tRanslatome: an R/Bioconductor package to portray translational control

Toma Tebaldi <t.tebaldi@unitn.it>,
Erik Dassi <erik.dassi@unitn.it>,
Galena Kostoska <kostoska.galena@unitn.it>,
Gabriella Viero <viero@fbk.eu>,
Alessandro Quattrone <alessandro.quattrone@unitn.it>,

Contents

1 Introduction

One way to achieve a comprehensive estimation of the influence of different layers of control on gene expression is to analyze the changes in abundances of molecular intermediates at different levels. For example, comparing changes between abundances of mRNAs in active translation with respect to the corresponding changes in abundances of total mRNAs (by mean of parallel high-throughput profiling) we can estimate the influence of translational controls on each transcript. The tRanslatome package represents a complete platform for comparing data coming from two parallel high-throughput assays, profiling two different levels of gene expression. The package focuses on the comparison between the translatome and the transcriptome or the proteome. It can be used to compare any variation monitored at two “-omics” levels (e.g. translatome versus proteome, translatome versus transcriptome or proteome versus transcriptome). The package provides a broad variety of statistical methods covering each step of the standard data analysis workflow: detection and comparison of differentially expressed genes (DEGs), detection and comparison of enriched biological themes through Gene Ontology (GO) annotation. The package provides tools to visually compare/contrast the results. An additional feature lies in the possibility to detect enrichment of targets of translational regulators using the experimental annotation contained in the AURA database (http://aura.science.unitn.it/).

2 First Worked example

The following code illustrates a standard analysis pipeline with tRanslatome. For demonstrating tRanslatome in practice we use a dataset coming from (Parent *et al.*, 2007). The dataset is named ”tRanslatomeSampleData”. In this study, the authors profiled the total and the polysome-bound transcripts in differentiated and undifferentiated human HepaRG cells. Therefore this example presents two levels of gene expression analysis (transcriptome, labelled as ”tot” and translatome, labeled as ”pol”) on cells in two different conditions (undifferentiated, labeled as ”undiff” vs. differentiated, labeled as ”diff”). Experiments were done by the authors in biological triplicate, labeled as ”a”, ”b” and ”c”. All the steps contained in the code will be explained in more detail in the following sections.

```
> ##loading the tRanslatome package
> library(tRanslatome)
> ##loading the training data set
> data(tRanslatomeSampleData)
> translatome.analysis <- newTranslatomeDataset(expressionMatrix,
+ c("tot.undiff.a", "tot.undiff.b", "tot.undiff.c"),
+ c("tot.diff.a", "tot.diff.b", "tot.diff.c"),
+ c("pol.undiff.a", "pol.undiff.b", "pol.undiff.c"),
+ c("pol.diff.a", "pol.diff.b", "pol.diff.c"),
+ label.level= c("transcriptome", "translatome"),
+ label.condition=c("undifferentiated", "differentiated"))
> ##identification of DEGs with the use of the limma statistical method
> limma.DEGs <- computeDEGs(translatome.analysis,
+
> ##enrichment analysis of the selected DEGs
> CCEnrichment <- GOEnrichment(limma.DEGs,ontology="CC", classOfDEGs="up",
+
> ##performing a comparison of the biological themes enriched
> ##in the two levels of gene expression
> CCComparison <- GOComparison(CCEnrichment)
```

2.1 DEGs detection

The initial core of the package consists of the class holding input data, called TranslatomeDataset. Objects of this class can be created through the newTranslatomeDataset() function. This function takes as input a normalized data matrix coming from the high throughput experiment with entities (genes, transcripts, exons,proteins) in rows and samples (normalized signals coming from microarray, next generation sequencing, mass spectrometry) in columns. Since tRanslatome doesn't provide any normalization, signals contained in the data matrix should be normalized before, unless the DEGs selection method doesn't provide also a normalization step, as in the case of edgeR and DEseq. In our worked example microarray data were previously quantile normalized.

In addition to the data matrix, important input parameters are:

- ✓ **cond.a, cond.b, cond.c, cond.d**: vectors of column names belonging to expression matrix. These columns contain the signal intensity data coming from the samples of the: (a) first expression level of the control condition (in our example: total RNA, undifferentiated cells); (b) first expression level of the treatment condition (in our example: total RNA, differentiated cells); (c) second expression level of the control condition (in our example: polysomal RNA, undifferentiated cells); (d) second expression level of the treatment condition (in our example: polysomal RNA, differentiated cells).
- ✓ **label.level, label.condition**: character vectors specifying: (a) the names given to the two levels. By default levels are named ”1st level” and ”2nd level”, but the user can specify others: in our example the two levels are named ”transcriptome” and ”translatome”; (b) the names given to the two conditions. By default, these values are ”control” and ”treated”, but user can specify others: in our example the two levels are named ”undifferentiated” and ”differentiated”;
- ✓ **data.type**, character vector specifying the type of data contained in the expression matrix. By default it is set to array, the other accepted value is ngs.

Once the object is initialized, the next step is the identification of DEGs. The function performing this task is called computeDEGs(). This function takes as input a label specifying the method employed in order to detect DEGs and returns a table containing, for each row of the expression matrix (genes, transcripts, exons, proteins) the associated fold change, statistical significance and classification according to their expression in the two “-omics” levels.

In addition to the object of class TranslatomeDataset, important input parameters are:

- ✓ **method**: a label that specifies the statistical method for DEGs detection. This parameter can take the following values:
  1. **DESeq** (Anders and Huber, 2010). This method has been developed for the analysis of count data from high-throughput sequencing assays. The test for differential expression is based on a model using the negative binomial distribution
  2. **edgeR** (Robinson *et al.*, 2010). This method is addressed to expression analysis of RNA-seq and digital gene expression profiles with biological replication. The method uses empirical Bayes estimation and exact tests based on the negative binomial distribution. It is also useful for differential signal analysis with other types of genome-scale count data.
  3. **SAM** (Tusher *et al.*, 2001). Significance Analysis of Microarrays. This method detects differential expression for microarray data, sequencing data, and other data with a large number of features (e.g. genes), correlating them with an outcome variable, such as a group indicator, quantitative variable or survival time.
  4. **t-test** (Tian *et al.*, 2005). One of the more common statistical approaches to select genes differentially expressed between two groups.
  5. **RP** (Breitling *et al.*, 2004). RankProd, a non-parametric method for identifying differentially expressed (up- or down- regulated) genes based on the estimated percentage of false predictions (pfp). The method can combine data sets from different origins (meta-analysis) to increase the power of the identification.
  6. **limma** (Smyth, 2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. The methods are also applicable to expression data from non-microarray platforms, such as quantitative PCR or RNA-Seq, given a suitable matrix of expression values.
  7. **ANOTA** (Larsson *et al.*, 2011). Method developed to identify differential translation between two or more sample classes, i.e. differences in actively translated mRNA levels that are independent of underlying differences in cytosolic mRNA levels. The method uses partial variances and the random variance model.
  8. **TE**. Translation Efficiency. TE is traditionally defined as the ratio of polysomal RNA and subpolysomal RNA signals (Powley *et al.*, 2009), or the ratio of polysomal RNA and total RNA signals. In ribosome footprinting experiments, TE has been defined as the ratio of ribosome protected fragments (RPFs) and RNA-seq reads (Ingolia *et al.*, 2011). When protein levels are detected, TE is defined as the ratio between protein and transcript levels. The TE method first calculates TE as the ratio of the signals of the two "-omics" levels provided by the user, then select DEGs according to differences in the TE between the treatment condition and the control condition.
  9. **none**. No statistical method is applied to the analysis, DEGs are selected only according to the threshold on the fold change.

- ✓ **significance.threshold, FC.threshold**: thresholds on (a) the statistical significance to select differentially expressed genes (the default is set to 0.05); (b) the absolute log2 fold change to select differentially expressed genes (the default is set to 0).
- ✓ **mult.cor**: a boolean variable specifying whether the significance threshold is applied on the Benjamini-Hochberg multiple test corrected p-values obtained from the DEGs detection method. By default it is set to TRUE.

The function computeDEGs() generates an object of class DEGs, containing the result of the differential expression analysis comparing the two “-omics” levels. DEGs can then be later retrieved with the accessory getDEGs function.

One way to visualize the results obtained by the computeDEGs() function is to use the **Scatterplot()** method on the object of class DEGs, generating a plot in logarithmic scale where each biological feature (gene, transcript, exon, protein) is represented as a dot whose position is uniquely determined by its fold change (FC) in the first expression level (on the x-axis), and in the second expression level (on the y-axis). The scatterplot generated from the analysis of the worked example is displayed in Figure 1. The two expression levels are names ”transcriptome” and ”translatome”.

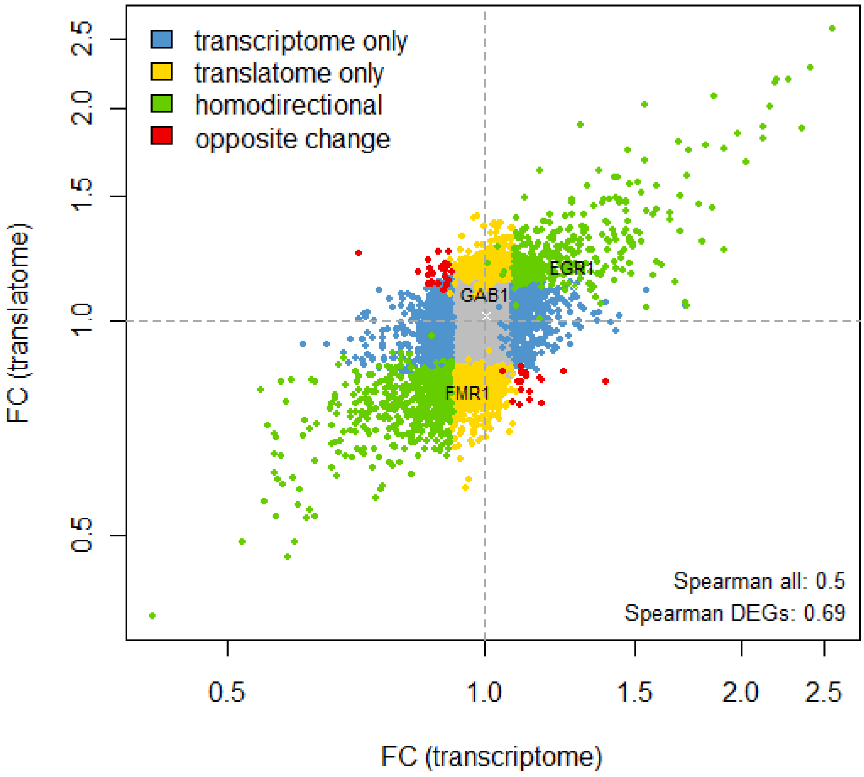

Figure 1: tRanslatome provides the method Scatterplot(), drawing a scatterplot in which each gene is mapped according to its fold change at the first level (on the x- axis) and the fold change at the second level (on the y-axis). The track parameter was set to c("GAB1","FMR1","EGR1"). We adopt a color code to label different classes of DEGs: blue for genes differentially expressed only at the first level; yellow for genes differentially expressed only at the second level, green for genes changing homodirectionally, red for genes with opposite changes.

This plot, as all the graphical outputs of tRanslatome, can be saved in various formats: jpeg (jpeg), postscript (postscript), pdf (pdf), or simply displayed on the screen (on screen, the default option). If the expression matrix is annotated with row names, specific rows can be tracked in the plot by specifying a character vector of gene names (the "track" parameter). The corresponding dot will be explicitly highlighted in the scatterplot. By default, the track parameter is empty.
A color code is adopted in the scatterplot to label different classes of DEGs: blue for genes differentially expressed only at the first level, yellow for genes differentially expressed only at the second level, green for genes showing concordant changes in the two "-omics" levels, red for genes changing with opposite directions in the two levels. The Spearman's Correlation Coefficient between the fold changes of all the genes and between the fold changes of all the DEGs is also displayed.

Another way to visualize the results obtained from the computeDEGs() function is to use the **Histogram()** method, showing a histogram with the number of biological features differentially expressed (up-regulated or down-regulated) in each level.
The user can set the parameter plottype to specify whether the histogram should be (a) a brief "summary" of the results, showing the number of genes up and down regulated in the first and the second "-omics" level (displayed in Figure 2 for the working example); (b) a "detailed" histogram presenting the distribution of all the classes of DEGs resulting from the comparison of the two "-omics" levels (displayed in Figure 3 for the working example).

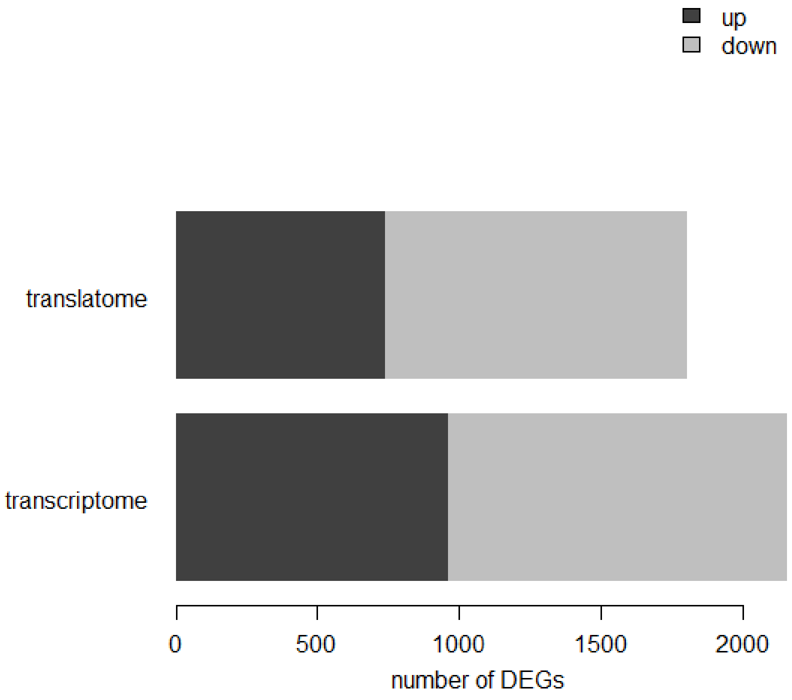

Figure 2: Summary histogram of the identified differentially expressed genes (DEGs). The histogram shows the distribution of up-regulated (in black) and down-regulated (in gray) genes in the two expression levels, transcriptome and translatome in our example.

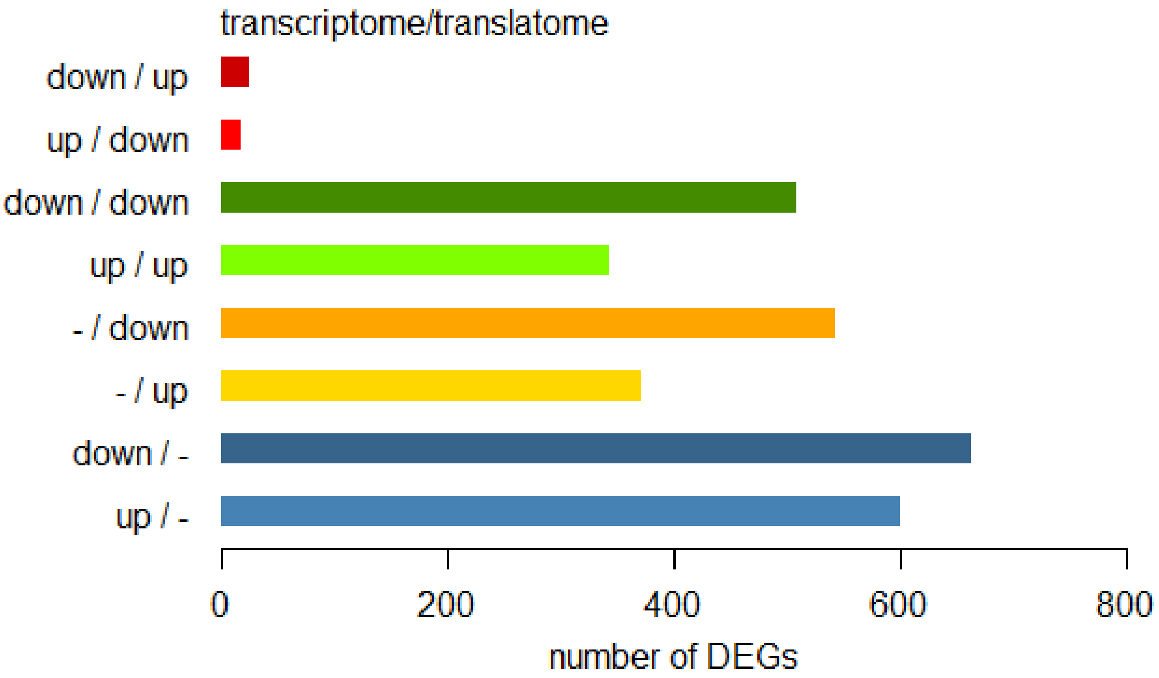

Figure 3: Detailed histogram of the identified differentially expressed genes (DEGs). The histogram displays the number of genes up or down regulated only in the first expression level ("up/-" or "down/-", in blue tones), only in the second expression level ("-/up" or "-/down", in yellow tones), in both expression levels with the same trend ("up/up" or "down/down", in green tones), in both expression levels with opposite directions ("up/down" or "down/up", in red tones).

2.2 Quality control

tRanslatome enables the generation of additional graphics helping the user to perform some basic quality controls on the data and the results of the identification of DEGs.

The MA-plots show in logarithmic scale the relationship between the average log2 signal intensity (A) and the log2 fold change (M) for each gene. The general assumption of genome-wide profiles is that most of the genes don't change between the two conditions under comparison. If this is not the case and the result is not justified by any biological consideration, an alternative normalization method should be applied.

tRanslatome enables the generation of MA-plots within the **MAplot()** method, which can be applied to any object of class DEGs. The MA-plot of the worked example is displayed in Figure 4. The upper panel in Figure 4 refers to the first expression level (transcriptome), whereas the lower panel refers to the second expression level (translatome). DEGs in each level are colored in blue and yellow, respectively.

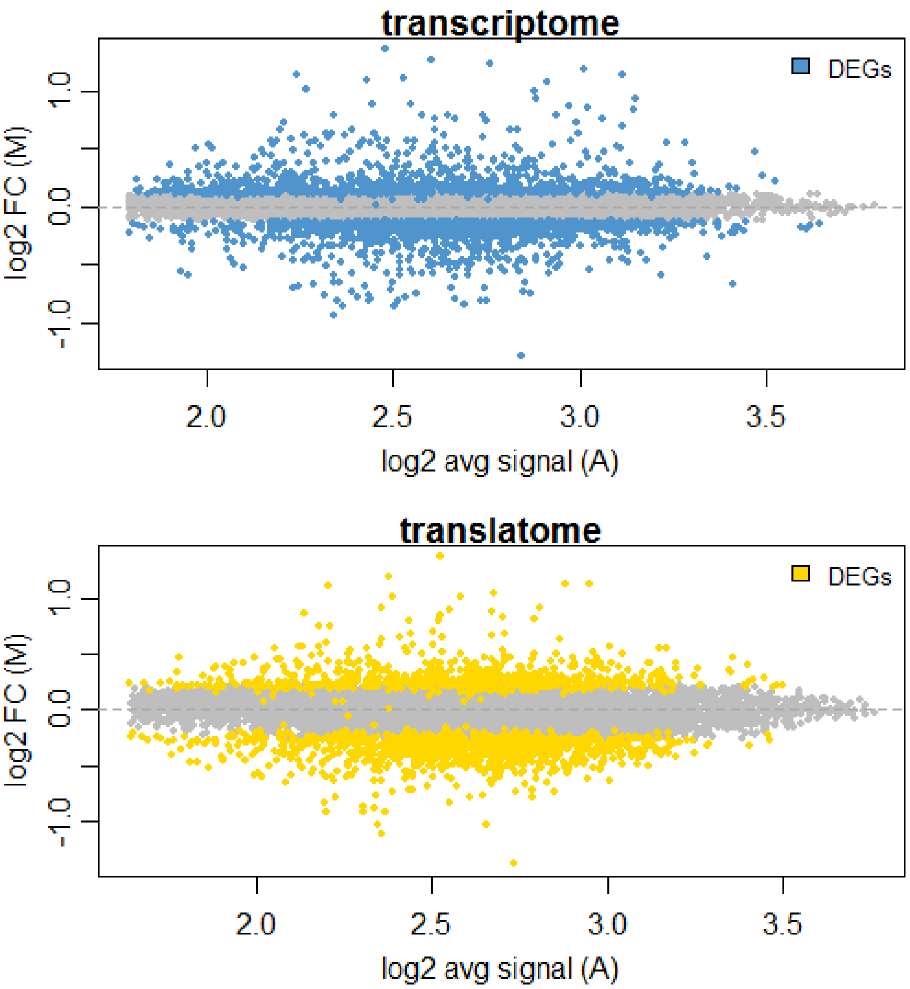

Figure 4: MA plots are displayed for the first expression level (transcriptome, in the upper panel, with DEGs labeled in blue) and the second expression level (translatome, in the lower panel, with DEGs labeled in yellow).

In tRanslatome there is also a method to visualize the relationship between the FC of the selected DEGs with respect to the standard deviation of their signals across the replicates. The **SDplot()** method plots, for each expression level, the standard deviation of the genes against their logarithmic fold change. The set of input parameters is exactly the same as in MAplot().

The SD plot of the worked example is displayed in Figure 5. The upper panel in Figure 5 refers to the first expression level (the transcriptome in the worked example), whereas the lower panel refers to the second expression level (the translatome in the worked example).

Alternatively, the relationship between the logarithmic fold change and the coefficient of variation (CV) of each gene can across the replicates can be visualized with the method **CVplot()**. The CV plot of the worked example is displayed in Figure 6. The upper panel in Figure 6 refers to the first expression level (the transcriptome in the worked example), whereas the lower panel refers to the second expression level (the translatome in the worked example).

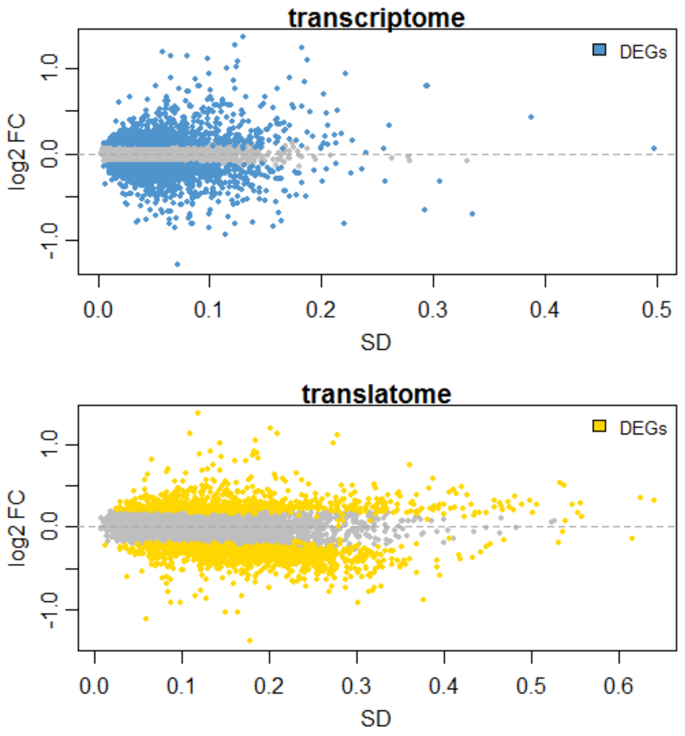

Figure 5: SD plots are displayed for the first expression level (transcriptome, in the upper panel, with DEGs labeled in blue) and the second expression level (translatome, in the lower panel, with DEGs labeled in yellow).

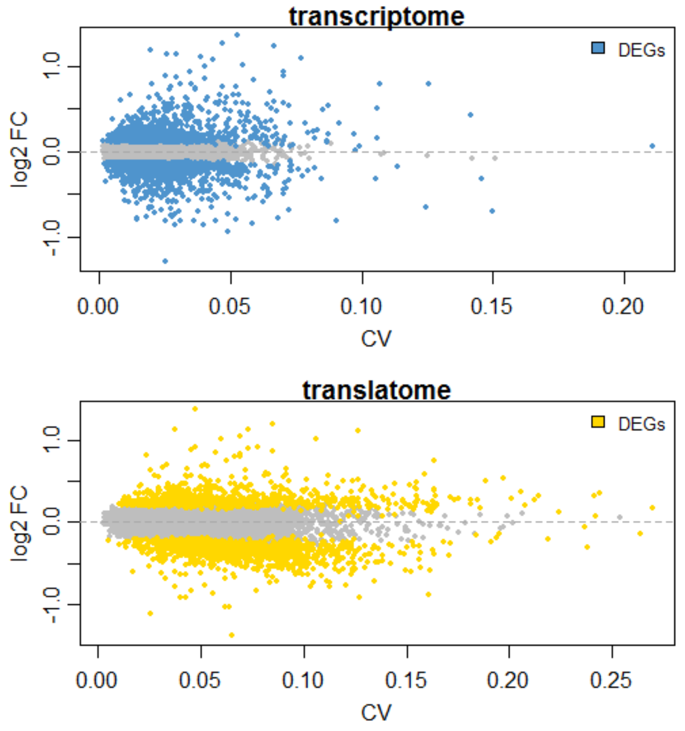

Figure 6: CV plots are displayed for the first expression level (transcriptome, in the upper panel, with DEGs labeled in blue) and the second expression level (translatome, in the lower panel, with DEGs labeled in yellow).

2.3 GO Enrichment

The Gene Ontology (GO) project provides a standardized controlled vocabulary to describe gene product attributes in all organisms (Ashburner *et al.*, 2000). GO consists of three hierarchically structured vocabularies (ontologies) describing gene products in terms of their associated biological processes (BP), cellular components (CC) and molecular functions (MF). One of the most frequent applications of the GO to the results of high-throughput experiments is the enrichment analysis, i.e. the identification of GO terms that are significantly over-represented in a given set of differentially expressed genes. The analysis of ontological enrichments helps to understand the possible functional characteristics associated to a given list of DEGs, suggesting the occurrence of common mechanisms of regulation coordinating the biological system under examination. tRanslatome offers different alternatives to perform the GO enrichment analysis of DEGs in any of the two "-omics" levels, by identifying the overrepresented GO terms in the lists of DEGs returned from the computeDEGs() function.

GOEnrichment() is a function in tRanslatome which identifies GO terms significantly enriched in the two lists of DEGs corresponding to the two "-omics" levels under analysis. The enrichment analysis can be performed on the whole set of GO ontologies (by default), or restricted to one single ontology (either molecular function, cellular component or biological process). Moreover, the enrichment analysis can be performed on different classes of DEGs: only up-regulated genes, only down-regulated genes or the union of up-regulated and down-regulated genes (i.e., the whole set of DEGs).

The function GOEnrichment() takes as input an object of class DEGs. In addition, important input parameters are:

- ✓ **test.method**. A character string specifying the statistical method to calculate the enrichment. By default it is set to classic (meaning that the enrichment is measured with the classic Fisher exact test). This parameter can be set to elim, weight, weight01 or parentchild. All these methods are implemented in the topGO Bioconductor package (http://www.bioconductor.org/packages/release/bioc/html/topGO.html). These alternative testing methods account for the topology of the GO graph, providing solutions to eliminate local similarities and dependencies between GO terms (Alexa *et al.*, 2006).
- ✓ **test.threshold**. The significance threshold to select over-represented terms. The default is se to 0.05.
- ✓ **mult.cor**. A boolean variable specifying whether the significance threshold is applied on the Benjamini-Hochberg multiple test corrected p-values obtained from the enrichment method. By default it is set to TRUE.

The output of the function GOEnrichment() is an object of class GOsets, containing the results of the enrichment analysis in tabular form.

The method **Radar()** can be applied to any object of class GOsets in order to display the most enriched GO terms in the first and in the second expression level in a radar plot display. The radar plot is useful to visualize differential enrichment of certain GO terms in the two lists of DEGs associated to the two "-omics" levels. The radar plot resulting from the GO Cellular Component analysis of the worked example is displayed in Figure 7.

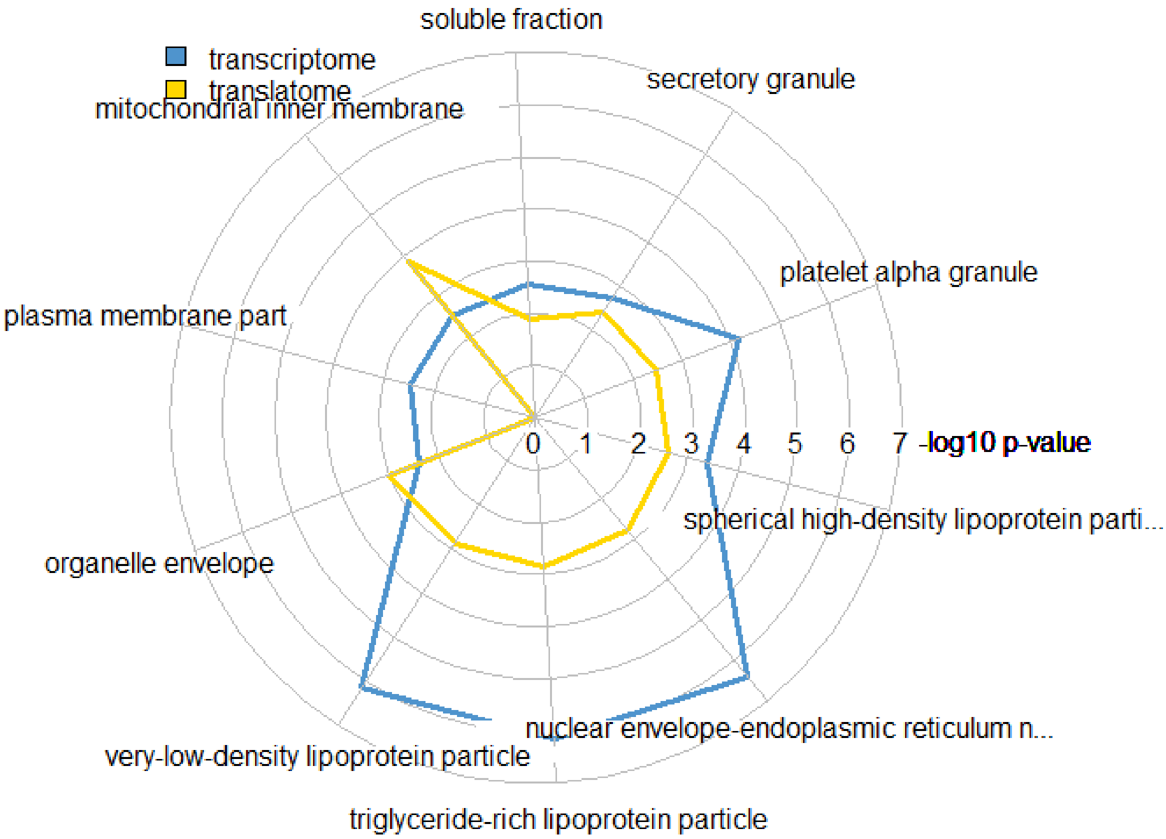

Figure 7: The nine top enriched cellular component GO terms are displayed in form of a radar plot. Enrichment p-value for the first expression level (the transcriptome in the worked example) are labeled in blue, whereas the enrichment p-values for the second expression level (the translatome in the worked example) are labeled in yellow.

The Radar() method takes as input an object of class GOsets, a label specifying the ontology of interest (either CC, BP or MF), and the number of top enriched GO terms that will be represented on the plot for the two "-omics" levels. By default the value is set to 5 for both the levels.

A second way to display and compare GO enrichment results is provided by the method **Heatmap()**, taking the same input parameters as the Radar() method. The heatmap resulting from the GO Cellular Component analysis of the worked example is displayed in Figure 8.

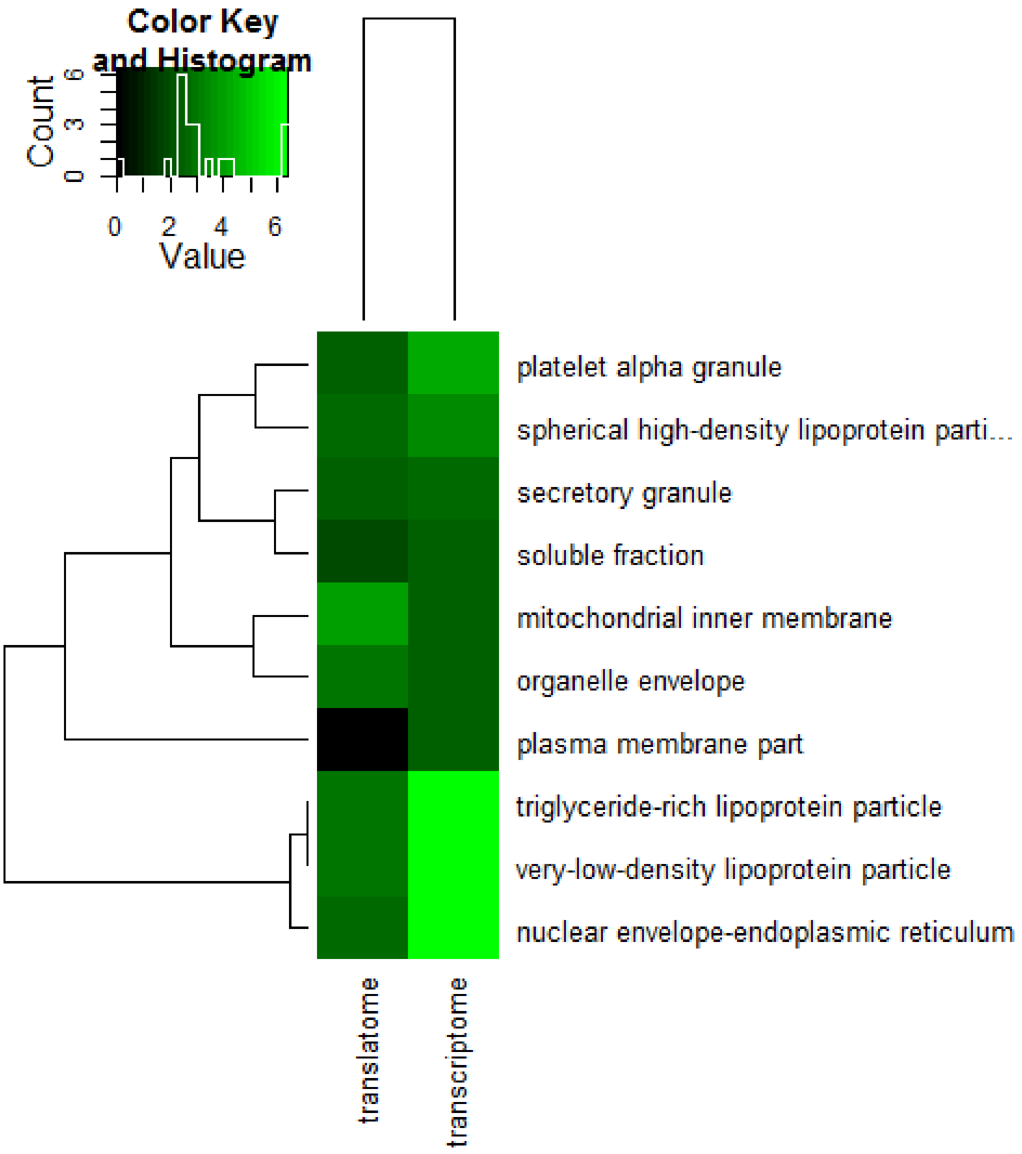

Figure 8: Heatmap of the top enriched GO Cellular Component terms for each expression level, transcriptome and translatome in the worked example. The color scale is based on the -log10 transformed enrichment p-value. Black cells are associated to not significant enrichments. GO terms are clustered according to the enrichment p-values across the two "-omics" levels.

2.4 GO Comparison

In order to further compare the functional enrichments associated to the two "-omics" levels, the function **GOComparison()** takes as input an object of class GOsets and returns as output an object of class GOsims, containing a variety of tables with comparisons between the lists of enriched GO terms. The comparison include the calculation of semantic similarity scores between the two lists of enriched terms, using the Wang method (Wang *et al.*, 2007) implemented in the GOSemSim Bioconductor package (http://www.bioconductor.org/packages/2.12/bioc/html/GOSemSim.html).
The function has only one input parameter, an object of class GOsets. The output contains the comparison of the enriched terms (named "identity comparison") and the semantic similarity measure between the GO terms enriched in the two expression levels under analysis (names "semantic similarity comparison").

The results of GOComparison can be displayed with the method **IdentityPlot()**, displaying in a barplot the number of GO terms showing enrichment in both expression levels or only in one expression level. The result of the method, applied to the worked example, is shown in Figure 9.

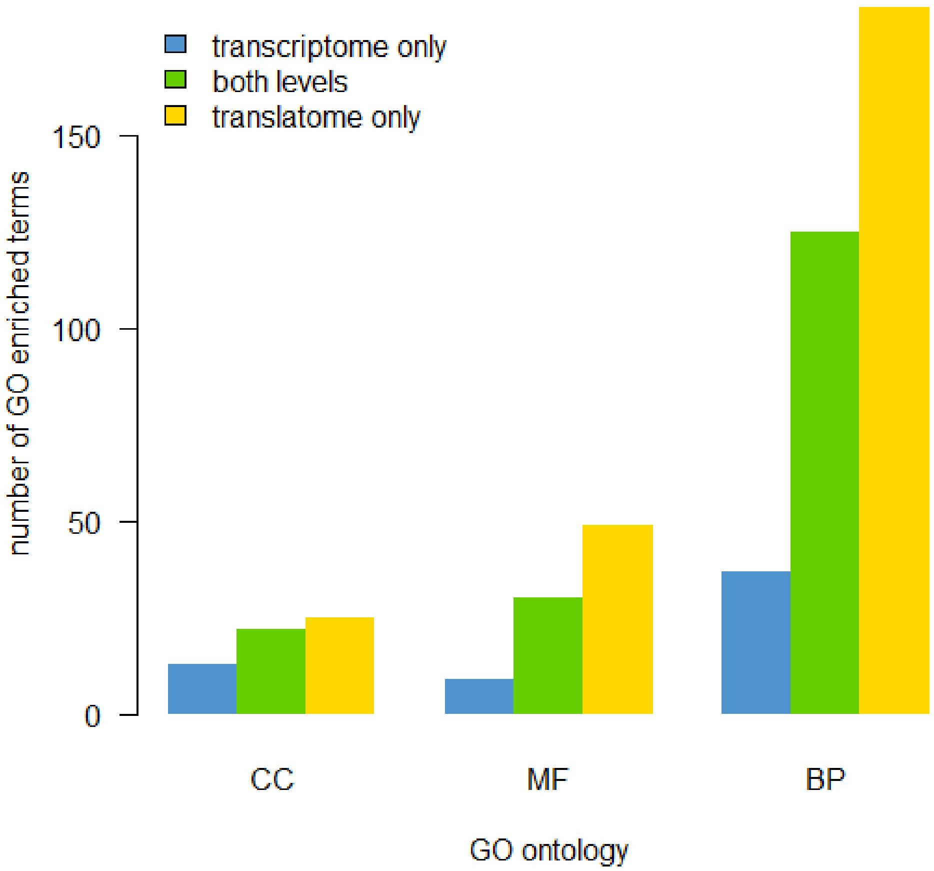

Figure 9: Barplot of the number of GO terms (CC, MF and BP) showing significant enrichment in both expression levels (transcriptome and translatome in the worked example, in green), only in the first level (in blue), only at the second level (in yellow).

The method SimilarityPlot(), applied to an object of class GOsims, displays a barplot with the semantic similarity values between the two lists of enriched GO terms, associated to the two "-omics" levels. Semantic similarity values range from 0 to 1. The result of the method, applied to the worked example, is shown in Figure 10.

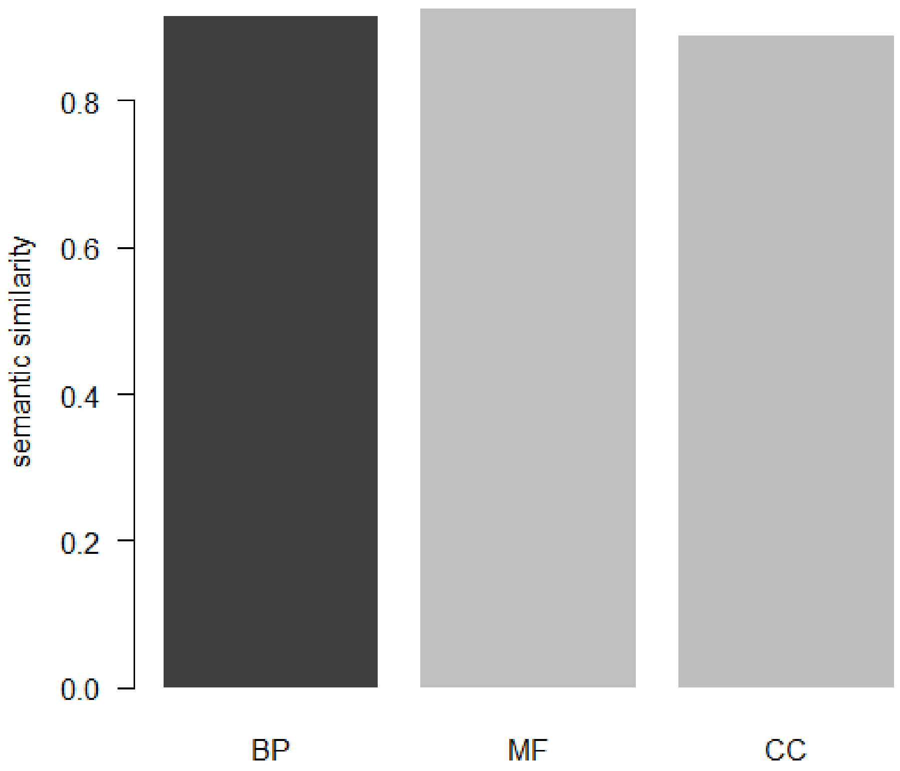

Figure 10: Barplot of the average semantic similarity values between GO terms (CC, MF and BP) significantly enriched in the first or in the second expression level (transcriptome and translatome in the worked example).

2.5 Regulatory Enrichment

RegulatoryEnrichment() is a function which, given as input an object of class DEGs, identifies overrepresented post-transcriptional regulators (such as RNA-binding proteins, microRNA) possibly coordinating the differential expression of their target DEGs. The default analysis is based on a dataset of experimentally determined post-transcriptional interactions, extracted from the AURA database (http://aura.science.unitn.it). The user can also specify a custom dataset of annotations onto which the same analysis can be performed. Moreover, the function can identify enriched regulators for separate classes of DEGs: only up-regulated genes, only down-regulated genes or their union. The method works by using two lists of genes: the first containing all the genes targeted by each of the post-transcriptional regulators, and the second containing the number of regulated and non-regulated genes for each of these post-transcriptional regulators in the background gene set (the whole genome). The function computes a Fisher enrichment p-value indicating whether a significant group of genes in the DEGs list is likely to be regulated by these post-transcriptional regulators. The output of the function is an object of class EnrichedSets, containing the results of the enrichment analysis. Both the methods Radar() and Heatmap(), previously introduced in section 2.3, can be applied also to objects of class EnrichedSets in order to show the top enriched regulatory elements associated to the two "-omics" levels.

2.6 Translational Efficiency

As already specified in the previous sections, tRanslatome allows the user to select the TE method when using the computeDEGs() function. The concept of translation efficiency has been associated to multiple interpretations in different publications, according to the experimental procedure. When performing polysomal profiling, TE is traditionally defined as the ratio of polysomal RNA and subpolysomal RNA signals (e.g. Powley et al., 2009), or the ratio of polysomal RNA and total RNA signals. With the development of ribosome footprinting, TE has been defined as the ratio of ribosome protected fragments (RPFs) and RNA-seq reads (Ingolia et al., 2011). When protein levels are detected, TE is defined as the ratio between protein and transcript levels, or, in a kinetic context, as the number of proteins produced per mRNA per hour (Schwanhäusser et al, 2011).

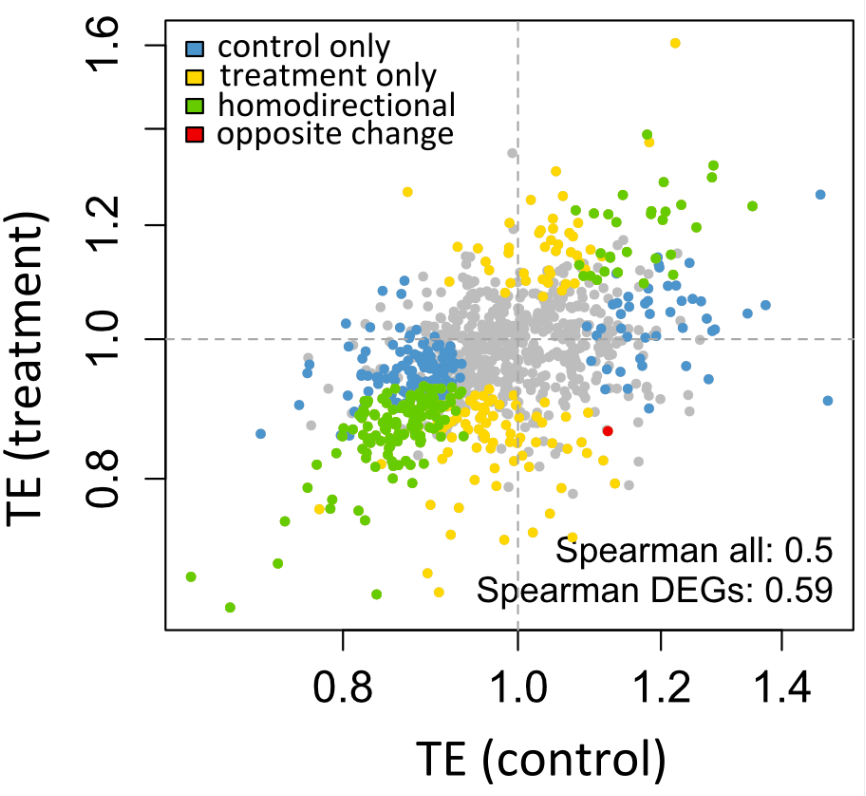

Figure 11: Scatterplot of translational efficiencies (TEs). Choosing the TE method in the computeDEGs function, tRanslatome calculates and displays the translational efficiencies of each gene. In this example, TEs are defined as the ratio of polysomal and total signals. The scatterplot displays each gene according to its translational efficiency in the first condition (the control condition in this example, represented on the x-

axis) and the translational efficiency in the second condition (the treatment condition, represented on the y-axis). As with the other DEGs selection methods, we adopt a color code to label different classes of DEGs: blue for genes with a TE significantly different from 1 only in the control condition; yellow for genes with TE significantly different from 1 only in the treatment condition, green for genes with a TE significantly different from 1 in both the conditions, red for genes whose TE changes antidirectionally between the two conditions.

In tRanslatome, TE is intended as any ratio between two reading levels. The TE method first calculates TE as the ratio of the signals of the two “-omics” levels provided by the user, then select DEGs according to differences in the TE between the treatment condition and the control condition. After choosing the TE method, all results are centered around the TE values, not the fold change values. The scatterplot resulting from the TE method applied to the worked example are displayed in Figure 11.

3 Second worked example

The second worked example shows how transcriptome and proteome data can be compared between two different human cell lines, HeLa and MCF7. The dataset used for this example is taken from (Stevens and Brown, 2013). In this work, the authors study the transcript levels and the protein levels in five different human cell lines. This worked example presents two levels of gene expression analysis (the transcriptome and the proteome) on two different cell lines (HeLa, labeled as ”HeLA” vs. MCF7, labeled as ”MCF7”). This dataset provides only one replicate, therefore DEGs are selected applying exclusively a threshold on the log2 fold change and choosing “none” as DEGs selection method.

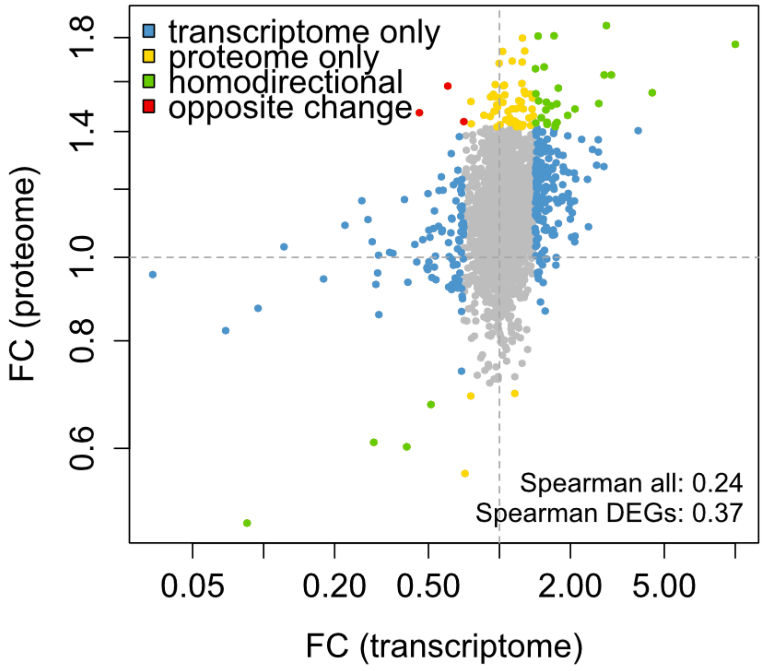

Figure 12: figure resulting from the method Scatterplot(), applied to the second worked example. Each gene is mapped according to its fold change at the first level (on the x- axis) and the fold change at the

second level (on the y-axis). The track parameter was set to c("GAB1","FMR1","EGR1"). We adopt a color code to label different classes of DEGs: blue for genes differentially expressed only at the first level; yellow for genes differentially expressed only at the second level, green for genes changing homodirectionally, red for genes with opposite changes.

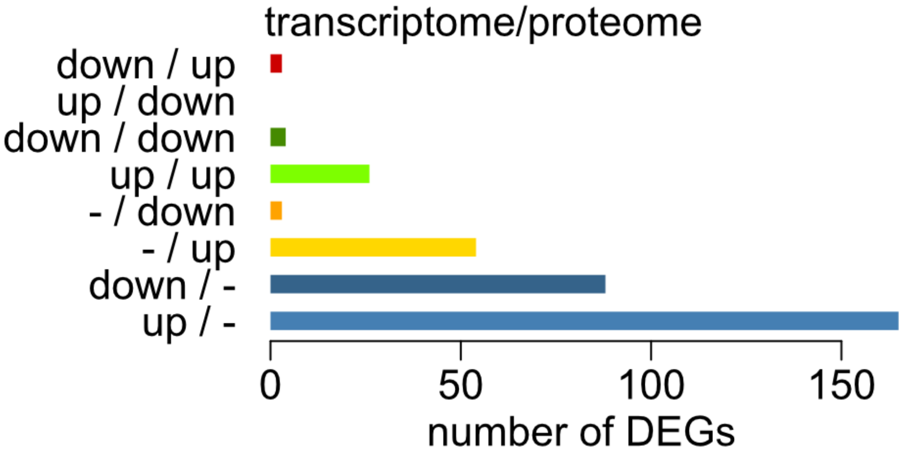

Figure 13: Detailed histogram of the identified differentially expressed genes (DEGs). The histogram displays the number of genes up or down regulated only in the transcriptome (”up/-” or ”down/-”, in blue tones), only in the proteome (”-/up” or ”-/down”, in yellow tones), in both expression levels with the same trend (”up/up” or ”down/down”, in green tones), in both expression levels with opposite directions (”up/down” or ”down/up”, in red tones).

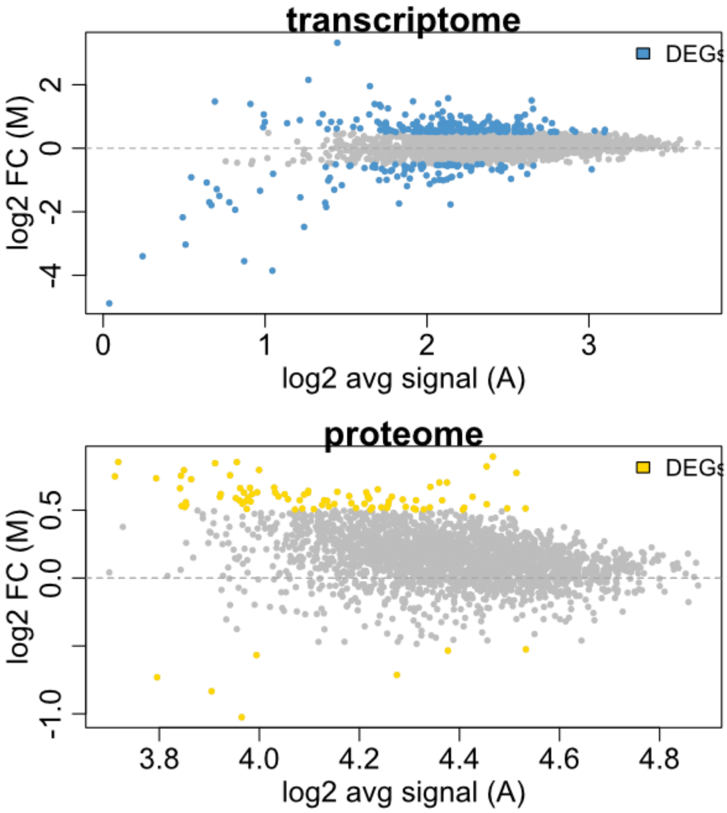

Figure 14: MA plots are displayed for the first expression level (the transcriptome, in the upper panel, with DEGs labeled in blue) and the second expression level (the proteome, in the lower panel, with DEGs labeled in yellow).

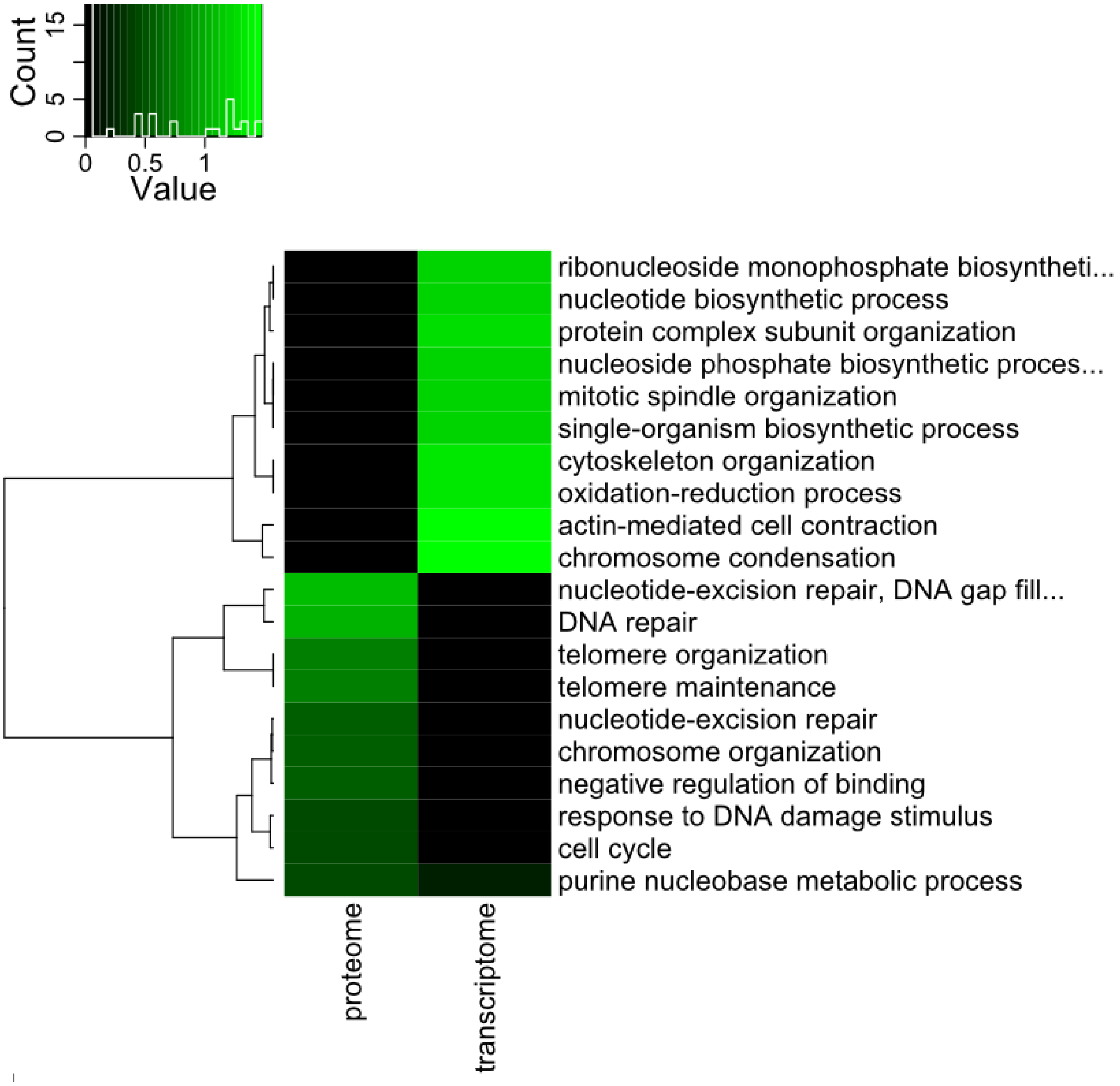

Figure 15: Heatmap of the top enriched GO Biological Process terms for each expression level, transcriptome and proteome in the worked example. The color scale is based on the -log10 transformed enrichment p-value. Black cells are associated to not significant enrichments. GO terms are clustered according to the enrichment p-values across the two "-omics" levels.

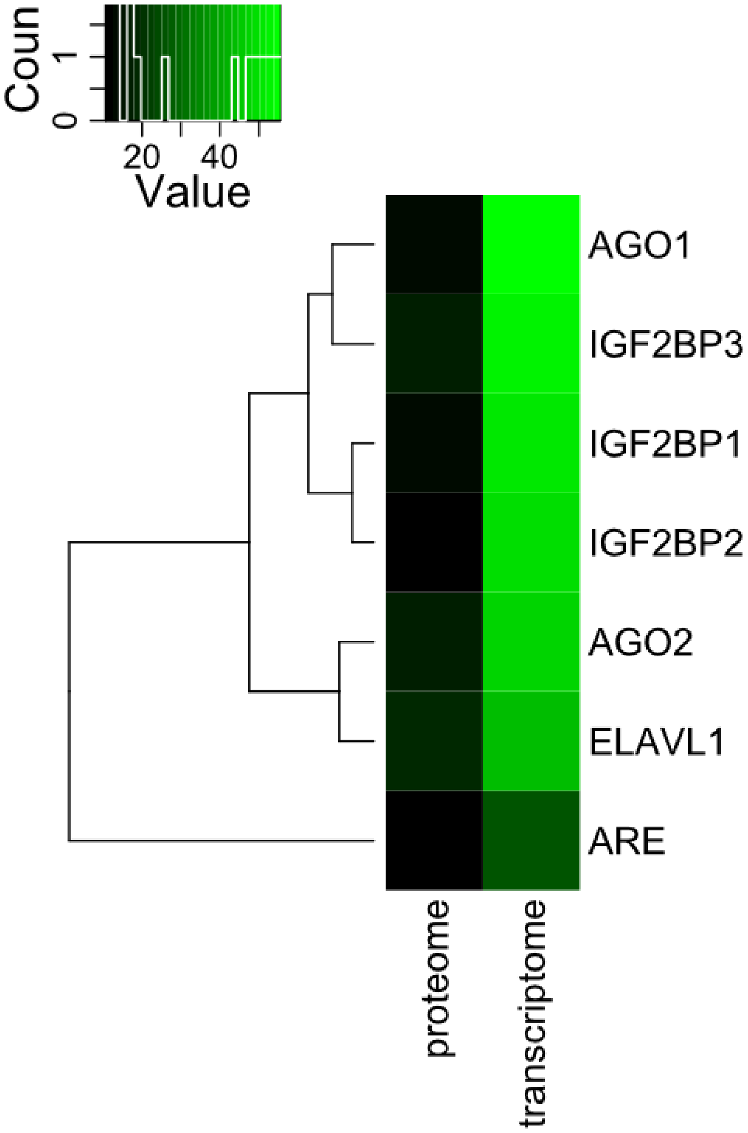

Figure 16: Heatmap of the top enriched AURA post-transcriptional regulatory elements for each expression level, transcriptome and proteome in the worked example. The color scale is based on the -log10 transformed enrichment p-value. Black cells are associated to not significant enrichments. Regulatory elements are clustered according to the enrichment p-values across the two "-omics" levels.

4 Bibliography

Alexa,A. *et al.* (2006) Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*, **22**, 1600–7.

Anders,S. and Huber,W. (2010) Differential expression analysis for sequence count data. *Genome Biol.*, **11**, R106.

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–9.

Breitling,R. *et al.* (2004) Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments. *FEBS Lett.*, **573**, 83–92.

Ingolia,N.T. *et al.* (2011) Ribosome profiling of mouse embryonic stem cells reveals the complexity and dynamics of mammalian proteomes. *Cell*, **147**, 789–802.

Larsson,O. *et al.* (2011) anota: Analysis of differential translation in genome-wide studies. *Bioinformatics*, **27**, 1440–1.

Parent,R. *et al.* (2007) Mammalian target of rapamycin activation impairs hepatocytic differentiation and targets genes moderating lipid homeostasis and hepatocellular growth. *Cancer Res.*, **67**, 4337–45.

Powley,I.R. *et al.* (2009) Translational reprogramming following UVB irradiation is mediated by DNA-PKcs and allows selective recruitment to the polysomes of mRNAs encoding DNA repair enzymes. *Genes Dev.*, **23**, 1207–20.

Robinson,M.D. *et al.* (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–40.

Smyth,G.K. (2004) Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat. Appl. Genet. Mol. Biol.*, **3**, Article3.

Stevens,S.G. and Brown,C.M. (2013) In silico estimation of translation efficiency in human cell lines: potential evidence for widespread translational control. *PLoS One*, **8**, e57625.

Tian,L. *et al.* (2005) Discovering statistically significant pathways in expression profiling studies. *Proc. Natl. Acad. Sci. U. S. A.*, **102**, 13544–9.

Tusher,V.G. *et al.* (2001) Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. U. S. A.*, **98**, 5116–21.

Wang,J.Z. *et al.* (2007) A new method to measure the semantic similarity of GO terms. *Bioinformatics*, **23**, 1274–81.