

Users' Guide MORE

Sonia Tarazona Campos
Maider Agueralde Martin
Blanca Tomás Riquelme

October 16, 2023

Contents

1	Introduction	2
2	Getting started	3
3	Input data	3
4	Generating the regression models with MORE	5
4.1	Arguments for more() function	5
4.2	more output	8
4.3	Running an example	12
5	Retrieving significant regulations from MORE results	14
5.1	RegulationPerCondition input parameters	14
5.2	Interpreting RegulationPerCondition output with an example	14
6	Plotting MORE results	15
6.1	plotmore input parameters	15
6.2	Interpretation of MORE plots	16
7	How to use MORE with R Shiny	20
7.1	MORE Shiny application example	22
8	How to cite MORE package	23

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(GeneExpressionDE)
      Batch_1_Ctr_0H Batch_2_Ctr_0H Batch_4_Ctr_0H Batch_1_Ctr_2H Batch_3_Ctr_2H Batch_4_Ctr_2H Batch_1_Ctr_6H Batch_3_Ctr_6H Batch_4_Ctr_6H
ENSMUSG000000000028 13.812612 14.03159 13.901656 13.929891 13.876085 14.126789 14.171622 13.872382 14.199299
ENSMUSG000000000056 13.470168 13.29205 12.828267 12.623622 12.997942 12.535726 12.991138 13.113399 12.767678
ENSMUSG000000000078 10.682646 10.35727 10.501151 10.588055 10.451858 10.444312 11.008301 10.434204 9.839811
ENSMUSG000000000093 6.910878 6.69301 7.139726 8.094464 7.535058 8.683042 7.659847 7.736197 8.327506
ENSMUSG000000000131 14.241408 14.39647 14.345537 14.381840 14.401952 14.302514 14.607419 14.443613 14.562286
ENSMUSG000000000134 10.963512 10.94442 11.147515 10.809237 11.009085 10.662868 10.994689 10.936161 10.942475
      Batch_1_Ctr_12H Batch_3_Ctr_12H Batch_4_Ctr_12H Batch_1_Ctr_18H Batch_3_Ctr_18H Batch_4_Ctr_18H Batch_1_Ctr_24H Batch_3_Ctr_24H
ENSMUSG000000000028 13.947157 13.958910 14.16333 13.924163 13.771880 14.043439 14.08774 14.006538
ENSMUSG000000000056 12.910736 12.670369 12.51459 12.992014 12.812824 12.490432 13.02470 12.888504
ENSMUSG000000000078 10.432909 10.979929 10.16331 11.325414 11.212417 10.705251 11.40895 11.134874
ENSMUSG000000000093 7.206325 6.810825 7.58449 7.093517 7.248715 7.952783 7.83131 6.891414
ENSMUSG000000000131 14.428646 14.508502 14.40876 14.373203 14.255694 14.630406 14.48066 14.534599
ENSMUSG000000000134 10.929800 10.964434 10.77230 11.145207 11.125737 11.092071 11.09345 10.956520
      Batch_4_Ctr_24H Batch_1_Ik_0H Batch_2_Ik_0H Batch_4_Ik_0H Batch_1_Ik_2H Batch_3_Ik_2H Batch_4_Ik_2H Batch_1_Ik_6H Batch_3_Ik_6H Batch_4_Ik_6H
ENSMUSG000000000028 14.356622 13.726267 13.509564 13.826921 13.882908 13.981671 14.090817 14.248588 14.196102 14.133489
ENSMUSG000000000056 12.675347 13.519628 13.542471 13.013013 13.570048 13.370643 13.059786 13.939147 13.710870 13.350913
ENSMUSG000000000078 10.708153 10.903395 10.832873 10.366378 11.008338 10.907011 10.827029 11.475356 10.853306 11.080212
ENSMUSG000000000093 8.090818 5.909092 6.071665 5.430375 6.456153 7.079468 7.685181 7.726669 7.682333 9.293368
ENSMUSG000000000131 14.689092 14.277610 14.162259 14.283160 14.112652 14.356173 14.239701 14.224534 14.192340 14.065615
ENSMUSG000000000134 11.075132 11.083885 11.220484 11.193649 10.844606 10.597720 10.591733 10.344318 10.148949 10.572215
      Batch_1_Ik_12H Batch_3_Ik_12H Batch_4_Ik_12H Batch_1_Ik_18H Batch_3_Ik_18H Batch_4_Ik_18H Batch_1_Ik_24H Batch_3_Ik_24H Batch_4_Ik_24H
ENSMUSG000000000028 14.252858 14.180092 14.463167 14.004855 13.876694 13.774570 13.122652 13.25478 13.04647
ENSMUSG000000000056 14.278777 14.206229 14.129953 14.610906 14.352236 14.475980 14.536249 14.00465 14.59179
ENSMUSG000000000078 12.970246 12.950961 11.906972 13.845399 13.399489 13.019943 14.069741 13.96808 13.37502
ENSMUSG000000000093 8.237073 8.057052 8.362416 9.354684 9.470364 9.433288 9.908516 10.38476 10.00137
ENSMUSG000000000131 13.920578 14.088770 14.240393 13.373764 13.490624 13.319716 13.402803 13.61354 13.50323
ENSMUSG000000000134 10.368850 10.558487 10.099720 10.690735 10.598448 10.583839 10.640646 10.64785 10.87545
```

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. There is no restriction for the number of columns, but it must be taken into account that MORE will combine all the experimental covariates into a single variable. For instance, in the example below, the new single covariate would combine Time and Group2 to obtain the values: 1_0, 2_0, ..., 7_1, 8_1. Therefore, in this case, it makes more sense to exclude Time from the experimental design and just include the covariate Group2.

```
> edesign
      Time Ikaros
C0H_rep1    0    0
C0H_rep2    0    0
C0H_rep3    0    0
C2H_rep1    2    0
C2H_rep2    2    0
C2H_rep3    2    0
C6H_rep1    6    0
C6H_rep2    6    0
C6H_rep3    6    0
C12H_rep1   12    0
C12H_rep2   12    0
C12H_rep3   12    0
C18H_rep1   18    0
C18H_rep2   18    0
C18H_rep3   18    0
C24H_rep1   24    0
C24H_rep2   24    0
C24H_rep3   24    0
I0H_rep1    0    1
I0H_rep2    0    1
I0H_rep3    0    1
I2H_rep1    2    1
I2H_rep2    2    1
I2H_rep3    2    1
I6H_rep1    6    1
I6H_rep2    6    1
I6H_rep3    6    1
I12H_rep1   12    1
I12H_rep2   12    1
I12H_rep3   12    1
I18H_rep1   18    1
I18H_rep2   18    1
I18H_rep3   18    1
I24H_rep1   24    1
I24H_rep2   24    1
I24H_rep3   24    1
```

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(data.omics$miRNA)
      C.0.1 C.0.2 C.0.4 C.2.1 C.2.2 C.2.4 C.6.1 C.6.2 C.6.4 C.12.1 C.12.2 C.12.4 C.18.1
mmu-miR-126a-3p 5.928540 5.894740 6.129422 5.944441 6.124934 5.821428 5.836395 5.901268 5.937407 5.549141 5.3015202 5.507578 5.721821
mmu-miR-146a-5p 8.897409 8.821297 8.970465 9.064574 8.765698 9.272425 8.884604 8.640085 9.099084 8.876949 8.6946134 9.110137 8.985080
mmu-miR-149-5p 6.019200 6.005988 6.319872 6.139417 6.101543 5.617048 6.234469 5.699149 5.704736 5.993486 5.2159274 5.920638 6.109935
mmu-miR-151-3p 2.292712 2.999994 3.029195 3.059053 3.737898 3.244619 2.554377 2.403334 2.682804 2.053719 3.5042895 3.355643 2.886767
mmu-miR-152-5p 3.685103 3.463875 3.361227 2.864062 3.237542 3.445622 3.489312 2.584364 3.673261 3.280194 4.0722022 3.960440 3.851923
mmu-miR-152-3p 2.443389 2.111220 3.237679 1.598316 2.215593 2.895290 2.305020 2.676140 2.455047 1.411464 0.9177292 2.593676 2.062350
      C.18.2 C.18.4 C.24.1 C.24.2 C.24.4 I.0.1 I.0.2 I.0.4 I.2.1 I.2.2 I.2.4 I.6.1
mmu-miR-126a-3p 5.956604 6.238669 5.871480 6.083656 5.810169 7.553824 7.565700 7.545007 7.218401 7.624274 7.277898 7.043530
mmu-miR-146a-5p 8.897821 9.416885 8.550588 8.914520 9.046006 10.206891 10.239775 10.309064 9.999837 10.094911 10.482743 9.730803
mmu-miR-149-5p 5.717178 5.545810 6.014354 5.978960 6.094488 5.552401 5.488337 5.802932 5.917709 5.619711 5.300880 5.030013
mmu-miR-151-3p 2.596736 3.067459 2.271744 2.891071 2.798531 2.300836 1.577522 2.503443 2.333803 1.832808 2.611632 3.269841
mmu-miR-152-5p 3.581309 2.656558 3.494520 3.280588 3.311762 4.533935 4.561176 3.804664 3.968536 4.474504 4.672827 5.258265
mmu-miR-152-3p 2.394461 1.583293 2.548980 2.717757 2.484371 2.981937 2.495120 3.336771 3.089476 2.761697 2.790815 2.982509
      I.6.2 I.6.4 I.12.1 I.12.2 I.12.4 I.18.1 I.18.2 I.18.4 I.24.1 I.24.2 I.24.4
mmu-miR-126a-3p 7.478030 7.273539 6.933740 7.163386 7.028725 6.900495 6.801852 7.108389 6.129697 6.817801 6.834958
mmu-miR-146a-5p 9.750513 9.978647 9.372293 9.143953 9.616151 8.763435 8.818473 9.389263 8.392633 9.200723 9.129251
mmu-miR-149-5p 5.693333 5.526662 5.945054 5.875019 5.579455 6.395686 6.830532 6.352893 7.361691 7.290380 7.237634
mmu-miR-151-3p 2.915293 2.828587 3.715337 3.427472 3.594374 4.870808 4.952152 4.730803 4.357550 4.445414 5.132643
mmu-miR-152-5p 4.325155 5.052524 4.556959 4.685922 5.695175 4.886784 4.849114 5.148405 5.618543 5.619467 5.123283
mmu-miR-152-3p 3.230537 2.658470 3.404436 3.384151 2.752064 3.394418 3.331074 3.545058 3.597190 3.395399 3.485411
```

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(associations$miRNA)
      Ensembl.Gene.ID      miRNA
ENSMUSG00000021252_mmu-miR-1839-5p ENSMUSG00000021252      mmu-miR-1839-5p
ENSMUSG00000021252_mmu-miR-1843b-5p ENSMUSG00000021252      mmu-miR-1843b-5p
ENSMUSG00000007777_mmu-miR-669f-3p ENSMUSG00000007777      mmu-miR-669f-3p
ENSMUSG00000024442_mmu-miR-409-3p ENSMUSG00000024442      mmu-miR-409-3p
ENSMUSG00000024442_mmu-miR-693-3p ENSMUSG00000024442      mmu-miR-693-3p
ENSMUSG00000024442_mmu-miR-4661-3p ENSMUSG00000024442      mmu-miR-4661-3p
```

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 *GeneExpression* 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, epsilon = 0.00001, alfa = 0.05,
     family = gaussian(), elasticnet = NULL, interactions.reg = TRUE,
     min.variation = 0, min.obs = 10, col.filter = 'cor',
     correlation = 0.7, scaletype = 'auto', p.method = 'jack',
     vip = 0.8, method = 'glm')
```

4.1 Arguments for more() function

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.

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

alfa Significance level. By default, 0.05.

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

elasticnet 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.

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

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.

min.obs Minimum number of observations a gene must have to compute the GLM model. By default, 10.

col.filter 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 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.

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.

p.method This parameter is only required when adjusting a PLS model and is used 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 and compute then their associated p-value.

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 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.

MasterRegulators Vector of the top 10 regulators that relevantly regulate more genes.

HubGenes Vector of the top 10 genes that are relevantly regulated by more regulators.

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 0.8.

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 (RMSEE), 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.

MasterRegulators Vector of the top 10 regulators that significantly regulate more genes.

HubGenes Vector of the top 10 genes that are significantly regulated by more regulators.

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 master regulators will be plotted against the genes they regulate. By default, FALSE. It could be very time-consuming if the master 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 regulators for each.
6. Top 10 hub genes per omic and the number of relevant regulators for each.
7. Top 10 master regulators and the number of genes they regulate.
8. If required with `plot.more = TRUE` the plots of the master 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 = GeneExpressionDE,
associations = associations, data.omics = data.omics,
edesign = edesign[,-1, drop = FALSE], center = TRUE, scale = TRUE,
family = gaussian(), elasticnet = 0.5, interactions.reg = TRUE,
min.variation = NULL, min.obs = 10, omic.type = 0,
```

```
col.filter = 'cor', correlation = 0.7, method = 'glm')
```

The estimated coefficients of the relevant regulators in the final GLM model computed by **more** for the gene ENSMUSG00000000078 are

```
> SimMORE$ResultsPerGene$ENSMUSG00000000078$coefficients
              coefficient
(Intercept)    11.19681826
mmu-miR-674-3p -0.07629578
TF_mc1_1_R      0.71624672
Group1:TF_mc1_1_R 0.51890964
```

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 regulators mmu-miR-381-3p and mmu-miR-410-3p are correlated, and the last has been chosen as representative regulator (R). In addition, it is indicated that the correlation with the representative is positive (P). 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(SimMORE$ResultsPerGene$ENSMUSG00000000078$allRegulators)
      gene      regulator omic area      filter Rel
mmu-miR-381-3p ENSMUSG00000000078 mmu-miR-381-3p miRNA    mirRNA_mc1_2_P  0
mmu-miR-410-3p ENSMUSG00000000078 mmu-miR-410-3p miRNA    mirRNA_mc1_2_R  0
mmu-miR-674-3p ENSMUSG00000000078 mmu-miR-674-3p miRNA          Model  1
mmu-miR-466d-5p ENSMUSG00000000078 mmu-miR-466d-5p miRNA    TF_mc1_1_N  1
mmu-miR-1187    ENSMUSG00000000078 mmu-miR-1187 miRNA    TF_mc1_1_N  1
mmu-miR-669f-3p ENSMUSG00000000078 mmu-miR-669f-3p miRNA    TF_mc1_1_N  1
> tail(SimMORE$ResultsPerGene$ENSMUSG00000000078$allRegulators)
      gene      regulator omic area      filter Rel
Pou6f1    ENSMUSG00000000078 Pou6f1    TF promoter mirRNA_mc1_2_N  0
Rfxank    ENSMUSG00000000078 Rfxank    TF promoter TF_mc1_1_P  1
Satb1     ENSMUSG00000000078 Satb1    TF promoter mirRNA_mc1_2_N  0
Zfp692    ENSMUSG00000000078 Zfp692    TF promoter TF_mc1_1_N  1
TF_mc1_1_R ENSMUSG00000000078 TF_mc1_1_R TF promoter    Model  1
miRNA_mc1_2_R ENSMUSG00000000078 miRNA_mc1_2_R miRNA          Model  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(SimMORE)
```

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

```
> myresults[c(1:10,60:65,95:100),]
```

	gene	regulator	omic	area	representative	Group0	Group1
mmu-miR-342-3p	ENSMUSG00000000028	mmu-miR-342-3p	miRNA		Nr3c1	-0.18920	-0.2837
Bach1	ENSMUSG00000000028	Bach1	TF promoter		Nr3c1	-0.18920	-0.2837
Bach2	ENSMUSG00000000028	Bach2	TF promoter		Nr3c1	-0.18920	-0.2837
Crem	ENSMUSG00000000028	Crem	TF promoter		Sfp11	0.10670	0.2404
E2f2	ENSMUSG00000000028	E2f2	TF promoter		Sfp11	0.10670	0.2404
Elf1	ENSMUSG00000000028	Elf1	TF promoter		Nr3c1	-0.18920	-0.2837
Elk3	ENSMUSG00000000028	Elk3	TF promoter		Nr3c1	-0.18920	-0.2837
Ep300	ENSMUSG00000000028	Ep300	TF promoter		Nr3c1	-0.18920	-0.2837
Etv5	ENSMUSG00000000028	Etv5	TF promoter		Nr3c1	-0.18920	-0.2837
Foxo1	ENSMUSG00000000028	Foxo1	TF public		Nr3c1	-0.18920	-0.2837
Usf1	ENSMUSG00000000056	Usf1	TF promoter		Mxd4	0.52240	0.5224
Zbtb7b1	ENSMUSG00000000056	Zbtb7b	TF promoter		Mxd4	0.52240	0.5224
Zfp5131	ENSMUSG00000000056	Zfp513	TF promoter		Mxd4	0.52240	0.5224
Zfp7681	ENSMUSG00000000056	Zfp768	TF promoter		Mxd4	0.52240	0.5224
mmu-miR-674-3p	ENSMUSG00000000078	mmu-miR-674-3p	miRNA			-0.07630	-0.0763
mmu-miR-466d-5p1	ENSMUSG00000000078	mmu-miR-466d-5p	miRNA		Hivep2	-0.71620	-1.2350
Usf11	ENSMUSG00000000131	Usf1	TF promoter		Myc	0.27030	-0.4459
Zbtb7b2	ENSMUSG00000000131	Zbtb7b	TF promoter		Myc	0.27030	-0.4459
Zfp5132	ENSMUSG00000000131	Zfp513	TF promoter		Myc	0.27030	-0.4459
Zfp580	ENSMUSG00000000131	Zfp580	TF promoter		Myc	0.27030	-0.4459
Runx33	ENSMUSG00000000134	Runx3	TF promoter			0.00000	-0.1737
mmu-miR-188-5p	ENSMUSG00000000134	mmu-miR-188-5p	miRNA		Sp4	-0.04696	-0.2184

This table shows the relevant regulators for each gene. The **representative** column indicates if the regulator was chosen as the random 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 `Group0` corresponds to the first condition, and `Group1` 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 the function **plotGLM** 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, reguValues = NULL,
plotPerOmic = FALSE, gene.col = 1, regu.col = NULL, order = TRUE,
xlab = "", cont.var = NULL, cond2plot = NULL)
```

6.1 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.

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 Interpretation of MORE plots

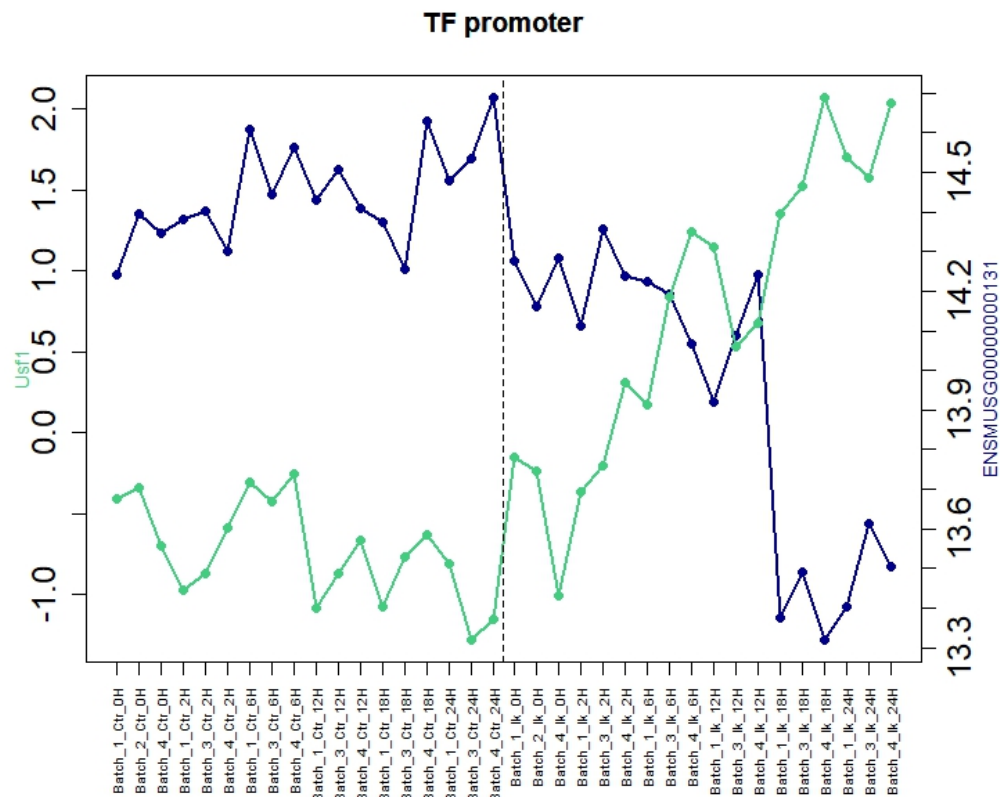
Following the previous example, the MORE graphic below represents the expression profile of a given gene (ENSMUSG00000000131) and the values for a relevant regulator of this gene (TF promoter regulator Usf1) and can be generated with the following code:

```
> plotmore(output = SimMORE,
  gene = "ENSMUSG00000000131",
  regulator = "Usf1",
  plotPerOmic = FALSE,
  gene.col = "blue4",
  order = FALSE,
  regu.col = "seagreen3")
```

The X-axis is divided into two conditions (1 or 2), and within each condition, the observations are displayed, which correspond to different time points in this case. 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).

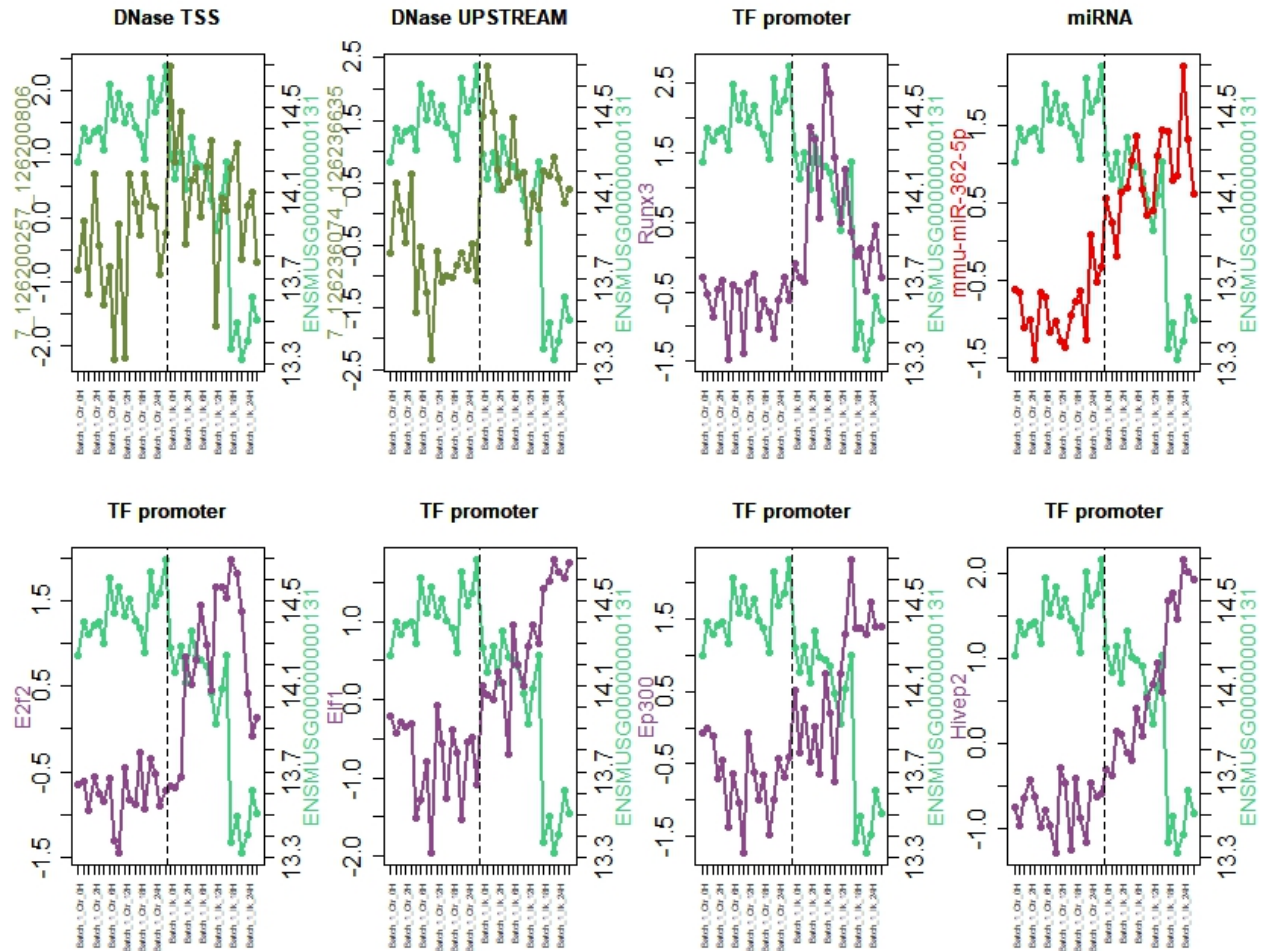
If we set the regulator argument to NULL, all the relevant regulators of gene ENSMUSG00000000131 will be plotted (18 regulators from three different omics: miRNA-seq, DNase-seq, and TF). Only 8 will be presented:

```
> par(mfrow = c(2,4))
> plotmore(output = SimMORE,
```

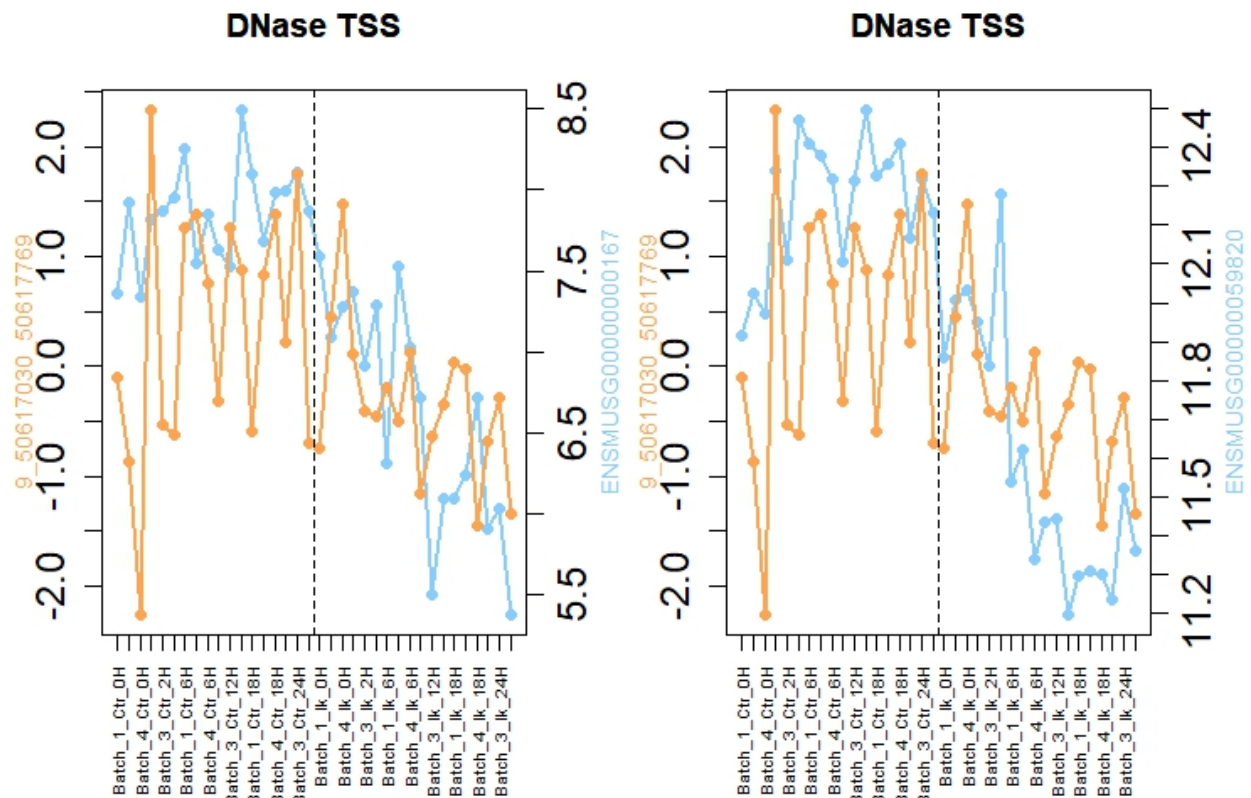
```
gene = "ENSMUSG00000000131",
regulator = NULL,
order =FALSE,
plotPerOmic = FALSE,
gene.col = "seagreen3")
```

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.



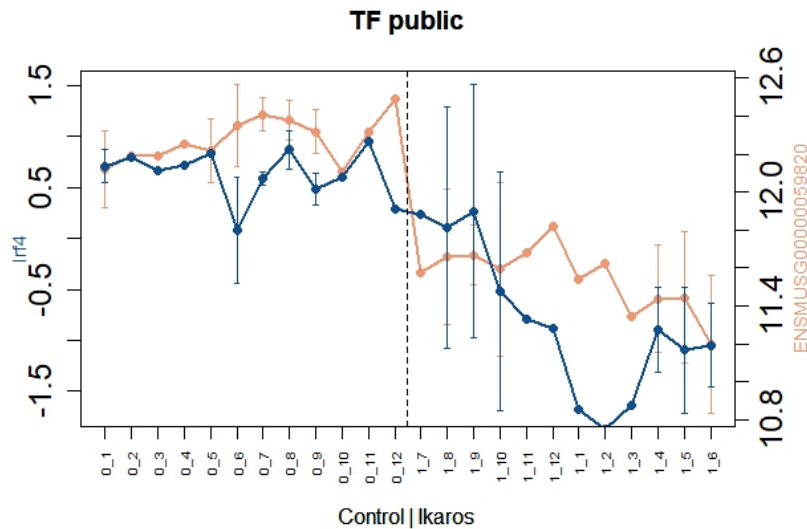
If we want to plot all the genes that are considered to be relevantly regulated by a given regulator (e.g. 9_50617030_50617769), we must set the gene argument to NULL as follows. In this case, the TF regulates two genes: ENSMUSG000000000167 and ENSMUSG000000059820.

```
> par(mfrow = c(1,2))
> plotmore(output = SimMORE,
  gene = NULL,
  regulator = "9_50617030_50617769",
  plotPerOmic = FALSE,
  order = FALSE,
  gene.col = "skyblue1",
  regu.col = "tan1")
```



Users can also define their own values for time points with the vector **cont.var**. In addition, they can assign a label for axis X to differentiate between two conditions, Control or Ikaros, which is the **xlab** parameter. 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 = "ENSMUSG000000059820",
  regulator = "Irf4",
  plotPerOmic = FALSE,
  gene.col = "lightsalmon2",
  regu.col = "dodgerblue4",
  cont.var = c(1,2,3,4,5,6,7,8,9,10,11,12),
  xlab = "Control_|_Ikaros")
```



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`.

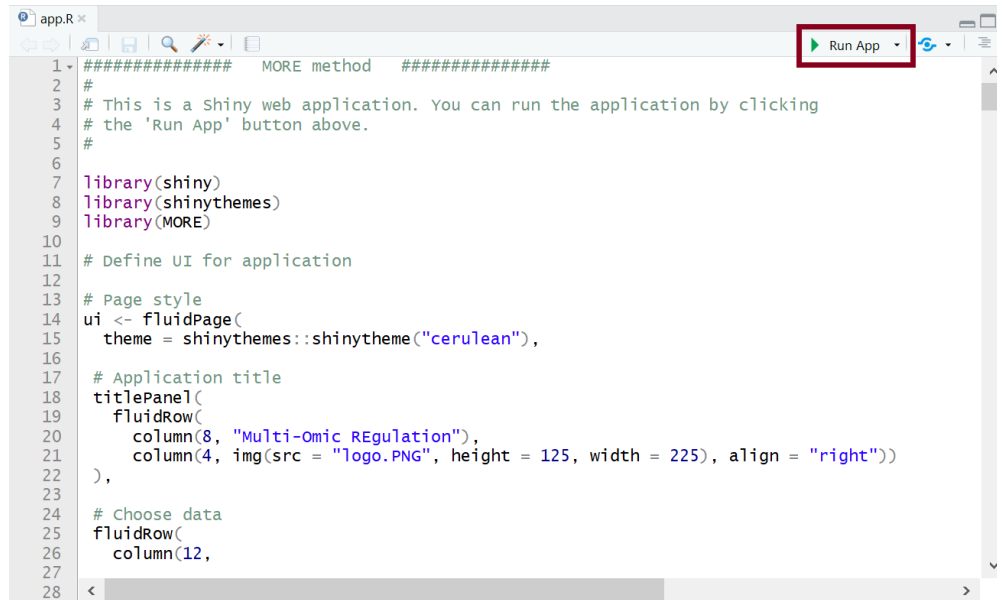
app.R Script to run the web application for MORE method. Users must open this file from RStudio to start using the application.

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).



```
1 ##### MORE method #####
2 #
3 # This is a Shiny web application. You can run the application by clicking
4 # the 'Run App' button above.
5 #
6
7 library(shiny)
8 library(shinythemes)
9 library(MORE)
10
11 # Define UI for application
12
13 # Page style
14 ui <- fluidPage(
15   theme = shinythemes::shinytheme("cerulean"),
16
17   # Application title
18   titlePanel(
19     fluidRow(
20       column(8, "Multi-Omic REGulation"),
21       column(4, img(src = "logo.PNG", height = 125, width = 225), align = "right"))
22   ),
23
24   # Choose data
25   fluidRow(
26     column(12,
27
28
```

Figure 1: Run MORE application, click Run App button

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

Multi-Omic REGulation

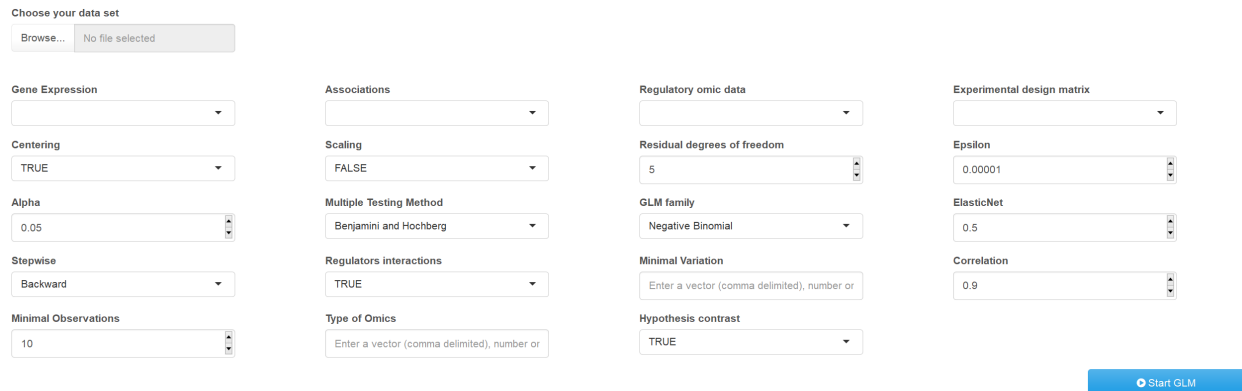


Figure 2: MORE Shiny

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>.

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 `RegulationPerCondition` 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`.

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 extension `.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.

Choose your data set

Browse... TestDataShiny.RData
Upload complete

Gene Expression
GeneExpressionDE

Centering
TRUE

Alpha
0.05

Stepwise
Backward

Minimal Observations
10

Associations
associations

Scaling
FALSE

Multiple Testing Method
Benjamini and Hochberg

Regulators Interactions
TRUE

Type of Omics
1,0,0

Regulatory omic data
data.omics

Residual degrees of freedom
5

GLM family
Negative Binomial

Minimal Variation
0

Hypothesis contrast
TRUE

Experimental design matrix
edesign

Epsilon
0.00001

ElasticNet
0.5

Correlation
0.9

Figure 3: MORE Shiny: inputs for running example

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**.

Show 10 entries

Search:

gene	regulator	omic	area	representative	Group0	Group1
ENSMUSG000000000078	mmu-miR-378a-5p	miRNA-seq			-1.224e-04	-1.224e-04
ENSMUSG000000000078	mmu-miR-139-3p	miRNA-seq			-3.021e-02	-3.021e-02
ENSMUSG000000000078	MeI2d	TF			-2.105e-05	-2.105e-05
ENSMUSG000000024873	mmu-miR-7081-5p	miRNA-seq			-3.052e-01	-3.052e-01
ENSMUSG000000024873	mmu-miR-34a-5p	miRNA-seq			-1.509e-04	-1.509e-04
ENSMUSG000000024873	mmu-miR-291a-5p	miRNA-seq			-8.512e-06	-8.512e-06
ENSMUSG000000024873	mmu-miR-879-5p	miRNA-seq			6.577e-04	6.577e-04
ENSMUSG000000015461	mmu-miR-376c-3p	miRNA-seq			3.343e-05	3.343e-05
ENSMUSG000000015461	Amt	TF			3.965e-04	3.965e-04
ENSMUSG000000058135	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

Previous 1 2 3 4 5 6 7 Next

MORE plots

Download

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

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.

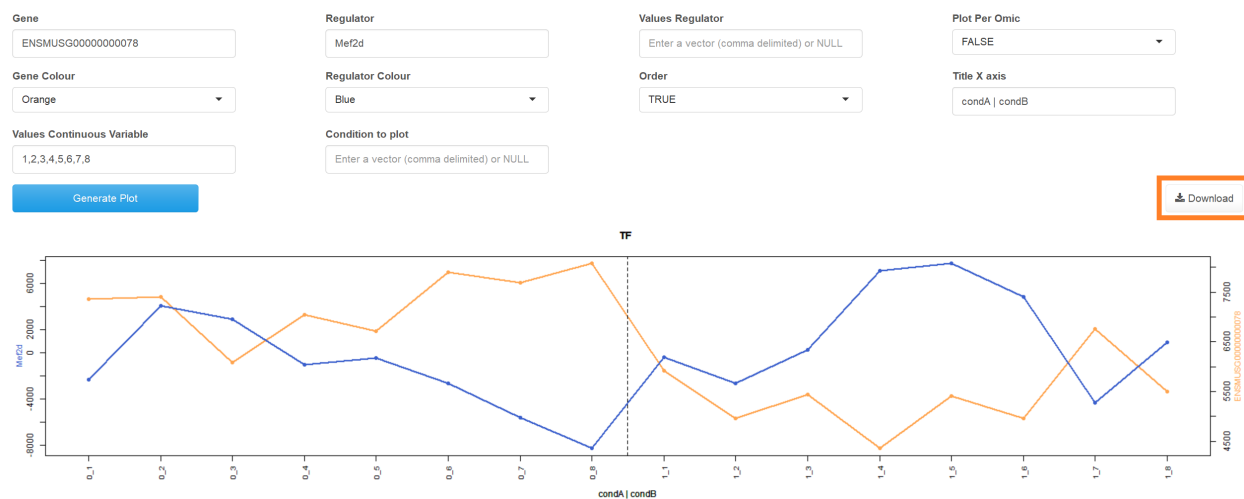


Figure 5: Inputs and plot for the pair gen–regulator (ENSMUSG00000000078–Mef2d). The user can download all plots by pressing the button Download (orange box)