Decoding Intracellular Pathogen of H3N2 at the Single-Cell level using Yeskit

Wei Zhang

2021/09/10

Contents

Taking the in-vitro experiment of H3N2 infection data (SRA Accession number: SRP239555) as an example,

we used PathogenTrack to identify H3N2 infected cells at the single-cell level and used Yeskit to analyze and explore the biological functions that may be related to H3N2 infection.

Install Yeskit from GitHub

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
if (!requireNamespace("devtools", quietly = TRUE))
    BiocManager::install("devtools")
#if (requireNamespace("Yeskit", quietly = TRUE))
#    devtools::install_github("ncrna/Yeskit")
```

Import Yeskit

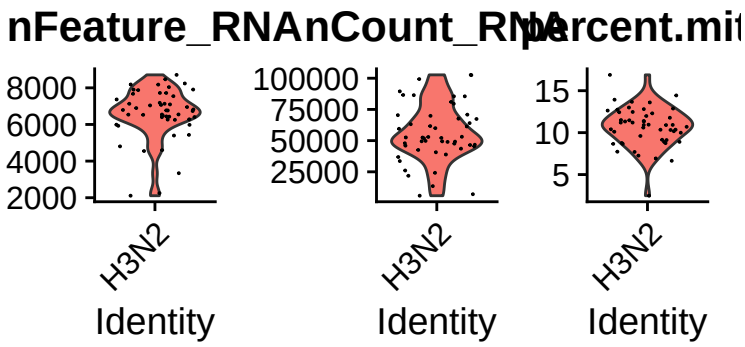
First, we load the package:

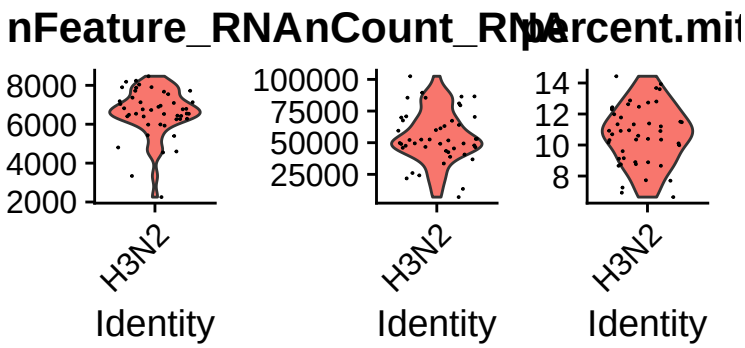
```
library(Yeskit)
library(topGO)
```

Importation

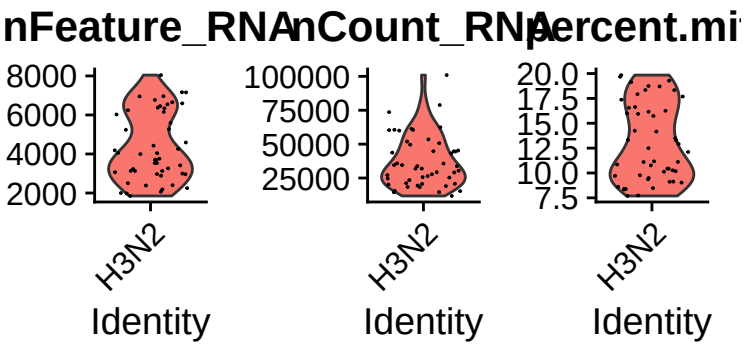
Now, let's load the single-cell count matrix:

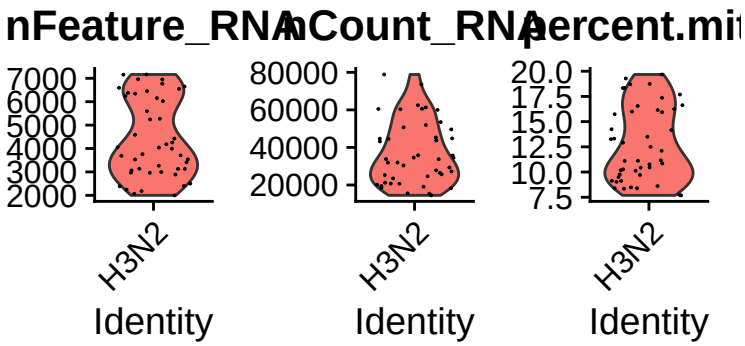
```
Bystander <- scRead(sample_name = "Bystander",
                    data_dir = system.file("extdata/H3N2_10X_matrix/Bystander/",
                                           package="Yeskit"),
                    gene_column = 2, project_name = "H3N2", group_name = "Bystander",
                    meta_file = system.file("extdata/H3N2_10X_matrix/Bystander/microbes.tsv",
                                            package="Yeskit")
                    )
```






```
Infected <- scRead(sample_name = "Infected",
                   data_dir = system.file("extdata/H3N2_10X_matrix/Infected/",
                                          package="Yeskit"),
                   gene_column = 2, project_name = "H3N2", group_name = "Infected",
                   meta_file = system.file("extdata/H3N2_10X_matrix/Infected/microbes.tsv",
                                           package="Yeskit"))
```

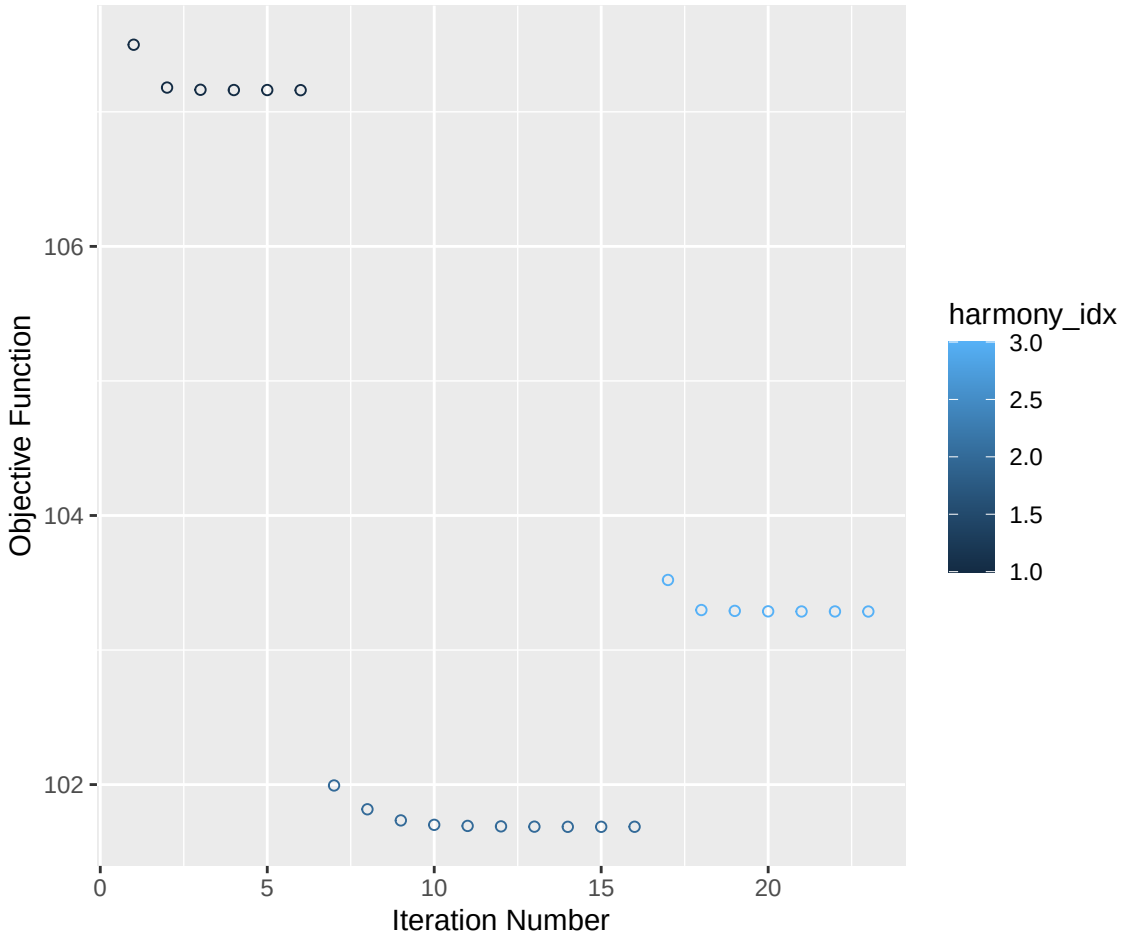




Integration

Then, we integrate these two `Seurat` object

```
H3N2_integrated <- scIntegrate(object.list=list(Bystander, Infected),
                               object.names = c("Bystander", "Infected"),
                               batch.rm = "harmony", res = 0.7)
```




```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 92
## Number of edges: 3856
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.3583
## Number of communities: 2
## Elapsed time: 0 seconds
```

FindMarkers

```
slot(H3N2_integrated, "misc")$Markers <- Seurat::FindAllMarkers(object = H3N2_integrated,
                                                                assay = 'RNA',
                                                                test.use = 'MAST')
```

Differential analysis

Differential analysis between Infected and Bystander

```
slot(H3N2_integrated, "misc")$Infected_vs_Bystander <- scDGE(object = H3N2_integrated,
                                                             comparison = c("Infected", "Bystander"),
                                                             group.by = "group", min.cells = 10,
                                                             logFC = 0.25, clusters = NULL)
```

```
##  done.
##  done.
```

Differential analysis between H3N2_positive and H3N2_negative

```
slot(H3N2_integrated, "misc")$H3N2 <- scPathogenDGE(object = H3N2_integrated,
                                                    species.by = "H3N2", min.cells = 5)
```

GO annotation

GO annotation of Markers

```
slot(H3N2_integrated, "misc")$Markers.GO <- scGO(object = H3N2_integrated,
                                                 key = "Markers",
                                                 logFC = 0.25,
                                                 only.pos = TRUE,
                                                 reference = "human")
```

```
## [1] "Running cluster 0"
## [1] "Running cluster 1"
```

GO annotation of DGEs between Infected and Bystander

```
slot(H3N2_integrated, "misc")$Infected_vs_Bystander.GO <- scGO(object = H3N2_integrated,
                                                               key = "Infected_vs_Bystander",
                                                               logFC = 0.25, only.pos = FALSE,
                                                               reference = "human")
```

```
## [1] "Running cluster 0"
## [1] "Running cluster 1"
```

GO annotation of DGEs between H3N2_positive and H3N2_negative

```
slot(H3N2_integrated, "misc")$H3N2.GO <- scPathogenGO(object = H3N2_integrated, key = "H3N2",
                                                      clusters = NULL, species = "H3N2",
                                                      logFC = 0.25)
```

MSigDB scoring

```
H3N2_integrated <- scMsigdbScoring(object = H3N2_integrated, category = "H", geneSets = NULL)
```

Visualization

Visualization of cell clusters by scDimPlot

```
scDimPlot(object = H3N2_integrated,
          reduction = "umap",
          cols = NULL,
          split.by = "sample",
          ncol = 2,
          pt.size = 2)
```

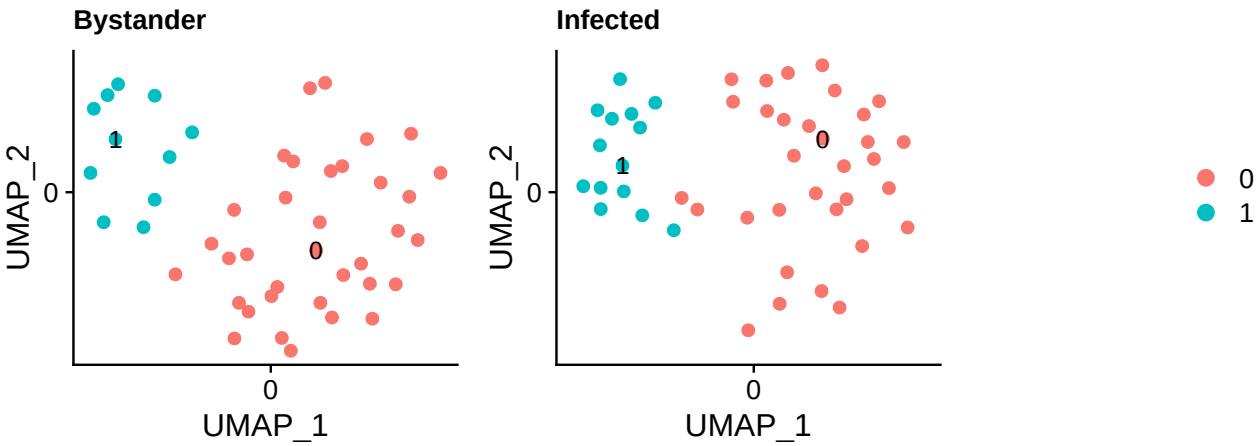


Figure 1: Cell DimPlot

Visualization of cell densities by scDensityPlot

```
scDensityPlot(object = H3N2_integrated,
              reduction = "umap",
              split.by = "sample",
              ncol = 2)
```

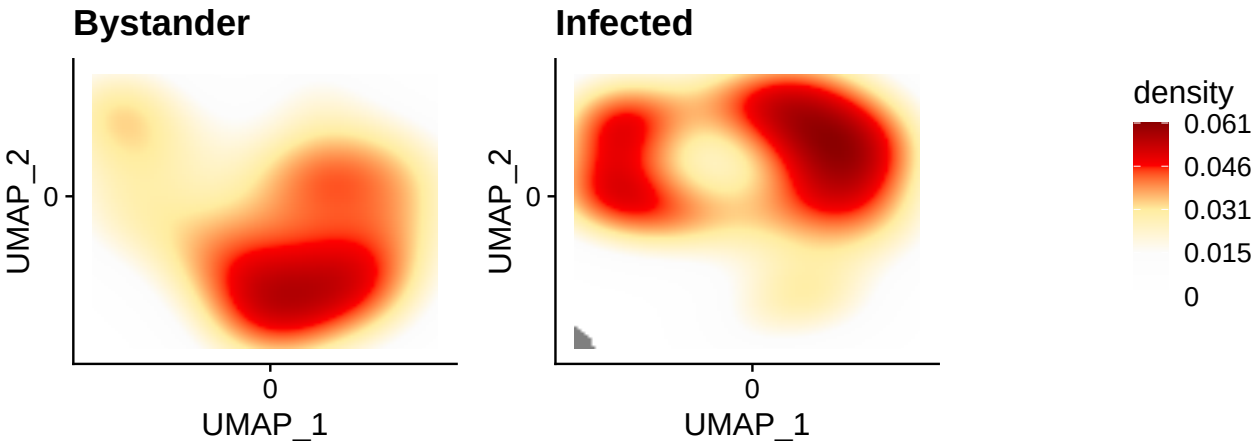


Figure 2: Cell Density Plot

Visualization of cell population fractions by scPopulationPlot, the x axis stands for clusters

```
scPopulationPlot(object = H3N2_integrated,
                 by = "cluster",
                 cols = "sc",
                 order = c("Bystander", "Infected"))
```

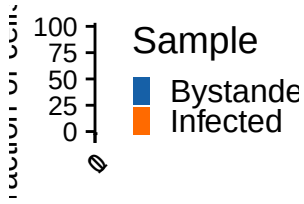


Figure 3: Cell Population Plot by cluster

Visualization of cell population fractions by scPopulationPlot, the x axis stands for samples

```
scPopulationPlot(object = H3N2_integrated,
                 by = "sample",
                 order = c("Bystander", "Infected"))
```

Figure 4: Cell Population Plot by sample

Visualization of meta data by scVizMeta

```
scVizMeta(object = H3N2_integrated,
          reduction = "umap",
          signature="H3N2",
          title = "H3N2",
          raster = TRUE,
          split.by = "sample",
          pt.size = 2,
          interval = c(
            Abundant = 1000,
            Large = 500,
            Medium = 100,
            Small = 10,
            Single = 1,
            None = 0)
          )
```

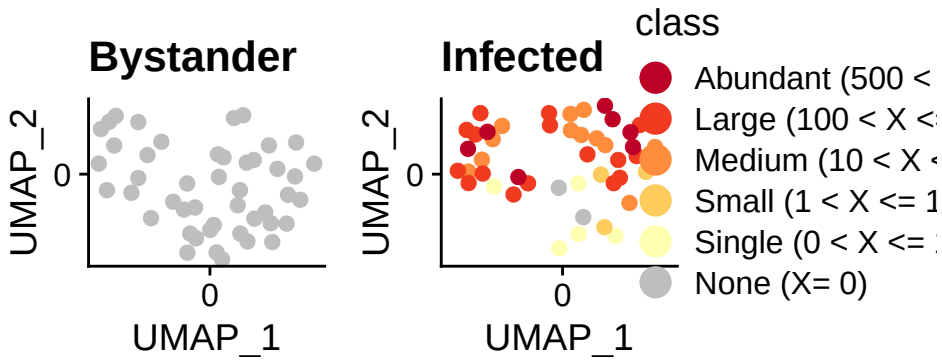


Figure 5: H3N2 DimPlot

Visualization of H3N2-infected cell fractions by scPathogenRatioPlot

```
scPathogenRatioPlot(object = H3N2_integrated,
                    species = "H3N2",
                    split.by = "sample",
                    ncol = 2)
```

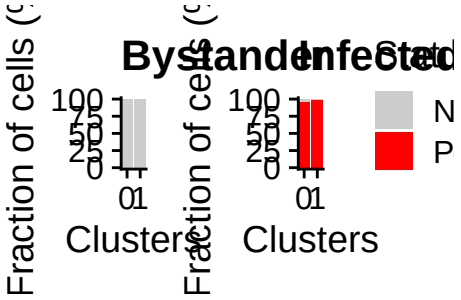


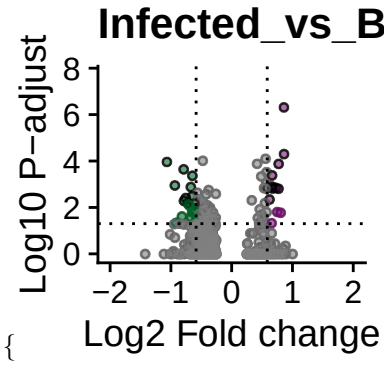
Figure 6: H3N2 Ratio Plot

Visualization of DGEs by scVolcanoPlot

```
scVolcanoPlot(H3N2_integrated,
              key = "Infected_vs_Bystander",
              cluster = "0",
              top_n = 10)
```

```
## Warning: ggrepel: 20 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

\begin{figure}



{

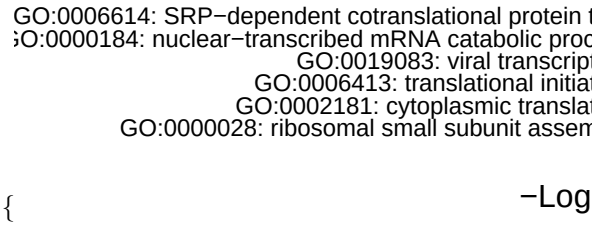
}

\caption{Volcano Plot of Infected_vs_Bystander} \end{figure}

Visualization of enriched GO terms for up-regulated genes by scGOBarPlot

```
scGOBarPlot(object = H3N2_integrated,
            key = "Infected_vs_Bystander.GO",
            ont = "BP",
            top_n = 6,
            direction = "up",
            cluster = "0")
```

\begin{figure}



{

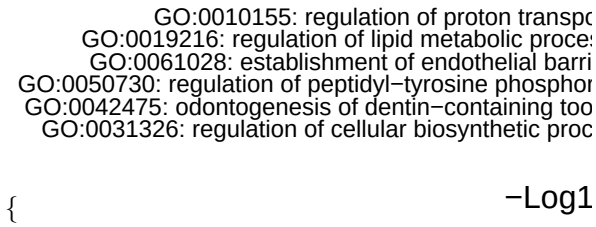
}

\caption{Infected_vs_Bystander.GO_up} \end{figure}

Visualization of enriched GO terms for down-regulated genes by scGOBarPlot

```
scGOBarPlot(object = H3N2_integrated,
            key = "Infected_vs_Bystander.GO",
            ont = "BP",
            top_n = 6,
            direction = "down",
            cluster = "0")
```

\begin{figure}



{

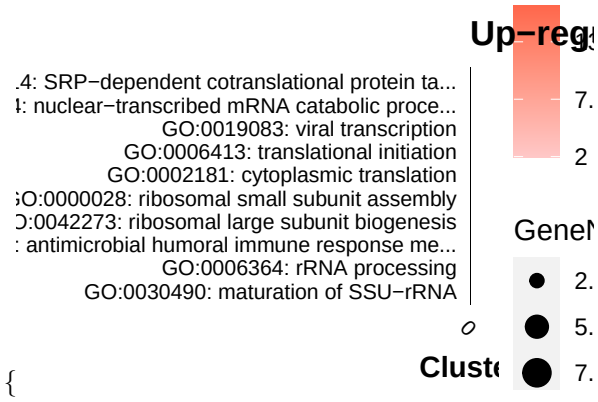
}

\caption{Infected_vs_Bystander.GO_down} \end{figure}

Visualization of enriched GO terms for up-regulated genes by scGODotPlot

```
scGODotPlot(object = H3N2_integrated,
            key = "Infected_vs_Bystander.GO",
            ont = "BP",
            direction = "up",
            top_n = 10,
            font.size = 8)
```

\begin{figure}



{

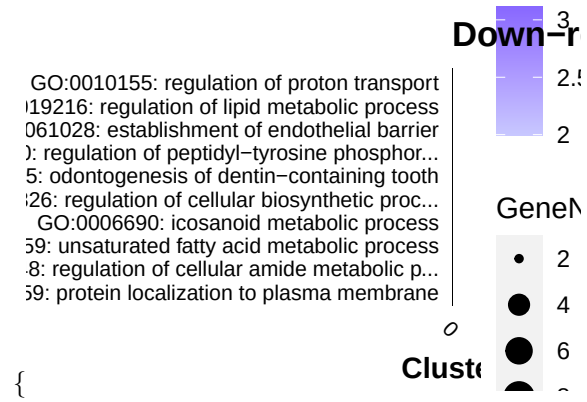
}

\caption{Infected_vs_Bystander.GO_up} \end{figure}

Visualization of enriched GO terms for down-regulated genes by scGODotPlot

```
scGODotPlot(object = H3N2_integrated,
            key = "Infected_vs_Bystander.GO",
            ont = "BP",
            direction = "down",
            top_n = 10,
            clusters = c("0"),
            font.size = 8)
```

\begin{figure}



{

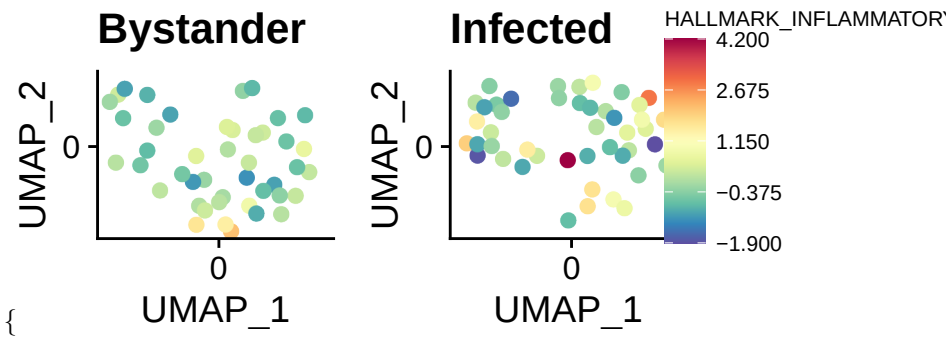
}

\caption{Infected_vs_Bystander.GO_down} \end{figure}

Visualization of HALLMARK_INFLAMMATORY_RESPONSE pathway

```
scScoreDimPlot(H3N2_integrated,
               signature = "HALLMARK_INFLAMMATORY_RESPONSE",
               split.by="sample",
               ncol = 2,
               pt.size = 2)
```

\begin{figure}



{

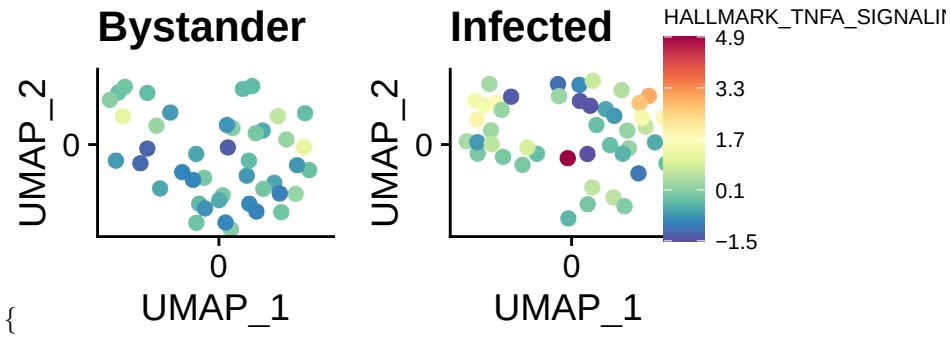
}

\caption{HALLMARK_INFLAMMATORY_RESPONSE} \end{figure}

Visualization of HALLMARK_TNFA_SIGNALING_VIA_NFKB pathway

```
scScoreDimPlot(H3N2_integrated,
               signature = "HALLMARK_TNFA_SIGNALING_VIA_NFKB",
               split.by="sample",
               ncol = 2,
               pt.size = 2)
```

\begin{figure}



{

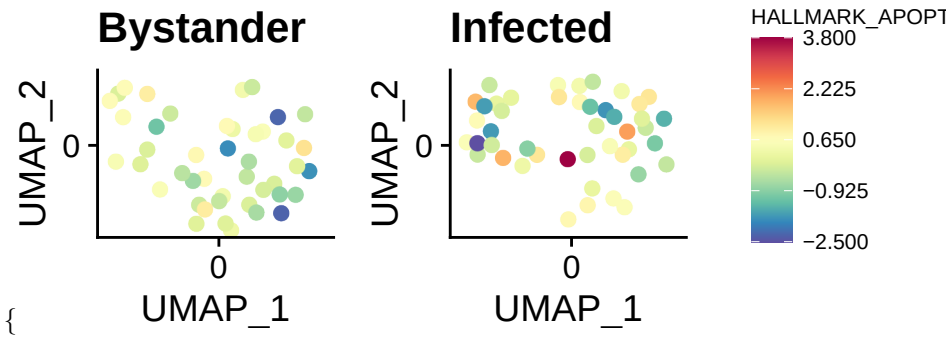
}

\caption{HALLMARK_TNFA_SIGNALING_VIA_NFKB} \end{figure}

Visualization of HALLMARK_APOPTOSIS pathway

```
scScoreDimPlot(H3N2_integrated,
               signature = "HALLMARK_APOPTOSIS",
               split.by="sample",
               ncol = 2,
               pt.size = 2)
```

\begin{figure}



{

}

\caption{HALLMARK_APOPTOSIS} \end{figure}

Session Information

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] zh_CN.UTF-8/zh_CN.UTF-8/zh_CN.UTF-8/C/zh_CN.UTF-8/zh_CN.UTF-8
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] org.Hs.eg.db_3.13.0  topGO_2.44.0         SparseM_1.81
##  [4] GO.db_3.13.0         AnnotationDbi_1.54.1 IRanges_2.26.0
##  [7] S4Vectors_0.30.0     Biobase_2.52.0       graph_1.70.0
## [10] BiocGenerics_0.38.0  Yeskit_0.99.0
##
## loaded via a namespace (and not attached):
##   [1] plyr_1.8.6                  igraph_1.2.6
##   [3] lazyeval_0.2.2              splines_4.1.1
##   [5] listenv_0.8.0               scattermore_0.7
##   [7] GenomeInfoDb_1.28.4         ggplot2_3.3.5
##   [9] digest_0.6.28               htmltools_0.5.2
##  [11] fansi_0.5.0                 magrittr_2.0.1
```

```
##  [13] memoise_2.0.0              tensor_1.5
##  [15] cluster_2.1.2              ROCR_1.0-11
##  [17] globals_0.14.0             Biostrings_2.60.2
##  [19] matrixStats_0.61.0         spatstat.sparse_2.0-0
##  [21] prettyunits_1.1.1          colorspace_2.0-2
##  [23] blob_1.2.2                 ggrepel_0.9.1
##  [25] xfun_0.25                  dplyr_1.0.7
##  [27] crayon_1.4.1               RCurl_1.98-1.4
##  [29] jsonlite_1.7.2             spatstat.data_2.1-0
##  [31] survival_3.2-13            zoo_1.8-9
##  [33] glue_1.4.2                 polyclip_1.10-0
##  [35] gtable_0.3.0               zlibbioc_1.38.0
##  [37] XVector_0.32.0             leiden_0.3.9
##  [39] DelayedArray_0.18.0        SingleCellExperiment_1.14.1
##  [41] future.apply_1.8.1         abind_1.4-5
##  [43] scales_1.1.1               DBI_1.1.1
##  [45] miniUI_0.1.1.1             Rcpp_1.0.7
##  [47] progress_1.2.2             viridisLite_0.4.0
##  [49] xtable_1.8-4               reticulate_1.20
##  [51] spatstat.core_2.3-0        bit_4.0.4
##  [53] htmlwidgets_1.5.4          httr_1.4.2
##  [55] RColorBrewer_1.1-2         ellipsis_0.3.2
##  [57] Seurat_4.0.4               ica_1.0-2
##  [59] pkgconfig_2.0.3            farver_2.1.0
##  [61] uwot_0.1.10                deldir_0.2-10
##  [63] utf8_1.2.2                 tidyselect_1.1.1
##  [65] labeling_0.4.2             rlang_0.4.11
##  [67] reshape2_1.4.4             later_1.3.0
##  [69] munsell_0.5.0              tools_4.1.1
##  [71] cachem_1.0.6               generics_0.1.0
##  [73] RSQLite_2.2.8              ggridges_0.5.3
##  [75] evaluate_0.14              stringr_1.4.0
##  [77] fastmap_1.1.0              yaml_2.2.1
##  [79] goftest_1.2-2              knitr_1.33
##  [81] bit64_4.0.5                fitdistrplus_1.1-5
##  [83] purrr_0.3.4                RANN_2.6.1
##  [85] KEGGREST_1.32.0            pbapply_1.5-0
##  [87] future_1.22.1              nlme_3.1-152
##  [89] mime_0.11                  ggrastr_0.2.3
##  [91] compiler_4.1.1             beeswarm_0.4.0
##  [93] plotly_4.9.4.1             png_0.1-7
##  [95] spatstat.utils_2.2-0       tibble_3.1.4
##  [97] stringi_1.7.4              highr_0.9
##  [99] RSpectra_0.16-0            lattice_0.20-44
## [101] Matrix_1.3-4               vctrs_0.3.8
## [103] pillar_1.6.2               lifecycle_1.0.0
## [105] spatstat.geom_2.2-2        lmtest_0.9-38
## [107] RcppAnnoy_0.0.19           data.table_1.14.0
## [109] cowplot_1.1.1              bitops_1.0-7
## [111] irlba_2.3.3                GenomicRanges_1.44.0
## [113] httpuv_1.6.3               patchwork_1.1.1
## [115] R6_2.5.1                   promises_1.2.0.1
## [117] KernSmooth_2.23-20         gridExtra_2.3
## [119] vipor_0.4.5                parallelly_1.28.1
```

```
## [121] codetools_0.2-18            MASS_7.3-54
## [123] SummarizedExperiment_1.22.0 MAST_1.18.0
## [125] withr_2.4.2                 SeuratObject_4.0.2
## [127] sctransform_0.3.2           harmony_0.1.0
## [129] GenomeInfoDbData_1.2.6      hms_1.1.0
## [131] mgcv_1.8-36                 grid_4.1.1
## [133] rpart_4.1-15                tidyr_1.1.3
## [135] rmarkdown_2.10              MatrixGenerics_1.4.3
## [137] Cairo_1.5-12.2              Rtsne_0.15
## [139] shiny_1.6.0                 ggbeeswarm_0.6.0
```