

Panobinostat GP example fits

Nils Kurzawa

3/13/2020

Load required libraries

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.2.1 --
## v ggplot2 3.2.1      v purrr  0.3.3
## v tibble  2.1.3      v dplyr  0.8.3
## v tidyr   1.0.0      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(NPARC)
library(GP4TPP)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##      select

reRun <- FALSE
```

Load data

```
panoTppData <- readRDS("../data/tppData_preprocessed.Rds") %>%
  filter(dataset == "Panobinostat")
```

Let's have a look at HDAC1 in isolation, to check how GP curves look like:

```
hdac1_treat_df <- filter(panoTppData, uniqueID == "HDAC1",
  compoundConcentration == 1) %>%
  dplyr::select(x = temperature,
    y = relAbundance)

hdac1_treat_df
```

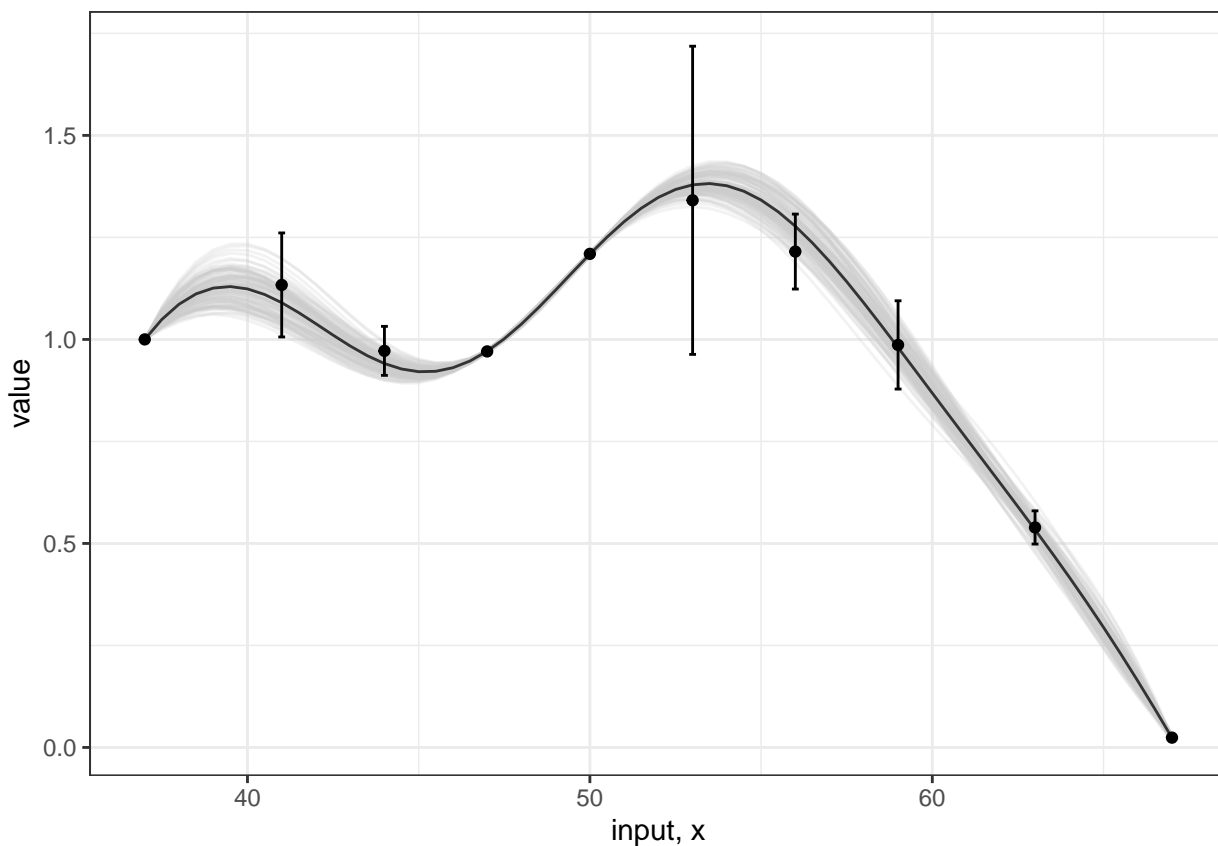
```
## # A tibble: 20 x 2
##       x     y
##   <dbl> <dbl>
## 1    37 1.000
## 2    41 1.09
## 3    44 0.951
## 4    47 0.972
## 5    50 1.21
## 6    53 1.21
## 7    56 1.18
```

```
## 8    59 0.948
## 9    63 0.554
## 10   67 0.0213
## 11   37 1.000
## 12   41 1.18
## 13   44 0.993
## 14   47 0.969
## 15   50 1.21
## 16   53 1.47
## 17   56 1.25
## 18   59 1.02
## 19   63 0.525
## 20   67 0.0268
```

Fit the melting profile of HDAC1 in the treatment condition

```
hdac1GpFit <- fitGP(in_df = hdac1_treat_df,
  x_range = seq(37, 67, 0.5),
  n_samples = 100,
  l = 8)
```

```
hdac1GpFit$plot
```



Let's also look at the vehicle condition

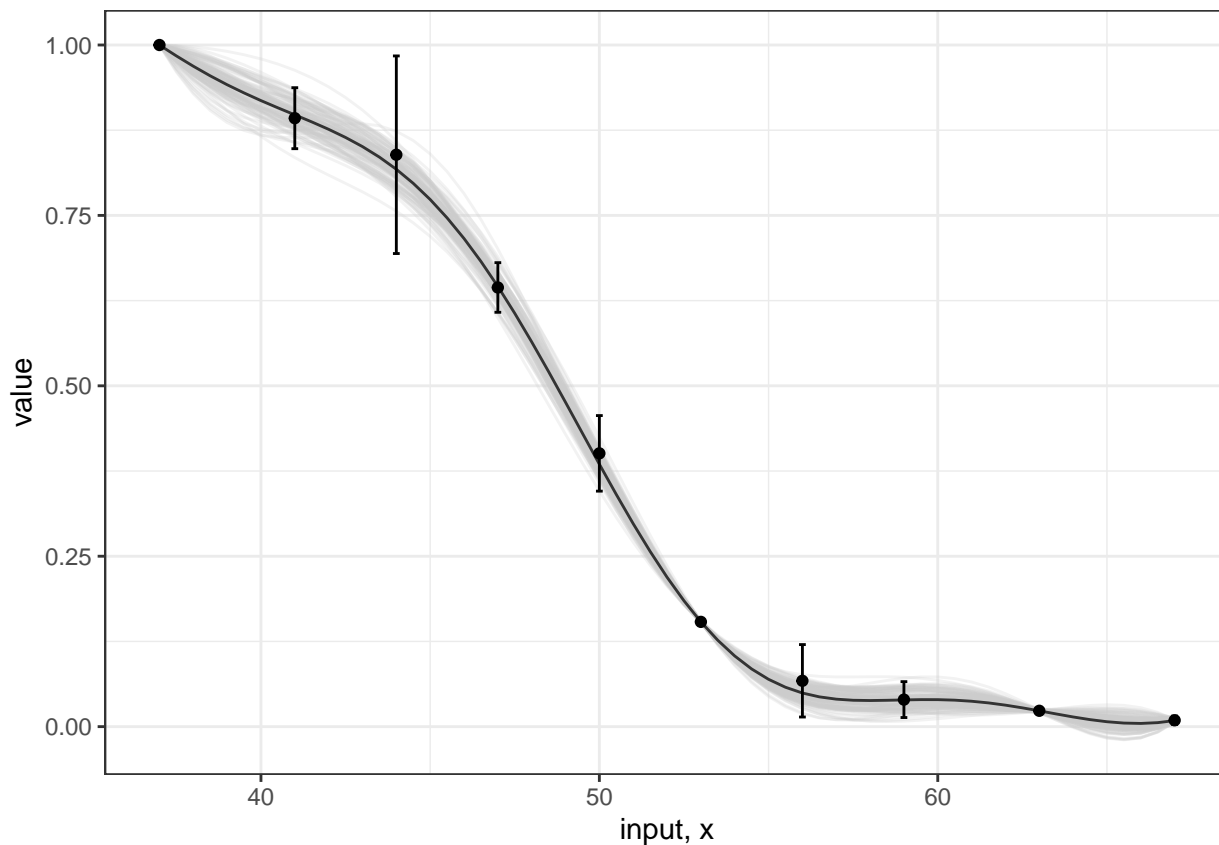
```
hdac1_veh_df <- filter(panoTppData, uniqueID == "HDAC1",
  compoundConcentration == 0) %>%
  dplyr::select(x = temperature,
    y = relAbundance)
```

```
hdac1_veh_df
```

```
## # A tibble: 20 x 2
##       x       y
##   <dbl> <dbl>
## 1    37 1.000
## 2    41 0.877
## 3    44 0.788
## 4    47 0.631
## 5    50 0.421
## 6    53 0.154
## 7    56 0.0860
## 8    59 0.0490
## 9    63 0.0245
## 10   67 0.00714
## 11   37 1.000
## 12   41 0.908
## 13   44 0.890
## 14   47 0.657
## 15   50 0.381
## 16   53 0.153
## 17   56 0.0484
## 18   59 0.0303
## 19   63 0.0220
## 20   67 0.0115
```

```
hdac1VehGpFit <- fitGP(in_df = hdac1_veh_df,
                        x_range = seq(37, 67, 0.5),
                        n_samples = 100,
                        l = 8)
```

```
hdac1VehGpFit$plot
```



Reformat data frame to be compatible with function

```
panoFitDf <- panoTppData %>%
  dplyr::select(gene_name = uniqueID,
                temperature,
                rel_value = relAbundance,
                var = compoundConcentration)
```

```
panoFitDf
```

```
## # A tibble: 145,960 x 4
```

```
##   gene_name temperature rel_value  var
##   <chr>         <dbl>     <dbl> <dbl>
## 1 AACS           37      1.000     1
## 2 AACS           41      0.854     1
## 3 AACS           44      0.621     1
## 4 AACS           47      0.223     1
## 5 AACS           50      0.132     1
## 6 AACS           53      0.0614    1
## 7 AACS           56      0.102     1
## 8 AACS           59      0.0378    1
## 9 AACS           63      0.0567    1
## 10 AACS          67      0.0187    1
```

```
## # ... with 145,950 more rows
```

```
h0_stats <- fitH0ModelTppGP(
  in_df = panoFitDf,
  n_samples = 100,
```

```

  l = 8
)

h1_stats <- fitH1ModelTppGP(
  in_df = panoFitDf,
  n_samples = 100,
  l = 8
)

load("../data/pano_eval_models.RData")

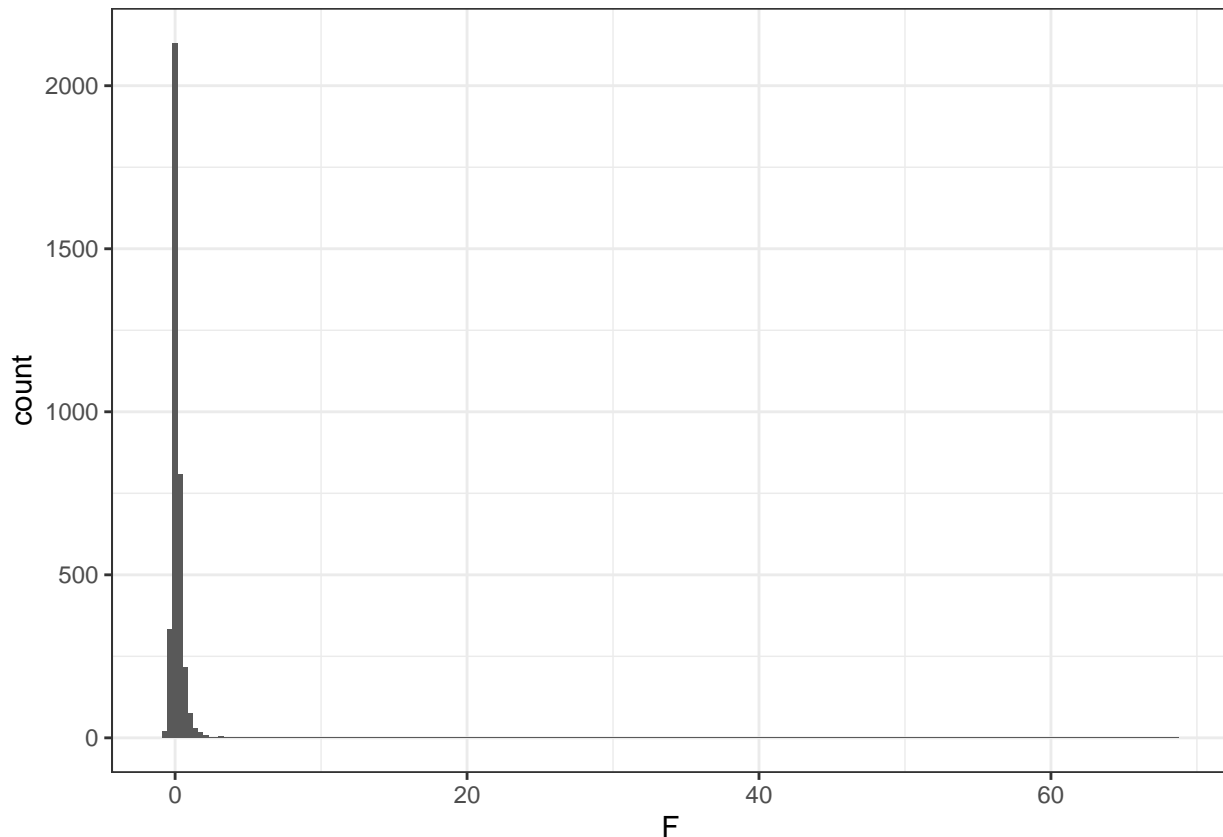
sum_df <- left_join(
  h0_stats, h1_stats,
  by = c("gene_name")) %>%
  mutate(rssH0_rssH1 = rssH0 - rssH1)

head(sum_df %>% mutate(F_stat = rssH0_rssH1/rssH1) %>% arrange(desc(F_stat)))

## # A tibble: 6 x 7
##   gene_name nObsH0  rssH0 nObsH1  rssH1 rssH0_rssH1 F_stat
##   <chr>      <int>  <dbl> <int>  <dbl>      <dbl>  <dbl>
## 1 HDAC1         40 4.87      40 0.0701      4.80    68.5
## 2 HDAC6         40 0.303      40 0.0290      0.274     9.44
## 3 TTC38         40 1.06      40 0.103      0.953     9.29
## 4 HDAC2         40 3.69      40 0.493      3.20     6.49
## 5 VCPIP1        40 0.0657      40 0.0112      0.0544    4.86
## 6 HDAC10        40 0.788      40 0.156      0.632     4.05

ggplot(sum_df, aes(rssH0_rssH1/rssH1)) +
  geom_histogram(bins = 200) +
  xlab('F') +
  theme_bw()

```



```
distr_pars = NPARC:::estimate_df(rss1 = sum_df$rssH1, rssDiff = sum_df$rssH0_rssH1)
sum_df$d1 <- distr_pars$d1
sum_df$d2 <- distr_pars$d2
sum_df$s0_sq = distr_pars$s0_sq
sum_df$rssDiff <- sum_df$rssH0_rssH1/sum_df$s0_sq
sum_df$rss1 = sum_df$rssH1/sum_df$s0_sq

sum_df <- sum_df %>%
  mutate(F_statistic = rssDiff / rss1 * (d2 / d1)) %>%
  mutate(p_val = 1 - pf(F_statistic, df1 = d1, df2 = d2)) %>%
  mutate(p_adj = p.adjust(p_val, method = "BH"))

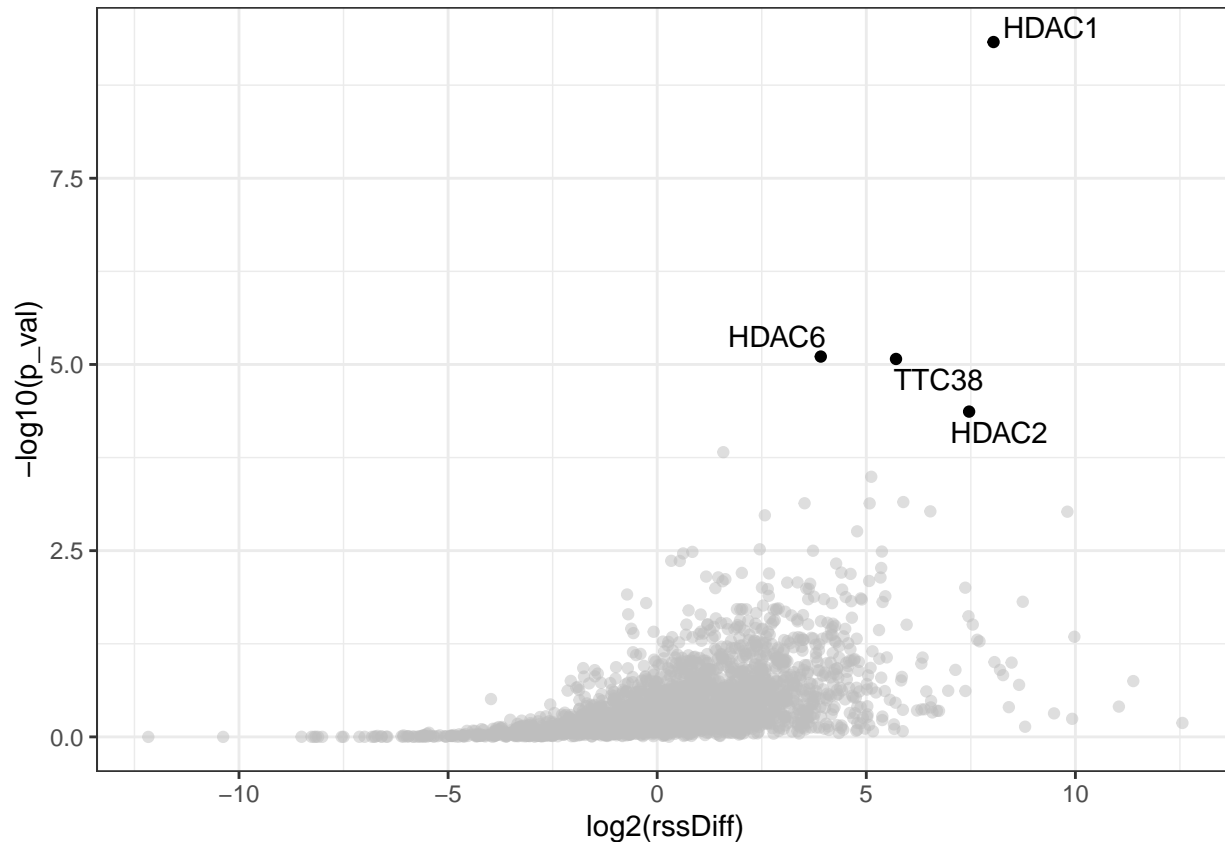
sum_df %>% filter(p_adj < 0.1)
```

```
## # A tibble: 4 x 14
##   gene_name nObsH0 rssH0 nObsH1  rssH1 rssH0_rssH1    d1    d2 s0_sq
##   <chr>      <int> <dbl>  <int>  <dbl>      <dbl> <dbl> <dbl> <dbl>
## 1 HDAC1        40 4.87    40 0.0701      4.80  2.27 10.3 0.0182
## 2 HDAC2        40 3.69    40 0.493      3.20  2.27 10.3 0.0182
## 3 HDAC6        40 0.303   40 0.0290      0.274 2.27 10.3 0.0182
## 4 TTC38        40 1.06    40 0.103      0.953 2.27 10.3 0.0182
## # ... with 5 more variables: rssDiff <dbl>, rss1 <dbl>, F_statistic <dbl>,
## #   p_val <dbl>, p_adj <dbl>
```

```
ggplot(sum_df, aes(log2(rssDiff), -log10(p_val))) +
  geom_point(color = "gray", alpha = 0.5) +
  geom_point(data = filter(sum_df, p_adj < 0.1)) +
  ggrepel::geom_text_repel(aes(label = gene_name),
```

```
data = filter(sum_df, p_adj < 0.1)) +  
theme_bw()
```

```
## Warning in FUN(X[[i]], ...): NaNs produced  
## Warning in FUN(X[[i]], ...): NaNs produced  
## Warning: Removed 1263 rows containing missing values (geom_point).
```



```
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] stats graphics grDevices utils datasets methods base  
##  
## other attached packages:  
## [1] GP4TPP_0.0.99 MASS_7.3-51.4 NPARC_0.99.5 forcats_0.4.0  
## [5] stringr_1.4.0 dplyr_0.8.3 purrr_0.3.3 readr_1.3.1
```

```

## [9] tidyr_1.0.0      tibble_2.1.3      ggplot2_3.2.1     tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
## [1] tidyselect_0.2.5  xfun_0.10         haven_2.1.1
## [4] lattice_0.20-38   colorspace_1.4-1  vctrs_0.2.0
## [7] generics_0.0.2    htmltools_0.4.0   yaml_2.2.0
## [10] utf8_1.1.4        rlang_0.4.1       pillar_1.4.2
## [13] glue_1.3.1        withr_2.1.2       BiocParallel_1.18.1
## [16] modelr_0.1.5      readxl_1.3.1      lifecycle_0.1.0
## [19] munsell_0.5.0     gtable_0.3.0      cellranger_1.1.0
## [22] rvest_0.3.4       evaluate_0.14      labeling_0.3
## [25] knitr_1.25        parallel_3.6.1     fansi_0.4.0
## [28] broom_0.5.2       Rcpp_1.0.2        scales_1.0.0
## [31] backports_1.1.5   jsonlite_1.6       hms_0.5.1
## [34] digest_0.6.22     stringi_1.4.3      ggrepel_0.8.1
## [37] grid_3.6.1        cli_1.1.0          tools_3.6.1
## [40] magrittr_1.5      lazyeval_0.2.2     crayon_1.3.4
## [43] pkgconfig_2.0.3   zeallot_0.1.0      ellipsis_0.3.0
## [46] xml2_1.2.2        lubridate_1.7.4    assertthat_0.2.1
## [49] rmarkdown_1.16    httr_1.4.1         rstudioapi_0.10
## [52] R6_2.4.0          nlme_3.1-141       compiler_3.6.1

```