

# Tabula Microcebus - MafB

Domien

2025-09-19 11:32:53 +0200

## Contents

Introduction	2
Load packages	2
Download Tabula Microcebus scanpy object from Figshare server	2
Convert Scanpy object to Seurat object	2
Load Tabula Microcebus Seurat object	2
Visualize clusters	2
Calculate correlation between MAFB expression and Mac signature score	3
Calculate correlation between MAFB expression and MafB target gene signature score	6

## 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 signature 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(readxl)  
})
```

## Download Tabula Microcebus scanpy object from Figshare server

```
https://figshare.com/articles/dataset/Tabula\_Microcebus\_v1\_0/14468196?file=31777475
```

## Convert Scanpy object to Seurat object

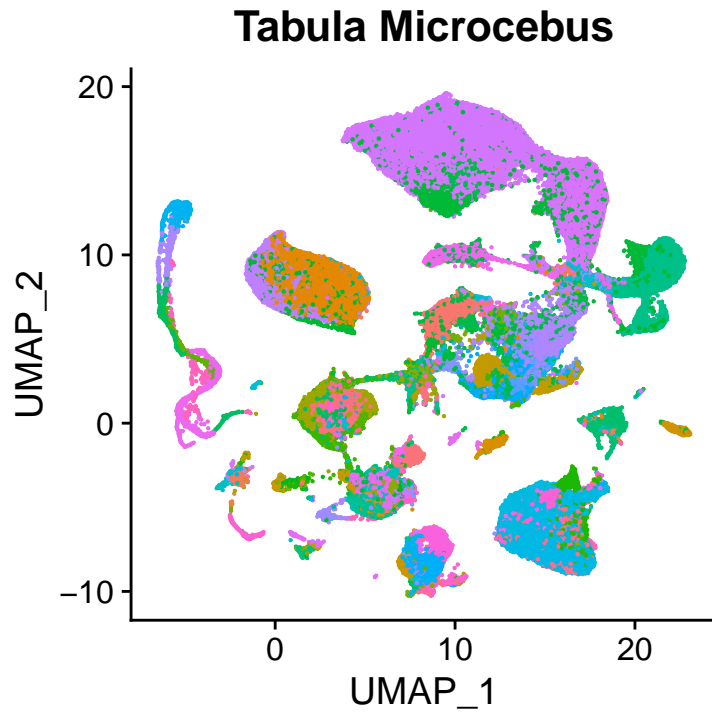
```
sceasy::convertFormat("LCA_complete_wRaw_toPublish.h5ad", from = "anndata",  
  to = "seurat", outFile = "TBM.rds")
```

## Load Tabula Microcebus Seurat object

```
TBM <- readRDS("/media/uliege/T7/SingleCellAtlas/TBM.rds")
```

## Visualize clusters

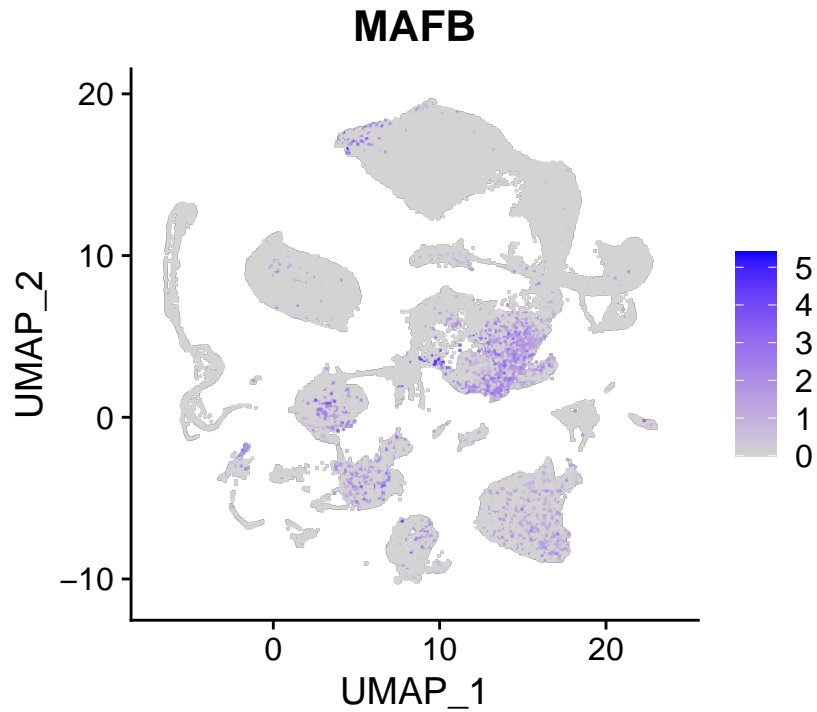
```
p1 <- DimPlot(TBM, group.by = "free_annotation_v1", raster = F) +  
  theme(legend.position = "none") + ggtitle("Tabula Microcebus")  
  
rasterize(p1, layers = "Point", dpi = 1200)
```



Calculate correlation between MAFB expression and Mac signature score

```
FeaturePlot(TBM, features = "MAFB", raster = T)
```

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

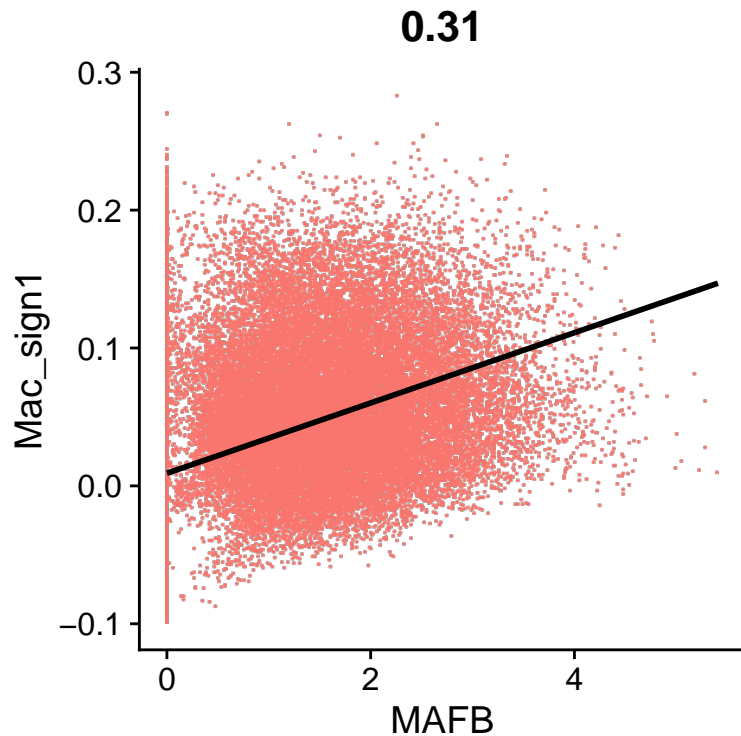


```
Mac_sign <- read_excel("Mac_sign_lemur.xlsx")
Mac_sign <- Mac_sign$ortholog_name
Mac_sign <- intersect(Mac_sign, rownames(TBM))
TBM <- AddModuleScore(TBM, features = list(c(Mac_sign)), name = "Mac_sign")
```

```
FeatureScatter(TBM, feature1 = "MAFB", 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(TBM, vars = "MAFB")
corr <- cbind(corr, FetchData(TBM, vars = "Mac_sign1"))

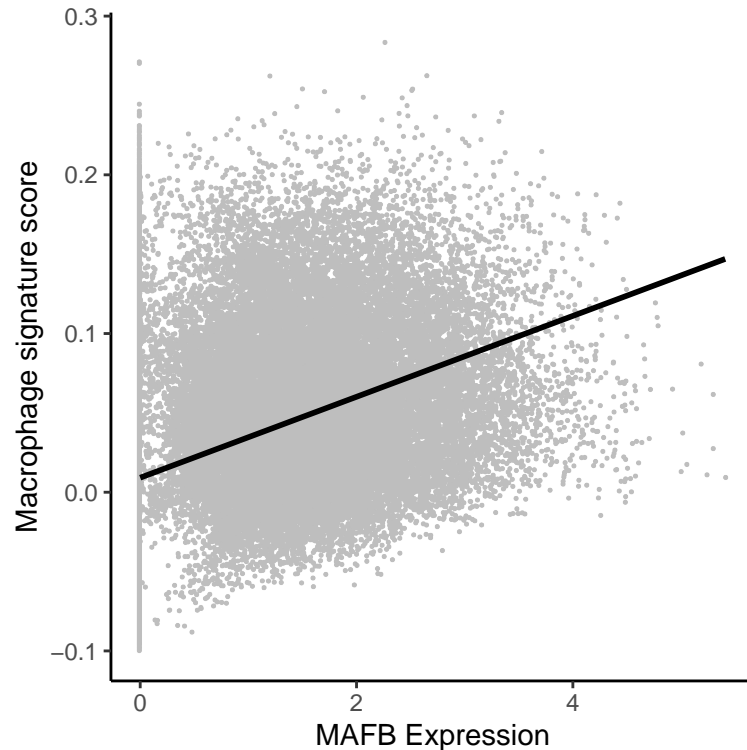
cor.test(corr$MAFB, corr$Mac_sign1, method = "pearson")

##
## Pearson's product-moment correlation
##
## data:  corr$MAFB and corr$Mac_sign1
## t = 160.21, df = 244079, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3048688 0.3120482
## sample estimates:
##      cor
## 0.3084629
```

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

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

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



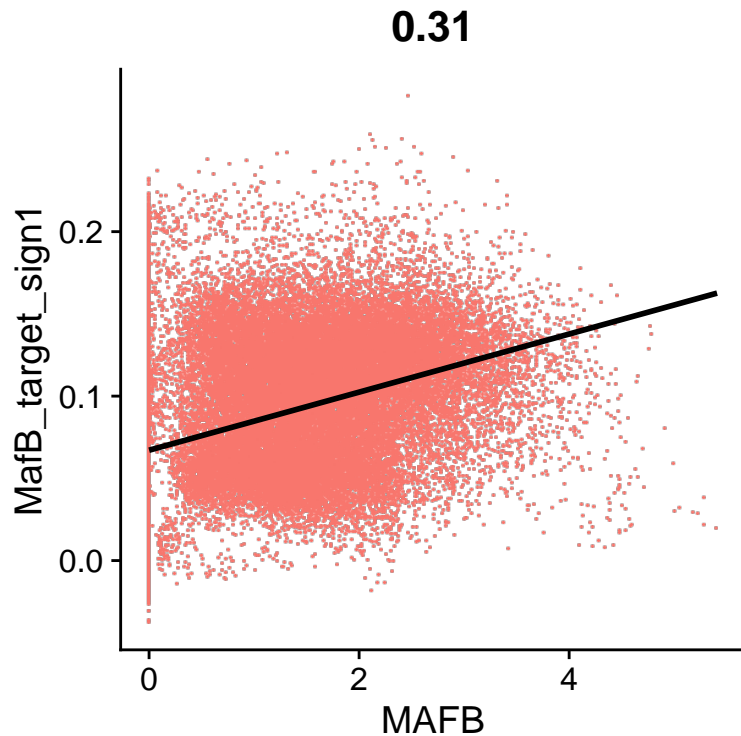
**Calculate correlation between MAFB expression and MafB target gene signature score**

```
Conserved_MafB_target_genes <- read.csv("Conserved_MafB_target_genes_microcebus.csv")
Conserved_MafB_target_genes <- Conserved_MafB_target_genes$ortholog_name
Conserved_MafB_target_genes <- intersect(Conserved_MafB_target_genes,
  rownames(TBM))
TBM <- AddModuleScore(TBM, features = list(c(Conserved_MafB_target_genes)),
  name = "MafB_target_sign")
```

```
FeatureScatter(TBM, feature1 = "MAFB", 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'
```

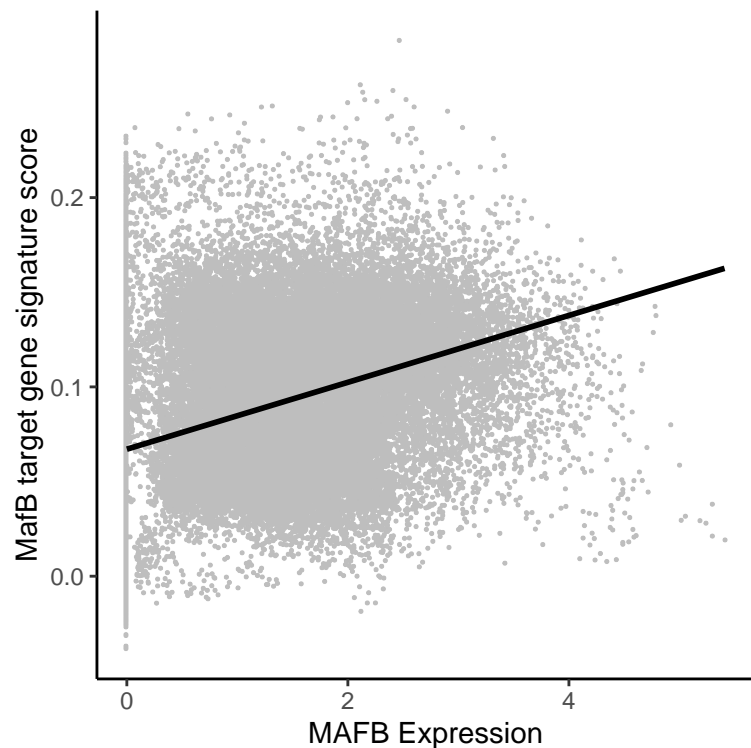


```
corr <- cbind(corr, FetchData(TBM, vars = "MafB_target_sign1"))
cor.test(corr$MAFB, corr$MafB_target_sign1, method = "pearson")
```

```
##
## Pearson's product-moment correlation
##
## data: corr$MAFB and corr$MafB_target_sign1
## t = 162.77, df = 244079, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3093311 0.3164885
## sample estimates:
##      cor
## 0.3129142
```

```
p3 <- ggplot(corr, aes(x = MAFB, y = MafB_target_sign1)) + geom_point(size = 0.1,
  colour = "grey") + geom_smooth(method = "lm", colour = "black") +
  xlab("MAFB Expression") + ylab("MafB target gene signature score") +
  theme_classic()
rasterize(p3, 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] readxl_1.4.3      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] deldir_2.0-4	pbapply_1.4-3	gridExtra_2.3
## [4] formatR_1.14	rlang_1.1.4	magrittr_2.0.3
## [7] RcppAnnoy_0.0.18	spatstat.geom_3.3-3	matrixStats_1.4.1
## [10] ggbridges_0.5.3	compiler_4.4.1	mgcv_1.9-1
## [13] png_0.1-8	vctrs_0.6.5	reshape2_1.4.4
## [16] stringr_1.5.1	pkgconfig_2.0.3	fastmap_1.2.0
## [19] labeling_0.4.3	utf8_1.2.4	promises_1.2.0.1
## [22] rmarkdown_2.28	ggbeeswarm_0.7.2	purrr_1.0.2
## [25] xfun_0.47	jsonlite_1.8.9	goftest_1.2-2
## [28] highr_0.11	later_1.2.0	spatstat.utils_3.1-0
## [31] irlba_2.3.5.1	parallel_4.4.1	cluster_2.1.6
## [34] R6_2.5.1	ica_1.0-2	stringi_1.6.2
## [37] RColorBrewer_1.1-3	spatstat.data_3.1-2	parallelly_1.26.0
## [40] spatstat.univar_3.0-1	cellranger_1.1.0	lmtest_0.9-38
## [43] scattermore_0.7	Rcpp_1.0.13	knitr_1.48
## [46] tensor_1.5	future.apply_1.7.0	zoo_1.8-9
## [49] sctransform_0.4.1	httpuv_1.6.1	Matrix_1.6-1.1
## [52] splines_4.4.1	igraph_1.2.6	tidyselect_1.2.1
## [55] abind_1.4-5	rstudioapi_0.16.0	yaml_2.2.1
## [58] spatstat.random_3.3-2	codetools_0.2-19	miniUI_0.1.1.1
## [61] spatstat.explore_3.3-2	listenv_0.8.0	lattice_0.22-5
## [64] tibble_3.2.1	plyr_1.8.6	withr_3.0.1
## [67] shiny_1.9.1	ROCR_1.0-11	evaluate_1.0.0
## [70] Rtsne_0.15	future_1.21.0	survival_3.7-0
## [73] polyclip_1.10-0	fitdistrplus_1.1-5	pillar_1.9.0
## [76] KernSmooth_2.23-24	plotly_4.10.4	generics_0.1.0
## [79] munsell_0.5.1	scales_1.3.0	globals_0.14.0
## [82] xtable_1.8-4	glue_1.7.0	lazyeval_0.2.2
## [85] tools_4.4.1	data.table_1.14.0	RANN_2.6.1
## [88] leiden_0.3.8	Cairo_1.6-2	cowplot_1.1.1
## [91] grid_4.4.1	tidyr_1.3.1	colorspace_2.1-1
## [94] nlme_3.1-165	patchwork_1.1.1	beeswarm_0.4.0
## [97] vipor_0.4.7	cli_3.6.3	spatstat.sparse_3.1-0
## [100] fansi_1.0.6	viridisLite_0.4.2	dplyr_1.1.4
## [103] uwot_0.2.2	gtable_0.3.5	digest_0.6.37
## [106] progressr_0.14.0	ggrepel_0.9.6	farver_2.1.2
## [109] htmlwidgets_1.5.3	htmltools_0.5.8.1	lifecycle_1.0.4
## [112] httr_1.4.7	mime_0.11	MASS_7.3-61