Step 系列：scRNA-seq 基本分析

2024-02-05

LiChuang Huang

@ 立效研究院

Contents

List of Figures

List of Tables

1 Step 系列共性特征

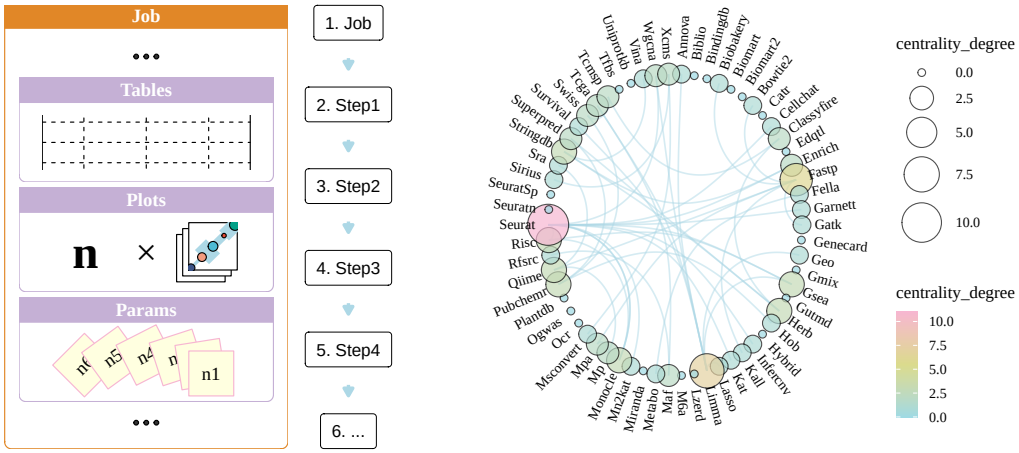

Figure 1: Workflow frame

1.1 理念

生物信息分析工具种类繁多，开发者出于各种原因，导致工具的适用性千差万别。学习和应用这些工具的成本可能是高昂的 (甚至它们可能有一些不为人知的漏洞)，想要把它们串连在一起分析，更是要付出一些分析之外的代价：像开发者一样调试这些程序。

Step 系列的分析方法为每一个制定好的方法、思路或工具 (一般是领域中的权威、经典或翘楚) 设立统一使用的标准，统一的数据存储，统一的应用流程，大幅度降低了学习成本、使用成本、应用成本。通过避免大量“分析之外”的繁琐工作，达到提高分析效率的目的。

Step 系列分析方法的一些基本特性：

- 统一的分析平台。Step 背后涵盖的工具可能涉及：R 包、Python 包、Java 包、Linux 命令行工具等。但最终用于分析的始终是 R 语言。Step 系列的方法通过在 R 中调用各种分析工具，实现不同分析平台之间的工具调用和数据对接 (降低了学习和应用成本)。
- 统一的数据存储单位。Step 系列所有的分析流程 (workflow) 都以一个对象 (Object) 存储。这避免了如果一个流程中涉及纷繁复杂的分析方法，人工存储数据 (中间数据、最终输出的图片、表格) 出错的可能 (提高效率)。必要时，通过统一的方法提取这些数据 (就像在图书馆的某一层的某一排的书架的某一个柜子取出一本书)。
- 统一的方法名称。Step 系列的所有 Workflow 的方法名称都是统一的 (step1、step2、step3…… map、vis 等)，但不会因为一同调用而出错。这是为了减免分析者的负担而设计的 (如果一次完整分析涉及十几种工具，每个工具又有十几种方法名称，那对分析者的记忆量和细心度的考验是惊人的) (提高效率)。
- 规范的分析流程。大多数的分析工具本身具备多种方法以适用灵活分析，但也提高了学习、应用成本。Step 系列的各个 Workflow 的流程都是单向的，对分析方法或思路的组合应用大都是建立在官方指南或教程的基础上，又或者是泛用性 (降低了学习成本)。如果有不同思路，可以通过关键参数调整方法，

或者视情况搭建一个额外的 Workflow。因此，每个 Workflow 是以分析思路为分门别类的，而不是工具本身。

- 提供权威、经典、泛用的分析方法。Workflow 创建前，会广泛查阅时下文献，对工具择优而选、择新而用 (比如，更趋向于选择更新的来自于 Nature Biotechnology 的分析方法) 。
- 提供泛用的组合思路。通过组合各种数据库、分析工具，应对千奇百怪的分析需求。同时，一些适当的组合，会发挥超出单一工具的价值 (Fig. **??** 的右图提供了现如今存在的许多组合思路；同时，每个 Workflow 内部又存在一些思路组合) 。
- 不断开发进化。每一个完成的 Workflow 不是固定不变的，在面临新的分析环境、新的分析需求，更多的功能会被加入到 "step" 方法中，让每一次的进步都直接应用于今后所有的同类分析。另外，一些大大小小的创新 (比如，更严谨或更酷炫的绘图) 也会不断追加到方法中。
- 效率至上。所有分析方法以输入命令形式实现，允许批量处理。
- 附带实用工具。分析流程相对固定，但通过提供一些小工具 (比如用于额外的绘图) ，可以将分析更加灵活。

1.2 局限性

- 适配性。由于多数生信工具都基于 Linux，此外 Step 系列在编写时，也只在 Linux 上调试过；所以，这些工具将只适用于 Linux 系统。

1.3 泛用方法

R input

```
sr <- job_seurat("./data")
sr <- step1(sr)
sr <- step2(sr)
sr <- step3(sr)
sr <- step4(sr)
sr <- step5(sr)
sr <- step6(sr)

mn <- asjob_monocle(sr)
mn <- step1(mn)
mn <- step2(mn)
mn <- step3(mn)
```

1.3.1 提取方法

1.3.2 存储方法

R input

```
autosv(sr@plots$step1)
```

1.3.3 空间管理

某些 R 包生成的数据或加载的数据可能极其占用运行内存 (RAM)。可以用 space() 函数查看当前内存占用:

R input

```
space()
```

1.3.4 查看 step 方法的默认参数

step 方法的参数力求精简，一般只保留关键的参数用以控制分析。但这些参数可能会在将来被拓展 (添加额外的参数) 以适应新的分析需求。

可以通过类似以下方式查看默认参数:

R input

```
## 示例: seurat 工作流的 step1 的默认参数
not(.job_seurat())
step1
```

```
## job_seurat:
##     x
##
## -- Methods parameters ----------------------------------------------------------------------
```

R input

```
## 示例: monocle 工作流 step2 的默认参数
not(.job_monocle())
step2
```

```
## job_monocle:
##     x, roots
##
## -- Methods parameters ----------------------------------------------------------------------
```

1.3.5 额外的信息

工作流创建时参考的信息源参考文献信息

R input

```
# 创建一个空的 Seurat 工作流对象 'sr_1'
sr_1 <- .job_seurat()
# 查看方法说明
sr_1@method
```

```
## [1] "The R package `Seurat` used for scRNA-seq processing; `SCSA` (python) used for cell type annota
```

R input

```
# 查看官方网站或信息源网站
sr_1@info
```

```
## [1] "Tutorial: https://satijalab.org/seurat/articles/pbmc3k_tutorial.html"
```

1.4 关于安装配置

所有 Step 系列的方法的安装配置会尽可能详细的罗列，但由于笔者 (开发者) 仅在自身的 Linux (Ubuntu 发行版) 系统做过调试，并不确定在其它的机器上会遇到哪些安装的特殊问题。如有疑问，或安装上的困难，请联系：Huang Lichuang (huanglichuang@wie-biotech.com) 。

1.5 关于本文档

2 Step 系列：scRNA-seq 基本分析

2.1 摘要

2.1.1 目的

2.1.2 解决的问题

2.2 适配平台

2.3 方法

Mainly used method:

- GEO https://www.ncbi.nlm.nih.gov/geo/ used for expression dataset aquisition.
- The R package `Seurat` used for scRNA-seq processing; `SCSA` (python) used for cell type annotation[1–3].
- Other R packages (eg., `dplyr` and `ggplot2`) used for statistic analysis or data visualization.

2.4 安装 (首次使用)

2.4.1 安装依赖

2.4.1.1 安装 Seurat v5

R input

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
if (!requireNamespace("remotes", quietly = TRUE))
    install.packages("remotes")
install.packages(c("sva"))
BiocManager::install(c('SparseArray', 'fastDummies', 'RcppHNSW', 'RSpectra'))
remotes::install_github("satijalab/seurat", "seurat5")
```

2.4.1.2 安装 seurat-wrappers

R input

```
remotes::install_github('satijalab/seurat-wrappers')
```

2.4.1.3 安装 monocle3

R input

```
BiocManager::install(c('BiocGenerics', 'DelayedArray', 'DelayedMatrixStats',
    'limma', 'lme4', 'S4Vectors', 'SingleCellExperiment',
    'SummarizedExperiment', 'batchelor', 'HDF5Array',
    'terra', 'ggrastr', 'rsample'))
remotes::install_github('cole-trapnell-lab/monocle3')
```

2.4.1.4 安装 CellChat

R input

```
BiocManager::install(c('NMF', 'circlize', 'ComplexHeatmap', 'BiocNeighbors'))
remotes::install_github("sqjin/CellChat")
```

2.4.1.5 安装 SCSA

bash input

```
git clone https://github.com/bioinfo-ibms-pumc/SCSA.git ~/SCSA
pip3 install numpy scipy openpyxl
pip3 install pandas==1.5.3
```

2.4.1.6 其它程序

以下可能是其它需要安装的程序：

R input

```
install.packages("reticulate")
reticulate::install_miniconda()
reticulate::py_install(packages = 'umap-learn')
```

你可能还想要安装：

R input

```
BiocManager::install(c("SingleR"))
BiocManager::install(c("celldex"))
```

2.4.2 安装主体

2.5 使用说明

2.6 示例分析

2.6.1 数据准备（从 GEO 的单细胞数据库开始分析）

10X Genomics (其他格式也行，但我遇到过的几乎都是 10X 的，所以其他格式的没机会调试) 的文件。

这里为了方便起见，我从 GEO 下载一批数据实时以代码示例。

2.6.1.1 快速获取示例数据

运行以下代码获取数据：

R input

```
geo <- job_geo("GSE171306")
geo <- step1(geo)
geo <- step2(geo)
untar("./GSE171306/GSE171306_RAW.tar", exdir = "./GSE171306")
prepare_10x("./GSE171306/", "ccRCC1", single = F)
# 创建 job_seurat 对象:
sr <- job_seurat("./GSE171306/GSM5222644_ccRCC1_barcodes")
```

2.6.1.2 一些补充说明（上述快速获取数据方式的实用价值）

job_geo 是另外一个可以用于高效获取、整理 GEO 数据集的 step 系列工作流 (以后介绍)，简而言之，它的作用在于帮我们极速整理好数据还有元数据 (整合大量数据集的时候很有用！)。例如，我们可以通过以下方式查看这批 GEO 数据的样品信息和数据前处理:

R input

```
geo$guess
```

```
## # A tibble: 2 x 3
##   rownames   title                                                tissue.ch1
##   <chr>      <chr>                                                <chr>
## 1 GSM5222644 ccRCC1, Right clear cell renal cell carcinoma Homogenate Right clear cell renal cell ca
## 2 GSM5222645 ccRCC2, Left clear cell renal cell carcinoma Homogenate  Left clear cell renal cell car
```

R input

```
geo$prods
```

```
##     Preliminary sequencing results (bcl files) were converted
##     to FASTQ files with Cell Ranger V3.1
##
##     R1 end, at the beginning of 16bp is CellBarcode sequence,
##     then 10bp is UMI sequence, R2 end, we can truncate 151bp to
##     98bp
##
##     The CellRanger (10X Genomics) secondary analysis pipeline
##     was used to generate a digital gene expression matrix
##
##     Normalization and additional analysis by Seurat R package
##
##     Genome_build: GRCh38 for human data
##
##     Supplementary_files_format_and_content: CellRanger output
```

```
##     files (the barcode, gene, expression matrix file of each
##     sample)
```

确认解压得到了些什么文件：

R input

```
list.files("./GSE171306/")
```

```
## [1] "GSE171306_RAW.tar"                  "GSM5222644_ccRCC1_barcodes"        "GSM5222645_ccRCC2_barco
## [4] "GSM5222645_ccRCC2_features.tsv.gz" "GSM5222645_ccRCC2_matrix.mtx.gz"
```

GEO 中的单细胞数据 (Supplementary file) 的存储形式不统一 (很随意)，一一整理起来用于输入很繁琐。`prepare_10x` 是我写的一个将同一样本的三个文件 (barcodes.tsv, matrix.mtx, features.tsv) 存放于一个目录中的高效办法。

R input

```
# 如果有三个文件：
prepare_10x("./GSE171306/", "ccRCC1", single = F)
# 如果只有一个 Matrix 文件：
prepare_10x("./GSE171306/", "ccRCC1", single = T)
```

2.6.2 分析流程

2.6.2.1 Job-seurat

在 2.6.1.1 中，已经运行过：

R input

```
sr <- job_seurat("./GSE171306/GSM5222644_ccRCC1_barcodes")
```

如你不想用 Step 系列的风格来分析，更想用 Seurat 的原生代码，那么你可以：

R input

```
seurat <- object(sr)
# 这样，`seurat` 就是你需要的数据对象。
# 你可以参考：<https://satijalab.org/seurat/articles/pbmc3k_tutorial.html>
# 不借助 Step 的默认设定，而开展自主分析。
```

2.6.2.2 Step1 数据质控

只需要运行：

R input

```
# 这一步不需要输入参数
sr <- step1(sr)
```

R input

```
# 获取运行的结果
sr@plots$step1$p.qc
```

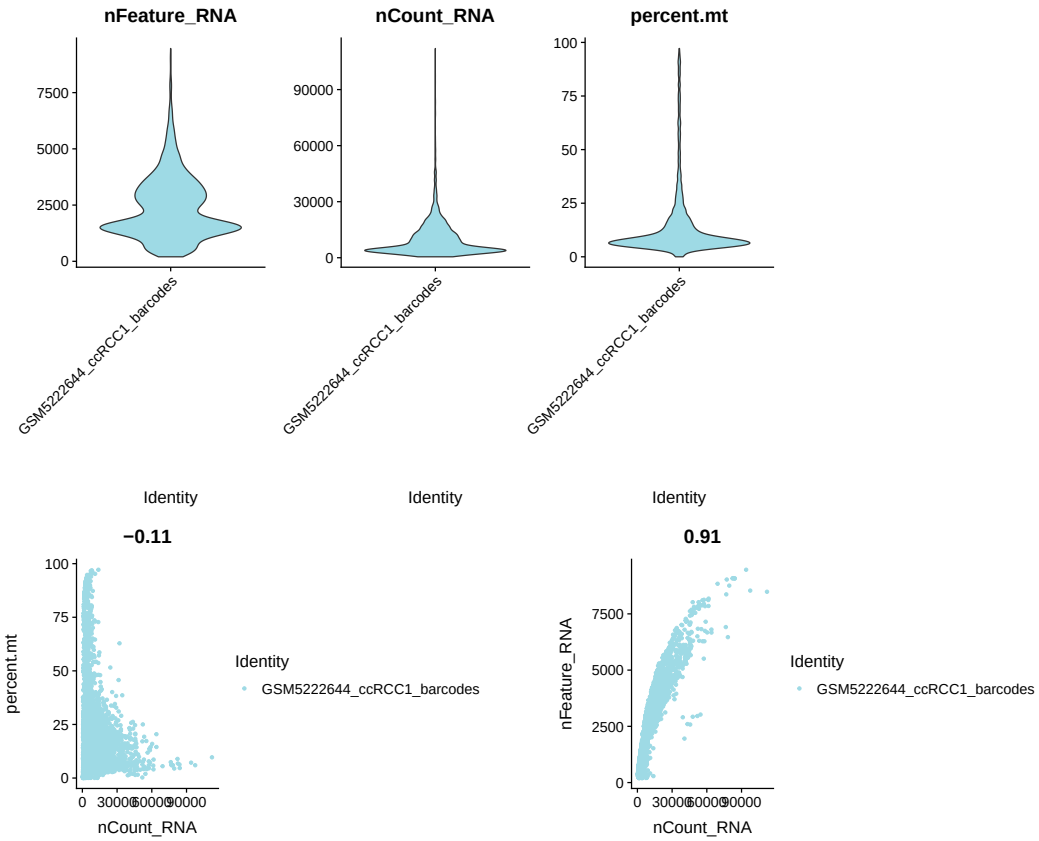

Figure 2: Quality Control

Fig. 2, 你可能会觉得 x 轴坐标太长了, 如果在此前 (创建对象的时候) 输入类似 `job_seurat("./GSE171306/", project = "Demo data")`, 就可以避免。

2.6.2.3 Step2 根据上一步 QC 作图过滤数据并标准化

这里的参数相当重要。

R input

```
sr <- step2(sr, 0, 7500, 35)
```

可以确认默认的参数:

R input

```
not(sr)
step2
```

```
## job_seurat:
##     x, min.features, max.features, max.percent.mt = 5, nfeatures = 2000, use = "nFeature_RNA"
##
## -- Methods parameters ----------------------------------------------------------------------
```

输入的三个参数对应以下：

- min.features
- max.features
- max.percent.mt

官网对 `percent.mt` 做了一些解释：

- The number of unique genes detected in each cell.
  - Low-quality cells or empty droplets will often have very few genes
  - Cell doublets or multiplets may exhibit an aberrantly high gene count
- Similarly, the total number of molecules detected within a cell (correlates strongly with unique genes)
  - The percentage of reads that map to the mitochondrial genome
  - Low-quality / dying cells often exhibit extensive mitochondrial contamination
- We calculate mitochondrial QC metrics with the PercentageFeatureSet() function, which calculates the percentage of counts originating from a set of features
  - We use the set of all genes starting with MT- as a set of mitochondrial genes

总而言之，这一步需要根据 Fig. 2 中的小提琴图选择合适的参数。

最后，这一步可以得到 4 个 Figure：

R input

```
sr@plots$step2$p.pca_pcComponents
sr@plots$step2$p.pca_1v2
sr@plots$step2$p.pca_heatmap
sr@plots$step2$p.pca_rank
```

这里不展开。其中，下一步的主要依据的是 `sr@plots$step2$p.pca_rank`，即 Fig. 3

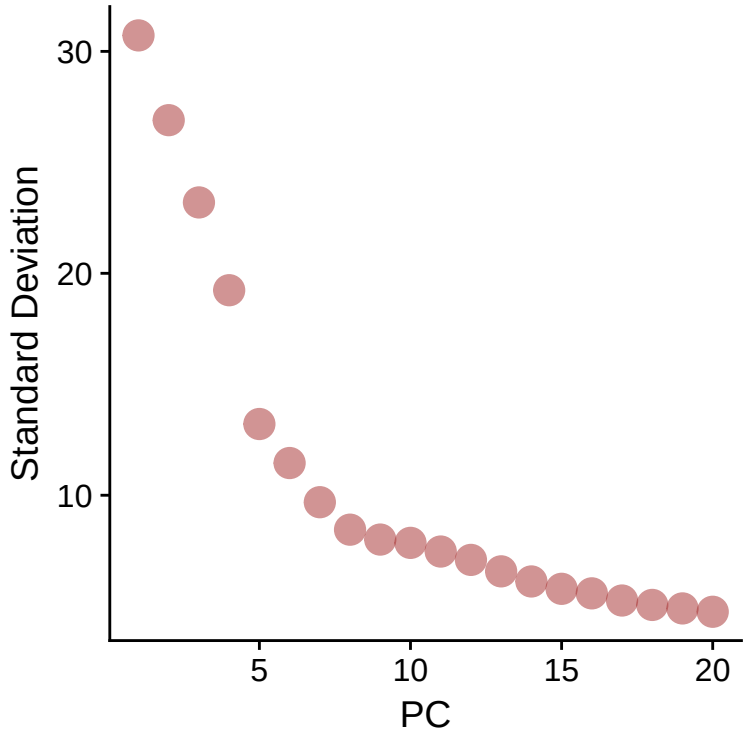

Figure 3: Ranking of principle components

2.6.2.4 Step3 完成降围、聚类和 Marker 筛选

R input

```
sr <- step3(sr, 1:15, 1.2)
```

可以确认默认的参数：

R input

```
step3
```

```
## job_seurat:
##     x, dims, resolution, force = F
##
## -- Methods parameters ------------------------------------------------------------------------
```

参数 `dim` 需要根据 Fig. ?? 判定。`resolution` 需要根据细胞数判定。这两个参数会传递到：

- Seurat::FindNeighbors
- Seurat::FindClusters

官方有一段解释：

The FindClusters() function implements this procedure, and contains a resolution parameter that sets the 'granularity' of the downstream clustering, with increased values leading to a greater number of clusters. We find that setting this parameter between 0.4-1.2 typically returns good results for single-cell datasets of around 3K cells. Optimal resolution often increases for larger datasets.

运行后将可以得到：

R input

```
sr@plots$step3$p.umap
```

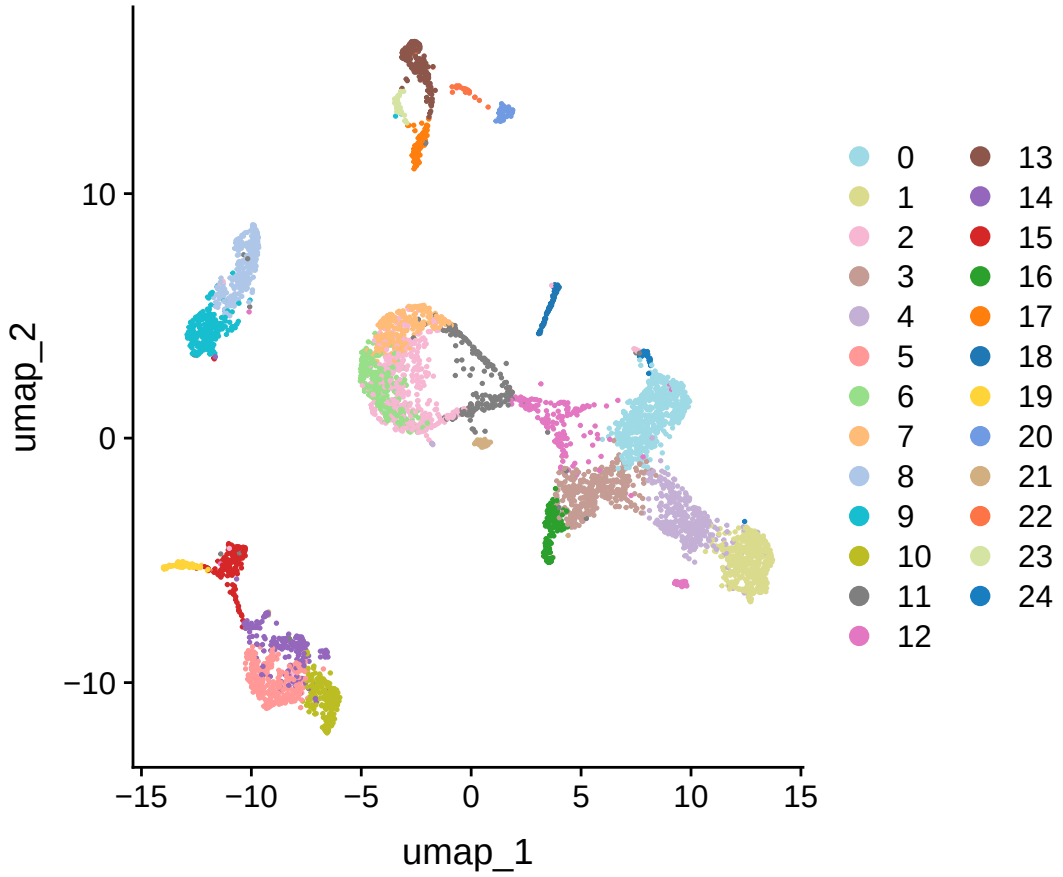

Figure 4: UMAP Clustering

2.6.2.5 Step4 使用 SingleR 注释细胞类

(我现在都不用 SingleR 注释了，因为我发现它时常把其他组织的细胞注释到目标组织上，比如，注释肾组织注释得到肝脏细胞; 默认使用的参考数据是：celldex::HumanPrimaryCellAtlasData，同样的，可以输入 `step4` 查看默认参数。)

如果要运行这一步 (挺耗时的，并需要确保你安装了 SingleR 和 celldex，详情见 2.4.1.6)：

R input

```
sr <- step4(sr, "SingleR")
```

跳过这一步：

R input

```
sr <- step4(sr, "")
```

如果你运行了这一步，你可以得到 (我没有运行，所以不展示了)：

R input

```
sr@plots$step4$p.score_SingleR
sr@plots$step4$p.map_SingleR
sr@tables$step4$anno_SingleR
```

2.6.2.6 Step5 计算各细胞类的 Marker 基因 (差异分析)

唯一需要输入的参数是线程数 (不知道是否是 Seurat 程序有问题，我好像基本都是单线程的，即使设置了这个参数)：

R input

```
## 设置成 NULL, 单线程
sr <- step5(sr, 5)
```

这样，你就能得到：

R input

```
sr@tables$step5$all_markers
sr@tables$step5$all_markers_no_filter
```

Table 1: All Markers

rownames	p_val	avg_lo...	pct.1	pct.2	p_val_adj	cluster	gene
GZMK	0	4.2402...	0.982	0.112	0	0	GZMK
CD27	0	3.2768...	0.747	0.094	0	0	CD27
CD8B	0	3.1693...	0.747	0.099	0	0	CD8B
DUSP4	0	3.0453...	0.774	0.144	0	0	DUSP4
CD8A	0	2.7603...	0.816	0.158	0	0	CD8A
GZMA	1.1171...	2.2365...	0.989	0.344	2.1412...	0	GZMA
CCL5	6.7488...	2.2809...	0.996	0.393	1.2935...	0	CCL5
CD3D	1.8166...	1.9909...	0.96	0.3	3.4820...	0	CD3D
TRAC	4.5242...	2.0246...	0.918	0.258	8.6716...	0	TRAC

rownames	p_val	avg_lo...	pct.1	pct.2	p_val_adj	cluster	gene
APOBEC3G	7.8727...	2.3392...	0.904	0.345	1.5089...	0	APOBEC3G
TRBC2	1.3387...	2.1611...	0.886	0.305	2.5659...	0	TRBC2
ITM2A	6.6648...	2.3921...	0.844	0.307	1.2774...	0	ITM2A
RPL28	4.5358...	1.0709...	1	1	8.6938...	0	RPL28
CST7	9.2732...	1.6915...	0.965	0.343	1.7773...	0	CST7
CD3E	3.8141...	1.8107...	0.914	0.294	7.3104...	0	CD3E
...	...	...	...	...	...	...	...

2.6.2.7 Step6 使用 SCSA 注释细胞类

`SCSA` 是 Python 编写的命令行工具。`step6` 已经调用以及数据的转换集成了，只需要运行如下，就能简便地得到结果：

R input

```
# 这个样本的组织来源是 Kidney
sr <- step6(sr, "Kidney")
```

(如果你没有按照默认的方式安装 SCSA，那么你可能需要额外输入 `cmd` 和 `db` 参数)

2.6.3 完整示例代码

2.7 技巧

Reference

1. Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *Cell* **184**, (2021).

2. Stuart, T. *et al.* Comprehensive integration of single-cell data. *Cell* **177**, (2019).

3. Cao, Y., Wang, X. & Peng, G. SCSA: A cell type annotation tool for single-cell rna-seq data. *Frontiers in genetics* **11**, (2020).