SeuratVizHelper Example: Stacked Violin Plot

SeuratVizHelper

2026-01-05

Contents

Introduction

This document demonstrates how to create a Scanpy-style stacked violin plot using the `StackVln` function from the SeuratVizHelper package.

Load Required Libraries

install.packages("devtools")

devtools::install_github("AyumuOkumura/SeuratVizHelper")

```
# Load required packages
library(Seurat)
library(SeuratVizHelper)

# Package required for dataset download
```

```r
if (!requireNamespace("SeuratData", quietly = TRUE)) {
  if (!requireNamespace("remotes", quietly = TRUE)) {
    install.packages("remotes")
  }
  remotes::install_github('satijalab/seurat-data')
}
library(SeuratData)
```

Downloading and loading the dataset

Download and load the `pbmc3k` dataset from the SeuratData package. This dataset is a suitable size for PBMC demonstrations.

```r
SeuratData::InstallData("pbmc3k")
pbmc <- SeuratData::LoadData("pbmc3k")

ndim <- 30
pbmc <- pbmc %>%
  NormalizeData() %>%
  FindVariableFeatures() %>%
  ScaleData() %>%
  RunPCA(verbose = FALSE) %>%
  FindNeighbors(dims = 1:ndim, verbose = FALSE) %>%
  FindClusters(resolution = 0.6, verbose = FALSE) %>%
  RunUMAP(dims = 1:ndim, n.neighbors = 30, verbose = FALSE)

DimPlot(pbmc, reduction = "umap", label = TRUE)
```
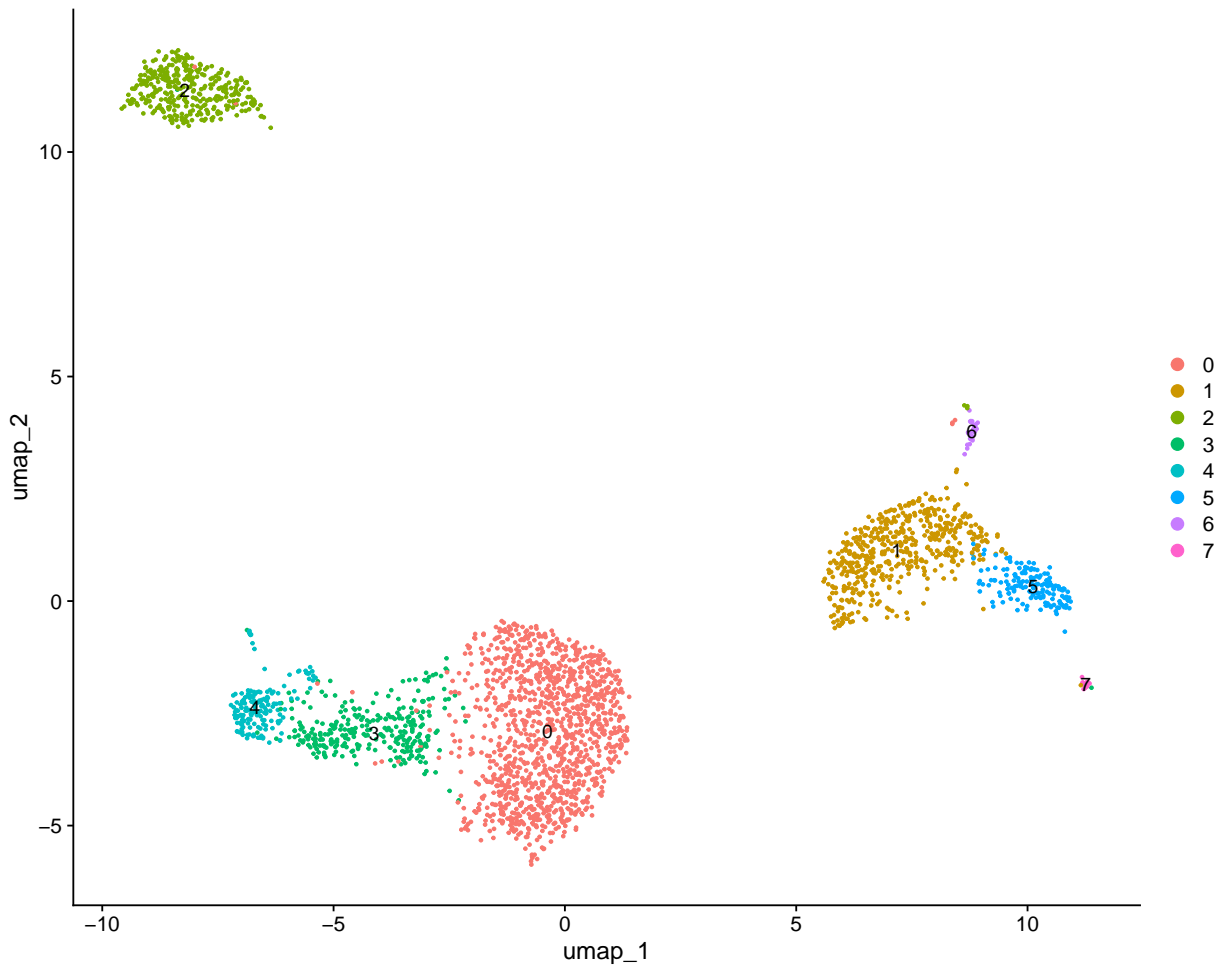
Specifying Marker Genes

Create a StackVln plot using the following marker genes:

```r
# List of marker genes
markers <- c('C1QA', 'PSAP', 'CD79A', 'CD79B', 'CST3', 'LYZ')

# Use only genes present in the dataset
available_markers <- markers[markers %in% rownames(pbmc)]

if (length(available_markers) == 0) {
  # If specified markers are not present, use alternative genes
  cat("Specified marker genes are not included in the dataset.\n")
  cat("Using alternative genes.\n")
  # Select from genes present in the dataset
  available_markers <- head(VariableFeatures(pbmc), 6)
}

cat("Marker genes to use:\n")
```

```
## Marker genes to use:
```
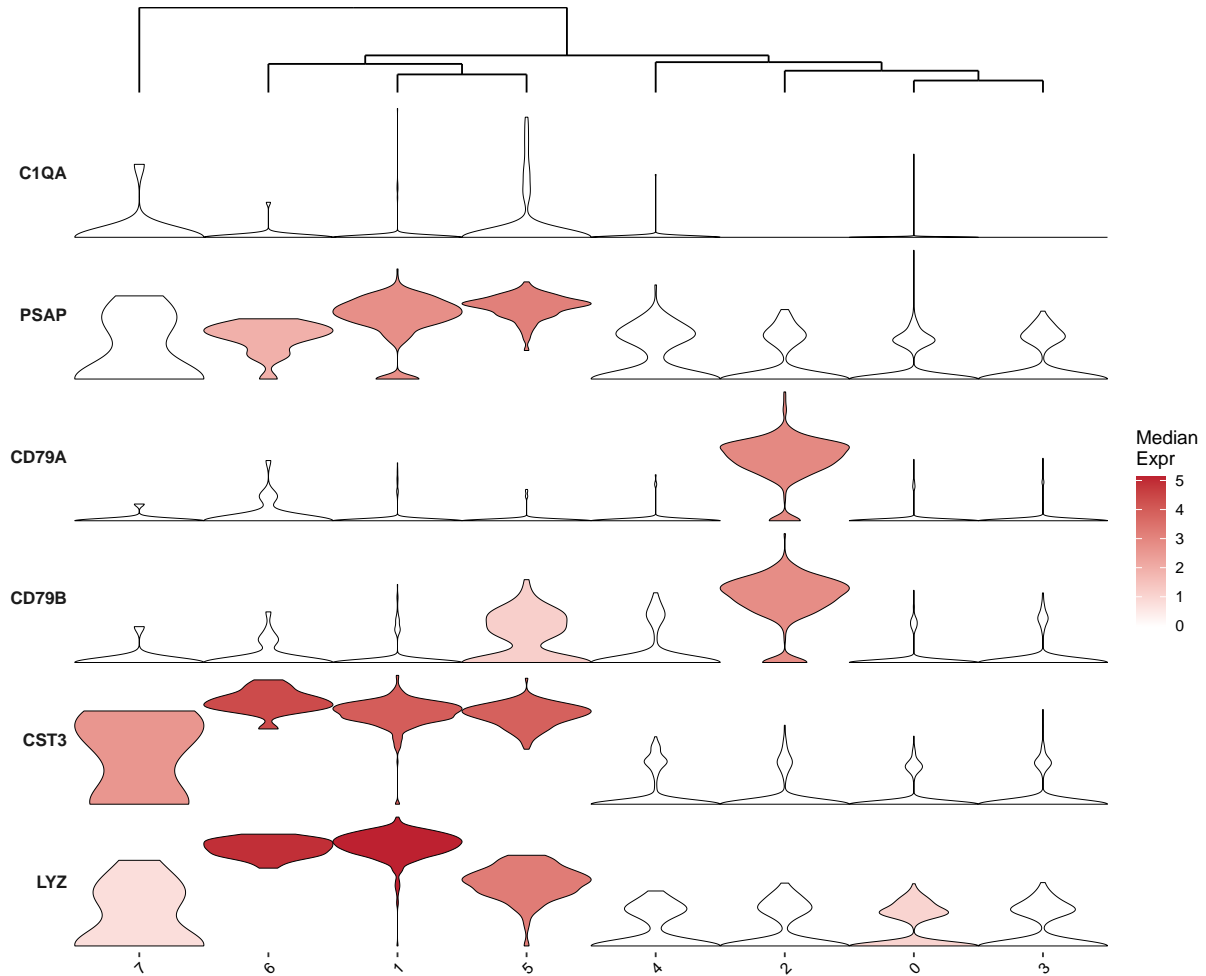
```
print(available_markers)
```

```
## [1] "C1QA"  "PSAP"  "CD79A" "CD79B" "CST3"  "LYZ"
```

Creating the StackVln plot

Use the `StackVln` function to create a Scanpy-style StackVln plot.

```
# Create StackVln plot
StackVln_p <- StackVln(
  seurat_object = pbmc,
  features = available_markers,
  group.by = "seurat_clusters",
  dendrogram_method = "dims",
  ndim = 30,
  plot_width = 5,
  plot_heights = c(0.3, 3),
  # Define the height ratio between the "top graph (dendrogram)" and "bottom graph (violin plot)"
  save_dir = file.path("png/ViolinPlot")
)

StackVln_p
```
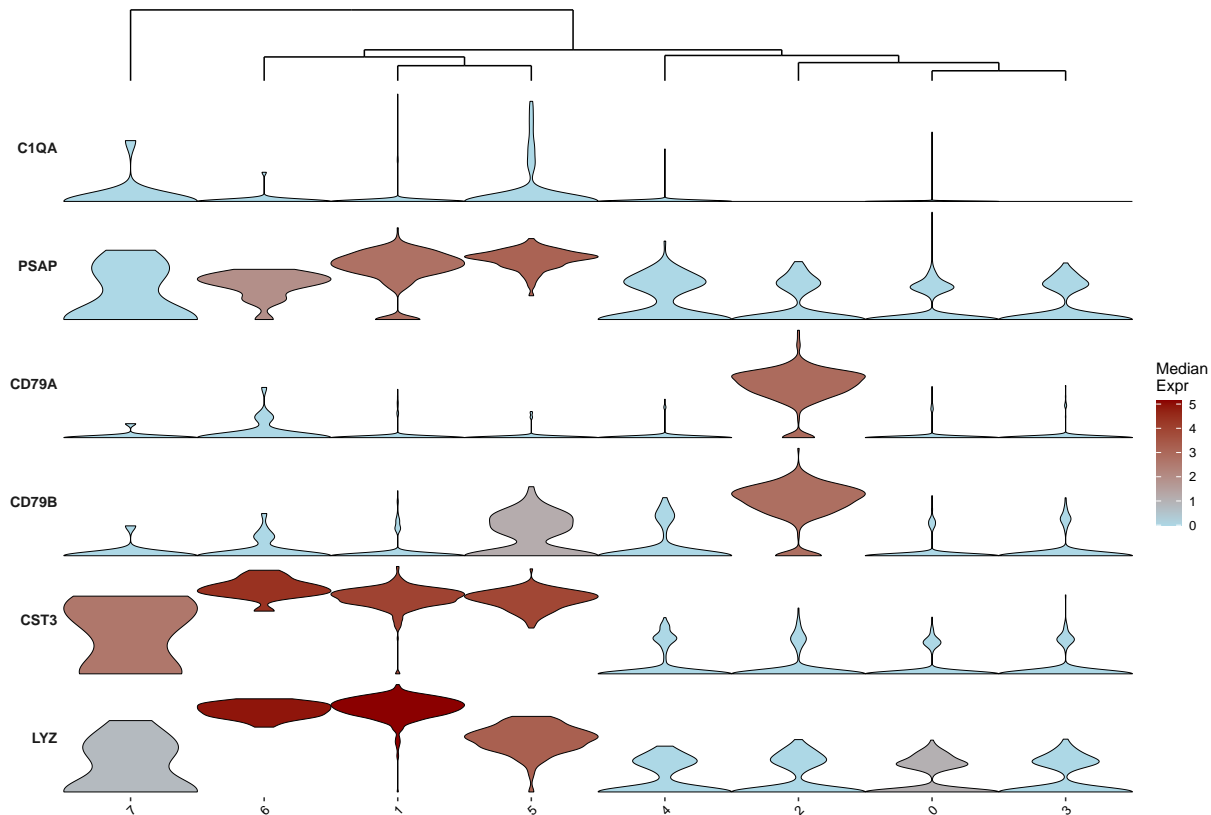
Customized Plot

Example with customized colors:

```r
# Plot with custom colors
StackVln_p <- StackVln(
  seurat_object = pbmc,
  features = available_markers,
  group.by = "seurat_clusters",
  dendrogram_method = "dims",
  ndim = 30,
  plot_width = 5,
  plot_heights = c(0.3, 3),
  # Define the height ratio between the "top graph (dendrogram)" and "bottom graph (violin plot)"
  color_low = 'lightblue',
  color_high = "darkred",
  save_dir = file.path("png/ViolinPlot")
)
StackVln_p
```

Saving Plots

Example of saving a plot to a file (the `plot_width` parameter is applied when saving the file):

```r
# Save the plot to a file
StackVln_p <- StackVln(
  seurat_object = pbmc,
  features = available_markers,
  color_high = "#BD2130",
  plot_width = 12,
  save_dir = "./figures"
)
StackVln_p
```

Session Information

```r
sessionInfo()
```

```
## R version 4.3.3 (2024-02-29 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
```

```
##
##
## locale:
## [1] LC_COLLATE=Japanese_Japan.utf8  LC_CTYPE=Japanese_Japan.utf8
## [3] LC_MONETARY=Japanese_Japan.utf8 LC_NUMERIC=C
## [5] LC_TIME=Japanese_Japan.utf8
##
## time zone: Asia/Tokyo
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] future_1.67.0              pbmc3k.SeuratData_3.1.4
## [3] SeuratData_0.2.2.9001      SeuratVizHelper_0.0.0.9000
## [5] Seurat_5.1.0               SeuratObject_5.2.0
## [7] sp_2.2-0
##
## loaded via a namespace (and not attached):
##   [1] deldir_2.0-4          pbapply_1.7-4         gridExtra_2.3
##   [4] rlang_1.1.6           magrittr_2.0.3        RcppAnnoy_0.0.22
##   [7] otel_0.2.0            spatstat.geom_3.6-0   matrixStats_1.5.0
##  [10] ggridges_0.5.7        compiler_4.3.3        systemfonts_1.3.1
##  [13] png_0.1-8             vctrs_0.6.5           reshape2_1.4.4
##  [16] stringr_1.6.0         crayon_1.5.3          pkgconfig_2.0.3
##  [19] fastmap_1.2.0         labeling_0.4.3        promises_1.5.0
##  [22] rmarkdown_2.30        ragg_1.5.0            purrr_1.2.0
##  [25] xfun_0.54             jsonlite_2.0.0        goftest_1.2-3
##  [28] later_1.4.4           spatstat.utils_3.2-0  irlba_2.3.5.1
##  [31] parallel_4.3.3        cluster_2.1.6         R6_2.6.1
##  [34] ica_1.0-3             stringi_1.8.7         RColorBrewer_1.1-3
##  [37] spatstat.data_3.1-9   reticulate_1.44.0     parallelly_1.45.1
##  [40] spatstat.univar_3.1-4 lmtest_0.9-40         scattermore_1.2
##  [43] Rcpp_1.1.0            knitr_1.50            tensor_1.5.1
##  [46] future.apply_1.20.0   zoo_1.8-14            sctransform_0.4.2
##  [49] httpuv_1.6.16         Matrix_1.6-5          splines_4.3.3
##  [52] igraph_2.2.1          tidyselect_1.2.1      abind_1.4-8
##  [55] rstudioapi_0.17.1     yaml_2.3.10           spatstat.random_3.4-2
##  [58] codetools_0.2-20      miniUI_0.1.2          spatstat.explore_3.5-3
##  [61] listenv_0.10.0        lattice_0.22-6        tibble_3.3.0
##  [64] plyr_1.8.9            withr_3.0.2           shiny_1.11.1
##  [67] ROCR_1.0-11           evaluate_1.0.5        Rtsne_0.17
##  [70] fastDummies_1.7.5     survival_3.5-8        polyclip_1.10-7
##  [73] fitdistrplus_1.2-4    pillar_1.11.1         KernSmooth_2.23-22
##  [76] plotly_4.11.0         generics_0.1.4        RcppHNSW_0.6.0
##  [79] ggplot2_3.5.2         scales_1.4.0          globals_0.18.0
##  [82] xtable_1.8-4          glue_1.8.0            lazyeval_0.2.2
##  [85] tools_4.3.3           data.table_1.17.8     RSpectra_0.16-2
##  [88] RANN_2.6.2            leiden_0.4.3.1        dotCall64_1.2
##  [91] cowplot_1.2.0         grid_4.3.3            ape_5.8-1
##  [94] tidyr_1.3.1           colorspace_2.1-1      nlme_3.1-164
##  [97] patchwork_1.3.2       cli_3.6.5             rappdirs_0.3.3
## [100] spatstat.sparse_3.1-0 textshaping_1.0.4     spam_2.11-1
```

```
## [103] viridisLite_0.4.2      ggdendro_0.2.0       dplyr_1.1.4
## [106] uwot_0.2.3             gtable_0.3.6         digest_0.6.37
## [109] progressr_0.18.0       ggrepel_0.9.6        htmlwidgets_1.6.4
## [112] farver_2.1.2           htmltools_0.5.8.1    lifecycle_1.0.4
## [115] httr_1.4.7             mime_0.13            MASS_7.3-60.0.1
```

Summary

This document demonstrated how to use the SeuratVizHelper package to:

1. Load a PBMC dataset
2. Specify marker genes
3. Create a StackVln plot

For detailed documentation, see the GitHub repository.