Supplementary Information: *Rtpca: an R package for differential thermal coaggregation analysis*

***Nils Kurzawa*[1], *André Mateus*[1], *and Mikhail M. Savitski*[1]**

[1]European Molecular Biology Laboratory (EMBL), Genome Biology Unit

07 April, 2020

Package

Rtpca 0.0.99

Contents

1 Introduction

Thermal proteome profiling (TPP) (Mateus et al., 2020; Savitski et al., 2014) is a mass spectrometry-based, proteome-wide implemention of the cellular thermal shift assay (Molina et al., 2013). It was originally developed to study drug-(off-)target engagement. However, it was realized that profiles of interacting protein pairs appeared more similar than by chance (Tan et al., 2018, Becher et al. (2018)) which was coined as 'thermal proximity co-aggregation' (TPCA) (Tan et al., 2018). The R package `Rtpca` enables analysis of TPP datasets using the TPCA concept for studying protein-protein interactions and protein complexes and also allows to test for differential protein-protein interactions across different conditions.
Here, we exemplify the analysis based on a dataset by Becher et al. (2018) which provides temperature range TPP (TPP-TR) experiments for synchronized HeLa cells in G1/S cell cycle stage versus M phase.
Note: The paper by Becher et al. (2018) also includes 2D-TPP (**???**) data which is in general more sensitive to changes in protein abundance or stability. This data can also be informative on dynamics of protein-protein interactions based on correlations analysis of 2D-TPP profiles of annotated interactors. However, the advantage of TPP-TR data is that one can test for coaggregation which, if significant, is directly indicative of protein-protein interaction or complex assembly.

2 Step-by-step walk through the data analysis

First, we need to load the required libraries (these need to be installed as specified in the comments):

```
library(dplyr) # install.packages("dplyr")
library(readxl) # install.packages("readxl")
library(Rtpca) # require(devtools); devtools::install_github("nkurzaw/Rtpca")
library(ggplot2) # install.packages("ggplot2")
```

Then, we download the supplementary data from Tan et al. which contains the TPP data which we'll be using:

```
if(!file.exists("1-s2.0-S0092867418303854-mmc4.xlsx")){
    download.file(
        url = "https://ars.els-cdn.com/content/image/1-s2.0-S0092867418303854-mmc4.xlsx",
        destfile = "1-s2.0-S0092867418303854-mmc4.xlsx")
}
```

Next, we read in the annotation information of the supplementary table as a data frame

```
supp_tab_becher_s4 <- read_xlsx("1-s2.0-S0092867418303854-mmc4.xlsx",
                                sheet = "TableS4_TPP-TR")

temperature_anno <-
    as.numeric(
        gsub("T", "", gsub("_.+", "", colnames(
            supp_tab_becher_s4 %>%
                dplyr::select(matches("mean\\.fc"))))))
```

We then extract the data for the

```
g1s_df <- supp_tab_becher_s4 %>%
    filter(cell.cycle == "G1_S") %>%
    dplyr::select(
        gene_name,
        replicates = found.in.reps,
        max_qupm = max.qupm,
        min_qupm = min.qupm,
        matches("mean\\.fc")) %>%
    filter(min_qupm > 3,
           replicates == 3)
```

```
g1s_mat <- as.matrix(
    g1s_df %>% dplyr::select(dplyr::matches("mean\\.fc"))
    )
rownames(g1s_mat) <- g1s_df$gene_name
attributes(g1s_mat)$temperature <- temperature_anno
```

```
m_df <- supp_tab_becher_s4 %>%
    filter(cell.cycle == "M") %>%
    dplyr::select(
        gene_name,
        replicates = found.in.reps,
        max_qupm = max.qupm,
        min_qupm = min.qupm,
        matches("mean\\.fc")) %>%
    filter(min_qupm > 3,
           replicates == 3)
```

```
m_mat <- as.matrix(
    m_df %>% dplyr::select(dplyr::matches("mean\\.fc"))
    )
rownames(m_mat) <- m_df$gene_name
attributes(m_mat)$temperature <- temperature_anno
```

First, we load an annotation of mammalian complexes by Ori et al. (2016), which comes with the `Rtpca` package:

```
data("ori_et_al_complexes_df")
```

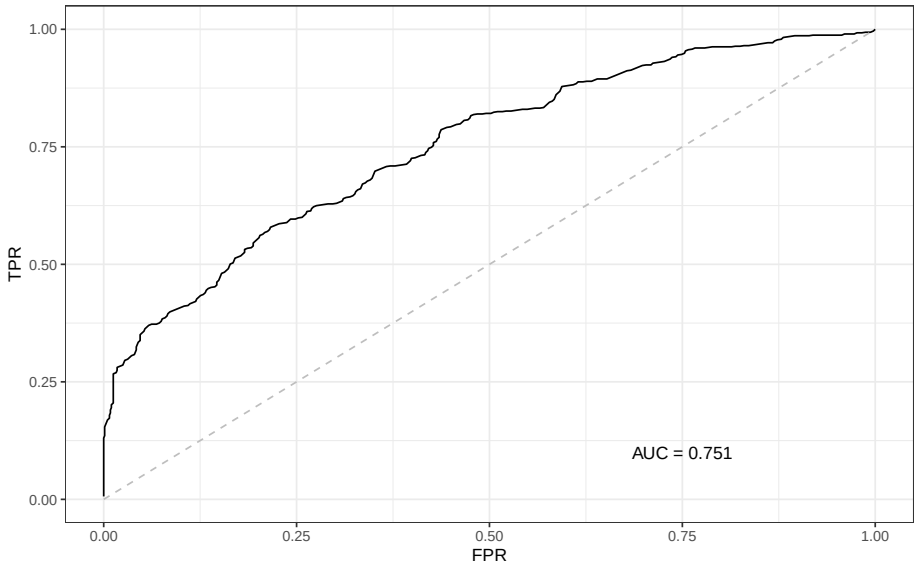
Then, we perform a TPCA anylsis only in the G1/S condition:

```
G1S_TPCA <- runTPCA(
    objList = list(g1s_mat),
    complexAnno = ori_et_al_complexes_df)
## Checking input arguments.
##
## Creating distance matrices.
##
## Testing for complex co-aggregation.
##
```

```
## Performning Complex ROC analysis.
```

We can plot the ROC curve for how predictive our data is on recovering protein complexes by evoking:

```
plotComplexRoc(G1S_TPCA, computeAUC = TRUE)
```



And we can inspect significantly co-melting protein complexes, like this:

```
G1S_TPCA@tpcaResultTable %>% filter(p_adj < 0.1)
## # A tibble: 46 x 5
##    complex_name                    count mean_dist  p_value    p_adj
##    <chr>                           <int>     <dbl>    <dbl>    <dbl>
##  1 26S Proteasome                     33    0.441  1.73e- 2 6.79e- 2
##  2 Nuclear pore complex (NPC)         24    0.341  1.00e- 4 1.34e- 3
##  3 BAF complex                         7    0.214  8.00e- 4 7.16e- 3
##  4 Spliceosome-U2                      9    0.333  1.48e- 2 6.44e- 2
##  5 Anaphase promoting complex (APC)    4    0.0948 2.22e-16 3.25e-15
##  6 multi-tRNAsynthase complex         10    0.123  2.22e-16 3.25e-15
##  7 RNA polymerase III core complex     3    0.189  2.39e- 2 8.75e- 2
##  8 RNA polymerase II core complex      4    0.190  6.90e- 3 4.11e- 2
##  9 COP9 signalosome                    8    0.236  8.00e- 4 7.16e- 3
## 10 MCM complex                         7    0.190  3.00e- 4 3.45e- 3
## # ... with 36 more rows
```

```
g1s_significant_complex_comelting <-
    filter(G1S_TPCA@tpcaResultTable,p_adj < 0.1)$complex_name
```

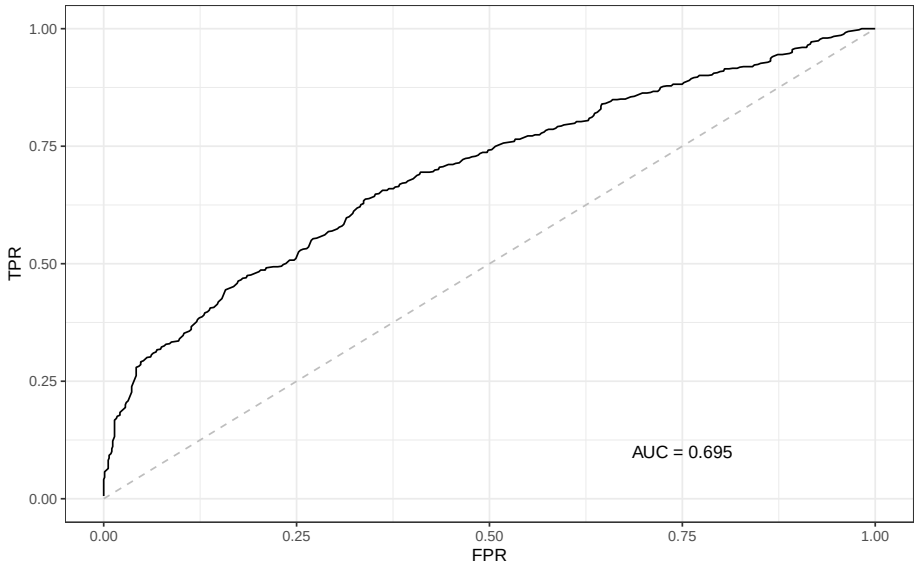
Next, we perform the same analysis for only the M phase condition:

```
M_TPCA <- runTPCA(
    objList = list(m_mat),
    complexAnno = ori_et_al_complexes_df)
## Checking input arguments.
```

```
##
## Creating distance matrices.
##
## Testing for complex co-aggregation.
##
## Performning Complex ROC analysis.
```

We can see that the predictive performance of this dataset for protein complexes is not quite as good as for the G1/S one:

```
plotComplexRoc(M_TPCA, computeAUC = TRUE)
```




```
M_TPCA@tpcaResultTable %>% filter(p_adj < 0.1)
## # A tibble: 56 x 5
##    complex_name                       count mean_dist  p_value    p_adj
##    <chr>                              <int>     <dbl>    <dbl>    <dbl>
##  1 Nuclear pore complex (NPC)            25    0.332  5.00e- 4 6.58e- 3
##  2 BAF complex                            9    0.140  1.00e- 4 1.90e- 3
##  3 Integrator                             7    0.254  9.90e- 3 4.84e- 2
##  4 NuRD complex                           9    0.254  3.60e- 3 2.05e- 2
##  5 Anaphase promoting complex (APC)       4    0.166  1.13e- 2 5.00e- 2
##  6 Cohesin complex                        7    0.245  7.80e- 3 3.92e- 2
##  7 Transcription-export (TREX) complex    8    0.312  2.78e- 2 8.49e- 2
##  8 multi-tRNAsynthase complex            10    0.0811 2.22e-16 5.42e-15
##  9 RANBP9-containing complex              3    0.0898 2.50e- 3 1.77e- 2
## 10 RNA polymerase II core complex         4    0.174  1.44e- 2 5.35e- 2
## # ... with 46 more rows
```

Based on the protein complexes which we find significantly assembled in either condition, we will select the protein-protein interactions to test for in a differential TPCA:

```
m_significant_complex_comelting <-
    filter(M_TPCA@tpcaResultTable,p_adj < 0.1)$complex_name
```

```
all_significant_complex_comelting <-
    unique(c(g1s_significant_complex_comelting,
             m_significant_complex_comelting))
```

We load the annotation of protein-protein interactions within complexes that is composed of PPIs from StringDb (Szklarczyk et al., 2019) and the complex annotation by Ori et al. (2016) and filter it for protein complexes that we have seen to coaggregate in the analysis above.

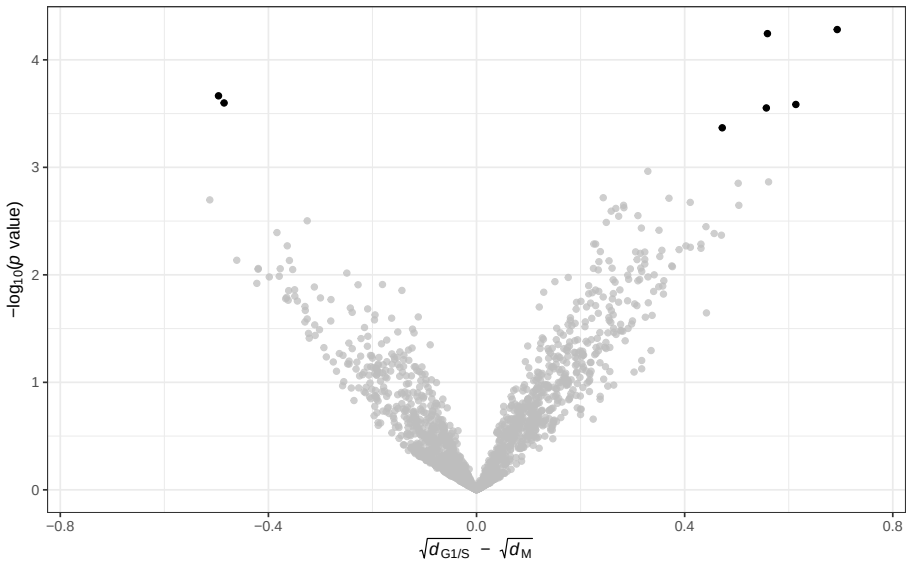
```
data("ori_et_al_complex_ppis")

filtered_complex_ppis <- ori_et_al_complex_ppis %>%
    filter(complex_name %in% all_significant_complex_comelting)
```

We now run the differential TPCA by evoking“”

```
set.seed(123)
M_vs_G1S_diff_TPCA <- runDiffTPCA(
    objList = list(g1s_mat),
    contrastList = list(m_mat),
    ctrlCondName = "G1/S",
    contrastCondName = "M",
    ppiAnno = filtered_complex_ppis,
    n = 10^6
)
## Checking input arguments.
## Creating distance matrices.
## Comparing annotated protein-pairs across conditions.
## Comparing random protein-pairs across conditions.
## Generating result table.
```

We can now plot the result in form of a volcano plot:

```
plotDiffTpcaVolcano(M_vs_G1S_diff_TPCA,
                    setXLim = TRUE,
                    xlimit = c(-0.75, 0.75))
```

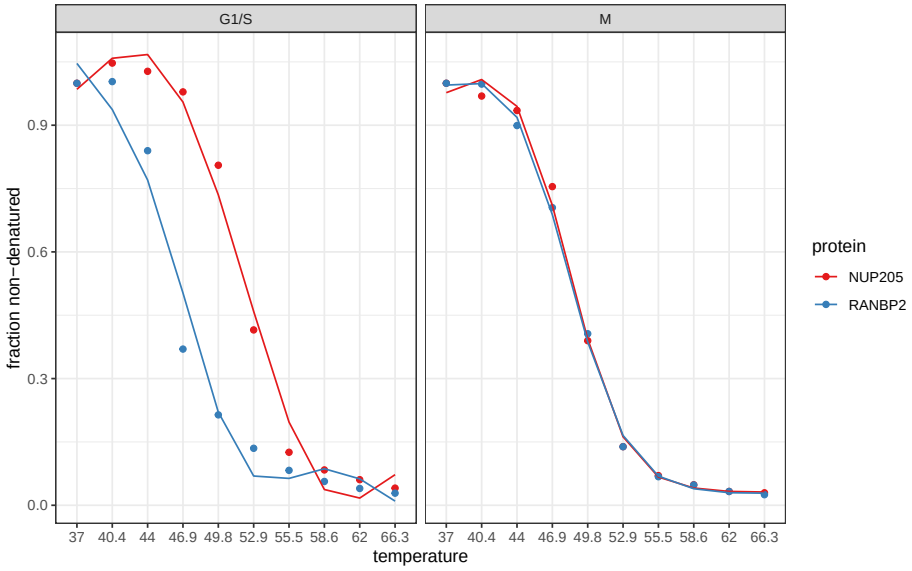


We can now inspect the significant results:

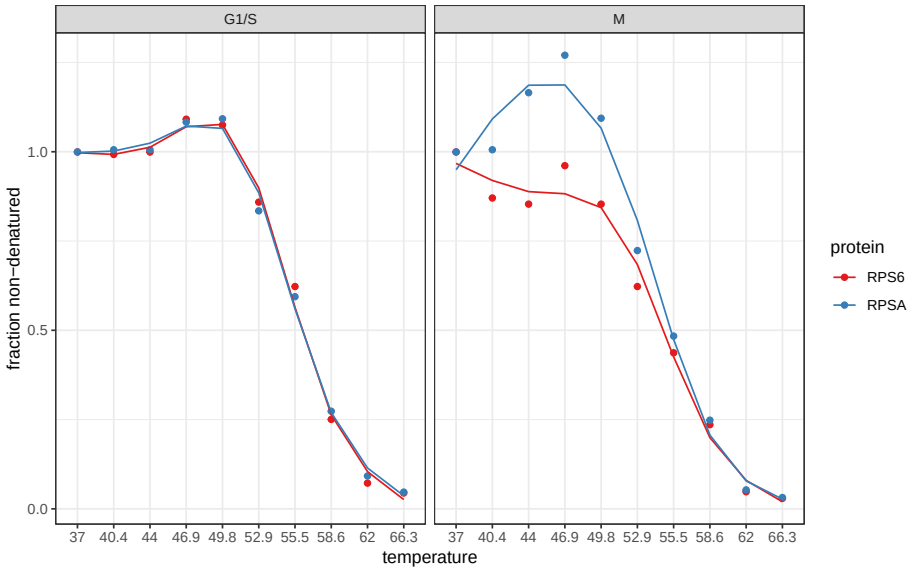
```
M_vs_G1S_diff_TPCA@diffTpcaResultTable %>%
    dplyr::select(pair, rssC1_rssC2, p_value, p_adj) %>%
    arrange(p_value)
## # A tibble: 1,549 x 4
##    pair          rssC1_rssC2   p_value  p_adj
##    <chr>               <dbl>     <dbl>  <dbl>
##  1 NUP205:RANBP2      0.834  0.0000523 0.0442
##  2 NUP88:RANBP2       0.357  0.0000570 0.0442
##  3 RPS6:RPSA         -0.279  0.000216  0.0724
##  4 RPS23:RPSA        -0.263  0.000252  0.0724
##  5 NUP93:RANBP2       0.679  0.000260  0.0724
##  6 NUP188:RANBP2      0.470  0.000280  0.0724
##  7 AAAS:TPR           0.260  0.000429  0.0949
##  8 PSMB1:PSMB4        0.0786 0.00109   0.207
##  9 NUP54:RANBP2       0.710  0.00137   0.207
## 10 NUP107:RANBP2      0.461  0.00141   0.207
## # ... with 1,539 more rows
```

To validate significant PPIs we can inspect their melting curves:

```
plotPPiProfiles(M_vs_G1S_diff_TPCA, pair = c("NUP205", "RANBP2"))
```




```
plotPPiProfiles(M_vs_G1S_diff_TPCA, pair = c("RPS6", "RPSA"))
```




```
sessionInfo()
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
```

```
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] ggplot2_3.2.1       Rtpca_0.0.99        tidyr_1.0.0
## [4] Biobase_2.44.0      BiocGenerics_0.30.0 readxl_1.3.1
## [7] dplyr_0.8.3         BiocStyle_2.12.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.2         RColorBrewer_1.1-2 plyr_1.8.4
##  [4] cellranger_1.1.0   pillar_1.4.2       compiler_3.6.1
##  [7] BiocManager_1.30.9 tools_3.6.1        zeallot_0.1.0
## [10] digest_0.6.22      evaluate_0.14      tibble_2.1.3
## [13] lifecycle_0.1.0    gtable_0.3.0       pkgconfig_2.0.3
## [16] rlang_0.4.1        cli_1.1.0          yaml_2.2.0
## [19] xfun_0.10          withr_2.1.2        stringr_1.4.0
## [22] knitr_1.25         pROC_1.15.3        vctrs_0.2.0
## [25] grid_3.6.1         tidyselect_0.2.5   glue_1.3.1
## [28] R6_2.4.0           fansi_0.4.0        fdrtool_1.2.15
## [31] rmarkdown_1.16     bookdown_0.14      farver_2.0.3
## [34] purrr_0.3.3        magrittr_1.5       ellipsis_0.3.0
## [37] splines_3.6.1      backports_1.1.5    scales_1.1.0
## [40] htmltools_0.4.0    assertthat_0.2.1   colorspace_1.4-1
## [43] labeling_0.3       utf8_1.1.4         stringi_1.4.3
## [46] lazyeval_0.2.2     munsell_0.5.0      crayon_1.3.4
```

References

Becher, I., Andrés-Pons, A., Romanov, N., Stein, F., Schramm, M., Baudin, F., Helm, D., Kurzawa, N., Mateus, A., Mackmull, M.-T., et al. (2018). Pervasive Protein Thermal Stability Variation during the Cell Cycle. Cell 1–13.

Mateus, A., Kurzawa, N., Becher, I., Sridharan, S., Helm, D., Stein, F., Typas, A., and Savitski, M.M. (2020). Thermal proteome profiling for interrogating protein interactions. Molecular Systems Biology 1–11.

Molina, D.M., Jafari, R., Ignatushchenko, M., Seki, T., Larsson, E.A., Dan, C., Sreekumar, L., Cao, Y., and Nordlund, P. (2013). Monitoring Drug Target Engagement in Cells and Tissues Using the Cellular Thermal Shift Assay. Science *341*, 84–88.

Ori, A., Iskar, M., Buczak, K., Kastritis, P., Parca, L., Andrés-pons, A., Singer, S., Bork, P., and Beck, M. (2016). Spatiotemporal variation of mammalian protein complex stoichiometries. Genome Biology 1–15.

Savitski, M.M., Reinhard, F.B.M., Franken, H., Werner, T., Savitski, M.F., Eberhard, D., Martinez Molina, D., Jafari, R., Dovega, R.B., Klaeger, S., et al. (2014). Tracking cancer drugs in living cells by thermal profiling of the proteome. Science *346*, 1255784.

Szklarczyk, D., Gable, A.L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., Simonovic, M., Doncheva, N.T., Morris, J.H., Bork, P., et al. (2019). STRING v11: Protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Research *47*, D607–D613.

Tan, C.S.H., Go, K.D., Bisteau, X., Dai, L., Yong, C.H., Prabhu, N., Ozturk, M.B., Lim, Y.T., Sreekumar, L., Lengqvist, J., et al. (2018). Thermal proximity coaggregation for system-wide profiling of protein complex dynamics in cells. Science *0346*, 1–12.