

Proteome wide screen for RNA-dependent Proteins in interphasic HeLa cells

Bolz, C., Bonsen, M., Pott, M., Simon, M.

2022-07-17

Introduction

RNA and proteins represent a symbiotic system. Proteins need RNA as a template for their biosynthesis as stated by Francis Crick in his “Central Dogma of Molecular Biology” (Crick, 1970). The opposite is also true, studies have shown that proteins need RNA for their catalytic activity, for instance in the RNA-induced silencing complex (Pratt and Macrae, 2009; Wilson and Doudna, 2013). Additionally, some RNA sequences depend on proteins for their synthesis and stability (Kishore *et al.*, 2010). A family of proteins which illustrates this symbiotic relationship are the RNA-binding proteins (RBPs). RBPs present a class of proteins whose interactome depends on RNA. They were shown to play a crucial role in RNA metabolism (Kishore *et al.*, 2010), cancer development (Wei *et al.*, 2022), and genetic disease (Gebauer *et al.*, 2021). Therefore, a deeper understanding of RNA binding and RNA-protein interaction strengthens our ability to adjust and manipulate the cellular mechanisms affected.

RBPs can be categorized into “true” RBPs (e.g. DICER, NPM3), those RNA-binding proteins which directly bind to RNA, and “RBP interacting proteins” which merely interact with “true” RBPs (eg. RBBP7). Furthermore, “true” RBPs can be subcategorized into “RNA-dependent”, meaning relying on RNA for their whole and correct function (eg. DICER) and “partially RNA dependent”, those that only require RNA for certain functions or transport (eg. NPM3) (see Fig. 1) (Caudron-Herger *et al.*, 2019; Corley *et al.*, 2020).

Figure 1. Categorization of RNA-binding proteins: RBPs can be categorized into “true” RBPs, those RNA-binding proteins which directly bind to RNA, “RBP interacting proteins” which merely interact with “true” RBPs. “True” RBPs are subcategorized into “RNA-dependent”, meaning relying on RNA for their whole and correct function and “partially RNA dependent”, those that only require RNA for certain functions or transport. DICER = Endoribonuclease Dicer; NPM3 = Nucleoplasm 3; RBBP7 = RNA-binding protein 7.

Several approaches to study RNA-protein interaction and to identify new RBPs were established such as RaPID and CLIP-Seq. These methods either analyze the interaction between the RNA of interest and additional proteins (RaPID) or the interaction between a protein of interest and the different RNA with which it is interacting (CLIP-Seq) (Qin *et al.*, 2021). Therefore, approaching the global study of RNA-binding protein, interaction networks (Sternburg and Karginov, 2020) has become a matter of interest. A method published by Caudron-Herger *et al.* enables analysis and quantification of whole cell interactomes. Furthermore, this method allows for identification of new RBPs through RNase treatment and density gradient ultracentrifugation. Subsequently, the resulting fraction shifts of proteins identified via mass spectrometry were analyzed using bioinformatic techniques (Caudron-Herger *et al.*, 2019, Caudron-Herger *et al.*, 2020).

In this project, we identified RBPs and possible RBP candidates using bioinformatics in R. Beyond that, we further identified contributing variables in our data differentiating RBPs without relying on our entire analysis protocol. We cross-referenced our results with known databases such as R-DeeP (<https://r-deep.dkfz.de/>; Caudron Herger *et al.*, 2019), UniProt (<https://www.uniprot.org/>) and RBP2GO (<https://rbp2go.dkfz.de/>; Caudron-Herger *et al.*, 2021). Our dataset focusing on interphase synchronized HeLa S3 cells was obtained by the method published by Caudron-Herger *et al.* cited above.

Methods

Generation of the dataset

Interphasic HeLa S3 cells were harvested and lyzed. The lyzate was split into two samples and one sample was treated with a RNase mixture, referred to as RNase (“RNA”). The other sample was left untreated, referred to as a control (“ctrl”). Both samples were loaded onto a 5 % to 25 % sucrose density gradient, grouped in 25 fractions and separated using ultracentrifugation. For either sample, technical triplicates were generated. Each fraction was then analyzed using mass spectrometry. Proteins were identified using UniProt. The data for the amount of protein per fraction, condition and replicate was collected in arbitrary units and stored in a .csv-file (for full protocol see: Caudron-Herger *et al.*, 2019).

Clean-Up and sorting of the dataset

Using the R package tidyverse, two separate data frames were generated containing either all untreated (“_ctrl”) or RNase treated (“_RNA”) replicates. Both data frames, as well as the raw data set, were screened for rows containing zeros only. Those zero rows were removed from the data frame and stored in a new data frame.

Normalization

To rule out batch to batch effects and technical error, the protein amount per replicate (“Rep”) and fraction (“Frac”) were additionally normalized with respect to each sample. The normalized results were then visualized by plotting the protein distribution in both samples.

Fraction-wise normalization

Normalization for each fraction was performed applying the following equation:

$$Protein(norm) = \frac{maxcolsums}{ColSumme} * Protein(before)$$

Protein(norm) describes the normalized amount of a single protein per fraction and replicate. *ColSumme* represents the total amount of protein per fraction and replicate. For each fraction, one maximum was chosen. *maxcolsums* is the selected maximum total protein amount per fraction and replicate. The parameter *Protein(before)* describes the arbitrary amount of a single protein per fraction and replicate prior to normalization.

Scaling the protein distribution per replicate and condition

For better comparison, the protein amount per fraction was converted to a relative percentage scale applying the following equation:

$$Protein(relative) = \frac{Protein(norm)}{RowSum} * 100$$

Protein(relative) describes the relative protein amount per fraction in relation to the total amount of the protein per replicate. *Protein(norm)* represents the normalized amount of a single protein per fraction and replicate. The total amount of a single protein per replicate is given by *RowSum*.

Create separate dataframes

Next, data frames for each individual fraction and condition were created, containing the triplicate values for each protein.

They follow the naming convention: 1. Fraction1_Ctrl for the triplicate values of the first fraction of the control group 2. Fraction_8_RNase for the triplicate values of the eighth fraction of the RNase treated sample

Calculation of means and standard deviation

The mean and standard deviation of the triplicates for each protein in each fraction were calculated using the built-in functions *mean()*-(\bar{x}) and *sd()*-*function*(σ) of R. Outliers were detected using $\bar{x} \pm 3\sigma$ as a cut-off. All values below and above the three sigma cut-off were replaced with NA and not considered in the following calculations.

Shapiro-Wilk Test

To test the normal distribution of the triplicate values, the Shapiro-Wilk test was chosen due to its high power in small populations compared to other tests. The test was performed using the built-in R function *shapiro_test()*.

Determination of Maxima

We determined the global maximum for the RNase-treated and untreated sample for all proteins. The protein content (y-value) was compared to the two neighbors right and left of the analyzed fraction (x-position). For fractions 1 and 25 only the neighbors right or left of the fraction could be compared due to border limitations. For fractions 2 and 24 only one neighbor could be compared left or right. Obtained values were stored in a sperate data frame.

Detection of Protein Shifts

In our analysis, we considered shifts in the global maximum comparing RNA-treated and untreated samples as a proxy for the presence of RNA. We required “shifting proteins” show both a significant x-shift and y-shift to improve precision.

x-shifts

First, we compared the x-position of the global maximum in both samples. To quantify the shift, we determined the difference in the fraction number of the control group and the RNase treated sample.

$$x - shift = |fracs_max_ctrl| - |fracs_max_rnase|$$

The variables *fracs_max_ctrl* and *fracs_max_rnase* describe the x-position of the global maxima for control and RNase treatment respectively. *x-shift* is the resulting value and used for Shift-direction determination.

The following convention regarding x-shifts was used:

1. Left-shift: Value < 1
2. Right-shift: Value > 1
3. No-Shift: Value = 0

y-shifts

The total value of the y-shift was calculated as the difference between the y-values of the global maxima for both conditions.

$$y - shift_total = |absolute_max_ctrl| - |absolute_max_rnase|$$

The variables *absolute_max_ctrl* and *absolute_max_rnase* describe the y-value of the global maxima for control and RNase treatment respectively. *y-shift_total* is the resulting value.

Statistical analysis

To identify significant y-shifts, we used statistical analysis to determine the difference in the relative protein amount (y-value) for each protein in the global maximum fraction of RNase-treated and untreated samples. All proteins with x-shift values $\geq |1|$ were considered.

F-Test

To determine whether our sample variances were comparable and therefore suitable for the two-tailed, unpaired t-test, a two-sided F-Test was performed on the triplicates of each protein and fraction using the built-in R function *var.test()*. The significance level (p-value) was set to $\alpha = 0.01$. The test was deemed positive if $p - value > 0.01$.

If *var.test()* was performed on all zero samples, NA/NaN was returned. Therefore, the F-test filtered out those samples that were not relevant for maxima analysis anyways. Since only the comparison of y-values at the x-positions of the global maxima mattered for our further analysis, proteins that failed the F-test (p-value < 0.01) at those x-positions were excluded. To quantify how many proteins failed F-test RNase- and ctrl-maximum spots were analyzed. To refer the F-test results back to the actual proteins, a more visual matrix was created labeled *p_value_matrix*.

Students T-Test

The two-tailed, unpaired t-test was used to identify proteins with significant y-value changes in global maxima x-positions after treatment. P-Values were calculated for each triplicate per protein per fraction using the built-in R function *t.test()*. Fractions with a global maximum were compared to the significance level $\alpha = 0.05$. The test was deemed positive if $p - value < 0.05$. Therefore, the change was considered significant.

Since the t-test was performed on y-values of x-positions of global maxima for both conditions separately, the returned results were either TRUE/TRUE, TRUE/FALSE, FALSE/TRUE or FALSE/FALSE for the ctrl and RNase maxima. Only proteins with positive t-tests (p-value < 0.01) for the global maximum in both conditions (TRUE/TRUE) were considered for further analysis.

Identification of potential RBPs by analysis of x- and y-shift

Potential RBPs were selected by filtering out proteins with significant y-shift following the t-test but no significant x-shift in the global maxima. Two different cut-off conditions were tested.

1. significant y-shift and x-shift < 1
2. significant y-shift and x-shift < 2

Proteins filtered out by this method were removed from the data frame and stored separately for further analysis.

Checking for false positive and false negative results

False-positive and false-negative proteins were determined by cross-referencing with two data sets which were provided by Maiwen Caudron-Herger at Prof. Dr. Sven Diederichs Lab (DFKZ) (unpublished Data). The data contains information on RBPs and non-RBPs previously identified by different researchers.

Identification of potential RBD Proteins

Potential RNA-binding dependent (RBD) protein candidates were identified by cross-referencing false-positive proteins with the data set provided by Maiwen Caudron-Herger at Prof. Dr. Sven Diederichs Lab (DFKZ) (unpublished Data).

k-means clustering

Using the libraries *corrplot*, *cluster* and *factoextra* k-means clustering was conducted using k-means algorithm. First we defined a function that applies the following equation to all proteins

$$Quotient = \frac{mean_RNAse(Protein)}{mean_ctrl(Protein)}$$

The resulting values were stored in a matrix labeled *q_Hela_mat*. *mean_RNAse(Protein)* describes the mean value for each protein and fraction in RNAse condition. *mena_ctrl(Protein)* describes the mean value for each protein and fraction in ctrl condition. Then we selected the maximum value (*maxi_RNAse*) for each protein and also calculated the overall quotient sum (*q_mat_sums*) for each protein and condition. These values were stored in the data frame *abs_mat_both_q*

Using the *factoextra* function *fviz_nbclust()* we created a silhouette plot to determine the optimal cluster number. With the built-in R function *k-means()* we clustered our proteins using the previous determined variables *q_mat_sums* and *maxi_RNAse* into three clusters.

Cluster A: Non-shifter

Cluster B: Potential shifter

Cluster C Sure-shifter

To check the quality of our clustering we created different silhouette plot using *factoextra*. To further analyse the results of our clustering we applied the functions created for cross-referencing our x-shift results. ## Linear regression analysis To train our regression model we selected 5000 random proteins from our raw data set. For these proteins we selected the values *maxi_RNAse* and *q_mat_sums* from the data frame *abs_mat_both_q*. We created a function to predict the x-shift based on those values. We plotted the maximum of quotients against the x-shifts.

To test our regression model the remaining 2081 proteins were treated

Results ## Clean-Up and normalization The initial data set consisted of 7086 proteins. After clean-up 7081 proteins remained for further analysis. Normalization was visualized for several reported sure-shifters and non-shifters (see Caudron-Herger *et al.*, 2019). The selected proteins are:

Sure-shifters: Sin3A_HUMAN, HDAC1_HUMAN, HNRPU_HUMAN, RFC2_HUMAN

Non-shifters: ASNS_HUMAN, MCM2_HUMAN, MCM3_HUMAN

Through normalization the arbitrary values were converted into relative percentage scale. Yet, the overall distribution of the protein amount per fraction remained comparable to the distribution prior to normalization. Therefore, the normalization worked for sure-shifters as well as non-shifters. For instance, figure 2 displays the normalization of a sure-shifter (RFC2_HUMAN) and a non-shifter (MCM2_HUMAN).

Figure 2: Comparing pre and post normalization (A): Comparison of the sure-shifter RFC2_HUMAN prior to and post normalization. (B): Comparison of the non-shifter MCM3_HUMAN prior to and post normalization. For both selected proteins, the overall trend of distribution remains comparable.

Calculation of mean, sd and Shapiro-Wilk Test

With the proposed 1σ rule we were able to detect not a single protein in which at least one replicate was not congruent with a normal distribution. To verify this result Shapiro-Wilk test was conducted. Test concluded that all values obtained for each protein were normally distributed for both conditions. Therefore they were suitable for statistical analysis with F-Test and two-tailed unpaired t-test.

Statistical analysis and Identification of potential RBPs by analysis of x- and y-shift

The obtained x- and y-shift as well as the fraction of the global maxima, shift direction and y-shift values were stored in a data frame labeled *abs_max_both*.

7081 proteins were tested with F-Test. Only the obtained p-values for fractions were a global maximum was detected were deemed relevant for identification of potential RBP candidates. Of 7081 proteins 1196 failed the test and were excluded from further analysis. 5885 remained. Quantification of FALSE Results in F-Test resulted that 10 % of all analyzed proteins fail F-Test. This means that either one or both obtained maxima for triplicate measurements vary in their variance in one treatment compared to the other. Because it seems like some FALSE F-test results originate from outlier values (see Discussion) and otherwise we would experience a great exclusion of data, significance level was set to $\alpha = 0.01$ instead of the previously usual $\alpha = 0.05$. With $\alpha = 0.01$ closer to 5 % of data will be excluded based on a FALSE F-test result. In this compromise some more likely faulty data will still be excluded while retaining as much data for further analysis as logically arguable.

Two-tailed unpaired t-test was applied to the remaining 5885 proteins per fraction ctrl vs RNase. P-Values were checked for significance at previous determined maxima positions. We identified 2929 potential shifters.

With the defined cut-off parameters significant y-shift and x-shift > 1 or significant y-shift and x-shift > 2 these potential shifters were further analysed.

The cut-off >1 for the x-shift allowed us to identify 513 true RBP with 120 false-positives out of 636 total. Of the 120 false-positives six were identified with cross-referencing as RBD proteins. In comparison, the cut-off >2 for the x-shift allowed us to identify 398 true RBP with 91 false-positives out of 489 total. Of the 91 false-positives six were identified with cross-referencing as RBD proteins. The six identified proteins were identical for both cut-off parameters (see Tab.1, data frames *RBD_true_port_df1* and *RBD_true_prot_df2*).

In both cases, about 80% of identified RBP were correct when cross-referenced. Since the >1 x-shift cut-off did not worsen this quote but instead delivered more results overall (636 compared to 489), >1 x-shift cut-off is used for our analysis.

Table.1: Amount of identified RBPs: True RBP quantification results after cross-referencing obtained proteins with the additional data provided by Caudron-Herger. TP = True positive, TN = True negative,

FP = False positive, FN = False negative.

Criteria	x-Shift > 1	x-shift > 2
Total identified	636	489
TP	513	398
FP	120	91
RBD	6	6

K-Means Clustering and Regression model

Discussion

Lorem ipsum set dolor