shinyÉPICo: the user's guide

Octavio Morante-Palacios[*1] ***and Esteban Ballestar***[†1]

[1]Epigenetics and Immune Disease Group, Josep Carreras Research Institute (IJC), 08916 Badalona, Barcelona, Spain and Germans Trias i Pujol Research Institute (IGTP), 08916 Badalona, Barcelona Spain

[*]omorante@carrerasresearch.org [†]eballestar@carrerasresearch.org

17 November 2020

Package

shinyepico 0.99.5

Contents

1 Introduction: What is ShinyÉPICo for?

shinyÉPICo is a web application based on shiny and created to accelerate and to make easier the study of Illumina Infinium DNA methylation arrays, including the 450k and EPIC. For this purpose, shinyÉPICo uses, among others, the widely used minfi (Aryee et al. 2014) and limma (Ritchie et al. 2015) packages for the array normalization and the differentially methylated positions (DMPs) calculation, respectively. Moreover, shinyÉPICo also allows calculation of differentially methylated regions calculation (DMRs), based on the mCSEA package (Martorell-Marugán, González-Rumayor, and Carmona-Sáez 2019).

It is conceived as a 'graphical pipeline' since, throughout the analysis, you can observe the result of the different options with multiple charts, and interactively modify decisions in current or prior steps. In addition, the application automates calculations that would otherwise take much more time and effort.

shinyÉPICo is a fully graphical application and the calculations performed in R are done entirely on the server-side. Therefore, the application can be used both on the local computer and through a web server, to which devices, such as computers, smartphones or tablets could be connected, without high RAM or CPU requirements and only a web browser as requirement.

The steps followed by the application includes the data importing, starting with iDAT files (Raw Illumina DNA methylation datasets), array normalization, quality control, and data exploring, and DMP and DMR calculation. Each step has multiple options and charts that we will describe along with this manual.

1.1 What do I need to use ShinyÉPICo?

ShinyÉPICo can run in GNU/Linux, Windows, or macOS. The package dependencies are automatically tried to install with the package is installing.

Since the application allows to follow interactively all the analysis process, many objects have to be stored in RAM memory. Therefore, the application can be memory demanding, especially when trying to analyze a large number of samples at the same time. We recommend at least 12GB of RAM for smooth use of the application, but depending on the number of samples analyzed and whether they are EPIC or 450k, the needs may be lower or higher.

1.2 How can I install ShinyÉPICo?

At the moment, you can install the application through github using the 'remotes' R package:

```
install.packages("remotes")
library("remotes")
install_github("omorante/shinyepico", upgrade="always", dependencies = TRUE)
```

1.3 Dependencies and implementation

ShinyÉPICo implements internal functions to follow the array normalization, and the DMP/DMR calculations, including, for example, the differential of beta values calculation between the groups. For the main features of the application, the package dependencies are the following:

- Application interface: **shiny**, **shinyWidgets**, and **DT**.
- Array reading and normalization: **minfi**
- Linear model generation and DMP calculation: **limma**
- DMR calculation and genomic graph: **mCSEA**
- Heatmaps: **gplots::heatmap.2** and **heatmaply**
- Other charts: **ggplot2** and **plotly**

The complete list of the application dependencies are:

```
R (>= 4.0.0),
DT (>= 0.15.0),
data.table (>= 1.13.0),
doParallel (>= 1.0.0),
dplyr (>= 1.0.0),
foreach (>= 1.5.0),
GenomicRanges (>= 1.38.0),
ggplot2 (>= 3.3.0),
gplots (>= 3.0.0),
heatmaply (>= 1.1.0),
limma (>= 3.42.0),
minfi (>= 1.32.0),
plotly (>= 4.9.2),
reshape2 (>= 1.4.0),
rlang (>= 0.4.0),
rmarkdown (>= 2.3.0),
rtracklayer (>= 1.46.0),
shiny (>= 1.5.0),
shinyWidgets (>= 0.5.0),
shinycssloaders (>= 0.3.0),
shinyjs (>= 1.1.0),
shinythemes (>= 1.1.0),
statmod (>= 1.4.0),
tidyr (>= 1.1.0),
zip (>= 2.1.0)
```

For the DMR calculation, additional suggested packages are required. You can use the application without installing this package, but you will not able to calculate DMRs. Moreover, for the array normalization, you will need to install annotation and manifest packages, for 450k or EPIC.

```
mCSEA (>= 1.10.0),
IlluminaHumanMethylation450kanno.ilmn12.hg19,
IlluminaHumanMethylation450kmanifest,
IlluminaHumanMethylationEPICanno.ilm10b4.hg19,
IlluminaHumanMethylationEPICmanifest
```

And, to run the application:

```
library("shinyepico")
set.seed(123)
run_shinyepico()
```

The function run_shinyepico has 4 parameters that can be modified:

- **n_cores:** The application is partially compatible with parallel computing. This numeric parameter controls the number of cores and, by default, it is half of the detected cores. If you have limited RAM (8GB or less) we recommend setting this to 1, in order to avoid the RAM overhead of multicore calculations.
- **max_upload_size:** By default, shiny applications have an upload limit (in MB), useful when the application is running on a web server. By default, this parameter is 2000MB.
- **host:** IP used to deploy the application. By default, this parameter is your local IP (127.0.0.1) which means that only you, from your computer, will have access to the application. However, it is possible to make the app reachable to other computers in the same LAN by changing the IP to 0.0.0.0.
- **port:** Port used to deploy the application. By default, a random free port.

As can be seen in the example, we used the set.seed function before starting ShinyÉPICo. We recommend always setting the same seed, which can be any number, to avoid uncertainties and ensure that the results obtained are reproducible. Particularly, DMRs calculation, that rely on the mCSEA package, utilizes permutations to estimate the p.value. For that reason, some uncertainty is expected, and results can be a bit different every time you run the application. Using a seed avoids this problem, as you will obtain always the same result with the same seed.

2 ShinyÉPICo workflow

shinyÉPICO workflow is divided into tabs that summarize the different steps to follow:

- **Data import**
- **Quality control and array normalization**
- **Differentially Methylated Positions calculation**
- **Calculation of Differentially Methylated Regions**
- **Data export**

While it is possible to go back to previous steps and change options at any point in the process, some buttons are disabled until the steps required to perform them are completed. For example, you will not be able to generate a heatmap if you do not complete the normalization of the arrays and the calculation of the DMPs. This design is intended to avoid errors and guide the user through the analysis steps.

In the next sections, we will discuss the different steps, the options, and chart interpretation. Moreover, we will use a public DNA methylation array dataset to show an example of application use.

3 Using ShinyÉPICO: an explanation of the options and an example with real data

In the following example, iDATs from a study using Illumina EPIC DNA methylation arrays will be used to illustrate the steps (Li et al. 2020). Although shinyÉPICo can be used with multiple groups and large cohorts, and in that case, iteration and automation are more useful, for this manual we have decided to use a minimal example that can be easily reproduced on almost any computer. Specifically, monocyte and monocyte-derived macrophages (produced with M-CSF) has been selected, only 6 samples (3 monocyte samples and 3 macrophage samples from 3 different healthy donors).

The .zip file of this dataset can be found in the Github repository

In this example, the seed 123 has been set using the function set.seed() before run_shinyepico().

3.1 Data Import and Sample Selection

The first step in the shinyÉPICo workflow is to prepare the data in the proper format. iDAT files should be compressed in a .zip file. The name of the files should follow the standard convention: **XXXXXXXXXXX_YYYYYY_ZZZ.idat** being XXXXXXXXXXX the Sentrix_ID, YYYYYY the Sentrix_Position, and ZZZ Grn or Red (corresponding, respectively, to the Red and Green signal file).

Moreover, a CSV (comma-separated) file with the annotation of the experiment should be included. It is mandatory to include the Sentrix_ID and Sentrix_Position columns that allow the software to find their respective iDAT files. Moreover, other columns should be added to reflect the different variables (e.g. sample name, health/disease, treatment/control, age, sex, hybridization day, etc.). Usually, iDATs and sample sheet have these features by default, and no additional work is required.

In this regard, 3 parameters are required in the Input tab to continue the analysis:

- A column with the **sample names**, that should be unique, without duplicates.
- A column with the **variable of interest**, in which sample groups will be found to calculate DMPs and DMRs.
- Optionally, a **donor variable** is also included in the Input tab. If you have an experiment where several samples come from the same donor, it is recommended to add a column with this information and select it in the form. This information is used in the SNPs heatmap of the Normalization section (as we will discuss later), and also for the DMPs/DMR calculation. If you select a valid donor variable, this information is automatically added as covariable of the linear model in the DMPs section, and, therefore a "**paired analysis**" will be performed. If you do not have or you would not like to use this information, you can select the same column as sample names.

An aspect to take into account is that shinyÉPICo autodetects the variable types (numerical or categorical) depending on whether or not they can be coerced to numbers. For this purpose, variables are coerced to a numeric vector, and when the generated NAs are less than 75% of the total, the variable is set as numeric, and, otherwise, as categorical. Moreover, not informative categorical variables are excluded (e.g., a variable with the same value in all the samples, or with not equal values). Therefore, numbers should not be used for categorical

variables in the sample sheet. Numerical variables with some NAs but less than 75% can be used in the exploratory analysis but not as covariables of the linear model, since Limma needs a design matrix without missing values.

In the next table, you can see the sample sheet of the example dataset:

Sample_Name	Sample_Well	Sample_Plate	Sample_Group	Donor	Pool_ID	Sentrix_ID	Sentrix_Position
MAC A	B05	SMET0256	MAC	A	Epic	202163550095	R02C01
MO B	D05	SMET0256	MO	B	Epic	202163550095	R04C01
MAC C	F05	SMET0256	MAC	C	Epic	202163550095	R06C01
MO A	G05	SMET0256	MO	A	Epic	202163550095	R07C01
MO C	E06	SMET0256	MO	C	Epic	202163550097	R05C01
MAC B	G06	SMET0256	MAC	B	Epic	202163550097	R07C01

It includes the Sentrix_ID and Sentrix_Position mandatory columns, and also other columns relevant for the experiment such as Sample_name, Sample_group, and donor.

After selecting the proper column in each form field, and the samples to be included (in this example, all), it is possible to proceed with the next step, pressing the "Continue" button. When the application is performing an operation expected to take time, a progress bar is displayed at the bottom right of the window to inform the user that the application is busy.

Internally, shinyÉPICO uses the "**read.metharray.sheet**" and "**read.metharray.exp**" functions to read the sample sheet and load the DNA methylation data, respectively.

After loading the methylation data of the selected samples, the Normalization tab is automatically shown, where you can see an overview of the data quality and perform the array normalization.

3.2 Quality control charts

First, the quality control tab shows two useful charts to identify bad samples.

On the one hand, the **QC Signal plot** shows the median methylated (mMed) and unmethylated (uMed) of each sample array. When mMed or uMed of a sample is less than 10, it is considered "Suboptimal". Although this cutoff is arbitrary and it is the user who decides whether a sample is valid or not, a signal much lower than this threshold or very different from most samples indicates that there has been a problem with it. Depending on whether the signal is very far from this threshold and depending on the rest of the results, these types of samples should be excluded from the analysis. To exclude samples, you can go back to the Input tab to change the samples selected and load them again.

On the other hand, the **Bisulfite conversion plot** is calculated using the information of the bisulfite conversion II control probes of the 450k/EPIC arrays. When the bisulfite conversion reaction is successful, the probe of the control position will have intensity in the Red channel, whereas if the sample has unconverted DNA, the probe will have a high signal in the Green channel.

For each sample, we calculate all the Red/Green ratios for each control position, and the minimum ratio is shown in the chart. When a sample has a ratio lower than 1.5, we flag it as "Failed".

3.3 Array Normalization

A correct array normalization is essential for the results of the subsequent steps in the workflow. shinyÉPICo can use all the normalization methods available in the minfi package:

- **Raw**: Without normalization.
- **Illumina**: A reverse-engineered implementation of the Genome Studio normalization.
- **Funnorm**: A between-array normalization method that relies in data from control probes of the arrays. It performs also Noob normalization before the Functional Normalization.
- **Noob**: A within-array normalization method with dye-bias normalization.
- **SWAN**: A within-array normalization method that allows Infinium I and Infinium II probes to be normalized together.
- **Quantile**: A between-array normalization method that assumes no global differences in methylation between the samples. When global changes are expected, such as in cancer samples, other methods, such as Funnorm, are recommended.
- Additionally, it offers the option of performing Noob within array normalization followed by Quantile normalization (**Noob + Quantile**), analogously to the Funnorm method. This non-standard approach has empirically shown good results in our experience.

For details about the methods, the minfi documentation and the respective publications are good references. Although depending on the experimental design and the type of changes expected a normalization method can be expected to be better than another, usually the best method should be determined empirically among the valid options.

Additionally, shinyÉPICo offers three more options in the Normalization tab:

- **Drop CpHs**: When this option is selected, CH probes (non-CpG methylation) are removed. It uses the **minfi::dropMethylationLoci** function with standard parameters.
- **Drop SNPs**: When this option is selected, positions with SNPs annotated by Illumina are removed, with a MAF (minimum allele frequency) higher than the cutoff selected are removed. It uses the **minfi::dropLociwithSnps** function with standard parameters.
- **Drop X/Y Chr**: When this option is selected, all the positions of the sex chromosomes are removed. Since the methylation values of these chromosomes can be very different depending on the sex of the donors, removing them can help to homogenize the data. Alternatively, or additionally, sex information can be added to the linear model.

Moreover, Using the minfi recommended value, the genomic positions of the final obtained RGChannelSet object (Raw Data) are filtered by the detection p.value (using the minfi::detectionP function) in the normalized data, removing all the positions with an average p.value higher of 0.01.

shinyÉPICo makes it very simple to test different methods and options. When a normalization method is selected, clicking the "Select" button, the different charts are automatically generated.

In this example, we are going to select the Quantile normalization. We show different examples of the type of charts generated by the software.

3.3.1 Density plot

This chart shows the distribution of the beta values of every sample. A bimodal distribution is expected, with two peaks centered around 0 (unmethylated) and 1 (methylated) beta values. When more peaks are shown in the middle, or when the pattern of different samples is very different, it indicates problems in some step of the sample preparation or hybridization. After normalization, we expect an improved alignment of the sample patterns, with little discrepancies between them.

Raw:

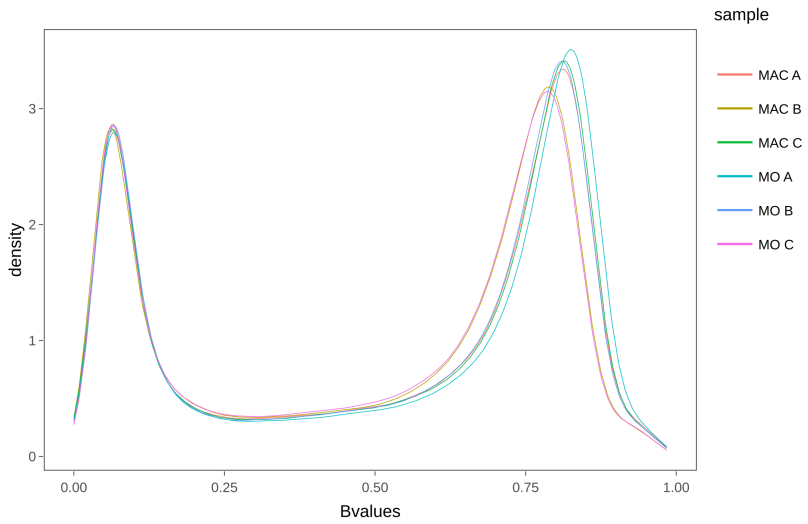


Figure 1: Density plot of raw beta values

Quantile:

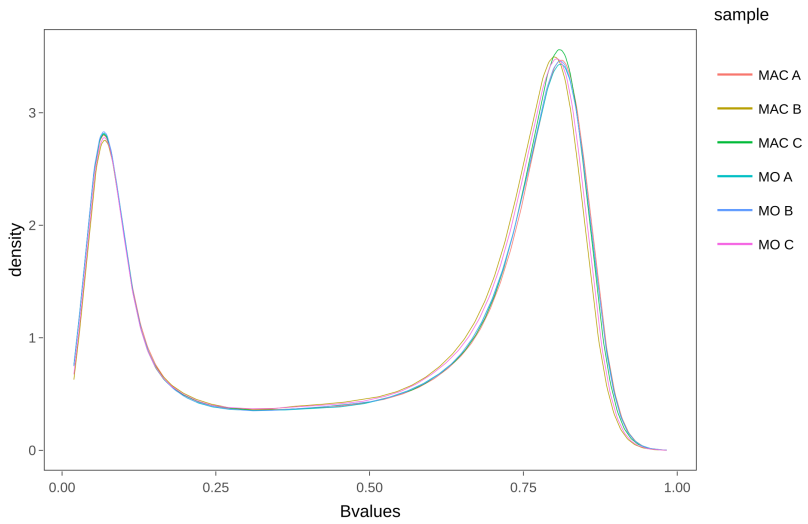


Figure 2: Density plot of quantile normalized beta values

3.3.2 Boxplot

This chart shows the box-and-whisker plot of beta values in each sample. The median and interquartile range of the samples should be similar, and more homogeneous after the normalization.

Raw:

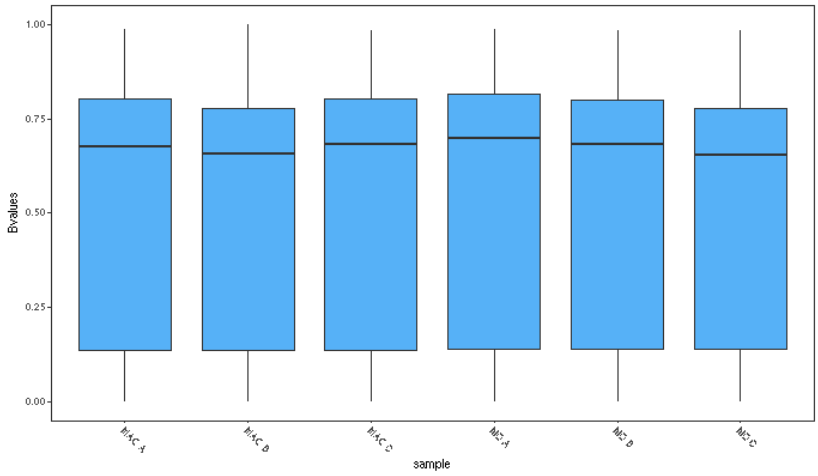


Figure 3: Boxplot of raw beta values

Quantile:

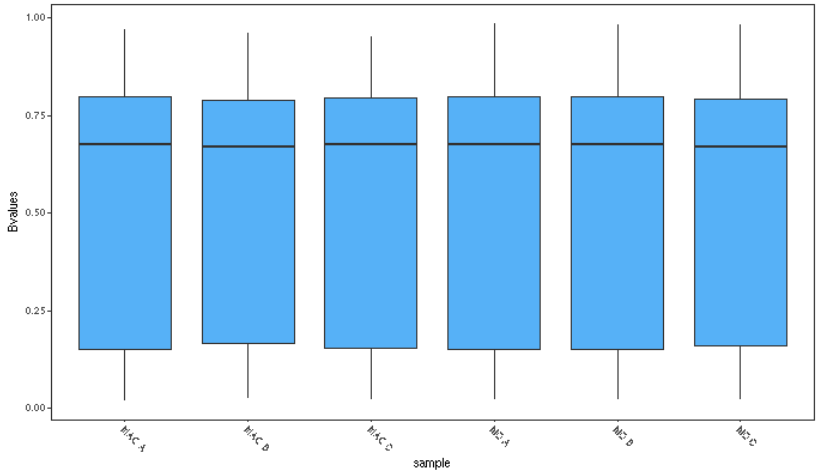


Figure 4: Boxplot of quantile normalized beta values

3.3.3 SNPs heatmap

This heatmap utilizes information of a special subset of single nucleotide polymorphism (SNP) probes in the 450k and EPIC arrays. For the representation, we use the beta values of these probes (extracted using the function minfi::getSnpBeta()). The heatmap shows a column clustering, that should correspond with the donor information. For example, in this experiment, with 3 different treatments and 3 donors, we can see a clear clustering by donor.

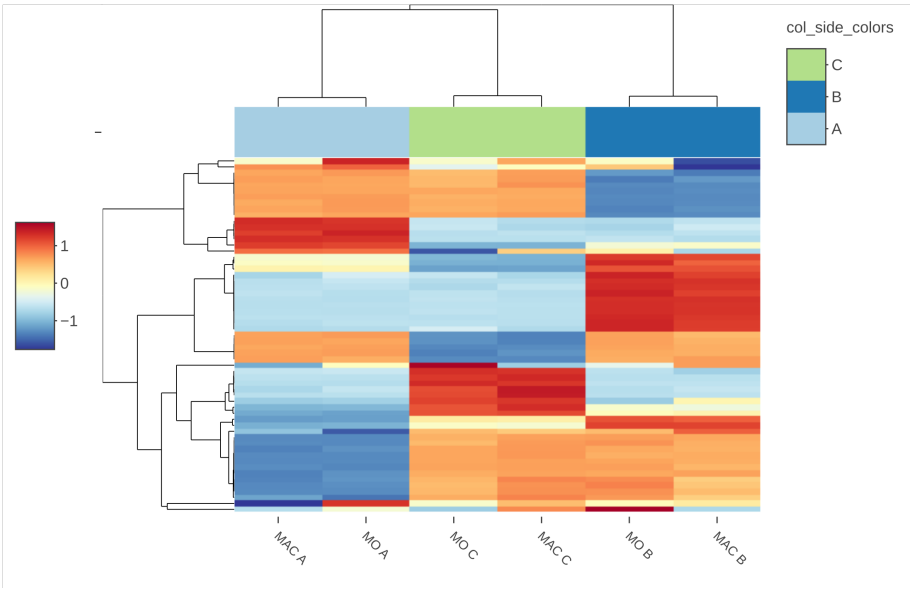
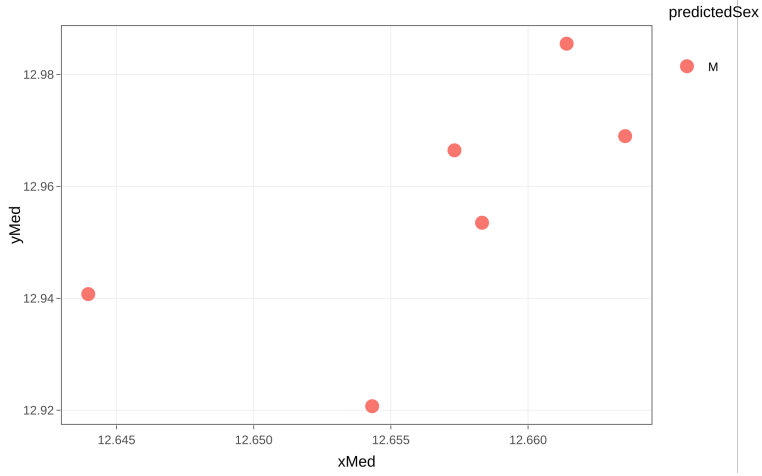


Figure 5: Heatmap of the SNP probes beta values

3.3.4 Sex prediction

Depending on the median X and Y chromosomes intensities, we show the sex prediction generated with the function minfi::getSex(). Note that, even if the Drop X/Y Chr. option is selected, this prediction is done before that step, and this graph will not be altered.

In this example, the three donors are males, as can be observed in the chart.



3.3.5 Exploratory analysis: PCA and Correlations

In order to get a first overview of the data before the DMP/DMR calculation, we provide two more charts.

- **PCA:**

First, we show the principal component analysis (PCA). Selecting specific principal components to plot in the x and y axis is possible, as well as changing the color variable. Moreover, a table showing the percentage of variance explained in each principal component is also depicted.

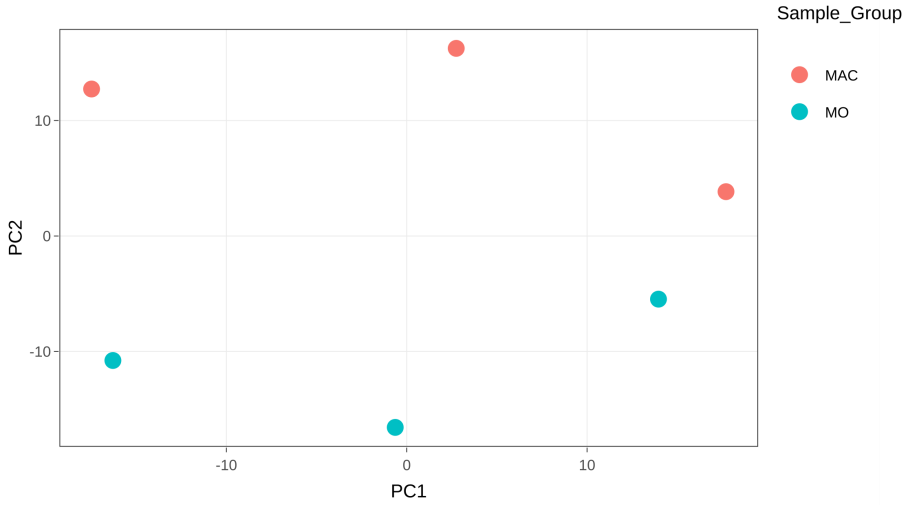


Figure 6: Principal Component 1 vs Principal Component 2

- **Correlations:**

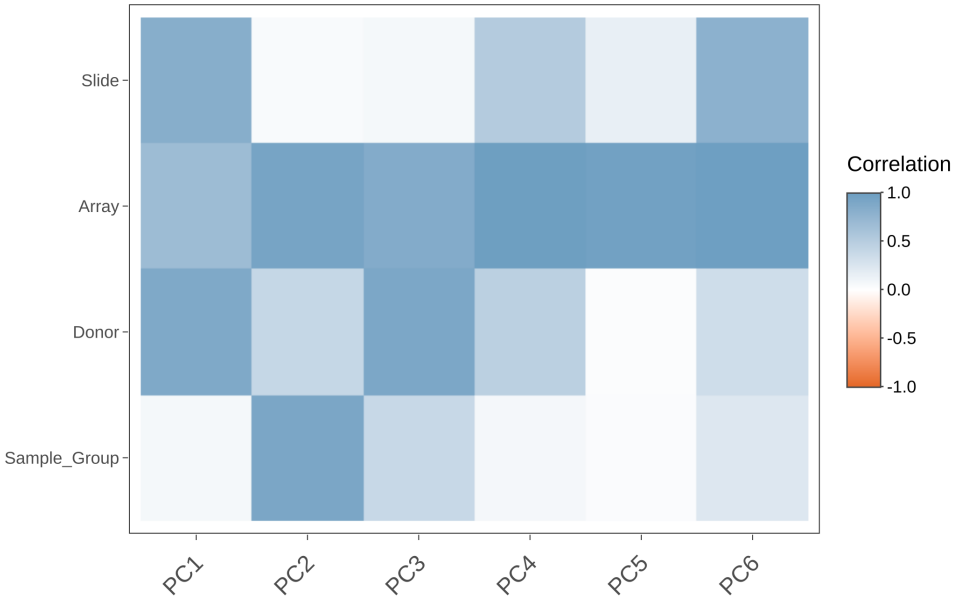
Secondly, we also provide a heatmap of the correlations between principal components and variables. As we explained before, variable types (categorical or numerical) are guessed, and not informative variables are excluded.

In the correlations tab, a detailed table of the variables is shown, including the type (categorical variables are shown as "factor", numerical variables as "numeric", and discarded variables as "discarded").

Pearson correlation is applied to correlate principal components with numerical variables. For categorical variables, linear models (principal component ~ categorical variable) are generated and R-squared statistics are shown in the representation. Alternatively, the p.values associated with these statistical approaches can also be plotted in a heatmap.

Both PCA and correlations plots can be also useful to find possible variables affecting methylation data, in order to select covariates and interactions for DMP/DMR calculation.

Correlation value:



P value:

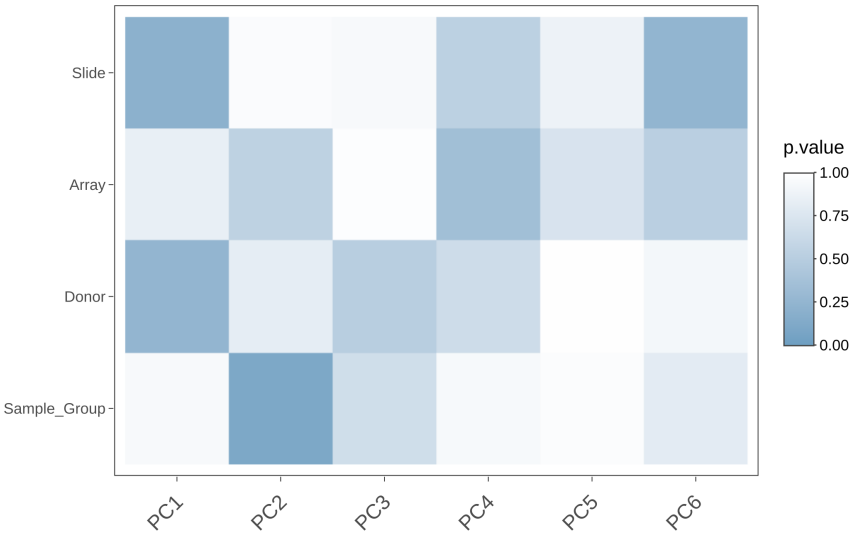


Figure 7: Correlation values of filtered variables with principal components

3.4 Differentially Methylated Positions (DMP) calculation

After normalizing the methylation data, the next tab, "DMPs", will become enabled. The process of DMP calculation is divided into two parts: model generation and contrasts calculation.

3.4.1 Model generation

In order to calculate DMPs between groups of the variable of interest, shinyÉPICo uses the limma package. First, shinyÉPICo fits a linear model using the **M values**, with the limma::lmFit function (see the limma user guide for more details.)

Beta values are used in all the charts of the application, because of its easier biological interpretation: Beta values range from 0 (totally unmethylated) to 1 (totally methylated).

However, for the statistical analysis with Limma, shinyÉPICo utilizes the M values (calculated as the logit of Beta values). M values range from -Infinite to Infinite.

Beta values are not as suitable to use with limma, because they have severe heteroscedasticity for highly methylated or unmethylated positions. For that reason, M values (homoscedastic data) are used for the limma model, whereas Beta values are used for representation (Du et al. 2010).

Several options can be set in this step:

- **Variable of interest**: This variable, that should be categorical, will be used to perform the contrasts to generate the DMPs.
- **Linear model covariables**: When other variables different from the variable of interest can be influencing the DNA methylation differences, they can be included in the linear model. Limma will take into account these differences and correct them in the result. For example, DNA methylation is very dependent on the donor, and, in experiments with several samples of the same donor, generally is recommendable to add the donor as a covariable. If you have selected a donor covariable in the Input tab, this option will be selected by default.
- **Linear model interactions**: When you suspect that a variable can be affecting DNA methylation depending on another variable (for example, changes of DNA methylation between the sex that are specific of age), an interaction term can be added to the model.
- **Array Weights**: If this option is enabled, estimated relative quality weights for each array are calculated and used in the limma model, using the function limma::arrayWeights. This option ponderates the influence of the arrays in the model depending on the calculated qualities. It is especially useful with large datasets of heterogeneous quality. Ritchie et al (2006)

When the linear model is generated, a diagnosis chart is plotted. It represents the square root of the standard deviation versus the average log expression of each position. We would expect that, as seen in the example, there is a flat relationship between both variables.

Moreover, the design matrix used to generate the model is also shown:

MAC	MO	DonorB	DonorC
1	0	0	0
0	1	1	0
1	0	0	1
0	1	0	0
0	1	0	1
1	0	1	0

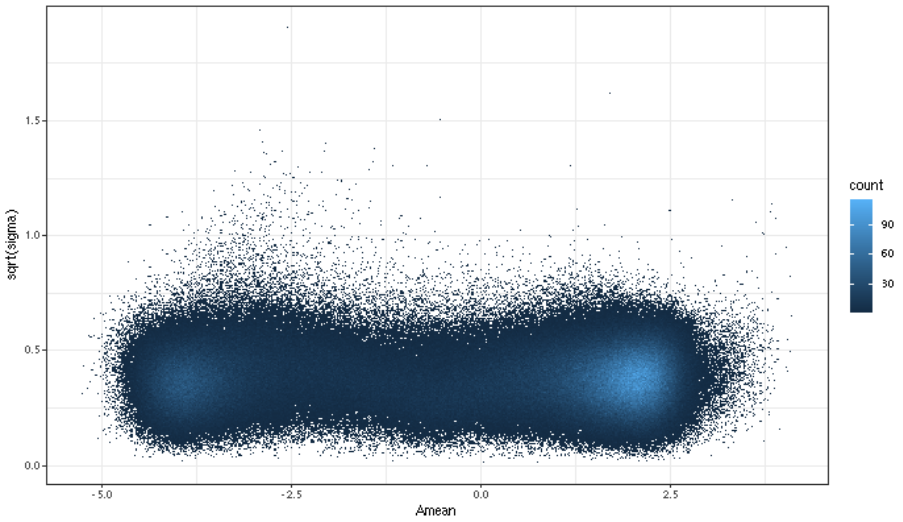


Figure 8: Mean vs Variance relationship of the linear model

3.4.2 Contrasts calculation

ShinyÉPICo autodetects all the possible comparisons between the groups found in the variable of interest. The number of pairwise combinations between the different groups is calculated as follows: n! / (2! (n — 2)!), where n is the number of groups.

Then, the application iterates over the contrasts and calculates the tables with the statistics produced by limma, using the functions limma::contrasts.fit, limma::eBayes and limma::topTable. Two more options can be set in this section:

- **eBayes Trend**: When this option is enabled, "an intensity-dependent trend is fitted to the prior variances s2.prior" (Law et al. 2014). This option is especially useful when a mean-variance relationship is observed in the prior plot.
- **eBayes Robust**: When this option is enabled, a robust empirical Bayes procedure is followed (Phipson et al. 2016). It can protect the empirical Bayes against hype-variable or hypo-variable positions.

Moreover, shinyÉPICo also calculates the differential of betas between groups, for each position. This information will be used in the next steps.

3.4.3 Heatmap customization

shinyÉPICo provides plenty of options to filter the statistics produced by limma and to generate a custom heatmap.

Filtering options

- **Min. DeltaBeta**: The threshold of the minimum differential of beta values between groups. By default, this value is 0.2.
- **Max. FDR**: The threshold of the maximum false discovery rate (adjusted p value). By default, this value is 0.05.
- **Max. p-value**: The threshold of the maximum unadjusted p value. By default, this value is 1.

With this options, a summary table with the DMPs found in each contrast is shown. In this example:

contrast	Hypermethylated	Hypomethylated	total
MAC-MO	2275	21	2296

The differential of beta is always calculated as the subtraction of the average value from the first group to the second group (in this example, MOavg - MACavg) and the positions are assigned to the "Hypermethylated" group if the differential is greater than 0, and to the "Hypomethylated" group if the differential is lower than 0.

Group options

- **Groups to plot**: Groups to be shown in the heatmap. When the column dendrogram is disabled, the order of the samples follows the order in this parameter. Drag and drop is enabled to reorder the groups in the form.
- **Contrasts to plot**: DMPs from the contrasts selected in this parameter are shown in the heatmap. Since several contrasts can share common DMPs, the heatmap shows only these common DMPs once, without duplicates.

Data options

- **Remove Batch Effect**: If this option is enabled, Beta values are modified with the function limma::removeBatchEffect and the information provided of covariables and interactions and the heatmap is generated with these values. For representation, this option can help to see a more clear pattern when technical variables, such as hybridization time, are affecting the results. Note that these corrected beta values are not used for any statistical test, and this option does not affect the differential of beta calculation neither.

Clustering options

- **Clustering algorithm**: Clustering algorithm to be passed to the hclust function. It includes single, average, complete, mcquitty, median and centroid algorithms. By default, it is average.
- **Distance Function**: Distance function to be used for the clustering. It can be Pearson correlation, Spearman correlation, Kendall or Euclidean. Whereas Pearson/Spearman correlation usually are useful distances when the scaling is performed, Kendall or Euclidean distance are more suitable when the heatmap is not scaled.
- **Scale**: This option indicates if scaling of the date should be performed. By default, a scaling of the data by row (position) is applied.

Additional options

- **Static Graph**: If this option is enabled (it is by default), the heatmap is plotted using the function gplots::heatmap.2 that produces a static image. If the option is disabled, the heatmap is plotted with the function heatmaply::heatmaply, which is less efficient and slower, but produce an interactive chart.
- **Column Dendro.**: This option indicates if columns (samples) should be ordered by the dendrogram (the default option) or if the columns should be ordered by group.
- **Column Colors**: When this option is enabled, a color legend of the different groups is shown above each column.
- **Row Colors**: When this option is enabled, row dendrogram is divided in the number of groups selected (k number) using the stats::cutree function. The resulting clusters are shown in different colors next to the row dendrogram.

Since heatmap plotting can be very time- and memory-consuming, we have fixed a limitation of 12.000 positions (rows) in the heatmap. If you try to plot more positions, you will see a message with this information. However, if you need a bigger heatmap, you can download the information to generate it in the Export tab, as we will discuss later.

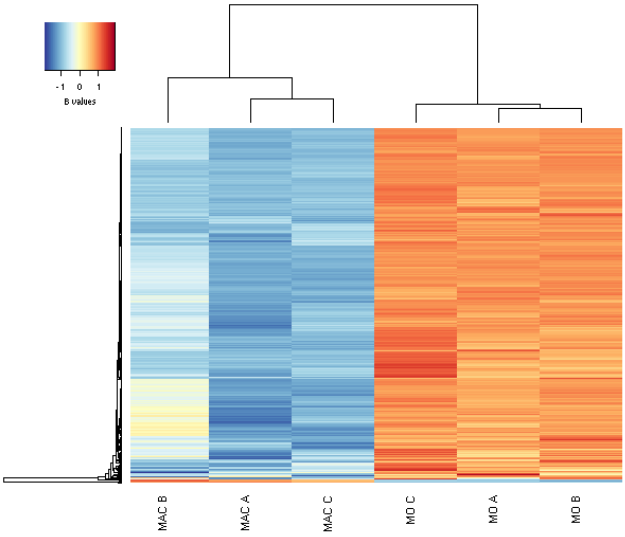


Figure 9: **DMP heatmap generated with default options**

3.4.4 DMPs Annotation

In the next tab of the DMPs section, DMPs annotation, the information about the genes associated with the DMPs is provided in a table. Moreover, the boxplot of a selected position can be plotted.

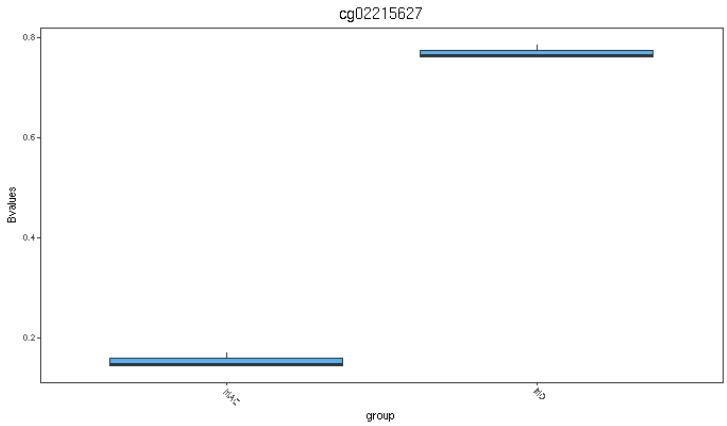


Figure 10: **DMP associated with the A2M gene promoter, found in the MAC vs MO contrast**
It shows demethylation in MAC.

The information on the table can be downloaded in CSV or XLSX (Excel) formats, using the buttons above it. However, only rows shown can be downloaded. It is necessary to display all the information in order to download it.

3.5 Differentially Methylated Regions (DMR) calculation

In addition to DMP calculation, shinyÉPICO also supports DMR calculation. A DMR is defined as a genomic region with differential methylation between groups. Instead of checking individual changes in genomic positions, DMR calculation involves the comparison of groups of genomic positions, which can be clusterized in several ways depending on the specific algorithm used.

DMR calculation of shinyÉPICO is based on the package mCSEA which searches DMRs in predefined regions: promoters, gene bodies and CpG islands.

mCSEA uses an algorithm based on the gene set enrichment analysis (Subramanian et al. 2005). Instead of genes, mCSEA uses a sorted list of genomic positions compared between two conditions.

Therefore, we use the eBayes result of limma of each contrast, sorted by the t-statistic, as the input of mCSEA. This implies that the parameters introduced in the limma model, including variables and covariates, are also taken into account for the calculation of DMRs, without the need of additional calculations.

3.5.1 mCSEA options

In the DMRs tab, some options can be set for the calculation generated by the mCSEA package:

- **Contrasts to calculate**: From all the contrasts possible for your variable of interest, you can select a specific subset.
- **Type of DMRs**: Among the three types of predefined genomic regions, you can select a specific subset.
- **Array platform**: Since the predefined genomic regions are dependent on the DNA methylation array used (450k or EPIC), you can select here the correct option. By default, this parameter is autodetected and set based on the input data.
- **Min. CpGs in DMR**: This parameter is a cutoff, that indicates the application to remove all the genomic regions with fewer positions than the minimum indicated.
- **Number of permutations**: mCSEA relies on permutations to estimate the p.values. Therefore, a higher permutation number will produce a more accurate result, but it would be more computationally demanding.

3.5.2 Heatmap customization

In the DMR section, shinyÉPICo provides the same options to filter the data and customize the data as in the DMP section.

Note that the default differential of beta threshold for DMRs is 0, in contrast with DMPs (0.2). This is justified considering that DMRs are calculated using the information of the methylation data of several positions, and that small changes may also be relevant if they are consistent. With this options, a summary table with the DMRs found in each contrast is shown. In this example:

contrast	Hypermethylated	Hypomethylated	total
MAC-MO	64	78	142
MAC-MO	160	111	271
MAC-MO	45	62	107

In order to calculate the differential of beta for DMRs, all the positions in DMRs are considered, and the average difference is used. The differential of beta is always calculated as the subtraction of the average value from the first group to the second group (in this example, MOavg - MACavg) and the positions are assigned to the "Hypermethylated" group if the differential is greater than 0, and to the "Hypomethylated" group if the differential is lower than 0.

All the options of the DMP heatmap are also in the DMR heatmap. The explanation of the options can be found in the DMPs section.

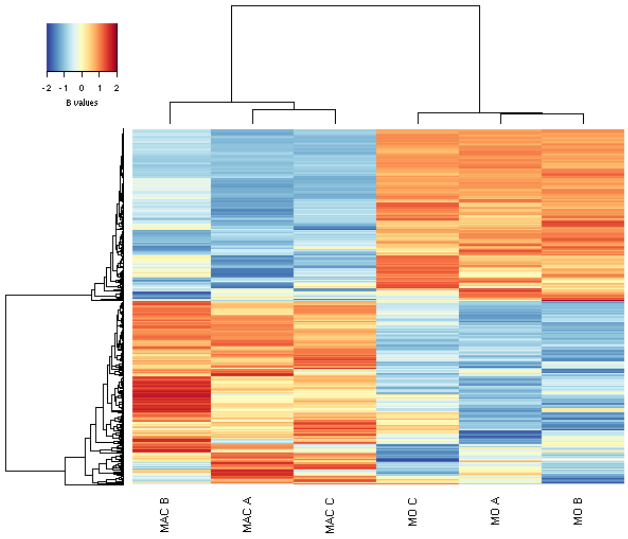


Figure 11: DMR heatmap generated with default options

3.5.3 Single DMR plot

In the Single DMR plot tab, a table with the DMRs of a selected contrast and region is shown. A DMR can be selected to be plotted in its genomic context.

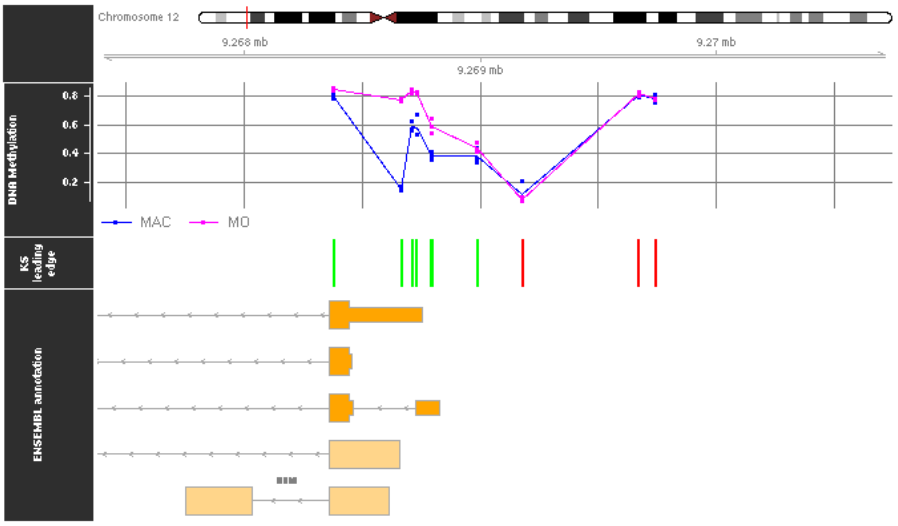


Figure 12: Genomic plot of the A2M promoter DMR
It shows demethylation in MAC

3.6 Exporting results

In the Export tab, shinyÉPICo allows the user to download the results of the analysis. Initially, all the download buttons are disabled, and they only become enabled when the information to generate the objects is available.

3.6.1 R Objects

Up to 9 R Objects can be download, selecting them in the form. They are generated in RDS format, using the saveRDS function. Therefore, they can be easily read in R using the function readRDS. We are going to describe the objects:

- **RGSet**: An object of the RGChannelSet class (see the minfi vignette) with the raw information of the loaded arrays.
- **GenomicRatioSet**: An object of the GenomicRatioSet class (see the minfi vignette) with the normalized information of the loaded arrays.
- **Fit**: An MArrayLM object generated using the limma::lmFit function. It contains the linear model used to calculate DMPs.
- **Design**: A matrix with the design information, with the variable of interest, covariates and interactions.
- **Ebayestables**: A named list with dataframes including the information of the limma statistics, one for each contrast.
- **Bvalues**: A dataframe with the normalized beta values of the experiment.
- **Mvalues**: A dataframe with the normalized M values of the experiment.
- **Globaldifs**: A dataframe with the calculated means and differentials of beta for each group and contrast, respectively.

- **DMR_results**: A list of objects generated by the mCSEA::mCSEATest function, with the statistics of the DMR analysis for each contrast.

3.6.2 Filtered bed files

DMPs or DMRs genomic positions can be downloaded in BED format. The list can be generated with the hypomethylated and hypermethylated lists of DMPs/DMRs in each contrast, or with the heatmap clustering (if the Row Colors option is enabled). Additionally, a list with the background (all the genomic positions or genomic regions tested) is also generated.

Moreover, the genome can also be selected (hg19 or hg38 genome). Since 450k and EPIC arrays are annotated in hg19, when hg38 is selected, shinyÉPICo uses the function rtracklayer::liftOver to convert the genomic coordinates. It should be note that, as a result of this process, some positions may not be converted, and will not appear in the final lists.

This option is convenient to produce some typical analysis before the DMP/DMR calculation, such as Gene Ontology Over-representation (with, for example, GREAT website) or motif enrichment analysis (with, for example, the HOMER application).

3.6.3 Workflow Report

An HTML file with all the options selected during the analysis, and most tables and charts produced, can be generated and downloaded. This report is intended as a reference that indicates all the details of the analysis that can be consulted at any time and reproduced again in shinyÉPICo if necessary.

3.6.4 Custom R Script

This option creates a custom R script, including the parameters selected in the analysis, the main functions that shinyÉPICo has used to produced the results and the pipeline followed, from the data importing to the DMP/DMR calculation, DMP/DMR filtering and heatmap(s) generation. The user only has to select the proper folder with the unzipped data used in the analysis (sample sheet and iDAT files), and running the script will produce the same objects as shinyÉPICo.

3.6.5 Heatmap(s)

Both DMP and DMR heatmaps can be downloaded in PDF format. This high quality vector format allows the user to modify heatmaps using software such as Inkscape or Adobe Illustrator.

4 Session info

```
#> R Under development (unstable) (2020-11-08 r79408)
#> Platform: x86_64-pc-linux-gnu (64-bit)
#> Running under: Ubuntu 18.04.3 LTS
#>
```

```
#> Matrix products: default
#> BLAS:   /home/omorante/R_devel/lib/R/lib/libRblas.so
#> LAPACK: /home/omorante/R_devel/lib/R/lib/libRlapack.so
#>
#> locale:
#>  [1] LC_CTYPE=es_ES.UTF-8       LC_NUMERIC=C
#>  [3] LC_TIME=en_AU.UTF-8        LC_COLLATE=es_ES.UTF-8
#>  [5] LC_MONETARY=en_AU.UTF-8    LC_MESSAGES=es_ES.UTF-8
#>  [7] LC_PAPER=en_AU.UTF-8       LC_NAME=C
#>  [9] LC_ADDRESS=C               LC_TELEPHONE=C
#> [11] LC_MEASUREMENT=en_AU.UTF-8 LC_IDENTIFICATION=C
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#> [1] BiocStyle_2.19.0
#>
#> loaded via a namespace (and not attached):
#>  [1] rstudioapi_0.13     knitr_1.30          magrittr_1.5
#>  [4] usethis_1.6.3       devtools_2.3.2      pkgload_1.1.0
#>  [7] R6_2.5.0            rlang_0.4.8         fansi_0.4.1
#> [10] stringr_1.4.0       tools_4.1.0         pkgbuild_1.1.0
#> [13] xfun_0.19           sessioninfo_1.1.1   cli_2.1.0
#> [16] withr_2.3.0         htmltools_0.5.0     ellipsis_0.3.1
#> [19] remotes_2.2.0       yaml_2.2.1          assertthat_0.2.1
#> [22] digest_0.6.27       rprojroot_1.3-2     bookdown_0.21
#> [25] crayon_1.3.4        processx_3.4.4      BiocManager_1.30.10
#> [28] callr_3.5.1         fs_1.5.0            ps_1.4.0
#> [31] testthat_3.0.0      evaluate_0.14       memoise_1.1.0
#> [34] glue_1.4.2          rmarkdown_2.5       stringi_1.5.3
#> [37] compiler_4.1.0      desc_1.2.0          backports_1.2.0
#> [40] prettyunits_1.1.1
```

References

Aryee, Martin J., Andrew E. Jaffe, Hector Corrada-Bravo, Christine Ladd-Acosta, Andrew P. Feinberg, Kasper D. Hansen, and Rafael A. Irizarry. 2014. "Minfi: A flexible and comprehensive Bioconductor package for the analysis of Infinium DNA methylation microarrays." *Bioinformatics* 30 (10): 1363–9. https://doi.org/10.1093/bioinformatics/btu049.

Du, Pan, Xiao Zhang, Chiang Ching Huang, Nadereh Jafari, Warren A. Kibbe, Lifang Hou, and Simon M. Lin. 2010. "Comparison of Beta-value and M-value methods for quantifying methylation levels by microarray analysis." *BMC Bioinformatics* 11 (1): 587. https://doi.org/10.1186/1471-2105-11-587.

Law, Charity W., Yunshun Chen, Wei Shi, and Gordon K. Smyth. 2014. "Voom: Precision weights unlock linear model analysis tools for RNA-seq read counts." *Genome Biology* 15 (2). https://doi.org/10.1186/gb-2014-15-2-r29.

Li, Tianlu, Antonio Garcia-Gomez, Octavio Morante-Palacios, Laura Ciudad, Sevgi Özkaramehmet, Evelien Van Dijck, Javier Rodríguez-Ubreva, Alejandro Vaquero, and Esteban Ballestar. 2020. "SIRT1/2 orchestrate acquisition of DNA methylation and loss of histone H3 activating marks to prevent premature activation of inflammatory genes in macrophages." *Nucleic Acids Research* 48 (2): 665–81. https://doi.org/10.1093/nar/gkz1127.

Martorell-Marugán, Jordi, Víctor González-Rumayor, and Pedro Carmona-Sáez. 2019. "MC-SEA: Detecting subtle differentially methylated regions." *Bioinformatics* 35 (18): 3257–62. https://doi.org/10.1093/bioinformatics/btz096.

Phipson, Belinda, Stanley Lee, Ian J. Majewski, Warren S. Alexander, and Gordon K. Smyth. 2016. "Robust hyperparameter estimation protects against hypervariable genes and improves power to detect differential expression." *Annals of Applied Statistics* 10 (2): 946–63. https://doi.org/10.1214/16-AOAS920.

Ritchie, Matthew E., Belinda Phipson, Di Wu, Yifang Hu, Charity W. Law, Wei Shi, and Gordon K. Smyth. 2015. "Limma powers differential expression analyses for RNA-sequencing and microarray studies." *Nucleic Acids Research* 43 (7): e47. https://doi.org/10.1093/nar/gkv007.

Subramanian, Aravind, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjee, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, et al. 2005. "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles." *Proceedings of the National Academy of Sciences of the United States of America* 102 (43): 15545–50. https://doi.org/10.1073/pnas.0506580102.