DiCoExpress: how to use the functions

ANALYSIS TUTORIAL USING THE BRASSICA NAPUS RNASEQ DATASET

Ilana Lambert, Christine Paysant-Le-Roux, Stefano Colella and Marie-Laure Martin-Magniette

2019-12-17

Aim of this document

We present in this document the analysis we performed using DiCoExpress on RNAseq transcriptome data published by Haddad et al. (2019). In this study, the authors studied the effects of silicon (Si) supply on the roots and mature leafs transcriptome in *Brassica napus L*. The *B. napus* RNAseq dataset is available at Brassica napus Genoscope.

This tutorial will allow a novel user to run data analysis in DiCoExpress using this example dataset and following our steps of the analysis in the paper. Note that some parts of the

script can be customized by the user for the analysis of his dataset. These arguments are found before the command line to launch a given function. Please, refer to the Reference Manual for details on arguments options.

Dataset description

This dataset contains roots and mature leafs expression data from *B. napus* with or without Si treatment. Three biological replicates are available for each data point. The two input files are available in the directory *DiCoExpress/Data*. We also added the annotation of *B. napus* v.5 downloaded from the Brassica Genome database in this same directory.

To begin

The script file that is used to run a given analysis is placed in the *Template_scripts* directory. In this example, we named it: **DiCoExpress_Brassica_napus.R**. The *Template_scripts* directory has to be the R working directory. If it is not the case, an error will appear to alert that the paths are not defined correctly. Once the directory is set the user can run this code:

```
source("../Sources/Load_Functions.R")
Load_Functions()

Working_Directory <- ".."
Data_Directory <- paste0(Working_Directory,"/Data")
Results_Directory <- paste0(Working_Directory,"/Results")
```

This part of the code will load all the R-packages needed to use the DiCoExpress workspace and set the *Data* and *Results* directories.

Load input files

For a correct automated load of all input files, the user should choose a project name that will be present as a prefix of the data files. In this way, the `Project_Name` in the code will allow DiCoExpress to find and load all input files of the given project:

```
Project_Name <- "Brassica_napus"
```

A filter option is available to choose a subset of samples to analyze. If `Filter = NULL`, the whole dataset is analyzed.

```
Filter=NULL

Data_Files <- Load_Data_Files(Data_Directory, Project_Name, Filter, Sep="\t")

Project_Name <- Data_Files$Project_Name
Target <- Data_Files$Target
Raw_Counts <- Data_Files$Raw_Counts
```

Running this function will return a list of 3 elements that are saved in three different R objects. They correspond to the Project_Name, the Target table, and the Raw Counts table.

[Optional] Using the filter argument: an example

The filter argument is a list of filter rules and a new project name for the filtered dataset. To use the filter argument, the user will have to add a line to the code with the following format:

```
Filter=list(c("Name_of_factor", "level_of_this_factor", TRUE/FALSE),"New_Proj
```

The filter rules are described by 3 characters: the name of the factor, the level of this factor, and TRUE or FALSE. TRUE states that the level of this factor is kept and FALSE it is removed. The last element of the list must be a string of characters to give a name to the project on the filtered dataset.

If in the *B. napus* dataset, the user wants to focus as an example only on the roots samples, without modifying the input files, he can use the `Filter` argument in two ways:

In the first case, the level “Root” of the “Tissue” factor is kept (TRUE) and the new project name is “Brassica_napus_Root”.

```
Filter=list(c("Tissue", "Root", TRUE),"Brassica_napus_Root")

Data_Files <- Load_Data_Files(Data_Directory, Project_Name, Filter, Sep="\t")

Project_Name <- DataFiles$Project_Name

Target <- Data_Files$Target
```

```
Raw_Counts <- Data_Files$Raw_Counts
```

In the second case, the level “MatureLeaf” of “Tissue” factor is deleted (FALSE) and the new project name is “Brassica_napus_Root”.

```
Filter=list(c("Tissue", "MatureLeaf", FALSE),"Brassica_napus_Root")

Data_Files <- Load_Data_Files(Data_Directory, Project_Name, Filter, Sep="\t")

Project_Name <- DataFiles$Project_Name
Target <- Data_Files$Target
Raw_Counts <- Data_Files$Raw_Counts
```

Quality control

The user will specify if an annotation file is available for the project, and DiCoExpress will load it during the analysis.
If no annotation file is available, the user will have to write:

```
Annotation_FileName <- NULL
```

If an annotation file is available, it has to be saved in the *Data* directory and its name specified as `Annotation_FileName` in the script as follows:

```
Annotation_FileName <- "Brassica_napus_Genome_Annot_20190811.txt"
```

In the Quality_Control function, 4 arguments can be changed. The defaults proposed in DiCoExpress are as follows:

```
Filter_Strategy="NbConditions"
Color_Group=c("darkolivegreen3","darkgreen","tan3","darkorange4")
CPM_Cutoff=1
Normalization_Method="TMM"

QualityControl(Data_Directory, Results_Directory, Project_Name, Target,
               Raw_Counts, Annotation_FileName, Filter_Strategy, Color_Group,
```

```
                CPM_Cutoff, Normalization_Method)
```

The Quality_Control function returns 7 output files saved in the directory *Results/Brassica_napus/Quality_Control*:

- **Brassica_napus_Normalization_Results.txt**
- **Brassica_napus_Low_counts_genes.txt**
- **Brassica_napus_NormCounts.txt**
- **Brassica_napus_NormCounts_log2.txt**
- **Brassica_napus_NormCounts_Mean_SD.txt**
- **Brassica_napus_NormCounts_log2_Mean_SD.txt**
- **Brassica_napus_Data_Quality_Control.pdf**

The user can check the number of genes before and after the filtering by open in a text editor the output file **Brassica_napus_Normalization_Results.txt** or can directly visualize these informations in the console pane:

```
## ################################################
## Filtering
## ################################################
##
## #### Description of Raw counts table ####
## Number of samples: 12
## Number of genes: 52962
##
## Number of features discarded by the filtering: 9233
## Number of analyzed genes after filtering: 43729
##
## ################################################
## Normalization
## ################################################
##

## Normalization factors:
##  0.7492924 0.7367225 0.7628112 0.7961395 0.8240471 0.8025353 1.246178 1.25
```

Differential expression analysis

The generalized linear model (glm) used to perform differential expression analysis is defined with the GLM_Contrasts function by specifying the arguments `Replicate` and `Interaction`.

Let μ_{tcr} be the log of the mean expression of a gene in Tissue t treated with Treatment c for the Replicate r.

- If `Replicate=FALSE` and `Interaction=FALSE`,

$$\mu_{tcr} = Intercept + T_t + C_c$$

- If `Replicate=TRUE` and `Interaction=FALSE`,

$$\mu_{tcr} = Intercept + T_t + C_c + R_r$$

- If `Replicate=FALSE` and `Interaction=TRUE`,

$$\mu_{tcr} = Intercept + T_t + C_c + (TC)_{tc}$$

- If `Replicate=TRUE` and `Interaction=TRUE`,

$$\mu_{tcr} = Intercept + T_t + C_c + R_r + (TC)_{tc}$$

where T_t states for the effect of Tissue, C_c the treatment effect, R_r the replicate effect and $(TC)_{tc}$ the interaction between Tissue and Treatment.

When two biological factors are available, an interaction term can be added. The interaction of factors analysis often helps to answer more directly to a biological question, and the user avoids a long interpretation work comparing different Differentially Expressed Genes (DEGs) lists.

For the *B. napus* dataset, we analyzed the complete glm with `Replicate=TRUE` and `Interaction=TRUE`.

```
Replicate=TRUE
Interaction=TRUE

GLM_Contrasts <- GLM_Contrasts(Results_Directory, Project_Name,
                               Target, Replicate, Interaction)

GLM_Model <- GLM_Contrasts$GLM_Model
Contrasts <- GLM_Contrasts$Contrasts
```

The model definition and the list of contrasts are available in the directory *Results/Brassica_napus/DiffAnalysis*:

- **Brassica_napus_GLM_Model.txt**
- **Brassica_napus_Contrasts_Matrix.txt**
- **Brassica_napus_GLM_Contrasts.txt** : useful to choose the contrasts of interest

Since the writing of the contrasts is automated, some of them might not be relevant. For this reason, the user has the option to select the contrasts that are relevant for a given project.
To do this, the user can open in a text editor the output file **Brassica_napus_GLM_Contrasts.txt** or can visualize the list of all contrasts in the console pane:

Inform the argument `Index_Contrast` giving the numbers of contrasts of interest. The other arguments have already been specified or have a default value.
For the *B. napus* dataset, we kept all the seven contrasts with `Index_Contrast=1:7`

```
Index_Contrast=1:7
Alpha=0.05
NbGenes_Profiles=20
NbGenes_Clustering=50

DiffAnalysis.edgeR(Data_Directory, Results_Directory, Project_Name, Target, R
                   GLM_Model, Contrasts, Index_Contrast, Annotation_FileName,
                   Filter_Strategy, Alpha, NbGenes_Profiles, NbGenes_Clusteri
                   CPM_Cutoff, Normalization_Method)
```

The DiffAnalysis_edgeR function returns 5 output files saved in the directory *Results/Brassica_napus/DiffAnalysis*:

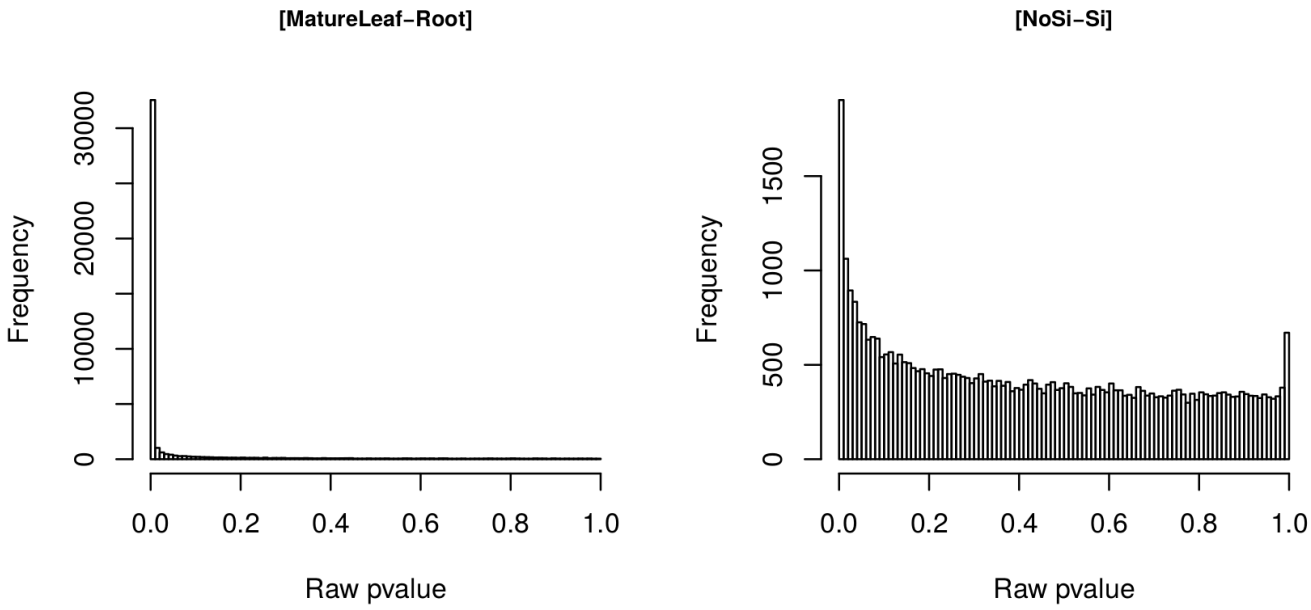
- **Brassica_napus_Compare_table.txt**
- **Brassica_napus_Contrasts_Interests_Matrix.txt**
- **Brassica_napus_DiffAnalysis_Comparisons.txt**
- **Brassica_napus_Down_Up_DEG.pdf**
- **Brassica_napus_Raw_pvalues_histograms.pdf**

The user can check the number of differentially expressed genes for each contrast by open in a text editor the output file **Brassica_napus_DiffAnalysis_Comparisons.txt** or can directly visualize these informations in the console pane:

```
## png
##   2
```

At this point in the analysis, it is important to look at the raw p-values histograms to be sure that the number of observations is high enough to estimate it. These graphs allow checking the quality of the differential analysis for each contrast of interest. The user can find all the raw p-values histograms in the file **Project_Name_Raw_pvalues_histograms.pdf**.

For example, we can observe the histograms of raw p-values for [MatureLeaf-Root] and for [NoSi-Si] contrasts included in **Brassica_napus_Raw_pvalues_histograms.pdf**:



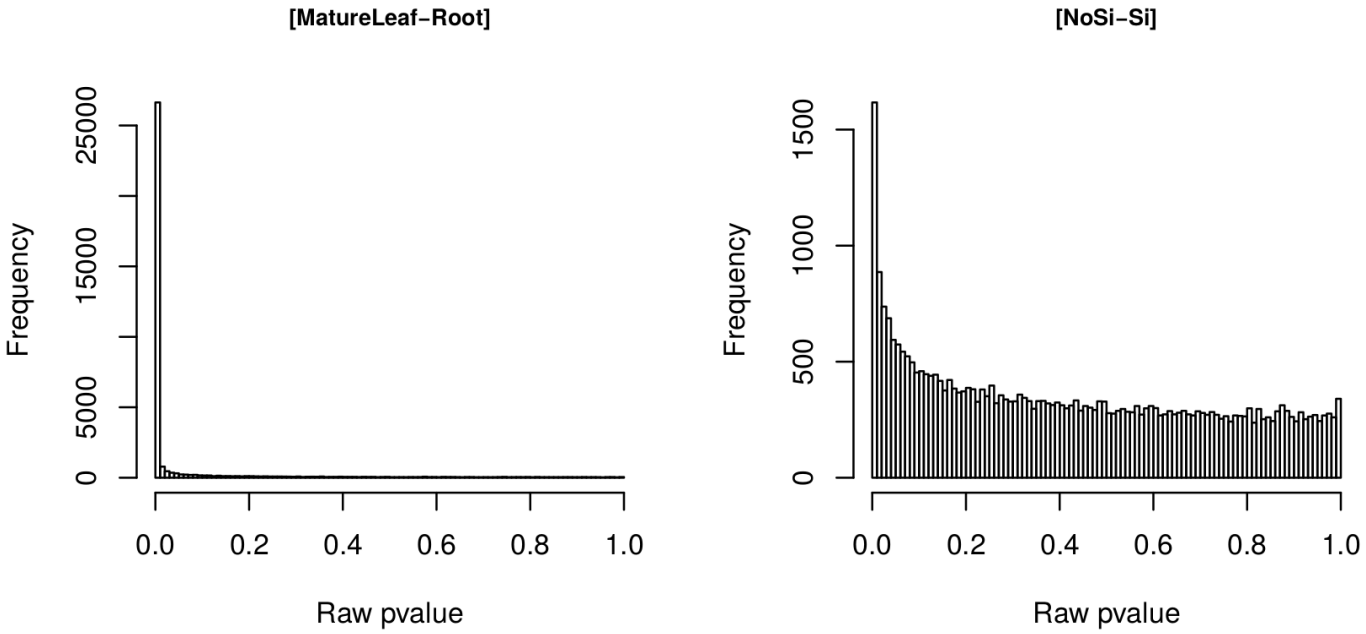
In this example, the histogram of raw p-values for the [MatureLeaf-Root] contrast is uniform and of good quality. However, for the [NoSi-Si] contrast, we observe a peak of the raw p-values equal to 1: this distribution reflects the fact that the model parameters estimation was not good and as a consequence, the results will not be reliable.

To correct this distribution, the user should choose a more stringent CPM cutoff. In our example, if we re-run the analysis using `CPM_Cutoff = 5` instead of `CPM_Cutoff = 1` as follows:

```
CPM_Cutoff=5

DiffAnalysis.edgeR(Data_Directory, Results_Directory, Project_Name, Target, R
                   GLM_Model, Contrasts, Index_Contrast, Annotation_FileName,
                   Filter_Strategy, Alpha, NbGenes_Profiles, NbGenes_Clusteri
                   CPM_Cutoff, Normalization_Method)
```

By increasing the stringency of filtering, we get good quality results for differential analysis of *B. napus* dataset.



If the raw p-values distribution remains unsatisfactory, the problem might come from the fact that the number of parameters is too large compared to the number of observations available to estimate them. In this case, we advise removing the interaction term in the GLM for the differential expression analysis.

For each contrast of interest, a subdirectory is created. For *B. napus* dataset, they are named:

- **[MatureLeaf-Root]**
- **[NoSi-Si]**
- **[NoSi_MatureLeaf-NoSi_Root]**
- **[Si_MatureLeaf-Si_Root]**
- **[MatureLeaf_NoSi-MatureLeaf_Si]**
- **[Root_NoSi-Root_Si]**
- **[MatureLeaf_NoSi-MatureLeaf_Si]-[Root_NoSi-Root_Si]**

And each one contains 8 files. For illustration, for the contrast [MatureLeaf-Root], they are named:

- **Brassica_napus_[MatureLeaf-Root]_LRT_BH.txt**
- **Brassica_napus_[MatureLeaf-Root]_DEG_BH.txt**
- **Brassica_napus_[MatureLeaf-Root]_id_DEG.txt**
- **Brassica_napus_[MatureLeaf-Root]_DEG_NormCounts.txt**

- **Brassica_napus_[MatureLeaf-Root]_DEG_log2_NormCounts.txt**
- **Brassica_napus_[MatureLeaf-Root]_plotSmear.pdf**
- **Brassica_napus_[MatureLeaf-Root]_Top20_Profile.pdf**
- **Brassica_napus_[MatureLeaf-Root]_Top50_Clustering.pdf**

Venn diagram and merge of DEG lists

To evaluate how many genes are impacted in their transcription by a given treatment, the user could be interested in looking at the genes differentially expressed in multiple contrasts. We cannot automate all possible combinations, but the Venn_Intersection_Union function offers some options to combine multiple lists obtained in the previous GLM analysis.

In this section, the user will choose a `Title` so that DiCoExpress will be able to create a subdirectory in *Results/Project_Name/Venn_Intersection_Union/Title*.
The user will then indicate the considered contrasts in the `Groups` argument.
Finally, the user will specify the `Operation` he wants to perform.
When `Operation="Union"`, the list of genes differentially expressed at least once is generated.
When `Operation="Intersection"`, the list of genes differentially expressed shared among all the contrasts chosen is generated.

For the *B. napus* dataset, we analyzed the treatment effect corresponding to the difference between the samples treated with Si and the samples without Si treatment (NoSi). Therefore, we chose to create the union list from 3 contrasts: [MatureLeaf_NoSi-MatureLeaf_Si], [Root_NoSi-Root_Si], [MatureLeaf_NoSi-MatureLeaf_Si]-[Root_NoSi-Root_Si] and we set `Title=NoSi-Si`.

```
Title="NoSi-Si"
Groups=c("[MatureLeaf_NoSi-MatureLeaf_Si]","[Root_NoSi-Root_Si]","[MatureLeaf
Operation="Union"

Venn_IntersectUnion(Data_Directory, Results_Directory, Project_Name,
                    Title, Groups, Operation, Annotation_FileName)
```

In this example, the Venn_IntersectUnion function returns 3 output files saved in the directory *Results/Brassica_napus/Venn_Intersection_Union/NoSi-Si*:

- **Brassica_napus_NoSi-Si_Union_List.txt**

- **Brassica_napus_NoSi-Si_Union_Summary_Table.txt**
- **Brassica_napus_NoSi-Si_Venn_Diagram.pdf**

In **Brassica_napus_NoSi-Si_Union_Summary_Table.txt** file, the user will find a column summarizing to which contrast a given gene belongs. As the contrast names are quite long, DiCoExpress uses a letter code to identify them. The user will find the legend of the groups in the **pdf file**.

Coexpression analysis

On any Venn_Interction_Union results, the user can perform a co-expression analysis. It is not necessary to specify the `Title` because it is already indicated in the Venn_Intersection_Union function.

The `A`, `B`, and `K` arguments of this function have values by default, which should be adapted for most analysis. Sometimes, based on the ICL curve, the dataset will call for an increment of the `B` argument for optimal results.

```
A=5
B=40
K=c(2,seq(5,30,by=5))

Coexpression_coseq(Data_Directory, Results_Directory, Project_Name,
           Title, Target, Raw_Counts, Annotation_FileName,
           Color_Group, A, B, K)
```

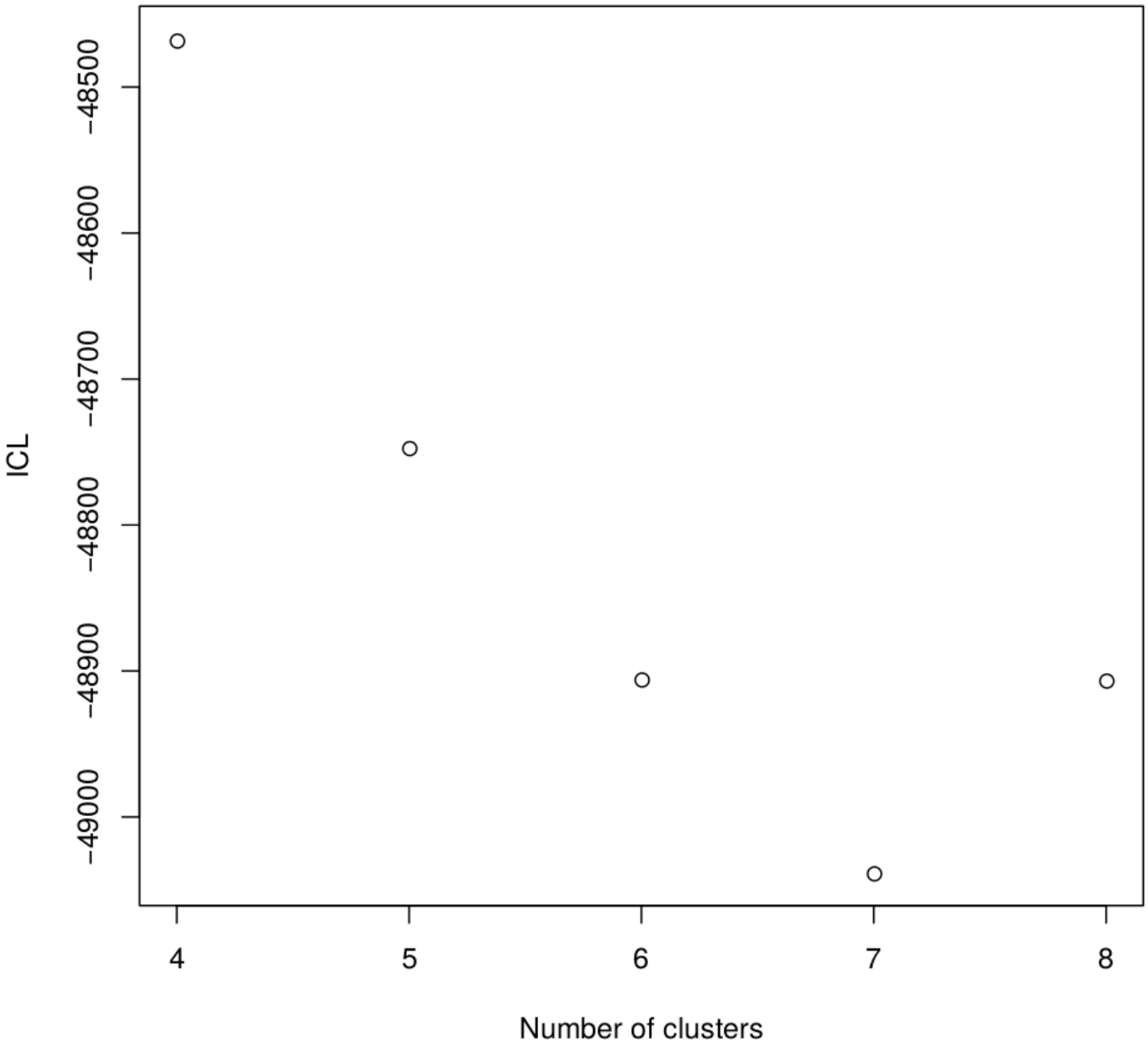
The Coexpression_coseq function returns 17 output files placed in the folder *Results/Brassica_napus/Coexpression/NoSi-Si*:

Some are plots concerning the analysis process:

- **Brassica_napus_NoSi-Si_Loop_1.pdf**
- **Brassica_napus_NoSi-Si_Loop_2.pdf**
- **Brassica_napus_NoSi-Si_coseq_final.RData**
- **Brassica_napus_NoSi-Si_coseq_loop_2.RData**
- **Brassica_napus_NoSi-Si_Results_First_Loop.txt**
- **Brassica_napus_NoSi-Si_Results_Second_Loop.txt**

For the optimal quality of the clusters, the user should check the ICL curve: this curve should be convex. *i.e.*, the minimum value should be clearly identifiable.

For *B. napus* dataset, we observe a clear minimum at 7 in the **Brassica_napus_NoSi-Si_Loop_2.pdf** file:



Some files contain the results:

- **Brassica_napus_NoSi-Si_AllClusters.txt**
- **Brassica_napus_NoSi-Si_Final_Coseq.pdf**
- **Brassica_napus_NoSi-Si_Results_Final.txt**
- **Brassica_napus_NoSi-Si_Boxplot_profiles_Coseq.pdf**
- **Brassica_napus_NoSi-Si_Cluster1_GeneID.txt**
- **Brassica_napus_NoSi-Si_Cluster2_GeneID.txt**
- **Brassica_napus_NoSi-Si_Cluster3_GeneID.txt**
- **Brassica_napus_NoSi-Si_Cluster4_GeneID.txt**
- **Brassica_napus_NoSi-Si_Cluster5_GeneID.txt**

- **Brassica_napus_NoSi-Si_Cluster6_GeneID.txt**
- **Brassica_napus_NoSi-Si_Cluster7_GeneID.txt**

Enrichment

To perform enrichment analysis, the user must have a Reference file. This file contains in the first column the Gene identifier and in the second column the annotation term corresponding to this gene. The second column must contain only one annotation term per line. If a gene has multiple annotations, its identifier must be repeated in the first column as many times as it has an annotation term. If any gene has no annotation, it must appear in the Reference file within the annotation column "NA".
For example, in *B. napus* dataset, we used the **Bna_Reference_Annotation_File.txt** file. Here below, some lines as an example:

```
##              Gene_ID                                        Annotations
## 1      BnaA01g00010D   [IPR]_Microsomal signal peptidase 12kDa subunit
## 244801 BnaA01g00010D      [MapMan]_[29.3.99]_protein.targeting.unknown
## 217789 BnaA01g00020D                                              <NA>
## 244802 BnaA01g00020D              [MapMan]_[35.2]_not assigned.unknown
## 2      BnaA01g00030D [IPR]_Protein of unknown function DUF3550/UPF0682
## 3      BnaA01g00030D           [IPR]_Regulator of G protein signalling
```

The Enrichment function allows performing automated enrichment analysis on all co-expression clusters or on a list of DEGs resulting from a given contrast list.

On the coexpression clusters

The user can perform hypergeometric tests to detect if the coexpressed groups are enriched or depleted in some annotation terms. The results of the enrichment analysis will help the user in the interpretation of the results.

the user must specify the reference file `Reference_FileName`, `Contrast_Name=NULL` to automatically analyze the coexpression clusters from the previous analysis, and choose the `Alpha` threshold of hypergeometric tests.

```
Contrast_Name=NULL
Reference_FileName="Bna_Reference_Annotation_File.txt"
Alpha=0.01

Enrichment(Data_Directory, Results_Directory, Project_Name, Title,
           Contrast_Name, Reference_FileName, Alpha)
```

The results are saved in *DiCoExpress/Results/Brassica_napus/Enrichment/NoSi-Si*:

- **Brassica_napus_NoSi-Si_Cluster_1_All_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_1_Over_Under_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_2_All_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_2_Over_Under_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_3_All_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_3_Over_Under_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_4_All_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_4_Over_Under_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_5_All_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_5_Over_Under_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_6_All_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_6_Over_Under_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_7_All_Results.txt**
- **Brassica_napus_NoSi-Si_Cluster_7_Over_Under_Results.txt**
- **Brassica_napus_NoSi-Si_Summary_Clusters_Enrichment.txt**

The choice of the p-value is not obvious; some software adjust the raw p-values; some others prefer to threshold the raw p-value to declare a term enriched.
In DiCoExpress, the term is declared enriched if its raw p-value is lower than `Alpha`. If the user wants to adjust the p-values, he can do so it on the files **All_Results** that contains the raw p-values for each cluster.
The user can use the file **Brassica_napus_NoSi-Si_Summary_Clusters_Enrichment.txt** to compare the annotation terms in all clusters.

On a list of DEGs

Enrichment analysis can also be performed on a list of DEGs resulting from a given contrast list.

In this case, the user before proceeding will specify a new title for this analysis so DiCoExpress can create a new subdirectory and the results files in it.

In the example below, the list of DEGs of the interaction contrast is tested:
`Contrast_Name="[MatureLeaf_NoSi-MatureLeaf_Si]-[Root_NoSi-Root_Si]"`.

In this example we chose `Title=Interaction`.

```
Title="Interaction"
Contrast_Name="[MatureLeaf_NoSi-MatureLeaf_Si]-[Root_NoSi-Root_Si]"
Reference_FileName="Bna_Reference_Annotation_File.txt"
Alpha=0.01

Enrichment(Data_Directory, Results_Directory, Project_Name, Title,
           Contrast_Name, Reference_FileName, Alpha)
```

The results are saved in *DiCoExpress/Results/Brassica_napus/Enrichment/interaction*:

- **Brassica_napus_Interaction_All_Results.txt**
- **Brassica_napus_Interaction_Over_Under_Results.txt**

Session Info

The user can find the **SessionInfo.txt** file in the *DiCoExpress/Results/Brassica_napus* directory. In this file, he can have informations about the versions of R, the OS and all the necessary packages.