

scAnno

Introduction

scAnno is an automated annotation tool for single-cell RNA sequencing datasets primarily based on the single cell cluster levels, using a joint deconvolution strategy and logistic regression.

Dependencies

- R version $\geq 3.5.0$.
- R packages: Seurat, dplyr, reticulate, MASS, irlba, future, progress, parallel, glmnet, knitr, rmark-down, devtools

It should be noted that this package requires interactive use of scikit-learn(version $\geq 0.24.2$) package in Python(version ≥ 3.7),so users are required to install Python and scikit-learn package in advance.If Python is not installed, we recommend installing anaconda at <https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>.

After downloading and installing

#Check to see if there is a Python environment

Enter in cmd:

python(Enter key)

exiting python(ctrl + z),input:

#Check if a conda environment is available

conda --version

#Check if a package is available in conda and the version of the package

conda list scikit-learn

#If scikit-learn package does not exist,install the package

conda install scikit-learn

scAnno will first search python in the environment.If python exists,the program will call it directly internally. If not, the program will prompt users to enter a python path.

Using python in R

```
library(reticulate)
```

```
#Check python path
```

```
py.path <- Sys.which("python")
```

```
#Here I use conda, so specify the python in conda
```

```
py.path
```

```
##                                python
## "F:\\\\PROGRA~3\\\\ANACON~1\\\\python.exe"

#Specify the python path
use_python(py.path)

library(scAnno)

##      Seurat

## Attaching SeuratObject

##      dplyr

##
##      'dplyr'

## The following objects are masked from 'package:stats':
##
##      filter, lag

## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union

#Import human cell type reference profile.
data(Human_cell_landscape)

#Import protein coding gene(19814 genes) to filter reference expression profile.
data(gene.anno)

#Import TCGA bulk data in pan-cancer.
data(tcga.data.u)

#A liver tissue data set to be annotated.
data(GSE136103)
```

Set parameters

Parameters	Description
query	Seurat object, which need to be annotated.
ref.expr	Reference gene expression profile.
ref.anno	Cell type information of reference profile, corresponding to the above ref.expr .
save.markers	Specified the filename of makers need to be saved.Default: markers.
cluster.col	Column name of clusters to be annotated in meta.data slot of query Seurat object. Default: seurat_clusters.
factor.size	Factor size for scaling the weight of gene expression. Default: 0.1.

Parameters	Description
seed.num	Number of seed genes of each cell type for recognizing candidate markers, only used when method = 'co.exp'. Default: 10.
redo.markers	Re-search candidate markers or not. Default: FALSE.
gene.anno	Gene annotation data.frame. Default: gene.anno.
permut.num	Number of permutations for estimating p-values of annotations. Default: 100.
show.plot	Show annotated results or not. Default: TRUE.
verbose	Show running messages or not. Default: TRUE.
tcga.data.u	bulk RNA-seq data of pan-cancer in TCGA.

Preparing data for input

```
# Seurat object, which need to be annotated.
obj.seu <- GSE136103

#Seurat object of reference gene expression profile.
ref.obj <- Human_cell_landscape

#Reference gene expression profile.
ref.expr <- GetAssayData(ref.obj, slot = 'data') %>% as.data.frame

#Cell type information of reference profile, corresponding to the above `ref.expr`.
ref.anno <- Idents(ref.obj) %>% as.character
```

scRNA-seq data annotation

```
results = scAnno(query = obj.seu,
  ref.expr = ref.expr,
  ref.anno = ref.anno,
  save.markers = "markers",
  cluster.col = "seurat_clusters",
  factor.size = 0.1,
  pvalue.cut = 0.01,
  seed.num = 10,
  redo.markers = FALSE,
  gene.anno = gene.anno,
  permut.num = 100,
  show.plot = FALSE,
  verbose = TRUE,
  tcga.data.u = tcga.data.u
)

## [INFO] Checking the legality of parameters
## [INFO] 30 cell types in reference, 35 clusters in query objects
## [INFO] Searching candidate marker genes...
## [INFO] Deconvolution by using RLM method
## [INFO] Logistic regression for cell-type predictions, waiting...
```

```
## [INFO] Merging the scores of both models, and assign annotations to clusters
## [INFO] Estimating p-values for annotations...
## [INFO] Finish!
```

Results

Details of the results is described in the table below.

output	details
query	Seurat object, which need to be annotated.
reference	Seurat object of reference gene expression profile.
pred.label	Cell types corresponding to each cluster.
pred.score	The prediction score for each cluster,corresponding to <code>pred.label</code> .
pvals	Significance level of the predicted scores, corresponding to <code>pred.score</code> .

```
results$query
```

```
## An object of class Seurat
## 21898 features across 16036 samples within 1 assay
## Active assay: RNA (21898 features, 2830 variable features)
## 2 dimensional reductions calculated: pca, umap
```

```
results$reference
```

```
## An object of class Seurat
## 17020 features across 5561 samples within 1 assay
## Active assay: RNA (17020 features, 0 variable features)
```

```
results$pred.label
```

```
##          C0          C1          C2
##      "T cell"      "T cell"      "T cell"
##          C3          C4          C5
##      "T cell"      "T cell"      "Dendritic cell"
##          C6          C7          C8
##      "T cell"      "T cell"      "T cell"
##          C9          C10         C11
##      "Monocyte"    "Epithelial cell"    "Macrophage"
##          C12          C13         C14
##      "T cell"      "Endothelial cell"    "Monocyte"
##          C15          C16         C17
##      "Endothelial cell"    "Endothelial cell"    "T cell"
##          C18          C19         C20
##      "Macrophage"    "Smooth muscle cell"    "T cell"
##          C21          C22         C23
##      "Smooth muscle cell"    "B cell"    "Monocyte"
##          C24          C25         C26
```

```
##           "T cell"           "T cell" "B cell (Plasmocyte)"
##           C27             C28             C29
##   "Dendritic cell"   "Endothelial cell"   "Endothelial cell"
##           C30             C31             C32
##           "B cell"       "Stromal cell"   "Endothelial cell"
##           C33             C34
##   "Dendritic cell"   "Epithelial cell"
```

```
results$pred.score
```

```
## [1] 0.9999973 0.9994229 0.9982720 1.0000000 1.0000000 0.9987077 0.9999760
## [8] 0.9988584 0.9987471 0.9997311 0.8919368 0.9956672 0.9999959 0.9989456
## [15] 0.9688465 0.9806819 0.9985180 1.0000000 0.9925140 0.9997775 1.0000000
## [22] 0.9999727 0.9987806 0.6087062 1.0000000 0.9990268 0.9992328 1.0000000
## [29] 0.9986609 0.9993443 0.9852378 0.6264032 0.9825261 1.0000000 1.0000000
```

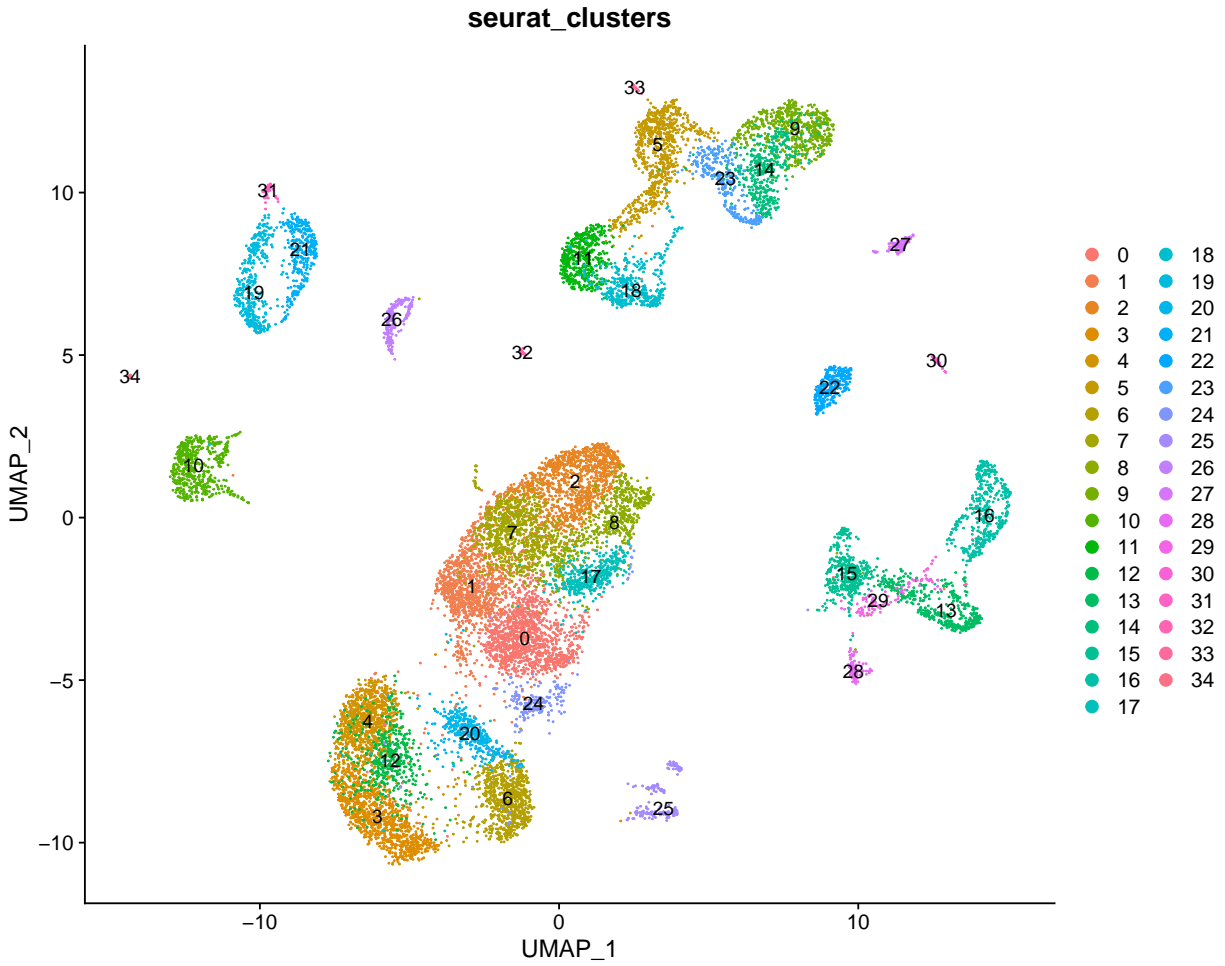
```
results$pvals
```

```
##           C0           C1           C2           C3           C4           C5
## 1.742825e-15 4.596242e-15 3.069306e-14 1.734694e-15 1.734694e-15 0.000000e+00
##           C6           C7           C8           C9           C10          C11
## 1.807108e-15 1.175180e-14 1.411631e-14 0.000000e+00 0.000000e+00 0.000000e+00
##           C12          C13          C14          C15          C16          C17
## 1.746914e-15 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 1.734694e-15
##           C18          C19          C20          C21          C22          C23
## 0.000000e+00 0.000000e+00 1.734694e-15 0.000000e+00 0.000000e+00 0.000000e+00
##           C24          C25          C26          C27          C28          C29
## 1.734694e-15 8.894709e-15 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
##           C30          C31          C32          C33          C34
## 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
```

Visualization

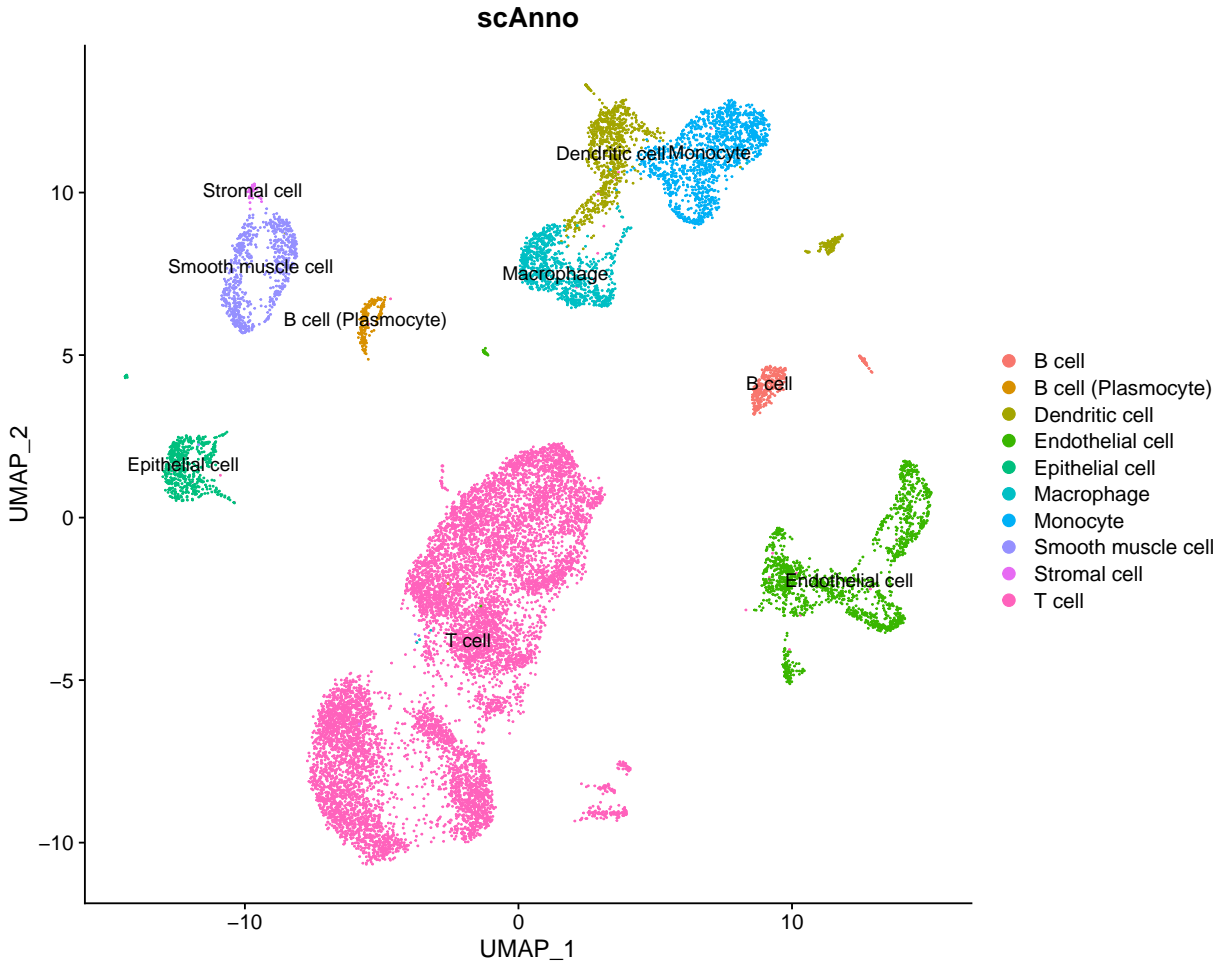
We plot query object in the two-dimensional space of UMAP based on the clustering information of `seurat_cluster`, which contains a total of 35 clusters.

```
DimPlot(results$query, group.by = "seurat_clusters", label = TRUE)
```



scAnno annotation results visualization,the identity of each cell in the scAnno column of the metadata of query Seurat object.

```
DimPlot(results$query, group.by = 'scAnno', label = TRUE)
```



sessionInfo()

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.936
## [2] LC_CTYPE=Chinese (Simplified)_China.936
## [3] LC_MONETARY=Chinese (Simplified)_China.936
## [4] LC_NUMERIC=C
## [5] LC_TIME=Chinese (Simplified)_China.936
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] scAnno_1.0.0      dplyr_1.0.10      SeuratObject_4.1.3 Seurat_4.3.0
```

```

## [5] reticulate_1.26
##
## loaded via a namespace (and not attached):
##   [1] Rtsne_0.16           colorspace_2.0-3      deldir_1.0-6
##   [4] ellipsis_0.3.2       ggribges_0.5.4        rstudioapi_0.14
##   [7] spatstat.data_3.0-1  farver_2.1.1          leiden_0.4.3
##  [10] listenv_0.9.0        ggrepel_0.9.2         fansi_1.0.3
##  [13] codetools_0.2-19     splines_4.1.1         knitr_1.42
##  [16] polyclip_1.10-4      jsonlite_1.8.3        ica_1.0-3
##  [19] cluster_2.1.3        png_0.1-7             uwot_0.1.14
##  [22] shiny_1.7.4          sctransform_0.3.5     spatstat.sparse_3.0-0
##  [25] compiler_4.1.1       httr_1.4.5            Matrix_1.5-1
##  [28] fastmap_1.1.0        lazyeval_0.2.2        cli_3.6.0
##  [31] later_1.3.0          prettyunits_1.1.1     htmltools_0.5.4
##  [34] tools_4.1.1          igraph_1.3.5          gtable_0.3.1
##  [37] glue_1.6.2           RANN_2.6.1            reshape2_1.4.4
##  [40] Rcpp_1.0.9           scattermore_0.8        vctrs_0.5.0
##  [43] nlme_3.1-157         spatstat.explore_3.0-5 progressr_0.13.0
##  [46] lmtest_0.9-40        spatstat.random_3.0-1 xfun_0.36
##  [49] stringr_1.5.0        globals_0.16.2        mime_0.12
##  [52] miniUI_0.1.1.1       lifecycle_1.0.3       irlba_2.3.5.1
##  [55] goftest_1.2-3        future_1.32.0          MASS_7.3-57
##  [58] zoo_1.8-11           scales_1.2.1          hms_1.1.2
##  [61] promises_1.2.0.1     spatstat.utils_3.0-2  parallel_4.1.1
##  [64] RColorBrewer_1.1-3   yaml_2.3.6            pbapply_1.7-0
##  [67] gridExtra_2.3        ggplot2_3.4.1         stringi_1.7.8
##  [70] highr_0.10           rlang_1.0.6           pkgconfig_2.0.3
##  [73] matrixStats_0.62.0   evaluate_0.20         lattice_0.20-45
##  [76] ROCR_1.0-11          purrr_0.3.4           tensor_1.5
##  [79] labeling_0.4.2       patchwork_1.1.2        htmlwidgets_1.6.1
##  [82] cowplot_1.1.1        tidyselect_1.2.0      parallelly_1.34.0
##  [85] RcppAnnoy_0.0.20     plyr_1.8.7            magrittr_2.0.3
##  [88] R6_2.5.1             generics_0.1.3        DBI_1.1.3
##  [91] withr_2.5.0          pillar_1.8.1          fitdistrplus_1.1-8
##  [94] survival_3.3-1       abind_1.4-5           sp_1.5-1
##  [97] tibble_3.1.8         future.apply_1.10.0   crayon_1.5.2
## [100] KernSmooth_2.23-20   utf8_1.2.2            spatstat.geom_3.0-3
## [103] plotly_4.10.1        rmarkdown_2.20        progress_1.2.2
## [106] grid_4.1.1           data.table_1.14.4     digest_0.6.30
## [109] xtable_1.8-4         tidyr_1.2.1           httpuv_1.6.6
## [112] munsell_0.5.0        viridisLite_0.4.1

```