Users' Guide
MORE

Sonia Tarazona Campos
Maider Aguerralde Martin
Blanca Tomás Riquelme

December 11, 2023

Contents

1 Introduction

One of the most common questions to be addressed when performing a multi-omics experiment is how the levels of given biological entities are being regulated by other biological entities under certain conditions. An example of this type of study would be understanding the regulatory mechanisms behind the changes in gene expression.

Potential regulators of a given gene such as miRNAs, transcription factors (TF), methylation sites, etc., can be either retrieved or predicted from public databases or obtained by a combination of experimental and computational procedures. However, a methodology for selecting the specific regulators of a particular biological system studied under certain experimental conditions is required. This is the goal of the MORE (Multi-Omics REgulation) method: modeling gene expression as a function of experimental variables, such as diseases or treatments, and the potential regulators of a given gene. The idea is to obtain more specific candidate regulators for the biological system under study by applying regression models, specifically generalized linear models (GLM), or by applying Partial Least Squares (PLS). MORE facilitates the application of GLM or PLS to multi-omic data and although it was originally conceived to study gene expression regulation, its usage can be extended to protein or metabolite levels, for instance.

MORE requires several data inputs: gene expression data, regulators' omic data, experimental design, and potential associations between genes and regulators. With this input data, MORE generates the initial model equation, which is different for each gene because each one of them has different potential regulators. MORE admits numerical omic data (continuous or discrete) or binary data.

It is strongly recommended to fit MORE models only to genes that present significant changes in any of the experimental conditions studied, that is, to differentially expressed genes (DEGs). DEGs can be selected with the standard procedures depending on the experimental design, but DEGs selection is not included in the MORE algorithm and must be done by the user.

This idea can be extended to potential regulators since regulators that do not change across conditions are not good candidates to regulate gene expression. Removing non-DE regulators will also help to reduce the number of predictors in the model since an excess of them would prevent the estimation of regression coefficients. Even so, MORE has several functionalities to filter regulators with missing values or low variation, highly correlated regulators, and perform variable selection.

MORE package also includes a function to retrieve the significant regulations and the magnitude of the regulatory effect under each experimental condition considered and an additional function to graphically investigate the relationship between genes and regulators.

2 Getting started

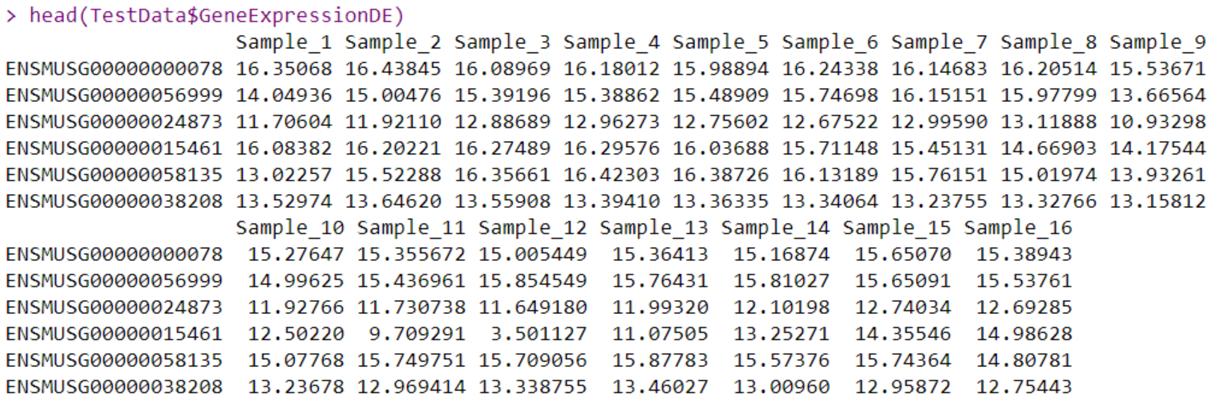
The MORE method is available as an R package from https://github.com/ConesaLab/MORE.git. As for other packages in GitHub, it can be installed from R with the following instructions:

```
> install.packages("devtools")
> devtools::install_github("ConesaLab/MORE")
```

3 Input data

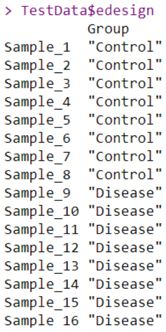
This section describes the main data files required by MORE to generate the regression models.

Gene expression data Expression values for each gene, in rows, under each experimental condition or replicate, in columns. MORE accepts either a **matrix** or a **data frame**. See an example below:



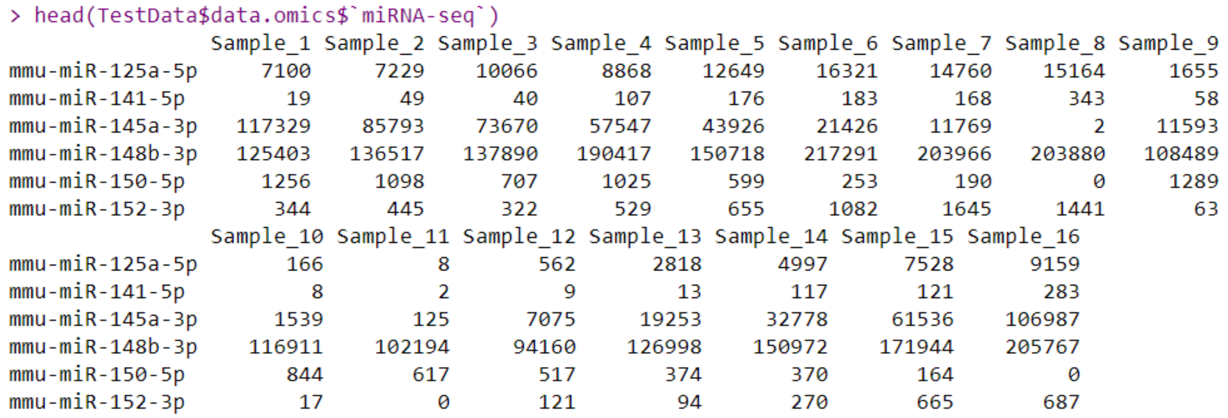
```
> head(TestData$GeneExpressionDE)
                   Sample_1 Sample_2 Sample_3 Sample_4 Sample_5 Sample_6 Sample_7 Sample_8 Sample_9
ENSMUSG00000000078 16.35068 16.43845 16.08969 16.18012 15.98894 16.24338 16.14683 16.20514 15.53671
ENSMUSG00000056999 14.04936 15.00476 15.39196 15.38862 15.48909 15.74698 16.15151 15.97799 13.66564
ENSMUSG00000024873 11.70604 11.92110 12.88689 12.96273 12.75602 12.67522 12.99590 13.11888 10.93298
ENSMUSG00000015461 16.08382 16.20221 16.27489 16.29576 16.03688 15.71148 15.45131 14.66903 14.17544
ENSMUSG00000058135 13.02257 15.52288 16.35661 16.42303 16.38726 16.13189 15.76151 15.01974 13.93261
ENSMUSG00000038208 13.52974 13.64620 13.55908 13.39410 13.36335 13.34064 13.23755 13.32766 13.15812
                   Sample_10 Sample_11 Sample_12 Sample_13 Sample_14 Sample_15 Sample_16
ENSMUSG00000000078  15.27647 15.355672 15.005449  15.36413  15.16874  15.65070  15.38943
ENSMUSG00000056999  14.99625 15.436961 15.854549  15.76431  15.81027  15.65091  15.53761
ENSMUSG00000024873  11.92766 11.730738 11.649180  11.99320  12.10198  12.74034  12.69285
ENSMUSG00000015461  12.50220  9.709291  3.501127  11.07505  13.25271  14.35546  14.98628
ENSMUSG00000058135  15.07768 15.749751 15.709056  15.87783  15.57376  15.74364  14.80781
ENSMUSG00000038208  13.23678 12.969414 13.338755  13.46027  13.00960  12.95872  12.75443
```

Experimental design Matrix or data frame containing the experimental covariates, such as treatments, diseases, strains, dose of a drug, etc. The rows of the object must be the same as the columns in **Gene expression data** and in the same order, as shown below.



```
> TestData$edesign
          Group
Sample_1  "Control"
Sample_2  "Control"
Sample_3  "Control"
Sample_4  "Control"
Sample_5  "Control"
Sample_6  "Control"
Sample_7  "Control"
Sample_8  "Control"
Sample_9  "Disease"
Sample_10 "Disease"
Sample_11 "Disease"
Sample_12 "Disease"
Sample_13 "Disease"
Sample_14 "Disease"
Sample_15 "Disease"
Sample 16 "Disease"
```

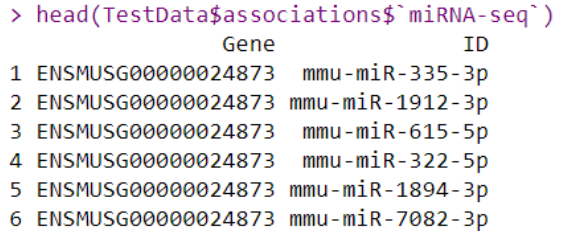
Regulatory omic data This object must be a list where each element is a matrix or data frame containing the data for each “regulatory” omic (miRNA expression, transcription factor expression, etc.), with a structure similar to gene expression data: regulators in rows and experimental conditions in columns (the columns must be the same as in gene expression and in the same order). See the example below (TestData$data.omics$‘miRNA-seq‘).



```
> head(TestData$data.omics$`miRNA-seq`)
                Sample_1 Sample_2 Sample_3 Sample_4 Sample_5 Sample_6 Sample_7 Sample_8 Sample_9
mmu-miR-125a-5p     7100     7229    10066     8868    12649    16321    14760    15164     1655
mmu-miR-141-5p        19       49       40      107      176      183      168      343       58
mmu-miR-145a-3p   117329    85793    73670    57547    43926    21426    11769        2    11593
mmu-miR-148b-3p   125403   136517   137890   190417   150718   217291   203966   203880   108489
mmu-miR-150-5p      1256     1098      707     1025      599      253      190        0     1289
mmu-miR-152-3p       344      445      322      529      655     1082     1645     1441       63
                Sample_10 Sample_11 Sample_12 Sample_13 Sample_14 Sample_15 Sample_16
mmu-miR-125a-5p       166         8       562      2818      4997      7528      9159
mmu-miR-141-5p          8         2         9        13       117       121       283
mmu-miR-145a-3p      1539       125      7075     19253     32778     61536    106987
mmu-miR-148b-3p    116911    102194     94160    126998    150972    171944    205767
mmu-miR-150-5p        844       617       517       374       370       164         0
mmu-miR-152-3p         17         0       121        94       270       665       687
```

Associations For each regulatory omic, associations between regulators and genes which indicate which are the potential regulators of each gene that will be consequently incorporated into the initial equation of the regression model. The association objects must be data frames and stored in a single **list** (attached below the example of miRNA-seq, TestData$associations$‘miRNA-seq‘). The names of the elements of this list must be the names of the list collecting regulatory omic data and must be in the same order.

If the user wants to consider all regulators of an omic as potential regulators they must set to NULL the object of this omic in the **associations** list. Moreover, if the user does not provide the list of **associations**, all regulators of all omics in **data.omics** will be considered potential regulators for all genes. However, this option is very time-consuming. By default, NULL.



```
> head(TestData$associations$`miRNA-seq`)
                Gene             ID
1 ENSMUSG00000024873  mmu-miR-335-3p
2 ENSMUSG00000024873 mmu-miR-1912-3p
3 ENSMUSG00000024873  mmu-miR-615-5p
4 ENSMUSG00000024873  mmu-miR-322-5p
5 ENSMUSG00000024873 mmu-miR-1894-3p
6 ENSMUSG00000024873 mmu-miR-7082-3p
```

4 Generating the regression models with MORE

The **more** function in MORE adjusts a generalized linear model (GLM) with elastic net regularization regression method for each gene (protein, metabolite, etc.) in the *GeneExpresion* object to determine which regulators and experimental covariates have a significant effect on the response variable (gene expression, protein levels, etc.) if the selected method by the user is 'glm'. If the selected method is 'pls', it adjusts a partial least squares (PLS) model instead of the GLM. These are the arguments the function accepts, described in detail in Section 4.1.

```
more(GeneExpression, data.omics, associations, omic.type = NULL,
     edesign = NULL, clinic = NULL, clinic.type = NULL,
     center = TRUE, scale = TRUE, scaletype = 'auto',
     epsilon = 0.00001, min.variation = 0, interactions.reg = TRUE,
     family.glm = gaussian(), elasticnet.glm = NULL,
     col.filter.glm = 'cor', correlation.glm = 0.7, alfa.pls = 0.05,
     p.method.pls = 'jack', vip.pls = 0.8, method  ='glm')
```

4.1 Arguments for more() function

Before explaining in more detail each input parameter of more() function, we make clear that inputs with *.glm* extensions are only used for GLM model construction, that inputs with *.pls* extensions apply only to PLS models (both for pls1 and pls2), and inputs with no extension apply to both methodologies.

GeneExpression Matrix or data frame containing gene expression data with genes in rows and experimental samples in columns. The row names must be the gene IDs.

data.omics List where each element corresponds to a different omic data type (miRNAs, transcription factors, methylation, etc.). The names of this list will be the omics, and each element of the list is a matrix or data frame with omic regulators in rows and samples in columns.

associations List where each element corresponds to a different omic data type (miRNAs, transcription factors, methylation, etc.). The names of the elements of the list will be the omics (in the same order as in **data.omics**). Each element is a data frame with two columns (optionally three) describing the potential interactions between genes and regulators for that omic. The first column must contain the regulators, the second the gene IDs, and an additional column can be added to describe the type of interaction (for example, in methylation data, if a CpG site is located in the promoter region of the gene, in the first exon, etc.). Optionally, the user can set the **associations** data frame of an omic equal to NULL if they want to consider all the regulators of that omic as potential regulators for all the genes. They can even set **associations** to NULL if they want to consider all regulators of all omics in

data.omics as potential regulators to all genes. Even if it can be done, we do not recommend the user to do it as it can be very time-consuming.

omic.type Vector with as many elements as the number of omics, indicating whether the omic values are numeric (0) or binary (1). When NULL is indicated, **MORE** will estimate which type of omics are provided and display them on the screen. If a single value is provided, the type for all the omics is set to that value. By default, NULL. If the estimated type of omics are incorrect, the user must halt the process and manually specify the omic.type.

edesign Data frame or matrix describing the experimental design. Rows must be the samples, that is, the columns in the **GeneExpression**, and columns must be the experimental covariates to be included in the model, such as disease, treatment, etc.

clinic Data frame or matrix containing clinical variables values where rows must represent samples and columns variables.

clinic.type Vector with as many elements as the number of clinical variables, indicating whether the variables values are numeric (0) or categorical/binary (1). When NULL is indicated, **MORE** will estimate which type of variables are provided and display them on the screen. If a single value is provided, the type for all the variables is set to that value. By default, NULL. If the estimated type of variables are incorrect, the user must halt the process and manually specify the clinic.type.

center If TRUE (default), the omic data are centered.

scale If TRUE (default), the omic data are scaled.

scaletype Type of scaling to be applied when adjusting a model if scaling is requested. It can be: 'auto', 'pareto', or 'block'. The first applies the autoscaling method; so that scales each variable independently. The second applies the Pareto scaling to the omics. The third applies the block scaling considering as block each of the omics in **data.omics** and the interactions of experimental design variables with them if they were.

epsilon A threshold for the positive convergence tolerance in the GLM model. By default, 0.00001.

min.variation Vector with as many elements as the number of omics (names of this vector will be the omics), indicating the minimum change in the standard deviation that a regulator must show across conditions in order not to be considered as having low variation and be removed from the regression models, for numerical regulators. Or the minimum change in the proportion a regulator must show across conditions for binary regulators. When a single value is given, the minimum change will be considered the same for all omics. The user has the option to set this value to NA if they do not want to provide a value but are sure that they want to filter more than constant regulators across conditions. In this case, the value will be calculated as the 10% of the maximum observed variability across conditions for continuous regulators and as the 10% of the maximum observed proportion difference across conditions for binary regulators. Additionally, the user can combine both functionalities; indeed, the user has the option to provide a vector containing the minimum change in the standard deviation for some omics and NA for those omics for which they do not want to provide a value. By default, its value is 0. If the user has been very restrictive an error message will be provided.

interactions.reg If TRUE (default), **MORE** allows for interactions between each regulator and the experimental covariate.

family.glm Error distribution and link function to be used in the GLM model (see `glm` for more information). By default, `gaussian()`.

elasticnet.glm ElasticNet mixing parameter (α). By default, NULL. These are the values that can be passed to this argument:

NULL α parameter will be automatically optimized for *cvup*, which is the mean cross-validated error plus the estimate of the standard error. For computational efficiency, it will only be tested with values ranging from 0 to 1 in increments of 0.1.

Value between 0 and 1 ElasticNet is applied with this α being the combination between ridge and lasso penalization.

Vector of α's ElasticNet will be applied for each of the α values provided in the vector, and the one that optimizes the *cvup* will be selected.

Value 0 The ridge penalty.

Value 1 The lasso penalty.

We make clear that the shrinkage parameter (λ) will in all cases be optimized by cross-validation.

col.filter.glm Type of filtering to be applied when adjusting a GLM model. This filter looks for highly correlated groups of regulators and considers the selected filter and the considered correlation threshold to select a representative. It can be 'cor' or 'pcor' if the partial correlation wants to be considered.

correlation.glm Correlation threshold (in absolute value) to decide which regulators are correlated, in which case, a representative of the group of correlated regulators is chosen to enter the model. By default, 0.7.

alfa.pls Significance level. By default, 0.05.

p.method.pls Type of methodology to apply for computing the p-values of the variables within the model. There are two available options: 'jack,' which refers to the Jack-

Knife resampling method, and ’perm,’ which corresponds to the response variable permutation method for obtaining the distribution of the coefficients under the null hypothesis and compute their associated p-value.

vip.pls The Variable Importance in Projection, VIP, score threshold to apply together with the p-value threshold to take a variable as significant; both requirements should be met to take a variable as significant. By default, 0.8.

method This parameter indicates whether a GLM model will be applied, ’glm’, or if a PLS model will be applied instead. In this case, the user can ask for a PLS1 model using ’pls1’ in which for each of the genes in **GeneExpression** a different PLS model will be applied or for a PLS2 model using ’pls2’ in which a single PLS model will be computed which creates a single model for all genes. The user must be aware that in the ’pls2’ model, the **association** list will not be considered and must be set to NULL.

4.2 more output

The object returned by the **more** function varies depending on the selected method.

4.2.1 more output for GLM

The object returned by the **more** when fitting a GLM model is a list that contains the following elements:

ResultsPerGene is a list with as many elements as genes in the GeneExpression object. For each gene, there is a list containing the following information:

Y Data frame with the response variable values for that gene (y), the values fitted by the model (fitted.y), and the residuals of the model (residuals).

X Data frame with all the predictors included in the final model.

coefficients Matrix with the estimated coefficients for the regulators selected as relevant by the elastic net regularization method.

allRegulators Data frame with all the initial potential regulators in rows and the following information in columns: gene, regulator, omic, area (the third optional column in associations), filter (if the regulator has been filtered out of the model, this column indicates the reason), and Rel (1 if the regulator is considered relevant and 0 if not). Regarding the filter column, several values are possible:

- *MissingValue*: If the regulator has been filtered out of the study because it has missing values.

- *LowVariation*: If the regulator has been filtered out of the study because it has lower variability than the threshold set by the user in min.variation parameter.
- *Model*: When the regulator is included in the initial equation model.
- *omic_mcX_X_X*: For example, *TF_mc1_1_R*. This notation is related to highly correlated regulators and how they are treated to avoid the multicollinearity problem. Following the *TF_mc1_1_R* example, two or more regulators, which potentially regulate the gene, are highly correlated (in absolute value). In such cases, one is chosen as the representative and indicated with _R. The rest of the regulators considered if they are directly highly correlated to it are labeled with _P, which means that they are positively correlated with the representative and with _N if negatively correlated. Once a representative is taken for them, if there are still highly correlated regulators, the process is repeated and indicated in the _1 of the example. An additional row is then added to this table, with the regulator *TF_mc1_1_R* and the filter label being *Model*, since only this representative is considered in the model. When there are several groups of correlated regulators for the same omic, it is indicated with _mc1_, _mc2_, etc.

relevantRegulators A character vector containing the relevant regulators.

GlobalSummary List that contains the following elements:

GoodnessOfFit Matrix that collects the R-squared value (which for GLMs is defined as the percentage of deviance explained by the model),the adjusted R-squared value, the Root Mean Square Error (RMSE), the Coefficient of Variation of the Root Mean Square Error (CV(RMSE)) and the number of relevant regulators for all the genes that had at least a relevant regulator.

ReguPerGene Matrix containing, for each omic and gene, the number of initial regulators, the number of regulators included in the initial model, and the number of relevant regulators.

GenesNOmodel List of genes for which the final GLM model with the elastic net regularization could not be obtained. There are three possible reasons for that, and they are indicated: “Too many missing values”, “-Inf/Inf values”, and ”No regulators left after NA/LowVar filtering”.

GenesNoregulators List of genes for which there were no initial regulators, only generated in case any gene was under this condition.

GlobalRegulators Vector that contains regulators that relevantly regulate more genes. It comprises the third quartile of regulators with the highest number of regulations. If the count is below 10, the top 10 regulators are selected.

HubGenes Vector that contains genes that are relevantly regulated by more regulators. It comprises the third quartile of genes with the highest number of regulators. If the count is below 10, the top 10 genes are selected.

Arguments List with the arguments used to generate the model: experimental design matrix, minimum degrees of freedom in the residuals, significance level, family distribution, etc.

4.2.2 more output for PLS

The object returned by the **more** when fitting a PLS (PLS1 or PLS2 indifferently) model is a list that contains the following elements:

ResultsPerGene is a list with as many elements as genes in the GeneExpression object. For each gene, there is a list containing the following information:

Y Data frame with the response variable values for that gene (y), the values fitted by the model (fitted.y), and the residuals of the model (residuals).

X Data frame with all the predictors included in the final model.

coefficients Matrix with the estimated coefficients for the regulators selected as significant by the selected p-value computation method (p.method) and whose Variable Importance in Projection (VIP) has been higher than the threshold.

allRegulators Data frame with all the initial potential regulators in rows and the following information in columns: gene, regulator, omic, area (the third optional column in associations), filter (if the regulator has been filtered out of the model, this column indicates the reason), and Sig (1 if the regulator is considered significant and 0 if not). Regarding the filter column, several values are possible:

- *MissingValue*: If the regulator has been filtered out of the study because it has missing values.
- *LowVariation*: If the regulator has been filtered out of the study because it has lower variability than the threshold set by the user in min.variation parameter.
- *Model*: When the regulator is included in the initial equation model.

significantRegulators A character vector containing the significant regulators.

GlobalSummary List that contains the following elements:

GoodnessOfFit Matrix that collects the R-squaredY value (the R squared of the response variable), the Q-squared (the goodness of prediction), the square root of the mean error between the actual and the predicted responses (RMSE), the Coefficient of Variation of the Root Mean Square Error (CV(RMSE)) and the number of significant regulators for all the genes that had at least a significant regulator.

ReguPerGene Matrix containing, for each omic and gene, the number of initial regulators, the number of regulators included in the initial model, and the number of significant regulators.

GenesNOmodel List of genes for which the final GLM model with the elastic net regularization could not be obtained. There are three possible reasons for that, and they are indicated: “Too many missing values”, “-Inf/Inf values”, and ”No regulators left after NA/LowVar filtering”.

GenesNoregulators List of genes for which there were no initial regulators, only generated in case any gene was under this condition.

GlobalRegulators Vector that contains regulators that significantly regulate more genes. It comprises the third quartile of regulators with the highest number of regulations. If the count is below 10, the top 10 regulators are selected.

HubGenes Vector that contains genes that are significantly regulated by more regulators. It comprises the third quartile of genes with the highest number of regulators. If the count is below 10, the top 10 genes are selected.

Arguments List with the arguments used to generate the model: experimental design matrix, minimum degrees of freedom in the residuals, significance level, family distribution, etc.

4.2.3 more summary

Making use of the output object returned by **more**, in both cases in GLM and PLS models, the user can ask for a summary of the results obtained by:

```
summary(object, plot.more=FALSE)
```

This summary takes two arguments as input:

object MORE object obtained from applying **more** function, indifferent to the method that has been used (’glm’, ’pls1’ or ’pls2’).

plot.more If TRUE, the top 10 global regulators will be plotted against the genes they regulate. By default, FALSE. It could be very time-consuming if the global regulators regulate a huge number of genes, so it is not recommended unless the user knows that there are only a few of them. Instead, it is recommended to use the **plotmore** function to plot the specific regulations.

Once the function is used the following information will be printed on the screen:

1. Number of genes for which a model was computed.

2. Number of genes that did not have initial regulators.
3. Number of genes for which the final model could not be obtained.
4. The mean of relevant/significant regulators of the genes.
5. Top 10 hub genes and the number of relevant/significant regulators for each.
6. Top 10 global regulators and the number of genes they regulate. They will have to regulate at least 10 genes to be considered global regulators.
7. If required with plot.more = TRUE, the plots of the global regulators against the genes they regulate.

4.3 Running an example

An example of the execution of **more** function for the 'glm' option is shown next by using simulated data. Even if the data file `TestData.RData` is available in the package; the results shown below are related to the STATegra database available here.

In this file, the gene expression matrix corresponds to the omic RNA-seq (**GeneExpressionDE**), and there is a list with four matrices of regulators in the **data.omics** object:

miRNA-seq miRNA expression data.

DNase-seq measures the chromatin accessibility expression.

Methyl-seq Methylation per CpG site (M values).

TF TF expression data.

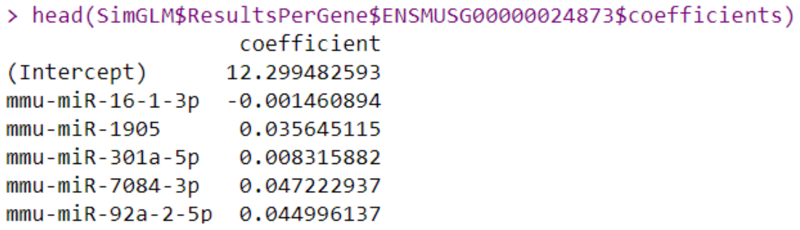
All values are normalized

The experimental design matrix (**edesign**) consists of 6 time points in two conditions, which results in a total of 12 experimental samples, but time is not to be considered as an experimental covariate since we are interested in comparing temporal profiles for the two experimental groups.

We can run the following **more** code to obtain the regression models for our genes:

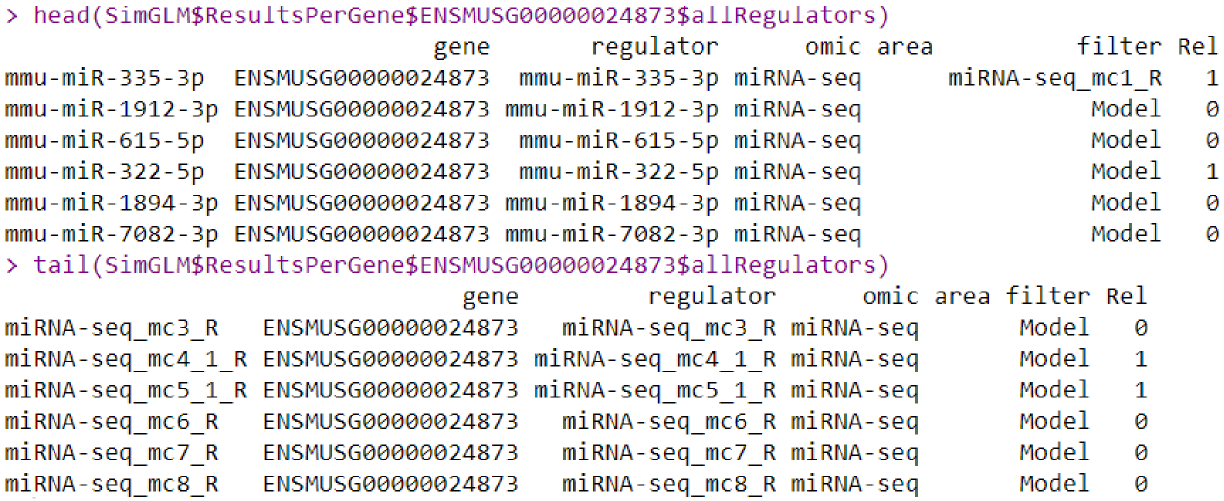
```
> set.seed(123)
> SimGLM = more(GeneExpression = TestData$GeneExpressionDE,
                data.omics = TestData$data.omics,
                associations = TestData$associations,
                omic.type = c(1,0,0), edesign = TestData$edesign,
                min.variation = 0, interactions.reg = TRUE,
                elasticnet.glm = NULL, col.filter.glm = 'cor',
                correlation.glm = 0.7, method = 'glm')
```

Some of the estimated coefficients of the relevant regulators in the final GLM model computed by **more** for the gene ENSMUSG00000024873 are



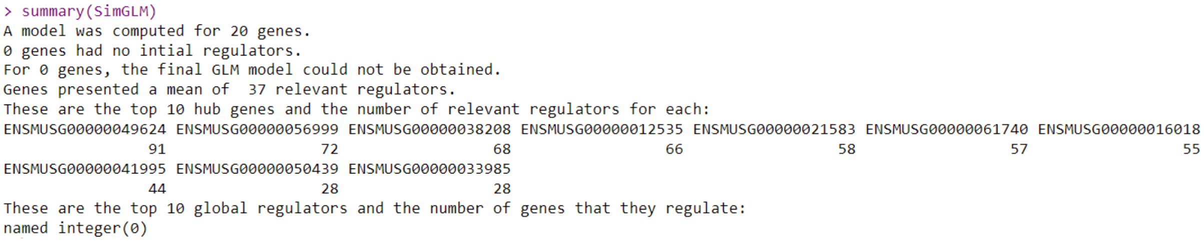
```
> head(SimGLM$ResultsPerGene$ENSMUSG00000024873$coefficients)
                 coefficient
(Intercept)      12.299482593
mmu-miR-16-1-3p  -0.001460894
mmu-miR-1905      0.035645115
mmu-miR-301a-5p   0.008315882
mmu-miR-7084-3p   0.047222937
mmu-miR-92a-2-5p  0.044996137
```

The **allRegulators** table shows, for each gene, their regulators, omic, area, the kind of filter applied, and if the regulator is considered relevant or not. In this case (see **filter** column), in miRNA-seq, the regulator mmu-miR-335-3p has been chosen as representative regulator (R) for being correlated to other regulators. In the filter column, the correlations are indicated using (R) for the representative, (P), and (N) for positive and negative correlation with the representative, respectively. In the same column, `Model` means that the regulator was included in the model by itself. On the other hand, the **Rel** column returns 1 if the regulator was considered relevant in the final model and 0 if not.



```
> head(SimGLM$ResultsPerGene$ENSMUSG00000024873$allRegulators)
                              gene        regulator      omic  area          filter  Rel
mmu-miR-335-3p  ENSMUSG00000024873   mmu-miR-335-3p miRNA-seq      miRNA-seq_mc1_R   1
mmu-miR-1912-3p ENSMUSG00000024873  mmu-miR-1912-3p miRNA-seq                Model   0
mmu-miR-615-5p  ENSMUSG00000024873   mmu-miR-615-5p miRNA-seq                Model   0
mmu-miR-322-5p  ENSMUSG00000024873   mmu-miR-322-5p miRNA-seq                Model   1
mmu-miR-1894-3p ENSMUSG00000024873  mmu-miR-1894-3p miRNA-seq                Model   0
mmu-miR-7082-3p ENSMUSG00000024873  mmu-miR-7082-3p miRNA-seq                Model   0
> tail(SimGLM$ResultsPerGene$ENSMUSG00000024873$allRegulators)
                                gene         regulator      omic area filter Rel
miRNA-seq_mc3_R   ENSMUSG00000024873   miRNA-seq_mc3_R miRNA-seq       Model   0
miRNA-seq_mc4_1_R ENSMUSG00000024873 miRNA-seq_mc4_1_R miRNA-seq       Model   1
miRNA-seq_mc5_1_R ENSMUSG00000024873 miRNA-seq_mc5_1_R miRNA-seq       Model   1
miRNA-seq_mc6_R   ENSMUSG00000024873   miRNA-seq_mc6_R miRNA-seq       Model   0
miRNA-seq_mc7_R   ENSMUSG00000024873   miRNA-seq_mc7_R miRNA-seq       Model   0
miRNA-seq_mc8_R   ENSMUSG00000024873   miRNA-seq_mc8_R miRNA-seq       Model   0
```

Finally, we can ask for a litlle summary of the created model calling summary function:



```
> summary(SimGLM)
A model was computed for 20 genes.
0 genes had no intial regulators.
For 0 genes, the final GLM model could not be obtained.
Genes presented a mean of  37 relevant regulators.
These are the top 10 hub genes and the number of relevant regulators for each:
ENSMUSG00000049624 ENSMUSG00000056999 ENSMUSG00000038208 ENSMUSG00000012535 ENSMUSG00000021583 ENSMUSG00000061740 ENSMUSG00000016018
                91                 72                 68                 66                 58                 57                 55
ENSMUSG00000041995 ENSMUSG00000050439 ENSMUSG00000033985
                44                 28                 28
These are the top 10 global regulators and the number of genes that they regulate:
named integer(0)
```

5 Retrieving significant regulations from MORE results

The function **RegulationPerCondition** is applied to the **more** output. It returns a summary table containing all the relevant/significant regulations, that is, all the pairs gene-regulator considered relevant/significant in MORE models (depending if a GLM or a PLS model was applied). Moreover, it provides the regression coefficient that relates the gene and the regulator for each experimental condition after testing if this coefficient is relevant/significant or not.

```
RegulationPerCondition(output)
```

5.1 RegulationPerCondition input parameters

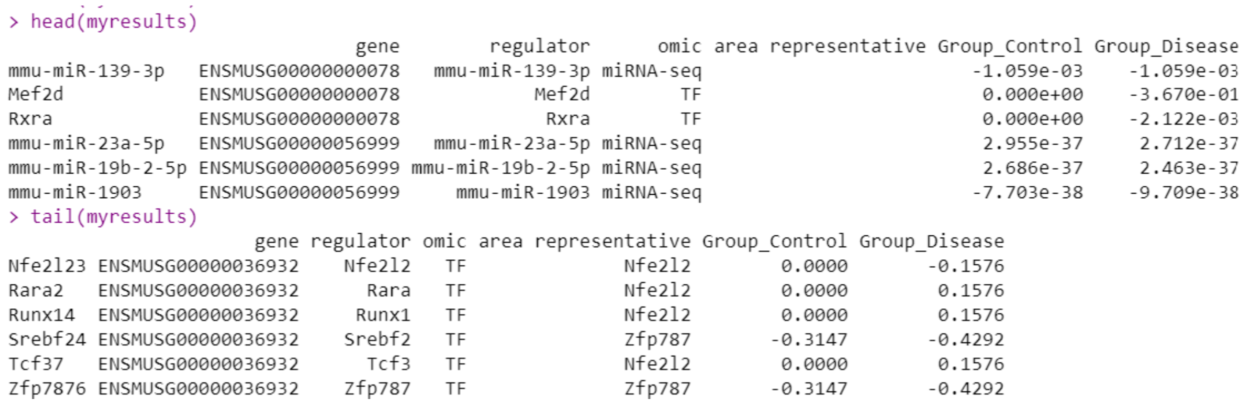
output Object containing the output of **more** function.

5.2 Interpreting RegulationPerCondition output with an example

Following the previous example, we can run the **RegulationPerCondition** function.

```
> myresults = RegulationPerCondition(SimGLM)
```

The output is the following table, where some pairs gene-regulator are selected to have a complete vision of the output of this function:



```
> head(myresults)
                               gene        regulator      omic area representative Group_Control Group_Disease
mmu-miR-139-3p   ENSMUSG00000000078   mmu-miR-139-3p miRNA-seq                         -1.059e-03    -1.059e-03
Mef2d            ENSMUSG00000000078            Mef2d        TF                          0.000e+00    -3.670e-01
Rxra             ENSMUSG00000000078             Rxra        TF                          0.000e+00    -2.122e-03
mmu-miR-23a-5p   ENSMUSG00000056999   mmu-miR-23a-5p miRNA-seq                          2.955e-37     2.712e-37
mmu-miR-19b-2-5p ENSMUSG00000056999 mmu-miR-19b-2-5p miRNA-seq                          2.686e-37     2.463e-37
mmu-miR-1903     ENSMUSG00000056999     mmu-miR-1903 miRNA-seq                         -7.703e-38    -9.709e-38
> tail(myresults)
                      gene regulator omic area representative Group_Control Group_Disease
Nfe2l23 ENSMUSG00000036932    Nfe2l2   TF              Nfe2l2        0.0000       -0.1576
Rara2   ENSMUSG00000036932      Rara   TF              Nfe2l2        0.0000        0.1576
Runx14  ENSMUSG00000036932     Runx1   TF              Nfe2l2        0.0000        0.1576
Srebf24 ENSMUSG00000036932    Srebf2   TF              Zfp787       -0.3147       -0.4292
Tcf37   ENSMUSG00000036932      Tcf3   TF              Nfe2l2        0.0000        0.1576
Zfp7876 ENSMUSG00000036932    Zfp787   TF              Zfp787       -0.3147       -0.4292
```

This table shows the relevant regulators for each gene. The **representative** column indicates if the regulator was chosen as the representative of a correlated group of regulators or, otherwise, which regulator was taken as the representative of the group. When no information is provided in this column, it means that the regulator was not part of a correlated group of regulators. Regulators correlated positively with the representative will have the same coefficients (same sign) as the representative, while negatively correlated regulators will have the same coefficients as the representative but with the opposite sign.

The final columns correspond to the regression coefficients of each regulator for each experimental group. In this case, the experimental design matrix (**edesign**) contained two conditions, so the column `Group_Control` corresponds to the first condition, and `Group_Disease` corresponds to the second one. These are the conclusions we can draw from the coefficients:

- If two experimental groups have the same coefficients, it means that the regulator has the same effect on the gene in both groups.
- If one of the coefficients is 0, it means that the regulator has no effect on the gene under this experimental condition.
- Experimental groups with different non-zero coefficients indicate that the regulator affects the gene in all these experimental groups but the magnitude of the effect is not the same for all these groups.

6 Plotting MORE results

MORE package includes several plots for the interpretation of the results.

6.1 summary_plot function

MORE package includes the function **summary_plot** to graphically represent the summary of the relationship between genes and regulators found when creating the models. It creates two types of summary plots depending on user specifications.

```
summary_plot(output, output_regpcond, by_genes =TRUE)
```

6.1.1 summary_plot input parameters

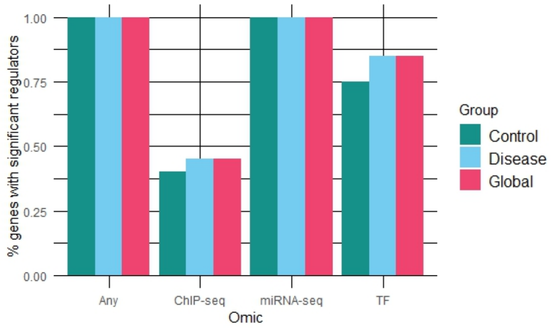
output Object generated by the function **more**.

output_regpcond Object generated by the function **RegulationPerCondition** when applied to a **more** object.

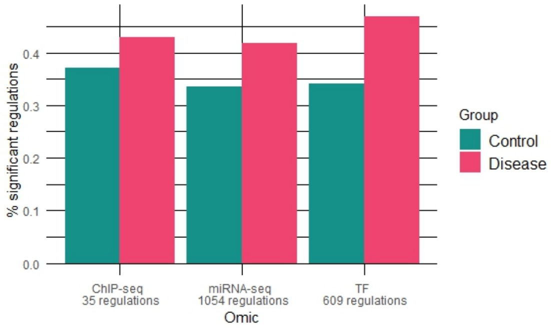
by_genes If TRUE (default), the function plots the percentage of genes with significant regulators globally and per omic. If FALSE, it plots the percentage of significant regulations per omic.

As an example we introduce the summary plots generated for the model previously created with **more**:

```
summary_plot(SimGLM, myresults)
```



```
summary_plot(SimGLM, myresults, FALSE)
```



6.2 plotmore function

MORE package includes the function **plotmore** to graphically represent the relationship between genes and regulators: for a given pair gene-regulator, to explore the regulators of a given gene, or to analyze which genes are regulated by a specific regulator.

```
plotmore(output, gene, regulator = NULL, simplify = FALSE,
         reguValues = NULL, plotPerOmic = FALSE, gene.col = 1,
         regu.col = NULL, order = TRUE, xlab = "", cont.var = NULL,
         cond2plot = NULL)
```

6.2.1 plotmore input parameters

output Object generated by the function **more**.

gene ID of the gene to be plotted.

regulator ID of the regulator to be plotted. If NULL (default value), all the regulators of the gene are plotted.

simplify If TRUE, a boxplot (if the regulator is binary) or a Scatterplot (otherwise) is plotted to represent the relationship between the gene and the regulator provided to the function. If FALSE (default), the GeneExpression and the regulator profiles will be plotted. When many samples are provided to create the models, it is hard to differentiate something in this second option.

reguValues Vector containing the values of a regulator that the user can optionally provide. If NULL (default value), these values are taken from **GLMoutput** as long as they are available.

plotPerOmic If TRUE, all the relevant regulators of the given gene and the same omic are plotted in the same graph. If FALSE (default value), each regulator is plotted in a separate plot.

gene.col Color to plot the gene. By default, 1 (black).

regu.col Color to plot the regulator. If NULL (default), a color will be assigned by the function, that will be different for each regulatory omic.

order If TRUE (default), the values in X-axis are ordered.

xlab Label for the X-axis.

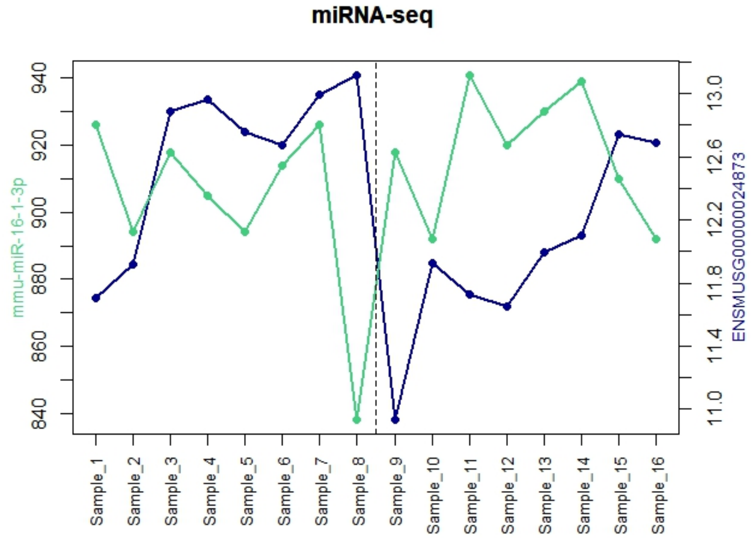
cont.var Vector with length equal to the number of observations in data, which optionally may contain the values of the numerical variable (e.g. time) to be plotted on the X-axis. By default, NULL. It plots a range for each observation in which the observation could take values taking into account the numerical variable introduced.

cond2plot Vector or factor indicating the experimental group of each value to represent. If NULL (default), the labels are taken from the experimental design matrix.

6.2.2 Interpretation of MORE plots

Following the previous example, the MORE graphic below represents the expression profile of a given gene (ENSMUSG00000024873) and the values for a relevant regulator of this gene (miRNA regulator, mmu-miR-16-1-3p). It can be generated with the following code:

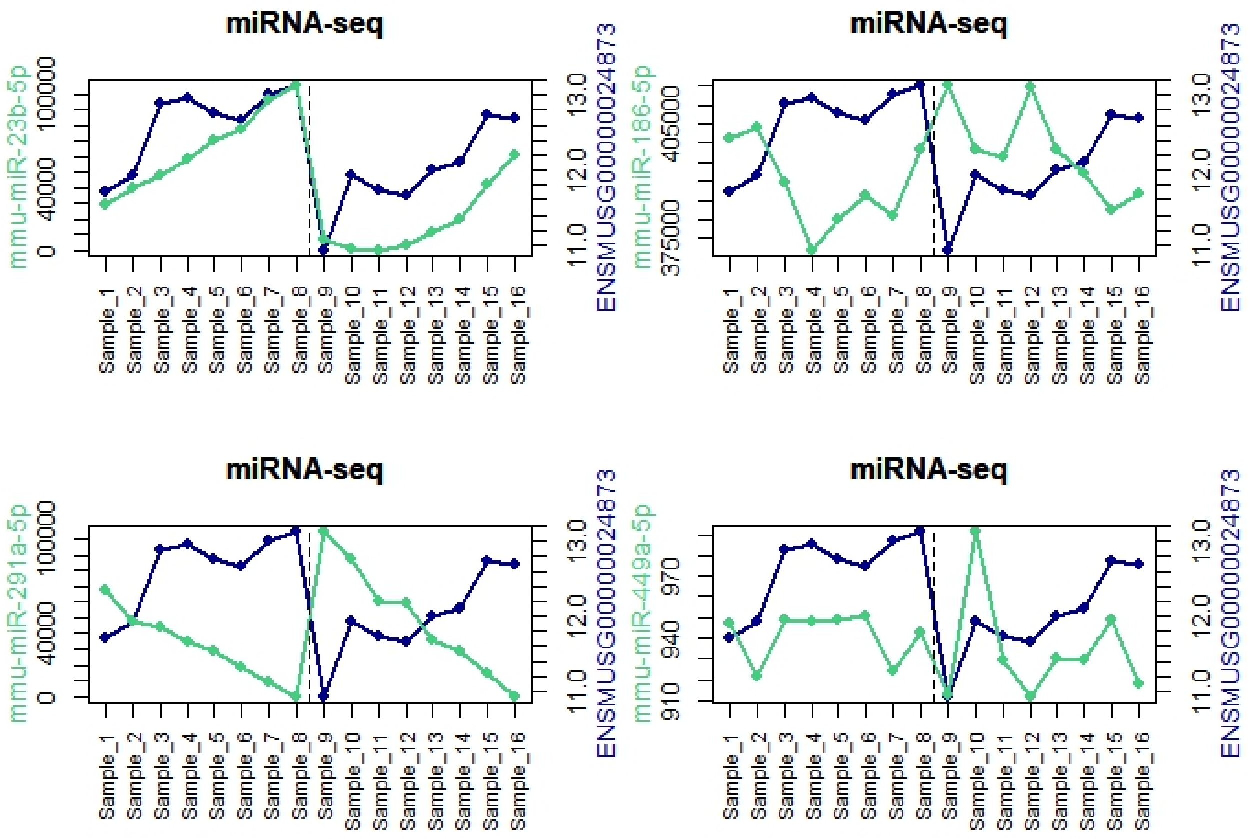
```
> plotmore(output = SimGLM, gene = "ENSMUSG00000024873",
           regulator = "mmu-miR-16-1-3p", plotPerOmic = FALSE,
           gene.col = "blue4", order = FALSE, regu.col = "seagreen3")
```



The X-axis is divided into two conditions (Control or Disease), and within each condition, the observations are displayed. The right Y-axis shows the expression values for the gene (plotted in blue), while the left Y-axis indicates the values for the regulator (plotted in green).

If we set the regulator argument to NULL, all the relevant regulators of gene ENSMUSG-00000024873 will be plotted (28 regulators). Only 4 will be presented:

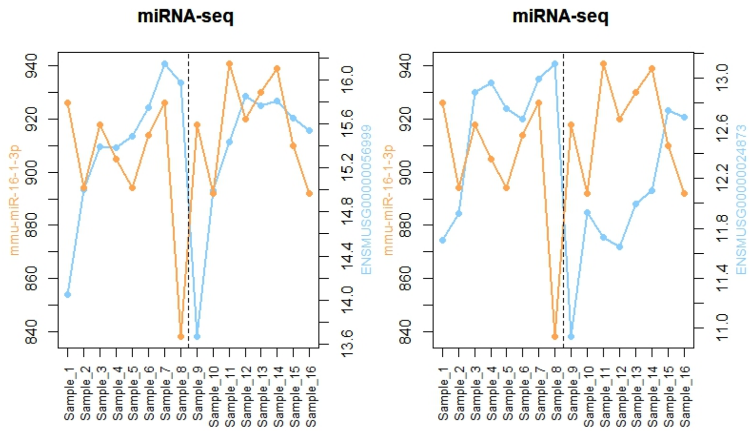
```
> par(mfrow = c(2,4))
> plotmore(output = SimGLM, gene = "ENSMUSG00000024873",
           regulator = NULL, order =FALSE,plotPerOmic = FALSE,
           gene.col = "seagreen3")
```



The title of each plot indicates the omic represented in that plot and the area. The values for relevant regulators are plotted in different colors according to the omic. The values for the gene are plotted in sea green, as indicated in the previous code.

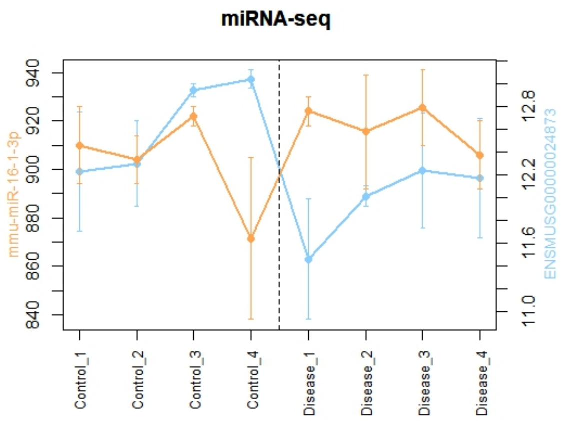
If we want to plot all the genes that are considered to be relevantly regulated by a given regulator (e.g. mmu-miR-16-1-3p), we must set the gene argument to NULL as follows. In this case, the miRNA regulates two genes: ENSMUSG00000056999 and ENSMUSG00000024873.

```
> par(mfrow = c(1,2))
> plotmore(output = SimGLM, gene = NULL,
           regulator = "mmu-miR-16-1-3p", plotPerOmic = FALSE,
           order = FALSE, gene.col = "skyblue1",
           regu.col = "tan1")
```



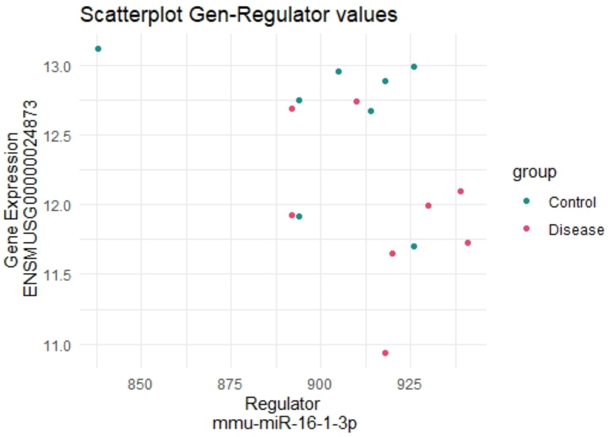
Additionally, if the user knows that the samples belong to different time points, they can provide this information in **cont.var** vector. A 'confidence interval' will be plotted for each of the time points to which the samples belong to. The code and the resulting graph, where the gene is plotted in light orange, and the regulator is plotted in blue, can be found below.

```
> plotmore(output = SimGLM, gene = "ENSMUSG00000024873",
         regulator = "mmu-miR-16-1-3p", plotPerOmic = FALSE,
         gene.col = "skyblue1", regu.col = "tan1",
         cont.var = c(1,2,3,4,1,2,3,4))
```

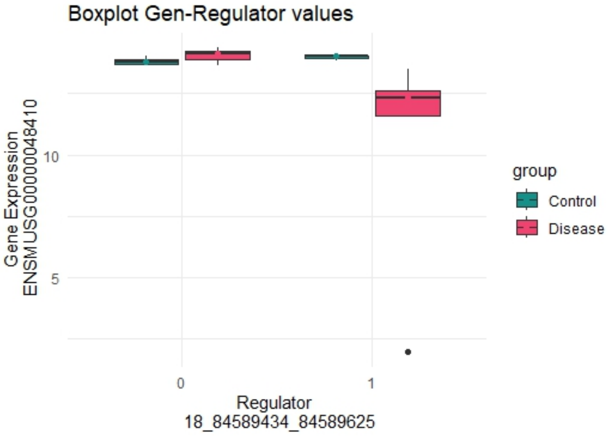


In this case, as only 16 samples are plotted the profiles can clearly be seen. However, in case there were difficulties, we encourage the user to apply simplify parameter to see the relationships between a gene and a regulator. An example for binary regulator and continuous regulator are shown:

```
> plotmore(output = SimGLM, gene = "ENSMUSG00000024873",
           regulator = "mmu-miR-16-1-3p", simplify =TRUE)
```



```
> plotmore(output = SimGLM, gene = "ENSMUSG00000048410",
           regulator = "18_84589434_84589625", simplify =TRUE)
```



6.3 network_more function

MORE package includes the function **network_more** to graphically represent the networks concluded by applying more function.

```
network_more(output_regpcond, cytoscape = TRUE, group1 = NULL,
             group2 = NULL)
```

6.3.1 network_more input parameters

output_regpcond Object generated by the application of **RegulationPerCondition** function to a **more** object.

cytoscape If TRUE (default), the function plots the network induced from **more** in *Cytoscape*. For that, it is necessary to install *RCy3* package and to maintain *Cytoscape* software opened while running the function. If FALSE, it plots this network in R using *igraph* package and saves it so that the user can introduce it later to *Cytoscape*. This option is not recommended for plotting huge networks as the visualization is complex.

group1 Name of the group to take as reference in the differential network creation. If it is not provided, the networks of all groups will be plotted. By default, NULL.

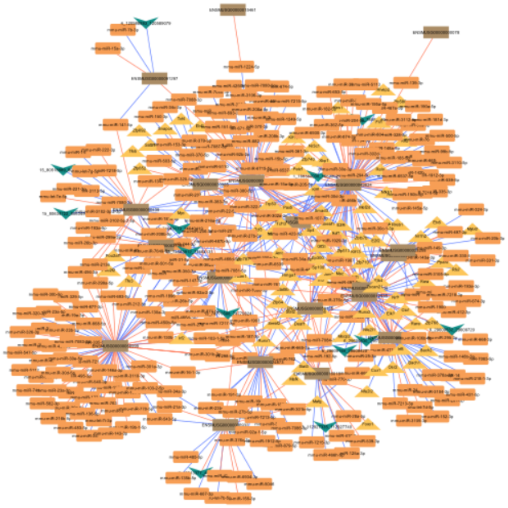
group2 Name of the group to compare to the reference in the differential network creation. If it is not provided, the networks of all groups will be plotted. By default, NULL.

NOTE 1: By default, regulators repressing the Gene expression will be connected by red edges and regulators activating the Gene expression will be connected by blue edges.

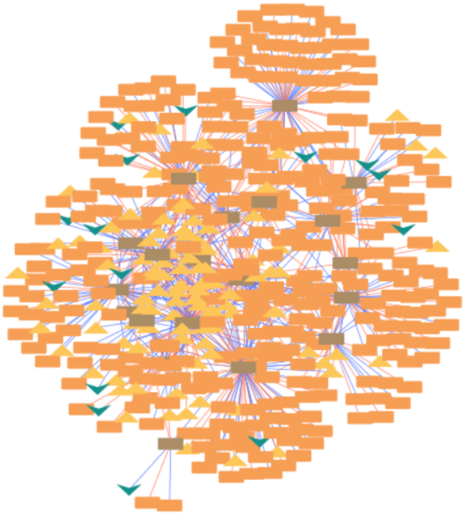
NOTE 2: By default, response variable (Gene Expression, protein expression,..), transcription factors (TFs) and microRNAs (miRNAs) will be plotted with rectangle, triangle and diamond shapes, respectively. Considering there is still no consensus in the literature on the shapes of all omics, we applied the ones by default to the rest of the omics. It is up to the user if they want to change it.

NOTE 3: In the differential network creation, dashed edges represent a change in the relation between the connected nodes compared to that present in the reference network and zig zag edges represent connections that do not exist in some of the provided groups.

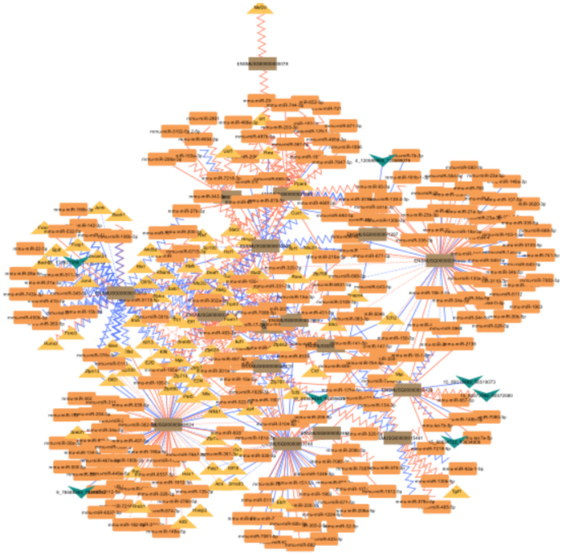
```
network_more(myresults, cytoscape = TRUE)

network_more(myresults, cytoscape = TRUE, group1 = 'Control',
             group2 = 'Disease')
```



Regulatory network in Control



Regulatory network in Disease



Differential regulatory
network in Disease-Control

7 How to use MORE with R Shiny

Shiny is an R package that allows for building web applications from R packages or scripts so users that are not familiar with R language can still easily use R packages. We have generated a MORE web application with R Shiny for this purpose. To use the MORE shiny tool, users must first install the Shiny and the Shiny themes packages from CRAN repository with `install.packages()` function.

```
> install.packages("shiny")
> install.packages("shinythemes")
```

Moreover, users must also have previously installed the MORE package described in section 2.

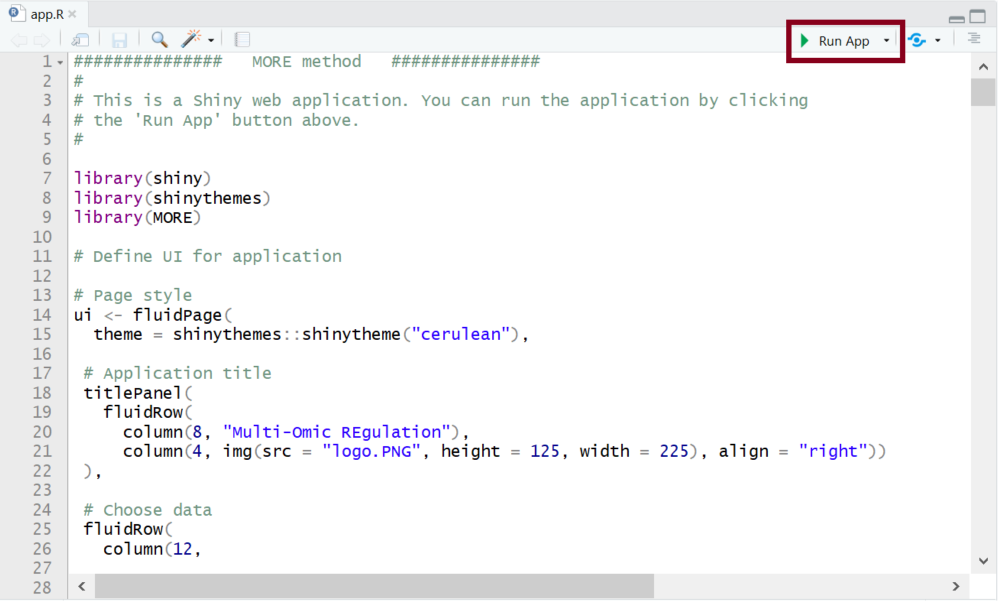
The MORE shiny scripts needed to run the tool are available in the Downloads bitbucket folder for MORE package (https://bitbucket.org/ConesaLab/more/downloads/). There is a file called app.R, an example dataset stored in the file TestDataShiny.RData and a folder called www.

app.R Script to run the web application for MORE method. Users must open this file from RStudio to start using the application.

TestDataShiny.RData Example data file to test the application. We used it as an example of an execution of MORE Shiny.

www Folder containing the style options. Please, do not delete anything in this folder.

Therefore, in order to run the application, please open the app.R file, where the MORE method application is located, and execute it using the button Run App (the red box in the following picture).

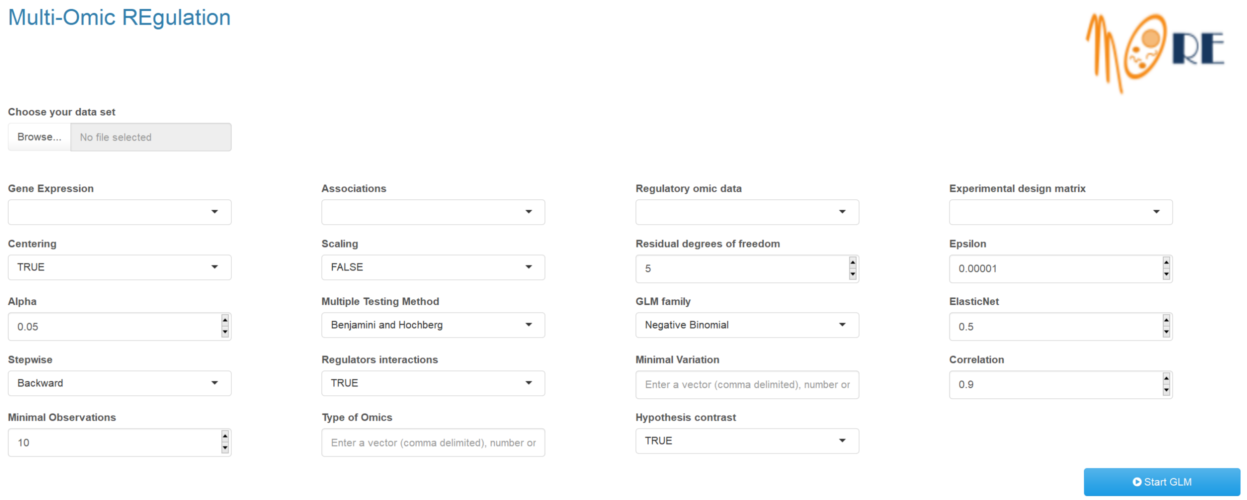


```
###############    MORE method    ###############
#
# This is a Shiny web application. You can run the application by clicking
# the 'Run App' button above.
#

library(shiny)
library(shinythemes)
library(MORE)

# Define UI for application

# Page style
ui <- fluidPage(
  theme = shinythemes::shinytheme("cerulean"),

 # Application title
 titlePanel(
   fluidRow(
     column(8, "Multi-Omic REgulation"),
     column(4, img(src = "logo.PNG", height = 125, width = 225), align = "right"))
 ),

 # Choose data
 fluidRow(
   column(12,

```

Figure 1: Run MORE application, click Run App button

A window will open with the MORE application, as shown in the following figure.

The different options have been explained in the previous sections, but next, we run an example to clarify how to use them. This example can also be visualized in this video: https://youtu.be/SSIaeFRNsXg.



Choose your data set

Gene Expression:

Centering: TRUE

Alpha: 0.05

Stepwise: Backward

Minimal Observations: 10

Associations:

Scaling: FALSE

Multiple Testing Method: Benjamini and Hochberg

Regulators interactions: TRUE

Type of Omics: Enter a vector (comma delimited), number or

Regulatory omic data:

Residual degrees of freedom: 5

GLM family: Negative Binomial

Minimal Variation: Enter a vector (comma delimited), number or

Hypothesis contrast: TRUE

Experimental design matrix:

Epsilon: 0.00001

ElasticNet: 0.5

Correlation: 0.9

Figure 2: MORE Shiny

7.1 MORE Shiny application example

Please take into account that MORE Shiny only supports `.RData` files at this time. Once the RData is loaded, the user must indicate the names of the data files in the RData that corresponds to (see the input data defined in the **section 3**): gene expression matrix, experimental design matrix, regulators matrix and association matrix. In summary (for more details see **section 3**):

Gene expression matrix Matrix or data frame that contains expression values for each gene, in rows, under each experimental condition or replicates, in columns.

Experimental design matrix Matrix or data frame that contains the experimental covariates, such as treatments, points of time...

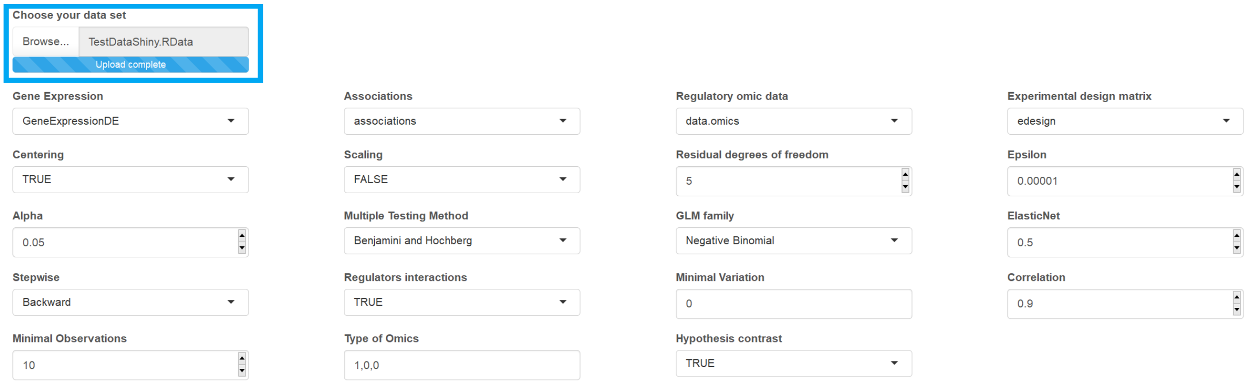
Regulatory omic data List that contains matrices or data frames containing the data for each regulatory omic, e.g. miRNA expression. The data frame structure is similar to gene expression data.

Association matrix List that contains data frames with the potential regulators for each regulatory omic considered. The association objects must be data frames and stored in a single list.

In this case, the file `TestDataShiny.RData` contains the objects described in **section 4.3**, unlike the experimental design matrix, which contains only one column with two conditions. By clicking on the button `Browse...`, users can choose their own data file (see Figure 3 blue box). Once the data is loaded, the user must enter the same input parameters of the GetGLM and RegulationPerConditon functions as defined in **section 4.1** and **section 5.1**.

It should be taken into account that if users want to enter a NULL value for a given parameter, they must leave the box empty.

In the example of the application, we will consider the input parameters shown in the following figure, leaving blank those we want to be NULL.



Choose your data set: TestDataShiny.RData

Gene Expression: GeneExpressionDE
Associations: associations
Regulatory omic data: data.omics
Experimental design matrix: edesign
Centering: TRUE
Scaling: FALSE
Residual degrees of freedom: 5
Epsilon: 0.00001
Alpha: 0.05
Multiple Testing Method: Benjamini and Hochberg
GLM family: Negative Binomial
ElasticNet: 0.5
Stepwise: Backward
Regulators interactions: TRUE
Minimal Variation: 0
Correlation: 0.9
Minimal Observations: 10
Type of Omics: 1,0,0
Hypothesis contrast: TRUE

Figure 3: MORE Shiny: inputs for running example

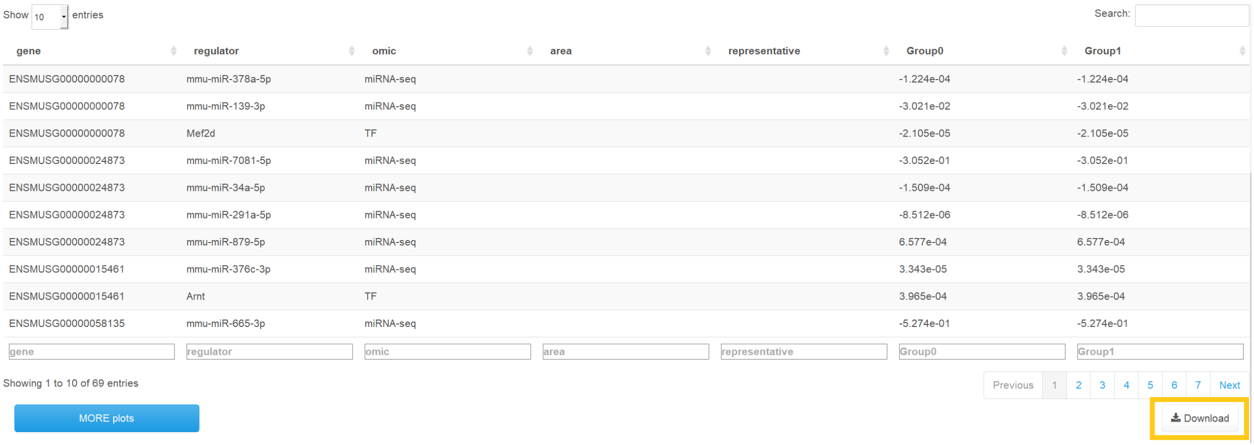
Now, clicking the button `Start GLM`, we will obtain a summary table (see Figure 4). Specifically, this is the table defined in the **section 5.2**, the output of the **RegulationPerCondition()** function. The user can download the table in CSV format by pressing the button `Download` (see Figure 4 orange box). It is necessary to save the file that will contain the table with name and extensión **.csv**.

The button `MORE plots` (see Figure 4) will generate plots to visualize the relationship between genes and regulators. The user can change the different parameters without re-executing the application to tune the plots or plot new elements.

Here we show the example for the gene ENSMUSG00000000078 (orange) and TF Mef2d (blue). Pressing the button `Generate Plot`, Shiny generates the first graph. However, if it is expected to obtain more than one graphic, the user can see all of them in a pdf file. This pdf file is generated by pressing the button `Download` (box orange in Figure 5) and saving the document, for example, as plotsGLM.pdf. It is essential to save the file with name and extension **.pdf**.

8 How to cite MORE package

Tarazona, S., Tomás-Riquelme, B., Martínez-Mira, C., Clemente-Císcar, M., Conesa, A. (2018). MORE: Multi-Omics REgulation by regression models. R package version 0.1.0.



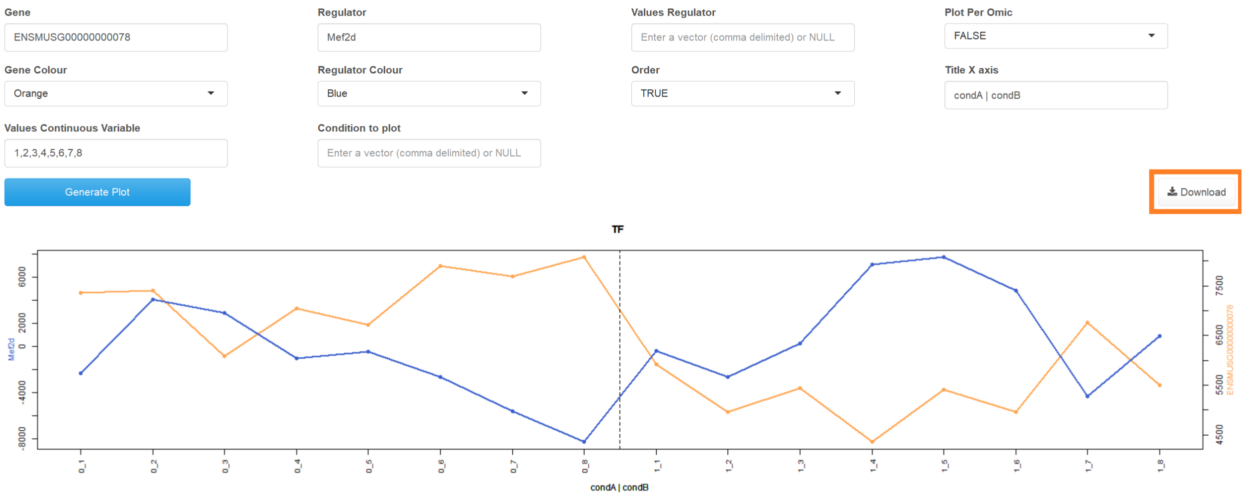
Show 10 entries

Search:

gene	regulator	omic	area	representative	Group0	Group1
ENSMUSG00000000078	mmu-miR-378a-5p	miRNA-seq			-1.224e-04	-1.224e-04
ENSMUSG00000000078	mmu-miR-139-3p	miRNA-seq			-3.021e-02	-3.021e-02
ENSMUSG00000000078	Mef2d	TF			-2.105e-05	-2.105e-05
ENSMUSG00000024873	mmu-miR-7081-5p	miRNA-seq			-3.052e-01	-3.052e-01
ENSMUSG00000024873	mmu-miR-34a-5p	miRNA-seq			-1.509e-04	-1.509e-04
ENSMUSG00000024873	mmu-miR-291a-5p	miRNA-seq			-8.512e-06	-8.512e-06
ENSMUSG00000024873	mmu-miR-879-5p	miRNA-seq			6.577e-04	6.577e-04
ENSMUSG00000015461	mmu-miR-376c-3p	miRNA-seq			3.343e-05	3.343e-05
ENSMUSG00000015461	Arnt	TF			3.965e-04	3.965e-04
ENSMUSG00000058135	mmu-miR-665-3p	miRNA-seq			-5.274e-01	-5.274e-01
gene	regulator	omic	area	representative	Group0	Group1

Showing 1 to 10 of 69 entries

Figure 4: Summary table and `MORE plots` button. The user can download the whole table by pressing the button `Download` (orange box)



Gene: ENSMUSG00000000078

Regulator: Mef2d

Values Regulator: Enter a vector (comma delimited) or NULL

Plot Per Omic: FALSE

Gene Colour: Orange

Regulator Colour: Blue

Order: TRUE

Title X axis: condA | condB

Values Continuous Variable: 1,2,3,4,5,6,7,8

Condition to plot: Enter a vector (comma delimited) or NULL

Figure 5: Inputs and plot for the pair gen–regulator (ENSMUSG00000000078–Mef2d). The user can download all plots by pressing the button `Download` (orange box)