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 3.0.0 (Last revision 2016-09-28)

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")
```

For the data manipulations in this vignette, we also load the *dplyr* and *magrittr* packages:

```
library("dplyr", quietly = TRUE)
library("magrittr", quietly = TRUE)
```

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("panobinostat_2DTPP_smallExample")
```

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 %>% head

##       Compound Experiment Temperature 126 127L  127H 128L 128H 129L 129H  130L 130H 131L
## 1 Panobinostat    X020466        42.0   5    1 0.143 0.02    0    -    -     -    -    -
## 2 Panobinostat    X020466        44.1   -    -     -    -    -    5    1 0.143 0.02    0
## 3 Panobinostat    X020467        46.2   5    1 0.143 0.02    0    -    -     -    -    -
## 4 Panobinostat    X020467        48.1   -    -     -    -    -    5    1 0.143 0.02    0
## 5 Panobinostat    X020468        50.4   5    1 0.143 0.02    0    -    -     -    -    -
## 6 Panobinostat    X020468        51.9   -    -     -    -    -    5    1 0.143 0.02    0
##   RefCol Path
## 1   128H
## 2   131L
## 3   128H
## 4   131L
## 5   128H
## 6   131L

data_tpp2d %>% str(1)

## List of 6
##  $ X020466:'data.frame': 484 obs. of  15 variables:
##  $ X020467:'data.frame': 478 obs. of  15 variables:
##  $ X020468:'data.frame': 448 obs. of  15 variables:
##  $ X020469:'data.frame': 372 obs. of  15 variables:
##  $ X020470:'data.frame': 306 obs. of  15 variables:
##  $ X020471:'data.frame': 261 obs. of  15 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 %>% extract2("X020466") %>% colnames
```

```
##  [1] "clustername"              "representative"           "msexperiment_id"
##  [4] "qupm"                     "qusm"                     "sumionarea_protein_126"
##  [7] "sumionarea_protein_127L" "sumionarea_protein_127H" "sumionarea_protein_128L"
## [10] "sumionarea_protein_128H" "sumionarea_protein_129L" "sumionarea_protein_129H"
## [13] "sumionarea_protein_130L" "sumionarea_protein_130H" "sumionarea_protein_131L"
```

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

```
tpp2dResults <- analyze2DTPP(configFile = config_tpp2d,
                             data = data_tpp2d,
                             idVar = "representative",
                             fcStr = NULL,
                             intensityStr = "sumionarea_protein_",
                             methods = "doseResponse",
                             qualColName = c("qupm", "qusm"),
                             addCol = c("clustername","msexperiment_id"),
                             nonZeroCols = "qupm",
                             nCores = 2)

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

```
##                      Protein_ID   norm_rel_fc_protein_0_unmodified
##  X020466_42_IPI00000001.2:   1    Min.   :1
##  X020466_42_IPI00000005.1:   1    1st Qu.:1
##  X020466_42_IPI00000690.1:   1    Median :1
##  X020466_42_IPI00000811.2:   1    Mean   :1
##  X020466_42_IPI00000875.7:   1    3rd Qu.:1
##  X020466_42_IPI00001466.2:   1    Max.   :1
##  (Other)                 :4650
##  norm_rel_fc_protein_0.02_unmodified norm_rel_fc_protein_0.143_unmodified
##  Min.   :0.1767                      Min.   :0.2612
##  1st Qu.:0.9192                      1st Qu.:0.9364
##  Median :1.0000                      Median :1.0000
##  Mean   :1.0035                      Mean   :1.0105
##  3rd Qu.:1.0727                      3rd Qu.:1.0632
##  Max.   :4.6565                      Max.   :5.8855
##
##  norm_rel_fc_protein_1_unmodified norm_rel_fc_protein_5_unmodified
##  Min.   : 0.2422                  Min.   : 0.2512
##  1st Qu.: 0.9344                  1st Qu.: 0.9337
##  Median : 1.0000                  Median : 1.0000
##  Mean   : 1.0163                  Mean   : 1.0259
##  3rd Qu.: 1.0654                  3rd Qu.: 1.0589
##  Max.   :10.0240                  Max.   :17.0405
##
##  norm_rel_fc_protein_0_normalized_to_lowest_conc
##  Min.   :1
##  1st Qu.:1
##  Median :1
##  Mean   :1
##  3rd Qu.:1
##  Max.   :1
##
##  norm_rel_fc_protein_0.02_normalized_to_lowest_conc
##  Min.   :0.1767
##  1st Qu.:0.9192
##  Median :1.0000
##  Mean   :1.0035
##  3rd Qu.:1.0727
##  Max.   :4.6565
##
##  norm_rel_fc_protein_0.143_normalized_to_lowest_conc
##  Min.   :0.2612
##  1st Qu.:0.9364
##  Median :1.0000
##  Mean   :1.0105
##  3rd Qu.:1.0632
##  Max.   :5.8855
##
##  norm_rel_fc_protein_1_normalized_to_lowest_conc
##  Min.   : 0.2422
##  1st Qu.: 0.9344
##  Median : 1.0000
##  Mean   : 1.0163
##  3rd Qu.: 1.0654
##  Max.   :10.0240
##
##  norm_rel_fc_protein_5_normalized_to_lowest_conc norm_rel_fc_protein_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_protein_0.02_transformed norm_rel_fc_protein_0.143_transformed
##  Min.   :-0.884                       Min.   :-1.201
##  1st Qu.:-0.154                       1st Qu.: 0.086
##  Median : 0.297                       Median : 0.376
##  Mean   : 0.302                       Mean   : 0.400
##  3rd Qu.: 0.614                       3rd Qu.: 0.662
##  Max.   : 2.542                       Max.   : 3.294
##  NA's   :4421                         NA's   :4421
##  norm_rel_fc_protein_1_transformed norm_rel_fc_protein_5_transformed     pEC50
##  Min.   :-0.961                    Min.   :0.000                    Min.   :5.728
##  1st Qu.: 0.095                    1st Qu.:0.000                    1st Qu.:6.696
##  Median : 0.313                    Median :0.000                    Median :7.778
##  Mean   : 0.400                    Mean   :0.379                    Mean   :7.346
##  3rd Qu.: 0.652                    3rd Qu.:1.000                    3rd Qu.:8.126
##  Max.   : 2.925                    Max.   :1.000                    Max.   :8.126
##  NA's   :4421                      NA's   :4421                     NA's   :4421
##      slope              R_sq          plot          compound_effect meets_FC_requirement
##  Min.   :-50.000   Min.   :-0.068   NA's:4656   destabilized: 146   Mode :logical
##  1st Qu.:-10.804   1st Qu.: 0.545               stabilized  :  89   FALSE:4537
##  Median : -1.000   Median : 0.723               NA's        :4421   TRUE :119
##  Mean   : -8.302   Mean   : 0.675                                   NA's :0
##  3rd Qu.:  1.159   3rd Qu.: 0.881
##  Max.   : 50.000   Max.   : 1.000
##  NA's   :4421      NA's   :4421
##  passed_filter   pEC50_outside_conc_range model_converged       pEC50_quality_check
##  Mode :logical   Mode :logical            Mode:logical    5.72818301656452:  12
##  FALSE:4601      FALSE:111                TRUE:235        6.07074587494624:   6
##  TRUE :55        TRUE :124                NA's:4421       7.44099730847312:   6
##  NA's :0         NA's :4421                               6.75587159170968:   2
##                                                           5.83469502048232:   1
##                                                           (Other)         :  84
##                                                           NA's            :4545
##  sufficient_data_for_fit protein_identified_in       representative      qupm
##  Mode:logical            Mode:logical            IPI00000001.2:  12   Min.   : 1.000
##  TRUE:235                TRUE:4656               IPI00000005.1:  12   1st Qu.: 3.000
##  NA's:4421               NA's:0                  IPI00000690.1:  12   Median : 7.000
##                                                  IPI00000811.2:  12   Mean   : 9.149
##                                                  IPI00000875.7:  12   3rd Qu.:12.000
##                                                  IPI00001914.1:  12   Max.   :87.000
##                                                  (Other)      :4584
##       qusm          clustername   msexperiment_id sumionarea_protein_5
##  Min.   :  1.00   A2M     :  12   Min.   :39093   Min.   :2.063e+05
##  1st Qu.:  5.00   ABHD10 :  12   1st Qu.:39101   1st Qu.:7.696e+07
##  Median : 11.00   ACAA1   :  12   Median :39106   Median :2.511e+08
##  Mean   : 19.57   ACO1    :  12   Mean   :39104   Mean   :7.182e+08
##  3rd Qu.: 23.00   ACO2    :  12   3rd Qu.:39108   3rd Qu.:7.382e+08
##  Max.   :263.00   ACTC1   :  12   Max.   :39110   Max.   :2.125e+10
##                   (Other):4584
##  sumionarea_protein_1 sumionarea_protein_0.143 sumionarea_protein_0.02
##  Min.   :3.819e+05    Min.   :3.579e+05         Min.   :4.335e+05
##  1st Qu.:7.604e+07    1st Qu.:8.079e+07         1st Qu.:8.401e+07
##  Median :2.512e+08    Median :2.591e+08         Median :2.739e+08
##  Mean   :7.542e+08    Mean   :7.554e+08         Mean   :8.100e+08
##  3rd Qu.:7.682e+08    3rd Qu.:7.857e+08         3rd Qu.:8.331e+08
##  Max.   :2.138e+10    Max.   :1.924e+10         Max.   :2.249e+10
##
##  sumionarea_protein_0  temperature     experiment  rel_fc_protein_5  rel_fc_protein_1
##  Min.   :2.925e+05    Min.   :42.0   X020466:968   Min.   : 0.3487   Min.   :0.2985
##  1st Qu.:7.345e+07    1st Qu.:46.2   X020467:950   1st Qu.: 0.7894   1st Qu.:0.8231
##  Median :2.574e+08    Median :50.4   X020468:894   Median : 0.8964   Median :0.9197
##  Mean   :8.599e+08    Mean   :51.6   X020469:738   Mean   : 0.9935   Mean   :0.9753
##  3rd Qu.:8.554e+08    3rd Qu.:56.1   X020470:600   3rd Qu.: 1.0878   3rd Qu.:1.0588
##  Max.   :2.644e+10    Max.   :63.9   X020471:506   Max.   :17.1835   Max.   :8.6463
##
##  rel_fc_protein_0.143 rel_fc_protein_0.02 rel_fc_protein_0
##  Min.   :0.3887       Min.   : 0.1882     Min.   :1
##  1st Qu.:0.8156       1st Qu.: 0.8413     1st Qu.:1
##  Median :0.9415       Median : 0.9601     Median :1
##  Mean   :1.0187       Mean   : 1.0974     Mean   :1
##  3rd Qu.:1.1447       3rd Qu.: 1.2027     3rd Qu.:1
##  Max.   :6.2354       Max.   :10.0917     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 <- tpp2dImportData(configTable = config_tpp2d,
                          data = data_tpp2d,
                          idVar = "representative",
                          fcStr = NULL,
                          intensityStr = "sumionarea_protein_",
                          qualColName = c("qupm", "qusm"),
                          addCol = c("clustername","msexperiment_id"))
```

```
head(data2d)
```

```
##   representative qupm qusm clustername msexperiment_id sumionarea_protein_5
## 1  IPI00028098.1    3    4       CCND1           39106            204841190
## 2  IPI00217151.3    1    1    C17ORF39           39106             65819416
## 3  IPI00170916.1    3    3      NECAP1           39106             98127667
## 4  IPI00000875.7   17   59       EEF1G           39106           3088494716
## 5  IPI00021917.1    5    5       RIPK2           39106            259734512
## 6  IPI00014263.1   21   45       EIF4H           39106           1309348011
##   sumionarea_protein_1 sumionarea_protein_0.143 sumionarea_protein_0.02
## 1            232467960                248774392               316622154
## 2             65633403                 99635379               112822532
## 3            119382560                113228677               217363144
## 4           3716161024               4008219610              4973078201
## 5            303419382                323066842               355720486
## 6           1469321178               1348496831              1630178705
##   sumionarea_protein_0 temperature experiment                unique_ID
## 1            370562621          42    X020466 X020466_42_IPI00028098.1
## 2            115419115          42    X020466 X020466_42_IPI00217151.3
## 3            159124932          42    X020466 X020466_42_IPI00170916.1
## 4           5214069781          42    X020466 X020466_42_IPI00000875.7
## 5            457237144          42    X020466 X020466_42_IPI00021917.1
## 6           2057977064          42    X020466 X020466_42_IPI00014263.1
```

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,
                                    dataTable = data2d,
                                    intensityStr="sumionarea_protein_")
```

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

```
head(fcData2d)
```

```
##   representative qupm qusm clustername msexperiment_id sumionarea_protein_5
## 1  IPI00028098.1    3    4       CCND1           39106            204841190
## 2  IPI00217151.3    1    1    C17ORF39           39106             65819416
## 3  IPI00170916.1    3    3      NECAP1           39106             98127667
## 4  IPI00000875.7   17   59       EEF1G           39106           3088494716
## 5  IPI00021917.1    5    5       RIPK2           39106            259734512
## 6  IPI00014263.1   21   45       EIF4H           39106           1309348011
##   sumionarea_protein_1 sumionarea_protein_0.143 sumionarea_protein_0.02
## 1            232467960                248774392               316622154
## 2             65633403                 99635379               112822532
## 3            119382560                113228677               217363144
## 4           3716161024               4008219610              4973078201
## 5            303419382                323066842               355720486
## 6           1469321178               1348496831              1630178705
##   sumionarea_protein_0 temperature experiment                unique_ID rel_fc_protein_5
## 1            370562621          42    X020466 X020466_42_IPI00028098.1        0.5527843
## 2            115419115          42    X020466 X020466_42_IPI00217151.3        0.5702644
## 3            159124932          42    X020466 X020466_42_IPI00170916.1        0.6166706
## 4           5214069781          42    X020466 X020466_42_IPI00000875.7        0.5923386
## 5            457237144          42    X020466 X020466_42_IPI00021917.1        0.5680521
## 6           2057977064          42    X020466 X020466_42_IPI00014263.1        0.6362306
##   rel_fc_protein_1 rel_fc_protein_0.143 rel_fc_protein_0.02 rel_fc_protein_0
## 1        0.6273379            0.6713424           0.8544363                1
## 2        0.5686528            0.8632485           0.9775030                1
## 3        0.7502442            0.7115709           1.3659905                1
## 4        0.7127179            0.7687315           0.9537805                 1
## 5        0.6635930            0.7065630           0.7779781                 1
## 6        0.7139638            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 <- tpp2dDoMedianNorm(configTable = config_tpp2d,
                                dataTable = fcData2d)

head(normData2d)
```

```
##   representative qupm qusm clustername msexperiment_id sumionarea_protein_5
## 1  IPI00028098.1    3    4       CCND1           39106            204841190
## 2  IPI00217151.3    1    1    C17ORF39           39106             65819416
## 3  IPI00170916.1    3    3      NECAP1           39106             98127667
## 4  IPI00000875.7   17   59       EEF1G           39106           3088494716
## 5  IPI00021917.1    5    5       RIPK2           39106            259734512
## 6  IPI00014263.1   21   45       EIF4H           39106           1309348011
##   sumionarea_protein_1 sumionarea_protein_0.143 sumionarea_protein_0.02
## 1            232467960                248774392               316622154
## 2             65633403                 99635379               112822532
## 3            119382560                113228677               217363144
## 4           3716161024               4008219610              4973078201
## 5            303419382                323066842               355720486
## 6           1469321178               1348496831              1630178705
##   sumionarea_protein_0 temperature experiment                unique_ID rel_fc_protein_5
## 1            370562621          42    X020466 X020466_42_IPI00028098.1        0.5527843
## 2            115419115          42    X020466 X020466_42_IPI00217151.3        0.5702644
## 3            159124932          42    X020466 X020466_42_IPI00170916.1        0.6166706
## 4           5214069781          42    X020466 X020466_42_IPI00000875.7        0.5923386
## 5            457237144          42    X020466 X020466_42_IPI00021917.1        0.5680521
## 6           2057977064          42    X020466 X020466_42_IPI00014263.1        0.6362306
##   rel_fc_protein_1 rel_fc_protein_0.143 rel_fc_protein_0.02 rel_fc_protein_0
## 1        0.6273379            0.6713424           0.8544363                 1
## 2        0.5686528            0.8632485           0.9775030                 1
## 3        0.7502442            0.7115709           1.3659905                 1
## 4        0.7127179            0.7687315           0.9537805                 1
## 5        0.6635930            0.7065630           0.7779781                 1
## 6        0.7139638            0.6552536           0.7921268                 1
##   norm_rel_fc_protein_5 norm_rel_fc_protein_1 norm_rel_fc_protein_0.143
## 1             0.9236180             0.8949713                 0.9714708
## 2             0.9528247             0.8112501                 1.2491699
## 3             1.0303623             1.0703116                 1.0296838
## 4             0.9897072             1.0167760                 1.1123984
## 5             0.9491282             0.9466935                 1.0224370
## 6             1.0630441             1.0185534                 0.9481894
##   norm_rel_fc_protein_0.02 norm_rel_fc_protein_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 folloeing analysis
fcStrUpdated <- "norm_rel_fc_protein_"
```

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

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

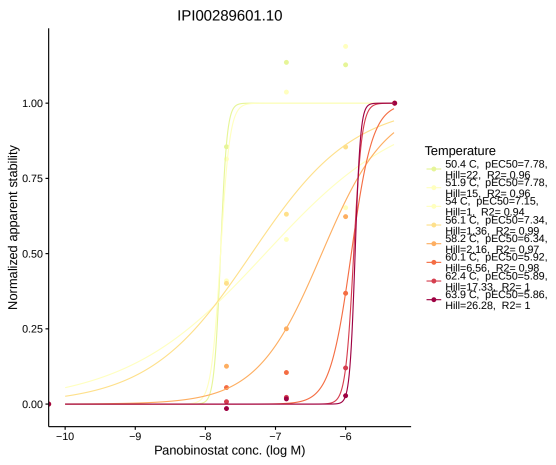
To run the TPP-CCR main function on our 2D-TPP data we now invoke:

```
ccr2dResults <- tpp2dRunTPPCCR(configFile = config_ccr,
                               dataTable = normData2d,
                               fcStr = fcStrUpdated,
                               idVar = "unique_ID")
```

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 with it's id IPI00289601.10:

```
goodCurves <- tpp2dPlotCCRGoodCurves(configTable = config_tpp2d,
                                     dataTable = ccr2dResults,
                                     idVar = "representative",
                                     fcStr = fcStrUpdated)
```

```
goodCurves[["IPI00289601.10"]]
```



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

```
singleCurve <- tpp2dPlotCCRSingleCurves(configTable = config_tpp2d,
                                        dataTable = ccr2dResults,
                                        idVar = "representative",
                                        fcStr = fcStrUpdated)
singleCurve[["IPI00289601.10"]][["54"]]
```

2.3 Quality control analyses

In order to access the quality of the experimental 2D-TPP data set acquired in a specific cell line, we recommend to compare the data with vehicle TR experiments (at least two replicates) of the same cell line. For the analysis of this data we supply a QC-workflow that enables comparison of treatment and non-treatment samples with reference data.

In order to start this workflow the first thing we need to do, is to generate a cell line specific TR reference object. We also need to specify the result path where this object should be stored:

```
resultPath = file.path(getwd(), 'Panobinostat_Vignette_Example_2D')
if (!file.exists(resultPath)) dir.create(resultPath, recursive = TRUE)
```

```
trConfig <- file.path(system.file("example_data", package="TPP"),
                      "2D_example_data/panobinostat_ex_confg.csv")

tpp2dCreateTPPTRreference(trConfigTable = trConfig,
                          resultPath = resultPath,
                          outputName = "some_cell_line_TR_reference",
                          createFCboxplots = FALSE)
```

For the purpose of explaining this worflow, we will use a reference data set of a HepG2 cell line supplied with this package. Originating from this object we can now perform various quality control steps. First of all by setting the *createFCboxplots* flag to true, we can generate box plot melting curves of the reference data which are first of all informative of the quality of the reference data and illustrate melting behavior of all proteins without any treatment.

Calling the function will generate a couple of output files in the indicated output directory.

- The `tppRefData.RData` file ist the most important one. This is the file that has to be referenced by indication of a system path to this file when calling functions to generate the 2D-TPP spline plots and perform an F test. When loaded in *R* the object `tppRefData` represents a list with the following elements:
  - tppCfgTable: the TPP-TR configtable which was used for generating this object
  - sumResTable a list of two elements:
    - detail: the exact result data from the TR analysis and
    - summary: a summary of the analyzed TR data comprising the median and standard deviation values of the measurements at the different temperatures (encoded by the isobaric labels)
    - temperatures: a table listing the temperatures which were used in the TR experiment in the different replicates
    - lblsByTemp: a table matching each temperature to an isobaric label
- An excel file which summarizes the data present in `tppRefData` on different sheets
- Textfiles representing the sheets of the excel file as plain text
- `normalizedData.RData` containing the TPP-TR data after normalization
- `resultTable.RData` containing the TPP-TR analysis result table

Secondly, we can generate plots which visualize the melting point temperatures of the 2D-TPP data in comparison to the TR reference data. Here we demonstrate this function on a subset of the proteins:

```
# set the system path for the HepG2 TR reference data set:
trRef <- file.path(system.file("data", package="TPP"), "HepG2_trRefData.RData")

plotData <- (ccr2dResults %>% filter(!is.na(compound_effect)) %>%
  arrange(representative))[1:10,]
pEC50QC <- tpp2dPlotQCpEC50(resultTable = plotData,
                            resultPath = resultPath,
                            trRef = trRef,
                            idVar = "representative")
```

We have therefore used the *ccr2dResults* data frame which we previously generated by invoking the TPP-CCR routine and the the respective configTable.

Moreover, we can generate plots that visualize the distributions of fold changes over the different treatment concentrations and temperatures and how the normalization affected them (of course only if we previously performed a normalization). The function automatically also visualizes various other characteristics of the data, such as how proteins behave in neighboring temperatures which are multiplexed. It can be invoked as follows:

```
tpp2dPlotQChist(configFile = config_tpp2d,
                resultTable = ccr2dResults,
                resultPath = resultPath,
                trRef = trRef)
```

2.4 Spline fits of treatment effects over temperature

In order to access whether the drug treatment has a significant impact on altering the thermal stability of specific proteins a function was implemented which illustrates the course of stability of a certain protein over different temperatures based on a reference data set. A natural cubic spline fitted to the reference data is then used to infer the relative stability curves of proteins with different concentrations of treatment which are in turn fitted by natural cubic splines. The cubic spline with n degrees of freedom on $[a, b]$ obeys:

- $S(x) \in C^2[a, b]$

- $a = t_0 < t_1 < ... < t_n = b$

and:

$$S(x) = \begin{cases} S_0(x) = a_0x^3 + b_0x^2 + c_0x + d_0, & t_0 \le x \le t_1 \\ S_1(x) = a_1x^3 + b_1x^2 + c_1x + d_1, & t_1 \le x \le t_2 \\ . \\ . \\ . \\ S_{n-1}(x) = a_{n-1}x^3 + b_{n-1}x^2 + c_{n-1}x + d_{n-1}, & t_{n-1} \le x \le t_n \end{cases} \tag{1}$$

a *natural cubic spline* additionally contrains that it's function has to be linear beyond the boundary knots with constrains that both the first and the last section of the cubic spline has to be linear.

The function to perform this analysis can be invoked by:

```
trRef <- file.path(system.file("data", package="TPP"), "HepG2_trRefData.RData")

analysisResults <- tpp2dSplineFitAndTest(data_2D = normData2d,
                                         trRefDataPath = trRef,
                                         idVar = "representative",
                                         fcStr = "norm_rel_fc_protein_",
                                         refFcStr = "norm_rel_fc_protein_",
                                         doPlot = FALSE,
                                         resultPath = resultPath)
```

Moreover, these fits can be used then, in order to access confidence on whether the curves fitting the relative treatment data points represent the data better than a model which does not distinguish between the different treatment concentrations. The confidence assessment is thereby based on a moderated F statistic adapted from a method by Storey and others [6] which they developed for microarray time course data. The method calculates a moderated F statistic following:

$$F = \frac{\mathrm{SS}_0 - \mathrm{SS}_1}{\tilde{s}^2(\sigma^2, \mathrm{df}_2)} \tag{2}$$

with SS_0 representing the sum of squares of the null model (fitting the data without distinguishing between different treatment concentrations) and SS_1 those of the full model (which fits the data by in this case 5 different splines for every treatment concentration respectively). With $\tilde{s}^2$ representing the empirical Bayes estimator for SS_1, with $\mathrm{df}_2 = n - \nu_1$, where ν_1 denoted the parameters of the full model and n denotes the number of data points.

```
analysisResults %>% filter(representative == "IPI00289601.10") %>%
  select(temperature, p_NPARC, p_adj_NPARC)

##    temperature p_NPARC p_adj_NPARC
## 1         42.0       0           0
## 2         44.1       0           0
## 3         46.2       0           0
## 4         48.1       0           0
## 5         50.4       0           0
## 6         51.9       0           0
## 7         54.0       0           0
## 8         56.1       0           0
## 9         58.2       0           0
## 10        60.1       0           0
## 11        62.4       0           0
## 12        63.9       0           0
```

By defining the `methods` argument to include `"splineFit"`, one prompts the main function `analyze2DTPP` to directly perform spline fits and a moderated F-test for each protein in the data set.

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.

[6] John D Storey, Wenzhong Xiao, Jeffrey T Leek, Ronald G Tompkins, and Ronald W Davis. Significance analysis of time course microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America*, 102(36):12837–42, 2005. URL: http://www.pnas.org/content/102/36/12837.abstract, doi:10.1073/pnas.0504609102.