Downstream analysis followig the image analysis to quanttify Dvl2_mEos codensates

Christian Scheeder [1] ***and Antionia Schubert*** [1]

[1]Division Signalling and Functional Genomics, German Cancer Research Center (DKFZ) Heidelberg

December 9, 2021

Contents

1 Load dependencies

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
library(here)
## here() starts at /Users/c.scheeder/Desktop/remotes/Supp_Schubert_2021/condensate_quantification
library(cowplot)
library(ggpubr)
##
## Attaching package: 'ggpubr'
## The following object is masked from 'package:cowplot':
##
##     get_legend
theme_set(theme_cowplot())
```

2 Preamble

The image analysis was carried out as follows: In a first step cells were identified by segmentation. Nuclei were identified based on a Hoechst staining (DAPI channel) and cell bodies based on a DyLight Phalloidin staining (Cy5 channel). After segmentation condensates inside cells were identified based on intensity-based thresholding (FITC channel) and counted. Condensate counts and cell counts (number of segmented cells) were saved for each image. An exemplary image analysis can be found with the script XXX.R in this repository. Four different cell lines were compared: - HEK cells with a DVL1-mEOS tag - HEK cells with a DVL1-mEOS tag and a EVI k.o. (clone 1) - HEK cells with a DVL1-mEOS tag and a EVI k.o. (clone 3) - HEK cells with a DVL1-mEOS tag and a FZD1 k.o. (clone 1) - HEK wild-type

For each cell line three replicates were carried out (E01,E02,E03). Per replicate multiple 384-well plates were seeded, fixed/stained and imaged. For each replicate representative images were selected. The image-level data (see description above) was further averaged per plate and differences between the cell lines were compared using a box plot and statistical testing with the ggbbubr package (non-parametric Wilcoxon rank-sum test).

3 Load the data after the image analysis

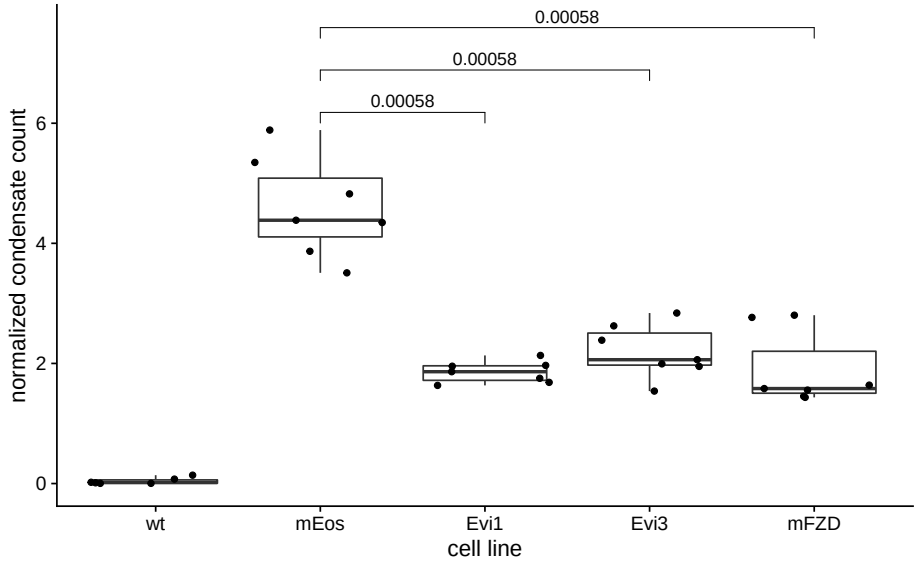
For each replicate the results were saved in a separate data frame.

```
results_path <- here("./raw_data/results_image_analysis/data_frames")
files <- dir(results_path, pattern = "*.rds")

raw_data <- files %>%
```

```
  map(~ read_rds(file.path(results_path, .))) %>%
  reduce(rbind) %>%
    mutate(cell_line=factor(cell_line,
                            levels = c("wt","mEos","Evi1","Evi3","mFZD")))
```

4 Prepare boxplot

```
boxplot <- raw_data %>%
  separate(well_id,c("well","field")) %>%
  mutate(norm_punct_count = punct_count_dapt/(cell_count/100)) %>%
  group_by(experiment,plate_id,cell_line) %>%
  summarise_if(is.numeric,mean) %>%
  ungroup() %>%
    ggplot(aes(x = cell_line,y=norm_punct_count)) +
    geom_boxplot() +
    geom_jitter() +
    stat_compare_means(comparisons = list(c("Evi1","mEos"),
                                          c("Evi3","mEos"),
                                          c("mFZD","mEos")),
                       method = "wilcox.test") +
    ylab("normalized condensate count") +
    xlab("cell line")


boxplot
```




```
ggsave(here("graphics/Figure4F","boxplot_figure4f_raw.pdf"),
       plot=boxplot,
       width = 6,
       height = 4)
```

5 Session info

```
sessionInfo()
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/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] ggpubr_0.4.0     cowplot_1.1.1    here_1.0.1       forcats_0.5.1
##  [5] stringr_1.4.0    dplyr_1.0.7      purrr_0.3.4      readr_2.1.1
##  [9] tidyr_1.1.4      tibble_3.1.6     ggplot2_3.3.5    tidyverse_1.3.1
## [13] BiocStyle_2.22.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7          lubridate_1.8.0     assertthat_0.2.1
##  [4] rprojroot_2.0.2     digest_0.6.29       utf8_1.2.2
##  [7] R6_2.5.1            cellranger_1.1.0    backports_1.4.0
## [10] reprex_2.0.1        evaluate_0.14       httr_1.4.2
## [13] pillar_1.6.4        rlang_0.4.12        readxl_1.3.1
## [16] rstudioapi_0.13     car_3.0-12          rmarkdown_2.11
## [19] labeling_0.4.2      munsell_0.5.0       broom_0.7.10
## [22] compiler_4.1.2      modelr_0.1.8        xfun_0.28
## [25] pkgconfig_2.0.3     htmltools_0.5.2     tidyselect_1.1.1
## [28] bookdown_0.24       fansi_0.5.0         crayon_1.4.2
## [31] tzdb_0.2.0          dbplyr_2.1.1        withr_2.4.3
## [34] grid_4.1.2          jsonlite_1.7.2      gtable_0.3.0
## [37] lifecycle_1.0.1     DBI_1.1.1           magrittr_2.0.1
## [40] scales_1.1.1        carData_3.0-4       cli_3.1.0
## [43] stringi_1.7.6       farver_2.1.0        ggsignif_0.6.3
## [46] fs_1.5.1            xml2_1.3.3          ellipsis_0.3.2
## [49] generics_0.1.1      vctrs_0.3.8         tools_4.1.2
## [52] glue_1.5.1          hms_1.1.1           abind_1.4-5
## [55] fastmap_1.1.0       yaml_2.2.1          colorspace_2.0-2
## [58] BiocManager_1.30.16 rstatix_0.7.0       rvest_1.0.2
## [61] knitr_1.36          haven_2.4.3
```