Loading Human scRNAseq V1 V2

Abinet Joan

2024-04-22 11:44:10 +0200

Contents

Loading packages

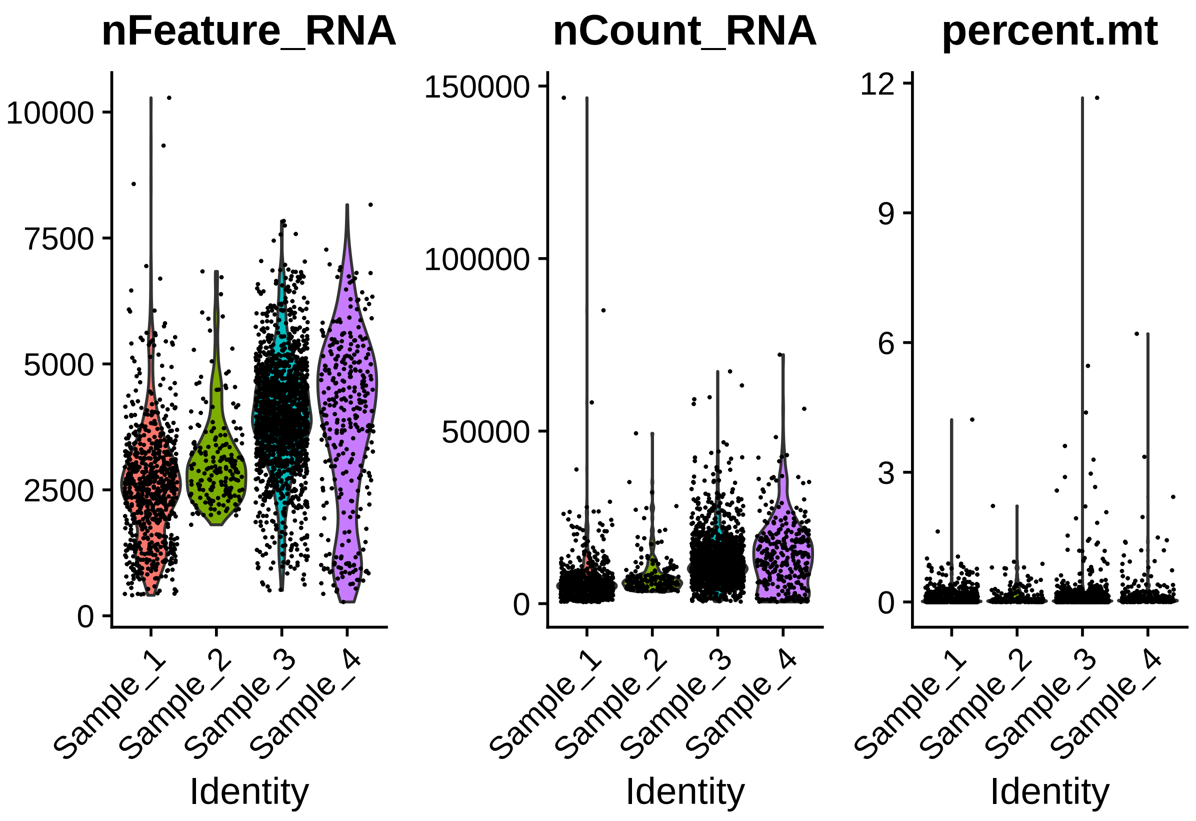
```
suppressMessages(library(dplyr))
suppressMessages(library(Seurat))
suppressMessages(library(patchwork))
suppressMessages(library(ggplot2))
suppressMessages(library(dittoSeq))
```

Loading 10X V1

```
# the data can be downloaded on GEO
all_dirs <- list.dirs(path = "Data", full.names = TRUE, recursive = F)

list_sample_name <- c("Sample_1", "Sample_2", "Sample_3", "Sample_4")

list_sample <- list()
for (i in 1:(length(all_dirs)/2)) {

  Seq_raw_file <- Read10X(data.dir = all_dirs[i])
  Seurat_file <- CreateSeuratObject(counts = Seq_raw_file, project = list_sample_name[i],
min.cells = 3, min.features =   200)
  list_sample <- append(list_sample, Seurat_file)
}
list_sample

V1_10x <- merge(list_sample[[1]], y = list_sample[-1], add.cell.ids = c("1","2","3",
"4"), project = "bronchoalveolar_lavage")
```

Quality control

```
V1_10x[["percent.mt"]] <- PercentageFeatureSet(V1_10x, pattern = "^MT-")# MT : human
cells
VlnPlot(V1_10x, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3,
pt.size = 0.1)
```



Normalisation + scalin

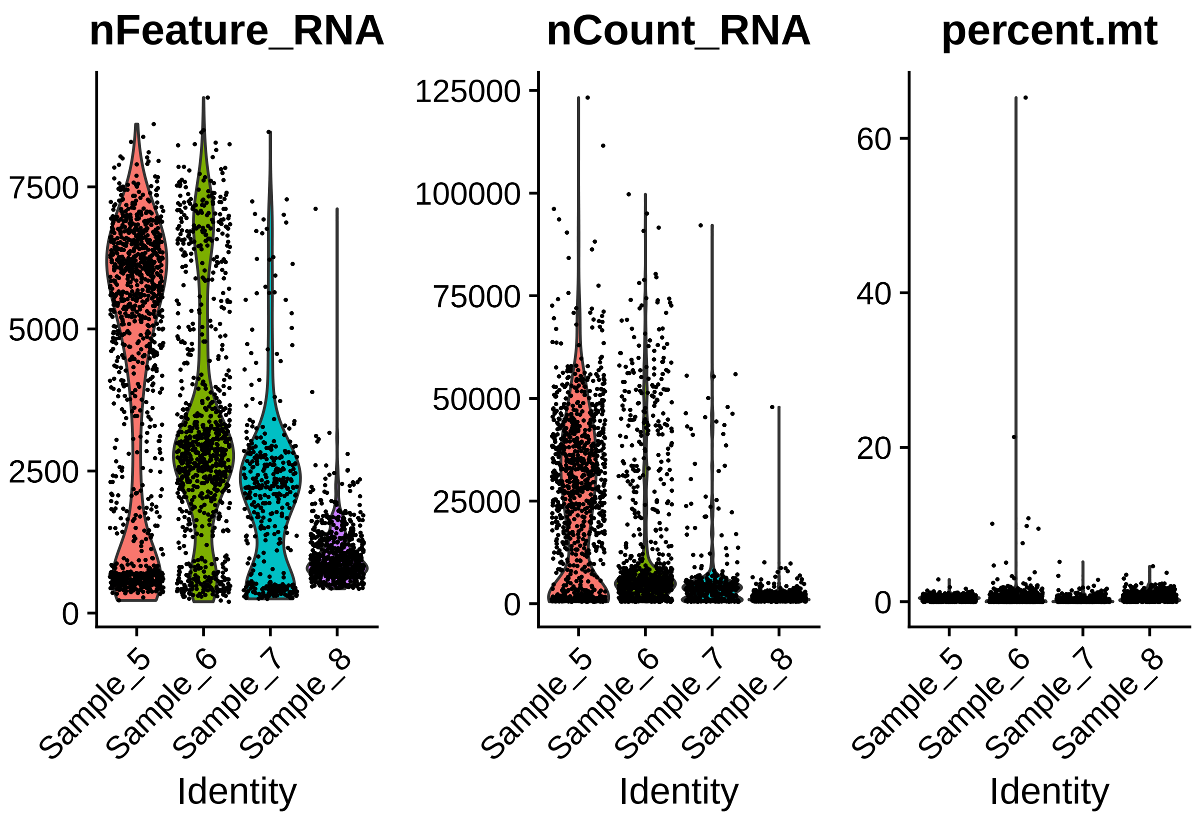
```
V1_10x <- subset(V1_10x, subset = nFeature_RNA > 200 & nFeature_RNA < 6000 & percent.mt <
5)

V1_10x <- NormalizeData(V1_10x, normalization.method = "LogNormalize", scale.factor =
10000)
V1_10x <- FindVariableFeatures(V1_10x, selection.method = "vst", nfeatures = 2000)

all.genes <- rownames(V1_10x)
V1_10x <- ScaleData(V1_10x, features = all.genes)

V1_10x <- RunPCA(V1_10x, features = VariableFeatures(object = V1_10x))
```

Loading 10X V2

```
# the data can be downloaded on GEO
all_dirs <- list.dirs(path = "Data", full.names = TRUE, recursive = F)

list_sample_name <- c("Sample_1", "Sample_2", "Sample_3", "Sample_4","Sample_5",
"Sample_6", "Sample_7", "Sample_8")

list_sample <- list()
for (i in 5:length(all_dirs)) {

  print(all_dirs[i])
  Seq_raw_file <- Read10X(data.dir = all_dirs[i])
  Seurat_file <- CreateSeuratObject(counts = Seq_raw_file, project = list_sample_name[i],
min.cells = 3, min.features =   200)
  list_sample <- append(list_sample, Seurat_file)
}
list_sample

V2_10x <- merge(list_sample[[1]], y = list_sample[-1], add.cell.ids = c("1","2","3",
"4"), project = "bronchoalveolar_lavage")
```

Quality control

```
V2_10x[["percent.mt"]] <- PercentageFeatureSet(V2_10x, pattern = "^MT-")# MT : human
cells
VlnPlot(V2_10x, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3,
pt.size = 0.1)
```



Normalisation + scalin

```
V2_10x <- subset(V2_10x, subset = nFeature_RNA > 400 & nCount_RNA > 800 & nFeature_RNA <
8000 & percent.mt < 5)

V2_10x <- NormalizeData(V2_10x, normalization.method = "LogNormalize", scale.factor =
10000)
V2_10x <- FindVariableFeatures(V2_10x, selection.method = "vst", nfeatures = 2000)

all.genes <- rownames(V2_10x)
V2_10x <- ScaleData(V2_10x, features = all.genes)

V2_10x <- RunPCA(V2_10x, features = VariableFeatures(object = V2_10x))
```

Integration

```
seurat_list <- list(V1_10x, V2_10x)

features <- SelectIntegrationFeatures(object.list = seurat_list)

cells.anchors <- FindIntegrationAnchors(object.list = seurat_list, anchor.features =
features, reduction = "rpca")
cells.combined <- IntegrateData(anchorset = cells.anchors)

cells.combined <- ScaleData(cells.combined, verbose = FALSE)
cells.combined <- RunPCA(cells.combined, npcs = 30, verbose = FALSE)
```

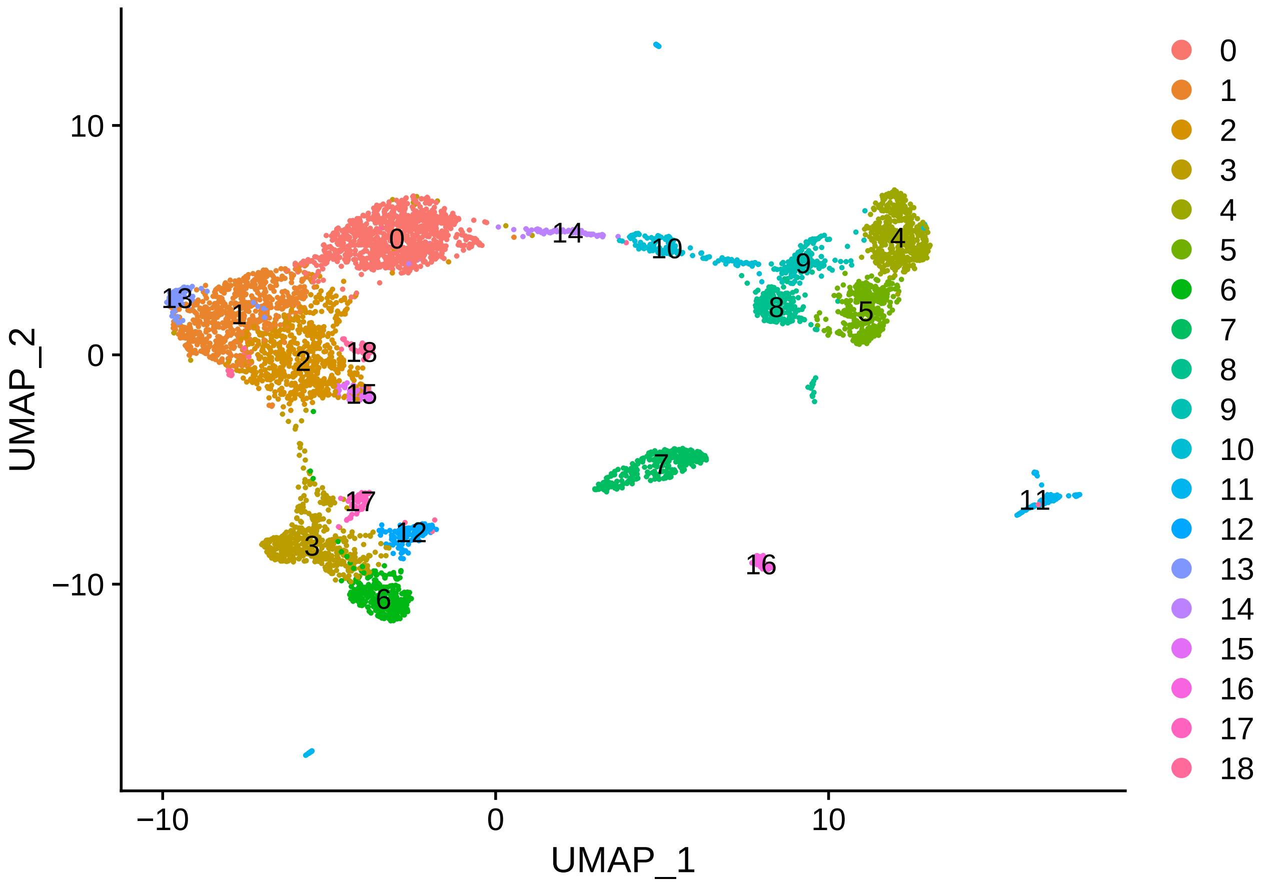
```r
cells.combined <- RunUMAP(cells.combined, reduction = "pca", dims = 1:16)
cells.combined <- FindNeighbors(cells.combined, reduction = "pca", dims = 1:16)
cells.combined <- FindClusters(cells.combined, resolution = 0.9)
```

Seurat doesn't allow us recreate the umap in a reproductible way in this repository. A previous embedding need to be loaded. However, the clustering was not affected.

```r
cells.combined <- subset(cells.combined, seurat_clusters %in% 19, invert = T)

Umap_embedding <- read.csv("umap_coordinate.csv", row.names = 1, header = T)

cells.combined@reductions$umap@cell.embeddings <- as.matrix(Umap_embedding)
```

Umap

```r
DimPlot(cells.combined, reduction = "umap", label = T)
```



Saving Data

```r
saveRDS(cells.combined, "cells.combined_Part1.rds")
```

```r
sessionInfo()
```

```
## R version 4.3.3 (2024-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.20.so;  LAPACK version 3.10.0
##
## 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] dittoSeq_1.12.0    ggplot2_3.4.2      patchwork_1.1.2    SeuratObject_4.1.3
## [5] Seurat_4.3.0       dplyr_1.1.2
##
## loaded via a namespace (and not attached):
##   [1] RColorBrewer_1.1-3          rstudioapi_0.14
##   [3] jsonlite_1.8.7              magrittr_2.0.3
##   [5] spatstat.utils_3.0-3        farver_2.1.1
##   [7] rmarkdown_2.23              zlibbioc_1.46.0
##   [9] vctrs_0.6.3                 ROCR_1.0-11
##  [11] spatstat.explore_3.2-1      RCurl_1.98-1.12
##  [13] S4Arrays_1.0.4              htmltools_0.5.5
##  [15] sctransform_0.3.5           parallelly_1.36.0
##  [17] KernSmooth_2.23-22          htmlwidgets_1.6.2
##  [19] ica_1.0-3                   plyr_1.8.8
##  [21] plotly_4.10.2               zoo_1.8-12
##  [23] igraph_1.5.0.1              mime_0.12
##  [25] lifecycle_1.0.3             pkgconfig_2.0.3
##  [27] Matrix_1.6-1                R6_2.5.1
##  [29] fastmap_1.1.1               GenomeInfoDbData_1.2.10
##  [31] MatrixGenerics_1.12.2       fitdistrplus_1.1-11
##  [33] future_1.33.0               shiny_1.7.4.1
##  [35] digest_0.6.33               colorspace_2.1-0
##  [37] S4Vectors_0.38.1            tensor_1.5
##  [39] irlba_2.3.5.1               GenomicRanges_1.52.0
##  [41] labeling_0.4.2              progressr_0.13.0
##  [43] fansi_1.0.4                 spatstat.sparse_3.0-2
##  [45] httr_1.4.6                  polyclip_1.10-4
##  [47] abind_1.4-5                 compiler_4.3.3
##  [49] withr_2.5.0                 highr_0.10
##  [51] R.utils_2.12.2              MASS_7.3-60
##  [53] DelayedArray_0.26.3         tools_4.3.3
##  [55] lmtest_0.9-40               httpuv_1.6.11
##  [57] future.apply_1.11.0         goftest_1.2-3
##  [59] R.oo_1.25.0                 glue_1.6.2
##  [61] nlme_3.1-163                promises_1.2.0.1
##  [63] grid_4.3.3                  Rtsne_0.16
##  [65] cluster_2.1.6               reshape2_1.4.4
##  [67] generics_0.1.3              gtable_0.3.3
##  [69] spatstat.data_3.0-1         R.methodsS3_1.8.2
##  [71] tidyr_1.3.0                 data.table_1.14.8
##  [73] XVector_0.40.0              sp_2.0-0
##  [75] utf8_1.2.3                  BiocGenerics_0.46.0
##  [77] spatstat.geom_3.2-4         RcppAnnoy_0.0.21
##  [79] ggrepel_0.9.3               RANN_2.6.1
##  [81] pillar_1.9.0                stringr_1.5.0
##  [83] spam_2.9-1                  later_1.3.1
##  [85] splines_4.3.3               lattice_0.22-5
##  [87] survival_3.5-8              deldir_1.0-9
##  [89] tidyselect_1.2.0            SingleCellExperiment_1.22.0
##  [91] miniUI_0.1.1.1              pbapply_1.7-2
##  [93] knitr_1.43                  gridExtra_2.3
##  [95] IRanges_2.34.0              SummarizedExperiment_1.30.2
##  [97] scattermore_1.2             stats4_4.3.3
##  [99] xfun_0.39                   Biobase_2.60.0
## [101] matrixStats_1.0.0           pheatmap_1.0.12
## [103] stringi_1.7.12              lazyeval_0.2.2
## [105] yaml_2.3.7                  evaluate_0.21
## [107] codetools_0.2-19            tibble_3.2.1
## [109] cli_3.6.1                   uwot_0.1.16
## [111] xtable_1.8-4                reticulate_1.30
## [113] munsell_0.5.0               Rcpp_1.0.11
## [115] GenomeInfoDb_1.36.0         globals_0.16.2
## [117] spatstat.random_3.1-5       png_0.1-8
## [119] parallel_4.3.3              ellipsis_0.3.2
## [121] dotCall64_1.0-2             bitops_1.0-7
## [123] listenv_0.9.0               viridisLite_0.4.2
## [125] scales_1.2.1                ggridges_0.5.4
## [127] crayon_1.5.2                leiden_0.4.3
## [129] purrr_1.0.1                 rlang_1.1.1
## [131] cowplot_1.1.1
```