Zebrafish Cell Landscape - MafB

Domien

2025-09-19 11:44:47 +0200

Contents

Introduction **2**

Load packages **2**

Download Zebrafish Cell Landscape Seurat object from Figshare server **2**

Load Zebrafish Cell Landscape Seurat object **2**

Reconstruct Zebrafish Cell Landscape Seurat object **2**

Visualize clusters **3**

Calculate correlation between mafb expression and Mac signature score **4**

Calculate correlation between mafb expression and MafB target gene signature score **9**

Introduction

To investigate the conservedness of MafB to imprint Mac identity across species we quantified the correlation between Mafb expression and the expression of macrophage sigature genes and MafB target genes in single cell atlases of mouse (Tabula Muris Senis), human (Tabula Sapiens), lemur (Tabula Microcebus), pig (Pig Cell Atlas), African clawed frog (Xenopus Cell Landscape) and zebrafish (Zebrafish Cell Landscape, ZCL).

Load packages

```
suppressMessages({
    library(SeuratObject)
    library(Seurat)
    library(readr)
    library(ggplot2)
    library(ggrastr)
    library(readxl)
    library(readr)
})
```

Download Zebrafish Cell Landscape Seurat object from Figshare server

```
rdata: https://figshare.com/s/1ab3c6d7648d12247eb2?file=36402453
cellInfo: https://figshare.com/s/1ab3c6d7648d12247eb2?file=36400452
```

Load Zebrafish Cell Landscape Seurat object

```
load("ZCDL.rdata")
ZCL <- pbmc
rm(pbmc)
```

Reconstruct Zebrafish Cell Landscape Seurat object

```
ZCDL_cellinfo <- read_csv("ZCDL_cellinfo.csv")
```

```
## New names:
## Rows: 1082680 Columns: 8
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (4): barcodes, stage, cell_type, cell_lineage dbl (4): ...1, cluster, tsne_x,
## tsne_y
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘‘ -> ‘...1‘
```

```
ZCDL_cellinfo$...1 <- NULL

rownames(ZCDL_cellinfo) <- ZCDL_cellinfo$barcodes
```

```
## Warning: Setting row names on a tibble is deprecated.
```

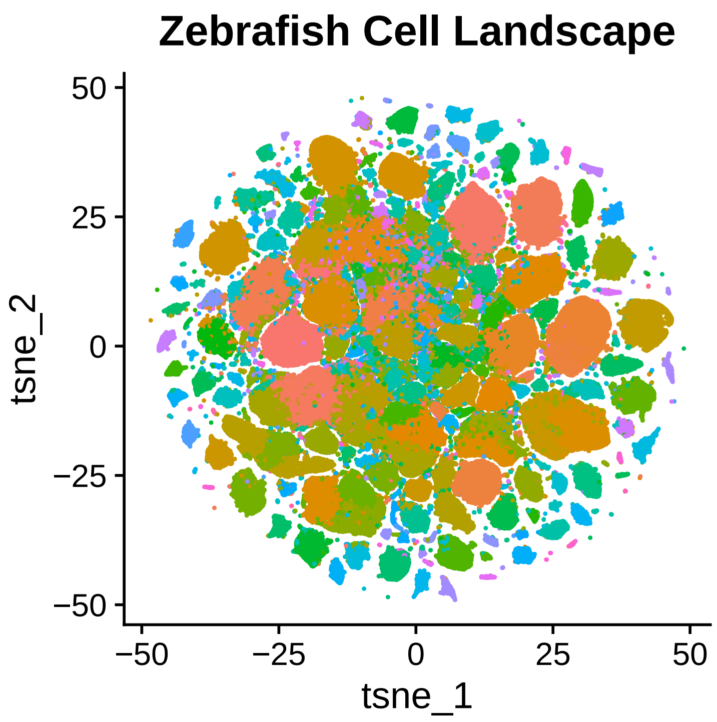
```
ZCL[["CellName"]] <- colnames(ZCL)

ZCL <- subset(ZCL, subset = CellName %in% rownames(ZCDL_cellinfo))

ZCL <- AddMetaData(ZCL, ZCDL_cellinfo)

mat <- ZCDL_cellinfo[, c(6, 7)]
rownames(mat) <- rownames(ZCDL_cellinfo)
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
colnames(mat) <- c("tsne_1", "tsne_2")

mat <- as.matrix(mat)

ZCL[["tsne"]] <- CreateDimReducObject(embeddings = mat, key = "tsne_",
    assay = "RNA")

ZCL <- NormalizeData(ZCL)
```

Visualize clusters

```
p1 <- DimPlot(ZCL, group.by = "cluster", raster = F) + theme(legend.position = "none") +
    ggtitle("Zebrafish Cell Landscape")

rasterize(p1, layers = "Point", dpi = 1200)
```

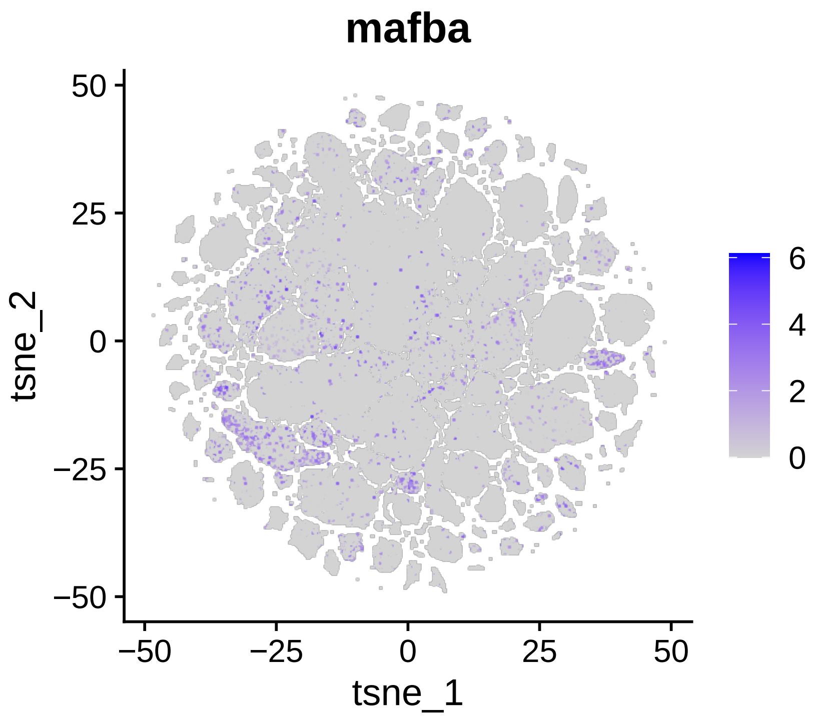


Calculate correlation between mafb expression and Mac signature score

Note: zebrafish have two MafB paralogs: mafba and mafbb

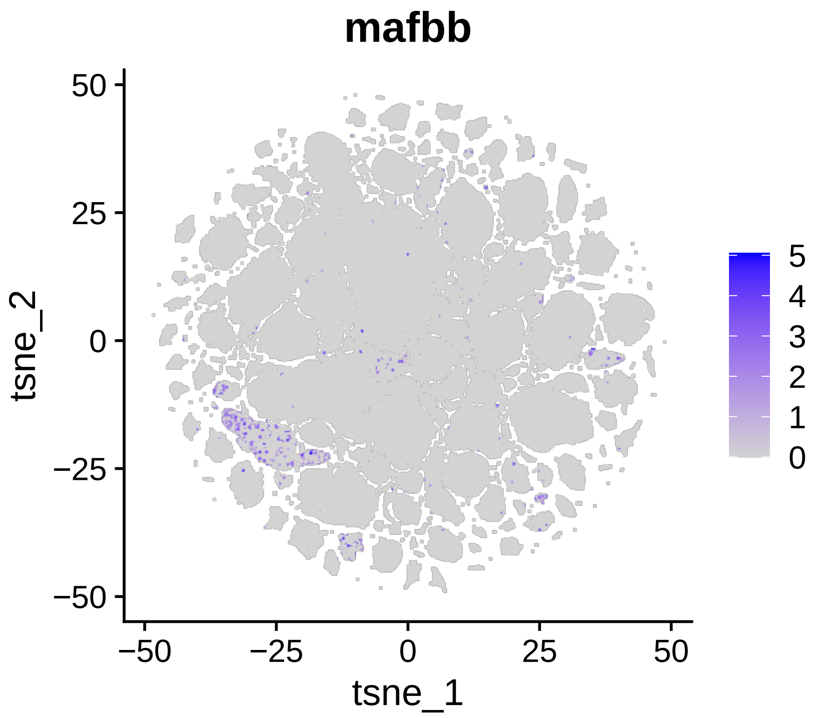
```
FeaturePlot(ZCL, features = "mafba", raster = T)
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set ‘raster=FALSE‘
```



```
FeaturePlot(ZCL, features = "mafbb", raster = T)
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set ‘raster=FALSE‘
```

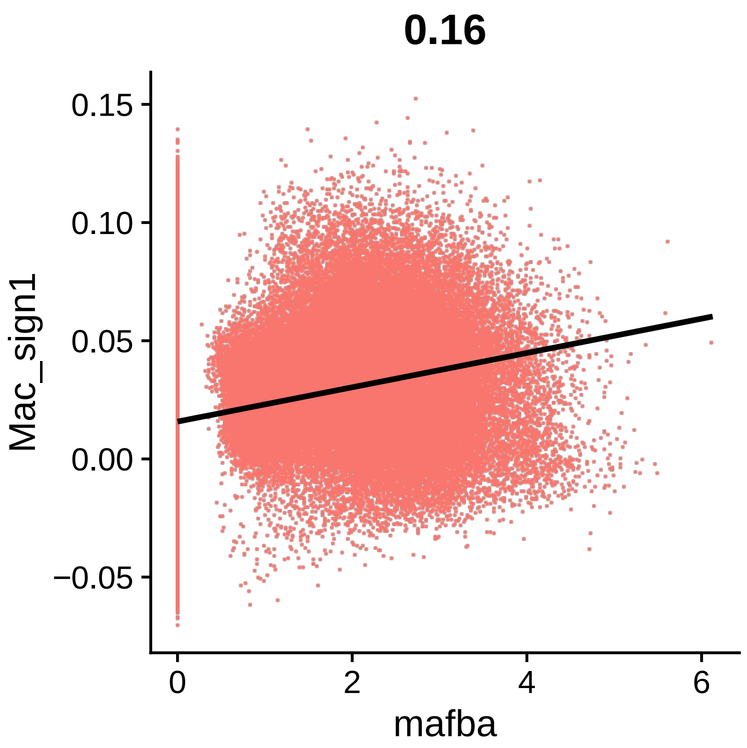


```r
Mac_sign_danio <- read_excel("Mac_sign_danio.xlsx")

Mac_sign_danio <- Mac_sign_danio$ortholog_name

Mac_sign_danio <- intersect(Mac_sign_danio, rownames(ZCL))


ZCL <- AddModuleScore(ZCL, features = list(c(Mac_sign_danio)),
    name = "Mac_sign")
```

```r
FeatureScatter(ZCL, feature1 = "mafba", feature2 = "Mac_sign1",
    raster = T) + geom_smooth(method = "lm", colour = "black") +
    theme(legend.position = "none")
```

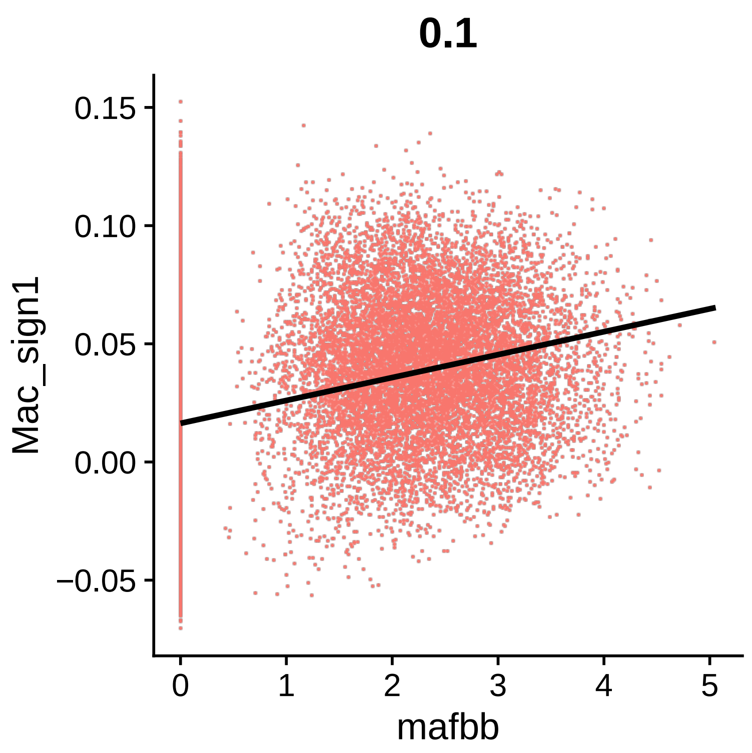
```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set `raster=FALSE`
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
FeatureScatter(ZCL, feature1 = "mafbb", feature2 = "Mac_sign1",
    raster = T) + geom_smooth(method = "lm", colour = "black") +
    theme(legend.position = "none")
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set `raster=FALSE`
## `geom_smooth()` using formula = 'y ~ x'
```



```
corr <- FetchData(ZCL, vars = "mafba")
corr <- cbind(corr, FetchData(ZCL, vars = "mafbb"))
corr <- cbind(corr, FetchData(ZCL, vars = "Mac_sign1"))

cor.test(corr$mafba, corr$Mac_sign1, method = "pearson")
```

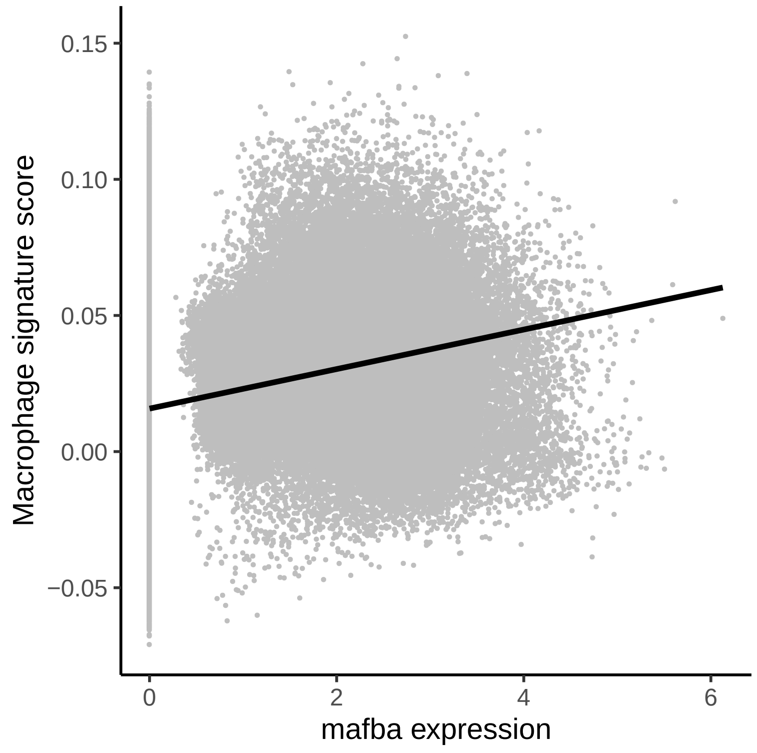
```
##
##  Pearson's product-moment correlation
##
## data:  corr$mafba and corr$Mac_sign1
## t = 164.72, df = 1082678, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1545193 0.1581945
## sample estimates:
##       cor
## 0.1563575
```

```
cor.test(corr$mafbb, corr$Mac_sign1, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  corr$mafbb and corr$Mac_sign1
## t = 104.57, df = 1082678, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.09812554 0.10185515
## sample estimates:
##       cor
## 0.0999907
```

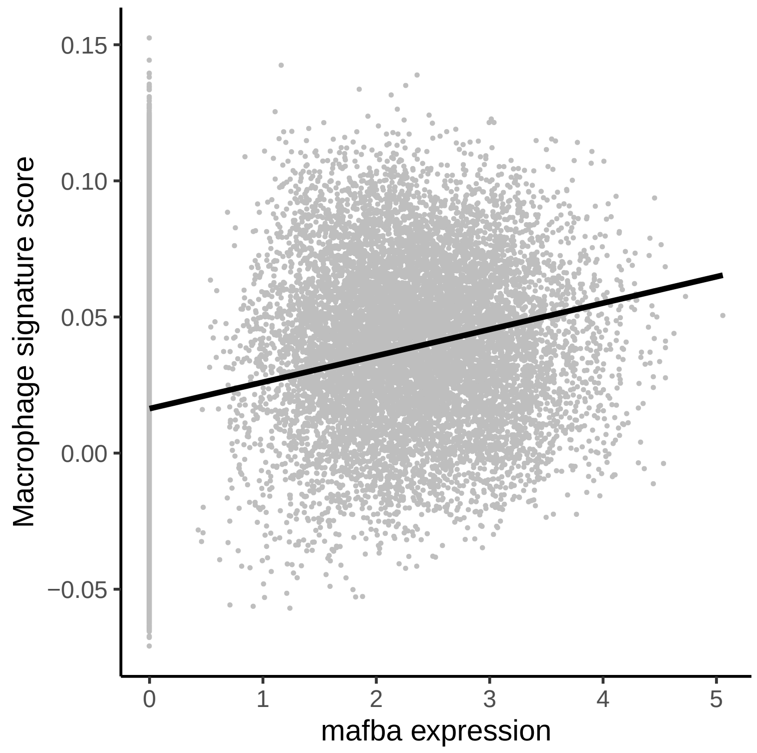
```
p2 <- ggplot(corr, aes(x = mafba, y = Mac_sign1)) + geom_point(size = 0.1,
    colour = "grey") + geom_smooth(method = "lm", colour = "black") +
    xlab("mafba expression") + ylab("Macrophage signature score") +
    theme_classic()

rasterize(p2, layers = "Point", dpi = 600)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
p3 <- ggplot(corr, aes(x = mafbb, y = Mac_sign1)) + geom_point(size = 0.1,
    colour = "grey") + geom_smooth(method = "lm", colour = "black") +
    xlab("mafba expression") + ylab("Macrophage signature score") +
    theme_classic()

rasterize(p3, layers = "Point", dpi = 600)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



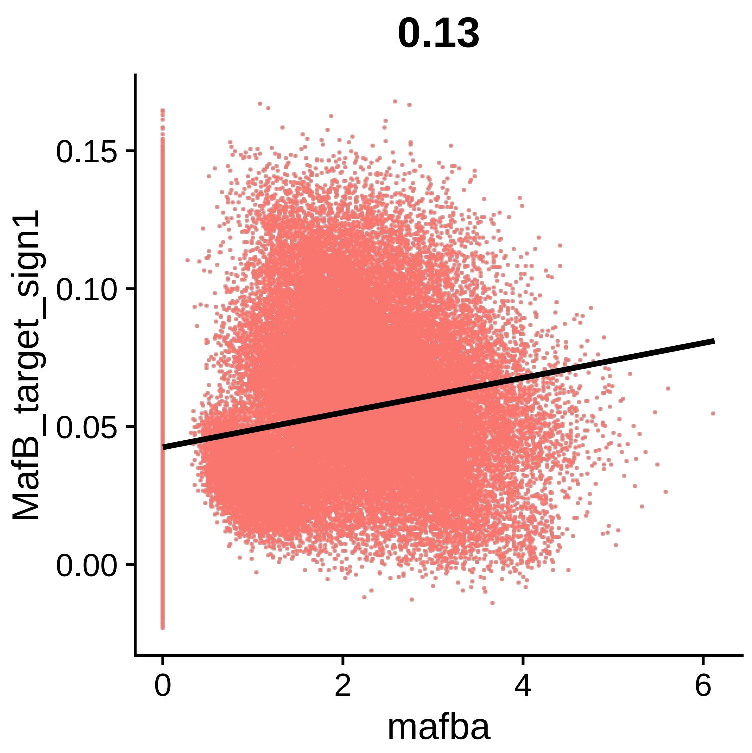
Calculate correlation between mafb expression and MafB target gene signature score

```
Conserved_MafB_target_genes <- read.csv("Conserved_MafB_target_genes_ZebraFish.csv")

Conserved_MafB_target_genes <- Conserved_MafB_target_genes$ortholog_name

Conserved_MafB_target_genes <- intersect(Conserved_MafB_target_genes,
    rownames(ZCL))


ZCL <- AddModuleScore(ZCL, features = list(c(Conserved_MafB_target_genes)),
    name = "MafB_target_sign")
```

```
FeatureScatter(ZCL, feature1 = "mafba", feature2 = "MafB_target_sign1",
    raster = T) + geom_smooth(method = "lm", colour = "black") +
    theme(legend.position = "none")
```

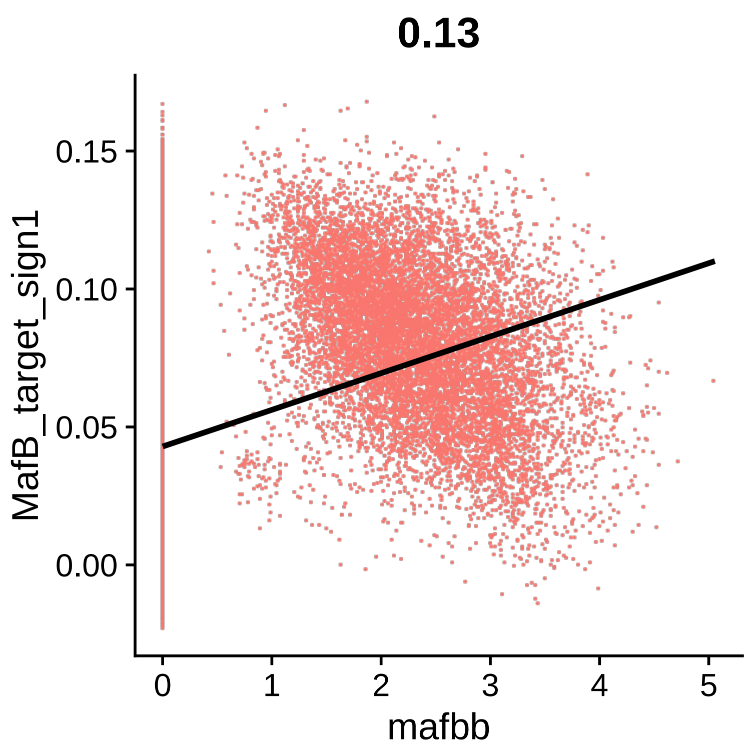
```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set ‘raster=FALSE‘
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```
FeatureScatter(ZCL, feature1 = "mafbb", feature2 = "MafB_target_sign1",
    raster = T) + geom_smooth(method = "lm", colour = "black") +
    theme(legend.position = "none")
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set `raster=FALSE`
## `geom_smooth()` using formula = 'y ~ x'
```



```r
corr <- cbind(corr, FetchData(ZCL, vars = "MafB_target_sign1"))

cor.test(corr$mafba, corr$MafB_target_sign1, method = "pearson")
```

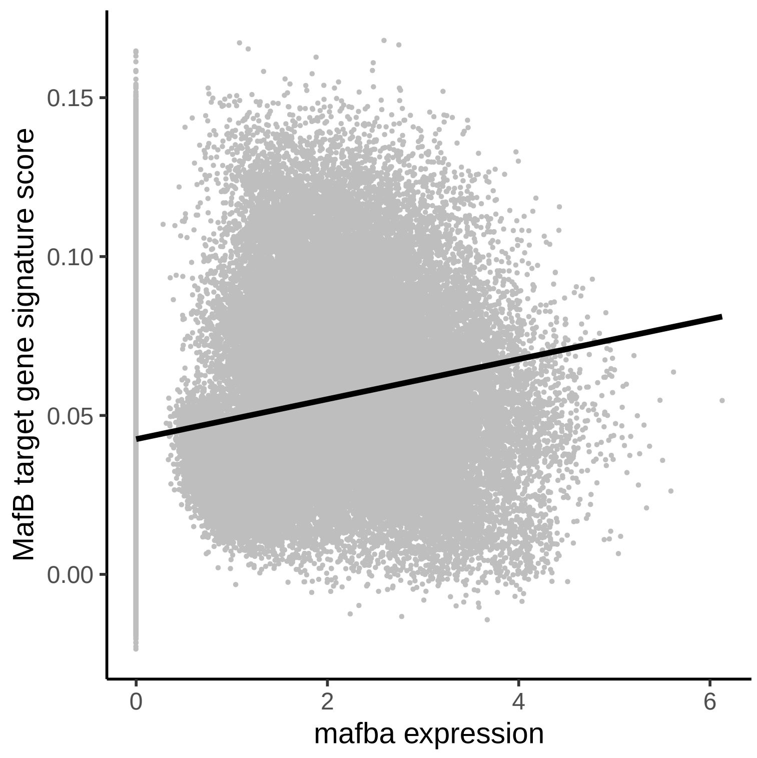
```
##
##  Pearson's product-moment correlation
##
## data:  corr$mafba and corr$MafB_target_sign1
## t = 134.23, df = 1082678, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1260923 0.1297979
## sample estimates:
##       cor
## 0.1279455
```

```r
cor.test(corr$mafbb, corr$MafB_target_sign1, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  corr$mafbb and corr$MafB_target_sign1
## t = 135.7, df = 1082678, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1274711 0.1311754
## sample estimates:
##       cor
## 0.1293237
```

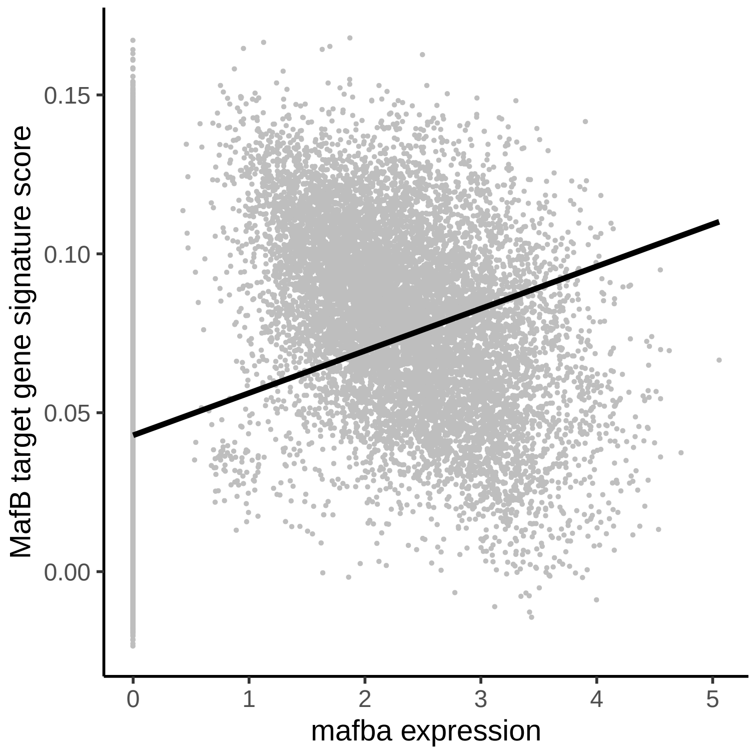
```
p4 <- ggplot(corr, aes(x = mafba, y = MafB_target_sign1)) + geom_point(size = 0.1,
    colour = "grey") + geom_smooth(method = "lm", colour = "black") +
    xlab("mafba expression") + ylab("MafB target gene signature score") +
    theme_classic()

rasterize(p4, layers = "Point", dpi = 600)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
p5 <- ggplot(corr, aes(x = mafbb, y = MafB_target_sign1)) + geom_point(size = 0.1,
    colour = "grey") + geom_smooth(method = "lm", colour = "black") +
    xlab("mafba expression") + ylab("MafB target gene signature score") +
    theme_classic()

rasterize(p5, layers = "Point", dpi = 600)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 20.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=fr_BE.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=fr_BE.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=fr_BE.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=fr_BE.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Brussels
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] readxl_1.4.3       ggrastr_1.0.2      ggplot2_3.4.4      readr_2.1.5
## [5] Seurat_4.3.0       SeuratObject_4.1.3 sp_2.1-4
##
## loaded via a namespace (and not attached):
```

```
##   [1] RColorBrewer_1.1-3     rstudioapi_0.16.0      jsonlite_1.8.9
##   [4] magrittr_2.0.3         spatstat.utils_3.1-0   ggbeeswarm_0.7.2
##   [7] farver_2.1.2           rmarkdown_2.28         vctrs_0.6.5
##  [10] ROCR_1.0-11            Cairo_1.6-2            spatstat.explore_3.3-2
##  [13] htmltools_0.5.8.1      cellranger_1.1.0       sctransform_0.4.1
##  [16] parallelly_1.26.0      KernSmooth_2.23-24     htmlwidgets_1.5.3
##  [19] ica_1.0-2              plyr_1.8.6             plotly_4.10.4
##  [22] zoo_1.8-9              igraph_1.2.6           mime_0.11
##  [25] lifecycle_1.0.4        pkgconfig_2.0.3        Matrix_1.6-1.1
##  [28] R6_2.5.1               fastmap_1.2.0          fitdistrplus_1.1-5
##  [31] future_1.21.0          shiny_1.9.1            digest_0.6.37
##  [34] colorspace_2.1-1       patchwork_1.1.1        tensor_1.5
##  [37] irlba_2.3.5.1          labeling_0.4.3         progressr_0.14.0
##  [40] fansi_1.0.6            spatstat.sparse_3.1-0  mgcv_1.9-1
##  [43] httr_1.4.7             polyclip_1.10-0        abind_1.4-5
##  [46] compiler_4.4.1         bit64_4.5.2            withr_3.0.1
##  [49] highr_0.11             MASS_7.3-61            tools_4.4.1
##  [52] vipor_0.4.7            lmtest_0.9-38          beeswarm_0.4.0
##  [55] httpuv_1.6.1           future.apply_1.7.0     goftest_1.2-2
##  [58] glue_1.7.0             nlme_3.1-165           promises_1.2.0.1
##  [61] grid_4.4.1             Rtsne_0.15             cluster_2.1.6
##  [64] reshape2_1.4.4         generics_0.1.0         gtable_0.3.5
##  [67] spatstat.data_3.1-2    tzdb_0.4.0             tidyr_1.3.1
##  [70] data.table_1.14.0      hms_1.1.3              utf8_1.2.4
##  [73] spatstat.geom_3.3-3    RcppAnnoy_0.0.18       ggrepel_0.9.6
##  [76] RANN_2.6.1             pillar_1.9.0           stringr_1.5.1
##  [79] vroom_1.6.5            later_1.2.0            splines_4.4.1
##  [82] dplyr_1.1.4            lattice_0.22-5         survival_3.7-0
##  [85] bit_4.5.0              deldir_2.0-4           tidyselect_1.2.1
##  [88] miniUI_0.1.1.1         pbapply_1.4-3          knitr_1.48
##  [91] gridExtra_2.3          scattermore_0.7        xfun_0.47
##  [94] matrixStats_1.4.1      stringi_1.6.2          lazyeval_0.2.2
##  [97] yaml_2.2.1             evaluate_1.0.0         codetools_0.2-19
## [100] tibble_3.2.1           cli_3.6.3              uwot_0.2.2
## [103] xtable_1.8-4           reticulate_1.39.0      munsell_0.5.1
## [106] Rcpp_1.0.13            globals_0.14.0         spatstat.random_3.3-2
## [109] png_0.1-8              spatstat.univar_3.0-1  parallel_4.4.1
## [112] listenv_0.8.0          viridisLite_0.4.2      scales_1.3.0
## [115] ggridges_0.5.3         leiden_0.3.8           purrr_1.0.2
## [118] crayon_1.4.1           rlang_1.1.4            cowplot_1.1.1
## [121] formatR_1.14
```