

Cell Type Anotation

Cell type annotation with clustering

Ximing Ran

2025-05-15

Contents

| | |
|-------------------------|---|
| 1. Read the merged Data | 2 |
| 2. Culstering | 3 |

```
library(tibble)
library(tidyr)
library(dplyr)
library(rtracklayer)
library(dplyr)
library(Seurat)
library(EnsDb.Hsapiens.v86)
library(ggplot2)
library(cowplot)
library(simspec)
library(cowplot)
library(AnnotationHub)
library(Seurat)
library(tidyverse)
library(openxlsx)
library(dplyr)
library(conflicted)
library(harmony)
library(data.table)
library(pheatmap)
library(ggplot2)
library(patchwork)

conflict_prefer("filter", "dplyr") # Prefer dplyr's filter()
conflict_prefer("select", "dplyr") # Prefer dplyr's filter()
conflict_prefer("lag", "dplyr")    # Prefer dplyr's lag()

# load function from local files
source(here::here("source", "sc_functions.R"))
```

1. Read the merged Data

```
seurat_obj <- readRDS(here::here("data", "synaptosomes_scRNA", "merged_RNA_2025-05-15.rds"))
# seurat_obj<- JoinLayers(seurat_obj)
```

2. Culstering

```
# create a directory to save the results
dir.create("result/01-Clustering/umap_plot", recursive = TRUE, showWarnings = FALSE)

# ct_label <- read.csv(here::here("data", "processed-data", "ct_manual_2025-04-09.csv"), row.names = 1)
# gene_list_clean <- unlist(strsplit(ct_label$marker, ","))
# gene_list_clean <- unique(trimws(gene_list_clean)) # Split by comma
# gene_list_clean <- gene_list_clean[gene_list_clean != ""]
# gene_list_clean <- gene_list_clean[gene_list_clean != ""]

seurat_obj %>%
  FindNeighbors() %>%
  FindClusters(resolution = seq(0.1, 0.5, 0.1)) -> seurat_obj

## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 53211
## Number of edges: 1720528
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9601
## Number of communities: 8
## Elapsed time: 11 seconds
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 53211
## Number of edges: 1720528
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9457
## Number of communities: 10
## Elapsed time: 9 seconds
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 53211
## Number of edges: 1720528
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9352
## Number of communities: 13
## Elapsed time: 9 seconds
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 53211
## Number of edges: 1720528
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9267
## Number of communities: 17
## Elapsed time: 9 seconds
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 53211
```

```

## Number of edges: 1720528
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9203
## Number of communities: 18
## Elapsed time: 9 seconds
# save the seurat object
saveRDS(seurat_obj, file = here::here("data", "synaptosomes_scRNA", "seurat-cluster_2025-05-15.rds"))

#####
## Run clustering res from 0.1 to 0.5 #####
for (res in seq(0.1, 0.5, by = 0.1)) {

  resolution <- sprintf("integrated_snn_res.%s", res)
  seurat_obj$seurat_clusters <- seurat_obj[[resolution]]
  Idents(seurat_obj) <- seurat_obj$seurat_clusters
  dir.create(sprintf("result/01-Clustering/umap_plot/res_%s", res), recursive = TRUE, showWarnings = FALSE)

  p <- DimPlot(seurat_obj,
                reduction = "umap",
                group.by = "seurat_clusters", label = TRUE) +
    ggtitle(paste("Resolution:", res)) +
    theme(legend.position = "none")

  print(p)
  print(sprintf("Resolution: %s", res))
  print(table(seurat_obj$seurat_clusters))

  # Merge all your per-sample layers into counts/data/scale.data...
  seurat_obj_marker <- JoinLayers(seurat_obj, assay = "RNA")

  FindAllMarkers(seurat_obj_marker, assay = "RNA", only.pos = T, densify = T) -> cluster.markers

  write.csv(cluster.markers, file = sprintf("result/01-Clustering/umap_plot/res_%s/cluster_markers_res_%s.csv", res, res))

  dir.create(sprintf("result/01-Clustering/umap_plot/res_%s", res), recursive = TRUE, showWarnings = FALSE)

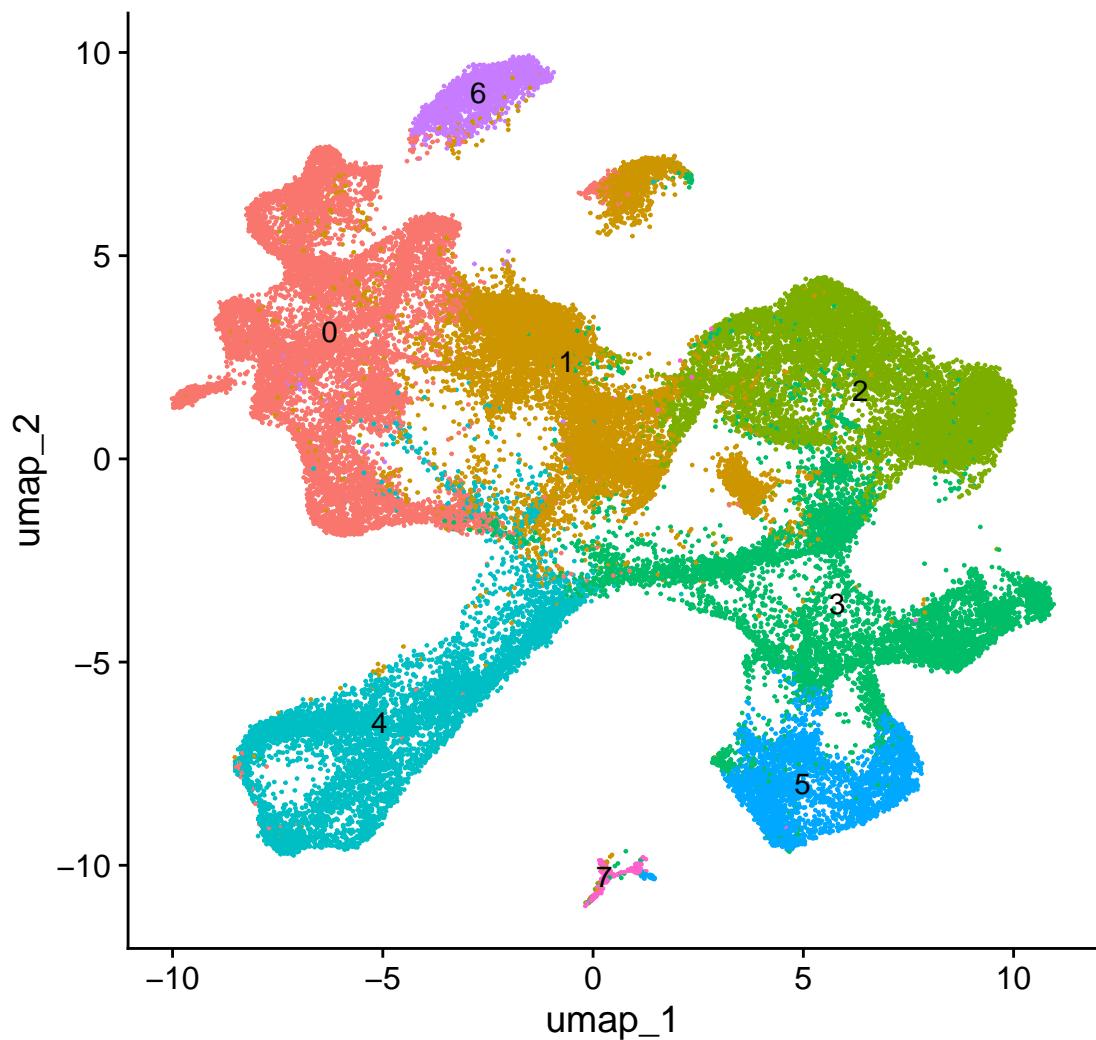
  ggsave(sprintf("result/01-Clustering/umap_plot/res_%s/umap_res_%s.png", res, res), plot = p, width = 8, height = 6)
  ggsave(sprintf("result/01-Clustering/umap_plot/res_%s/umap_res_%s.pdf", res, res), plot = p, width = 8, height = 6)

  # for(index in 1:length(gene_list_clean )) {
  #   gene <- gene_list_clean[index]
  #   file_path <- sprintf("result/01-Clustering/umap_plot/res_%s/%02d_%s", res, index, gene)
  #   p <- plot_sc_feature(seurat_obj, res = res, features = gene, save = TRUE,
  #                         group_by = "seurat_clusters",
  #                         group_by_label = sprintf("seurat_clusters_%s", res),
  #                         reduction_label = "umap",
  #                         output_prefix = file_path)
  # }

```

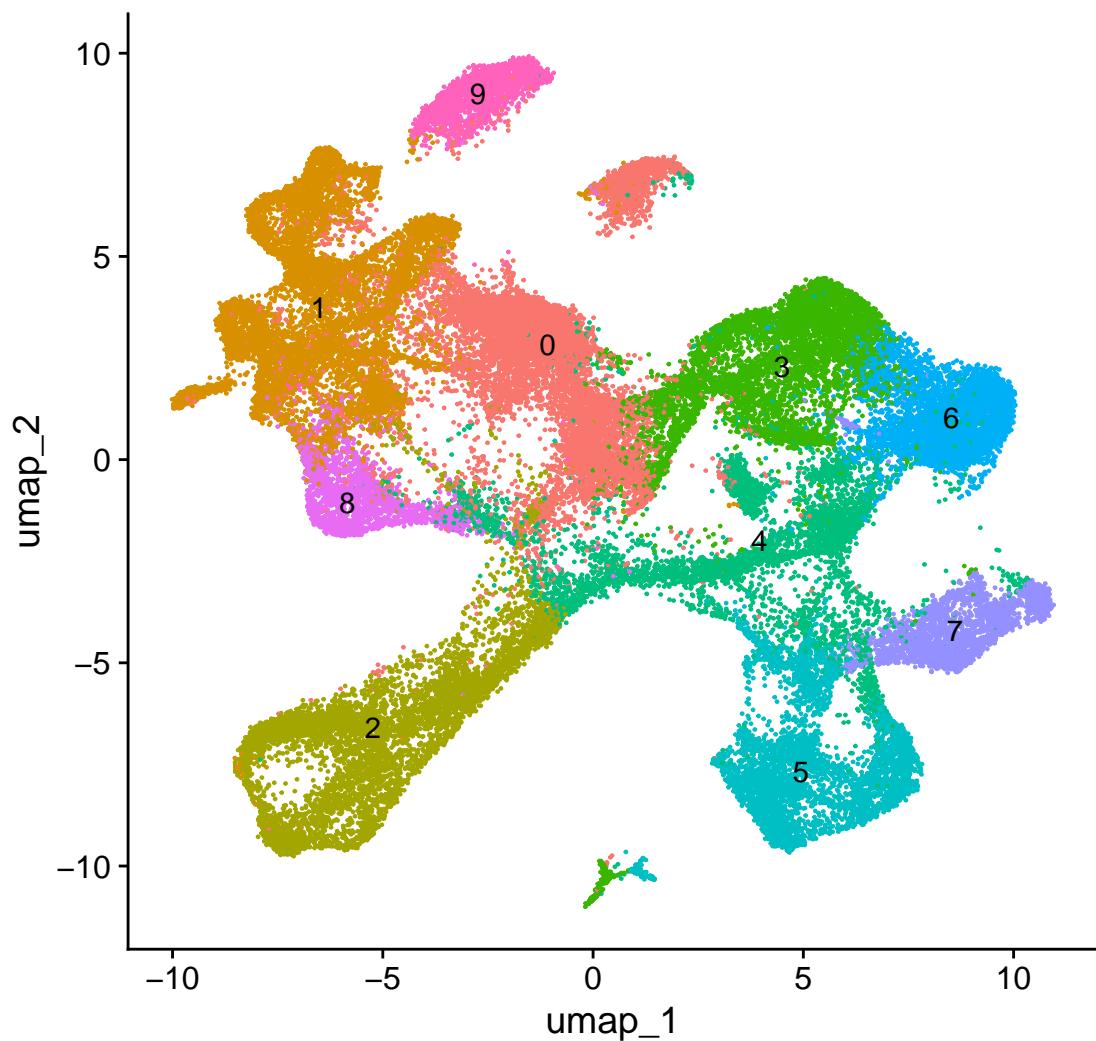
}

Resolution: 0.1



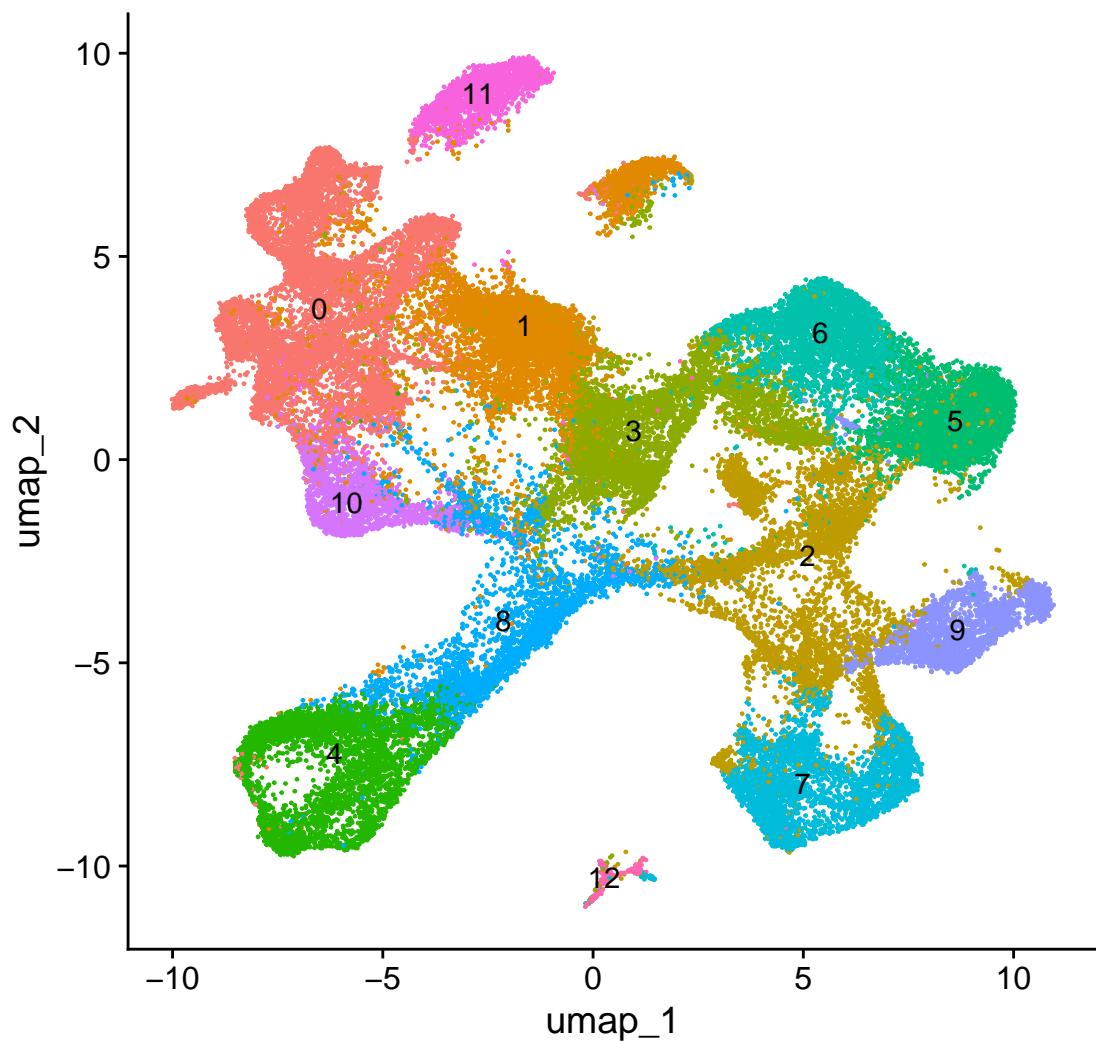
```
## [1] "Resolution: 0.1"
##
##      0     1     2     3     4     5     6     7
## 12785 11300 9195 7510 6654 3504 1960  303
```

Resolution: 0.2



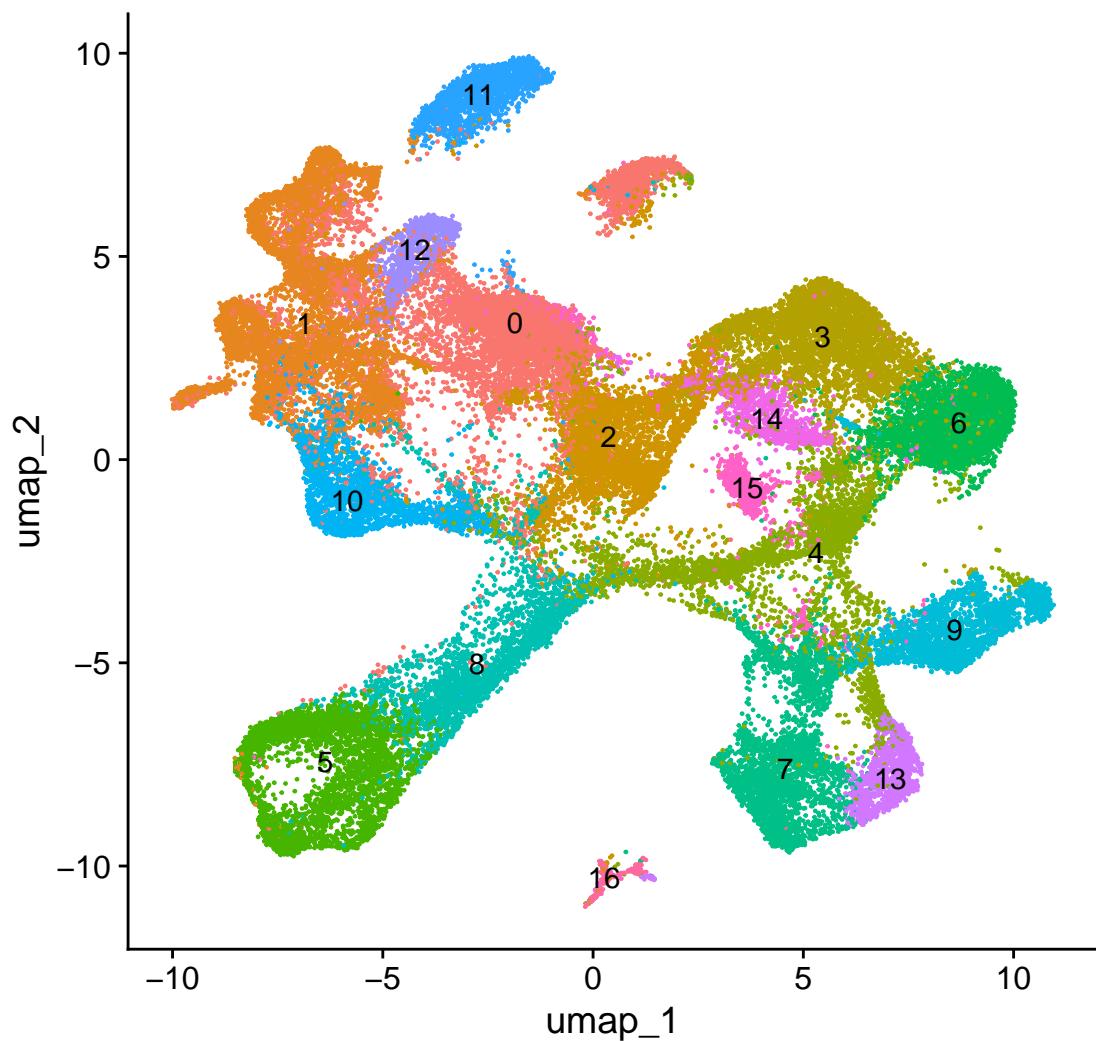
```
## [1] "Resolution: 0.2"  
##  
##      0     1     2     3     4     5     6     7     8     9  
## 10480 9917 6236 5850 5235 4646 4184 2550 2135 1978
```

Resolution: 0.3



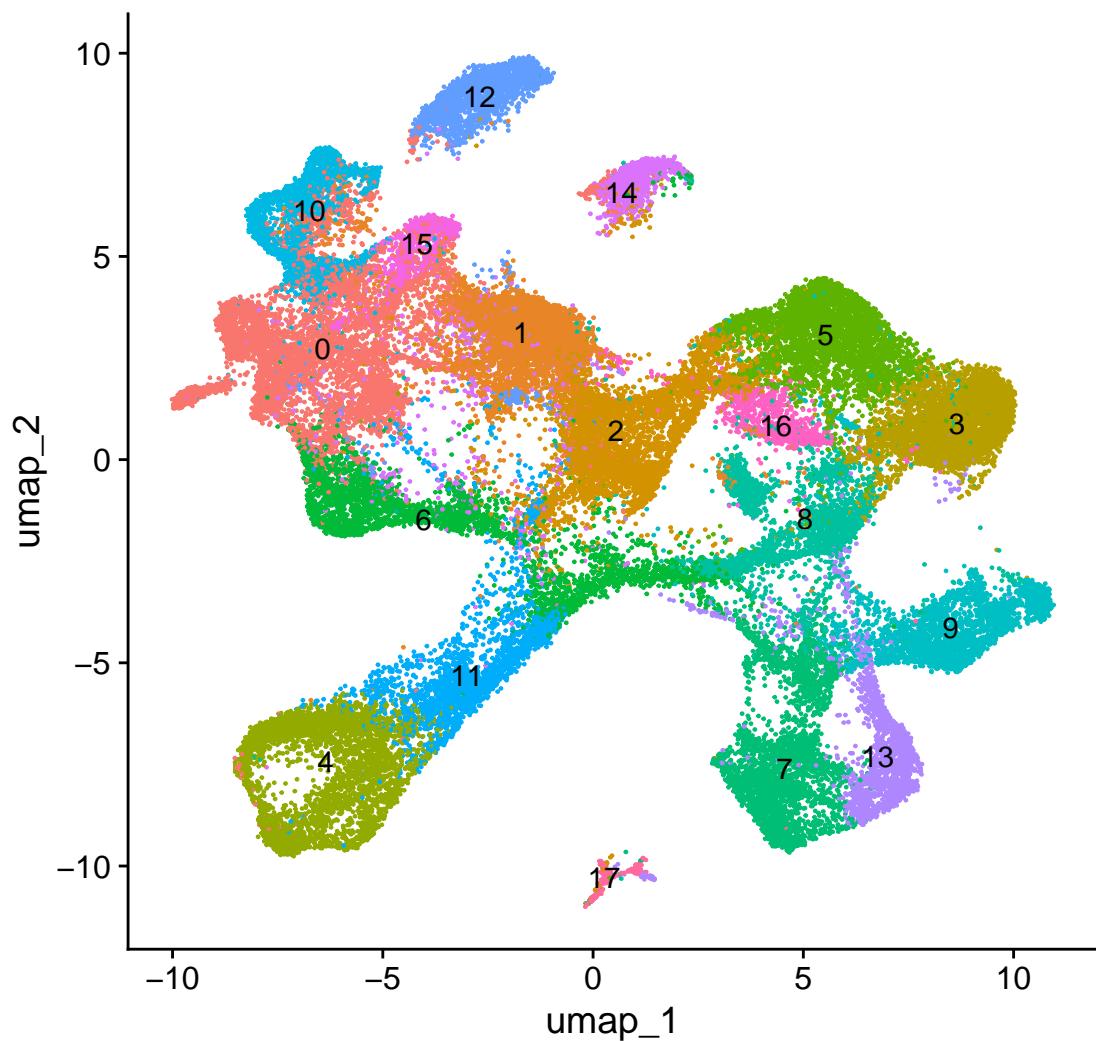
```
## [1] "Resolution: 0.3"  
##  
##      0   1   2   3   4   5   6   7   8   9   10  11  12  
## 9934 7473 5384 5227 4060 3843 3517 3504 3450 2454 2050 2010 305
```

Resolution: 0.4



```
## [1] "Resolution: 0.4"
##
##      0     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
## 8231 7452 4217 3940 3871 3676 3620 3348 2929 2570 2499 2037 1235 1159 1094 1029
##      16
##     304
```

Resolution: 0.5



```
## [1] "Resolution: 0.5"
##
##      0     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
## 7105 4958 4519 3816 3696 3637 3530 3298 2961 2772 2504 2497 2124 1828 1808  963
##      16     17
##    888    307
```

Session Information

```
sessionInfo()

## R version 4.4.0 (2024-04-24)
## Platform: aarch64-apple-darwin20
## Running under: macOS 15.4
##
## Matrix products: default
## BLAS:    /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
## LAPACK:  /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats4      stats       graphics   grDevices  utils      datasets   methods
## [8] base
##
## other attached packages:
##  [1] future_1.49.0           patchwork_1.3.0
##  [3] pheatmap_1.0.12          data.table_1.17.2
##  [5] harmony_1.2.3            Rcpp_1.0.14
##  [7] conflicted_1.2.0          openxlsx_4.2.8
##  [9] AnnotationHub_3.12.0     BiocFileCache_2.12.0
## [11] dbplyr_2.5.0              simspec_0.0.0.9000
## [13] cowplot_1.1.3             EnsDb.Hsapiens.v86_2.99.0
## [15] ensemblDb_2.28.1          AnnotationFilter_1.28.0
## [17] GenomicFeatures_1.56.0     AnnotationDbi_1.66.0
## [19] Biobase_2.64.0             Seurat_5.3.0
## [21] SeuratObject_5.1.0          sp_2.2-0
## [23] rtracklayer_1.64.0         GenomicRanges_1.56.2
## [25] GenomeInfoDb_1.40.1        IRanges_2.38.1
## [27] S4Vectors_0.42.1           BiocGenerics_0.50.0
## [29] knitr_1.50                  lubridate_1.9.4
## [31]forcats_1.0.0                stringr_1.5.1
## [33] dplyr_1.1.4                  purrr_1.0.4
## [35] readr_2.1.5                  tidyverse_2.0.0
## [37] tibble_3.2.1                 ggplot2_3.5.2
## [39] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] RcppAnnoy_0.0.22           splines_4.4.0
##  [3] later_1.4.2                BiocIO_1.14.0
##  [5] filelock_1.0.3              bitops_1.0-9
##  [7] polyclip_1.10-7             XML_3.99-0.18
##  [9] fastDummies_1.7.5           lifecycle_1.0.4
## [11] rprojroot_2.0.4              globals_0.18.0
## [13] lattice_0.22-7              MASS_7.3-65
## [15] magrittr_2.0.3               limma_3.60.6
## [17] plotly_4.10.4                rmarkdown_2.29
## [19] yaml_2.3.10                 httpuv_1.6.16
```

```

## [21] sctransform_0.4.2           zip_2.3.2
## [23] spam_2.11-1                 spatstat.sparse_3.1-0
## [25] reticulate_1.42.0          pbapply_1.7-2
## [27] DBI_1.2.3                  RColorBrewer_1.1-3
## [29] abind_1.4-8                zlibbioc_1.50.0
## [31] Rtsne_0.17                 presto_1.0.0
## [33] RCurl_1.98-1.17           rappdirs_0.3.3
## [35] GenomeInfoDbData_1.2.12    ggrepel_0.9.6
## [37] irlba_2.3.5.1              listenv_0.9.1
## [39] spatstat.utils_3.1-3       goftest_1.2-3
## [41] RSpecSpectra_0.16-2         spatstat.random_3.3-3
## [43] fitdistrplus_1.2-2         parallelly_1.44.0
## [45] codetools_0.2-20           DelayedArray_0.30.1
## [47] tidyselect_1.2.1           UCSC.utils_1.0.0
## [49] farver_2.1.2               matrixStats_1.5.0
## [51] spatstat.explore_3.4-2     GenomicAlignments_1.40.0
## [53] jsonlite_2.0.0              progressr_0.15.1
## [55] ggridges_0.5.6              survival_3.8-3
## [57] systemfonts_1.2.3           tools_4.4.0
## [59] ragg_1.4.0                 ica_1.0-3
## [61] glue_1.8.0                 gridExtra_2.3
## [63] SparseArray_1.4.8            here_1.0.1
## [65] xfun_0.52                  MatrixGenerics_1.16.0
## [67] withr_3.0.2                BiocManager_1.30.25
## [69] fastmap_1.2.0              digest_0.6.37
## [71] timechange_0.3.0            R6_2.6.1
## [73] mime_0.13                  textshaping_1.0.1
## [75] colorspace_2.1-1            scattermore_1.2
## [77] tensor_1.5                 dichromat_2.0-0.1
## [79] spatstat.data_3.1-6         RSQLite_2.3.11
## [81] generics_0.1.4              httr_1.4.7
## [83] htmlwidgets_1.6.4            S4Arrays_1.4.1
## [85] uwot_0.2.3                 pkgconfig_2.0.0.3
## [87] gtable_0.3.6                blob_1.2.4
## [89] lmtest_0.9-40               XVector_0.44.0
## [91] htmltools_0.5.8.1           dotCall164_1.2
## [93] ProtGenerics_1.36.0          scales_1.4.0
## [95] png_0.1-8                  spatstat.univar_3.1-3
## [97] rstudioapi_0.17.1           tzdb_0.5.0
## [99] reshape2_1.4.4              rjson_0.2.23
## [101] nlme_3.1-168              curl_6.2.2
## [103] zoo_1.8-14                cachem_1.1.0
## [105] BiocVersion_3.19.1          KernSmooth_2.23-26
## [107] parallel_4.4.0              miniUI_0.1.2
## [109] restfulr_0.0.15             pillar_1.10.2
## [111] grid_4.4.0                  vctrs_0.6.5
## [113] RANN_2.6.2                 promises_1.3.2
## [115] xtable_1.8-4                cluster_2.1.8.1
## [117] evaluate_1.0.3              tinytex_0.57
## [119] cli_3.6.5                  compiler_4.4.0
## [121] Rsamtools_2.20.0             rlang_1.1.6
## [123] crayon_1.5.3              future.apply_1.11.3
## [125] labeling_0.4.3              plyr_1.8.9
## [127] stringi_1.8.7              viridisLite_0.4.2

```

```
## [129] deldir_2.0-4           BiocParallel_1.38.0
## [131] Biostrings_2.72.1         lazyeval_0.2.2
## [133] spatstat.geom_3.3-6       Matrix_1.7-3
## [135] RcppHNSW_0.6.0            hms_1.1.3
## [137] bit64_4.6.0-1             statmod_1.5.0
## [139] KEGGREST_1.44.1           shiny_1.10.0
## [141] SummarizedExperiment_1.34.0 ROCR_1.0-11
## [143] igraph_2.1.4              memoise_2.0.1
## [145] bit_4.6.0
```