# GENIE3: documentation

Author: Vân Anh Huynh-Thu, vahuynh@uliege.be

This is the documentation for the R implementation (C wrapper) of GENIE3. This implementation is a research prototype and is provided "as is". No warranties or guarantees of any kind are given.

The GENIE3 method is described in the following paper:
Huynh-Thu V. A., Irrthum A., Wehenkel L., and Geurts P. (2010) Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE*, 5(9):e12776.

This R implementation wraps a C code, and was developed for the SCENIC pipeline to analyze single-cell RNA-seq data (Aibar Santos *et al.*, 2017. Nature Methods, 14:1083-1086).

The source code is provided in the two files "GENIE3.R" and "GENIE3.c".

## 1 Installation

1. You will have to install R:
   http://www.r-project.org

2. Compile the C code by typing the following command (you must be in the directory where "GENIE3.c" is located):

   ```
   R CMD SHLIB GENIE3.c
   ```

   This will create two files GENIE3.o and GENIE3.so

3. You must also install the following R packages, all available from CRAN:

   - *reshape2*
     https://cran.r-project.org/web/packages/reshape2/index.html
   - *doRNG*
     https://cran.r-project.org/web/packages/doRNG/index.html
   - *doParallel*
     https://cran.r-project.org/web/packages/doParallel/index.html

   This installation can be done from R with these commands:

   ```
   install.packages("reshape2")
   install.packages("doRNG")
   install.packages("doParallel")
   ```

# 2 Load the GENIE3 package

```
source("GENIE3.R")
```

This command will load the source file.

# 3 Run GENIE3

## Load the data

A file 'data.txt', containing an example of expression dataset (10 genes, 136 conditions), is provided for this tutorial. To load this data:

```
expr.matrix <- read.expr.matrix("data.txt", form="rows.are.samples")
```

This command will read the expression matrix from the file. The command will automatically recognize if gene names and/or sample names are provided in the first row and/or column. The *form* parameter is important to set correctly. It tells if every row in the expression file corresponds to a gene ("rows.are.genes"), or if every row corresponds to a sample ("rows.are.samples").
Additional function parameters and information are explained in the source file.

## Run GENIE3 with its default parameters

The only mandatory input argument of the function $GENIE3()$ is the gene expression matrix:

```
weight.matrix1 <- GENIE3(expr.matrix)
```

This command computes the weighted adjacency matrix of the gene network with the GENIE3 algorithm. In the weight matrix, element $(i, j)$ (row $i$, column $j$) gives the weight of the link from regulatory gene $i$ to target gene $j$, with high scores corresponding to more likely regulatory links.

## Restrict the candidate regulators to a subset of genes

You can specify that only a subset of the genes are to be used as candidate regulators, by passing an array of gene indices through the *regulators* parameter:

```
weight.matrix2 <- GENIE3(expr.matrix, regulators=3:5)
```

Here, for example, only genes from 3 to 5 (included) will be used as candidate regulators. This can be useful when you know which genes are transcription factors. If you have a list of gene names to be used as candidate regulators, you can also use it like this:

```
input.genes <- c("GATA5","XRCC2","OSR2","RAD51")
weight.matrix3 <- GENIE3(expr.matrix, regulators=input.genes)
```

### Change the tree-based method and its settings

```
# Use the Extra-Trees as tree-based method
tree.method <- "ET"

# Number of randomly chosen candidate regulators at each node of a tree
K <- 7

# Number of trees per ensemble
ntrees <- 50

# Run the method with these settings
weight.matrix4 <- GENIE3(expr.matrix, ...
  tree.method=tree.method, K=K, ntrees=ntrees)
```

### Obtain more information

Additional function parameters and information are explained in the source file.

## 4 Write the predictions

### Get the predicted ranking of all the regulatory links

```
link.list <- get.link.list(weight.matrix1)
head(link.list)
```

The output will look like this:

```
##  regulatory.gene target.gene    weight
## 1           XRCC2        TBX3 0.6066478
## 2            TBX3       XRCC2 0.5961631
## 3           CREB5        CD93 0.4408768
## 4            CD19       RAD51 0.4031206
## 5            CD93       CREB5 0.3886690
## 6           GATA5       CREB5 0.2937685
```

Each line corresponds to a regulatory link. The first column shows the regulator, the second column shows the target gene, and the last column indicates the score of the link.
Note that the ranking that is obtained will be slightly different from one run to another. This is due to the intrinsic randomness of the Random Forest and Extra-Trees methods. The variance of the ranking can be decreased by increasing the number of trees per ensemble.

### Get the first 5 links only

```
link.list <- get.link.list(weight.matrix1, report.max=5)
```

## Get the links with a score above 0.1

```
link.list <- get.link.list(weight.matrix1, threshold=0.1)
```

**Important note on the interpretation of the scores:** The weights of the links returned by *GENIE3()* **do not have any statistical meaning** and only provide a way to rank the regulatory links. There is therefore no standard threshold value, and caution must be taken when choosing one.

## Obtain more information

Information about function parameters are explained in the source file.