seneR Workflow

Overview

seneR is an R package that provides the senescence scoring framework based on the pretrained SenCID model by Tao *et al.*. It supports both bulk RNA-seq and single-cell RNA-seq data and includes visualization and trajectory analysis tools.

Important note for users:
This vignette shows all example code **but Python-dependent steps (e.g., SenCID prediction, trajectory inference, GSVA scoring) are set to `eval=FALSE`**.
This is required because:

- Vignettes built by CRAN and GitHub Actions cannot install Python modules (e.g., sklearn, numpy, joblib).
- Reticulate cannot create virtual environments during automated vignette building.
- Therefore, Python code must not be executed during vignette building.

Users running this vignette interactively can simply copy these commands into their R session (after running `env_Load()` once).

Installation

```
# install.packages("devtools")
devtools::install_github("dr-yi-zhang/seneR")
```

Load the package:

```
library(seneR)
# env_Load()  # Run this interactively, not during vignette building
```

Bulk RNA-seq Example

Load example data

```
example_expr <- read.csv(
  system.file("demo", "GSE246425_counts.csv", package = "seneR"),
  row.names = 1
)
example_expr[1:4,1:4]
#>          GSM7868701_Young_Male_A GSM7868702_Young_Male_B
#> 5S_rRNA                        1                       6
#> 5_8S_rRNA                      0                       0
#> 7SK                            9                       4
#> A1BG                          54                      41
#>          GSM7868703_Young_Female_A GSM7868704_Young_Female_B
#> 5S_rRNA                          1                         6
#> 5_8S_rRNA                        0                         0
#> 7SK                              4                         7
#> A1BG                            33                        24

example_meta <- read.csv(
  system.file("demo", "GSE246425_meta.csv", package = "seneR"),
  row.names = 1
)
head(example_meta)
#>                           group gender
#> GSM7868701_Young_Male_A   Young   Male
#> GSM7868702_Young_Male_B   Young   Male
#> GSM7868703_Young_Female_A Young Female
#> GSM7868704_Young_Female_B Young Female
#> GSM7868705_Young_Male_A   Young   Male
#> GSM7868706_Young_Male_B   Young   Male
```

Senescence Level (SID Score)

```
SID_res <- SenCID(example_expr)
score_res <- SID_res$score_res
head(score_res)
```

To compare groups:

```
plot_group(
  score_res,
  example_meta,
  group_col = "group",
  score_col = "SID_Score",
  comparisons = list(c("Old", "Young"))
)
```

Senescence-related Phenotype Assessment

```r
gsva_res <- seneGSVA(example_expr)
```

Plot phenotypes:

```r
plot_violin(
  gsva_res,
  example_meta,
  group_col = "group",
  adjust_fdr = TRUE,
  p_threshold = 0.05
)
```

---

scRNA-seq Example with Seurat

Load a test dataset:

```r
library(Seurat)
library(SeuratData)

options(timeout = 600)
SeuratData::InstallData("pbmc3k")
pbmc3k <- LoadData("pbmc3k", type = "pbmc3k.final")
```

Compute cell-level SID scores:

```r
expr_matrix <- GetAssayData(pbmc3k, layer = "counts")
SID_res <- SenCID(expr_matrix, binarize = TRUE)
pbmc3k@meta.data <- cbind(pbmc3k@meta.data, SID_res$score_res)
```

Visualize:

```r
VlnPlot(pbmc3k, features = "SID_Score", group.by = "seurat_annotations")
DimPlot(pbmc3k)
DimPlot(pbmc3k, group.by = "RecSID")
DimPlot(pbmc3k, group.by = "Binarization")
FeaturePlot(pbmc3k, features = "SID_Score")
```

Senescence Trajectory Analysis

```r
library(monocle)
seneCDS <- sene_traject(pbmc3k, color = "SID_Score", cores = 7)
```

Trajectory heatmap:

```r
sene_heatmap(seneCDS, num = 20, num_clusters = 3)
```

Why Many Chunks Use `eval=FALSE`

Because seneR depends on Python libraries, the following Python modules must exist for full execution:

- numpy
- sklearn
- joblib

But:

- **CRAN**
- **GitHub install_github()**
- **Automated checks**

do **not** allow Python installation during vignette building.

Therefore, Python-dependent code **cannot run** inside the vignette, or the build will fail.

Users running seneR locally can simply run all commands normally after calling:

```r
env_Load()
```

This loads the Python environment and enables full functionality.

Feedback

For questions or suggestions, please contact:
2211563@tongji.edu.cn

References

Tao et al. (2024). SenCID: A machine learning model for cellular senescence prediction.