

FigureS9_forebrain_midbrain_late

This documents details the steps for the generation of pseudotemporal expression profiles described in “Temporal patterning of the central nervous system by a shared transcription factor code” by Sagner et al. 2021.

Connect loom file from La Manno et al. 2020.

```
## connect sc.loom file downloaded from mousebrain.org
sc.loom <- loomR::connect(filename = paste0(dir, "/input/dev_all.loom"), mode = 'r+', skip.validate = TRUE)

## Generate sc.meta file by extracting parameters from connected sc.loom file
sc.meta <- data.frame(sc.loom$col.attrs$Age[],
                      sc.loom$col.attrs$PseudoAge[],
                      sc.loom$col.attrs$Tissue[],
                      sc.loom$col.attrs$PseudoTissue[],
                      sc.loom$col.attrs$Class[],
                      sc.loom$col.attrs$Clusters[],
                      10000 / sc.loom$col.attrs$TotalUMI[],
                      sc.loom$col.attrs$CellID[],
                      sc.loom$col.attrs$SampleID[])

colnames(sc.meta) <- c("age", "pseudoage", "tissue", "pseudotissue", "class", "clusters", "normalization")
```

Late midbrain stages / comparison tTFs

```
tissue = "Midbrain"
celltype = "Neuron"
timepoints = c('e16.0', 'e16.25', 'e16.5', 'e17.0', 'e17.5', 'e18.0')

tissue.id <- which(grepl(tissue, unique(sc.loom$col.attrs$Tissue[])) == TRUE)
cell.id <- intersect(which(sc.meta$tissue %in% unique(sc.loom$col.attrs$Tissue[])[tissue.id] & sc.meta$age %in% timepoints))

exp.mat <- sc.loom[["matrix"]][cell.id,]

colnames(exp.mat) <- sc.loom$row.attrs$Gene[]
rownames(exp.mat) <- sc.meta$cellID[cell.id]

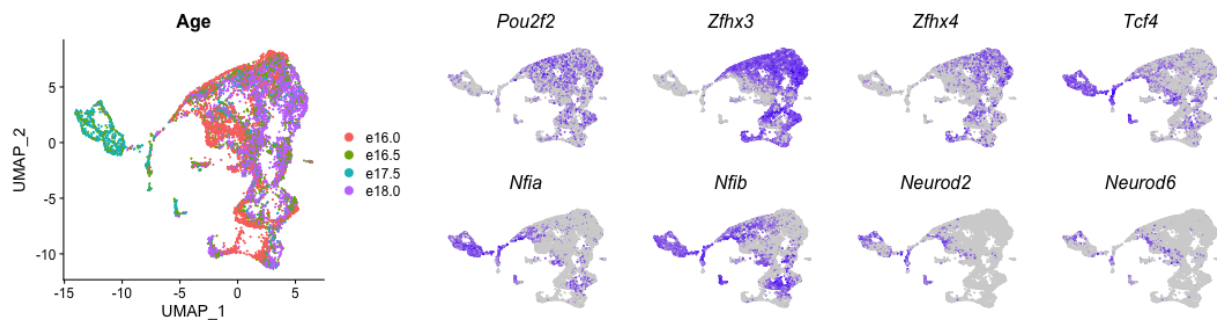
mb.seurat.late <- CreateSeuratObject(counts = t(exp.mat),
                                     meta.data = sc.meta[cell.id,] %>%
                                       as.tibble() %>%
                                       tibble::column_to_rownames("cellID"))

mb.seurat.late[["percent.mt"]] <- PercentageFeatureSet(mb.seurat.late, pattern = "^mt-")
```

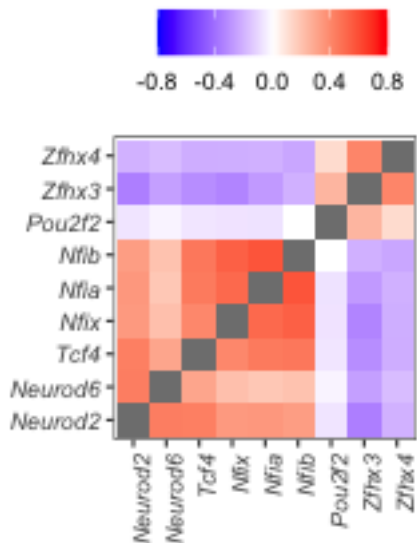
```
mb.seurat.late <- mb.seurat.late %>%
  subset(subset = nFeature_RNA > 600 & nFeature_RNA < 6000 & percent.mt < 6) %>%
  SCTransform(vars.to.regress = 'sampleID') %>%
  NormalizeData(verbose=FALSE) %>%
  ScaleData(verbose = FALSE) %>%
  FindVariableFeatures(selection.method = "vst", verbose = FALSE) %>%
  RunPCA(npcs = 30, verbose = FALSE) %>%
  RunUMAP(reduction = "pca", dims = 1:30) %>%
  FindNeighbors(dims = 1:30) %>%
  FindClusters(resolution = 0.5)
```

```
mb.umap.late <- cowplot::plot_grid(DimPlot(mb.seurat.late, reduction = 'umap', group.by = "age") + ggtitle("Age"),
  FeaturePlot(mb.seurat.late, features = c('Pou2f2', 'Zfhx3', 'Zfhx4', 'Tcf4',
    'Nfia', 'Nfib', 'Neurod2', 'Neurod6'),
    ncol = 4) & NoLegend() & NoAxes() & theme(plot.title = element_text(face = "bold", size = 12),
    nrow = 1, rel_widths = c(1,2))
```

```
mb.umap.late
```

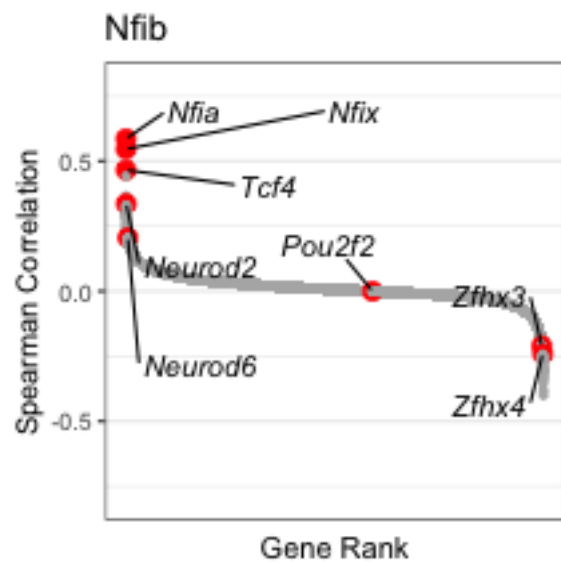
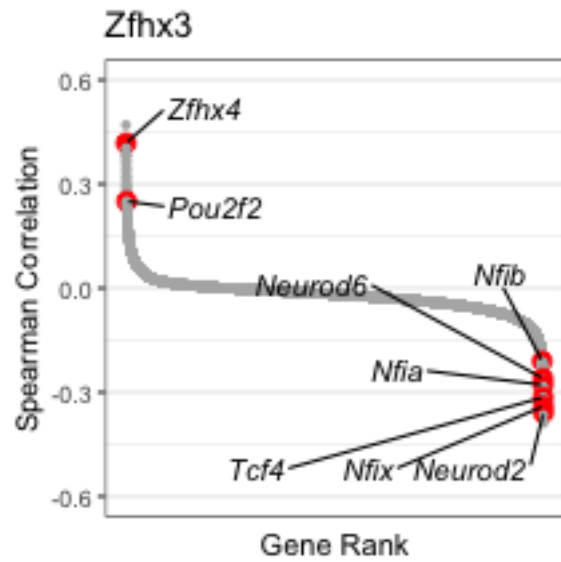


```
mb.corr.mtx <- plot.correlation.mtx(
  input = mb.seurat.late,
  input.type = c("Seurat"),
  correlation.genes = c(
    "Pou2f2", "Zfhx3", "Zfhx4", "Nfia", "Nfib", "Nfix", "Neurod2", 'Neurod6', "Tcf4"
  ),
  min = -0.8,
  max = 0.8
)
mb.corr.mtx
```



```
mb.corr.rank <- cowplot::plot_grid(
  plot.correlation.ranks(
    input = mb.seurat.late,
    input.type = c("Seurat"),
    plot.gene = 'Zfhx3',
    correlation.genes = c(
      "Pou2f2", "Zfhx3", "Zfhx4", "Nfia", "Nfib", "Nfix", "Neurod2", "Neurod6", "Tcf4"
    ),
    min = -0.6,
    max = 0.6
  ),
  plot.correlation.ranks(
    input = mb.seurat.late,
    input.type = c("Seurat"),
    plot.gene = 'Nfib',
    correlation.genes = c(
      "Pou2f2", "Zfhx3", "Zfhx4", "Nfia", "Nfib", "Nfix", "Neurod2", "Neurod6", "Tcf4"
    ),
    min = -0.8,
    max = 0.8
  ),
  ncol = 1
)

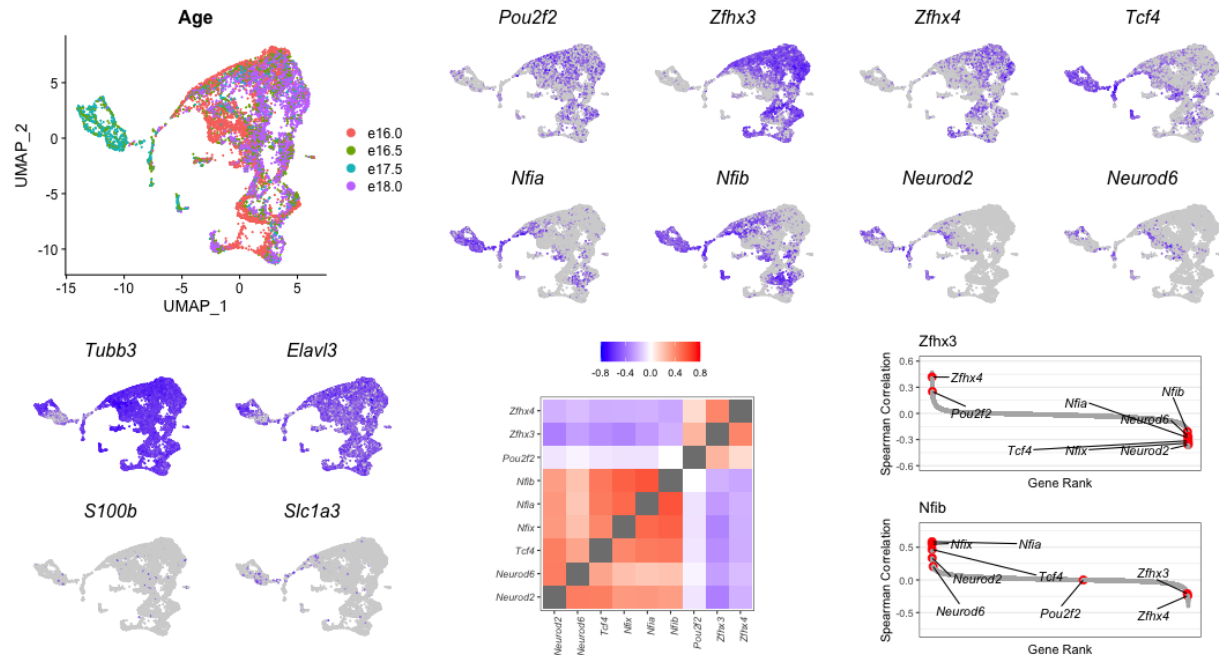
mb.corr.rank
```



```
bottom <- cowplot::plot_grid(FeaturePlot(mb.seurat.late, features = c('Tubb3', 'Elavl3', 'Slc1',
  NoLegend() & NoAxes() & theme(plot.title = element_text(face = 'italic'))),
  mb.corr.mtx,
  mb.corr.rank + theme(aspect.ratio = 1) & theme(plot.title = element_text(f
  nrow = 1)

mb.plot <- cowplot::plot_grid(mb.umap.late, bottom,
  nrow = 2)

mb.plot
```



Late forebrain stages / comparison tTFs

```

tissue = "Forebrain"
celltype = "Neuron"
timepoints = c('e16.0', 'e16.25', 'e16.5', 'e17.0', 'e17.5', 'e18.0')

tissue.id <- which(grepl(tissue, unique(sc.loom$col.attrs$Tissue[])) == TRUE)
cell.id <- intersect(which(sc.meta$tissue %in% unique(sc.loom$col.attrs$Tissue[])[tissue.id] & sc.meta$
                      which(sc.meta$age %in% timepoints))

exp.mat <- sc.loom[["matrix"]][cell.id,]

colnames(exp.mat) <- sc.loom$row.attrs$Gene[]
rownames(exp.mat) <- sc.meta$cellID[cell.id]

fb.seurat.late <- CreateSeuratObject(counts = t(exp.mat),
                                     meta.data = sc.meta[cell.id,] %>%
                                     as.tibble() %>%
                                     tibble::column_to_rownames("cellID"))

fb.seurat.late[["percent.mt"]] <- PercentageFeatureSet(fb.seurat.late, pattern = "^mt-")

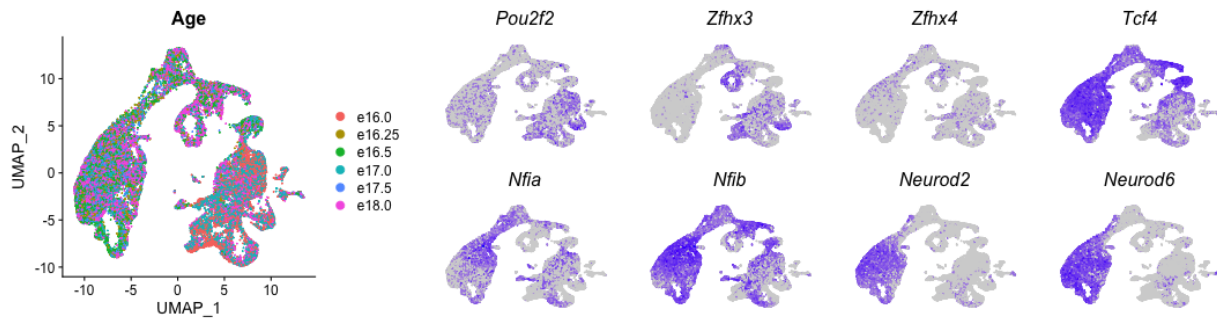
fb.seurat.late <- fb.seurat.late %>%
  subset(subset = nFeature_RNA > 600 & nFeature_RNA < 6000 & percent.mt < 6) %>%
  SCTransform(vars.to.regress = 'sampleID') %>%
  NormalizeData(verbose=FALSE) %>%
  ScaleData(verbose = FALSE) %>%
  FindVariableFeatures(selection.method = "vst", verbose = FALSE) %>%
  RunPCA(npcs = 30, verbose = FALSE) %>%
  RunUMAP(reduction = "pca", dims = 1:30) %>%

```

```
FindNeighbors(dims = 1:30) %>%
FindClusters(resolution = 0.5)
```

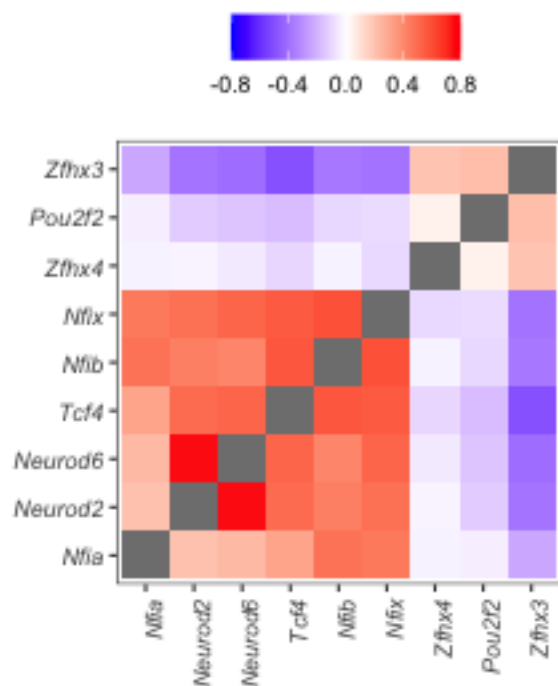
```
fb.umap.late <- cowplot::plot_grid(DimPlot(fb.seurat.late, reduction = 'umap', group.by = "age") + ggtitle("UMAP plot"),
  FeaturePlot(fb.seurat.late, features = c('Pou2f2', 'Zfhx3', 'Zfhx4', 'Tcf4',
    'Nfia', 'Nfib', 'Neurod2', 'Neurod6'),
    ncol = 4) & NoLegend() & NoAxes() & theme(plot.title = element_text(face = "bold", size = 12),
    nrow = 1, rel_widths = c(1,2))

fb.umap.late
```



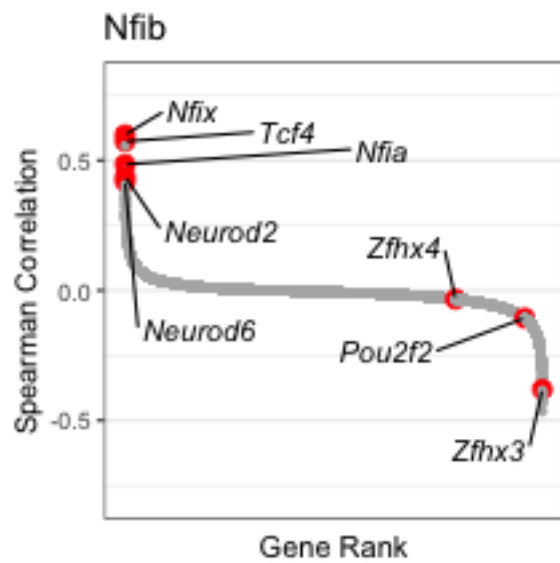
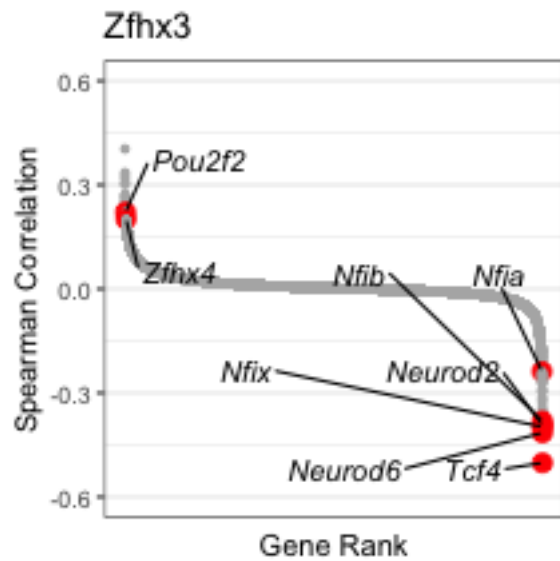
```
fb.corr.mtx <- plot.correlation.mtx(
  input = fb.seurat.late,
  input.type = c("Seurat"),
  correlation.genes = c(
    "Pou2f2", "Zfhx3", "Zfhx4", "Nfia", "Nfib", "Nfix", "Neurod2", 'Neurod6', "Tcf4"
  ),
  min = -0.8,
  max = 0.8
)

fb.corr.mtx
```



```
fb.corr.rank <- cowplot::plot_grid(
  plot.correlation.ranks(
    input = fb.seurat.late,
    input.type = c("Seurat"),
    plot.gene = 'Zfhx3',
    correlation.genes = c(
      "Pou2f2", "Zfhx3", "Zfhx4", "Nfia", "Nfib", "Nfix", "Neurod2", "Neurod6", "Tcf4"
    ),
    min = -0.6,
    max = 0.6
  ),
  plot.correlation.ranks(
    input = fb.seurat.late,
    input.type = c("Seurat"),
    plot.gene = 'Nfib',
    correlation.genes = c(
      "Pou2f2", "Zfhx3", "Zfhx4", "Nfia", "Nfib", "Nfix", "Neurod2", "Neurod6", "Tcf4"
    ),
    min = -0.8,
    max = 0.8
  ),
  nrow = 2
)
```

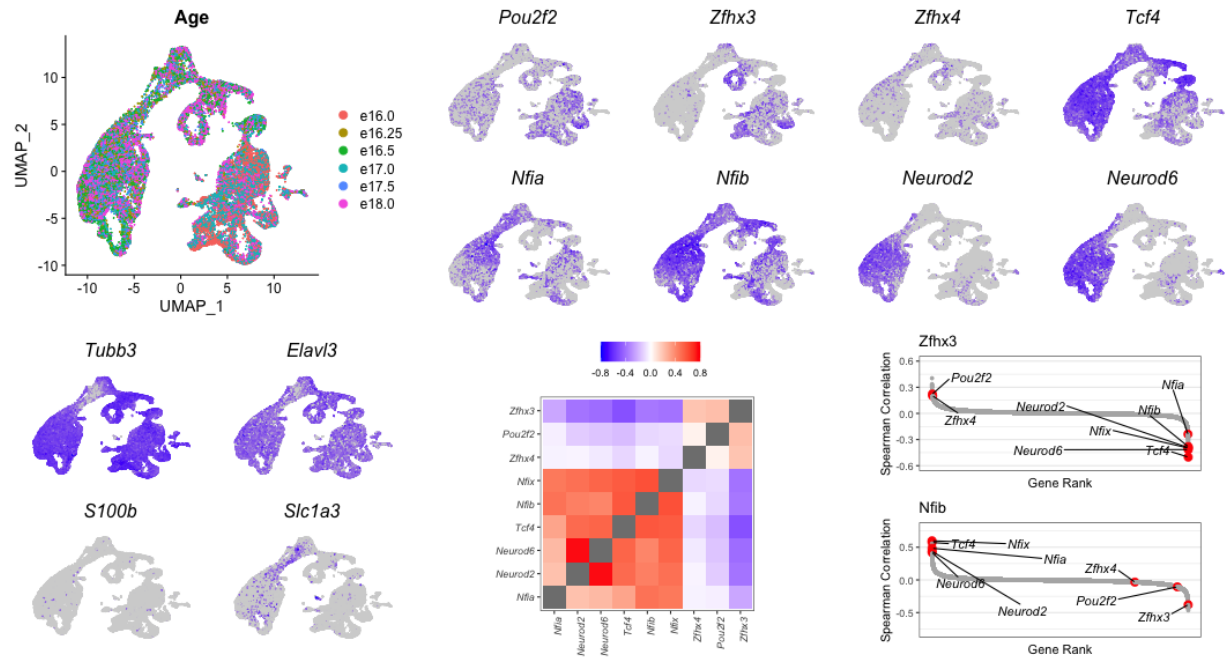
```
fb.corr.rank
```



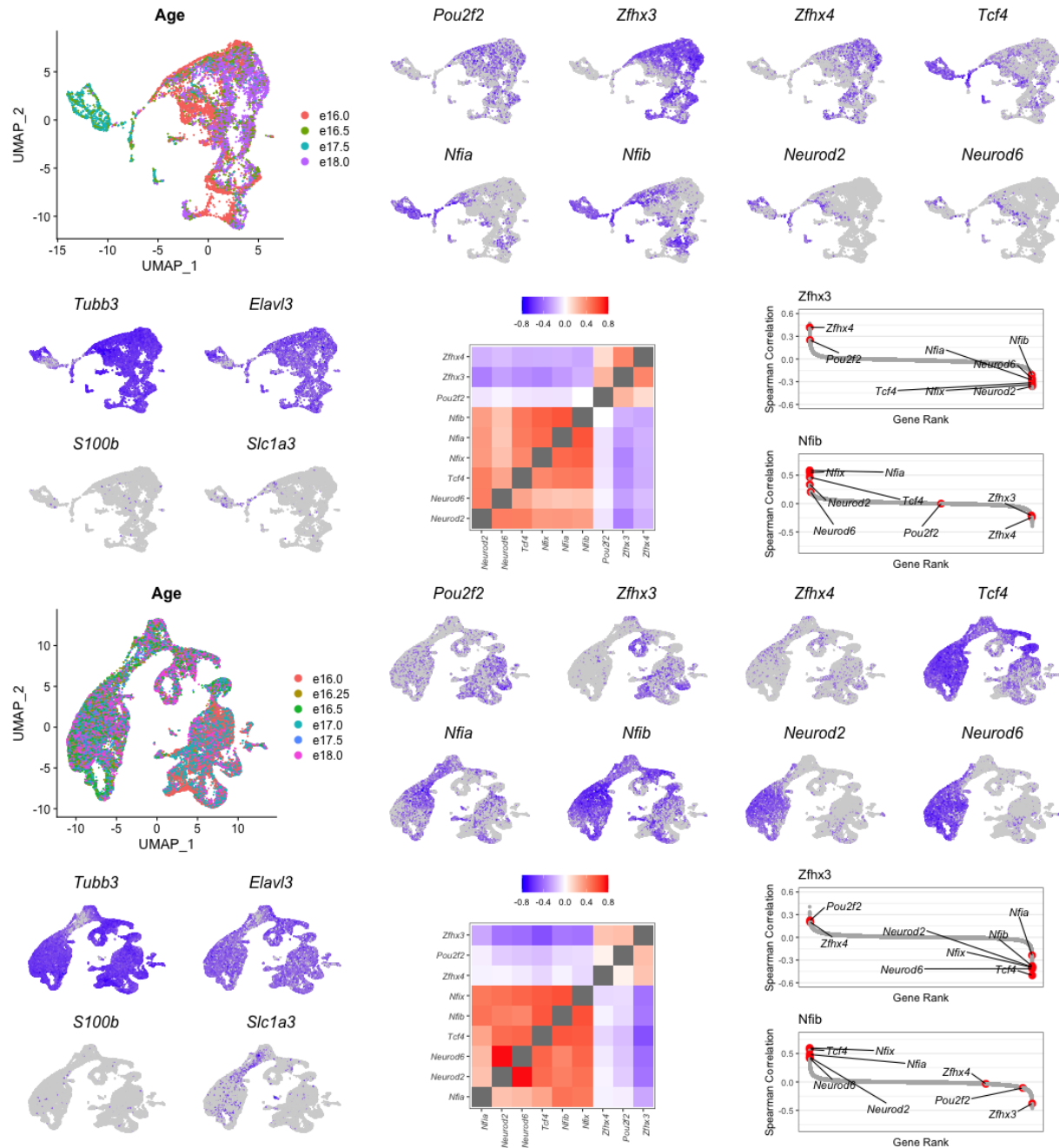
```
bottom <- cowplot::plot_grid(FeaturePlot(fb.seurat.late, features = c('Tubb3', 'Elavl3', 'Slc100b', 'Slc100a1'),
  NoLegend() & NoAxes() & theme(plot.title = element_text(face = 'italic')),
  fb.corr.mtx,
  fb.corr.rank + theme(aspect.ratio = 1) & theme(plot.title = element_text(face = 'italic')),
  nrow = 1)

fb.plot <- cowplot::plot_grid(fb.umap.late, bottom,
  nrow = 2)

fb.plot
```

```
cowplot::plot_grid(mb.plot, fb.plot, nrow = 2)
```



```
sessionInfo()
```

```
## R version 4.0.4 (2021-02-15)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
```

```

## [1] en_GB.UTF-8/en_GB.UTF-8/en_GB.UTF-8/C/en_GB.UTF-8/en_GB.UTF-8
##
## attached base packages:
## [1] parallel stats      graphics  grDevices utils      datasets  methods
## [8] base
##
## other attached packages:
## [1] ggrepel_0.9.1      pbapply_1.4-3      viridis_0.6.0
## [4] viridisLite_0.4.0  tibble_3.1.1      ggplot2_3.3.3
## [7] loomR_0.2.1.9000   hdf5r_1.3.2        R6_2.5.0
## [10] scales_1.1.1       dplyr_1.0.5        SeuratObject_4.0.0
## [13] Seurat_4.0.1       slingshot_1.8.0    prncurve_2.1.6
## [16] Biobase_2.50.0     BiocGenerics_0.36.1
##
## loaded via a namespace (and not attached):
## [1] Rtsne_0.15          colorspace_2.0-0
## [3] deldir_0.2-10       ellipsis_0.3.1
## [5] gggridges_0.5.3     XVector_0.30.0
## [7] GenomicRanges_1.42.0 spatstat.data_2.1-0
## [9] farver_2.1.0        leiden_0.3.7
## [11] listenv_0.8.0       bit64_4.0.5
## [13] RSpectra_0.16-0     fansi_0.4.2
## [15] codetools_0.2-18    splines_4.0.4
## [17] knitr_1.32          polyclip_1.10-0
## [19] jsonlite_1.7.2      ica_1.0-2
## [21] cluster_2.1.2       png_0.1-7
## [23] uwot_0.1.10         spatstat.sparse_2.0-0
## [25] shiny_1.6.0         sctransform_0.3.2
## [27] compiler_4.0.4      httr_1.4.2
## [29] assertthat_0.2.1    Matrix_1.3-2
## [31] fastmap_1.1.0       lazyeval_0.2.2
## [33] later_1.1.0.1       htmltools_0.5.1.1
## [35] tools_4.0.4         igraph_1.2.6
## [37] gtable_0.3.0        glue_1.4.2
## [39] GenomeInfoDbData_1.2.4 RANN_2.6.1
## [41] reshape2_1.4.4      Rcpp_1.0.6
## [43] scattermore_0.7     vctrs_0.3.7
## [45] ape_5.4-1           nlme_3.1-152
## [47] lmtest_0.9-38       xfun_0.22
## [49] stringr_1.4.0       globals_0.14.0
## [51] mime_0.10           miniUI_0.1.1.1
## [53] lifecycle_1.0.0     irlba_2.3.3
## [55] goftest_1.2-2       future_1.21.0
## [57] zlibbioc_1.36.0     MASS_7.3-53.1
## [59] zoo_1.8-9           spatstat.core_2.1-2
## [61] spatstat.utils_2.1-0 promises_1.2.0.1
## [63] MatrixGenerics_1.2.1 SummarizedExperiment_1.20.0
## [65] RColorBrewer_1.1-2  SingleCellExperiment_1.12.0
## [67] yaml_2.2.1          reticulate_1.18
## [69] gridExtra_2.3       rpart_4.1-15
## [71] stringi_1.5.3       highr_0.9
## [73] S4Vectors_0.28.1    GenomeInfoDb_1.26.7
## [75] rlang_0.4.10        pkgconfig_2.0.3
## [77] matrixStats_0.58.0  bitops_1.0-6

```

## [79] evaluate_0.14	lattice_0.20-41
## [81] tensor_1.5	ROCR_1.0-11
## [83] purrr_0.3.4	labeling_0.4.2
## [85] patchwork_1.1.1	htmlwidgets_1.5.3
## [87] bit_4.0.4	cowplot_1.1.1
## [89] tidysselect_1.1.1	parallelly_1.25.0
## [91] RcppAnnoy_0.0.18	plyr_1.8.6
## [93] magrittr_2.0.1	IRanges_2.24.1
## [95] generics_0.1.0	DelayedArray_0.16.3
## [97] DBI_1.1.1	withr_2.4.2
## [99] mgcv_1.8-35	pillar_1.6.0
## [101] fitdistrplus_1.1-3	abind_1.4-5
## [103] survival_3.2-10	RCurl_1.98-1.3
## [105] future.apply_1.7.0	crayon_1.4.1
## [107] KernSmooth_2.23-18	utf8_1.2.1
## [109] spatstat.geom_2.1-0	plotly_4.9.3
## [111] rmarkdown_2.7	grid_4.0.4
## [113] data.table_1.14.0	digest_0.6.27
## [115] xtable_1.8-4	tidyr_1.1.3
## [117] httpuv_1.5.5	stats4_4.0.4
## [119] munsell_0.5.0	