MIPRIP 2.0 User Manual

January 13, 2019

This manual describes the functions of MIPRIP 2.0 R package and their usage.
MIPRIP 2.0 is a software package for R (www.r-project.org) to predict regulators of a gene of interest from gene expression profiles and known regulator binding information (from e.g. ChIP-seq/ChIP-chip databases). MIPRIP 2.0 can straightforwardly be applied to human and mouse gene expression data to study the regulatory relationships within one group of samples (single-mode), between two different groups (e.g. treatment vs. control) (dual-mode) and also for more than two groups (multi-mode).
For more details on the method, please see Poos *et al.* (2016 and 2019).

MIPRIP 2.0 author:

Alexandra Poos

Availability:

https://www.leibniz-hki.de/en/miprip.html and
https://github.com/network-modeling/MIPRIP

Publications:

Poos AM, Maicher A, Dieckmann AK, Oswald M, Eils R, Kupiec M, Luke B, König R (2016) Mixed Integer Linear Programming based machine learning approach identifies regulators of telomerase in yeast. *Nucleic Acids Research, 44, e93*

Poos AM, Kordaß T, Kolte, A, Ast V, Oswald M, Rippe K, König R (2019) Modelling *TERT* regulation across 19 different cancer types based on the MIPRIP 2.0 gene regulatory network approach, bioRxiv.

Manual author:

Alexandra Poos <Alexandra.Poos@med.uni-jena.de>

1. Installation

1.1. Required software

For installing and running MIPRIP 2.0 the following software must be installed:

a) **R** (version 3.5.1 or later); available from https://www.r-project.org/
b) **RStudio** (version 1.2.907 or later); available from https://www.rstudio.com/
c) **Gurobi** (version 8.0.1 or later); available from http://www.gurobi.com/index

For the use of Gurobi a licence is necessary. For academic usage this licence is free and can be obtained after registration from http://www.gurobi.com/downloads/download-center (for more details see Quick Start Guide (http://www.gurobi.com/documentation/)). You have to make sure that the activation (grbgetkey) of the Gurobi software is within the university/academic network. Otherwise, an online course licence can be obtained, which is limited to 2000 constraints per model.

After installation and activation (grbgetkey) of the Gurobi software, the Gurobi R application programming interface (API) has to be installed. The easiest way to install this is by using Rstudio:

(Tools → Install Packages → Install from Package Archive File).

For win64, the package archive file can be found at: "C:/gurobi<VERSION>/win64/R/gurobi_<VERSION>.zip";

analogous for Mac OS X

at "/Library/gurobi<VERSION>/mac64/R/gurobi_<VERSION>.tgz ".

For Linux you can find the package archive file at "/opt/<USER>/gurobi<VERSION>/linux64/R/gurobi_<VERSION>_R_x86_64_<GNU_VERSION>.tar.gz".

In addition, you need to install the package "slam" using the R command "install.packages("slam")".

After installing the Gurobi R-API, make sure that the library can be loaded on the R console. For this simply type: library("gurobi").

1.2. Installation of MIPRIP 2.0

To install MIPRIP 2.0, download the package from https://www.leibniz-hki.de/en/miprip.html or https://github.com/network-modeling/MIPRIP.

On a Unix/Linux system, execute the following command from a shell

```
R CMD INSTALL MIPRIP_2.0.tar.gz
```

or from the R command line

```
install.packages("MIPRIP_2.0.tar.gz")
```

Or you can also install it directly from RStudio similar to the Gurobi R API (click on Tools → Install Packages → Install from Package Archive File and select the file MIPRIP_2.0.tar.gz).

1.3. Loading MIPRIP

To load the package within the R command line simply type:

```
library("MIPRIP2")
```

2. Running MIPRIP 2.0

MIPRIP 2.0 is an extension of the previous MIPRIP package and is now applicable also for human and mouse gene expression data. Compared to the first version, MIPRIP 2.0 can deal with weighted edges and the user can distinguish between three different modes:

(1) **single-mode**: identifies the most important regulators of a gene of interest in one group of samples

(2) **dual-mode**: compares the regulatory processes of the gene of interest between two different groups or conditions (e.g. treatment vs. control)

(3) **multi-mode**: identifies the most common but also group-specific regulators of more than two groups

2.1. Preprocessed data provided with the MIPRIP 2.0 package

- Generic human regulatory network
- Generic mouse regulatory network

- z-transformed gene expression data of 115 melanoma skin cancer samples from The Cancer Genome Consortium (TCGA) (the whole dataset with real sample IDs is freely available at the TCGA Genome Data Analysis Center (GDAC, http://gdac.broadinstitute.org/))
- *TERT* promoter mutation status (mutated (group1) or wild-type (group2)) of the 115 melanoma skin cancer samples (from (Cancer Genome Atlas, 2015))

2.2. Workflow of MIPRIP 2.0

The basic idea of MIPRIP is to identify the most relevant regulators of a particular target gene by predicting the target gene's expression using all potential regulators putatively binding to its promoter based on a Mixed Integer linear Programming based approach. The gene expression value $\tilde{g}_{i,k}$ of the target gene i is predicted for each sample by the following model:

$$\tilde{g}_{i,k} = \beta_0 + \sum_{t=1}^{T} \beta_t \cdot es_{ti} \cdot act_{tk} \qquad (1)$$

where β_0 is an additive offset, T the number of all regulators binding to the gene’s promoter, β_t is the optimization parameter for regulator t, es_{ti} is the edge strength score between regulator t and its putative target gene i and act_{tk} the activity of regulator t in sample k.

Input data: The algorithm needs gene expression data and a regulatory network with all the regulator to target gene interactions as input. A regulatory network for human, mouse or yeast can be downloaded from https://www.leibniz-hki.de/en/miprip.html or https://github.com/network-modeling/MIPRIP.

The gene expression data should look like this. In the columns are the samples and in the rows are the genes (official gene symbol only). For the dual- and the multi-mode, MIPRIP 2.0 expects a list with two or more gene expression matrices as input (see Example).

The algorithm proceeded as follows:

(1) **Preprocessing of the data**:
   a. check if the genes in the expression dataset and in the network are matching
   b. all samples with no expression value for the gene of interest are removed

(2) **Activity calculation**: Instead of using the gene expression value of a regulator, we calculate an activity value act_{tk} for each regulator and each sample based on the expression of all its putative target genes g_{ik} by

$$act_{tk} = \frac{\sum_{i=1}^{n} es_{ti} \cdot g_{ik}}{\sum_{i=1}^{n} es_{ti}} \tag{2}$$

The activity is the cumulative effect of a regulator on all its target genes, normalized by the sum of all target genes. To calculate the activity value, we excluded the expression value of the gene of interest itself.

Please note: you can also provide your own activity matrix (Act). Please, make sure than that the dimensions of the gene expression matrix, the network and the activity matrix are matching.

(3) **Mode selection**: the user selects between the single, dual or multi-mode based on the number of different groups/datasets.

(4) **Modelling step**: All linear equations are optimized using the Gurobi optimizer to minimize the difference between the measured transcript level (from the gene expression matrix) $g_{i,k}$ and the predicted gene expression $\tilde{g}_{i,k}$ value. This equals to minimizing the error terms e_{ik} in

$$min \sum_{k=1}^{l} \left| g_{ik} - \tilde{g}_{i,k} \right| = \sum_{k=1}^{l} e_{ik}. \tag{3}$$

To avoid overfitting, for each dataset we constructed models constraining the number of regulators starting with one regulator up to e.g. 10 regulators. By default, the model performs a ten-times threefold cross-validation yielding 300 models for each dataset. The correlation between the measured and the predicted gene expression values from the models indicates the prediction performance.

(5) **Statistical analysis**:

- Single mode: none
- Dual-mode: a Fisher's Exact Test is performed to identify significant regulators between the two groups

- Multi-mode:
    a. The most common regulators of all groups are identified based on a Rank product test.
    b. For each group a one-sided Wilcoxon test is performed with the regulator frequencies with one group vs. all other groups, which leads to the group-specific regulators.

To run MIPRIP 2.0, use the following R command:

```
miprip.result <- miprip.run(mode=c("single", "dual", "multi"), group_names=c(), target_gene,
          num_repeats=10, num_cv=3, num_parameter=10,
          gurobi_parameter=list(timeLimit = 5, OutputFlag=0), X=expression, ES=network)
```

2.2.1. Parameters

The following parameters MUST be specified within the function miprip.run:

- mode=c("single", "dual", "multi"): specifies which MIPRIP mode is used
- group_names=c(): vector of the group names (only if dual- or multi-mode is used)
- target_gene: specifies the official gene symbol of the target gene (please check that this gene is present in the network and in your expression dataset)
- num_repeats: number of repeats; how often the cross-validation (defined with num_cv) should be repeated (default=10)
- num_cv: specification of cross-validation runs, e.g. num_cv=3 for a three-fold cross-validation (default=3)
- num_parameter: number of maximal parameters used for the modelling (default=10)
- gurobi_parameter: setting the parameters for the Gurobi optimization software, e.g. TimeLimit (otherwise a default value of 5s per modelling step is used; further details about the parameters can be found at https://www.gurobi.com/documentation/8.0/refman/parameters.html#sec:Parameters)
- X: defines the name of the expression matrix or list (default=expression)
- ES: defines the name of the network object (default=network)

See the help for further information: ?miprip.run

3. Output of MIPRIP 2.0

A summary of the results of MIPRIP 2.0 are saved in an RData object during the modelling. Boxplots of the performances are saved as pdfs.

3.1. Using the single-mode:

A detailed output is saved in the file "MIPRIP_results_singleMode_<target_gene>.RData". To have a look at the details, load the file into a R session:

```
result=load("MIPRIP_results_singleMode_<target_gene>.RData")
```

- $frequency_group1: shows how often each regulator was chosen by the model for all regulator combinations for all runs
- $performance_single: correlation between the predicted value and the gene expression value from the dataset for all cross-validation runs and repeats
- $results_complete_group1: results of group1 for all cross-validation runs and repeats. For each run a table with the number of parameters, the chosen regulators, the beta values of these regulators plus beta zero as well as the performance (Pearson correlation between the predicted and the gene expression values in the dataset) of the model is listed
- $predictions_group1: predicted value of each validation sample over all tested regulator combinations compared to the gene expression value of the dataset

The performance of all the runs is plotted for each number of regulators to see which number of parameters leads to the best modelling result (saved as pdf "Performance_singleMode_<target_gene>.pdf").

3.2. Using the dual-mode:

The main output of MIPRIP 2.0 using two groups is a table with significant regulators of group1 compared to group2. For this a Fisher Exact Test with the regulator frequencies (how often the regulators were used over all combinations and all cross-validation runs) is performed. The table contains the frequency of the regulators in group1 and group2, the p-value and the Benjamini-Hochberg corrected p-value for each regulator (for more details see section 2.2.). This table and a boxplot with the overall performance of group1 compared to group2 is saved in a separate file, while all details about all models can be found in the file "MIPRIP_results_dualMode<target_gene>.RData".

3.3. Using the multi-mode:

The output of the multi-mode MIPRIP 2.0 analysis is similar to the single- and the dual-mode analysis. Instead of the significant regulators between group1 and group 2, the multi-mode analysis provides

(1) a table with the common regulators of the target gene over all groups.

(2) one table for each group with the significant regulators of this group compared to the other groups based on the regulator frequencies.

4. Example

In the following an example based on our performed melanoma skin cancer case study is described. All necessary data is available on our website and using the data, the example can be easily repeated. Here, we compare the regulation of *TERT* between melanoma samples with a *TERT* promoter mutation and melanoma samples with wild-type *TERT* promoter.

4.1. Running MIPRIP 2.0 with the example data (step by step)

(1) Loading the packages: After opening R from the terminal or using RStudio, the packages “gurobi” as well as “MIPRIP2” have to be loaded:

```
library(gurobi)
library(MIPRIP2)
```

(2) Loading of the dataset and the human generic regulatory network with all the transcription factor to target interactions.

```
skcm<-read.delim(file="Example_expression_data.tsv", header=TRUE,
check.names=FALSE)

network=read.delim(file="Generic_regulatory_network_human.tsv", header=TRUE,
check.names=FALSE)
```

(3) For a classification into groups:

```
annot=read.table(file="Annotation_table.txt", header=TRUE)
#1=mutated, 2=wild-type
```

```
mut=skcm[,colnames(skcm) %in% annot[,1][annot[,2]== "group1"]]
wt=skcm[,colnames(skcm) %in% annot[,1][annot[,2]== "group2"]]

both=list(mut, wt)
# for the dual- and the multi-mode MIPRIP 2.0 expects a list with an expression matrix
for each group/dataset/condition
```

(4) Running MIPRIP 2.0 with dual-mode:

```
example= miprip.run(mode="dual", group_names=c("mutated", "wildtype"),
target_gene="TERT", num_repeats=3, num_cv = 3, num_parameter = 5, X=both)
```

4.2. Results:

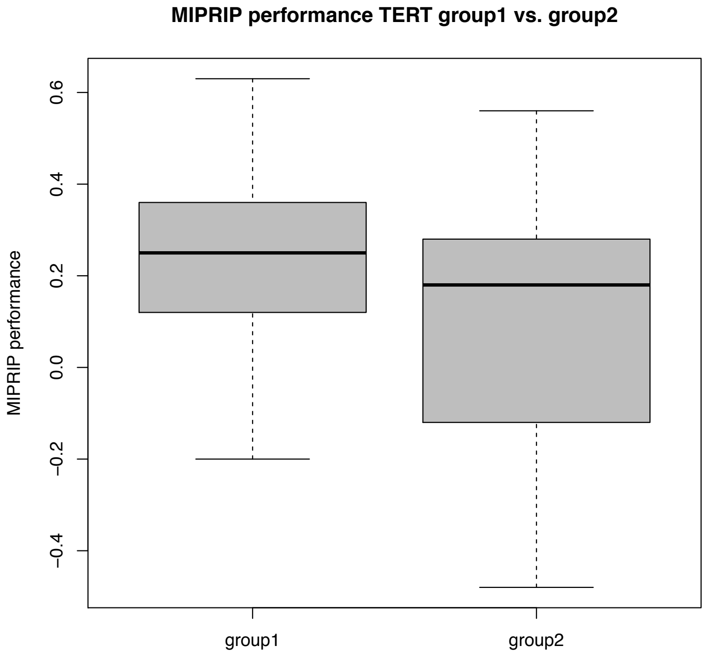
The results of both groups over all runs are saved in the file "MIPRIP_results_dualMode_TERT.RData".

The significant regulators (corrected p-value < 0.05) of group1 (samples with a *TERT* promoter mutation) compared to group2 (samples with wild-type *TERT* promoter) are printed on the screen and are also saved in the table "Significant_regulators_<target_gene>_ <group_names[1]>_vs_<group_names[2]>.txt".

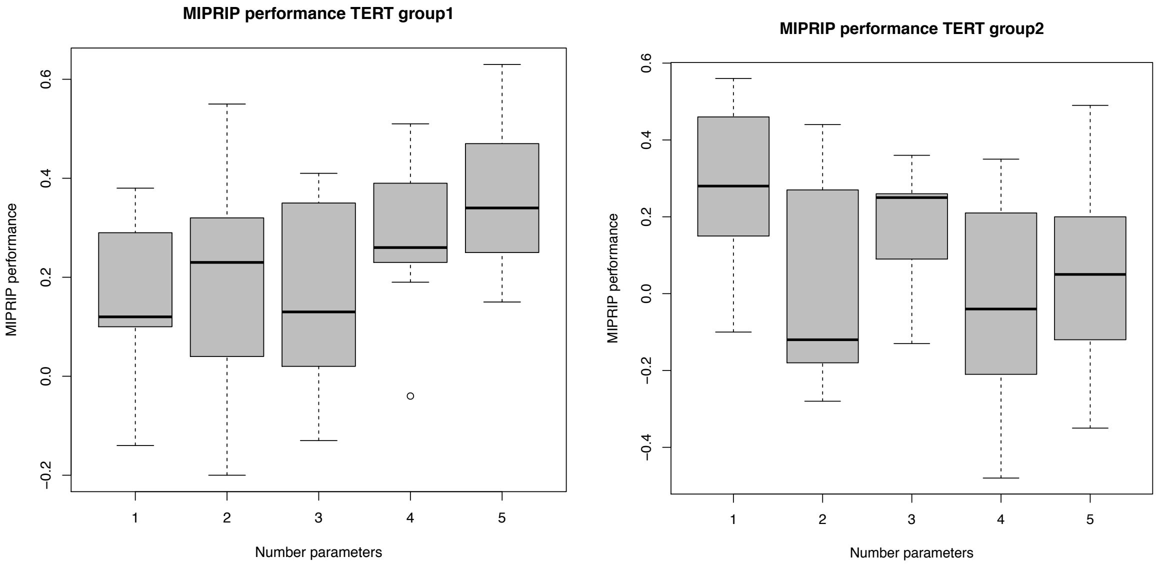
TF	Frequency_group1	Frequency_group2	p-value	p-value_BH
WT1	0	14	3.69e-05	2.77e-03
AR	17	2	1.63e-04	6.12e-03
ETS1	16	2	3.65e-04	9.12e-03
HIF.1	0	10	1.12e-03	2.04e-02
HMGA2	7	22	1.36e-03	2.04e-02

Hint: Compared to our case study as described in Poos *et al.* (2019), we run this example with less repeats and a smaller number of parameters because otherwise it will take several hours. With these parameter restrictions, the example can be run within 10-15 minutes on a local computer.

Besides the table, a boxplot with the performance over all models of group1 and group2 is saved as a separate file. For the example data with the simplified parameter settings it looks like this:



Furthermore, the performance of all models with a different number of regulators is plotted for each group. This shows which number of regulators leads to the best performance.



The detailed results of all runs and for both groups can be found in "MIPRIP_results_dualMode_TERT.RData". It is shown here exemplarily for group1.

```
>example$result_complete_group1[[1]]
```

```
                    TF
```

```
2                   E2F2
3               ETS1, TAF1
4           AR, ETS1, MYCN
5       EGR1, ETS1, HEY1, MITF
6 AR, BCL11A, MAZ, REST, TFAP2C
                                                                                Betas_TF
2                                                          0.843909641668916;-0.0142296755027596
3                                           3.76189029227422;-26.9876047229246;0.101857546042487
4    -1.56672735622951;1.14242031857615;1.3664982275431;-0.0077283925615985
5                                                                    2.53877785804811;6.27232204810903;-
71.2504270635463;0.769465773719372;0.188003243364635
6                                                                    -5.33395308660083;24.007050791514;-
9.73968775338207;86.8801181555237;1.10611890333321;0.188960006991155

 Amount TF           correlation
2        1          0.287679246226749
3        2          0.271192848989937
4        3          0.407774715936029
5        4          0.246441217834961
6        5          0.251351048995436

…

$frequency_group1
           [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
AP-2        "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"
AR          "2"  "2"  "3"  "3"  "2"  "1"  "0"  "3"  "1"
BATF        "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"
BCL11A      "1"  "3"  "0"  "0"  "0"  "0"  "0"  "0"  "1"
BHLHE40     "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"
CEBPA       "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"
CTCF        "0"  "0"  "0"  "0"  "1"  "0"  "0"  "0"  "0"
CTCFL       "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"
E2F1        "0"  "0"  "2"  "1"  "0"  "0"  "2"  "1"  "1"
E2F2        "1"  "0"  "1"  "1"  "1"  "0"  "0"  "2"  "0"
…
```

```
$predictions:group1[[1]]
                      1           2           3          4           5     g_real
Sample_109  0.418786007  0.337043524  0.116013903  1.1203121  0.06019459 -0.14136761
Sample_112 -0.604341569  0.641313465 -0.179251198  0.3111869 -0.52819262 -1.32567450
Sample_2    0.002135515  0.365319135  0.175612904 -0.3048363  0.54415186  0.95997318
Sample_25  -0.580074581 -0.309993838 -0.395754658 -0.6634316  0.06261036 -0.11797294
Sample_26  -0.405624637 -0.255126089 -0.533583973 -0.9974604 -0.26327050  0.39921397
...
```

```
$performance_group1
     1     2    3    4     5    6    7     8    9
1 0.29 -0.14 0.38 0.12 -0.09 0.11 0.10  0.29 0.19
2 0.27  0.00 0.55 0.32  0.23 0.17 0.04 -0.20 0.41
3 0.41 -0.10 0.36 0.35 -0.13 0.12 0.02  0.13 0.28
4 0.25 -0.04 0.51 0.45  0.26 0.23 0.19  0.39 0.36
5 0.25  0.19 0.47 0.34  0.35 0.32 0.15  0.61 0.63
```

Acknowledgements:

Thanks to Rainer König, Volker Ast, Marcus Oswald, Theresa Kordaß and Amol Kolte for their support.

References:

Cancer Genome Atlas, N. Genomic Classification of Cutaneous Melanoma. *Cell* 2015;161(7):1681-1696.