Image analysis to quantify Dvl2_mEos codensates

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

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

December 10, 2021

Contents

1 Load dependencies 2

2 Preamble 2

3 Load the exmaple images 2

4 Perform image analysis 5

5 Session info 11

1 Load dependencies

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(fig.align="center")
knitr::opts_chunk$set(dev="png")
library(EBImage)
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::combine()   masks EBImage::combine()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::transpose() masks EBImage::transpose()
library(here)
## here() starts at /Users/c.scheeder/Desktop/remotes/Supp_Schubert_2021/condensate_quantification
```

2 Preamble

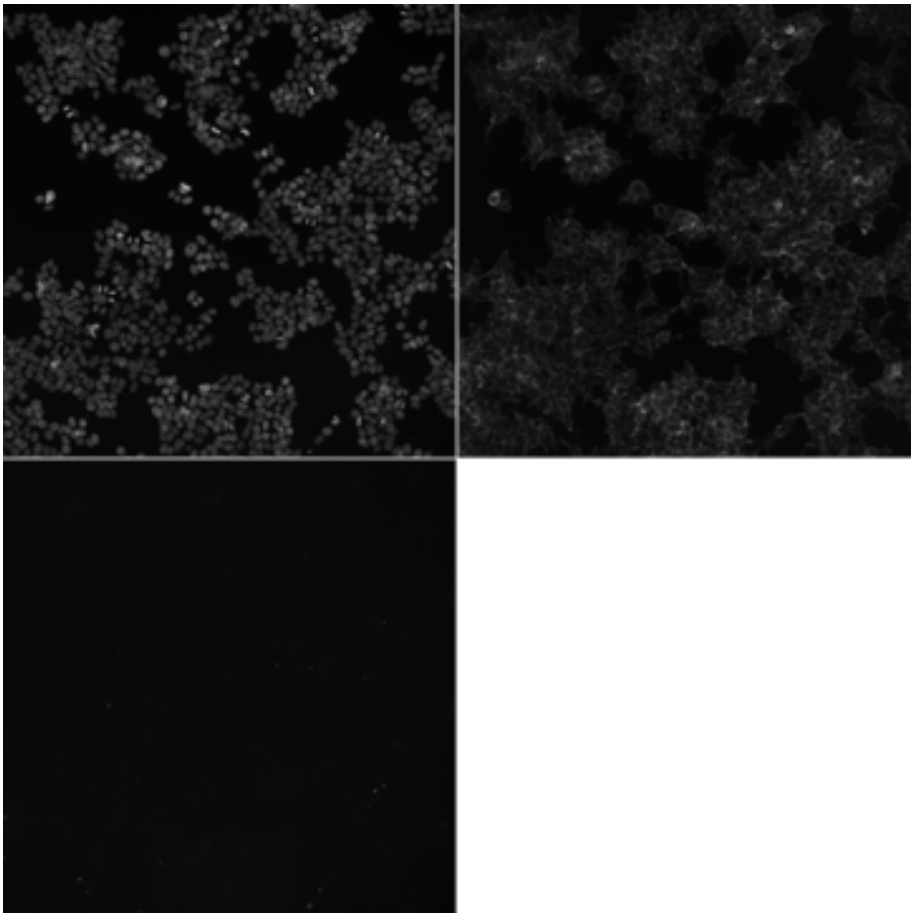
Script to document the image analysis carried out to quantify condensates. In brief, The image analysis was carried out as follows: In a first step cells were identified by segmentation. Nuclei were identified based on a Hoechst staining (DAPI channel) and cell bodies based on a DyLight Phalloidin staining (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. 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
The image analysis shown here is exemplary carried out for two cell lines (DVL1-mEOS and DVL1-mEOS_EVIk.o._clone1).

3 Load the exmaple images

```
source_dir_DVL1_mEOS <-
  here("./raw_data/example_images/HC10984t_HC1098_E01_Evi1_1_2020.10.21.23.32.00")

source_dir_DVL1_Evi1 <-
  here("./raw_data/example_images/HC10984t_HC1098_E01_mEos_2_2020.10.21.15.56.44")

images_DVL1_mEOS <- EBImage::combine(
  readImage(
    file.path(source_dir_DVL1_mEOS,
              list.files(
                source_dir_DVL1_mEOS)[grepl("DAPI",
```

```
                                                list.files(source_dir_DVL1_mEOS))])),
          readImage(
            file.path(source_dir_DVL1_mEOS,
                      list.files(
                        source_dir_DVL1_mEOS)[grepl("Cy5",
                                                list.files(source_dir_DVL1_mEOS))])),
          readImage(
            file.path(source_dir_DVL1_mEOS,
              list.files(
                source_dir_DVL1_mEOS)[grepl("FITC",
                                      list.files(source_dir_DVL1_mEOS))]))
          )
        ## Warning in readTIFF(x, all = all, ...): TIFFReadDirectory: Unknown field with
        ## tag 65400 (0xff78) encountered

        ## Warning in readTIFF(x, all = all, ...): TIFFReadDirectory: Unknown field with
        ## tag 65400 (0xff78) encountered

        ## Warning in readTIFF(x, all = all, ...): TIFFReadDirectory: Unknown field with
        ## tag 65400 (0xff78) encountered


        display(normalize(images_DVL1_mEOS),
                method = "raster",
                all=T)
```

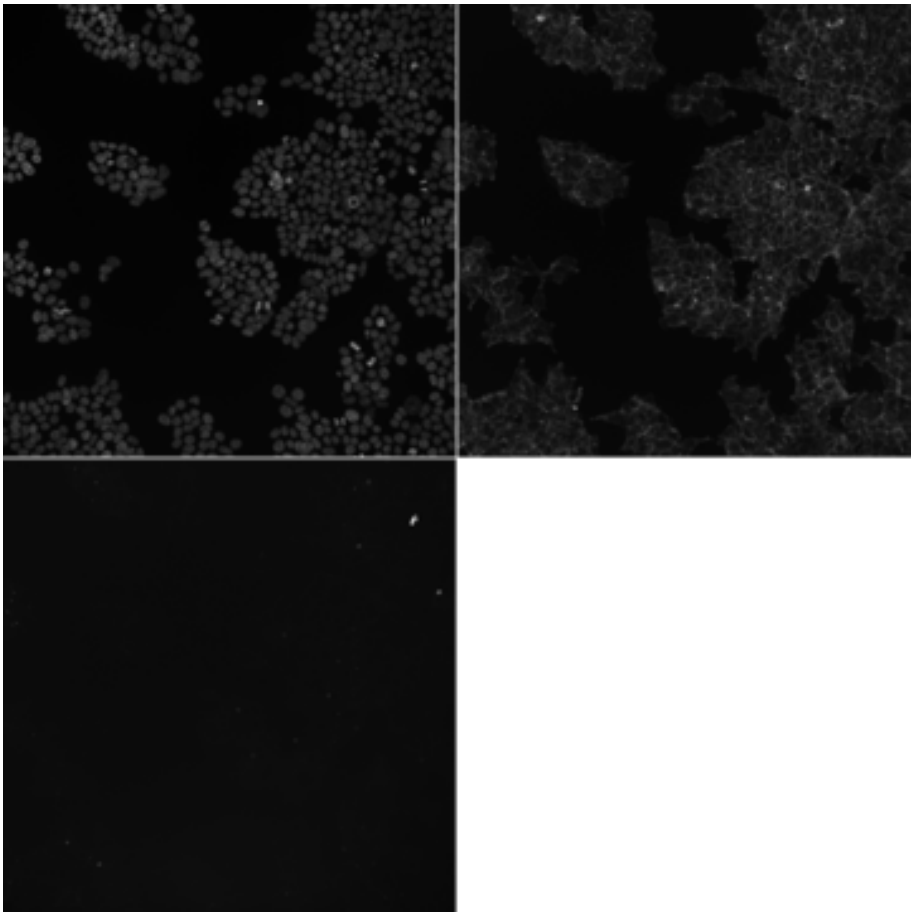
Image analysis to quantify Dvl2_mEos codensates



```
images_DVL1_Evi1 <- EBImage::combine(
  readImage(
    file.path(source_dir_DVL1_Evi1,
              list.files(
                source_dir_DVL1_Evi1)[grepl("DAPI",
                                            list.files(source_dir_DVL1_Evi1))])),
  readImage(
    file.path(source_dir_DVL1_Evi1,
              list.files(
                source_dir_DVL1_Evi1)[grepl("Cy5",
                                            list.files(source_dir_DVL1_Evi1))])),
  readImage(
    file.path(source_dir_DVL1_Evi1,
      list.files(
        source_dir_DVL1_Evi1)[grepl("FITC",
                                    list.files(source_dir_DVL1_Evi1))]))
  )
## Warning in readTIFF(x, all = all, ...): TIFFReadDirectory: Unknown field with
## tag 65400 (0xff78) encountered

## Warning in readTIFF(x, all = all, ...): TIFFReadDirectory: Unknown field with
```

```
## tag 65400 (0xff78) encountered

## Warning in readTIFF(x, all = all, ...): TIFFReadDirectory: Unknown field with
## tag 65400 (0xff78) encountered
```

```
display(normalize(images_DVL1_Evi1),
        method = "raster",
        all=T)
```

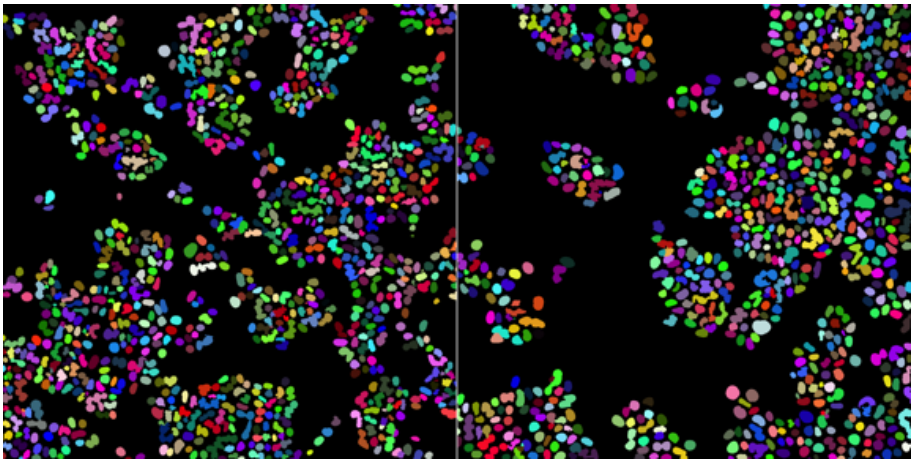


4 Perform image analysis

```
# blur the images
images_DVL1_mEOS_smooth <- images_DVL1_mEOS

images_DVL1_mEOS_smooth[,,1] <- gblur(images_DVL1_mEOS_smooth[,,1],
                                      radius = 51,
                                      sigma = 1)
```

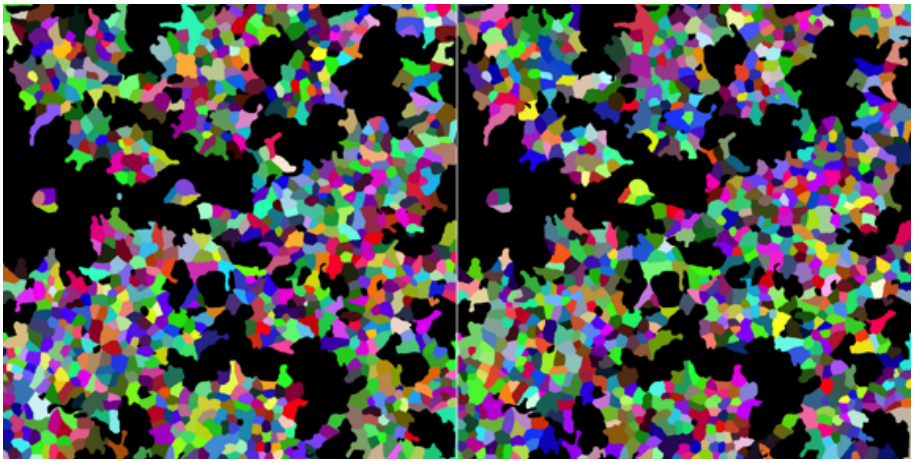
```
images_DVL1_mEOS_smooth[,,2] <- gblur(images_DVL1_mEOS_smooth[,,2],
                                      radius  = 51,
                                      sigma = 4)

images_DVL1_mEOS_smooth[,,3] <- gblur(images_DVL1_mEOS_smooth[,,3],
                                      radius  = 51,
                                      sigma = 4)


images_DVL1_Evi1_smooth <- images_DVL1_Evi1

images_DVL1_Evi1_smooth[,,1] <- gblur(images_DVL1_Evi1[,,1],
                                      radius = 51,
                                      sigma = 1)

images_DVL1_Evi1_smooth[,,2] <- gblur(images_DVL1_Evi1[,,2],
                                      radius  = 51,
                                      sigma = 4)

images_DVL1_Evi1_smooth[,,3] <- gblur(images_DVL1_Evi1[,,3],
                                      radius  = 51,
                                      sigma = 4)



# segment the nuclei
segmentNuclei <- function(inputImage){
  nucleusTresh = thresh(inputImage,
                        w = 20, h = 20,
                        offset = 0.004)

  nucleusTresh = fillHull(opening(nucleusTresh,
                                  kern=makeBrush(9, shape="disc")))

  nucleusFill = fillHull(thresh(inputImage,
                                w = 30, h = 30,
                                offset = 0.001))

  nucleusRegions = propagate(inputImage,
                             seed=bwlabel(nucleusTresh),
                             mask=nucleusFill)

  return(nucleusRegions)

}


nuclei_DVL1_mEOS <- segmentNuclei(inputImage=images_DVL1_mEOS_smooth[,,1])
nuclei_DVL1_Evi1 <- segmentNuclei(inputImage=images_DVL1_Evi1_smooth[,,1])
```

```
display(
  EBImage::combine(
    colorLabels(nuclei_DVL1_mEOS),
    colorLabels(nuclei_DVL1_Evi1)
  ),
  all=T,
  method = "raster"
  )
```



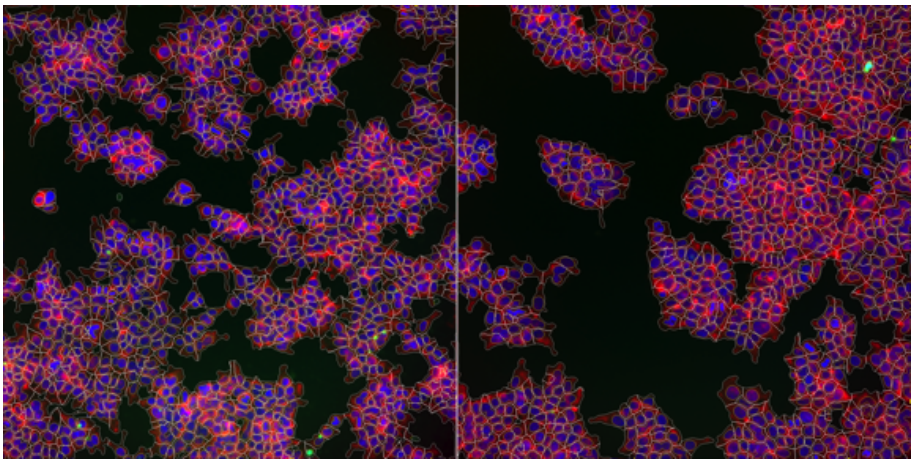
```
# Get outlines of cell bodies
getCellBodies <- function(inputImage,nucleiMask){
   cytoplasmThresh = thresh(inputImage,
                            w = 28,
                            h = 28,
                            offset =  0.001)

    cytoplasmOpening = opening(cytoplasmThresh,
                               kern=makeBrush(9,shape="disc"))

    cytoplasmOpening2 = opening(inputImage > 0.025)


    nucleusRegions2 <- nucleiMask
    nucleusRegions2[nucleiMask != 0] <- 1


    cytoplasmCombined = cytoplasmOpening | cytoplasmOpening2  | nucleusRegions2


    storage.mode(cytoplasmCombined) = "integer"


    cytoplasmRegions = propagate(x = inputImage,
                                 seeds = nucleiMask,
```

```
                                    lambda=1e-04,
                                    mask=cytoplasmCombined)

    cytoplasmRegions = fillHull(cytoplasmRegions)

    return(cytoplasmRegions)
}

cellBodies_DVL1_mEOS <- getCellBodies(inputImage=images_DVL1_mEOS_smooth[,,2],
                                      nucleiMask = nuclei_DVL1_mEOS)

cellBodies_DVL1_Evi1 <- getCellBodies(inputImage=images_DVL1_Evi1_smooth[,,2],
                                      nucleiMask = nuclei_DVL1_Evi1)


display(
  EBImage::combine(
    colorLabels(cellBodies_DVL1_mEOS),
    colorLabels(cellBodies_DVL1_mEOS)
  ),
  all=T,
  method = "raster"
  )
```



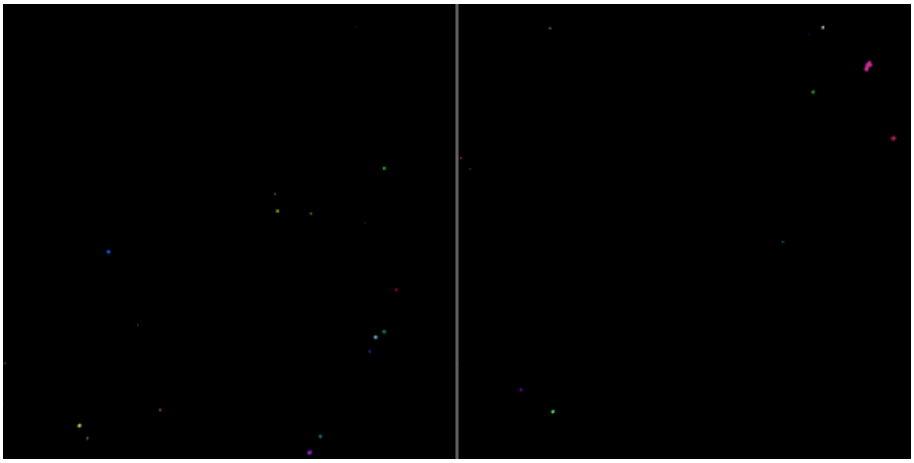
```
# display the fluorescence image as normalized and pseudo-colored RGB
#   with segmentation borders painted
ImgColor_DVL1_mEOS = rgbImage(2*normalize(images_DVL1_mEOS_smooth[,,2]),
                    2*normalize(images_DVL1_mEOS_smooth[,,3]),
                    2*normalize(images_DVL1_mEOS_smooth[,,1]))

ImgOut_DVL1_mEOS = paintObjects(cellBodies_DVL1_mEOS,
                       paintObjects(nuclei_DVL1_mEOS,
                                    ImgColor_DVL1_mEOS,
                                    col='yellow'),
```

```
                          col='white')


ImgColor_DVL1_Evi1 = rgbImage(2*normalize(images_DVL1_Evi1_smooth[,,2]),
                    2*normalize(images_DVL1_Evi1_smooth[,,3]),
                    2*normalize(images_DVL1_Evi1_smooth[,,1]))

ImgOut_DVL_Evi1 = paintObjects(cellBodies_DVL1_Evi1,
                       paintObjects(nuclei_DVL1_Evi1,
                                    ImgColor_DVL1_Evi1,
                                    col='yellow'),
                       col='white')

display(
  EBImage::combine(
    ImgOut_DVL1_mEOS,
    ImgOut_DVL_Evi1
  ),
  all=T,
  method = "raster"
  )
```



```
# Detect mEos condensates in images by adaptive thresholding
getCondensates <- function(inputImage,cellBodyMask){
  condensates_adaptTresh <- thresh(inputImage,
                                   w = 20, h = 20,
                                   offset = 0.006)

  condensates_adaptTresh[cellBodyMask == 0] <- 0

  condensates_adaptTresh_out <- colorLabels(bwlabel(condensates_adaptTresh))

  return(condensates_adaptTresh_out)
}
```

```
DVL1_mEOS_conTresh <- getCondensates(inputImage = images_DVL1_mEOS_smooth[,,3],
                                     cellBodyMask = cellBodies_DVL1_mEOS)

DVL1_Evi1_conTresh <- getCondensates(inputImage = images_DVL1_Evi1_smooth[,,3],
                                     cellBodyMask = cellBodies_DVL1_Evi1)


display(
  EBImage::combine(
    DVL1_mEOS_conTresh,
    DVL1_Evi1_conTresh
  ),
  method = 'raster',
  all=T
)
```



```
# the number of condensates corresponds to the number of objects in
# the binary image with labels

print(paste0(
  "Number of detected condensates in the image with DVL1-mEOS cells: ",
  length(unique(bwlabel(DVL1_mEOS_conTresh)[bwlabel(DVL1_mEOS_conTresh)!=0])),
  " in ",
  length(unique(cellBodies_DVL1_mEOS[cellBodies_DVL1_mEOS!=0])),
  " cells."
  )
)
## [1] "Number of detected condensates in the image with DVL1-mEOS cells: 20 in 973 cells."

print(paste0(
  "Number of detected condensates in the image with DVL1-mEOS Evi1 k.o. cells: ",
  length(unique(bwlabel(DVL1_Evi1_conTresh)[bwlabel(DVL1_Evi1_conTresh)!=0])),
  " in ",
```

```
  length(unique(cellBodies_DVL1_Evi1[cellBodies_DVL1_Evi1!=0])),
  " cells."
  )
)
## [1] "Number of detected condensates in the image with DVL1-mEOS Evi1 k.o. cells: 11 in 779 cells."
```

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] here_1.0.1       forcats_0.5.1    stringr_1.4.0    dplyr_1.0.7
##  [5] purrr_0.3.4      readr_2.1.1      tidyr_1.1.4      tibble_3.1.6
##  [9] ggplot2_3.3.5    tidyverse_1.3.1  EBImage_4.36.0   BiocStyle_2.22.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7          locfit_1.5-9.4      lubridate_1.8.0
##  [4] lattice_0.20-45     fftwtools_0.9-11    png_0.1-7
##  [7] rprojroot_2.0.2     assertthat_0.2.1    digest_0.6.29
## [10] utf8_1.2.2          R6_2.5.1            tiff_0.1-10
## [13] cellranger_1.1.0    backports_1.4.0     reprex_2.0.1
## [16] evaluate_0.14       httr_1.4.2          pillar_1.6.4
## [19] rlang_0.4.12        readxl_1.3.1        rstudioapi_0.13
## [22] magick_2.7.3        rmarkdown_2.11      htmlwidgets_1.5.4
## [25] RCurl_1.98-1.5      munsell_0.5.0       broom_0.7.10
## [28] compiler_4.1.2      modelr_0.1.8        xfun_0.28
## [31] pkgconfig_2.0.3     BiocGenerics_0.40.0 htmltools_0.5.2
## [34] tidyselect_1.1.1    bookdown_0.24       fansi_0.5.0
## [37] withr_2.4.3         crayon_1.4.2        tzdb_0.2.0
## [40] dbplyr_2.1.1        bitops_1.0-7        grid_4.1.2
## [43] jsonlite_1.7.2      gtable_0.3.0        lifecycle_1.0.1
## [46] DBI_1.1.1           magrittr_2.0.1      scales_1.1.1
## [49] cli_3.1.0           stringi_1.7.6       fs_1.5.1
## [52] xml2_1.3.3          ellipsis_0.3.2      generics_0.1.1
## [55] vctrs_0.3.8         tools_4.1.2         glue_1.5.1
## [58] hms_1.1.1           jpeg_0.1-9          abind_1.4-5
```

```
## [61] fastmap_1.1.0       yaml_2.2.1          colorspace_2.0-2
## [64] BiocManager_1.30.16 rvest_1.0.2         knitr_1.36
## [67] haven_2.4.3
```