CASh pseudo-code and implementation

In this section we give a more detailed description of Algorithm 1. In addition, we show how this algorithm has been implemented.

We start from an alternative representation of the Shapley value, that can be given in terms of the unanimity coefficients \( (\lambda_S(v))_{S \subseteq 2^N \setminus \emptyset} \), that is:

\[
\phi_i(v) = \sum_{S \subseteq N : i \in S} \frac{\lambda_S(v)}{s}
\]

for each \( i \in N \), where \( s \) is the cardinality of \( S \).

In Methods we introduced the definition of microarray game corresponding to a given Boolean matrix. An equivalent way to calculate the microarray game \( \tilde{v} \) corresponding to a Boolean matrix \( B \) is as a sum of unanimity games as follows

\[
\tilde{v} = \frac{1}{k} \sum_{j \in K : sp(B_j) \neq \emptyset} u_{sp(B_j)},
\]

where \( u_{sp(B_j)} \) is the unanimity game on \( sp(B_j) \subseteq N \), for each \( j \in K \).

Note that the computation of the Shapley value \( \phi(\tilde{v}) \) in a microarray game \( \tilde{v} \), in virtue of equations (1) and (2), is very easy, independently of the number of players in \( N \). More precisely, we have that

\[
\phi_i(\tilde{v}) = \frac{1}{k} \sum_{j=1}^{k} R_{ij}
\]

for each \( i \in N \), where the matrix \( R \in [0, 1]^{n \times k} \) is such that for each \( i \in N \) and each \( j \in K \)

\[
R_{ij} = \begin{cases} 
0 & \text{if } B_{ij} = 0 \\
\frac{1}{\text{card}(sp(B_j))} & \text{otherwise}
\end{cases}
\]

and \( \text{card}(sp(B_j)) \) is the cardinality of \( sp(B_j) \).

We describe the nonparametric Bootstrap approach to estimate the (unadjusted for multiple comparisons) \( p \)-values in the next algorithm:

**Algorithm 1**

**INPUT:**
- Two Boolean matrices \( B^1 \in \{0, 1\}^{n \times k} \), \( B^2 \in \{0, 1\}^{n \times h} \), with \( n, k, h \in \{1, 2, \ldots\} \);
- an integer number \( b \) of Bootstrap re-samples (with replacement).

**OUTPUT:**
- a Bootstrap estimation of the null distribution of Shapley value differences on the $n$ genes;
- a vector of $n$ (un-adjusted for multiple comparisons) estimated $p$-values.

**Begin:**

**step 1**: Compute the matrix $R^1 \in [0, 1]^{n \times k}$ according to relation (4) applied to Boolean matrix $B^1$; compute the matrix $R^2 \in [0, 1]^{n \times h}$ according to relation (4) applied to Boolean matrix $B^2$.

**step 2**: Compute the observed Shapley value difference $\delta_i(\phi(\bar{v}_1), \phi(\bar{v}_2)) = \frac{1}{k} \sum_{j=1}^{k} R^1_{ij} - \frac{1}{h} \sum_{j=1}^{h} R^2_{ij}$ for each $i \in N$;

**step 3**: for $r : 1$ to $b$ {

**step 4.r** Let $s^{r,1} = (s_{j}^{r,1})_{j \in \{1, \ldots, k\}} \in \{1, \ldots, k\}^k$ and $s^{r,2} = (s_{j}^{r,2})_{j \in \{1, \ldots, h\}} \in \{1, \ldots, h\}^h$ be the vectors representing the $r$-th Bootstrap re-sample (with replacement) on the indexes $\{1, \ldots, k\}$ and $\{1, \ldots, h\}$, respectively.

**step 5.r** Consider the new Boolean matrix $B^{s^{r,1}} \in \{0, 1\}^{n \times k}$ such that $B^{s^{r,1}}_j = B^1_{s^{r,1}_j}$ for each $j \in \{1, \ldots, k\}$ and the Boolean matrix $B^{s^{r,2}} \in \{0, 1\}^{n \times h}$ such that $B^{s^{r,2}}_j = B^2_{s^{r,2}_j}$ for each $j \in \{1, \ldots, h\}$.

**step 6.r** Compute the Bootstrap Shapley value difference

$$
\beta^r_i(\phi(\bar{v}_1^r), \phi(\bar{v}_2^r)) := \left| (\phi_i(\bar{v}_1^r) - \phi_i(\bar{v}_1^1)) - (\phi_i(\bar{v}_2^r) - \phi_i(\bar{v}_2^1)) \right|,
$$

(5)

for each $i \in N$, where $\bar{v}_1^r, \bar{v}_2^r$ are the microarray games corresponding to the Boolean matrix $B^{s^{r,1}}$ and $B^{s^{r,2}}$, respectively.

}

**step 7**: for each $i \in N$, compute the (un-adjusted for multiple comparisons) estimate Achieved Significance Level (ASL) or $p$-value $p_i$ of each gene $i \in N$ as follows

$$
p_i = \frac{\text{card}\{r : \beta^r_i(\phi(\bar{v}_1^r), \phi(\bar{v}_2^r)) \geq \delta_i(\phi(\bar{v}_1), \phi(\bar{v}_2))\}}{b}.
$$

(6)

**End.**

**Remark 1** In order to preserve the ties among genes in each sample, on step 5.r, the entire columns of the Boolean matrices $B^1$ and $B^2$ are re-sampled according to the vectors $s^{r,1}$ and $s^{r,2}$ defined on step 4.r, for each $r \in \{1, \ldots, b\}$.

**Remark 2** Subtracting $\phi(\bar{v}_1)$ and $\phi(\bar{v}_2)$ in (5) from the Bootstrap Shapley values in the game defined on matrix $B^{s^{r,1}}$ and $B^{s^{r,2}}$, respectively, makes the Bootstrap Shapley values correspond to the null hypotheses $\delta_i(\phi(\bar{v}_1), \phi(\bar{v}_2)) = 0$. 

2
Implementation of Algorithm 1

Algorithm 1 was implemented using the following R (www.r-project.org) code:

R1 <- shapleymat(B1)
R2 <- shapleymat(B2)
R <- matrix(NA, n, (k+h))
R[,1:k] <- R1
R[,,(k+1):(k+h)] <- R2
classes <- c(array(1, dim=c(k)), array(2, dim=c(h)))
library(multtest)
OUTPUT <- MTP(X=R, Y=classes, standardize=FALSE, B=b)

Where shapleymat() is the R function provided as additional file, MTP is function to perform resampling-based multiple hypothesis testing of the Bioconductor (www.bioconductor.org) package multtest. In arguments of MTP function, classes is a vector of class labels for column of matrix B, representing the two conditions 1 and 2 and b is the number of Bootstrap re-samples. In order to see the Bootstrap estimation of the null distribution of Shapley value differences for all of the n genes type
> OUTPUT@rawp
In order to see the vector of n (un-adjusted for multiple comparisons) estimated p-values type
> OUTPUT@statistic