Exploring protein interactome data with IPinquiry: statistical analysis and data visualization by spectral counts

Lauriane Kuhn, Timothée Vincent, Philippe Hammann, Hélène Zuber

Exploring protein interactome data with IPinquiry: statistical analysis and data visualization by spectral counts

Lauriane Kuhn, Timothée Vincent, Philippe Hammann and Hélène Zuber

Abstract

Immunoprecipitation mass spectrometry (IP-MS) is a popular method for the identification of protein-protein interactions. This approach is particularly powerful when information is collected without *a priori* knowledge and has been successively used as a first key step for the elucidation of many complex protein networks. IP-MS consists in the affinity purification of a protein of interest and of its interacting proteins followed by protein identification and quantification by mass spectrometry analysis.

Lauriane Kuhn

Plateforme protéomique Strasbourg Esplanade du CNRS, Université de Strasbourg, 67000 Strasbourg, France e-mail: l.kuhn@ibmc-cnrs.unistra.fr

Timothée Vincent

Institut de biologie moléculaire des plantes, CNRS, Université de Strasbourg, 12 rue Zimmer, 67000 Strasbourg, France e-mail: vincent.timothee07@gmail.com

Philippe Hammann

Plateforme protéomique Strasbourg Esplanade du CNRS, Universitéde Strasbourg, 67000 Strasbourg, France e-mail: p.hammann@ibmc-cnrs.unistra.fr

Hélène Zuber

Institut de biologie moléculaire des plantes, CNRS, Université de Strasbourg, 12 rue Zimmer, 67000 Strasbourg, France e-mail: helene.zuber@ibmp-cnrs.unistra.fr

We developed an R package, named IPinquiry, dedicated to IP-MS analysis and based on the spectral count quantification method. The main purpose of this package is to provide a simple R pipeline with a limited number of processing steps to facilitate data exploration for biologists. This package allows to perform differential analysis of protein accumulation between two groups of IP experiments, to retrieve protein annotations, to export results and to create different types of graphics. Here we describe the step-by-step procedure for an interactome analysis using IPinquiry from data loading to result export and plot production.

Key words Immunoprecipitation, Mass spectrometry, Data processing, Differential analysis, Volcano plots, Spectral counts, R package

1 Introduction

Affinity purification mass spectrometry (AP-MS) or immunoprecipitation mass spectrometry (IP-MS) is a popular without *a priori* method for the identification of protein-protein interactions that has been successfully used for resolving numerous complex protein networks [1–3]. IP-MS is a first key step of the experimental workflow for protein partner identification and usually precedes validation using alternatives approaches. IP-MS starts with the affinity purification of the protein of interest, referred to as the bait protein, and of its interacting proteins by using a resin coupled to an antibody recognizing either the bait itself or an epitope tag expressed fused to the bait. The eluted protein mixture is then subjected to proteolytic digestion and identified by MS analysis (see Figure 1). The latter classically involves peptide separation by reverse-phase liquid chromatography combined with tandem mass spectrometry (LC-MS/MS). Experimental design of IP-MS approaches differs according to biological questions and available biological material. In particular, the use of appropriate controls is crucial as the eluted protein mixture contains *bona fide*

protein partners but also various non-specific interactors, such as proteins binding to the epitope tag or to the resin. A good control should enable assessing protein backgrounds resulting from the various contamination sources and its choice should not be neglected. Classically, when the aim is to analyze the protein interactome of a protein of interest by using cells expressing a tagged bait protein, control IPs are performed using wild-type cells, that do not express the tagged bait protein, and/or using cells that express an unrelated tagged protein. Proteins found to be enriched in bait compared to control IPs are then considered as potential protein partners. Alternatively, when the question is to test the impact of a particular protein motif or domain on the protein interactome, IPs using the wild-type version of the bait protein are compared to the one using a mutated version. The potentially interesting proteins correspond then to those depleted in mutant IPs. Finally, a frequent question is also to test the impact of different conditions or treatments on the interactome of a protein of interest. IPs performed from samples of different conditions are then compared and both significantly enriched and depleted proteins are considered as potentially interesting. In all cases, the data analysis consists in comparing the differential accumulation of proteins between two groups of IPs. Two metrics can be used for protein quantification: the spectral count, defined as the total number of spectra identified for a protein, and the peptide abundance derived from MS1 peak area [4]. The second strategy is now often preferred notably because of its higher performance for the detection of low abundant proteins. Yet, the spectral count quantification method still represents a popular fast and simple approach that demonstrates its efficiency in IP-MS approaches to resolve protein interactomes.

Here we describe the analysis of IP-MS data based on spectra counts using the `IPinquiry` R package. The main purpose of this package is to provide a simple R pipeline with a limited number of processing steps to facilitate as much possible data

exploration and plot creation for biologists. `IPinquiry` compiles several functions to: i) identify proteins significantly enriched or depleted between two groups of IP experiments, ii) retrieve annotations for detected proteins, iii) export result tables, and iv) create different graph types, such as interactive volcanoplots that display protein changes (fold changes) according to statistical significance (p-value). In order to calculate p-values associated with protein accumulation changes, the package uses the negative binomial generalized linear models, with or without quasi-likehood tests, implemented in the `EdgeR` package [5, 6]. `EdgeR` GLM models were developed for RNA-seq analysis to assess gene differential analysis between two conditions or genotypes. RNA-seq and proteomic data share common features in a statistical point of view: both types of data are discrete, are usually linked to high biological dispersion and to a reduced number biological replicates, often below 5. Because of these common properties, the `EdgeR` GLM model was previously proposed for analyzing MS-MS data [7] and was already successfully applied to explore protein interactome based on IP-MS experiments [8–10]. We detailed hereafter the step-by-step procedure for data analysis based on spectral counts using `IPinquiry` from data loading to result export and plot production. The package includes example datasets from [11] to help users apprehending `IPinquiry` utilization.

2 Material

2.1 Considerations for IP-MS approaches

1. Choosing a good antibody.

   (a) When available, IP can be performed using an antibody against the protein of interest. This is the ideal situation as protein-protein interactions can be analysed at physiological level.

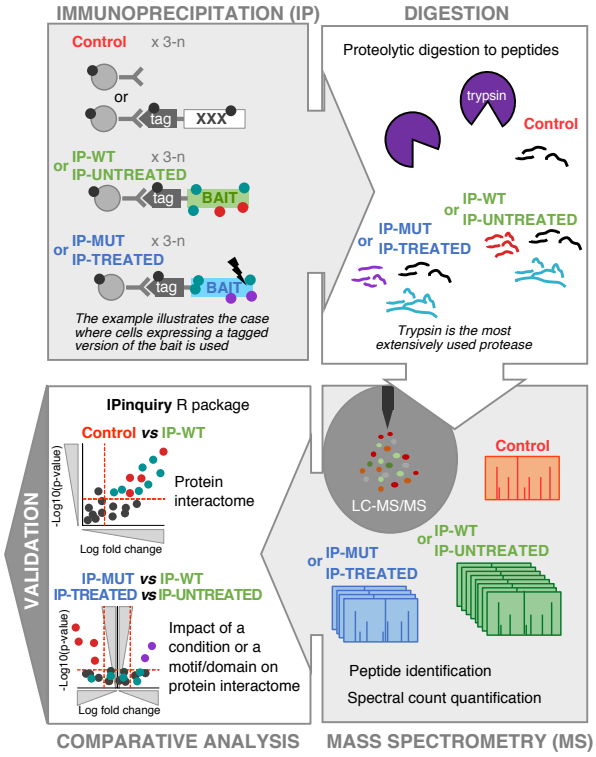

Fig. 1 Schematic overview of main steps of IP-MS approaches

(b) When such an antibody is not available, the protein of interest fused to an epitope tag needs to be of expressed in cells (*see* **Note 1**).

2. Choosing good controls for IPs, the objective being to remove as much as possible contaminant proteins.
3. Optimizing affinity purification conditions (*see* **Note 2**): optimization of the sample lysis, homogenization and grinding, choice of detergents and bead type, adjustment of wash and elution stringency, etc. The goal is to obtain an efficient and reproducible purification with a good balance between the number of detected proteins and the number of contaminants.

4. Optimizing and standardizing the quantity of starting material for IP and MS. Once optimized, the same number of cells or the same dry weight from plant has to be used.
5. Selecting the number of biological replicates (*see* **Note 3**). This choice depends on the expected variation and on the biological and technical variability. We recommend analyzing at least three biological replicates (*see* **Note 4**). We also encourage the analysis of different transgenic lines, when using cells expressing the tagged bait protein.
6. Selecting the method for protease digestion. Gel-free trypsin digestion is widely used and well suited in most cases as it allows to better control the reproducibility of the process compared to gel-purified protein complexes.
7. Selecting ion source. Electrospray Ionization Source (ESI) is the most widely used ion source.
8. Selecting mass analyzers. Sensitivity and high duty cycle of actual mass spectrometers allow to efficiently identify moderately complex mixtures, like in the case of an AP-MS sample.
9. Selecting and optimizing LC-MS parameters. For protein quantification with LC-MS, we favor long chromatographic gradients and a single injection under discovery mode for each sample. Several methods of data acquisition are also available such as data-dependent acquisition, targeted acquisition and data-independent acquisition. Finally, a key parameter is the dynamic exclusion time that needs to be optimized to obtain a good balance between the number of detected proteins and the number of spectra obtained for each protein (*see* **Note 5**).
10. Selecting protein identification method. It depends on the MS acquisition strategy. Identification should be validated according to the actual guidelines (FDR<1% on both spectral and protein level) and protein redundancy should be carefully managed.

11. Selecting the quantification method, *i.e.* spectral count or peptide abundance derived from MS1 peak area. This chapter is dedicated to the analysis based on spectral counts (*see* **Note 6**).

2.2 Requirements

`IPinquiry` is a package written in R [12]. If not already done, R needs to be downloaded and installed. We also recommend the use of RStudio, which provides a nice R user interface making life easier for R beginners. In addition, the following R packages are needed (*see* **Note 7**).

1. for statistical analysis (required for package installation): `EdgeR` [5, 6], `limma` [13], `statmod` [14]
2. for the creation of interative volcanoplots : `plotly` [15], `htmlwidgets` [16]
3. for the creation of interactive tables : `DT` [17], `htmlwidgets` [16]
4. for the creation of others graphs : `ggplot` [18], `pheatmap` [19], `RColorBrewer` [20]
5. for protein annotation: `biomaRt` [21, 22]
6. for saving result tables as excel files: `xlsx` [23]

2.3 Software installation

`IPinquiry` can be downloaded and installed from Github using devtools [24]. If needed, install devtools and load the library:

```
install.packages("devtools")
library(devtools)
```

`IPinquiry` package can then be installed (*see* **Note 8**).

```
install_github("https://github.com/hzuber67/IPinquiry4")
```

2.4 Data format

Input data consist in two files :

1. a **Count table** (text file with tab-separated values) that contains spectral counts for all proteins detected in IPs. Each row corresponds to one protein detected in IP and each column corresponds to one IP experiment (see Figure 2).

	accession	Mut_L0_1	Mut_L0_2	Mut_L3_1	Mut_L3_2	URT1_L12_1_2019	URT1_L12_2_2019	URT1_L17_1_2019	URT1_L17_2_2019
1	AT1G01080.2	0	0	1	1	0	1	1	1
2	AT1G01090.1	1	2	2	2	1	2	1	2
3	AT1G01100.1	0	1	0	1	0	2	0	0
4	AT1G01300.1	11	11	10	9	8	10	9	14
5	AT1G01320.1	1	1	1	0	2	1	0	1

Fig. 2 Screenshot of the count table for the first dataset (top part)

2. a **Sample table** (text file with tab-separated values) that gives information about samples. First column indicates the IP names, second column the conditions and finally the third column is optional and allows for indicating potential batch effect, related to different experiment times for example (see Figure 3) (*see* **Note 9**).

	IP_names	sample
1	Mut_L0_1	M1
2	Mut_L0_2	M1
3	Mut_L3_1	M1
4	Mut_L3_2	M1
5	URT1_L12_1_2019	urt1
6	URT1_L12_2_2019	urt1
7	URT1_L17_1_2019	urt1
8	URT1_L17_2_2019	urt1

	IP_names	sample	batch
1	F016864_2014_S17_control_C1	control	one
2	F016865_2014_S17_control_C2	control	one
3	F016867_2014_S17_URT1_H1	urt1	one
4	F016878_2014_S25_control_C5	control	one
5	F016880_2014_S25_URT1_H5	urt1	one
6	F016882_2015_S12_CTRL_1	control	two
7	F016883_2015_S12_CTRL_2	control	two
8	F016884_2015_S12_CTRL_3	control	two
9	F016886_2015_S12_URT1_myc2	urt1	two
10	F016887_2015_S12_URT1_myc3	urt1	two

Fig. 3 Screenshots of sample tables for the first (on the left) and the second (on the right) datasets

2.5 Example dataset

Two example datasets corresponding to IP experiments in Arabidopsis [11] (*see* **Note 10**) are included in the package.

1. The first dataset contains results for two groups of IPs performed from plants expressing a wild-type version of the URT1 TUTase fused to an epitope tag, named URT1-myc, or a mutated version, named m1URT1-myc. Each group is composed of four replicates. The goal was to test the impact of the M1 motif of URT1 on its interactome *in planta* (*see* **Note 11**). Directories for the example and sample tables are:

   ```
   > CountTable1 <- system.file("extdata", "CountTable1.txt",
   package = "IPinquiry4")
   > SampleTable1 <- system.file("extdata", "SampleTable1.txt",
   package = "IPinquiry4")
   ```

   Top part of the count table can be vizualized as follow (see Figure 2):

   ```
   > Count_tb1 <- read.table(CountTable1, sep="\t", header=TRUE)
   > head(Count_tb1)
   ```

   Sample table can be vizualized as follow (see Figure 3):

   ```
   > Sample_tb1 <- read.table(SampleTable1, sep="\t", header=TRUE)
   > print(Sample_tb1)
   ```

   Conditions in the sample table are named "urt1" and "M1" for URT1-myc or m1URT1-myc IPs, respectively.

2. The second dataset contains results for four replicates of IPs performed from plants expressing the wild-type version of URT1 fused to an epitope tag. Control

IPs were performed in parallel using wild-type plants that do not express the tagged URT1 with six biological replicates. The goal here was to identify protein partners of the URT1 TUTase *in planta*. IP experiments were performed for two different tissues at two different times inducing a batch effect that will be latter taken into account in the statistical model. Directories for the example count and sample tables are:

```
> CountTable2 <- system.file("extdata", "CountTable2.txt",
package = "IPinquiry4")
> SampleTable2 <- system.file("extdata", "SampleTable2.txt",
package = "IPinquiry4")
```

Top part of the count table can be vizualized as follow:

```
> Count_tb2 <- read.table(CountTable2, sep="\t", header=TRUE)
> head(Count_tb2)
```

Sample table can be vizualized as follow (see Figure 3):

```
> Sample_tb2 <- read.table(SampleTable2, sep="\t", header=TRUE)
> print(Sample_tb2)
```

Conditions in the sample table are named "urt1" and "control" for bait and control IPs, respectively. The sample table contains a third column indicating a batch effect.

2.6 Data loading

1. To load `IPinquiry` library in your R environment, enter in the R console:

```
> library(IPinquiry4)
```

2. IP data can then be loaded using `load_IP_data` function (*see* **Note 12**). Here, the two example datasets are successively loaded by indicating their directories as defined above (*see* Subheading 2.5).

```
> # Load dataset1
> IP_data1 <- load_IP_Data(CountTable1, SampleTable1)
> # Load dataset2
> IP_data2 <- load_IP_Data(CountTable2, SampleTable2)
```

3. Arguments taken by the function are the directories for count and sample tables. When analyzing your own data, simply indicate their directories on your computer. For example :

```
> my_IP_data <- load_IP_Data("/Users/me/Documents/my_count_table.txt",
"/Users/me/Documents/my_sample_table.txt")
```

3 Methods

3.1 Visualization of the overall variability between samples

Multidimensional scaling (MDS) plots can be used to visualize distances or dissimilarities between the different IP experiments. Here, the Euclidean distance is used to perform the MDS.

1. MDS can be plotted based on raw data (norm="nothing", by default) or on normalized data either based on the total number of counts (`norm="total"`) or on the median-to-ratios method as used in DESeq2 R package [25] (`norm="DEseq"`) (*see* **Note 13**).

   MDS without prior normalization can be obtained as follow (see Figure 4):

```
# MDS for the first dataset
```

```
> MDSplot(IP_data1)
```

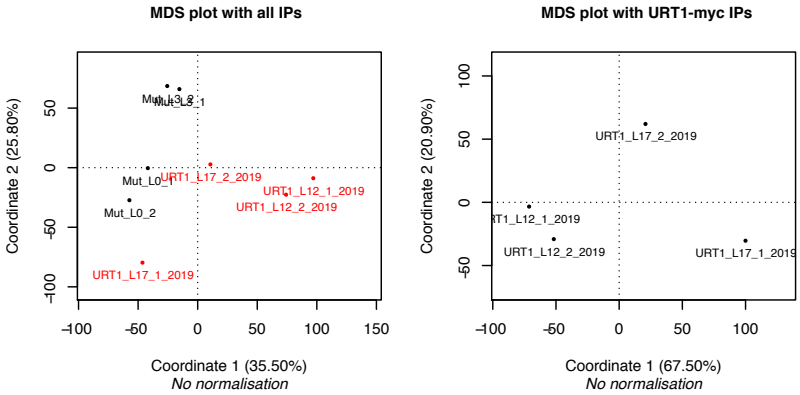

Fig. 4 MDS plot for the first dataset with all or with URT1-myc IPs

2. Functions `subset{\_}IPObj{\_}treat` and `subset{\_}IPObj{\_}batch` can also be used to subset the dataset prior to plot the MDS. subset_IPObj_treat allows the selection of specific treatments/conditions. subset_IPObj_batch allows the selection of specific batches.

   Here we plot the MDS for the treatment "urt1" (see Figure 4):

   ```
   # MDS for the first dataset for the treatment "urt1"
   IP_urt1 <- subset_IPObj_treat(IP_data2, "urt1")
   MDSplot(IP_urt1)
   ```

3.2 Statistical analysis for differential analysis

The statistical analysis is based on the GLM model developed by the `EdgeR` package [5, 6]. This model was previously proposed to analyze MS data based on spectral counts in the `msmsTests` package [7]. A refined description of these statistical

models are provided in Chapter XXX. By default, `IPinquiry` package uses the Genewise Negative Binomial Generalized Linear Model with Quasi-likelihood Tests implemented in `EdgeR` (*see* **Note 14**).

1. Statistical comparison needs to be performed for each pairwise comparison. For the first dataset, we have two treatments, `"urt1"` and `"M1"`, and :

   (a) Low abundance protein are filtered out before the calculation of the dispersion. Here, only proteins with a total sum of counts above 10 are used. This correspond to the 50% most abundant proteins (`min.disp=10`) (*see* **Note 15**).

   (b) The size correction factor (offset) is calculated using the median to ratio method [25] (`div="DEseq"`, (*see* **Note 16**)).

   ```
   > test1 <- stat_test(IP_data1, "urt1", treatment = "M1",
   div="DEseq", min.disp=10)
   ```

2. This function return a data frame with six columns: Protein ID as row.names, LogFC, quasi-likelihood F-statistics, P-values as calculated by `EdgeR`, P-values ajusted according to the Benjamini and Hochberg method and the protein rank based on ajusted p-values.
3. In the case of the first dataset, we are interested in identifying proteins that are significantly depleted when URT1 M1 motif is mutated. The list of proteins significantly depleted in m1URT1-myc compared URT1-myc IPs can be visualized as follows (see Figure 5):

   ```
   > print(subset(test1, test1$adjp<0.05&test1$LogFC<0))
   ```

4. A batch effect can be taken into account into the statistical model by adding `batch="TRUE"` (*see* **Note 17**). If so, a third column has to be added in the sample table to indicate the batch of each IPs (*see* Subheading 2.4). For the second dataset, we take into account the batch effect as two sets of experiments were performed.

	LogFC	F	p.value	adjp	number
AT1G26110.1	-2.6756838	126.78115	9.505361e-19	1.559830e-15	1
AT5G45330.1	-6.0490263	75.81482	2.016451e-11	1.102999e-08	3
AT2G45810.1	-1.2679256	50.79735	1.550362e-08	6.360360e-06	4
AT4G00660.2	-1.0963299	41.08400	3.488022e-07	1.144769e-04	5
AT3G13300.1	-0.9810462	35.63486	1.990222e-06	4.665648e-04	7
AT3G61240.1	-1.0198458	33.62240	4.228655e-06	8.674029e-04	8
AT4G20360.1	-0.8466879	31.30745	7.568390e-06	1.379970e-03	9
AT1G27090.1	-2.1257515	30.70430	2.544579e-05	4.175654e-03	10
AT1G48410.2	-1.7462989	27.82227	6.489081e-05	9.680530e-03	11
AT3G58510.1	-0.8828883	24.12936	1.003564e-04	1.372373e-02	12
AT3G58570.1	-0.9571052	22.12210	2.273244e-04	2.664566e-02	14
AT2G42520.1	-0.9563024	20.29599	4.466130e-04	4.311129e-02	16
AT5G47010.1	-1.4477632	21.46352	4.218413e-04	4.311129e-02	17
AT4G38680.1	-1.0716476	20.39479	5.152760e-04	4.632652e-02	18
AT5G40490.1	-1.5120287	20.85829	5.363826e-04	4.632652e-02	19

Fig. 5 Data frame with statistical results for significantly depleted proteins

```
> test2 <- stat_test(IP_data2, "control", treatment = "urt1",
div="DEseq", batch=TRUE)
```

5. In the case of the second dataset, we are interested in identifying proteins that are significantly enriched in URT1 IPs compared to control IPs. These proteins will be considered as potential protein partners of URT1. The list of proteins significantly enriched can be vizualized as follow:

```
> # Subset enriched proteins
> test2_enriched <- subset(test2, test2$adjp<0.05&test2$LogFC>0)
> # Print subtable
> print(test2_enriched)
```

6. By adding the argument `glm="classic"`, you can use instead the `EdgeR` function based on the Genewise Negative Binomial Generalized Linear Models without Quasi-likelihood Tests (*see* **Note 18**).
7. An additional low abundance filter can be added, *e.g.* `filter=5` (*see* **Note 19**). If a filter value is indicated, the output data frame includes a seventh column

indicating if, "YES" or "NO", proteins meet this additional criterion. This filter does not affect the statistics calculation.

3.3 Retrieve annotations for each protein

Functional annotations are retrieved using the `biomaRt` package [21, 22]. Annotation are collected from the Ensembl database [26]. Active internet connection is necessary to access the remote database and query it on-line.

1. Here, annotations from *Arabidopsis thaliana* are retrieved.

```
> annotated_table_At <- addBiomaRtAnnotation(test,
biomart="plants_mart", dataset="athaliana_eg_gene")
```

2. By default, the function searches for Ensembl peptide identifiers. This argument needs to be adjusted according to the identifiers used in the row names of the count table. It can be `"ensembl_peptide_id"`, `"ensembl_transcript_id"`, `"ensembl_gene_id"` or `"external_gene_name"`.
3. The new output data frame contains three additional columns: `ensembl ID`, `external gene name` and `description`
4. Of course, this function can be used for all other species for which annotations are available at Ensembl. `biomart`, `dataset` and `host` arguments need to be adjusted according to the analyzed species.

   (a) For listing available databases:

```
> library(biomaRt)
> listMarts()
> dataset_list <- listDatasets(useMart("ENSEMBL_MART_ENSEMBL"))
> print(dataset_list)
```

(b) For example, for *Drosophila melanogaster*:

```
> annotated_table_Dm <- addBiomaRtAnnotation(droso_results,
biomart = "ENSEMBL_MART_ENSEMBL",
dataset = "dmelanogaster_gene_ensembl", host = "www.ensembl.org")
```

(c) Another example, for human:

```
> annotated_table_Hs <- addBiomaRtAnnotation(human_results,
biomart = "ensembl", dataset = "hsapiens_gene_ensembl",
host = "www.ensembl.org", features="external_gene_name")
```

3.4 Create and export an html table

`IPinquiry` package includes a function based on `DT package` [17] to create an interactive table with results.

1. The following code creates an interactive table from the `annotated_table_At` data frame that contains statistical results and protein annotations:

```
> # Interactive table for dataset 1
> createTable(annotated_table_At)
```

2. This table is interactive and can be used to sort and search proteins, select and copy interesting rows, export results, etc.(see Figure 6)
3. This interactive table can also be saved as an *html* file.

```
> p <- createTable(annotated_table_At)
> htmlwidgets::saveWidget(p,"interactive_table_1.html",
selfcontained = TRUE)
```

4. You can also specify the directory where the file has to be saved (*see* **Note 20**):

Copy CSV Excel Search:

	LogFC	adjp	external_gene_name	description
AT1G26110.1	-2.67568375781914	1.5598298212177e-15	DCP5	Protein decapping 5 [Source:UniProtKB/Swiss-Prot;Acc:Q9C658]
AT5G41790.1	2.07394390136945	3.6645744207509e-9	CIP1	COP1-interactive protein 1 [Source:UniProtKB/Swiss-Prot;Acc:F4JZY1]
AT5G45330.1	-6.04902632434931	1.10299859702332e-8	DCP5-L	Decapping 5-like protein [Source:UniProtKB/Swiss-Prot;Acc:Q9FH77]
AT2G45810.1	-1.26792563395301	0.00000636036009242441	RH6	DEAD-box ATP-dependent RNA helicase 6 [Source:UniProtKB/Swiss-Prot;Acc:Q94BV4]
AT4G00660.2	-1.09632992307729	0.000114476886893135	RH8	DEAD-box ATP-dependent RNA helicase 8 [Source:UniProtKB/Swiss-Prot;Acc:Q8RXK6]
AT3G45140.1	0.650623330055275	0.000237262866856185	LOX2	Lipoxygenase 2, chloroplastic [Source:UniProtKB/Swiss-Prot;Acc:P38418]
AT3G13300.1	-0.981046170327883	0.000466564823690176	VCS	Enhancer of mRNA-decapping protein 4 [Source:UniProtKB/Swiss-Prot;Acc:Q9LTT8]
AT3G61240.1	-1.01984579214868	0.000867402913446643	RH12	DEAD-box ATP-dependent RNA helicase 12 [Source:UniProtKB/Swiss-Prot;Acc:Q9M2E0]

Fig. 6 Screenshot of the interactive table with statistical results and protein annotations. The interative table allows data sorting, section and export.

```
> htmlwidgets::saveWidget(p,
"/Users/me/Documents/My_results/interactive_table_1.html",
selfcontained = TRUE)
```

3.5 Export result table as excel or text file

Alternatively, results table can also be saved:

1. As an excel file, using the `xlsx` package with the following command line:

```
> library(xlsx)
> write.xlsx(annotated_table_At, "IP_results.xlsx",
sheetName = "Statistics")
```

2. As a text file, using `write.table` R function:

```
> write.table(annotated_table_At, "IP_results.txt", sep="\t",
col.names=NA, quote=FALSE)
```

3. As previously (*see* Subheading 3.4), you can specify the directory where the file has to be saved for both functions. For example for `write.table` function:

```
> write.table(annotated_table_At,
"/Users/me/Documents/My_results/IP_results.txt",
sep="\t", col.names=NA, quote=FALSE)
```

3.6 Create interactive volcanoplot

1. The interactive volcanoplot created by `IPinquiry` is based on the `Plotly` R package [15]. The volcanoplot shows the log2 fold change according to p-value or to adjusted p-value.

   Volcanoplot for the first dataset (see Figure 7):

```
> #Dataset 1 volcanoplot
> htmlPlot(annotated_table_At, sign="adjp")
```

   For the second dataset, we are interested in the proteins that are enriched in URT1 IP when compared to control IP. The volcanoplot is drawn only for enriched proteins, with `LogFC>0` (see Figure 7).

   Volcanoplot for the second dataset:

```
> #Dataset 2 volcanoplot for enriched proteins
> htmlPlot(subset(test2, test2$LogFC>0), sign="adjp")
```

2. By default, point labels correspond to row names of the input table accompanied with the point coordinates. Custom texts can also be used instead using the `custom_text` argument. For example, the R code below allows using the 30 first letters of protein annotation found in the description column of the result table.

```
> # extract the 30 first letters of the description column
```

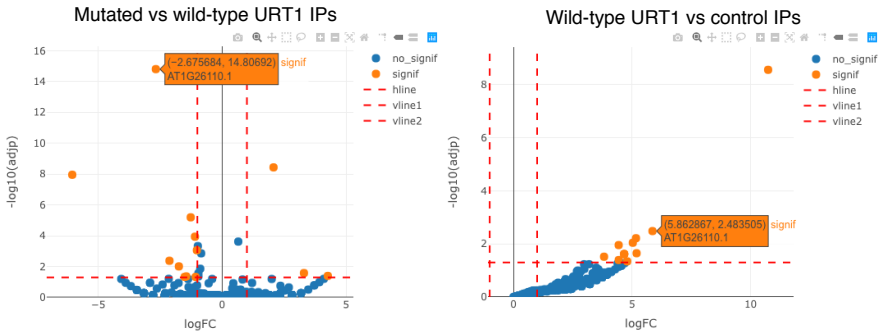

Fig. 7 Screenshots of interactive volcanoplots obtained for example datasets 1 and 2 shown on the left and on the right, respectively. Text labels of points appear when the cursor is moved over them.

```
> my_test = paste(row.names(annotated_table_At) , "_",
substr(annotated_table_At$description,1,30))
> # add annotation as point label for the dataset 1 volcanoplot
> htmlPlot(annotated_table_At,  custom_text=my_test)
```

3. By default, point colors are set according to the significance and dotted lines are set both according to p-value and LogFC. P-value and LogFC cut-offs can be ajusted using `max.pval` and `min.LFC` arguments. Defaut values are 0.05 and 1, respectively. Point colors can also be used to highlight specific proteins. For example , the R code below is used to pinpoint three proteins linked to decapping (*see* **Note 21**).

```
> # List of interesting proteins
> smallTrueList <- c("AT1G26110.1", "AT5G45330.1", "AT3G13300")
> # Create interactive plot
> htmlPlot(annotated_table_At, listGenes = smallTrueList,
custom_text=my_test)
```

4. There is also the possibility to set colors according to a supplemental column. This column can contain additionnal information for example concerning gene

ontology. IPinquiry package contains a text file with supplemental information related to the first dataset. This file is composed of two columns, a first one with protein identifiers and a second one with protein classifications based on their molecular function.

(a) Directory for this information table is :

```
> Supplemental <- system.file("extdata", "Supplemental_information.txt",
package = "IPinquiry4")
```

(b) When analyzing your own data, simply indicate the directory on your computer of the text table containing the information of interest. For example :

```
> Supplemental_me <- "/Users/me/Documents/Interesting_information.txt"
```

(c) The add_suppl_information function of IPinquiry can be used to combine your result table with another table containing classification criteria. The code below combines the result table for the first dataset with the information table.

```
> # Add a supplemental column with criteria for color classification
> annotated_table_At2 <- add_suppl_information(annotated_table_At,
Supplemental)
> head(annotated_table_At2)
```

(d) This new column can then be used to set point colors.

```
> # Create the volcanoplot with colors according to this new column
> htmlPlot(annotated_table_At2,
colforcolor = annotated_table_At2$Classification)
```

5. The interactive volcanoplot can be directly saved under html format.

```
> p <- htmlPlot(annotated_table_At2,
```

```
colforcolor = annotated_table_At2$Classification)
> htmlwidgets::saveWidget(p,"interactive_volcanoplot_plot.html",
selfcontained = TRUE)
```

3.7 Create ggplot2 based volcanoplot

1. `IPinquiry` also includes a function, named `PDF_Plot`, to create a volcanoplot based on the `ggplot2` package [18]. As previously, the volcanoplot shows the log2 fold change according to p-value or to adjusted p-value. The advantage of using `ggplot2` is that the volcanoplot can then be saved as a vector image, using pdf or eps format (*see* **Note 22**).

```
> # Volcanoplot for example dataset1
> PDF_Plot(annotated_table_At2)
> # Volcanoplot for example dataset2
> PDF_Plot(subset(test2, test2$LogFC>0), sign="adjp")
```

2. `PDF_Plot` function contains many arguments that can be adjusted (see Figure 8):

   (a) Point colors and sizes.

   (b) Axis limits.

   (c) p-value and LogFC cut-offs. By default, `max.pval = 0.05` and `min.LFC = 1`.

   (d) Text labels and font size. Text labels are added only for proteins with a significant p-value.

   (e) Text labels and cut-off red lines can also be removed.

```
> # Custom volcanoplot for example dataset 1
> graph1 <- PDF_Plot(annotated_table_At2, sign="p.value", max.pval = 0.05,
min.LFC = 1, line=TRUE, point_color= c("gray", "purple"), min_x=-6, max_x=6,
min_y=0, max_y=20, point_size=3, label=TRUE, label_size=2,
```

```
custom_text=annotated_table_At2$external_gene_name,
title="Mutated vs wild-type URT1 IPs")
> graph1
>
> # Custom volcanoplot for example dataset 2
> graph2 <- PDF_Plot(subset(test2, test2$LogFC>0), sign="adjp",
max.pval = 0.05, line=FALSE, point_color= c("gray", "darkred"),
min_x=-0.5, max_x=11, min_y=-0.5, max_y=10, point_size=3,
label=TRUE, label_size=2, title="Wild-type URT1 vs control IPs")
> graph2
```

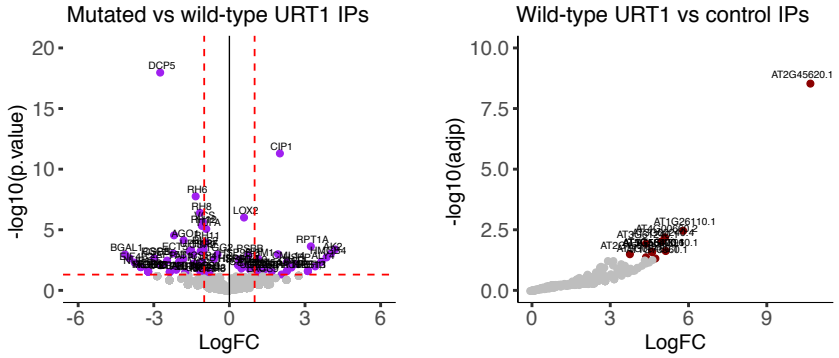

Fig. 8 Volcanoplots obtained using `PDF_Plot` function for example datasets 1 and 2 shown on the left and on the right, respectively. These graphs can be saved as pdf file using the `ggsave` function.

3. Volcanoplots can be saved as a pdf file using the ggsave function of `ggplot2` R package:

```
> library(ggplot2)
> ggsave("Volcanoplot1.pdf", graph1, height=3, width=5)
> ggsave("Volcanoplot2.pdf", graph2, height=3, width=5)
```

3.8 Create heatmap

1. The heatmap allows the visualization of protein expression pattern between samples. It can be useful when you have multiple groups and you want to sort your interesting proteins based on their abundance in the different groups of IPs. The function `IP_heatmap` creates an heatmap for a list of selected protein. This heatmap is performed based on the `pheatmap` package [19].
2. Heatmap can be drawn for all detected proteins or for a subset of interesting proteins, for example proteins that show differential accumulation according to the conditions. For the example dataset 1, the heatmap can be drawn for proteins related to RNA metabolism based on the functional classification in the last column of the result table (*see* Subheading 3.6).

```
> # Make a table with only proteins with classification linked
> # to RNA metabolism.
> # The code below removes proteins with empty classification (NA).
> class <- annotated_table_At2[
!is.na(annotated_table_At2$Classification),]
```

3. Nicely, `pheatmap` package allows also to add a color code based annotation for columns or rows. For example for the dataset 1, row color code can be added to indicate classification of the proteins used for the heatmap (see Figure 9). This color code can be added by specifying, as `annotation_row` argument, a dataframe with protein identifiers as row names and their corresponding classification as first column.

```
> # Creation of a one-column dataframe with
> # the classification for each selected protein
> # and the protein ID as row.names
> class2 <- class[,"Classification", drop=F]
```

4. Heatmap can be plotted based on raw data (`norm="nothing"`, by default), or on normalized data either based on the total number of count (`norm="total"`) or on the median-to-ratios method as used in DESeq2 R package [25] (`norm="DEseq"`). Here, the median to ratio method (DEseq2) was used to normalize data (*see* **Note 23**).

```
> IP_pheatmap(IP_data1, GeneList=row.names(class), norm="DEseq",
annotation_row = class2, fontsize_row=8)
```

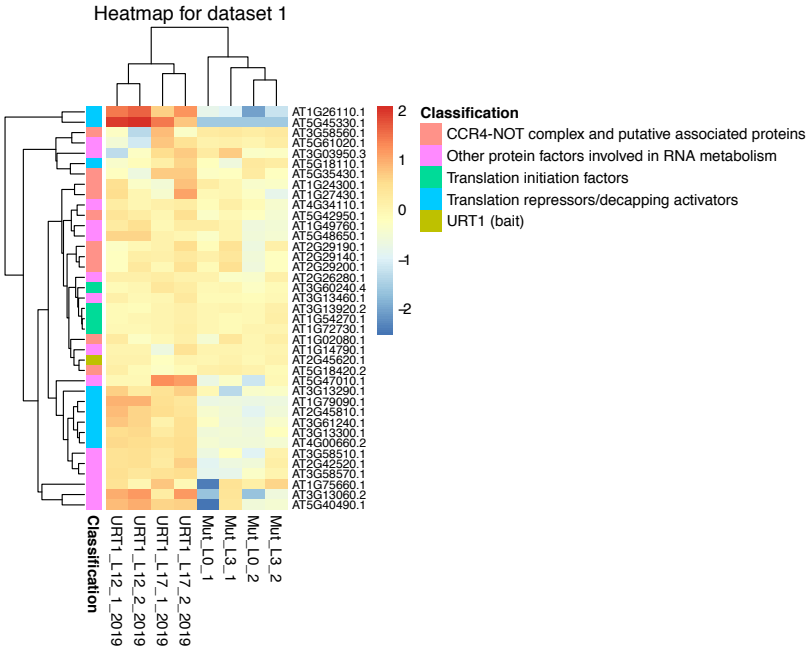

Fig. 9 Heatmap for the example dataset 1 drawn for proteins related to RNA metabolism. Color code on the left of the heatmap indicates the different classifications. These graphs can be saved as pdf file using `pdf` and `dev.off` functions.

5. Several other arguments of `IP_pheatmap` can also be adjusted:

   (a) font sizes.

   (b) title and its font size.

(c) hierarchical clustering of columns or rows can be removed.

For argument usages enter :

```
>help(IP_heatmap)
```

6. The heatmap can also be saved as a pdf file using `pdf` and `dev.off` R function.

```
> pdf("Heatmap_dataset1.pdf", width=8, height=6)
> IP_pheatmap(IP_data, GeneList=row.names(class), norm="DEseq",
annotation_row = class2, fontsize_row=8,title="Heatmap for dataset 1")
> dev.off()
```

4 Notes

1. Stable expression system should be favored and, when possible, the expression level of the tagged protein should be as close as possible of the endogenous level of the protein of interest in order to better reflect physiological protein-protein interactions.
2. In this chapter, we focus on IP-MS approach but other affinity approaches are often used in interactomics, as for example the Tandem Affinity Purification tag system (TAP-tag) [27].
3. Biological replicates are parallel measurements of biologically distinct samples that reflect random biological variation whereas technical replicates are repeated measurements of the same biological sample that reflect random technical variability [28]. In the context of IP-MS, technical variability can be linked to sample preparation or affinity purification.
4. In addition to biological replicates, we also encourage analyzing affinity replicates in a first approach to evaluate technical variability related to affinity purification.
5. Dynamic exclusion can be enabled or not for spectral count based quantification. Yet, [29] showed that enabling dynamic exclusion leads to higher peptide counts and

better reproducibility for the detection of relatively low abundant proteins. They found that the optimal duration of this exclusion depends on the average width of the chromatographic peak, mass spectrometry parameters and sample complexity.

6. The quantification method based on spectral counts is a fast and simple approach to resolve a short list of potential protein interactors. One shortcoming of the spectral count approach is its limitation towards the detection of low abundant proteins that can lead to an underestimation of differentially accumulated proteins. Alternatively, or as a complement, MS1 peak area quantification can be performed on the same MS raw files. Of note, the statistical model implemented in `IPinquiry` is not appropriate for statistical analysis based on MS1 peak area quantification

7. These packages are not automatically installed when installing `IPinquiry` and have to be installed from CRAN [12] or Bioconductor [30].

8. `IPinquiry` is meant to evolve in order to allow for bug fixes and/or improvements. Please update `IPinquiry` regularly.

9. Sample names in count and sample tables have to be identical and must not start with numbers.

10. In this study [11], Scheer, de Almeida et *al.* performed interactomic and functional analyses of the TUTase URT1, the main enzyme responsible for mRNA uridylation in Arabidopsis. Their data supports that URT1 participates in a molecular network connecting several translational repressors/decapping activators.

11. M1 motif is a short linear motif in the N-terminal region of URT1. In [11], M1 was shown to mediate direct interaction between URT1 and DCP5, a decapping activator.

12. Documentation can be accessed by using the R function `help` for each function of the `IPinquiry` package.

13. `IPinquiry` includes three methods of data normalization. When `norm="nothing"` is used, scale factor is set to 1. When `norm="total"` is used, spectral counts are di-

vided by the total number of counts and scale factors are calculating using the R code `div<- apply(data,2,sum)`. Finally, when `norm="DEseq"` is used, counts are divided by sample-specific size factors determined by median ratio of spectral counts relative to geometric mean per protein. The geometric mean is calculated using the R code `prod(x)^(1/n)` with `n <- length(x)`. The scale factor is calculated using the R code `div <- apply((data+1)/ apply(data + 1,1,gmean),2, median)`. By default, `norm="total"`.

14. When `glm=QL`, the `stat_test` function applies the three following `EdgeR` functions: estimateDisp, glmQLFit, glmQLFTest.

15. Low abundance proteins can adversely affect the dispersion estimation. The `min.disp` argument allows users to set an appropriate cut-off value for the calculation of the dispersion. Only proteins with total sum of counts above this value are used. By default, the cut-off value used by the `EdgeR` `estimateDisp` function is 5.

16. GLM models implemented in `EdgeR` and `msmsTest` packages normalize data with the help of an offset term in the model. `IPinquiry` includes three alternative ways for the offset calculation : no normalization (`norm="nothing"`), normalization using the total number of counts (`norm="total"`) and normalization based on the median-to ratio method (`norm="DEseq"`) (*see* **Note 13** for details about scale factor calculation).

17. If `batch=TRUE`, the batch variable is added as a blocking factor in the GLM model.

18. when `glm=classic`, the stat_test function applies the three following `EdgeR` functions: `estimateDisp`, `glmFit`, `glmLRT`. Output includes the same elements except that quasi-likelihood F-statistics values are replaced by likelihood ratio statistics value. This model is the one included in `msmsTest` package [7].

19. "YES" or "NO" tags indicate proteins with sum of counts across all IPs higher or lower than this filter value, respectively.

20. The directory where data are saved can be specified for all functions allowing data export, for example in this pipeline for `saveWidget`, `write.xlsx`, `write.table`, `ggsave` and `pdf`.

21. Decapping is a critical step of mRNA degradation and consists in the hydrolysis of the 5' cap structure of mRNA. Data in Scheer, de Almeida et *al.* suggest that URT1 connects decapping activators.

22. The aim of `PDF_Plot` is to facilitate the creation of graphs that are suitable for pdf or eps saving. Of course, if you are familiar with the `ggplot2` package, you can skip the `PDF_Plot` function and use the `ggplot2` suite of functions. You will then have the possibility to control more graphical parameters.

23. Counts are log2 transformed using the equation :

$$log2(x+1) \tag{1}$$

Acknowledgements The authors gratefully acknowledge Dominique Gagliardi for proofreading of the manuscript, Johana Chicher and all persons involved in `IPinquiry` testing at IBMP. Activities in our groups are currently supported by the Centre National de la Recherche Scientifique (CNRS) and research grants from the French National Research Agency as part of the "Investments for the Future" program under the framework of the LABEX: ANR-10-LABX-0036_NETRNA and ANR-17-EURE-0023. The work was also supported by an IdEx grant from the University of Strasbourg to HZ and by an IdEx grant from the University of Strasbourg for the funding of a QExactive Plus mass spectrometer.

References

[1] Dunham WH, Mullin M, Gingras AC (2012) Affinity-purification coupled to mass spectrometry: Basic principles and strategies. Proteomics 12(10):1576–1590, https://doi.org/10.1002/pmic.201100523

[2] Smits AH, Vermeulen M (2016) Characterizing protein–protein interactions using mass spectrometry: challenges and opportunities. Trends in biotechnology 34(10):825–834, https://doi.org/10.1016/j.tibtech.2016.02.014

[3] Yugandhar K, Gupta S, Yu H (2019) Inferring protein-protein interaction networks from mass spectrometry-based proteomic approaches: a mini-review. Computational and Structural Biotechnology Journal 17:805–811, https://doi.org/10.1016/j.csbj.2019.05.007

[4] Bubis JA, Levitsky LI, Ivanov MV, Tarasova IA, Gorshkov MV (2017) Comparative evaluation of label-free quantification methods for shotgun proteomics. Rapid Communications in Mass Spectrometry 31(7):606–612, https://doi.org/10.1002/rcm.7829

[5] Robinson MD, McCarthy DJ, Smyth GK (2010) edger: a bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26(1):139–140, https://doi.org/10.1093/bioinformatics/btp616

[6] McCarthy DJ, Chen Y, Smyth GK (2012) Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. Nucleic acids research 40(10):4288–4297, https://doi.org/10.1093/nar/gks042

[7] Gregori J, Sanchez A, Villanueva J (2019) msmsTests: LC-MS/MS Differential Expression Tests. R package version 1.22.0

[8] Chicois C, Scheer H, Garcia S, Zuber H, Mutterer J, Chicher J, Hammann P, Gagliardi D, Garcia D (2018) The upf1 interactome reveals interaction networks between rna degradation and translation repression factors in arabidopsis. The Plant Journal 96(1):119–132, https://doi.org/10.1111/tpj.14022

[9] Lange H, Ndecky SY, Gomez-Diaz C, Pflieger D, Butel N, Zumsteg J, Kuhn L, Piermaria C, Chicher J, Christie M, et al. (2019) Rst1 and ripr connect the cy-

tosolic rna exosome to the ski complex in arabidopsis. Nature communications 10(1):1–12, https://doi.org/10.1038/s41467-019-11807-4

[10] Bouchoucha A, Waltz F, Bonnard G, Arrivé M, Hammann P, Kuhn L, Schelcher C, Zuber H, Gobert A, Giegé P (2019) Determination of protein-only rnase p interactome in arabidopsis mitochondria and chloroplasts identifies a complex between prorp1 and another nyn domain nuclease. The Plant Journal 100(3):549–561, https://doi.org/10.1111/tpj.14458

[11] Scheer H, de Almeida C, Ferrier E, Simonnot Q, Poirier L, Pflieger D, Sement FM, Koechler S, Piermaria C, Krawczyk P, et al. (2021) The tutase urt1 connects decapping activators and prevents the accumulation of excessively deadenylated mrnas to avoid sirna biogenesis. Nature communications 12(1):1–17, https://doi.org/10.1038/s41467-021-21382-2

[12] R Core Team (2017) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/

[13] Ritchie ME, Phipson B, Wu D, Hu Y, Law CW, Shi W, Smyth GK (2015) limma powers differential expression analyses for rna-sequencing and microarray studies. Nucleic acids research 43(7):e47–e47, https://doi.org/10.1101/2020.05.26.114322

[14] Giner G, Smyth GK (2016) statmod: probability calculations for the inverse gaussian distribution. R Journal 8(1):339–351, https://journal.r-project.org/archive/2016-1/giner-smyth.pdf

[15] Sievert C (2020) Interactive Web-Based Data Visualization with R, plotly, and shiny. Chapman and Hall/CRC, https://plotly-r.com

[16] Vaidyanathan R, Xie Y, Allaire J, Cheng J, Russell K (2019) htmlwidgets: HTML Widgets for R. R package version 1.5.1, https://CRAN.R-project.org/package=htmlwidgets

[17] Xie Y, Cheng J, Tan X (2020) DT: A Wrapper of the JavaScript Library 'DataTables'. R package version 0.13, https://CRAN.R-project.org/package=DT

[18] Wickham H (2016) ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, https://ggplot2.tidyverse.org

[19] Kolde R (2019) pheatmap: Pretty Heatmaps. R package version 1.0.12, https://CRAN.R-project.org/package=pheatmap

[20] Neuwirth E (2014) RColorBrewer: ColorBrewer Palettes. R package version 1.1-2, https://CRAN.R-project.org/package=RColorBrewer

[21] Durinck S, Moreau Y, Kasprzyk A, Davis S, De Moor B, Brazma A, Huber W (2005) Biomart and bioconductor: a powerful link between biological databases and microarray data analysis. Bioinformatics 21:3439–3440, https://doi.org/10.1093/bioinformatics/bti525

[22] Durinck S, Spellman PT, Birney E, Huber W (2009) Mapping identifiers for the integration of genomic datasets with the r/bioconductor package biomart. Nature Protocols 4:1184–1191, https://doi.org/10.1038/nprot.2009.97

[23] Dragulescu A, Arendt C (2020) xlsx: Read, Write, Format Excel 2007 and Excel 97/2000/XP/2003 Files. R package version 0.6.3, https://CRAN.R-project.org/package=xlsx

[24] Wickham H, Hester J, Chang W (2020) devtools: Tools to Make Developing R Packages Easier. R package version 2.3.1, https://CRAN.R-project.org/package=devtools

[25] Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for rna-seq data with deseq2. Genome Biology 15:550, DOI 10.1186/s13059-014-0550-8, https://doi.org/10.1186/s13059-014-0550-8

[26] Hubbard T, Barker D, Birney E, Cameron G, Chen Y, Clark L, Cox T, Cuff J, Curwen V, Down T, et al. (2002) The ensembl genome database project. Nucleic acids research 30(1):38–41, https://doi.org/10.1093/nar/30.1.38

[27] Gerace E, Moazed D (2015) Affinity purification of protein complexes using tap tags. In: Methods in enzymology, vol 559, Elsevier, pp 37–52, https://doi.org/10.1016/bs.mie.2014.11.007

[28] Blainey P, Krzywinski M, Altman N (2014) Replication: quality is often more important than quantity. Nature Methods 11(9):879–881, https://doi.org/10.1038/nmeth.3091

[29] Zhang Y, Wen Z, Washburn MP, Florens L (2009) Effect of dynamic exclusion duration on spectral count based quantitative proteomics. Analytical chemistry 81(15):6317–6326, https://doi.org/10.1021/ac9004887

[30] Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, et al. (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome biology 5(10):R80, https://doi.org/10.1186/gb-2004-5-10-r80