Xenopus Cell Landscape - MafB

Domien

2024-12-16 14:24:14 +0100

Contents

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(reticulate)
    library(sceasy)
    library(ggplot2)
    library(ggrastr)
    library(readr)
})
```

Download meta.data, counts and cellInfo from Figshare

```
Extract meta.data from scanpy object:
https://figshare.com/articles/dataset/Cell_Atlas_of_the_Xenopus_Laevis_at_Single-Cell_Resolution/191528

Donload counts from:
https://figshare.com/articles/dataset/Cell_Atlas_of_the_Xenopus_Laevis_at_Single-Cell_Resolution/191528

cellInfo:
https://figshare.com/articles/dataset/Cell_Atlas_of_the_Xenopus_Laevis_at_Single-Cell_Resolution/191528
```

Recontruct Seurat Object

```
XCA <- CreateSeuratObject(counts = counts, assay = "RNA", min.cells = 3,
    meta.data = meta.data)

cellInfo$...1 <- NULL

rownames(cellInfo) <- cellInfo$cellID

mat <- cellInfo[, c(1, 2)]
rownames(mat) <- rownames(cellInfo)

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

mat <- as.matrix(mat)
```

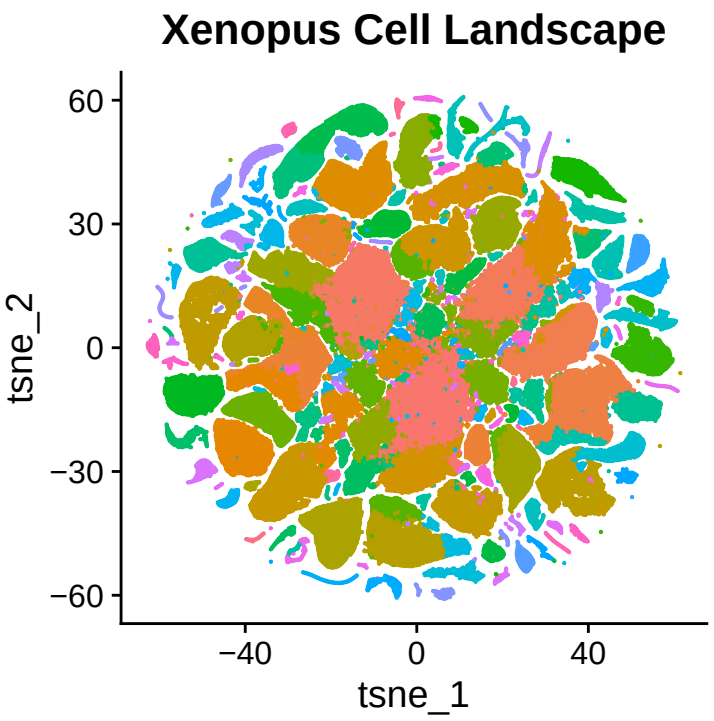
```
XCA[["tsne"]] <- CreateDimReducObject(embeddings = mat, key = "tsne_",
    assay = "RNA")

XCA <- NormalizeData(XCA)

saveRDS(XCA, file = "XCA_full.rds")
```

Load Xenopus Cell Landscape Seurat object

```
XCA <- readRDS("XCA_full.rds")
```

Visualize clusters

```
p1 <- DimPlot(XCA, group.by = "leiden", raster = F) + theme(legend.position = "none") +
    ggtitle("Xenopus Cell Landscape")

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

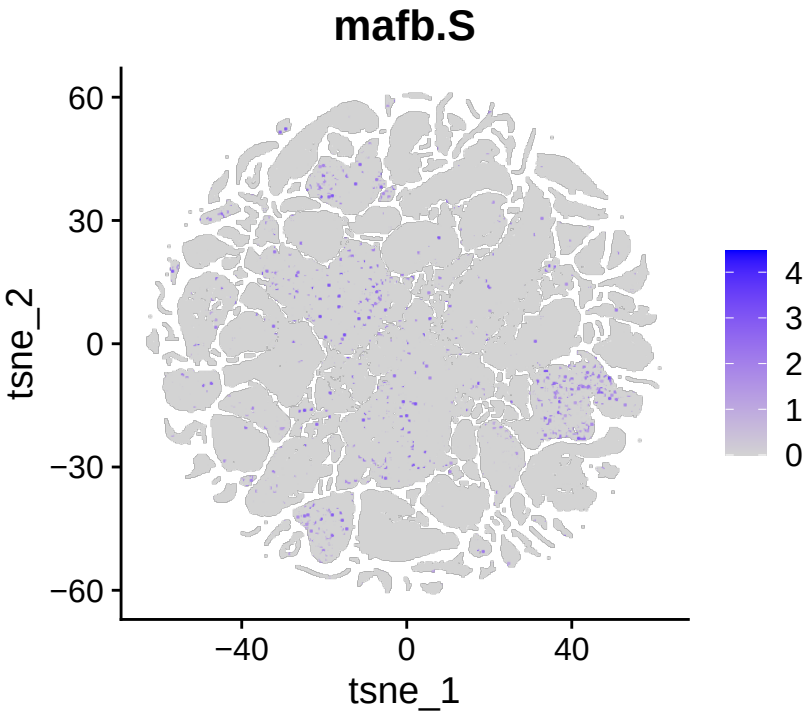


Calculate correlation between mafb expression and Mac signature score

Note: Xenupus laevis has two MafB paralogs: mafb.S and mafb.L

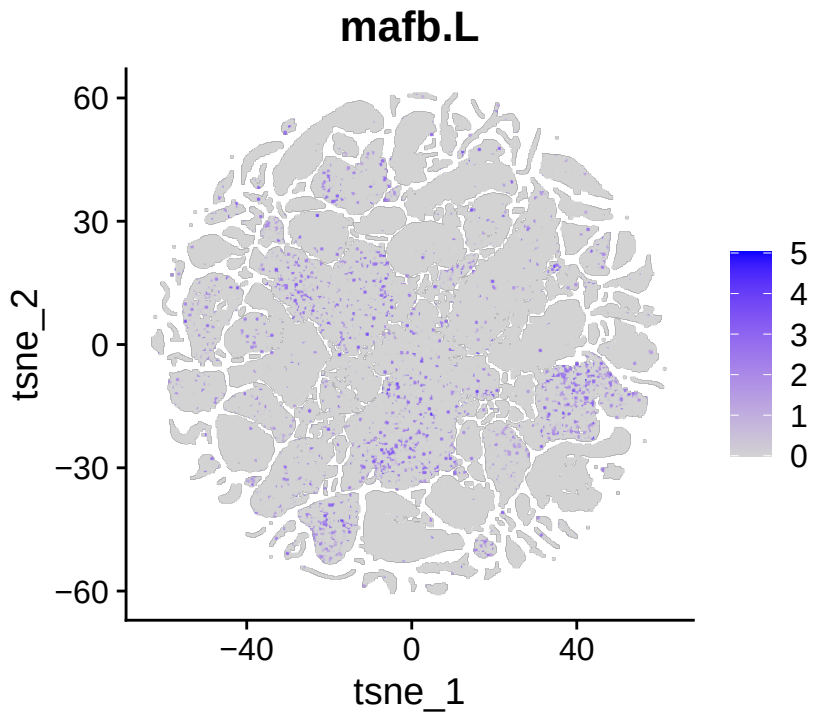
```
FeaturePlot(XCA, features = "mafb.S", raster = T)
```

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




```
FeaturePlot(XCA, features = "mafb.L", raster = T)
```

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




```
Mac_sign <- read_csv("Mac_sign_Xlaevis.csv")
```

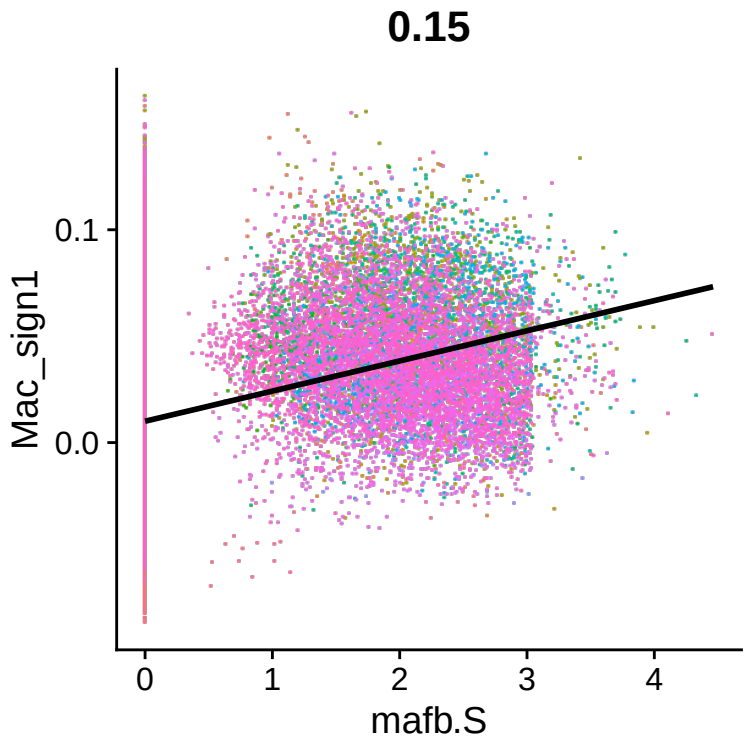
```
## New names:
## Rows: 1166 Columns: 14
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (11): Gene1ID, Gene1Symbol, Gene1SpeciesTaxonID, Gene1SpeciesName, Gene2... dbl
## (3): ...1, AlgorithmsMatch, OutOfAlgorithms
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
Mac_sign <- Mac_sign$Gene2Symbol

Mac_sign <- intersect(Mac_sign, rownames(XCA))

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

```
FeatureScatter(XCA, feature1 = "mafb.S", 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`
```

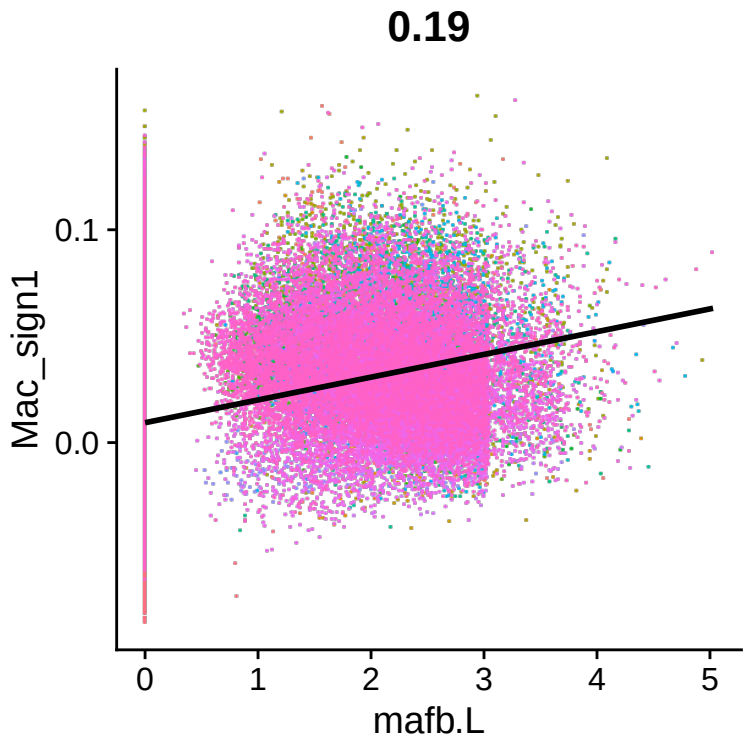
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
FeatureScatter(XCA, feature1 = "mafb.L", 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(XCA, vars = "mafb.S")
corr <- cbind(corr, FetchData(XCA, vars = "mafb.L"))
corr <- cbind(corr, FetchData(XCA, vars = "Mac_sign1"))

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

```
##
##  Pearson's product-moment correlation
##
## data:  corr$mafb.S and corr$Mac_sign1
## t = 105.27, df = 501356, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1443415 0.1497579
## sample estimates:
##       cor
## 0.1470508
```

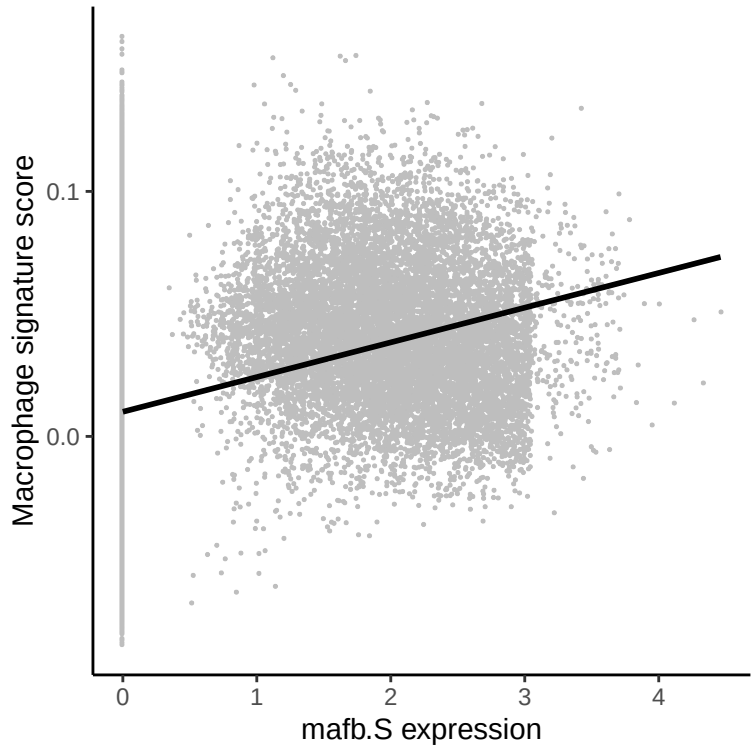
```
cor.test(corr$mafb.L, corr$Mac_sign1, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  corr$mafb.L and corr$Mac_sign1
## t = 135.39, df = 501356, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1851375 0.1904784
```

```
## sample estimates:
##       cor
## 0.1878094
```

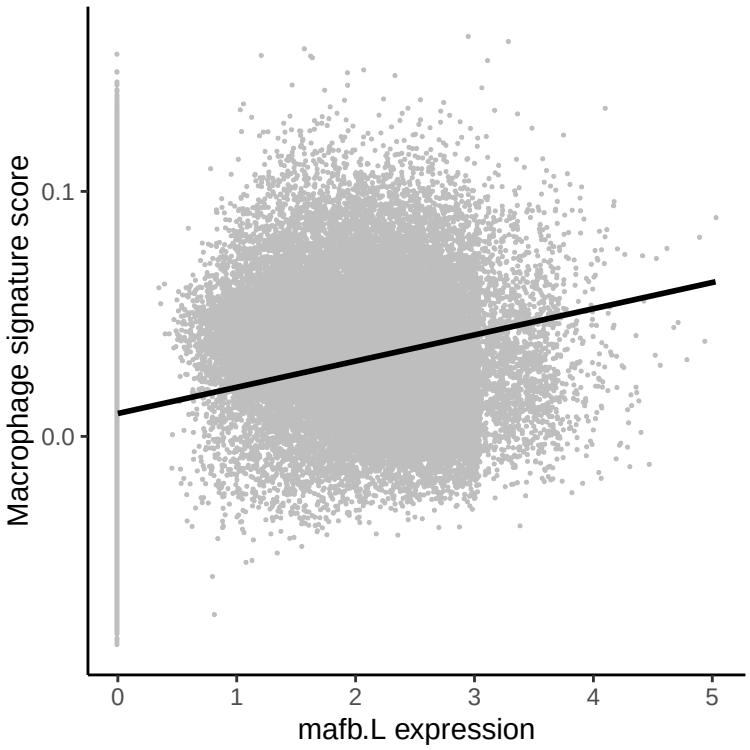
```
p2 <- ggplot(corr, aes(x = mafb.S, y = Mac_sign1)) + geom_point(size = 0.1,
    colour = "grey") + geom_smooth(method = "lm", colour = "black") +
    xlab("mafb.S expression") + ylab("Macrophage signature score") +
    theme_classic()

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

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




```
p3 <- ggplot(corr, aes(x = mafb.L, y = Mac_sign1)) + geom_point(size = 0.1,
    colour = "grey") + geom_smooth(method = "lm", colour = "black") +
    xlab("mafb.L 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

```
MafB_target_sign <- read_csv("MafB_target_Xlaevis.csv")
```

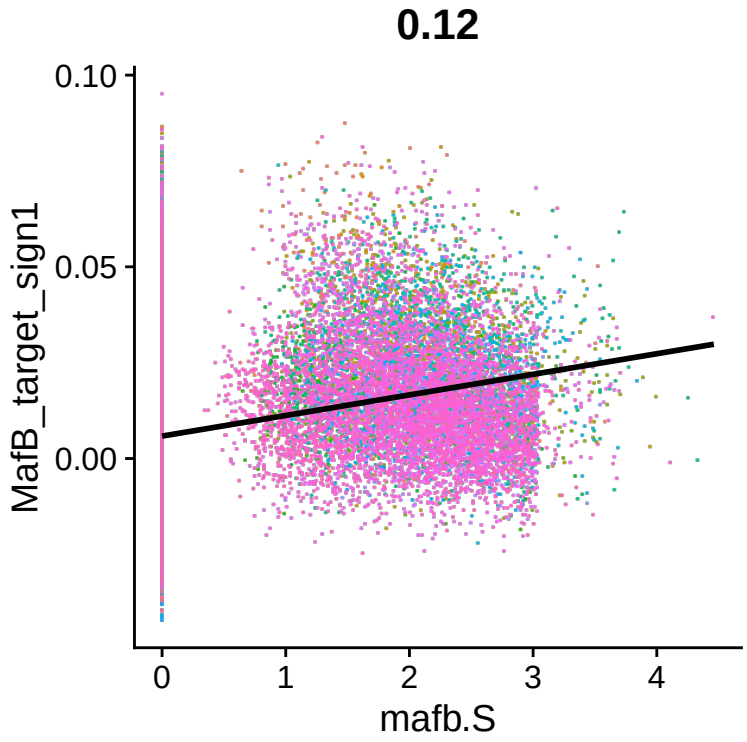
```
## New names:
## Rows: 1807 Columns: 14
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (11): Gene1ID, Gene1Symbol, Gene1SpeciesTaxonID, Gene1SpeciesName, Gene2... dbl
## (3): ...1, AlgorithmsMatch, OutOfAlgorithms
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
MafB_target_sign <- MafB_target_sign$Gene2Symbol

MafB_target_sign <- intersect(MafB_target_sign, rownames(XCA))

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

```
FeatureScatter(XCA, feature1 = "mafb.S", 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'
```

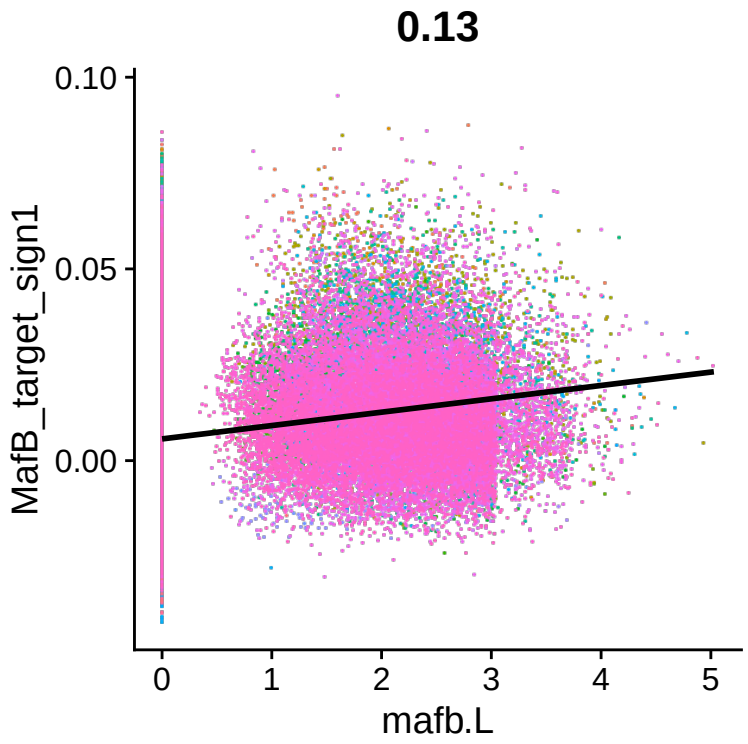
```
## 'geom_smooth()' using formula = 'y ~ x'
```




```
FeatureScatter(XCA, feature1 = "mafb.L", 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'
```




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

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

```
##
##  Pearson's product-moment correlation
##
## data:  corr$mafb.S and corr$MafB_target_sign1
## t = 86.628, df = 501356, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1187108 0.1241652
## sample estimates:
##       cor
## 0.1214389
```

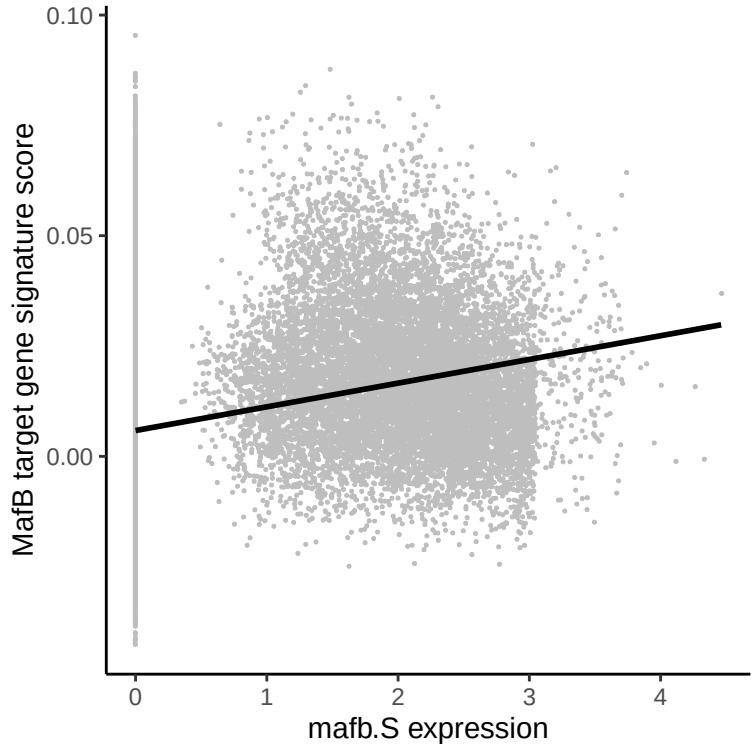
```
cor.test(corr$mafb.L, corr$MafB_target_sign1, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  corr$mafb.L and corr$MafB_target_sign1
## t = 95.262, df = 501356, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1306168 0.1360545
## sample estimates:
##       cor
```

```
## 0.1333367
```

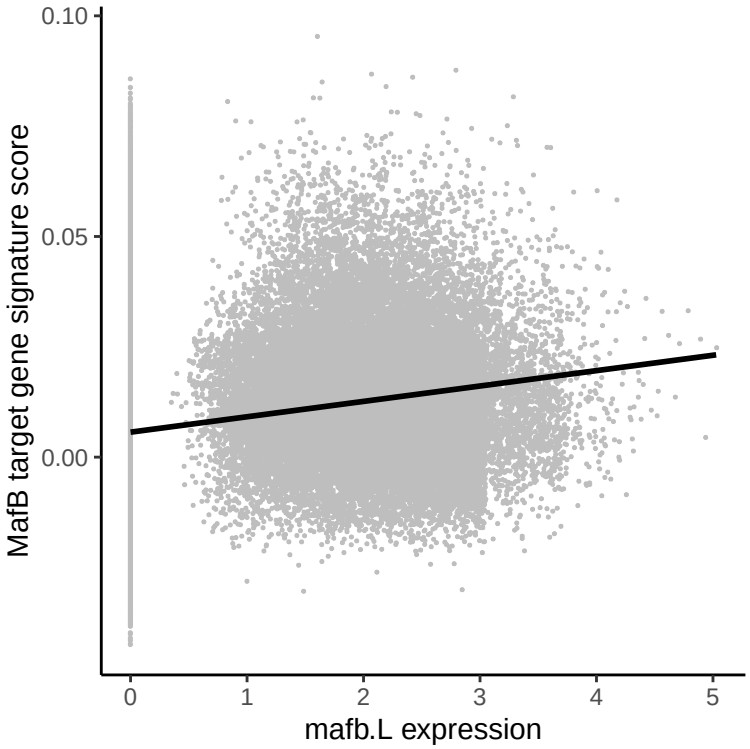
```
p4 <- ggplot(corr, aes(x = mafb.S, y = MafB_target_sign1)) +
    geom_point(size = 0.1, colour = "grey") + geom_smooth(method = "lm",
    colour = "black") + xlab("mafb.S expression") + ylab("MafB target gene signature score") +
    theme_classic()

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

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




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

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

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




```
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] readr_2.1.5        ggrastr_1.0.2      ggplot2_3.4.4      sceasy_0.0.7
## [5] reticulate_1.39.0  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      sctransform_0.4.1      parallelly_1.26.0
##  [16] KernSmooth_2.23-24     htmlwidgets_1.5.3      ica_1.0-2
##  [19] plyr_1.8.6             plotly_4.10.4          zoo_1.8-9
##  [22] igraph_1.2.6           mime_0.11              lifecycle_1.0.4
##  [25] pkgconfig_2.0.3        Matrix_1.6-1.1         R6_2.5.1
##  [28] fastmap_1.2.0          fitdistrplus_1.1-5     future_1.21.0
##  [31] shiny_1.9.1            digest_0.6.37          colorspace_2.1-1
##  [34] patchwork_1.1.1        tensor_1.5             irlba_2.3.5.1
##  [37] labeling_0.4.3         progressr_0.14.0       fansi_1.0.6
##  [40] spatstat.sparse_3.1-0  mgcv_1.9-1             httr_1.4.7
##  [43] polyclip_1.10-0        abind_1.4-5            compiler_4.4.1
##  [46] bit64_4.5.2            withr_3.0.1            highr_0.11
##  [49] MASS_7.3-61            tools_4.4.1            vipor_0.4.7
##  [52] lmtest_0.9-38          beeswarm_0.4.0         httpuv_1.6.1
##  [55] future.apply_1.7.0     goftest_1.2-2          glue_1.7.0
##  [58] nlme_3.1-165           promises_1.2.0.1       grid_4.4.1
##  [61] Rtsne_0.15             cluster_2.1.6          reshape2_1.4.4
##  [64] generics_0.1.0         gtable_0.3.5           spatstat.data_3.1-2
##  [67] tzdb_0.4.0             tidyr_1.3.1            data.table_1.14.0
##  [70] hms_1.1.3              utf8_1.2.4             spatstat.geom_3.3-3
##  [73] RcppAnnoy_0.0.18       ggrepel_0.9.6          RANN_2.6.1
##  [76] pillar_1.9.0           stringr_1.5.1          vroom_1.6.5
##  [79] later_1.2.0            splines_4.4.1          dplyr_1.1.4
##  [82] lattice_0.22-5         bit_4.5.0              survival_3.7-0
##  [85] deldir_2.0-4           tidyselect_1.2.1       miniUI_0.1.1.1
##  [88] pbapply_1.4-3          knitr_1.48             gridExtra_2.3
##  [91] scattermore_0.7        xfun_0.47              matrixStats_1.4.1
##  [94] stringi_1.6.2          lazyeval_0.2.2         yaml_2.2.1
##  [97] evaluate_1.0.0         codetools_0.2-19       tibble_3.2.1
## [100] cli_3.6.3              uwot_0.2.2             xtable_1.8-4
## [103] munsell_0.5.1          Rcpp_1.0.13            globals_0.14.0
## [106] spatstat.random_3.3-2  png_0.1-8              spatstat.univar_3.0-1
## [109] parallel_4.4.1         listenv_0.8.0          viridisLite_0.4.2
## [112] scales_1.3.0           ggridges_0.5.3         crayon_1.4.1
## [115] leiden_0.3.8           purrr_1.0.2            rlang_1.1.4
## [118] cowplot_1.1.1          formatR_1.14
```