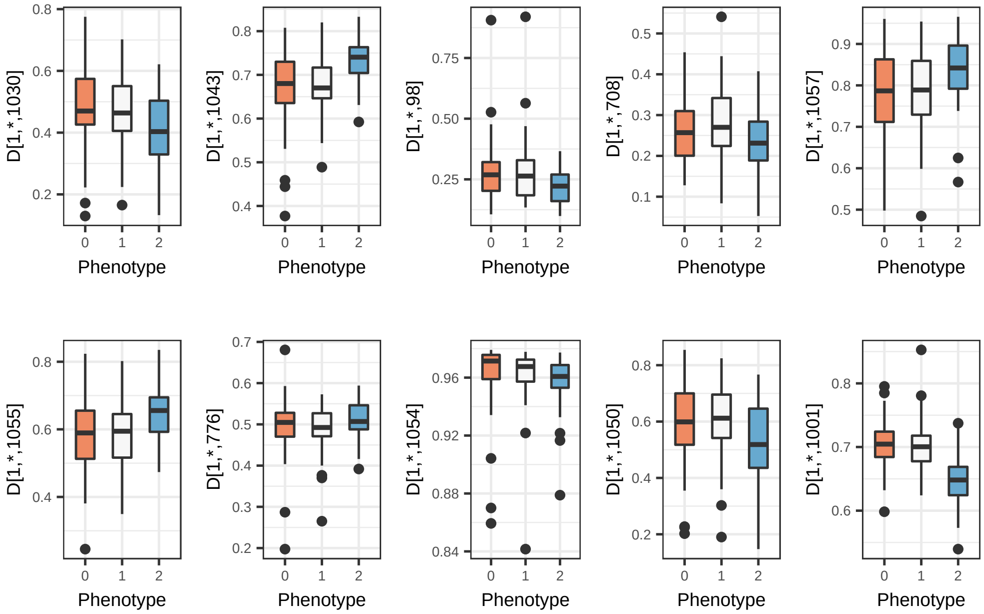
```
 [1] "cg12876356" "cg11207515" "cg25987452" "cg19945931" "cg23916896"
 [6] "cg23161492" "cg13751956" "cg23079012" "cg18316974" "cg03636183"
[11] "cg16733643" "cg10500026" "cg23576855" "cg05575921" "cg12803068"
[16] "cg01940273" "cg08757924" "cg09469111" "cg01351337" "cg19670431"
```

```
[21] "cg02157475" "cg21242417" "cg03991871" "cg08880327" "cg23520688"
[26] "cg16505233" "cg06178322" "cg04621997" "cg00710180" "cg25648203"
[31] "cg24090911" "cg00969573" "cg09417849" "cg11577329" "cg26806511"
[36] "cg17863681" "cg26729913" "cg00078857" "cg26703534" "cg20457894"
[41] "cg26192556" "cg09095364" "cg02378784" "cg22352709" "cg07720851"
[46] "cg09509179" "cg26963277" "cg23771366" "cg00719224" "cg25921609"
[51] "cg01615339" "cg11660018" "cg12409982" "cg09935388" "cg23114183"
[56] "cg10606240" "cg24083631" "cg25575628" "cg21103074" "cg00699461"
[61] "cg21121843" "cg01249187"
```

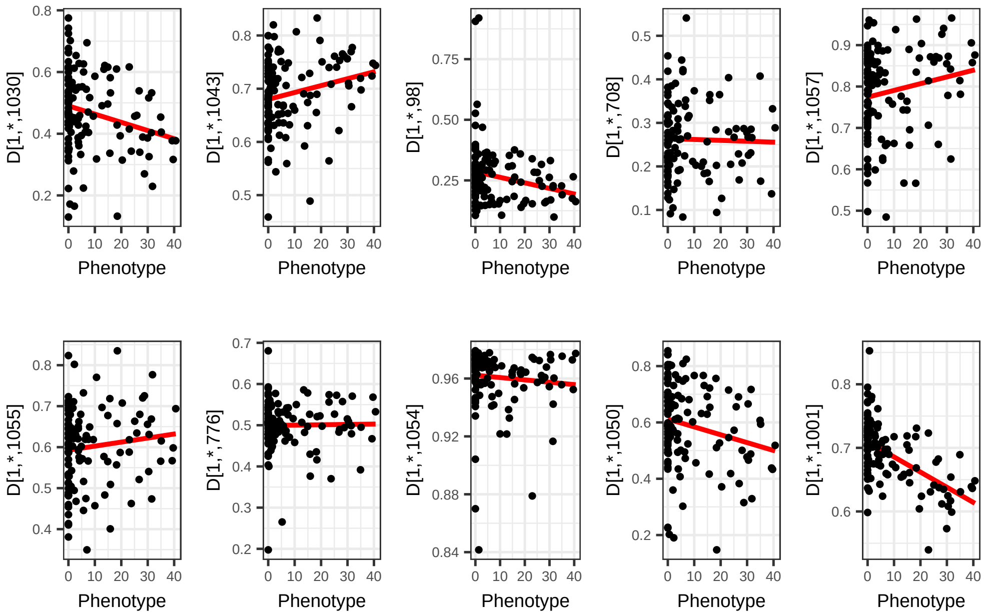
2.7 Feature profile

Finally, since we have selected features, the coresponding molecular profiles should be checked against phenotypes of interest. In this scenario, the DNA methylation profiles of selected features can be plotted against smoking status and smoking pack years by function feature_profile. By default, top 10 features are shown in the plot.

```
> (profile.p <- feature_profile(Data = buccalbloodtensor$data,
+ phenotype.v = buccalbloodtensor$pheno.l$SmokingStatus, phenotype.is.categorical = T,
+ feature.index = feature.k$pred.idx[[1]][1:10], tissue.index = 1));
```



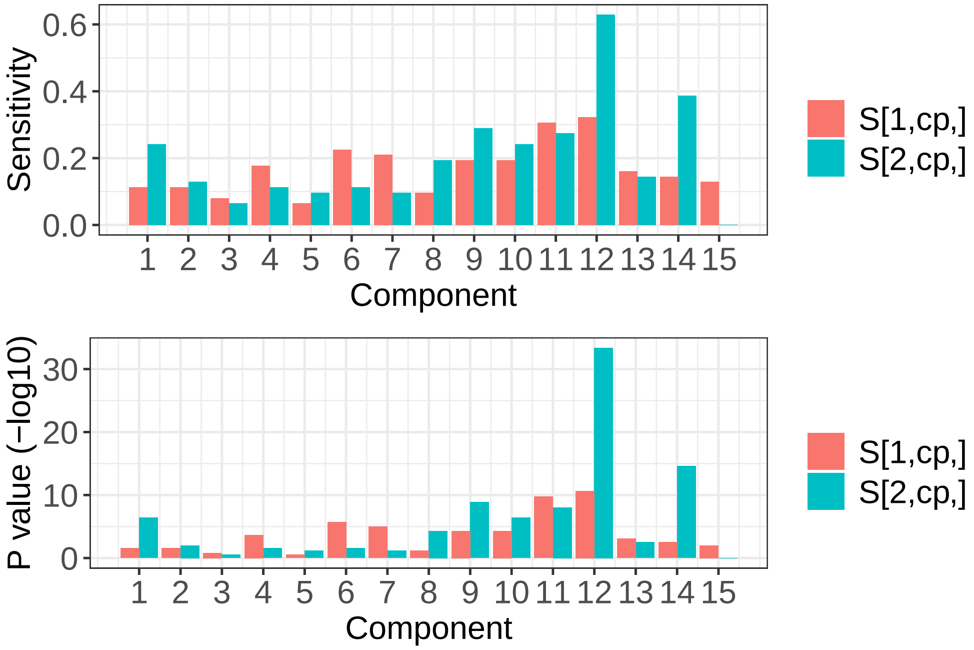
```
> (profile.p <- feature_profile(Data = buccalbloodtensor$data,
+ phenotype.v = buccalbloodtensor$pheno.l$SmokingPackYear, phenotype.is.categorical = F,
+ feature.index = feature.k$pred.idx[[1]][1:10], tissue.index = 1));
```



2.8 Check sensitivity

To check the accuracy of the methods, we can use function EstSE to estimate the sensitivity if the true positive set is known.

```
> require(cowplot)
> SE <- EstSE(tica.o = tica.o, tp = 1001:1062);
> plot_grid(SE$se.p, SE$pv.p, nrow = 2);
```



3. Citation

Teschendorff AE, Han J, Paul D, Virta J, Nordhausen K. *Tensorial Blind Source Separation for Improved Analysis of Multi-Omic Data.* Genome Biology (2018) 19:76.

4. Session information:

Output of sessionInfo on the system on which this document was compiled:

- R version 3.5.2 (2018-12-20), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `CentOS Linux 7 (Core)`
- Matrix products: default
- BLAS: `/usr/local/lib64/R/3.5.2/lib64/R/lib/libRblas.so`
- LAPACK: `/usr/local/lib64/R/3.5.2/lib64/R/lib/libRlapack.so`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: JADE 2.0-1, bindrcpp 0.2.2, cowplot 0.9.3, dplyr 0.7.8, fastICA 1.2-1, ggplot2 3.1.0, isva 1.9, qvalue 2.14.0, tensorICA 1.0.0, tidyr 0.8.2
- Loaded via a namespace (and not attached): GGally 1.4.0, ICS 1.3-1, ICSNP 1.1-1, ICtest 0.3-1, Matrix 1.2-15, R6 2.3.0, RColorBrewer 1.1-2, Rcpp 1.0.0, TTR 0.23-4, assertthat 0.2.0, bindr 0.1.1, boot 1.3-20, clue 0.3-56, cluster 2.0.7-1, colorspace 1.3-2, compiler 3.5.2, crayon 1.3.4, curl 3.2, digest 0.6.18, forecast 8.4, fracdiff 1.4-2, glue 1.3.0, grid 3.5.2, gtable 0.2.0, knitr 1.21, labeling 0.3, lattice 0.20-38, lazyeval 0.2.1, lmtest 0.9-36, magrittr 1.5, munsell 0.5.0, mvtnorm 1.0-8, nlme 3.1-137, nnet 7.3-12, parallel 3.5.2, pillar 1.3.1, pkgconfig 2.0.2, plyr 1.8.4, png 0.1-7, purrr 0.2.5, quadprog 1.5-5, quantmod 0.4-13, reshape 0.8.8, reshape2 1.4.3, rlang 0.3.0.1, scales 1.0.0, splines 3.5.2, stringi 1.2.4, stringr 1.3.1, survey 3.35, survival 2.43-1, tensor 1.5, tensorBSS 0.3.4, tibble 1.4.2, tidyselect 0.2.5, timeDate 3043.102, tools 3.5.2, tsBSS 0.5.2, tseries 0.10-46, urca 1.3-0, uroot 2.0-9, withr 2.1.2, xfun 0.4, xts 0.11-2, zoo 1.8-4