Users' Guide
MultiPower

Sonia Tarazona (sotacam@eio.upv.es)

Version 0.5.0 (September 18, 2019)

Content

Introduction

The statistical power of a method, which is its ability to detect features with a true change between experimental groups, is determined by the within-group variability, the size of the effect to be detected, the significance level to be achieved and the number of replicates per experimental group (sample size). Evaluating the power in omic experiments is a challenging task because a statistical test per each omic feature is performed. This implies that the within-condition variability might be different for different features, and that the significance level must be adapted to take into account the multiple testing correction. It may also be difficult to decide the effect size to detect, especially when the natural dynamic range of the data has changed due to previous normalization procedures. Moreover, different omic platforms have varying levels of noise, dynamic ranges, etc., which may result in non-comparable performance of differential analysis methods. Consequently, in a multi-omic experiment, independently computing the statistical power for each omic might not be the best strategy if the omics are to be analysed in an integrative fashion. In this case, a joint power study for all the omics is more appropriate, since the lack of power for one of them can be compensated with the power of the others.

The MultiPower R method performs a joint power study in which the cost of the multi-omic experiment is minimized while both a minimum power for each omic must be independently achieved and an average power for all of them is required. The parameters required to compute power can be either estimated by MultiPower from multi-omic pilot or available data or set by users. MultiPower considers the multiple testing correction by adjusting the significance level to control the False Discovery Rate (FDR). Normally distributed data , count data and binary data (0/1 or TRUE/FALSE) are accepted, and optimal sample size for each omic can be computed either with the same or different sample sizes for each omic. In this last case, the cost of generating each omic sample is also considered as an additional parameter in the power maximization problem. The MultiPower method is designed to help users not only in the design of the multi-omic experiment but also to assess if an already generated multi-omic data set provides enough power for differential expression and to highlight potential limitations.

Getting started

The MultiPower method is available as an R package from https://bitbucket.org/ConesaLab/multipower/. As for other packages in bitbucket, it can be installed from R with the following instructions:

```
> install.packages("devtools")
> devtools::install_bitbucket("ConesaLab/multipower")
```

However some additional libraries must be previously installed, as described next.

Required R libraries

These are the libraries needed from the CRAN repository that can be installed with `install.packages()` function and then loaded with `library()`.

- slam
- Lpmodeler
- Rsymphony

Other required library is RnaSeqSampleSize from Bioconductor repository, that must be installed following the Bioconductor instructions at:

https://bioconductor.org/packages/release/bioc/html/RnaSeqSampleSize.html

Installing Rsymphony in Linux systems

Installing Rsymphony library in Linux systems requires some additional steps. First, the last version of SIMPHONY must be installed from the command line as follows:

```
$ svn checkout https://projects.coin-or.org/svn/SYMPHONY/releases/5.6.16
SYMPHONY-5.6.16
$ cd SYMPHONY-5.6.16
$ ./configure
$ make
$ make install
```

Other components will also be needed:

```
$ sudo apt-get install coinor-libcgl-dev coinor-libclp-dev
coinor-libcoinutils-dev coinor-libosi-dev
$ sudo apt-get install coinor-libsymphony-dev
$ sudo apt-get install autotools-dev
```

Next we can download the R package from CRAN (https://cran.r-project.org/src/contrib/Rsymphony_0.1-26.tar.gz) and install it from R:

```
> install.packages("Rsymphony_0.1-26.tar.gz", repos = NULL)
```

Optimal sample size estimation from available data

MultiPower method needs some input parameters for each omic data type in order to compute the statistical power for a two-group comparison. We recommend to estimate such input parameters from available data (pilot data or data from previous studies). In that case, or in the case that MultiPower is applied to evaluate the limitations of an already generated multi-omic data set, this is the wrapper function to be used to estimate the optimal sample size for the multi-omic experiment:

```
MultiPower(data, groups, type, d0 = 0.8, p1 = 0.2, omicPower = 0.6,
averagePower = 0.85, dispPerc = 75, fdr = 0.05, alpha = 0.05, cost = 1,
equalSize = TRUE, max.size = 200, omicCol = NULL, powerPlots = TRUE)
```

The arguments of the `MultiPower()` function are described in detail in the next sections.

Input data

data: List with as many elements as omic data types. The names of the omics should be the names of the list. Each element in this list must be a raw count data matrix, and in this case MultiPower will take into account the library sizes to estimate power; a normally distributed data matrix which must have been already pre-processed and normalized; or a binary data matrix (with 0/1 or TRUE/FALSE values). In any case, for each one of these matrices, rows must correspond to omic features (genes, methylation sites, ChIP-seq peaks, etc.) and columns to observations (biological samples, patients, etc.).

	TCGA-02-0432-01	TCGA-08-0245-01	TCGA-28-1750-01	TCGA-06-1084-01	TCGA-02-0064-01
FSTL1	6.750974	9.081284	9.961373	9.801834	10.822028
AACS	6.392410	6.572695	7.102250	6.924443	6.586677
RPS11	11.010146	10.781873	10.583565	10.540609	10.749480
CREB3L1	4.569818	4.722297	4.472831	4.290404	4.712333
ELMO2	7.789996	6.224809	6.914030	6.155211	6.817825
PNMA1	9.914452	9.418119	9.029799	8.005483	9.677593

groups: List with as many elements as omic data types. The names of the omics should be the names of the list. Each element in this list must be a vector with a length equal to the number of observations for that omic in `data` argument. Each element of this vector must indicate the experimental group where each observation belongs. Only two experimental groups are allowed.

```
[1] "Proneural"   "Proneural"   "Mesenchymal" "Mesenchymal" "Mesenchymal" "Proneural"
```

type: Vector with length equal to the number of omic data types. Each element of this vector must be a 1, 2 or 3 to indicate whether the omic data are count data (1), continuous data approximately following a normal distribution (2) or binary data (3).

Power parameters

d0: Initial Cohen's d values. It can be either a single numeric value if the same Cohen's d is required for all the omics, or a vector with a length equal to the number of omics containing the targeted Cohen's d value for each omic. The Cohen's d is defined as Δ/σ, where Δ is the absolute difference of means for each experimental group and σ is the pooled standard deviation of both groups. The Cohen's d is not dependent on the scale of the data as it happens with Δ value, so it is fair to set the same value for all the omics. Cohen [1] and Sawilowsky [2] suggested the classification in the following table to aid users to set the Cohen's d value. MultiPower computes the Cohen's d value for all the omic features and uses it to estimate the set M1 of differentially expressed (DE) features (those with d>d0), which are called pseudo-DE features. By default, d0 = 0.8. Note that for binary data, d0 is the Cohen's h value.

Rules of thumb for effect sizes measured by Cohen's d or Cohen's h

Cohen's d / Cohen's h	**Effect size**
0.01	Very small
0.2	Small
0.5	Medium
0.8	Large
1.2	Very large
2.0	Huge

p1: The expected proportion of DE features between the two groups compared. It must be a vector with length equal to the number of omics. If it is a single number, this same number will be used for all the omics. The values for p1 must be set by the user, who can decide to estimate them from pilot data by performing a differential expression analysis (this option is not provided by MultiPower). By default, p1 = 0.2.

omicPower: The minimum power that must be achieved for each omic. It must be a vector with a length equal to the number of omics. If it is a single number, this same number will be used for all the omics. By default, omicPower = 0.6.

averagePower: The minimum average power that must be globally achieved. By default, averagePower = 0.85.

dispPerc: Dispersion percentile to be used to estimate power. By default, 75 (75th percentile).

fdr: False Discovery Rate level to be used. It is the significance level after multiple testing correction. By default, fdr = 0.05. If no multiple testing correction is to be applied, this argument must be set to NULL and then alpha argument is required.

alpha: Significance level to be used only when multiple testing correction is not to be applied (fdr = NULL). By default, alpha = 0.05.

Other parameters

cost: The cost to generate a replicate (a sample) for each omic. It must be a vector with a length equal to the number of omics. If it is a single number, this same number will be used for all the omics. This argument will only be used when a different sample size per omic is allowed. By default, cost = 1 (which means that all the omics will be assumed to have the same cost).

equalSize: If TRUE (default), the same optimal sample size will be estimated for all the omics. If FALSE, omics are allowed to have different sample sizes.

max.size: Maximum allowed sample size. By default, max.size = 200.

omicCol: The color that will be used to plot each omic. It must be a vector with a length equal to the number of omics. If it is NULL (default), default colors are used.

powerPlots: If TRUE (default), power plots will be generated.

Interpretation of results

When applying `MultiPower()` function, the result is a list containing the following elements:

parameters: List with as many elements as omic data types. For each omic, each element of the list is another list containing the different parameters used to compute power, either estimated from the pilot data or provided by the user: type, allDispersions, dispersion, p1, d, delta, minFC, meanCounts, m, m1, and alld.

optimalSampleSize: List with the following elements:

n0: Sample size to achieve the minimum omic power (`omicPower`) for each omic.

n: Optimal sample size.

finalPower: Power at the optimal sample size for each omic.

fdr: See `fdr` parameter.

omicPower: See `omicPower` parameter.

averagePower: See `averagePower` parameter.

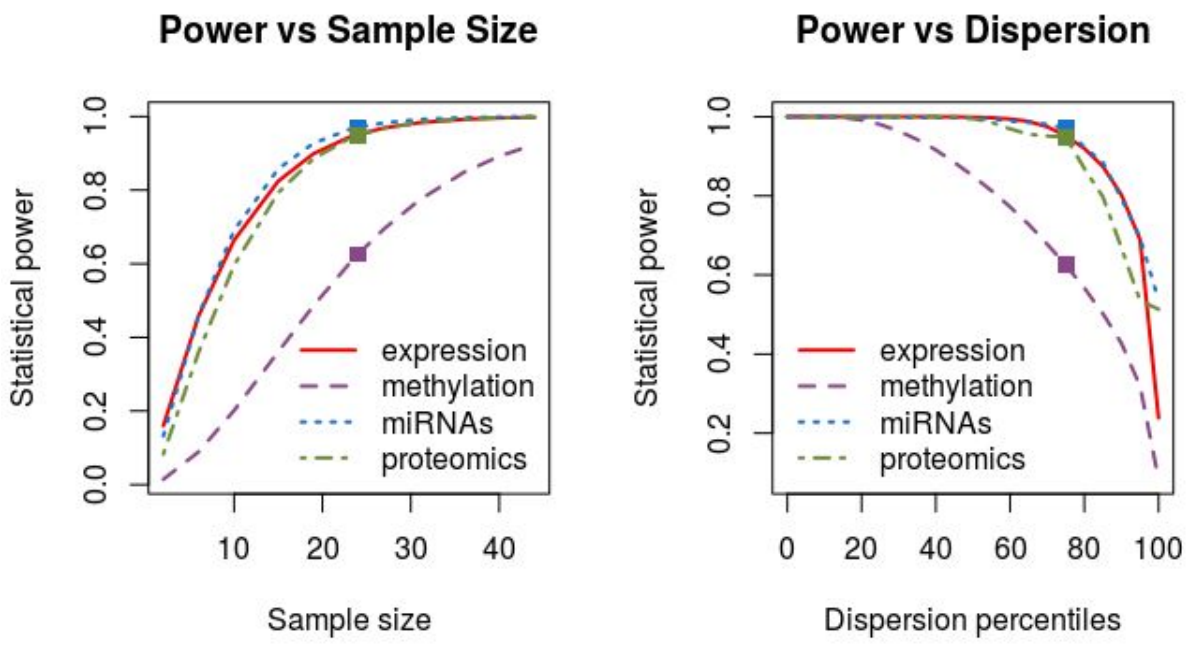
cost: See `cost` parameter.

summary: Table summarizing MultiPower results (see an example below). The columns are: the omic data names (omic), the omic data types (type), the number of omic features for each omic (numFeat), the expected proportion of DE features set by the user (DEperc), the initial Cohen's d set by the user (CohenD), the estimated delta (Δ) value, which is the absolute difference of group means for normal data and the proportions per group in binary data, the estimated fold-change in count data (minFC), the estimated mean of counts for count data (meanCounts), the estimated dispersion (dispersion), the FDR value (FDR), the minimum power to be achieve for each omic (minPower), the average power to be achieved in the multi-omic experiment (averPower), the cost per omic (cost), the minimum sample size needed for each omic to achieve minPower (minSampleSize), the optimal sample size (optSampleSize), and the power at this optimal sample size (power).

	omic	type	numFeat	DEperc	CohenD	delta	minFC	meanCounts	dispersion	minPower	averPower	cost	minSampleSize
expression	expression	2	12042	0.64	0.8	0.7856	NA	NA	0.8034	0.6	0.8	1	9
methylation	methylation	2	384349	0.14	0.8	0.8347	NA	NA	0.9441	0.6	0.8	1	24
miRNAs	miRNAs	2	534	0.56	0.8	0.8688	NA	NA	0.7948	0.6	0.8	1	9
proteomics	proteomics	2	171	0.44	0.8	0.6798	NA	NA	0.6263	0.6	0.8	1	11

	optSampleSize	power
expression	24	0.9513
methylation	24	0.6250
miRNAs	24	0.9703
proteomics	24	0.9486

data2plot: Data generated to create the power plots that are also returned by the function (see example below).



These plots display the statistical power curve for each omic at different sample sizes (left) or at different dispersion values from the minimum to the maximum dispersion (right). The squares represent the values for the optimal sample size.

When optimal sample size is out of the budget

Sometimes, the optimal sample size computed by MultiPower can be prohibitive in terms of cost. What options do we have then if we do not want to sacrifice power? We can allow for a different number of replicates per omic (see next section) since the omics with higher power might require a lower number or replicates and then the cost of the experiment can be reduced at the expense of not having the ideal scenario to perform an integration analysis. A second option could be to accept a higher FDR, which would increase the number of false positives in the statistical analysis. And finally, the most recommendable option is to increase the effect size to detect. In this case, MultiPower package has the function `postMultiPower()` to find out which effect size can be detected for a given sample size and which the resulting power would be. These are the arguments of the function that can be used when the same effect size (d0) was set for all the omics:

data: As provided to MultiPower() function. List with as many elements as omic data types. The names of the omics should be the names of the list. Each element in this list must be a raw count data matrix, and in this case MultiPower will take into account the library sizes to estimate power; a normally distributed data matrix which must have been already pre-processed and normalized; or a binary data matrix (with 0/1 or TRUE/FALSE values). In any case, for each one of these

matrices, rows must correspond to omic features (genes, methylation sites, ChIP-seq peaks, etc.) and columns to observations (biological samples, patients, etc.).

groups: As provided to MultiPower() function. List with as many elements as omic data types. The names of the omics should be the names of the list. Each element in this list must be a vector with length equal to the number of observations for that omic in `data` argument. Each element of this vector must indicate the experimental group where each observation belong. Only two experimental groups are allowed.

optResults: Object returned by MultiPower() function.

max.size: Maximum sample size allowed by the user. It will be used to determine the effect size that can be detected (by default, 5).

omicCol: The color that will be used to plot each omic. It must be a vector with length equal to the number of omics. If it is NULL (default), default colors are used.

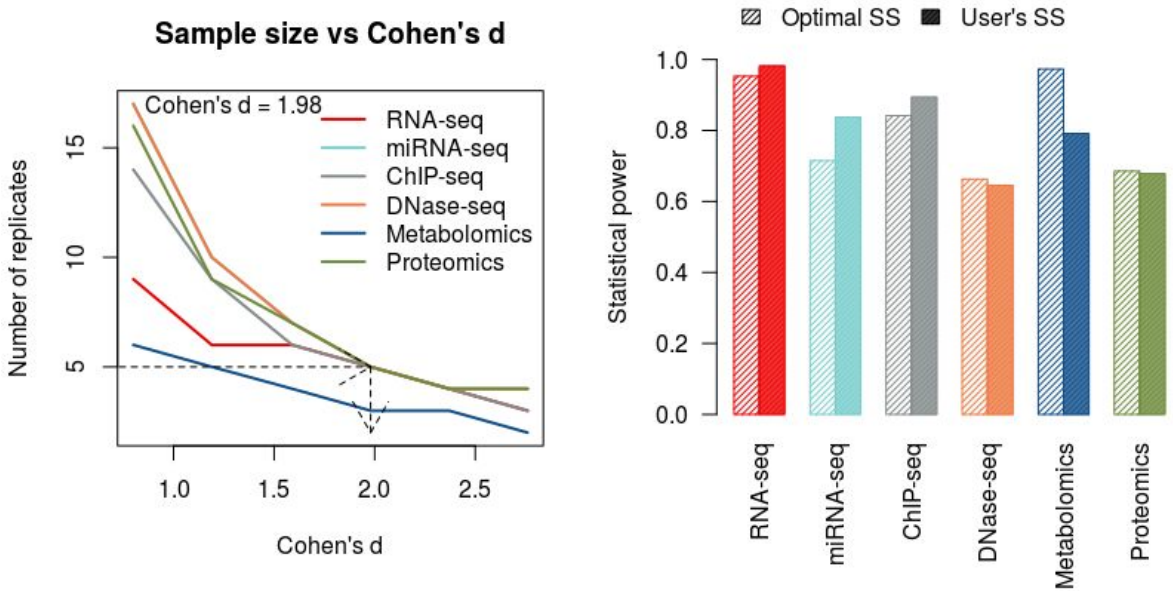
The `postMultiPower()` function returns a list with the following elements:

SampleSize: Matrix containing the optimal sample size for each omic data type (in rows) and for different values of Cohen's d (in columns).

Power: Matrix containing the statistical power at the optimal sample size for each omic data type (in rows) and for different values of Cohen's d (in columns).

d: Values of Cohen's d for which the optimal sample size has been estimated. These are the values represented in the columns of the above matrices.

In addition, a plot is displayed (see an example below) that summarizes these results.



In this example, a different sample size was allowed for each omic. That is why we have a different line for each omic in the left plot. The maximum sample size was set to 5 and the effect size to be detected at this sample size is given by a Cohen's d of 1.98. In this case, we can observe that the required sample size for Metabolomics would be lower than 5. The right plot compares the effective power at the optimal sample size and with the power at the selected sample size (n=5).

Optimal sample size estimation without pilot or previous data

When multi-omic pilot data sets are not available, users must set the values for the input parameters and then use the `optimalRep()` function, which requires the following arguments:

parameters: List with as many elements as omic data types. For each omic, each element of this list is another list containing the different parameters needed to compute power which, in this case, must be set by the user. See next section for more details.

omicPower: The minimum power that must be achieved for each omic. It must be a vector with a length equal to the number of omics. If it is a single number, this same number will be used for all the omics. By default, omicPower = 0.6.

averagePower: The minimum average power that must be globally achieved. By default, averagePower = 0.85.

fdr: False Discovery Rate level to be used. It is the significance level after multiple testing correction. By default, fdr = 0.05. If no multiple testing correction is to be applied, this argument must be set to NULL and then alpha argument is required.

alpha: Significance level to be used only when multiple testing correction is not to be applied (fdr = NULL). By default, alpha = 0.05.

cost: The cost to generate a replicate (a sample) for each omic. It must be a vector with a length equal to the number of omics. If it is a single number, this same number will be used for all the omics. This argument will only be used when a different sample size per omic is allowed. By default, cost = 1 (which means that all the omics will be assumed to have the same cost).

equalSize: If TRUE (default), the same optimal sample size will be estimated for all the omics. If FALSE, omics are allowed to have different sample sizes.

max.size: Maximum allowed sample size. By default, max.size = 30.

Setting input parameters

When no data sets are available, users must create an R list containing the input parameters required to estimate power for each omic: `type`, `allDispersions`, `dispersion`, `p1`, `d`, `delta`, `minFC`, `meanCounts`, `m`, `m1`, and `alld`. The descriptions for all these elements can be found in the previous sections. Here we provide an example of the R code that can be used to define the parameters object for an experiment with three omics. Please do not consider the parameter values in this example as suggested realistic values, you should carefully decide which are the most appropriate values for your experiment. Remember that values for the parameters `dispersion`, `delta`, `minFC` or `meanCounts` are related only to DE features. The parameters `allDispersions` and `alld` can be set to NULL because they are not really used to compute the optimal sample size.

1) Define the omics and the parameters for each one of them:

```
myomics = c("RNA-seq", "miRNA-seq",  "RRBS-seq")
mytype = c(1, 1, 2)
sdValue = c(2.5,3.7,0.7)  # pooled standard deviation
p1omics = c(0.4, 0.2, 0.1) # expected proportion of DE features
d0 = 1
minFC0 = c(1.5, 1.5, NA) # fold-change to be detected in DE features
```

```
meanCounts0 = c(20, 15, NA) # mean counts of DE features
M = c(12000, 500, 600000)  # number of features
```

2) Create the parameters object needed by the `optimalRep()` function:

```
myparam = vector("list", length = 3)  # define parameters object
names(myparam) = myomics
for (i in 1:length(myparam)) {
  if (mytype[i] == 1) {
    disp = max((sdValue[i]^2 - meanCounts0[i]) / meanCounts0[i]^2, 0.0001)
  }
  if (mytype[i] == 2) { disp = sdValue[i] }
  myparam[[i]] = list("type" = mytype[i], "allDispersions" = NULL,
                      "dispersion" = disp,"p1" = p1omics[i], "d" = d0,
                      "delta" = d0*sdValue[i], "minFC" = minFC0[i],
                      "meanCounts" = meanCounts0[i], "m" = M[i],
                      "m1" = round(M[i]*p1omics[i], 0), "alld" = NULL,
                      "dispPerc" = 75, "w" = 1)
}
```

Estimating the optimal sample size

The following R code is an example on how to run the `optimalRep()` function once the parameters are set, how to retrieve the results, and how to summarize them in a table:

```
optimalSS = optimalRep(parameters = myparam, omicPower = 0.6,
                       averagePower = 0.8, fdr = 0.05, alpha = 0.05,
                       cost = 1, equalSize = TRUE, max.size = 30)
optimalSS$n  # optimal sample size
powerSummary(parameters = myparam, optimalSampleSize = optimalSS)
```

When no data are being used to estimate power parameters, we can only generate one plot to study the statistical power versus the sample size. It is not possible to obtain the plot of the power versus the dispersion because we do not have the dispersion percentiles.

```
mycol = c("red2", "darkslategray3", "orchid4")
names(mycol) = myomics
plotData = powerPlot(parameters = myparam, optimalSampleSize = optimalSS,
                     omicCol = mycol)
```

Power study for available sample size

When users need to perform a power study on an available multi-omic data set with a fixed sample size and a given effect size, it must be taken into account that the statistical power, in this case, strongly depends on the data variability. To perform this study, `MultiPower()` function must be first applied in order to get the parameters needed to compute power estimated. These parameters are then used for the `PowerDispersionPlot()` as follows:

```
PowerDispersionPlot(n = 5, parameters, fdr = 0.05, alpha = 0.05,
                    omicCol = NULL)
```

Input parameters

n: Sample size available. It can be an integer if the same number of replicates is available for all the omic data types or a vector if different. By default, n = 5.

parameters: List with as many elements as omic data types. For each omic, each element of this list is another list containing the different parameters needed to compute power returned by MultiPower() function. For example, if *myresults* is the object returned by MultiPower(), *myresults$parameters* is the object needed for this argument.

fdr: False Discovery Rate level to be used. It is the significance level after multiple testing correction. By default, fdr = 0.05. If no multiple testing correction is to be applied, this argument must be set to NULL and then alpha argument is required.

alpha: Significance level to be used only when multiple testing correction is not to be applied (fdr = NULL). By default, alpha = 0.05.

omicCol: The color that will be used to plot each omic. It must be a vector with length equal to the number of omics. If it is NULL (default), default colors are used.

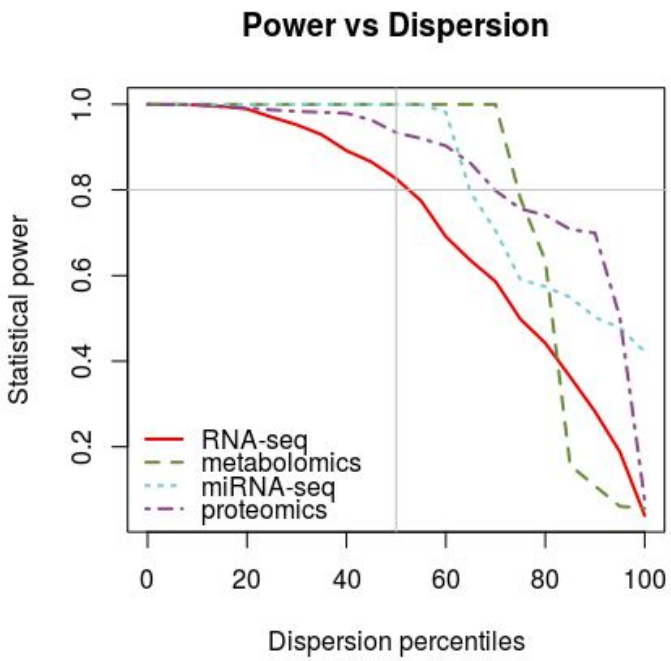
Interpretation of results

The `PowerDispersionPlot()` function generates a plot like the one below and the interpretation is similar to the plot returned by the `MultiPower()` function and explained in previous sections. The grey guide lines indicating the median dispersion and a power of 0.8 have been added with the following code. In this case, no multiple testing correction was applied.

```
PowerDispersionPlot(n = c(6,7,6,7), parameters = myresults$parameters,
                    fdr = NULL, alpha = 0.05, omicCol = NULL)
```

```
abline(v = 50, col = "grey")

abline(h = 0.8, col = "grey")
```



MultiPower for more than two groups

When more than two experimental groups are to be studied, and pilot or public data are available for all these groups, users could apply MultiPower function as many times as different pairwise comparisons need to be done. In order to facilitate this task, the `MultiGroupPower()` function has been included in the package. This function will apply MultiPower for all the pairwise comparisons required by the user or for all the possible comparisons if no comparisons are specified. The individual MultiPower results for each comparison are provided in a list and, in addition, also a global summary is displayed. In this summary, the optimal sample size is computed as the maximum sample size required in all the comparisons to reach the desired power. Please note that if a different sample size were allowed for each group, a different (and smaller) optimal sample size would be obtained for some groups. However, this option is not available in the package right now since it returns the same sample size for all groups.

The `MultiGroupPower()` function can be used as follows:

```
MultiGroupPower(data, groups, type, comparisons = NULL, d0 = 0.8, p1 = 0.2,
omicPower = 0.6, averagePower = 0.85, dispPerc = 75, fdr = 0.05,
alpha = 0.05, cost = 1, equalSize = TRUE, max.size = 200, omicCol = NULL,
powerPlots = FALSE, summaryPlot = TRUE)
```

Most of the arguments of the `MultiGroupPower()` function were already described in detail in the MultiPower section. Next, we describe those that are specific of MultiGroupPower or that may have a different format.

comparisons: Pairwise comparisons to be done between groups. If NULL (default option), the function will generate all the possible comparisons between the groups that are available for all omics. If users wish to indicate the comparisons to be done, they must provide a matrix with two rows and as many columns as comparisons. Each column will be then a two-element vector with the two groups to be compared. An easy way to generate this matrix is using the `combn()` function that returns a matrix with all the possible comparisons. Users can then remove the columns of the comparisons that are not interesting for them.

p1: The expected proportion of DE features for each comparison. If it is a single value, the same expected DE proportion will be used for all omics and comparisons. If it is a vector, it must have a length equal to the number of omics and the same vector will be used for each comparison. If it is a matrix, it must have as many rows as omic data types and as many columns as comparisons. The comparisons must be in the same order as in the comparisons argument. Please note that if comparisons = NULL, the order in both matrices could not coincide and therefore wrong results could be obtained. By default, p1 = 0.2.

powerPlots: If TRUE (FALSE is the default), power plots will be generated for each individual comparison as in MultiPower function.

summaryPlot: If TRUE (default), summary plots for sample size and power will be generated including the results for all comparisons and the global result, that is, the maximum sample size for all comparisons (“optimal” sample size) and the corresponding statistical power for each omic.

Interpretation of results

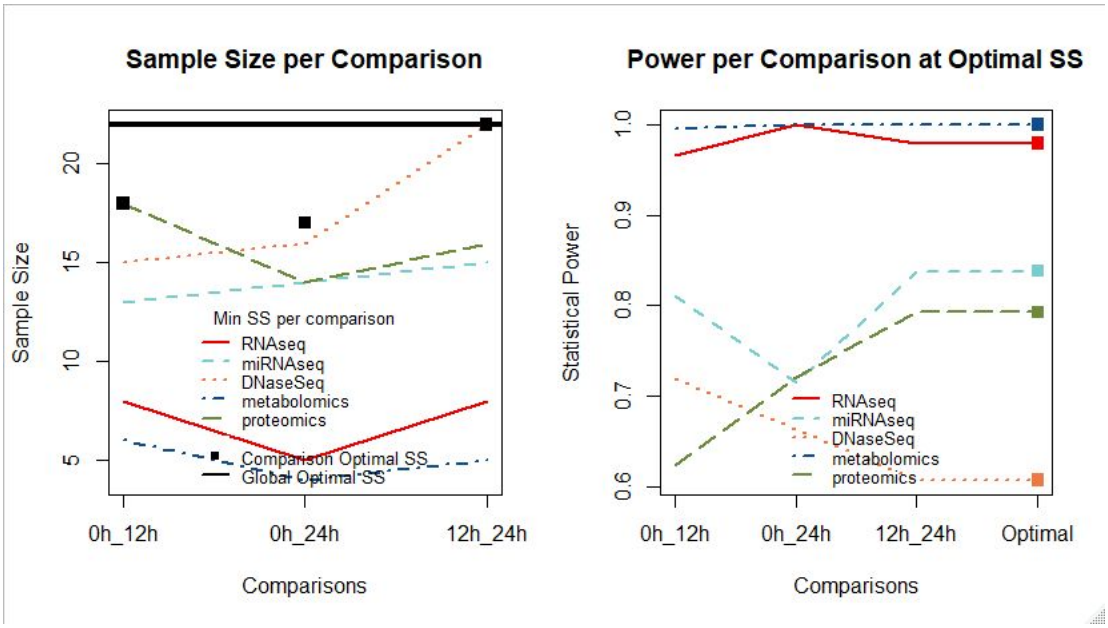
When applying `MultiGroupPower()` function, the result is a list containing as many elements (lists) as the number of comparisons and an additional element with the global summary of the results. The list obtained for each individual comparison is the same as the one returned by the MultiPower function (see the corresponding section), while the GlobalSummary element contains the following information:

omic	type	numFeat	DEperc	CohenD	delta	minFC	meanCounts	dispersion	minPower	averPower	cost
RNAseq	2	12762	0.4	0.8	0.4672-0.6074	NA	NA	0.3163-0.3892	0.6	0.8	1
miRNAseq	2	469	0.2	0.8	0.502-0.5582	NA	NA	0.4572-0.4761	0.6	0.8	1
DNaseSeq	2	52788	0.2	0.8	0.1662-0.5143	NA	NA	0.193-0.4882	0.6	0.8	1
metabolomics	2	60	0.6	0.8	0.6541-1.2028	NA	NA	0.3784-0.619	0.6	0.8	1
proteomics	2	1077	0.2	0.8	0.755-1.1722	NA	NA	0.7315-1.2018	0.6	0.8	1

minSampleSize	optSampleSize	power
5-8	22	0.9791
13-15	22	0.8378
15-22	22	0.6074
4-6	22	0.9999
14-18	22	0.7929

This summary table is very similar to the summary table of an individual pairwise comparison. The only difference is that some columns as the expected proportion of DE features set by the user (DEperc), the estimated delta (Δ) value, the estimated fold-change in count data (minFC), the estimated mean of counts for count data (meanCounts), the estimated dispersion (dispersion) or the minimum sample size needed for each omic to achieve minPower (minSampleSize) may contain either a single value or a range of values when the value was different at each comparison.

In addition to this summary table, the following summary plots are also generated when the summaryPlot parameter is set to TRUE:



The sample size plot (left plot) is slightly different when equal or different sample size is allowed for each group. When equal sample size is required (example above), the color lines show the minimum sample size required for each omic at each comparison, while the black square is the optimal sample size at each comparison and the black line the global optimal sample size (the maximum of them). The power plot displays the statistical power for each omic at the optimal

sample size. This information is given for each comparison and for the global optimal sample size (squares).

This plots can also be generated from the object returned by `MultiGroupPower()` function by using the `MultiCompaPlot()` function.

How to cite MultiPower

Soon in bioRxiv !!

References

[1] Cohen, Jacob. Statistical power analysis for the behavioral sciences. Hilsdale. NJ: Lawrence Earlbaum Associates 2 (1988).
[2] Sawilowsky, S. S. (2009). New effect size rules of thumb. Journal of Modern Applied Statistical Methods, 8(2), 597 – 599.