FNA transcriptomics

Ed Carr, Linterman lab

8 Dec 2020

```
library(Seurat)
########
# Wrapper function
# Read in each sample, annotate with day
SRRs2Seurat <- function(x) {
  data <- Read10X(data.dir = as.character(x$path))
  obj <- CreateSeuratObject(data,
                            project =
                              "FNA")

  obj <- AddMetaData(obj, metadata = c(as.character(x$day),
                                       x$lib,
                                       paste(
                                         as.character(x$day),
                                    x$lib, sep = "_")
                                    ),
                     col.name = c("day", "lib","day_lib"))

  # % mito reads
  obj[["percent.mt"]] <- PercentageFeatureSet(obj, pattern = "^MT-")

  # simple QC: >200 features/cell, less than 7k features/cell and <12.5% mitochondrial reads.
  obj <- subset(obj, subset = nFeature_RNA > 200 & nFeature_RNA < 7000 & percent.mt < 12.5)
}
#########

# Build a dataframe with sample annotation:

SRRtable <- cbind("path" = c("cellranger/TurnerSRR11233645/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233646/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233647/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233648/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233649/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233650/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233651/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233653/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233654/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233655/outs/filtered_feature_bc_matrix/",
                             "cellranger/TurnerSRR11233656/outs/filtered_feature_bc_matrix/"),
                  "day" = c("d60",
                            "d60",
                            "d28",
```

```
                            "d28",
                            "d12",
                            "d12",
                            "d12",
                            "d5",
                            "d5",
                            "d0",
                            "d0"),
                  # SRA lists FNA_2 or _3 (presumably each core is a #)
                  # And lib 1 or 2 (presumambly 10x channel replicates)
                  # Only some of these combinations exist
                  # Presumably some cores were too low cell number to process.
                  # Presumably some 10x channels clearly failed in prep prior to sequencing.
                  # these are simplified into (1, 2) here.
                  "lib" = c(1,
                            2,
                            2,
                            1,
                            3,
                            2,
                            1,
                            2,
                            1,
                            2,
                            1))


SRRtable <- data.frame(SRRtable)
SRRtable



# Make a list
fna.list <- list()

# Loop for each entry in SRR table to do the import to Seurat.
for(i in 1:nrow(SRRtable)){

  fna.list[[i]] <- SRRs2Seurat(SRRtable[i,])

}


names(fna.list) <- paste(SRRtable$day, SRRtable$lib, sep = "_")
```

```
# First time running, this chunk needs eval=TRUE This is a
# slow chunk, so run once normalised data and written to
# disk.  hence eval=FALSE

for (i in 1:length(fna.list)) {
    fna.list[[i]] <- NormalizeData(fna.list[[i]], verbose = FALSE)
    fna.list[[i]] <- FindVariableFeatures(fna.list[[i]], selection.method = "vst",
        nfeatures = 2000, verbose = FALSE)
}
```

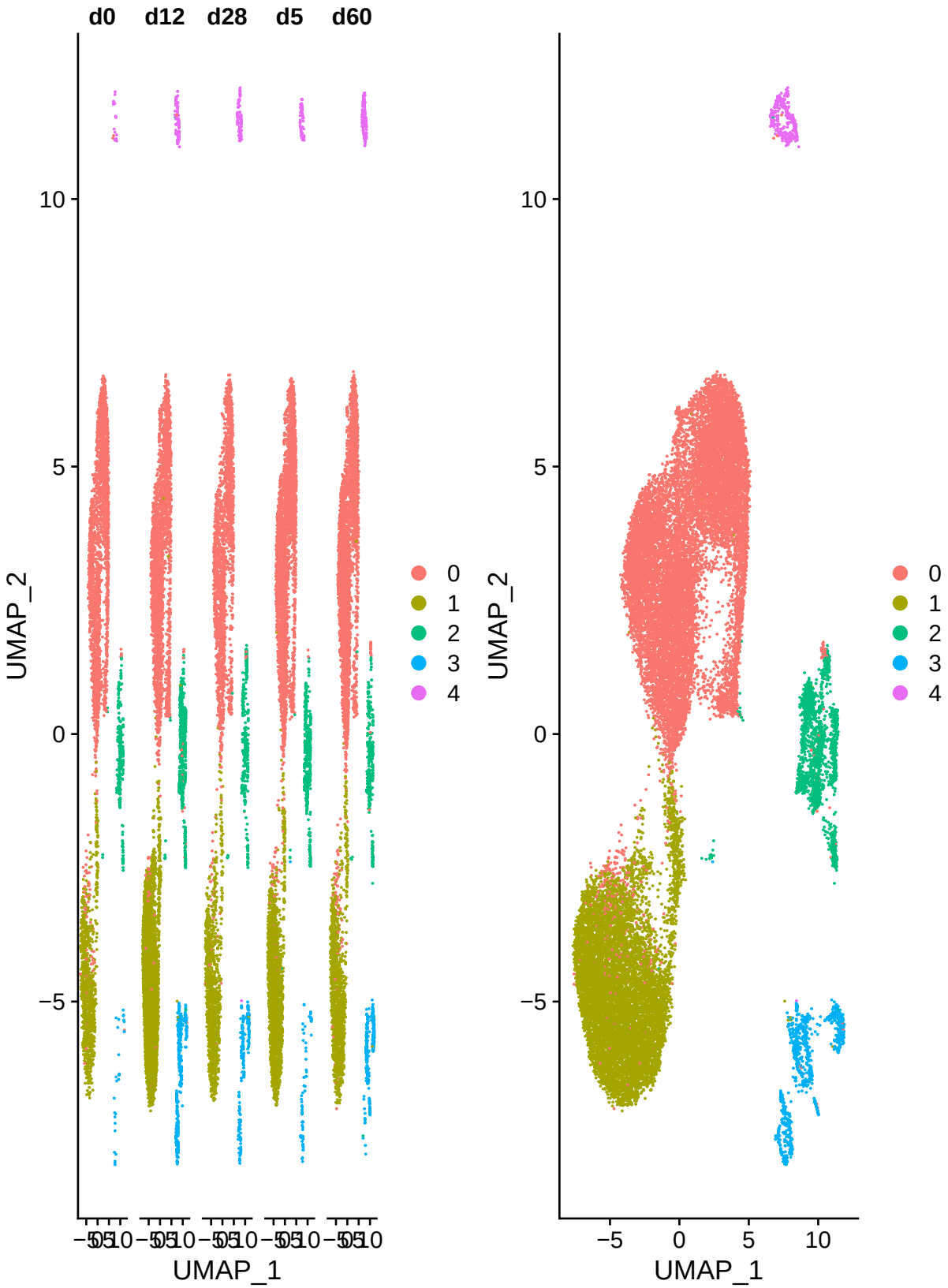
```
# Set reference as a day 12 sample (will have some GC-B, and
# all other cell types) (the Seurat vignette keeps one
# dataset back to 'project' anchors onto)
reference.list <- which(names(fna.list) == "d12_1")

# Find anchors:
fna.anchors <- FindIntegrationAnchors(object.list = fna.list,
    normalization.method = "LogNormalize", reference = reference.list,
    dims = 1:30)

fna.integrated <- IntegrateData(anchorset = fna.anchors, dims = 1:30)

system("mkdir seurat_objects")

save(fna.integrated, file = "seurat_objects/FNA_transcriptomes_AnchorIntegrated.RData")
```

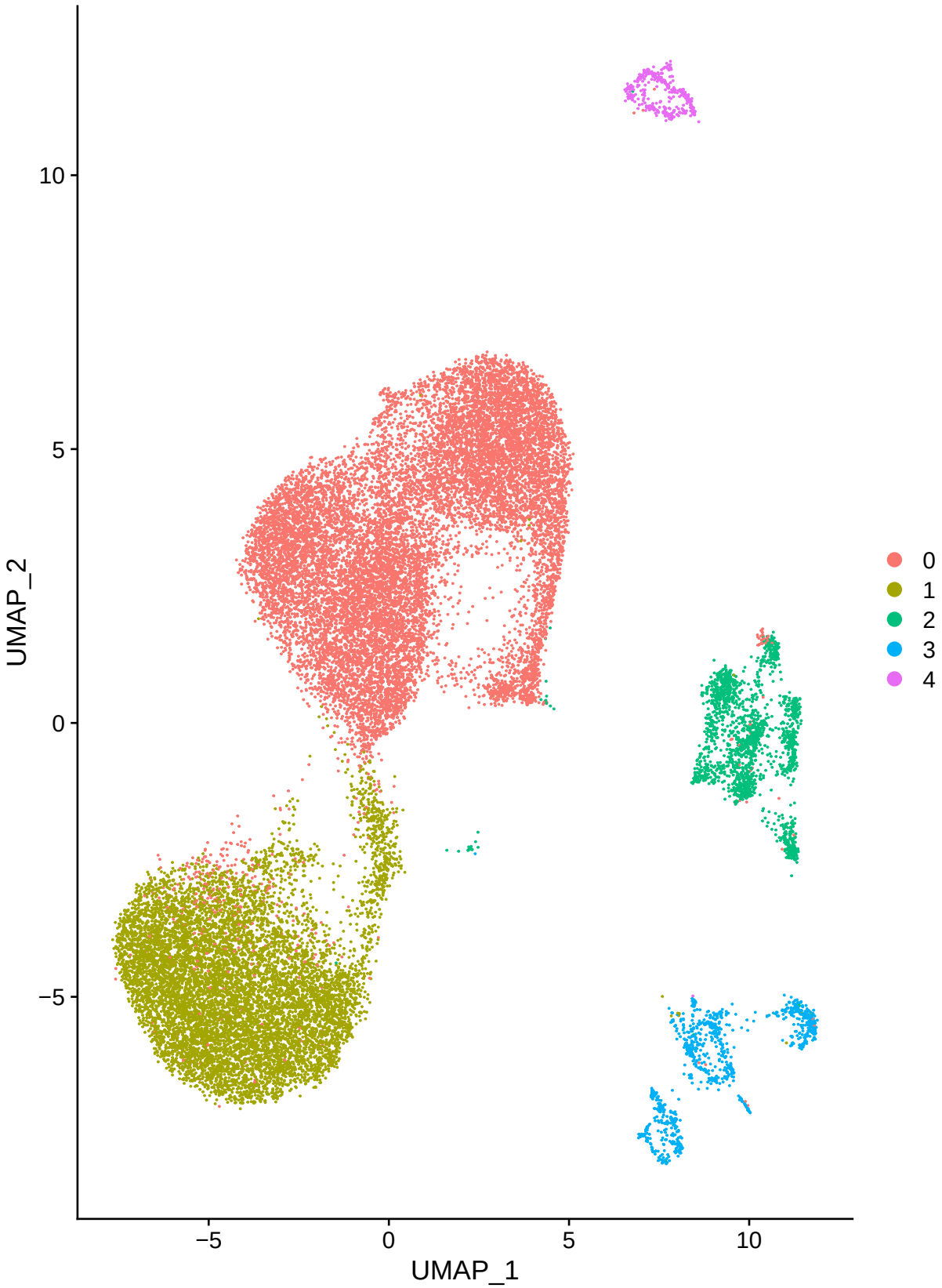
```
load("seurat_objects/FNA_transcriptomes_AnchorIntegrated.RData")

# switch to integrated assay. The variable features of this
# assay are automatically set during IntegrateData
DefaultAssay(fna.integrated) <- "integrated"

# Run the standard workflow for visualization and clustering
fna.integrated <- ScaleData(fna.integrated, verbose = FALSE)
fna.integrated <- RunPCA(fna.integrated, npcs = 20, verbose = FALSE)

fna.integrated <- FindNeighbors(fna.integrated, dims = 1:20)
fna.integrated <- FindClusters(fna.integrated, resolution = 0.05)

fna.integrated <- RunUMAP(fna.integrated, reduction = "pca",
    dims = 1:20)
p1 <- DimPlot(fna.integrated, reduction = "umap", group.by = "day")
p2 <- DimPlot(fna.integrated, reduction = "umap", group.by = "seurat_clusters")
# p1 + p2
```

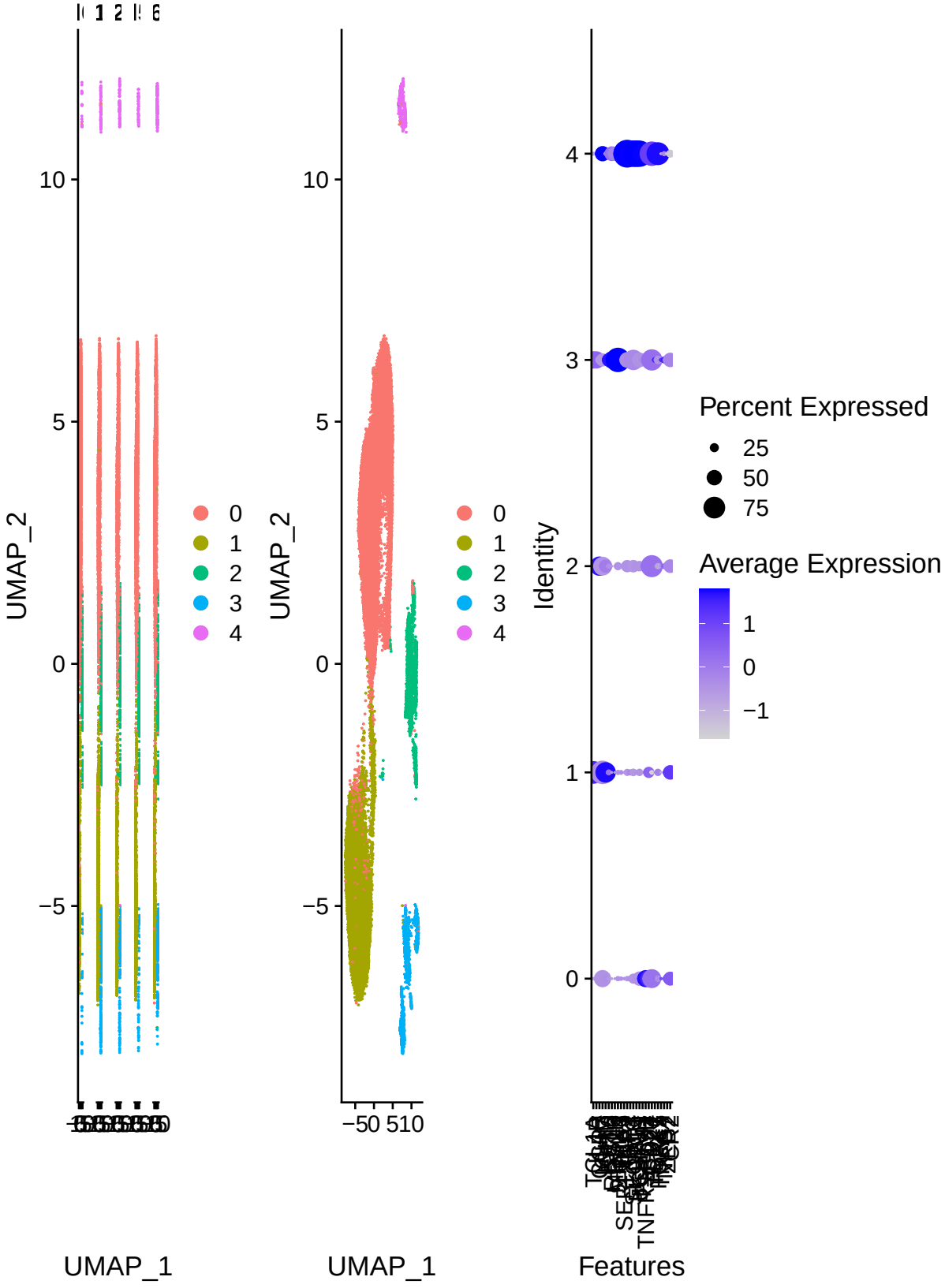
```
# identify B cells Should use the 'RNA' assay for
# visualisation like this.

DefaultAssay(fna.integrated) <- "RNA"
# Normalize RNA data for visualization purposes
fna.integrated <- NormalizeData(fna.integrated, verbose = FALSE)

## Turner et al. use these markers for PBMC/FNA cluster
## assignment:
markers.to.plot <- c("MS4A1", "CD19", "CD79A", "CD3D", "CD3E",
    "CD3G", "IL7R", "CD4", "CD8A", "GZMB", "GNLY", "NCAM1", "CD14",
    "LYZ", "CST3", "MS4A7", "IL3RA", "CLEC4C", "PPBP")


p3 <- DotPlot(fna.integrated, features = c(markers.to.plot)) +
    ggpubr::rotate_x_text()
```

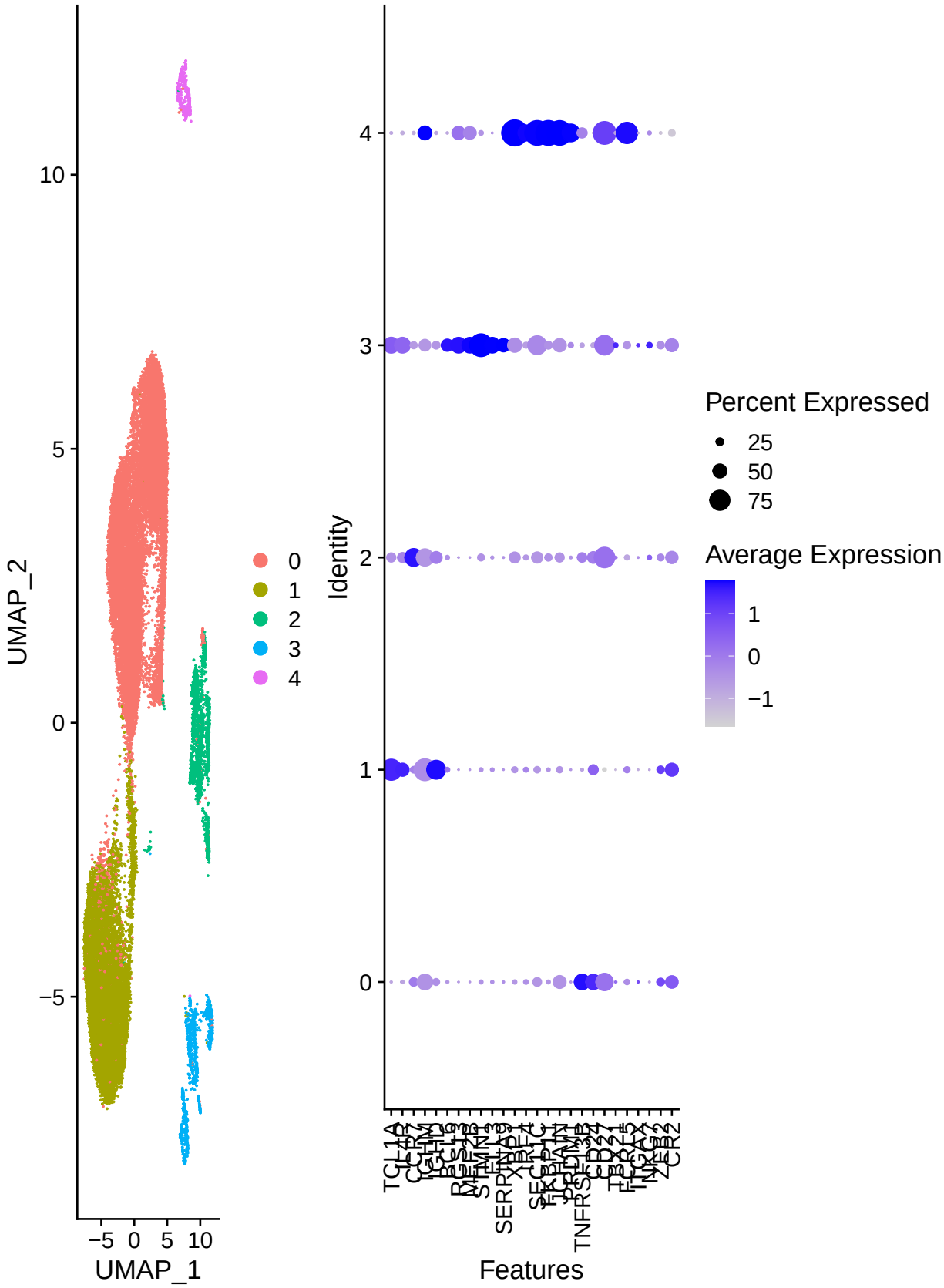
```r
# Clusters 1, 4 and 6 are B cells
fna.B <- subset(fna.integrated, idents = c("1", "4", "6"))

# Turner et al check for housekeeping genes:
# DotPlot(fna.integrated, features = c('ACTB', 'GAPDH',
# 'B2M', 'HSP90AB1', 'GUSB', 'PPIH', 'PGK1', 'TBP', 'TFRC',
# 'SDHA', 'LDHA'))

# Save B cells
DefaultAssay(fna.B) <- "integrated"  # set explicitly, so we re-cluster on integrated anchors.
save(fna.B, file = "seurat_objects/1_FNA_transcriptomes_AnchorIntegrated_just_B_cells.RData")
```

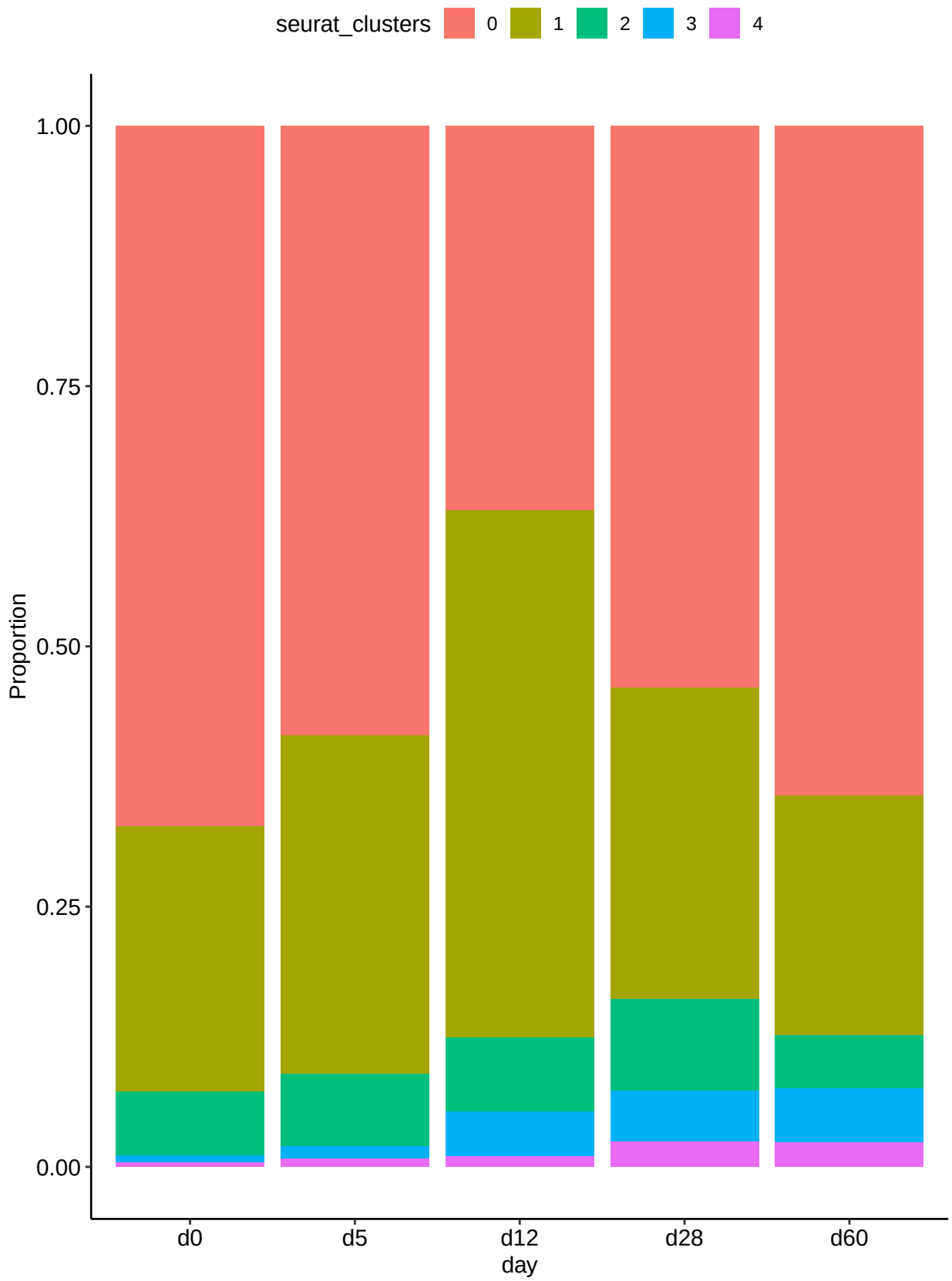
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 27265
## Number of edges: 849631
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9724
## Number of communities: 5
## Elapsed time: 8 seconds
```


d0
d12
d28
d5
d60
UMAP_2
UMAP_1
10
5
0
−5
−5
0
5
10
0
1
2
3
4


10
5
0
−5
UMAP_2
−5
0
5
10
UMAP_1
0
1
2
3
4


UMAP_2
UMAP_1
0
1
2
3
4
Identity
Features
Percent Expressed
25
50
75
Average Expression
1
0
−1


UMAP_2
UMAP_1
10
5
0
−5
−5 0 5 10
0
1
2
3
4
Identity
Features
Percent Expressed
25
50
75
Average Expression
1
0
−1


seurat_clusters
0
1
2
3
4
1.00
0.75
0.50
0.25
0.00
Proportion
d0
d5
d12
d28
d60
day

SessionInfo

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: CentOS Linux 7 (Core)
##
## Matrix products: default
## BLAS:   /bi/apps/R/3.6.1/lib64/R/lib/libRblas.so
## LAPACK: /bi/apps/R/3.6.1/lib64/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C         LC_TIME=C
##  [4] LC_COLLATE=C         LC_MONETARY=C        LC_MESSAGES=C
##  [7] LC_PAPER=C           LC_NAME=C            LC_ADDRESS=C
## [10] LC_TELEPHONE=C       LC_MEASUREMENT=C     LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.4.0   stringr_1.4.0   dplyr_1.0.2     purrr_0.3.3
##  [5] readr_1.3.1     tidyr_1.0.0     tibble_3.0.4    tidyverse_1.3.0
##  [9] patchwork_1.0.0 cowplot_1.0.0   ggplot2_3.3.2   Seurat_3.2.2
##
## loaded via a namespace (and not attached):
##   [1] Rtsne_0.15             colorspace_1.4-1       ggsignif_0.6.0
##   [4] deldir_0.1-23          ellipsis_0.3.0         ggridges_0.5.2
##   [7] fs_1.3.1               rstudioapi_0.10        spatstat.data_1.4-3
##  [10] leiden_0.3.4           ggpubr_0.2.4           listenv_0.8.0
##  [13] farver_2.0.1           ggrepel_0.8.1          fansi_0.4.0
##  [16] lubridate_1.7.4        RSpectra_0.16-0        xml2_1.2.2
##  [19] codetools_0.2-16       splines_3.6.1          knitr_1.26
##  [22] polyclip_1.10-0        jsonlite_1.6           broom_0.7.3
##  [25] ica_1.0-2              dbplyr_1.4.2           cluster_2.1.0
##  [28] png_0.1-7              uwot_0.1.5             shiny_1.4.0
##  [31] sctransform_0.3.2.9000 compiler_3.6.1         httr_1.4.1
##  [34] backports_1.1.5        assertthat_0.2.1       Matrix_1.2-17
##  [37] fastmap_1.0.1          lazyeval_0.2.2         cli_2.0.0
##  [40] later_1.0.0            formatR_1.7            htmltools_0.4.0
##  [43] tools_3.6.1            rsvd_1.0.2             igraph_1.2.4.2
##  [46] gtable_0.3.0           glue_1.4.2             RANN_2.6.1
##  [49] reshape2_1.4.3         Rcpp_1.0.3             spatstat_1.64-1
##  [52] cellranger_1.1.0       vctrs_0.3.6            gdata_2.18.0
##  [55] nlme_3.1-141           lmtest_0.9-37          xfun_0.11
##  [58] globals_0.13.1         rvest_0.3.5            mime_0.7
##  [61] miniUI_0.1.1.1         lifecycle_0.2.0        irlba_2.3.3
##  [64] gtools_3.8.1           goftest_1.2-2          future_1.20.1
##  [67] MASS_7.3-51.4          zoo_1.8-8              scales_1.1.0
##  [70] hms_0.5.2              promises_1.1.0         spatstat.utils_1.17-0
##  [73] parallel_3.6.1         RColorBrewer_1.1-2     yaml_2.2.0
##  [76] reticulate_1.14        pbapply_1.4-2          gridExtra_2.3
##  [79] rpart_4.1-15           stringi_1.4.3          caTools_1.17.1.3
##  [82] rlang_0.4.10           pkgconfig_2.0.3        matrixStats_0.55.0
##  [85] bitops_1.0-6           evaluate_0.14          lattice_0.20-38
```

```
##  [88] ROCR_1.0-7            tensor_1.5            htmlwidgets_1.5.1
##  [91] labeling_0.3          tidyselect_1.1.0      parallelly_1.21.0
##  [94] RcppAnnoy_0.0.14      plyr_1.8.5            magrittr_1.5
##  [97] R6_2.4.1              gplots_3.0.1.1        generics_0.0.2
## [100] DBI_1.1.0             haven_2.2.0           pillar_1.4.7
## [103] withr_2.1.2           mgcv_1.8-28           fitdistrplus_1.1-1
## [106] survival_2.44-1.1     abind_1.4-5           future.apply_1.6.0
## [109] modelr_0.1.5          crayon_1.3.4          KernSmooth_2.23-15
## [112] plotly_4.9.1          rmarkdown_2.0         readxl_1.3.1
## [115] grid_3.6.1            data.table_1.12.8     reprex_0.3.0
## [118] digest_0.6.23         xtable_1.8-4          httpuv_1.5.2
## [121] RcppParallel_4.4.4    munsell_0.5.0         viridisLite_0.3.0
```