Step 系列：scRNA-seq 癌细胞鉴定

2024-02-22

LiChuang Huang

@ 立效研究院

Contents

List of Figures

List of Tables

1 摘要

1.1 目的

解决癌组织单细胞数据集中，癌细胞鉴定的难题，并延续一般性的单细胞数据分析流程。

1.2 解决的问题（技术性的）

不同的 R 包或其他工具之间的数据转换和衔接。

2 前情资料

这份文档是以下的补充资料（以下的配置和使用是前提条件）：

- 《Step 系列：scRNA-seq 基本分析》

如果你还不知道 Step 系列的基本特性以及一些泛用的提取和存储方法，请先阅读：

- 《Step 系列：Prologue and Get-start》

3 适配性

同《Step 系列：scRNA-seq 基本分析》。

4 方法

以下是我在这个工作流中涉及的方法和程序：

Mainly used method:

- R package `copyKAT` used for aneuploid cell or cancer cell prediction[1].
- R package `Monocle3` used for cell pseudotime analysis[2,3].
- The R package `Seurat` used for scRNA-seq processing; `SCSA` (python) used for cell type annotation[4–6].
- Other R packages (eg., `dplyr` and `ggplot2`) used for statistic analysis or data visualization.

5 安装（首次使用）

5.1 安装依赖

《Step 系列：scRNA-seq 基本分析》已经例举了大部分程序的安装方法，以下，仅展示 R 包 `copykat` 的安装。

R input

```
devtools::install_github("navinlabcode/copykat")
```

6 示例分析

在《Step 系列：scRNA-seq 基本分析》中，我以 GSE171306 做了 scRNA-seq 一般性的示例。但其实 GSE171306 是一批癌组织数据集，理应鉴定癌细胞，而 SCSA 和多数其他自动注释工具，都无法鉴定出癌细胞。这样，就有必要专门鉴定癌细胞了。

以下针对癌细胞鉴定的问题展开示例分析，并依然使用 GSE171306 这批数据。

注：只要是《Step 系列：scRNA-seq 基本分析》中提及的，以下就不再赘述；只细述新的内容。

6.1 数据准备

6.1.1 快速获取示例数据

运行以下代码获取数据 (和《Step 系列：scRNA-seq 基本分析》中的是一样的)：

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

6.2 分析流程

6.2.1 单细胞数据的质控、聚类、Marker 鉴定、细胞注释等

以下直接给出代码 (和《Step 系列：scRNA-seq 基本分析》相同)：

R input

```
sr <- step1(sr)
sr <- step2(sr, 0, 7500, 35)
sr <- step3(sr, 1:15, 1.2)
sr <- step4(sr, "")
sr <- step5(sr, 5)
sr <- step6(sr, "Kidney")
```

6.2.2 癌细胞鉴定

6.2.2.1 As-job-kat 将前处理完毕的 job_seurat 数据对象转化

R input

```
kat <- asjob_kat(sr)
```

注意，这一步默认将 sr@object (也就是 seurat 数据对象) 中的 assays 数据槽中的第一个数据集用于鉴定癌细胞。转化完成后，你会在 kat@object 看到这个矩阵数据集。一般情况下，使用的都是第一个数据集。

你可以手动指定数据集，例如：

R input

```
# 不要运行
kat <- asjob_kat(sr, use = names(x@object@assays)[[1]])
```

6.2.2.2 Step1 根据变异拷贝数鉴定癌细胞

copykat 有一个优点 (相对于 inferCNV)，不需要手动指定参考细胞，程序会自主在数据集中选择参考细胞。所以，将所有的细胞表达数据输入就可以了。

(这一步会比较耗时，1-5 小时)

R input

```
# 后一个参数指定线程数
kat <- step1(kat, 8)
# 你还可以通过添加 `path` 参数，指定其他文件夹用以存放中间数据
```

这里，我有必要做一些说明。copykat 内置一些参考数据集，用以参考和计算。对于人类，它内置的是 'hg20' 参考集。目前遇到更多的可能是 'hg38'。copykat 内部并不会对基因符号 (symbol) 后缀的版本信息进行替换和匹配，因此，如果输入的基因即使是同一种，然而因为版本不同，也会无法匹配上 (例如，你输入的是 'AHR.5'，内置的参考集包含的是 'AHR.1')。

因此，我在 job_kat 的 step1 中补充了这一部分工作，能够让这一步能够顺利进行下去。但我无法预料是否还会有其他特殊情况，所以以上事项需知悉。

如果你同样对以上事项保持警惕，可以用以下确认我做了哪些补充工作：

R input

```
selectMethod(step1, "job_kat")
```

我还需要备注的是，由于我只做到过人类的癌细胞鉴定，所以小鼠的数据集目前还不支持。

6.2.2.3 Step2 可视化鉴定结果

copykat 自带可视化的函数；然而，由于其中的图例绘制的太糟糕，因此，这里我去除了 copykat 绘制的热图的图例，重新添加了一个新的图例。

这一步也会比较耗时 (热图很大)。

R input

```
kat <- step2(kat)
```

该热图可以直接提取查看；但最好不要这么做，因为加载这个热图太耗时：

- `kat@plots$step2$p.copykat`

推荐的做法是 (见 6.2.2.4)，运行下一部分 (`clear`) 之后，再将输出的 png 图片查看。

这里，我们也可以直接获取鉴定结果表格：

R input

```
kat@tables$step2$res_copykat
```

Table 1: CopyKAT results

cell.names	copykat.pred	copykat_cell
AAACCCAAGCTGTGCC-1	diploid	Normal cell
AAACCCACAGCCGGTT-1	diploid	Normal cell
AAACCCATCATGAGGG-1	diploid	Normal cell
AAACGAAAGTCACAGG-1	diploid	Normal cell
AAACGAACACACAGAG-1	diploid	Normal cell
AAACGAACACCTCTAC-1	diploid	Normal cell
AAACGAATCAACACGT-1	diploid	Normal cell
AAACGAATCACTACTT-1	diploid	Normal cell
AAACGAATCGGCATAT-1	aneuploid	Cancer cell
AAACGCTAGACAGTCG-1	diploid	Normal cell
AAACGCTAGTCCCAAT-1	aneuploid	Cancer cell
AAACGCTCATCAGTCA-1	diploid	Normal cell
AAACGCTCATGACTCA-1	diploid	Normal cell
AAACGCTGTAGGACCA-1	diploid	Normal cell
AAACGCTGTGAGCAGT-1	diploid	Normal cell
...	...	...

6.2.2.4 （额外的）保存 `job_kat` 并输出结果

R input

```
kat <- clear(kat)
```

这样，就能取得想要的结果了：

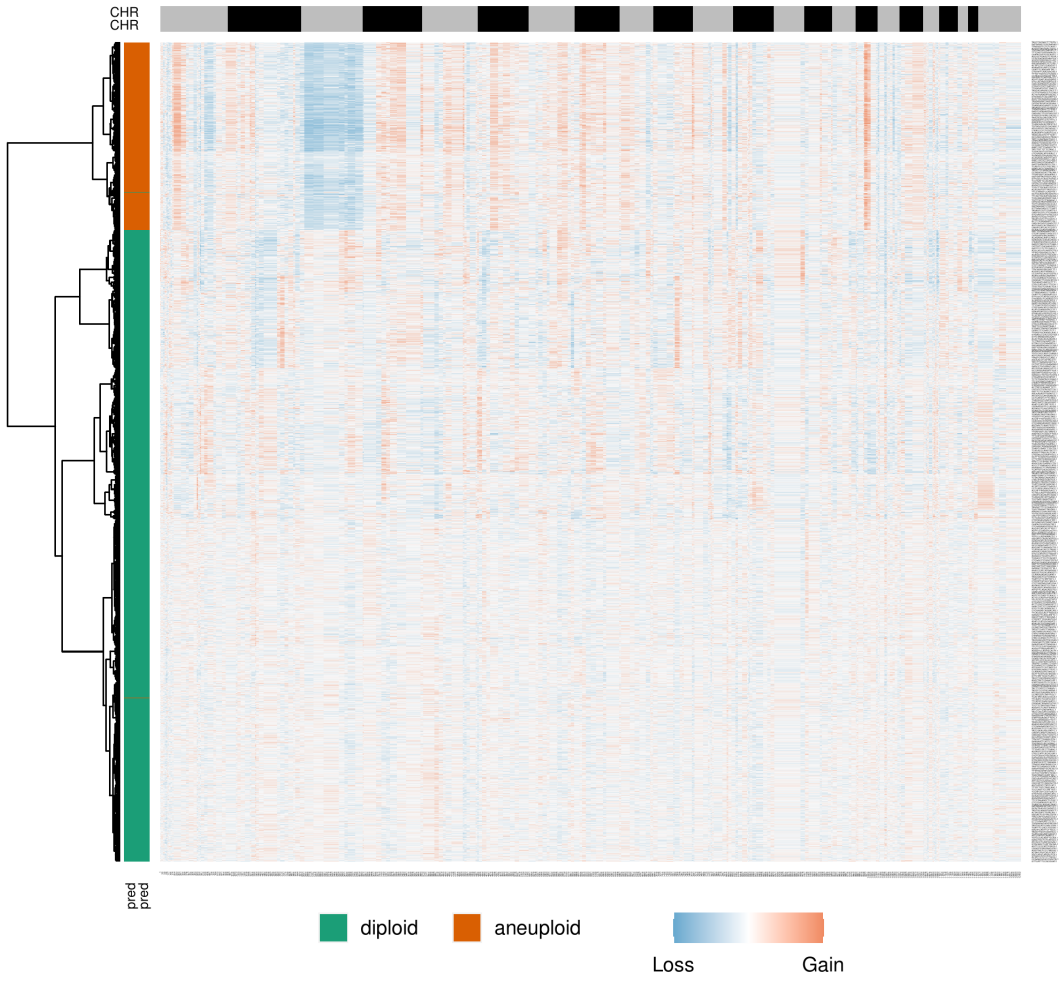

Figure 1: Copykat prediction

Fig. 1，图中的 'aneuploidy' 即为癌细胞。

6.2.2.5 Map 将结果映射回 Seurat

R input

```
sr <- map(sr, kat)
p.sr_vis <- vis(sr, "scsa_copykat")
p.sr_vis
```

这样我们就能在 Fig. 2 中看到，被注释为 'Cancer cell' 的细胞群体。

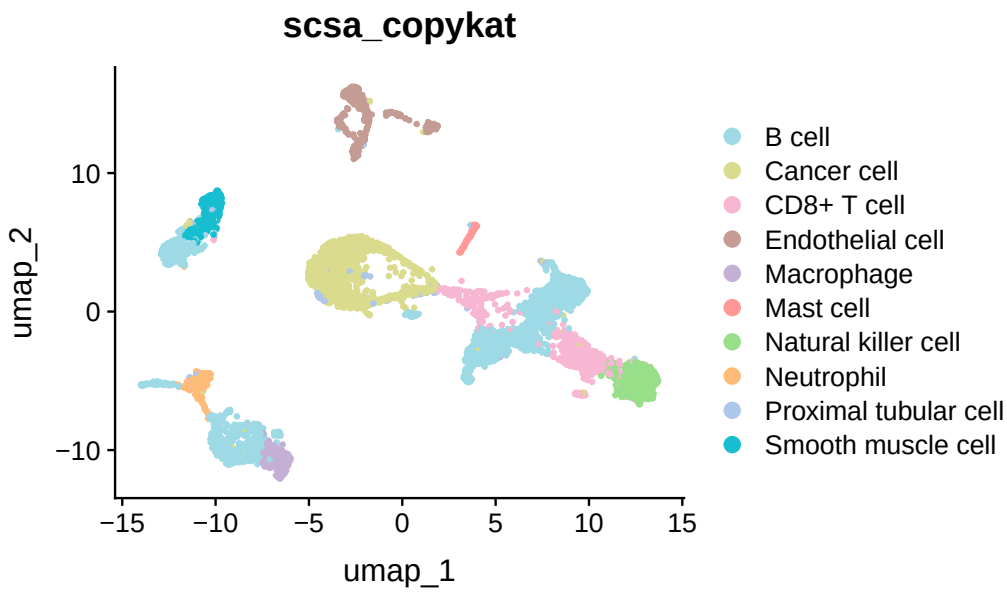

Figure 2: The scsa copykat

我们不妨对比一下 SCSA 的注释结果 (Fig. 3)，看看 'Cancer cell' 可能的来源细胞。

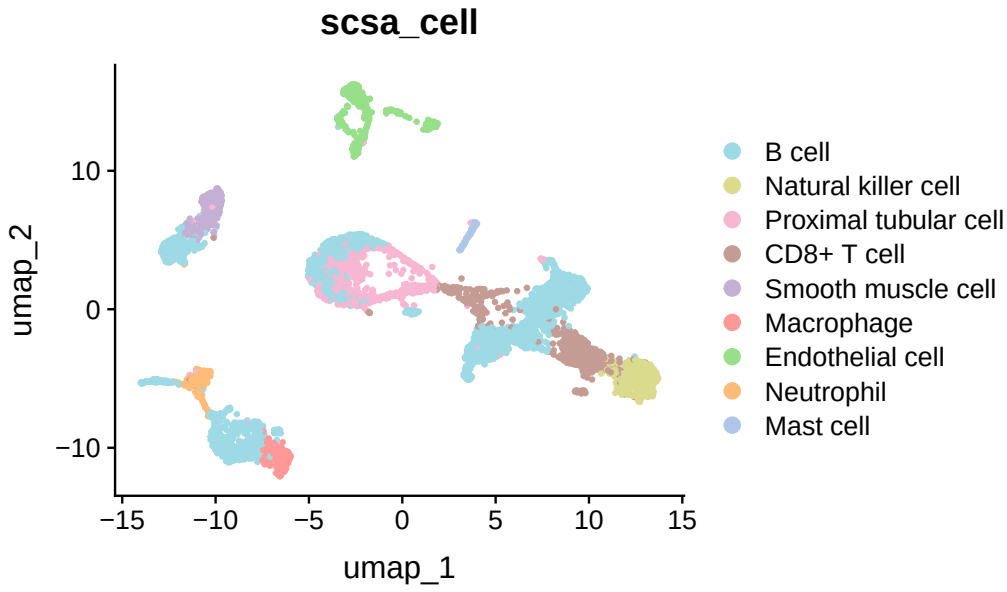

Figure 3: SCSA Cell type annotation

现在可以推断，'Cancer cell' 主要来源于 'Proximal tubular cell'。

6.2.3 拟时分析

其实到 6.2.2.5 为止，本文档的主要内容，即鉴定癌细胞，已经结束了。

然而如果我们进一步思考，如果将 `copykat` 依据变异拷贝数鉴定癌细胞的原理推进，结合拟时分析，也许

就能分析出，癌细胞或正常细胞是如何沿着‘拟时轨迹’，转变成癌细胞了。

这会是一种可以泛用于肿瘤组织单细胞数据的分析方法。

6.2.3.1 do-monocle 对癌细胞进行拟时分析

按照上述思路，这里需要将癌细胞单独取出，用以拟时分析。

我提供了一个便捷的方法，以快速达成这一目的：

R input

```
mn <- do_monocle(sr, kat)
```

在整个 Step 系列方法中，`do_*` 形式的目前还很少；这是一种根据传入的前两个参数的类来决定调用的函数的系列方法。而 `asjob_*` 形式的，只根据第一种参数来决定。

其实，`do_monocle` 内部重新对传入的 `sr` 数据运行了 `step3`，即对分离的癌细胞重新聚类，区分出更多的群体 (可能会是亚型)

6.2.3.2 Step1 构建拟时轨迹

R input

```
mn <- step1(mn)
```

Fig. 4 是用来确认选择拟时起点的。

R input

```
# 以 wrap 调整了长宽比例
wrap(mn@plots$step1$p.prin, 6, 4)
```

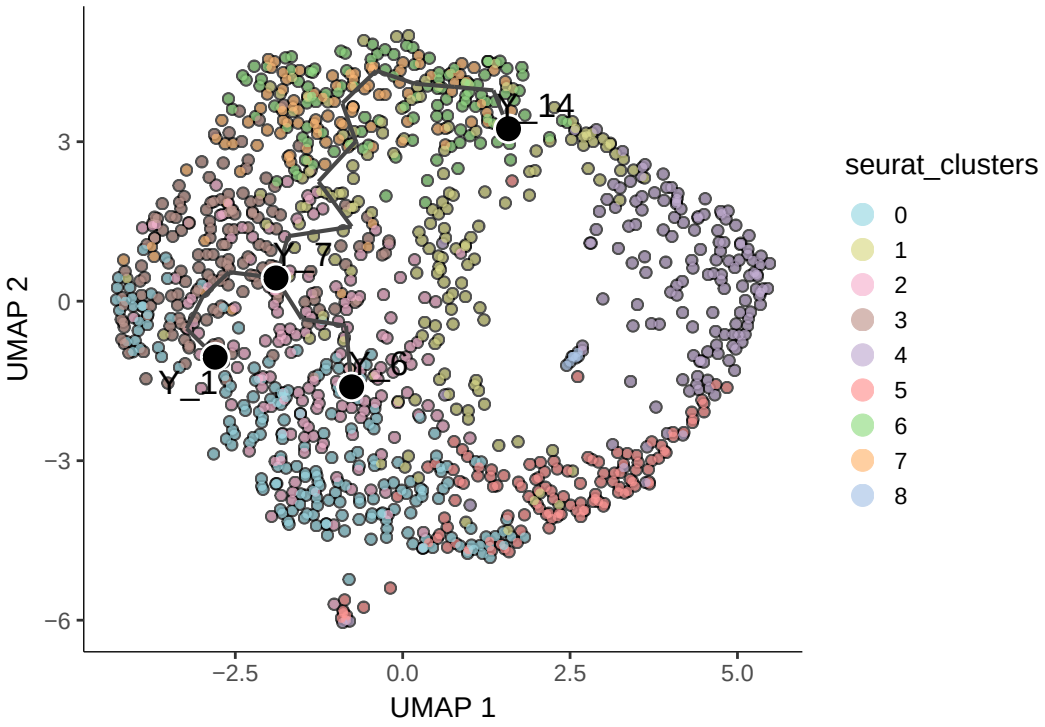

Figure 4: Cancer cell prin

6.2.3.3 Step2 选择拟时起点

选择合适的拟时起点依然会是一个难题。这里提供了一种可借鉴的，用以选择癌细胞拟时起点的思路 (仅供参考) 。见 Fig. 1, 对于 'aneuploid'，即癌细胞，'gain'、'loss' 水平更接近 'diploid' 的细胞，或许更适合作为拟时起点。具体而言，根据 Fig. 1 的侧边聚类树，我们或许可以试着将肿瘤细胞切分为多个小群体，然后根据它们相较于正常细胞的近似程度，选择拟时起点。那么切分之后，哪一群体更加接近正常细胞呢?

见 Fig. 1，如果切分为两个群体，下方高度 (Height) 更低的群体更近似正常细胞。这样，我们就能大致决定：Height 更低的群体作为拟时起点。

我们可以把 `copyKAT` 的聚类结果隐射到 UMAP 聚类上：`mn` 来自于 `do_monocle(sr, kat)`，相较于《Step 系列：scRNA-seq 基本分析》中所述的，有一点特殊之处，即，额外生成了图片，也就是我们需要的映射图，只不过，它一共做了 1-30 种切分 (注意，该切分是针对 Fig. 1 中的所有细胞进行的切分，而不是单单癌细胞) 。

R input

```
mn@plots$step1$p.cancer_position
```

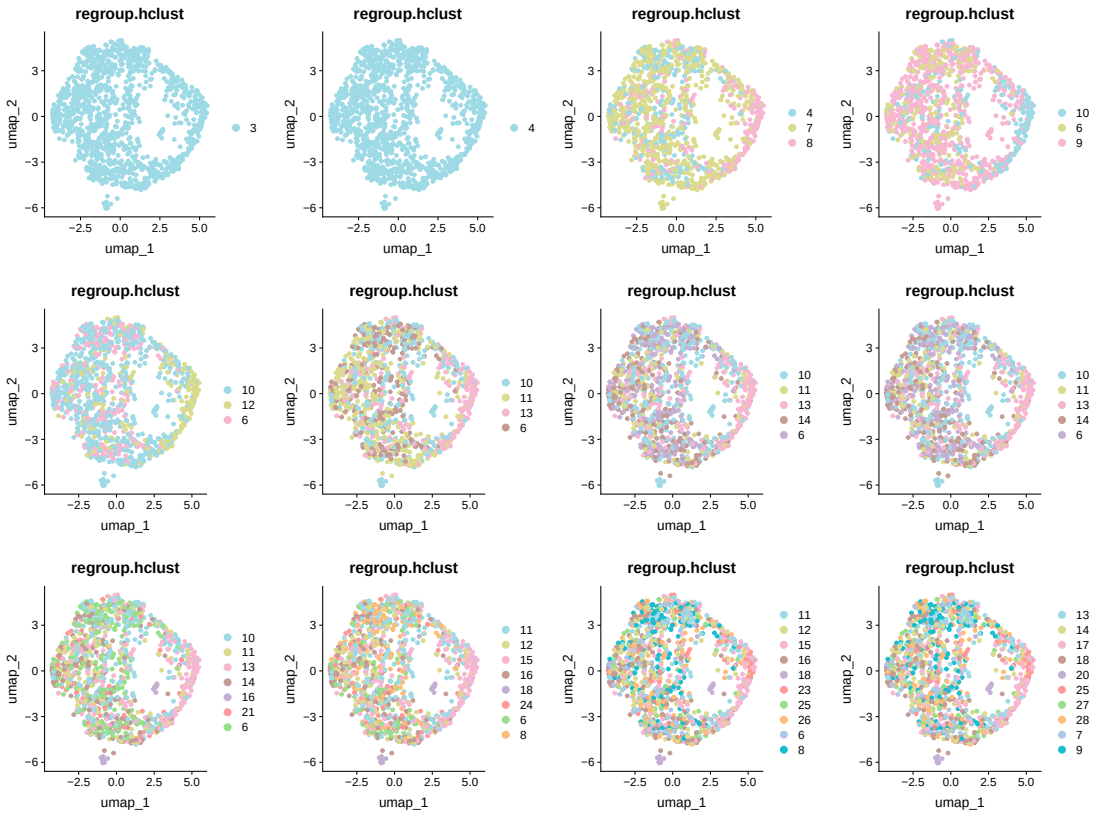

Figure 5: Cut tree

见 Fig. 5，图中的数值越小，代表 Height 越低。如果我们观察最后一副子图，会发现 '9' 代表的群体，主要集中在 UMAP 的上半部分。因此，这里我们试着将更上半部分的细胞作为拟时起点。那么，也就是 Fig. 4 中的 'Y_14'。

R input

```
mn <- step2(mn, "Y_14")
```

这样，我们就能得到：

R input

```
mn@plots$step2$p.pseu
```

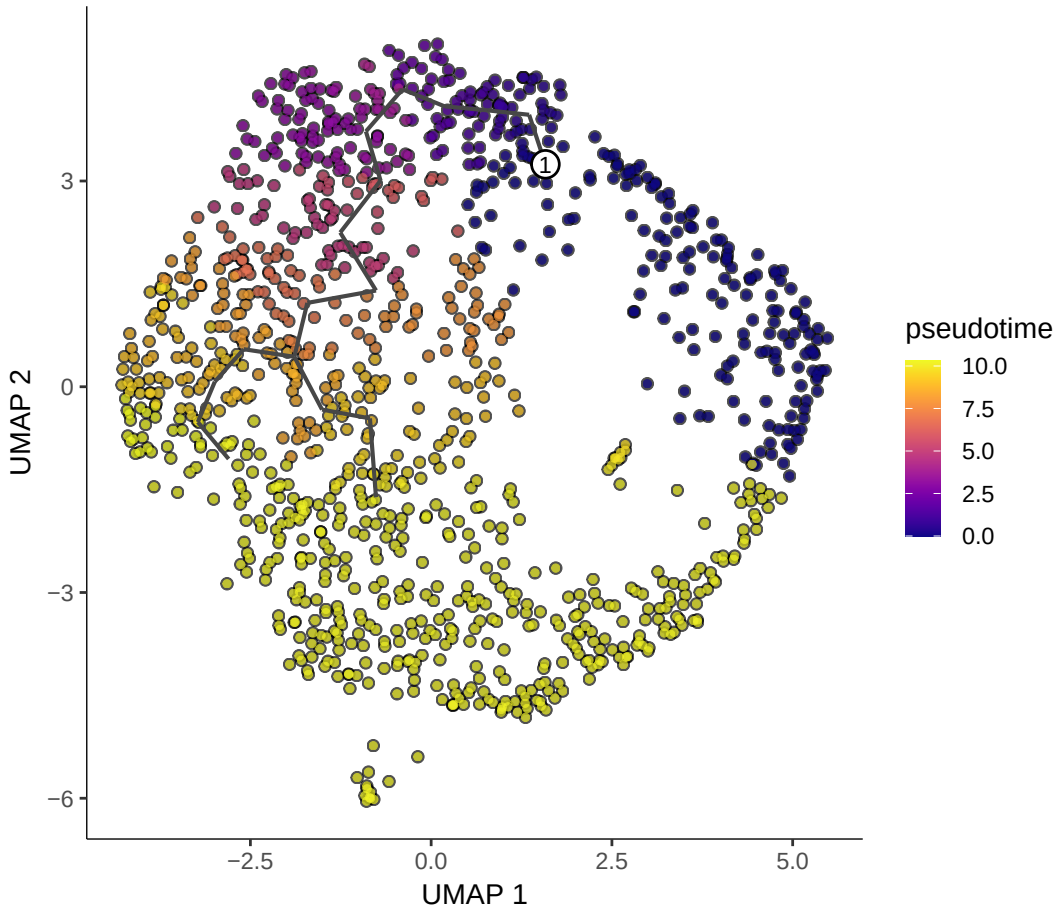

Figure 6: Pseudotime

6.2.3.4 Step3 拟时分析基础上的差异分析和基因表达模块

R input

```
mn <- step3(mn)
```

这样可以得到：

R input

```
# 以下未展示
mn@plots$step3$gene_module_heatdata$graph_test.sig
mn@tables$step3$graph_test
```

注意到了吗，从这里开始，分析已经回到了《Step 系列：scRNA-seq 基本分析》中的拟时分析后的思路上了；如果你想要根据拟时细胞，尝试划分癌细胞的亚型，那么请参考其中的：“(进阶) 根据拟时分析结果重新划分细胞群体”

之后，还可以根据使用者的需求，开展细胞通讯分析或其他分析。

6.3 完整示例代码

R input

```
# 获取示例数据
geo <- job_geo("GSE171306")
geo <- step1(geo)
geo <- step2(geo)
untar("./GSE171306/GSE171306_RAW.tar", exdir = "./GSE171306")
prepare_10x("./GSE171306/", "ccRCC1", single = F)

sr <- job_seurat("./GSE171306/GSM5222644_ccRCC1_barcodes")
sr <- step1(sr)
sr <- step2(sr, 0, 7500, 35)
sr <- step3(sr, 1:15, 1.2)
sr <- step4(sr, "")
sr <- step5(sr, 5)
sr <- step6(sr, "Kidney")

kat <- asjob_kat(sr)
kat <- step1(kat, 8)
kat <- step2(kat)
kat <- clear(kat)

# 将癌细胞鉴定结果映射回 `job_seurat`
sr <- map(sr, kat)

mn <- do_monocle(sr, kat)
mn <- step1(mn)
# 拟时起点需要根据实际情况选择
mn <- step2(mn, "Y_14")
mn <- step3(mn)
```

6.4 Session Info

R input

```
sessionInfo()
```

```
## R version 4.3.2 (2023-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Pop!_OS 22.04 LTS
##
```

```
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.10.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8        LC_COLLATE=en_U
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8    LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATI
##
## time zone: Asia/Shanghai
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  grid      methods   base
##
## other attached packages:
## [1] Seurat_4.9.9.9067       SeuratObject_4.9.9.9091 utils.tool_0.0.0.9000   MCnebula2_0.0.9000      g
## [6] sp_2.0-0                nvimcom_0.9-146
##
## loaded via a namespace (and not attached):
##   [1] IRanges_2.34.1              R.methodsS3_1.8.2           BPCells_0.1.0               urlchecker_
##   [5] nnet_7.3-19                 goftest_1.2-3               vctrs_0.6.3                 spatstat.ra
##   [9] digest_0.6.33               png_0.1-8                   proxy_0.4-27                ggrepel_0.9
##  [13] deldir_1.0-9                parallelly_1.36.0           magick_2.7.5                MASS_7.3-60
##  [17] reshape2_1.4.4              httpuv_1.6.11               BiocGenerics_0.46.0         withr_2.5.0
##  [21] xfun_0.40                   ggfun_0.1.2                 ellipsis_0.3.2              survival_3
##  [25] memoise_2.0.1               s2_1.1.4                    profvis_0.3.8               ggsci_3.0.0
##  [29] systemfonts_1.0.4           tidytree_0.4.5              ragg_1.2.5                  zoo_1.8-12
##  [33] pbapply_1.7-2               R.oo_1.25.0                 spData_2.3.0                Formula_1.2
##  [37] prettyunits_1.1.1           promises_1.2.1              httr_1.4.6                  globals_0.1
##  [41] fitdistrplus_1.1-11         ps_1.7.5                    rstudioapi_0.15.0           units_0.8-3
##  [45] miniUI_0.1.1.1              generics_0.1.3              base64enc_0.1-3             processx_3
##  [49] S4Vectors_0.38.1            zlibbioc_1.46.0             ggraph_2.1.0                polyclip_1
##  [53] GenomeInfoDbData_1.2.10     xtable_1.8-4                stringr_1.5.0               desc_1.4.2
##  [57] evaluate_0.21               S4Arrays_1.0.5              GenomicRanges_1.52.0        bookdown_0
##  [61] irlba_2.3.5.1               colorspace_2.1-0            ROCR_1.0-11                 reticulate_
##  [65] spatstat.data_3.0-1         magrittr_2.0.3              lmtest_0.9-40               spdep_1.2-8
##  [69] later_1.3.1                 viridis_0.6.4               lattice_0.22-5              spatstat.ge
##  [73] future.apply_1.11.0         scattermore_1.2             XML_3.99-0.14               cowplot_1.1
##  [77] matrixStats_1.0.0           RcppAnnoy_0.0.21            class_7.3-22                Hmisc_5.1-0
##  [81] pillar_1.9.0                nlme_3.1-163                SeuratWrappers_0.3.1        compiler_4
##  [85] RSpectra_0.16-1             stringi_1.7.12              sf_1.0-14                   tensor_1.5
```

```
##  [89] minqa_1.2.5                SummarizedExperiment_1.30.2 devtools_2.4.5              plyr_1.8.8
##  [93] crayon_1.5.2               abind_1.4-5                 gridGraphics_0.5-1          ggtext_0.1
##  [97] graphlayouts_1.0.0         terra_1.7-39                dplyr_1.1.2                 codetools_
## [101] textshaping_0.3.6          openssl_2.1.0               monocle3_1.3.4              e1071_1.7-
## [105] plotly_4.10.2              mime_0.12                   leidenbase_0.1.25           splines_4.
## [109] Rcpp_1.0.11                fastDummies_1.7.3           grr_0.9.5                   gridtext_0
## [113] knitr_1.43                 utf8_1.2.3                  lme4_1.1-34                 fs_1.6.3
## [117] listenv_0.9.0              checkmate_2.2.0             pkgbuild_1.4.2              ggplotify_
## [121] Matrix_1.6-1               tibble_3.2.1                callr_3.7.3                 svglite_2.
## [125] tweenr_2.0.2               pkgconfig_2.0.3             tools_4.3.2                 cachem_1.0
## [129] viridisLite_0.4.2          DBI_1.1.3                   fastmap_1.1.1               rmarkdown_
## [133] scales_1.2.1               usethis_2.2.2               pbmcapply_1.5.1             ica_1.0-3
## [137] officer_0.6.2              patchwork_1.1.2             BiocManager_1.30.22         dotCall64_
## [141] RANN_2.6.1                 rpart_4.1.23                ggimage_0.3.3               farver_2.1
## [145] tidygraph_1.2.3            wk_0.7.3                    yaml_2.3.7                  MatrixGene
## [149] foreign_0.8-86             cli_3.6.1                   purrr_1.0.2                 stats4_4.3
## [153] leiden_0.4.3               lifecycle_1.0.3             askpass_1.1                 uwot_0.1.1
## [157] Biobase_2.60.0             sessioninfo_1.2.2           backports_1.4.1             gtable_0.3
## [161] ggridges_0.5.4             progressr_0.14.0            parallel_4.3.2              ape_5.7-1
## [165] testthat_3.1.10            jsonlite_1.8.7              RcppHNSW_0.4.1              bitops_1.0
## [169] assertthat_0.2.1           brio_1.1.3                  Rtsne_0.16                  yulab.utils
## [173] spatstat.utils_3.0-3       zip_2.3.0                   R.utils_2.12.2              lazyeval_0
## [177] shiny_1.7.5                htmltools_0.5.6             sctransform_0.4.0           tinytex_0.
## [181] glue_1.6.2                 spam_2.9-1                  XVector_0.40.0              RCurl_1.98
## [185] qpdf_1.3.2                 treeio_1.24.3               rprojroot_2.0.3             classInt_0
## [189] jpeg_0.1-10                gridExtra_2.3               boot_1.3-28                 igraph_1.5
## [193] R6_2.5.1                   tidyr_1.3.0                 labeling_0.4.2              cluster_2.
## [197] pkgload_1.3.2.1            grImport2_0.2-0             aplot_0.2.0                 GenomeInfo
## [201] nloptr_2.0.3               DelayedArray_0.26.7         tidyselect_1.2.0            htmlTable_
## [205] ggforce_0.4.1              xml2_1.3.5                  future_1.33.0               rsvd_1.0.5
## [209] munsell_0.5.0              KernSmooth_2.23-22          BiocStyle_2.28.0            rsvg_2.4.0
## [213] data.table_1.14.8          htmlwidgets_1.6.2           RColorBrewer_1.1-3          rlang_1.1.
## [217] spatstat.sparse_3.0-2      spatstat.explore_3.2-1      uuid_1.1-0                  remotes_2.
## [221] fansi_1.0.4
```

Reference

1. Gao, R. *et al.* Delineating copy number and clonal substructure in human tumors from single-cell transcriptomes. *Nature Biotechnology* **39**, 599–608 (2021).

2. Qiu, X. *et al.* Reversed graph embedding resolves complex single-cell trajectories. *Nature Methods* **14**, (2017).

3. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology* **32**, (2014).

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

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

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