Introduction to the *TPP* package for analyzing Thermal Proteome Profiling data: 2D-TPP experiments

Dorothee Childs, Nils Kurzawa
European Molecular Biology Laboratory (EMBL),
Heidelberg, Germany
dorothee.childs@embl.de

TPP version 3.0.3 (Last revision 2016-10-25)

Abstract

Thermal Proteome Profiling (TPP) combines the cellular thermal shift assay concept [1] with mass spectrometry based proteome-wide protein quantitation [2]. Thereby, drug-target interactions can be inferred from changes in the thermal stability of a protein upon drug binding, or upon downstream cellular regulatory events, in an unbiased manner.

The package *TPP* facilitates this process by providing exectuable workflows that conduct all necessary data analysis steps. Recent advances in the field have lead to the development of so called 2D Thermal Proteome Profiling (2D-TPP) experiments [3]. Recent advances in the field have lead to the development of so called 2D Thermal Proteome Profiling (2D-TPP) experiments [3]. Similar as for the TPP-TR and the TPP-CCR analysis, the function analyze2DTPP executes the whole workflow from data import through normalization and curve fitting to statistical analysis. Nevertheless, all of these steps can also be invoked separately by the user. The corresponding functions can be recognized by their suffix tpp2d.

Here, we first show how to start the whole analysis using analyze2DTPP. Afterwards, we demonstrate how to carry out single steps individually.

For details about the analysis of 1D TR- or CCR experiments [2, 4], please refer to the vignette `TPP_introduction_1D`.

Contents

1 Installation

To install the package, type the following commands into the *R* console

```
source("http://bioconductor.org/biocLite.R")
biocLite("TPP")
```

The installed package can be loaded by

```
library("TPP")
```

1.1 Special note for Windows users

The *TPP* package uses the *openxlsx* package to produce Excel output [5]. *openxlsx* requires a zip application to be installed on your system and to be included in the path. On Windows, such a zip application ist not installed by

default, but is available, for example, via Rtools. Without the zip application, you can still use the 'TPP' package and access its results via the dataframes produced by the main functions.

2 Analyzing 2D-TPP experiments

2.1 Overview

Before you can start your analysis, you need to specify information about your experiments:

The mandatory information comprises a unique experiment name, as well as the isobaric labels and corresponding temperature values for each experiment. The package retrieves this information from a configuration table that you need to specify before starting the analysis. This table can either be a data frame that you define in your R session, or a spreadsheet in .xlsx or .csv format. In a similar manner, the measurements themselves can either be provided as a list of data frames, or imported directlyfrom files during runtime.

We demonstrate the functionality of the package using the dataset Panobinostat_2DTPP_smallExampleData. It contains an illustrative subset of a larger dataset which was obtained by 2D-TPP experiments on HepG2 cells treated with the histone deacetylase (HDAC) inhibitor panobinostat in the treatment groups and with vehicle in the control groups. The experiments were performed for different temperatures. The raw MS data were processed with the Python package isobarQuant, which provides protein fold changes relative to the protein abundance at the lowest temperature as input for the TPP package [3].

2.2 Performing the analysis

Fist of all, we load an example data set:

```
data("panob2D_isobQuant_example")
```

Using this command we load two objects:

1. `Panobinostat_2DTPP_smallExampleData`: a list of data frames that contain the measurements to be analyzed,
2. `hdac2D_config`: a configuration table with details about each experiment.

```
config_tpp2d <- panobinostat_2DTPP_config
data_tpp2d <- panobinostat_2DTPP_data

config_tpp2d
```

```
##        Compound  Experiment Temperature 126 127L  127H 128L 128H 129L 129H  130L 130H
## 1  Panobinostat Experiment1        42.0   5    1 0.143 0.02    0    -    -     -    -
## 2  Panobinostat Experiment1        44.1   -    -     -    -    -    5    1 0.143 0.02
## 3  Panobinostat Experiment2        46.2   5    1 0.143 0.02    0    -    -     -    -
## 4  Panobinostat Experiment2        48.1   -    -     -    -    -    5    1 0.143 0.02
## 5  Panobinostat Experiment3        50.4   5    1 0.143 0.02    0    -    -     -    -
## 6  Panobinostat Experiment3        51.9   -    -     -    -    -    5    1 0.143 0.02
## 7  Panobinostat Experiment4        54.0   5    1 0.143 0.02    0    -    -     -    -
## 8  Panobinostat Experiment4        56.1   -    -     -    -    -    5    1 0.143 0.02
## 9  Panobinostat Experiment5        58.2   5    1 0.143 0.02    0    -    -     -    -
## 10 Panobinostat Experiment5        60.1   -    -     -    -    -    5    1 0.143 0.02
## 11 Panobinostat Experiment6        62.4   5    1 0.143 0.02    0    -    -     -    -
## 12 Panobinostat Experiment6        63.9   -    -     -    -    -    5    1 0.143 0.02
##    131L RefCol Path
## 1     -   128H
## 2     0   131L
## 3     -   128H
## 4     0   131L
## 5     -   128H
## 6     0   131L
## 7     -   128H
## 8     0   131L
## 9     -   128H
## 10    0   131L
```

```
## 11    -   128H
## 12    0   131L
```

```
data_tpp2d %>% str(1)
```

```
## List of 6
##  $ Experiment1:'data.frame': 484 obs. of  13 variables:
##  $ Experiment2:'data.frame': 478 obs. of  13 variables:
##  $ Experiment3:'data.frame': 448 obs. of  13 variables:
##  $ Experiment4:'data.frame': 372 obs. of  13 variables:
##  $ Experiment5:'data.frame': 306 obs. of  13 variables:
##  $ Experiment6:'data.frame': 261 obs. of  13 variables:
```

The data object `Panobinostat_2DTPP_smallExampleData` is organized as a list of data frames which contain the experimental raw data of an 2D-TPP experiment. The names of the list elements correspond to the different multiplexed experiments. Each experimental dataset constains the following columns:

```
data_tpp2d$Experiment1 %>% colnames
```

```
##  [1] "gene_name"       "qupm"            "qssm"            "signal_sum_126"
##  [5] "signal_sum_127L" "signal_sum_127H" "signal_sum_128L" "signal_sum_128H"
##  [9] "signal_sum_129L" "signal_sum_129H" "signal_sum_130L" "signal_sum_130H"
## [13] "signal_sum_131L"
```

In ordern to perform the complete workflow we can now simply use:

```
tpp2dResults <- analyze2DTPP(configFile = config_tpp2d,
                             data = data_tpp2d,
                             fcStr = NULL,
                             methods = "doseResponse",
                             nCores = 2)


tpp2dResults %>% mutate_if(is.character, factor) %>% summary
```

```
##                   Protein_ID   norm_rel_fc_0_unmodified norm_rel_fc_0.02_unmodified
##  Experiment1_42_A2M    :   1   Min.   :1                Min.   :0.1767
##  Experiment1_42_ABHD10 :   1   1st Qu.:1                1st Qu.:0.9192
##  Experiment1_42_ABHD14B:   1   Median :1                Median :1.0000
##  Experiment1_42_ACAA1  :   1   Mean   :1                Mean   :1.0035
##  Experiment1_42_ACBD5  :   1   3rd Qu.:1                3rd Qu.:1.0727
##  Experiment1_42_ACO1   :   1   Max.   :1                Max.   :4.6565
##  (Other)               :4650
##  norm_rel_fc_0.143_unmodified norm_rel_fc_1_unmodified norm_rel_fc_5_unmodified
##  Min.   :0.2612               Min.   : 0.2422          Min.   : 0.2512
##  1st Qu.:0.9364               1st Qu.: 0.9344          1st Qu.: 0.9337
##  Median :1.0000               Median : 1.0000          Median : 1.0000
##  Mean   :1.0105               Mean   : 1.0163          Mean   : 1.0259
##  3rd Qu.:1.0632               3rd Qu.: 1.0654          3rd Qu.: 1.0589
##  Max.   :5.8855               Max.   :10.0240          Max.   :17.0405
##
##  norm_rel_fc_0_normalized_to_lowest_conc norm_rel_fc_0.02_normalized_to_lowest_conc
##  Min.   :1                               Min.   :0.1767
##  1st Qu.:1                               1st Qu.:0.9192
##  Median :1                               Median :1.0000
##  Mean   :1                               Mean   :1.0035
##  3rd Qu.:1                               3rd Qu.:1.0727
##  Max.   :1                               Max.   :4.6565
##
##  norm_rel_fc_0.143_normalized_to_lowest_conc norm_rel_fc_1_normalized_to_lowest_conc
##  Min.   :0.2612                              Min.   : 0.2422
##  1st Qu.:0.9364                              1st Qu.: 0.9344
```

```
##  Median :1.0000                                  Median : 1.0000
##  Mean   :1.0105                                  Mean   : 1.0163
##  3rd Qu.:1.0632                                  3rd Qu.: 1.0654
##  Max.   :5.8855                                  Max.   :10.0240
##
##  norm_rel_fc_5_normalized_to_lowest_conc norm_rel_fc_0_transformed
##  Min.   : 0.2512                         Min.   :0.000
##  1st Qu.: 0.9337                         1st Qu.:0.000
##  Median : 1.0000                         Median :1.000
##  Mean   : 1.0259                         Mean   :0.621
##  3rd Qu.: 1.0589                         3rd Qu.:1.000
##  Max.   :17.0405                         Max.   :1.000
##                                          NA's   :4421
##  norm_rel_fc_0.02_transformed norm_rel_fc_0.143_transformed norm_rel_fc_1_transformed
##  Min.   :-0.884               Min.   :-1.201               Min.   :-0.961
##  1st Qu.:-0.154               1st Qu.: 0.086               1st Qu.: 0.095
##  Median : 0.297               Median : 0.376               Median : 0.313
##  Mean   : 0.302               Mean   : 0.400               Mean   : 0.400
##  3rd Qu.: 0.614               3rd Qu.: 0.662               3rd Qu.: 0.652
##  Max.   : 2.542               Max.   : 3.294               Max.   : 2.925
##  NA's   :4421                 NA's   :4421                 NA's   :4421
##  norm_rel_fc_5_transformed     pEC50           slope              R_sq          plot
##  Min.   :0.000             Min.   :5.728   Min.   :-50.000   Min.   :-0.068   NA's:4656
##  1st Qu.:0.000             1st Qu.:6.696   1st Qu.:-10.804   1st Qu.: 0.545
##  Median :0.000             Median :7.778   Median : -1.000   Median : 0.723
##  Mean   :0.379             Mean   :7.346   Mean   : -8.302   Mean   : 0.675
##  3rd Qu.:1.000             3rd Qu.:8.126   3rd Qu.:  1.159   3rd Qu.: 0.881
##  Max.   :1.000             Max.   :8.126   Max.   : 50.000   Max.   : 1.000
##  NA's   :4421              NA's   :4421    NA's   :4421      NA's   :4421
##      compound_effect meets_FC_requirement passed_filter   pEC50_outside_conc_range
##  destabilized: 146   Mode :logical        Mode :logical   Mode :logical
##  stabilized  :  89   FALSE:4537           FALSE:4601      FALSE:111
##  NA's        :4421   TRUE :119            TRUE :55        TRUE :124
##                      NA's :0              NA's :0         NA's :4421
##
##
##
##  model_converged       pEC50_quality_check sufficient_data_for_fit protein_identified_in
##  Mode:logical    5.72818301656452:  12     Mode:logical            Mode:logical
##  TRUE:235        6.07074587494624:   6     TRUE:235                TRUE:4656
##  NA's:4421       7.44099730847312:   6     NA's:4421               NA's:0
##                  6.75587159170968:   2
##                  5.83469502048232:   1
##                  (Other)         :  84
##                  NA's            :4545
##    gene_name          qupm             qssm         signal_sum_5        signal_sum_1
##  A2M    :  12   Min.   : 1.000   Min.   :  1.00   Min.   :2.063e+05   Min.   :3.819e+05
##  ABHD10 :  12   1st Qu.: 3.000   1st Qu.:  5.00   1st Qu.:7.696e+07   1st Qu.:7.604e+07
##  ACAA1  :  12   Median : 7.000   Median : 11.00   Median :2.511e+08   Median :2.512e+08
##  ACO1   :  12   Mean   : 9.149   Mean   : 19.57   Mean   :7.182e+08   Mean   :7.542e+08
##  ACO2   :  12   3rd Qu.:12.000   3rd Qu.: 23.00   3rd Qu.:7.382e+08   3rd Qu.:7.682e+08
##  ACTC1  :  12   Max.   :87.000   Max.   :263.00   Max.   :2.125e+10   Max.   :2.138e+10
##  (Other):4584
##  signal_sum_0.143    signal_sum_0.02      signal_sum_0        temperature
##  Min.   :3.579e+05   Min.   :4.335e+05   Min.   :2.925e+05   Min.   :42.0
##  1st Qu.:8.079e+07   1st Qu.:8.401e+07   1st Qu.:7.345e+07   1st Qu.:46.2
##  Median :2.591e+08   Median :2.739e+08   Median :2.574e+08   Median :50.4
##  Mean   :7.554e+08   Mean   :8.100e+08   Mean   :8.599e+08   Mean   :51.6
```

```
##  3rd Qu.:7.857e+08   3rd Qu.:8.331e+08   3rd Qu.:8.554e+08   3rd Qu.:56.1
##  Max.   :1.924e+10   Max.   :2.249e+10   Max.   :2.644e+10   Max.   :63.9
##
##        experiment     rel_fc_5          rel_fc_1       rel_fc_0.143     rel_fc_0.02
##  Experiment1:968   Min.   : 0.3487   Min.   :0.2985   Min.   :0.3887   Min.   : 0.1882
##  Experiment2:950   1st Qu.: 0.7894   1st Qu.:0.8231   1st Qu.:0.8156   1st Qu.: 0.8413
##  Experiment3:894   Median : 0.8964   Median :0.9197   Median :0.9415   Median : 0.9601
##  Experiment4:738   Mean   : 0.9935   Mean   :0.9753   Mean   :1.0187   Mean   : 1.0974
##  Experiment5:600   3rd Qu.: 1.0878   3rd Qu.:1.0588   3rd Qu.:1.1447   3rd Qu.: 1.2027
##  Experiment6:506   Max.   :17.1835   Max.   :8.6463   Max.   :6.2354   Max.   :10.0917
##
##     rel_fc_0
##  Min.   :1
##  1st Qu.:1
##  Median :1
##  Mean   :1
##  3rd Qu.:1
##  Max.   :1
##
```

Moreover, we can also invoke the single functions of the workflow manually. Therefore, we start with importing the data. Using the import function the data is subsequently imported and stored in a single dataframe containing all the required data columns and those that the user likes to take along through the analysis to be displayed together with the results of this workflow.

```
data2d <- tpp2dImport(configTable = config_tpp2d,
                      data = data_tpp2d,
                      fcStr = NULL)

head(data2d)
```

```
##   gene_name qupm qssm signal_sum_5 signal_sum_1 signal_sum_0.143 signal_sum_0.02
## 1     CCND1    3    4    204841190    232467960        248774392       316622154
## 2  C17ORF39    1    1     65819416     65633403         99635379       112822532
## 3    NECAP1    3    3     98127667    119382560        113228677       217363144
## 4     EEF1G   17   59   3088494716   3716161024       4008219610      4973078201
## 5     RIPK2    5    5    259734512    303419382        323066842       355720486
## 6     EIF4H   21   45   1309348011   1469321178       1348496831      1630178705
##   signal_sum_0 temperature  experiment              unique_ID
## 1    370562621          42 Experiment1    Experiment1_42_CCND1
## 2    115419115          42 Experiment1 Experiment1_42_C17ORF39
## 3    159124932          42 Experiment1   Experiment1_42_NECAP1
## 4   5214069781          42 Experiment1    Experiment1_42_EEF1G
## 5    457237144          42 Experiment1    Experiment1_42_RIPK2
## 6   2057977064          42 Experiment1    Experiment1_42_EIF4H
```

If we haven't computed fold changes from the raw "sumionarea" data, as it is the case in this example, we can invoke the function *tpp2dComputeFoldChanges* in order to do so:

```
fcData2d <- tpp2dComputeFoldChanges(configTable = config_tpp2d,
                                    data = data2d)
```

Thereon the function adds additional columns to our dataframe containing corresponding fold changes:

```
head(fcData2d)
```

```
##   gene_name qupm qssm signal_sum_5 signal_sum_1 signal_sum_0.143 signal_sum_0.02
## 1     CCND1    3    4    204841190    232467960        248774392       316622154
## 2  C17ORF39    1    1     65819416     65633403         99635379       112822532
## 3    NECAP1    3    3     98127667    119382560        113228677       217363144
## 4     EEF1G   17   59   3088494716   3716161024       4008219610      4973078201
```

```
## 5     RIPK2    5    5    259734512    303419382        323066842       355720486
## 6     EIF4H   21   45   1309348011   1469321178       1348496831      1630178705
##   signal_sum_0 temperature  experiment               unique_ID  rel_fc_5  rel_fc_1
## 1    370562621          42 Experiment1    Experiment1_42_CCND1 0.5527843 0.6273379
## 2    115419115          42 Experiment1 Experiment1_42_C17ORF39 0.5702644 0.5686528
## 3    159124932          42 Experiment1   Experiment1_42_NECAP1 0.6166706 0.7502442
## 4   5214069781          42 Experiment1    Experiment1_42_EEF1G 0.5923386 0.7127179
## 5    457237144          42 Experiment1    Experiment1_42_RIPK2 0.5680521 0.6635930
## 6   2057977064          42 Experiment1    Experiment1_42_EIF4H 0.6362306 0.7139638
##   rel_fc_0.143 rel_fc_0.02 rel_fc_0
## 1    0.6713424   0.8544363        1
## 2    0.8632485   0.9775030        1
## 3    0.7115709   1.3659905        1
## 4    0.7687315   0.9537805        1
## 5    0.7065630   0.7779781        1
## 6    0.6552536   0.7921268        1
```

We can then normalize the data by performing a median normalization on the fold changes, in order to account for experiment specific noise.

```
normData2d <- tpp2dNormalize(configTable = config_tpp2d,
                             data = fcData2d)

head(normData2d)
```

```
##   gene_name qupm qssm signal_sum_5 signal_sum_1 signal_sum_0.143 signal_sum_0.02
## 1     CCND1    3    4    204841190    232467960        248774392       316622154
## 2  C17ORF39    1    1     65819416     65633403         99635379       112822532
## 3    NECAP1    3    3     98127667    119382560        113228677       217363144
## 4     EEF1G   17   59   3088494716   3716161024       4008219610      4973078201
## 5     RIPK2    5    5    259734512    303419382        323066842       355720486
## 6     EIF4H   21   45   1309348011   1469321178       1348496831      1630178705
##   signal_sum_0 temperature  experiment               unique_ID  rel_fc_5  rel_fc_1
## 1    370562621          42 Experiment1    Experiment1_42_CCND1 0.5527843 0.6273379
## 2    115419115          42 Experiment1 Experiment1_42_C17ORF39 0.5702644 0.5686528
## 3    159124932          42 Experiment1   Experiment1_42_NECAP1 0.6166706 0.7502442
## 4   5214069781          42 Experiment1    Experiment1_42_EEF1G 0.5923386 0.7127179
## 5    457237144          42 Experiment1    Experiment1_42_RIPK2 0.5680521 0.6635930
## 6   2057977064          42 Experiment1    Experiment1_42_EIF4H 0.6362306 0.7139638
##   rel_fc_0.143 rel_fc_0.02 rel_fc_0 norm_rel_fc_5 norm_rel_fc_1 norm_rel_fc_0.143
## 1    0.6713424   0.8544363        1     0.9236180     0.8949713         0.9714708
## 2    0.8632485   0.9775030        1     0.9528247     0.8112501         1.2491699
## 3    0.7115709   1.3659905        1     1.0303623     1.0703116         1.0296838
## 4    0.7687315   0.9537805        1     0.9897072     1.0167760         1.1123984
## 5    0.7065630   0.7779781        1     0.9491282     0.9466935         1.0224370
## 6    0.6552536   0.7921268        1     1.0630441     1.0185534         0.9481894
##   norm_rel_fc_0.02 norm_rel_fc_0
## 1        0.9793027             1
## 2        1.1203543             1
## 3        1.5656149             1
## 4        1.0931650             1
## 5        0.8916710             1
## 6        0.9078873             1
```

```
# we have to update our fcStr, if we want the normalized columns to be used in the following analysis
fcStrUpdated <- "norm_rel_fc_"
```

A configuration file for the TPP-CCR function can be then generated using the function `tpp2dCreateCCRConfigFile`

```
config_ccr <- tpp2dCreateCCRConfigFile(configTable = config_tpp2d)
```

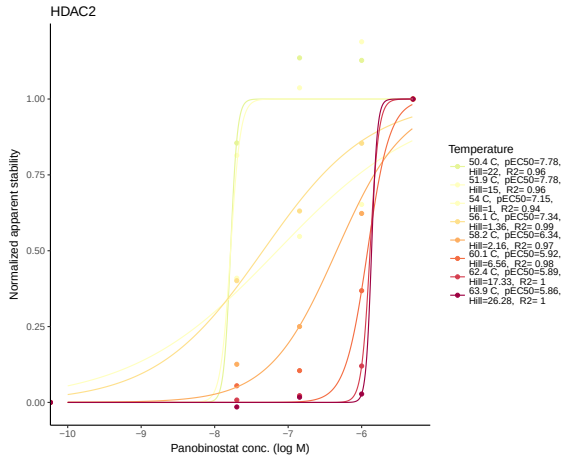
To run the TPP-CCR main function on our 2D-TPP data we now invoke:

```
ccr2dResults <- tpp2dCurveFit(configFile = config_ccr,
                              data = normData2d,
                              fcStr = fcStrUpdated)
```

Now we can plot the curves for any of the proteins for which at least one CCR curve could be fitted. In this case we choose HDAC2:

```
goodCurves <- tpp2dPlotCCRGoodCurves(configTable = config_tpp2d,
                                     data = ccr2dResults,
                                     fcStr = fcStrUpdated)
```

```
goodCurves[["HDAC2"]]
```



And we can also plot the single curves for each of the proteins with:

```
singleCurve <- tpp2dPlotCCRSingleCurves(configTable = config_tpp2d,
                                        data = ccr2dResults,
                                        fcStr = fcStrUpdated)
singleCurve[["HDAC2"]][["54"]]
```

References

[1] Daniel Martinez Molina, Rozbeh Jafari, Marina Ignatushchenko, Takahiro Seki, E Andreas Larsson, Chen Dan, Lekshmy Sreekumar, Yihai Cao, and Paer Nordlund. Monitoring drug target engagement in cells and tissues using the cellular thermal shift assay. *Science*, 341(6141):84–7, 2013.

[2] Mikhail M Savitski, Friedrich BM Reinhard, Holger Franken, Thilo Werner, Maria Fälth Savitski, Dirk Eberhard, Daniel Martinez Molina, Rozbeh Jafari, Rebecca Bakszt Dovega, Susan Klaeger, et al. Tracking cancer drugs in living cells by thermal profiling of the proteome. *Science*, 346(6205):1255784, 2014.

[3] Isabelle Becher, Thilo Werner, Carola Doce, Esther A Zaal, Cecilia R Berkers, Ina Tögel, Elsa Salzer, Marcus Bantscheff, and Mikhail M Savitski. Comprehensive thermal and chemoproteomics profiling identifies phenylalanine hydroxylase as a potent off-target of the histone deacetylase inhibitor panobinostat. *in submission*, 2016.

[4] Holger Franken, Toby Mathieson, Dorothee Childs, Gavain Sweetman, Thilo Werner, Wolfgang Huber, and Mikhail M Savitski. Thermal proteome profiling for unbiased identification of drug targets and detection of downstream effectors. *Nature protocols*, 10(10):1567 – 1593, 2015.

[5] Alexander Walker. *openxlsx: Read, Write and Edit XLSX Files*, 2015. R package version 2.4.0. URL: http://CRAN.R-project.org/package=openxlsx.