Non-parametric analysis of thermal proteome profiles

Dorothee Childsh

08 August 2018, 21:49:36

Contents

1 Introduction

This workflow shows how to reproduce the analysis described by Childs, Bach, Franken et al. (2018): Non-parametric analysis of thermal proteome profiles reveals novel drug-binding proteins.

2 Preparation

Load necessary packages:

```r
library(tidyverse)
library(broom)
```

3 Data import

First we load the data from the different TPP experiments. All data have been downloaded from the supplements of the respective publications (M. M. Savitski et al. 2014, Franken et al. (2015), Reinhard et al. (2015)) converted into tidy format, and concatenated into one table. This table will be made available as supplementary material to the paper. Until then, it can be found in the same folder as this vignette.

```r
tppData <- readRDS("tppData.Rds")
```

Let's take a look at the imported data:

```r
tppData %>%
  mutate(molarDrugConcentration = factor(molarDrugConcentration),
         replicate = factor(replicate),
         dataset = factor(dataset)) %>%
  summary %>%
  knitr::kable()
```

	dataset	uniqueID	relAbundance	temperature	molarDrugConcentration	replicate	unic
	ATP :268000	Length:1432280	Min. : 0.0	Min. :25.00	0 :716140	1:716140	Min
	Dasatinib 0.5:308520	Class :character	1st Qu.: 0.1	1st Qu.:44.00	5e-07:154260	2:716140	1st
	Dasatinib 5 :308520	Mode :character	Median : 0.6	Median :52.50	1e-06:120080		Mec
	Panobinostat :240160		Mean : 0.6	Mean :51.86	5e-06:154260		Mea
	Staurosporine:307080		3rd Qu.: 1.0	3rd Qu.:59.00	2e-05:153540		3rd
			Max. :577.6	Max. :67.00	0.002:134000		Max
			NA's :372809				NA'

4 Data preprocessing

First, we remove all decoy proteins remaining in the panobinostat data. They can be recognized by the prefix `###`, which was assigned by the quantification software `isobarQuant`.

```r
tppData <- tppData %>% filter(!grepl("###[[:alnum:]]*###", uniqueID))
```

Next, we remove all proteins that were not found with at least one unique peptide

```r
tppData <- filter(tppData, uniquePeptideMatches >= 1)
```

Next, we remove all proteins that only contain missing values

```r
tppData <- tppData %>% filter(!is.na(relAbundance))
```

Finally, we remove all proteins not reproducibly observed with full melting curves in both replicates and treatment groups per dataset. A full melting curve is defined by the presence of measurements at all 10 temperatures for the given experimental group.

```r
tppData <- tppData %>%
  group_by(dataset, uniqueID) %>%
  mutate(n = n()) %>%
  group_by(dataset) %>%
  mutate(max_n = max(n)) %>%
  filter(n == max_n) %>%
  dplyr::select(-n, -max_n)
```

4.1 Reproduce Table 1 of the paper

Count the numbers of proteins remaining in each dataset. They coincide with the values reported in Table 1.

```r
tppData %>%
  distinct(dataset, uniqueID) %>%
  distinct %>%
  group_by(dataset) %>%
  tally %>%
  knitr::kable()
```

dataset	n
ATP	4177
Dasatinib 0.5	4625
Dasatinib 5	4154
Panobinostat	3649
Staurosporine	4505

5 Illustrative example

We first illustrate the principles of nonparametric analysis of response curves (NPARC) on an example protein (STK4) from the staurosporine dataset. The same protein is shown in Figures 1 and 2 of the paper.

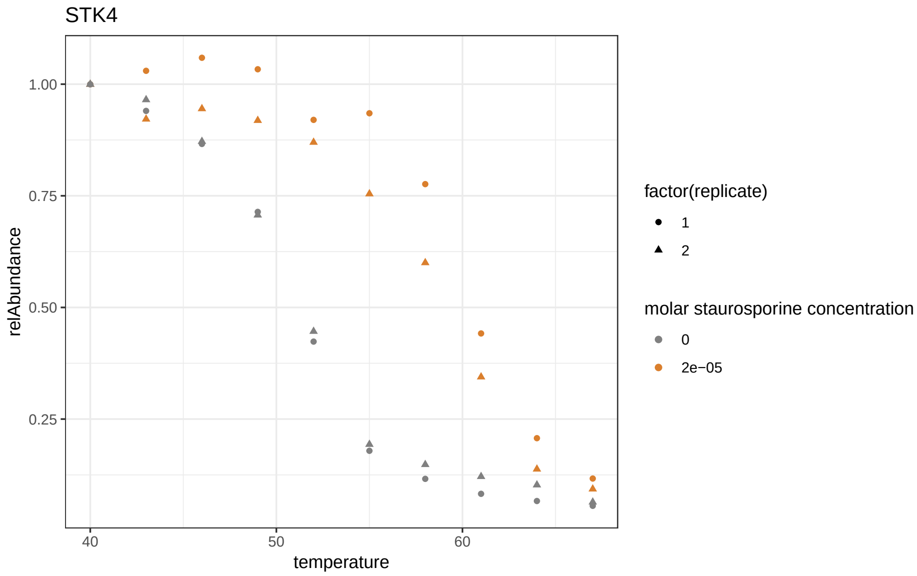
Select protein:

```
stk4 <- filter(tppData, dataset == "Staurosporine", grepl("STK4", uniqueID))
```

Plot measurements:

```
stk4_plot <- ggplot(stk4, aes(x = temperature, y = relAbundance)) +
  geom_point(aes(shape = factor(replicate), color = factor(molarDrugConcentration))) +
  theme_bw() +
  ggtitle("STK4") +
  scale_color_manual("molar staurosporine concentration",
                     values = c("#808080", "#da7f2d"))

print(stk4_plot)
```



To assess whether there is a significant difference between both treatment groups, we fit a null and an alternative models to the data. The null model fits a sigmoid melting curve through all data points irrespective of experimental condition. The alternative model fits separate melting curves per experimental group (vehicle: 0 muM staurosporine, treatment: 20 muM staurosporine).

Because we have to repeat the fitting several times in this workflow, we define a function that we can call repeatedly:

```
fitSingleSigmoid <- function(x, y){
  m <- try(
    nls(formula= y ~ (1 - Pl)  / (1+exp((b - a/x))) + Pl,
        start=c(Pl = 0, a = 550, b = 10),
        data=list(x=x, y=y),
        na.action = na.exclude,
        algorithm = "port",
        lower = c(0.0, 1e-5, 1e-5),
        upper = c(1.5, 15000, 250),
        control = nls.control(maxiter=50, warnOnly = TRUE)),
    silent = TRUE)
  return(m)
}
```

5.1 Fit null models

First, we use this function to fit the null model. In order to add the predicted curve to our data frame, we use the function `augment` from the `broom` package to obtain the predictions in tabular format. It also returns the residuals which we will need later for the hypothesis test.

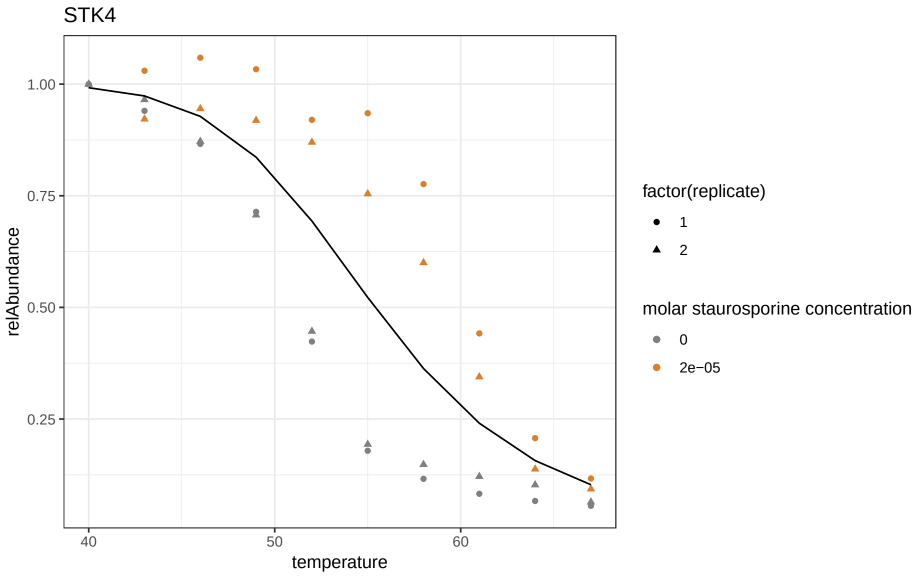
```
nullFit <- fitSingleSigmoid(x = stk4$temperature, y = stk4$relAbundance)
nullPredictions <- broom::augment(nullFit)

stk4$nullPrediction <- nullPredictions$.fitted
stk4$nullResiduals <- nullPredictions$.resid
```

Plot the curve predicted by the null model:

```
stk4_plot <- stk4_plot +
  geom_line(data = stk4, aes(y = nullPrediction))

print(stk4_plot)
```



5.2 Fit alternative models

Next we fit the alternative model. Again, we compute the predicted values and the corresponding residuals by the `broom` package.

```
alternativePredictions <- stk4 %>%
# Fit separate curves per treatment group:
  group_by(molarDrugConcentration) %>%
  do({
    fit = fitSingleSigmoid(x = .$temperature,
                           y = .$relAbundance)
    broom::augment(fit)
  }) %>%
  ungroup %>%
  # Rename columns for merge to data frame:
  dplyr::rename(alternativePrediction = .fitted,
                alternativeResiduals = .resid,
                temperature = x,
                relAbundance = y)
```

Add the predicted values and corresponding residuals to our data frame:

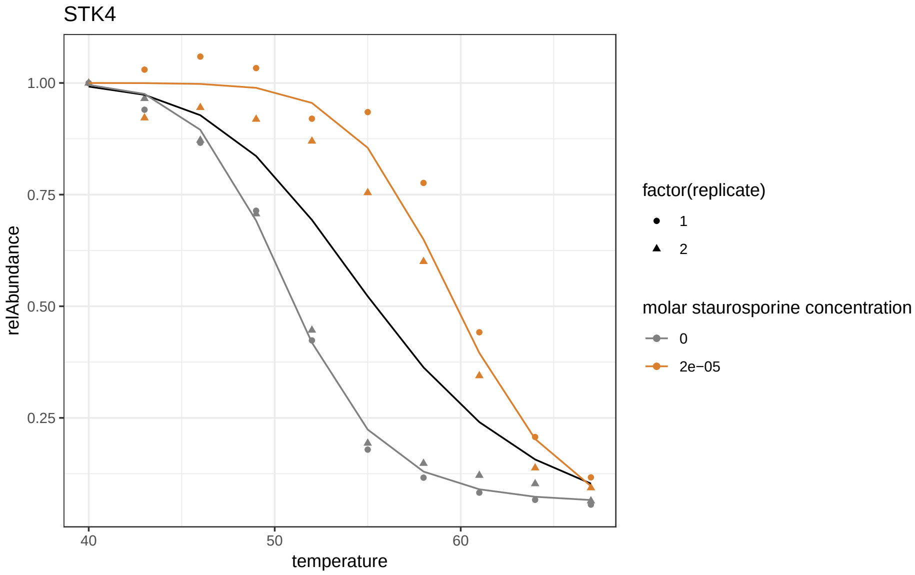
```
stk4 <- stk4 %>%
  left_join(alternativePredictions,
            by = c("relAbundance", "temperature", "molarDrugConcentration")) %>%
  distinct()
```

5.3 Reproduce Figure 2 (A)/(B) of the paper

Add the curves predicted by the alternative model to the plot:

```
stk4_plot <- stk4_plot +
  geom_line(data = distinct(stk4, temperature, molarDrugConcentration, alternativePrediction),
            aes(y = alternativePrediction, color = factor(molarDrugConcentration)))

print(stk4_plot)
```



This plot corresponds to Figures 2(A) and 2(B) in the paper.

5.4 Compute RSS values

In order to quantify the improvement in goodness-of-fit of the alternative model relative to the null model, we compute the sum of squared residuals (RSS).

```r
rssPerModel <- stk4 %>%
  summarise(rssNull = sum(nullResiduals^2),
            rssAlternative = sum(alternativeResiduals^2))

knitr::kable(rssPerModel, digits = 4)
```

dataset	rssNull	rssAlternative
Staurosporine	1.2181	0.0831

These values will be used to construct the F-statistic according to

$$\mathrm{F}_i = \frac{\mathrm{d}_{i2}}{\mathrm{d}_{i1}} \cdot \frac{\mathrm{RSS}^0_i - \mathrm{RSS}^1_i}{\mathrm{RSS}^1_i}. \tag{1}$$

To compute this statistic and to derive a p-value, we need the degrees of freedom $\mathrm{d}_{i1}, \mathrm{d}_{i2}$. They cannot be analytically derived due to the correlated nature of the measurements. The paper describes how to estimate these values from the RSS-values of all proteins in the dataset. In the following Section, we illustrate how to repeat the model fitting for all proteins of a dataset and how to perform hypothesis testing on these models.

6 Analyzing the datasets

Bibliography

Franken, Holger, Toby Mathieson, Dorothee Childs, Gavain M A Sweetman, Thilo Werner, Ina Tögel, Carola Doce, et al. 2015. "Thermal Proteome Profiling for Unbiased Identification of Direct and Indirect Drug Targets Using Multiplexed Quantitative Mass Spectrometry." *Nat. Protoc.* 10 (10): 1567–93.

Reinhard, Friedrich B M, Dirk Eberhard, Thilo Werner, Holger Franken, Dorothee Childs, Carola Doce, Maria Fälth Savitski, et al. 2015. "Thermal Proteome Profiling Monitors Ligand Interactions with Cellular Membrane Proteins." *Nat. Methods* 12 (12): 1129–31.

Savitski, Mikhail M, Friedrich B M Reinhard, Holger Franken, Thilo Werner, Maria Fälth Savitski, Dirk Eberhard, Daniel Martinez Molina, et al. 2014. "Tracking Cancer Drugs in Living Cells by Thermal Profiling of the Proteome." *Science* 346 (6205): 1255784.