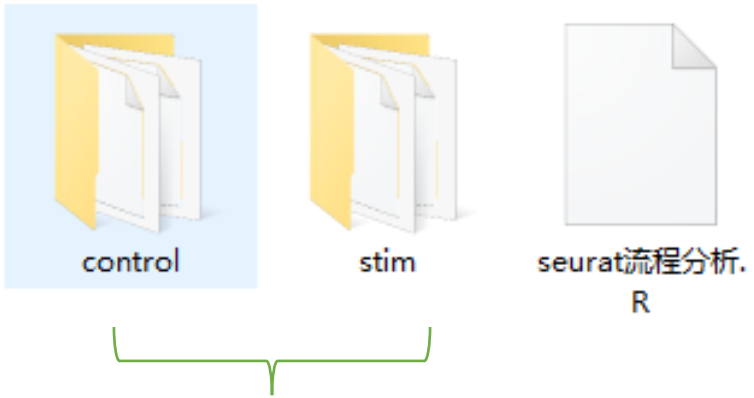
基迪奥生物 GENE DENOVO | 专注科研服务

基于seurat的基础分析流程

基迪奥生物

课程附件说明

- 示例数据与R脚本



1. 案例介绍：系统性红斑狼疮患者干扰素治疗前PBMC样本（con）和干扰素治疗后(stim)的10X数据
2. 课件中的R脚本，可用记事本、Rstudio等打开

课前准备

- 安装软件：R，Rstudio
- 安装R包：Seurat、dplyr、patchwork、harmony

R包安装方法

- #安装所需R包，安装过的无需安装；
- install.packages（“Seurat”）
- install.packages（“dplyr")
- install.packages（“patchwork”）
- install.packages（“ggplot2”）
- devtools::install_github("immunogenomics/harmony")

- #加载R包，无报错信息表示安装成功
- library(Seurat)
- library(dplyr)
- library(patchwork)
- library(ggplot2)

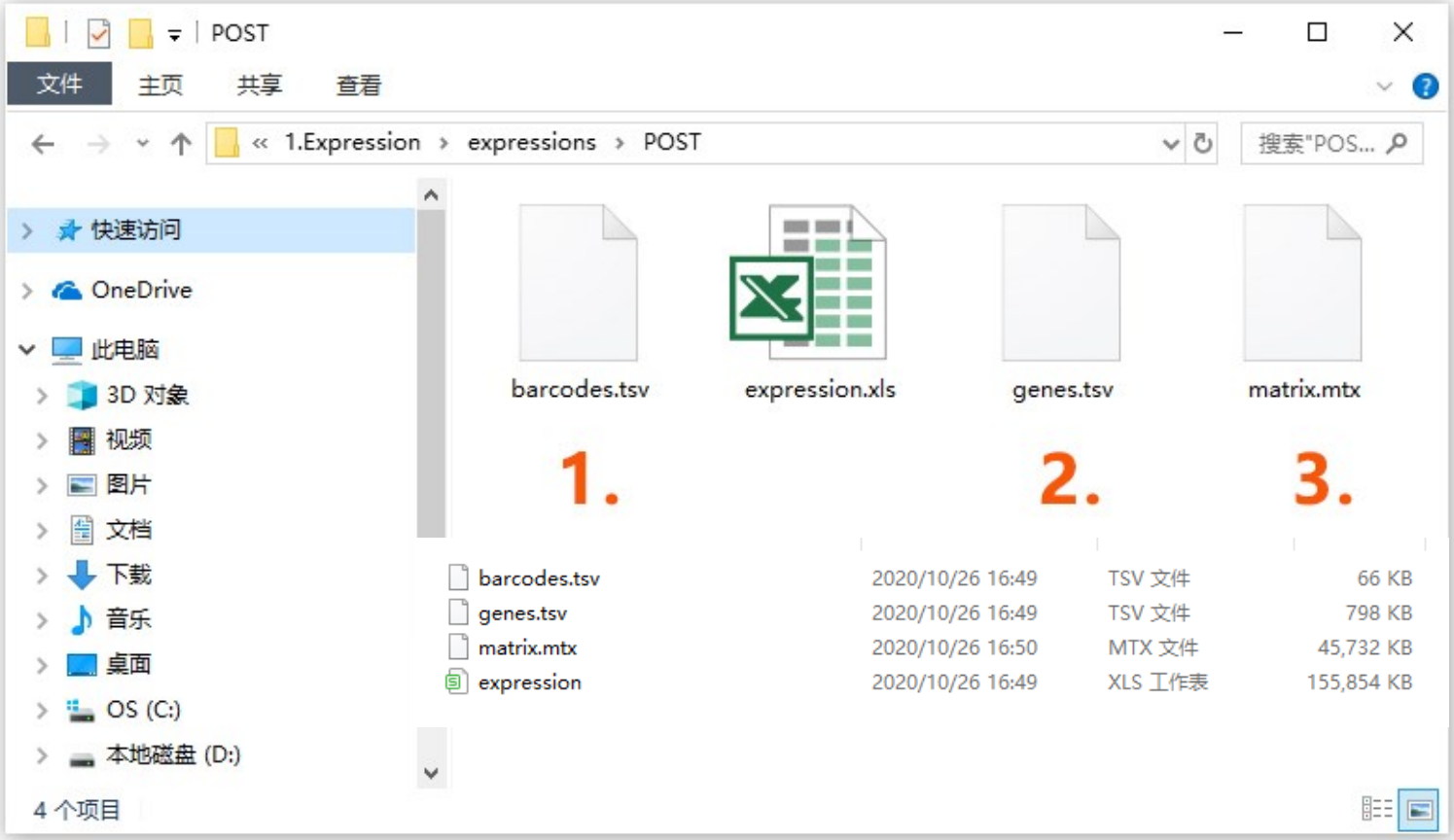
目录

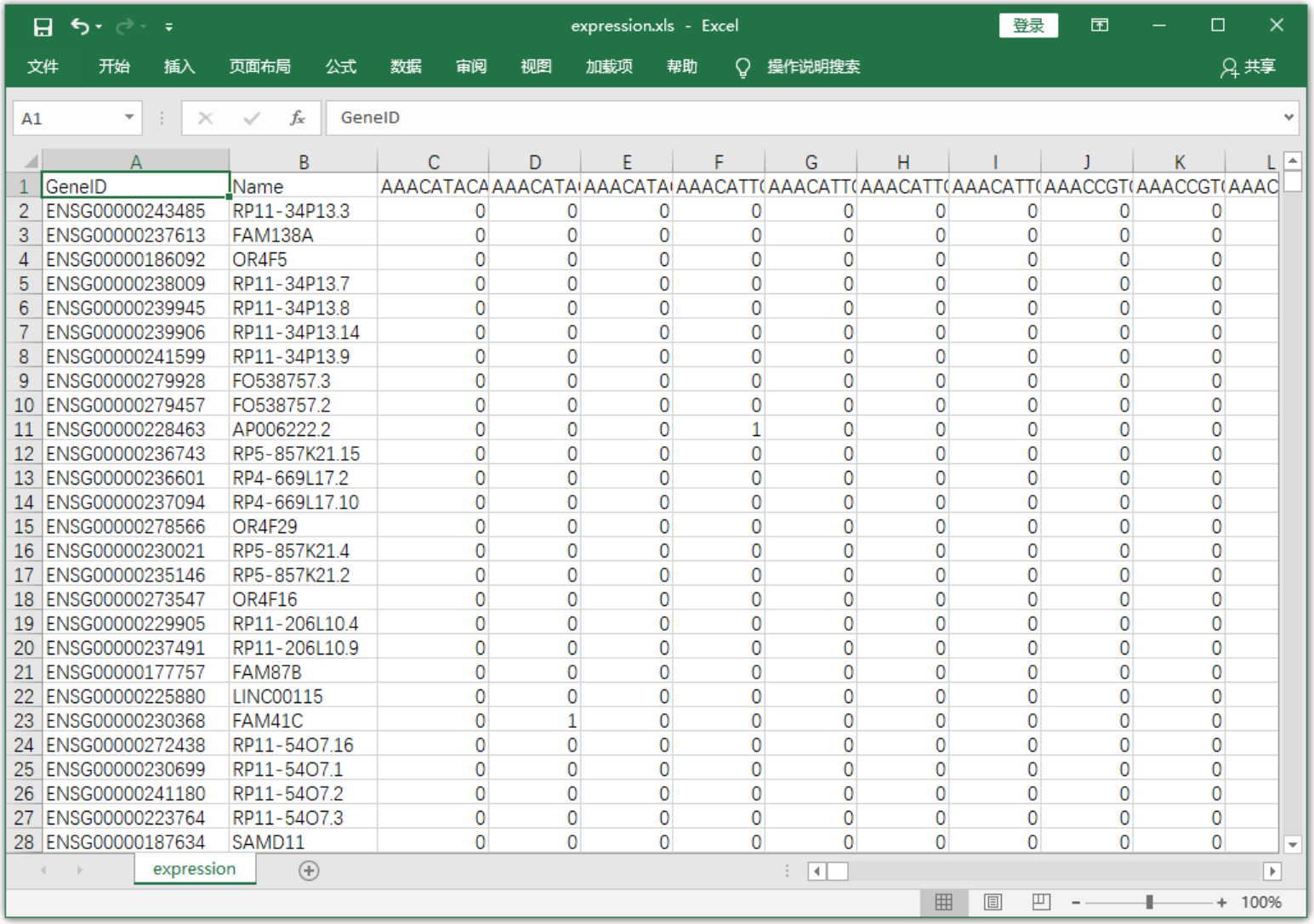
标准流程分析

1. 数据导入
2. 创建Seurat对象与数据过滤
3. 标准化
4. 细胞分类
5. 非线性降维可视化
6. 为分群重新指定细胞类型

1. 数据导入

数据文件内容，如下图。1，2，3为Cell Ranger生成的稀疏矩阵，xls后缀的为常规矩阵。





expression.xls - Excel

登录

文件 开始 插入 页面布局 公式 数据 审阅 视图 加载项 帮助 操作说明搜索 共享

A1 GeneID

	A	B	C	D	E	F	G	H	I	J	K	L
1	GeneID	Name	AAACATACA	AAACATA(	AAACATA(	AAACATT(	AAACATT(	AAACATT(	AAACATT(	AAACCGT(	AAACCGT(	AAAC
2	ENSG00000243485	RP11-34P13.3	0	0	0	0	0	0	0	0	0	
3	ENSG00000237613	FAM138A	0	0	0	0	0	0	0	0	0	
4	ENSG00000186092	OR4F5	0	0	0	0	0	0	0	0	0	
5	ENSG00000238009	RP11-34P13.7	0	0	0	0	0	0	0	0	0	
6	ENSG00000239945	RP11-34P13.8	0	0	0	0	0	0	0	0	0	
7	ENSG00000239906	RP11-34P13.14	0	0	0	0	0	0	0	0	0	
8	ENSG00000241599	RP11-34P13.9	0	0	0	0	0	0	0	0	0	
9	ENSG00000279928	FO538757.3	0	0	0	0	0	0	0	0	0	
10	ENSG00000279457	FO538757.2	0	0	0	0	0	0	0	0	0	
11	ENSG00000228463	AP006222.2	0	0	0	1	0	0	0	0	0	
12	ENSG00000236743	RP5-857K21.15	0	0	0	0	0	0	0	0	0	
13	ENSG00000236601	RP4-669L17.2	0	0	0	0	0	0	0	0	0	
14	ENSG00000237094	RP4-669L17.10	0	0	0	0	0	0	0	0	0	
15	ENSG00000278566	OR4F29	0	0	0	0	0	0	0	0	0	
16	ENSG00000230021	RP5-857K21.4	0	0	0	0	0	0	0	0	0	
17	ENSG00000235146	RP5-857K21.2	0	0	0	0	0	0	0	0	0	
18	ENSG00000273547	OR4F16	0	0	0	0	0	0	0	0	0	
19	ENSG00000229905	RP11-206L10.4	0	0	0	0	0	0	0	0	0	
20	ENSG00000237491	RP11-206L10.9	0	0	0	0	0	0	0	0	0	
21	ENSG00000177757	FAM87B	0	0	0	0	0	0	0	0	0	
22	ENSG00000225880	LINC00115	0	0	0	0	0	0	0	0	0	
23	ENSG00000230368	FAM41C	0	1	0	0	0	0	0	0	0	
24	ENSG00000272438	RP11-54O7.16	0	0	0	0	0	0	0	0	0	
25	ENSG00000230699	RP11-54O7.1	0	0	0	0	0	0	0	0	0	
26	ENSG00000241180	RP11-54O7.2	0	0	0	0	0	0	0	0	0	
27	ENSG00000223764	RP11-54O7.3	0	0	0	0	0	0	0	0	0	
28	ENSG00000187634	SAMD11	0	0	0	0	0	0	0	0	0	

expression

100%

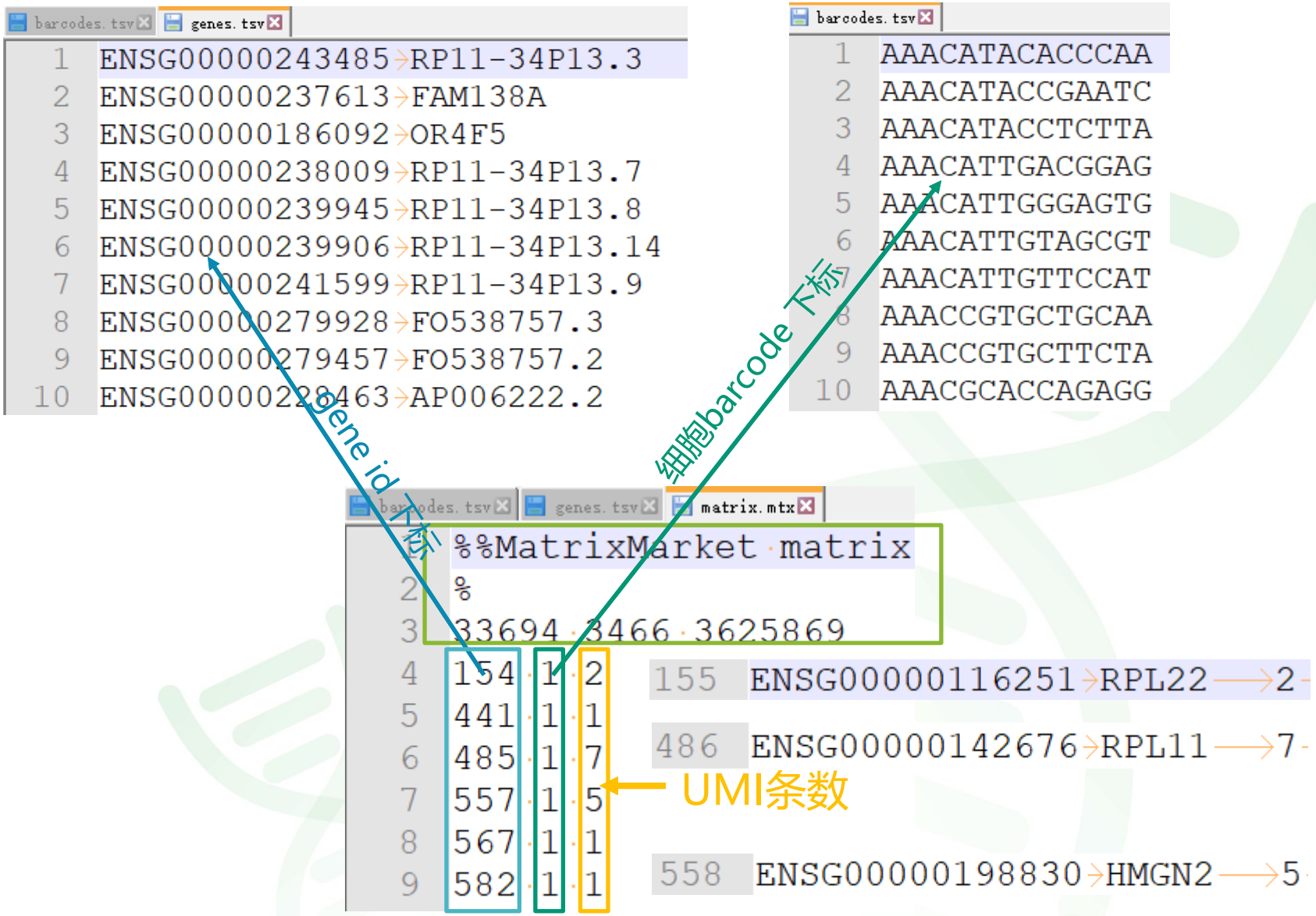
注：Excel打开需要2分钟……

Sparse matrix(稀疏矩阵)

在矩阵中，若数值为0的元素数目远多于非0元素，并且非0元素分布无规律时，则称该矩阵为稀疏矩阵。 /spɑ:s/

基本的定义是矩阵中的大多数元素为零，并且可以利用零元素**节约大量存储、程序运行时间**。

单细胞转录组的表达量数据，有大量的表达量为0的数据，符合稀疏矩阵的特点。


barcodes.tsv
genes.tsv
1 ENSG00000243485→RP11-34P13.3
2 ENSG00000237613→FAM138A
3 ENSG00000186092→OR4F5
4 ENSG00000238009→RP11-34P13.7
5 ENSG00000239945→RP11-34P13.8
6 ENSG00000239906→RP11-34P13.14
7 ENSG00000241599→RP11-34P13.9
8 ENSG00000279928→FO538757.3
9 ENSG00000279457→FO538757.2
10 ENSG00000228463→AP006222.2
barcodes.tsv
1 AAACATACACCCAA
2 AAACATACCGAATC
3 AAACATACCTCTTA
4 AAACATTGACGGAG
5 AAACATTGGGAGTG
6 AAACATTGTAGCGT
7 AAACATTGTTCCAT
8 AAACCGTGCTGCAA
9 AAACCGTGCTTCTA
10 AAACGCACCAGAGG
gene id 下标
细胞barcode 下标
barcodes.tsv
genes.tsv
matrix.mtx
%%MatrixMarket matrix
%
33694 3466 3625869
4 154 1 2
5 441 1 1
6 485 1 7
7 557 1 5
8 567 1 1
9 582 1 1
155 ENSG00000116251→RPL22 2
486 ENSG00000142676→RPL11 7
UMI条数
558 ENSG00000198830→HMGN2 5

1.1 干扰素治疗前样本的读取

#系统性红斑狼疮患者干扰素治疗前PBMC样本的读取

data_dir <- "E:/单细胞培训班/课件ppt/NO.1 10X单细胞概述及R语言入门/流程分析/数据及脚本/control/" ##指定数据所在目录

list.files(data_dir) ##列出文件名

```
> list.files(data_dir)  ##列出文件名
[1] "barcodes.tsv"    "expression.xls" "genes.tsv"      "matrix.mtx"
```

con_expression_matrix <- Read10X(data.dir = data_dir) ##读取数据

dim(con_expression_matrix) #查看维度，即基因数和细胞数

```
> dim(con_expression_matrix)    #查看维度，即基因数和细胞数
[1] 35635 14619
```

con[1:10,1:6] #查看矩阵（1~10行，1~6列，.表示0）

```
> con[1:10,1:6] #查看矩阵（1~10行，1~6列，.表示0）
10 x 6 sparse Matrix of class "dgCMatrix"
           con_CCTGCAACTCATTC-1 con_CCCAACTGGTGCAT-1 con_TACGGCCTTCAGAC-1 con_CCAACCTGTGCCAA-1 con_AGATCGTGTTGCGA-1
MIR1302-10                    .                    .                    .                    .                    .
FAM138A                       .                    .                    .                    .                    .
OR4F5                         .                    .                    .                    .                    .
RP11-34P13.7                  .                    .                    .                    .                    .
RP11-34P13.8                  .                    .                    .                    .                    .
AL627309.1                    .                    .                    .                    .                    .
RP11-34P13.14                 .                    .                    .                    .                    .
RP11-34P13.9                  .                    .                    .                    .                    .
AP006222.2                    .                    .                    .                    .                    .
RP4-669L17.10                 .                    .                    .                    .                    .
           con_GCCTACACATAAGG-1
MIR1302-10                    .
FAM138A                       .
OR4F5                         .
RP11-34P13.7                  .
RP11-34P13.8                  .
AL627309.1                    .
RP11-34P13.14                 .
RP11-34P13.9                  .
AP006222.2                    .
RP4-669L17.10                 .
```

1.2 干扰素治疗前样本的读取

#系统性红斑狼疮患者干扰素治疗前PBMC样本的读取

data_dir <- "E:/单细胞培训班/课件ppt/NO.1 10X单细胞概述及R语言入门/流程分析/数据及脚本/control/"

list.files(data_dir)

```
> list.files(data_dir)  ##列出文件名
[1] "barcodes.tsv"    "expression.xls" "genes.tsv"       "matrix.mtx"
```

stim_expression_matrix <- Read10X(data.dir = data_dir)

dim(stim_expression_matrix)

```
> dim(stim_expression_matrix)
[1] 35635 14446
```

stim[1:10,1:6] #查看矩阵（1~10行，1~6列，.表示0）

```
> stim[1:10,1:6] #查看矩阵（1~10行，1~6列，.表示0）
10 x 6 sparse Matrix of class "dgCMatrix"
           stim_CACAACGAGGGTGA-1 stim_GAGGCAGATCATTC-1 stim_GTCACCTGGGTCAT-1 stim_GCGTATGACTTCCG-1
MIR1302-10                     .                     .                     .                     .
FAM138A                        .                     .                     .                     .
OR4F5                          .                     .                     .                     .
RP11-34P13.7                   .                     .                     .                     .
RP11-34P13.8                   .                     .                     .                     .
AL627309.1                     .                     .                     .                     .
RP11-34P13.14                  .                     .                     .                     .
RP11-34P13.9                   .                     .                     .                     .
AP006222.2                     .                     .                     .                     .
RP4-669L17.10                  .                     .                     .                     .
           stim_CTTCACCTGTCGTA-1 stim_GCAGCGTGTAGTCG-1
MIR1302-10                     .                     .
FAM138A                        .                     .
OR4F5                          .                     .
RP11-34P13.7                   .                     .
RP11-34P13.8                   .                     .
AL627309.1                     .                     .
RP11-34P13.14                  .                     .
RP11-34P13.9                   .                     .
AP006222.2                     .                     .
RP4-669L17.10                  .                     .
```

2. 创建seurat对象与数据过滤

```
#创建seurat对象和数据过滤
#数据集中测到的少于200个基因的细胞（min.features = 200）和少于3个细胞覆盖的基因（min.cells = 3）被过滤掉

con <- CreateSeuratObject(counts = con, project = "control", min.cells = 3, min.features = 200)

stim <- CreateSeuratObject(counts = stim, project = "stimulus", min.cells = 3, min.features = 200)

#两个不同的样本合并
seurat_object <- merge(con,stim)
```

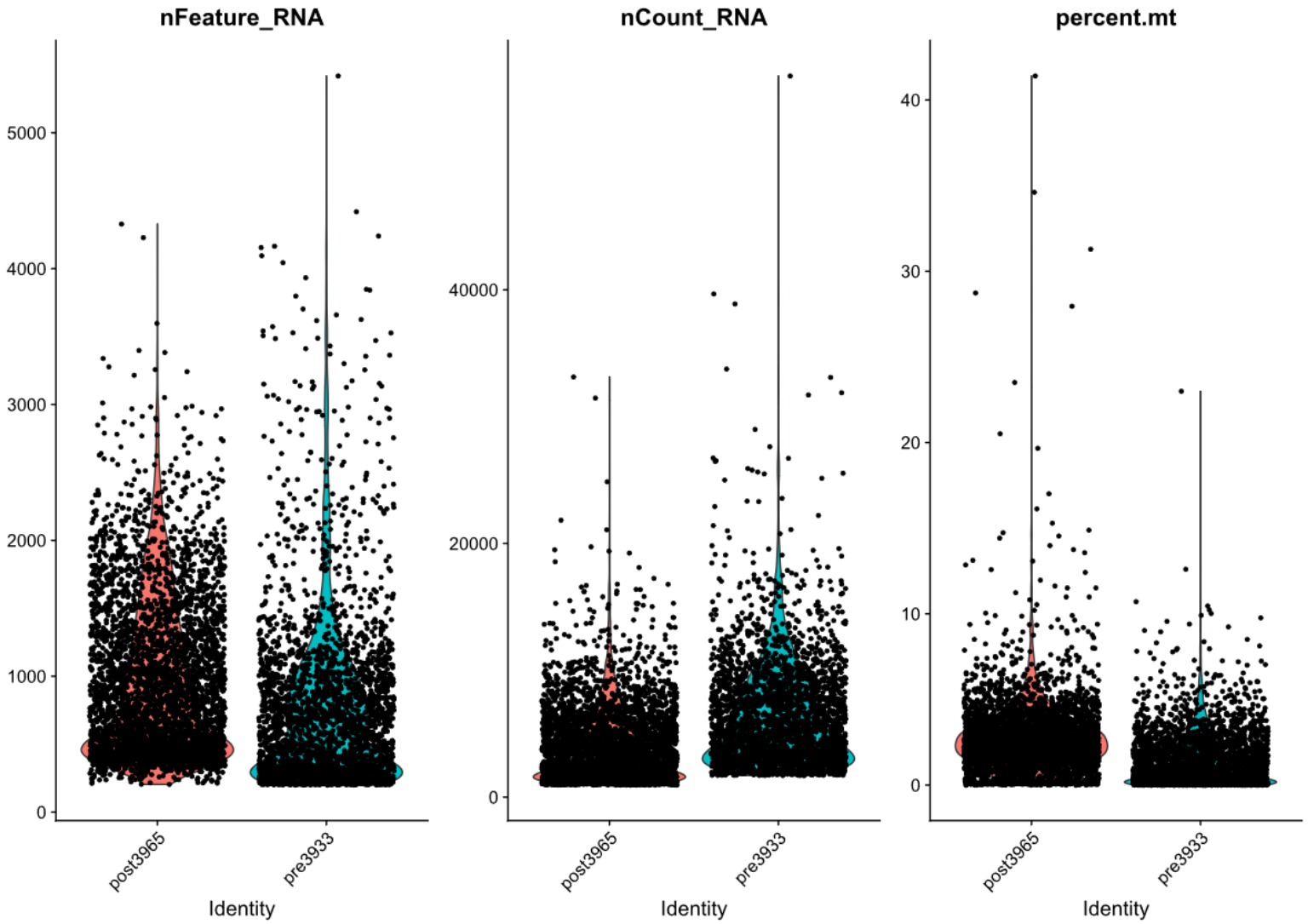
#计算每个细胞的线粒体基因转录本数的百分比（%），使用[[ ]] 操作符存放到 metadata 中;

```
seurat_object[["percent.mt"]] <- PercentageFeatureSet(seurat_object, pattern = "^MT-")
```

#nFeature_RNA代表每个细胞测到的基因数目，nCount代表每个细胞测到所有基因的表达量之和，percent.mt代表测到的线粒体基因的比例。

```
VlnPlot(seurat_object, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
```

基因数，细胞数和线粒体占比



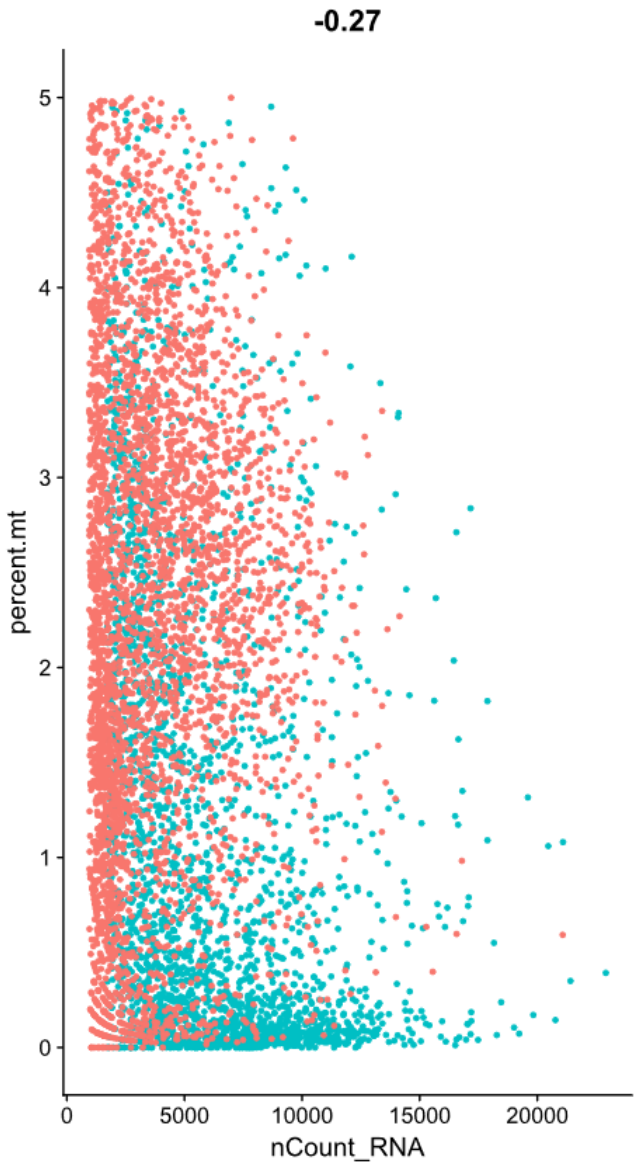
#过滤细胞：保留 gene 数大于 200 小于 2500 的细胞；目的是去掉空 GEMs 和 1 个 GEMs 包 含 2 个以上细胞的数据；而保留线粒体基因的转录本数低于 5%的细胞，为了过滤掉死细胞 等低质量的细胞数据。

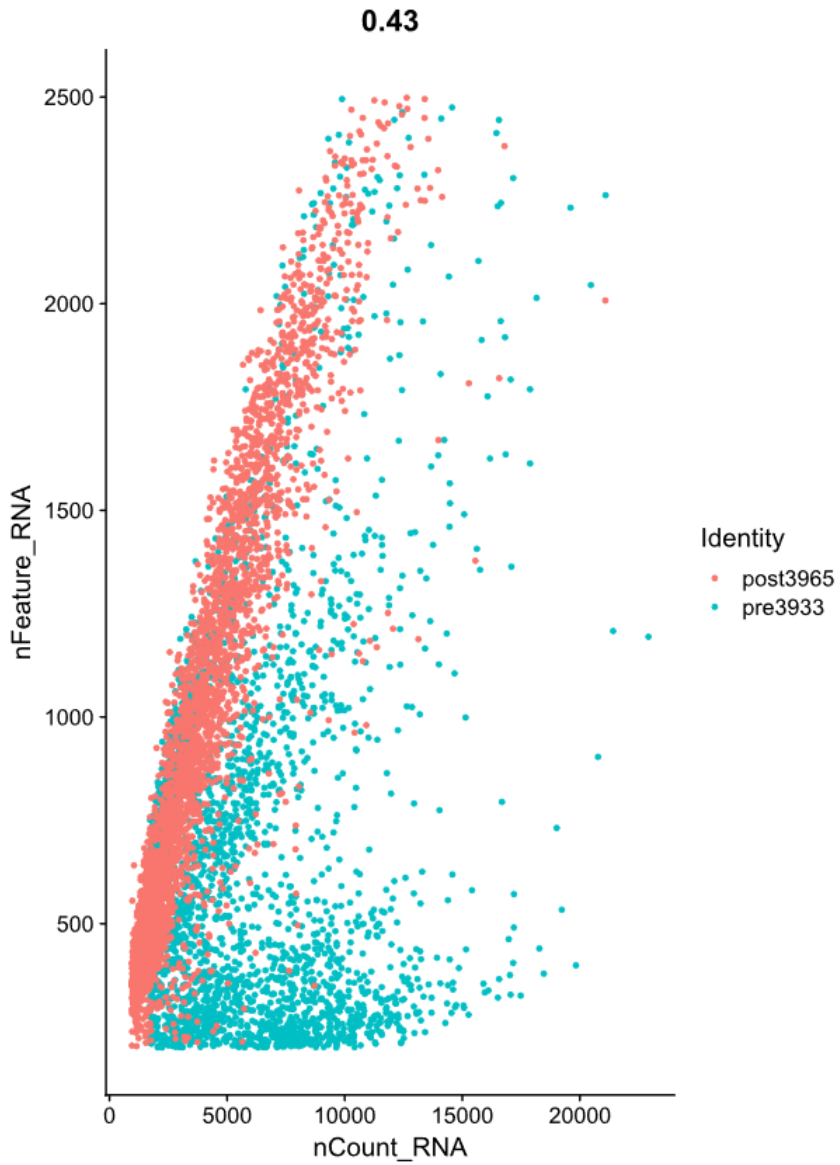
```
seurat_object <- subset(seurat_object, subset = nFeature_RNA > 200 & nFeature_RNA < 1500 & percent.mt < 5)
```

对过滤后的 QC metrics 进行可视化（绘制散点图）；

```
plot1 <- FeatureScatter(seurat_object, feature1 = "nCount_RNA", feature2 = "percent.mt")
plot2 <- FeatureScatter(seurat_object, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
plot1 + plot2
```

过滤后的 QC metrics 可视化





3. 标准化

#表达量数据标准化：LogNormalize 的算法：A = log(1 + (UMIA ÷ UMITotal) × 10000)

```
seurat_object <- NormalizeData(seurat_object, normalization.method = "LogNormalize", scale.factor = 10000)
```

#鉴定细胞间表达量高变的基因（feature selection），用于下游分析，PCA

#这一步的目的是鉴定出细胞与细胞之间表达量相差很大的基因，用于后续鉴定细胞类型，我们使用默认参数，即“vst”方法选取2000个高变基因。

```
seurat_object <- FindVariableFeatures(seurat_object, selection.method = "vst", nfeatures = 2000)
```

提取表达量变变化最高的 10 个基因;

top10 <- head(VariableFeatures(seurat_object), 10)

Top10

```
> top10 <- head(VariableFeatures(seurat_object), 10)
> top10
 [1] "HBB"      "HBA2"     "HBA1"     "APOBEC3B" "CCL4"     "CCL3"     "CCL7"     "CCL8"
 [9] "IL1B"     "CCL2"
```

绘制带有和不带有标签的变量特征的散点图

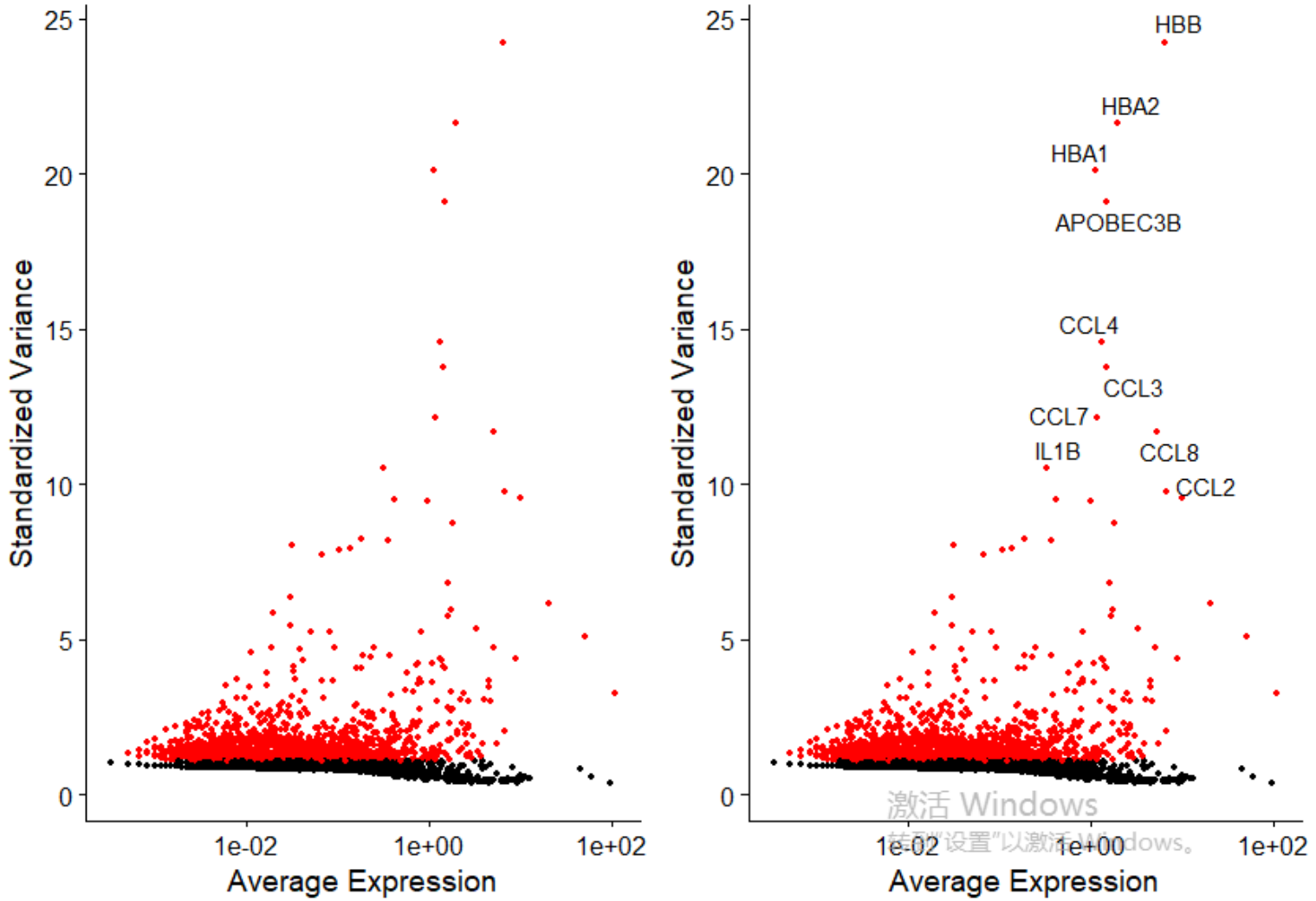
plot3 <- VariableFeaturePlot(seurat_object)+NoLegend()

plot4 <- LabelPoints(plot = plot3, points = top10, repel = TRUE,

xnudge=0, ynudge=0)

plot3+plot4

变量特征的散点图



4. 细胞分类

4.1 对数据集进行降维

4.2 定义数据集的分群个数

4.3 细胞分类

4.1 对数据集进行降维

```
#使用ScaleData()进行数据标准化；默认只是标准化高变基因（2000 个），速度更快，不影响 PCA 和分群，但影响热图的绘制。
seurat_object <- ScaleData(seurat_object,vars.to.regress = "percent.mt")
#而对所有基因进行标准化的方法如下：
all.genes <- rownames(seurat_object)
seurat_object <- ScaleData(seurat_object, features = all.genes,
vars.to.regress = "percent.mt") ##耗时2min

#线性降维（PCA）,默认用高变基因集，但也可通过 features 参数自己指定；
seurat_object <- RunPCA(seurat_object, features =
VariableFeatures(object = seurat_object))
```

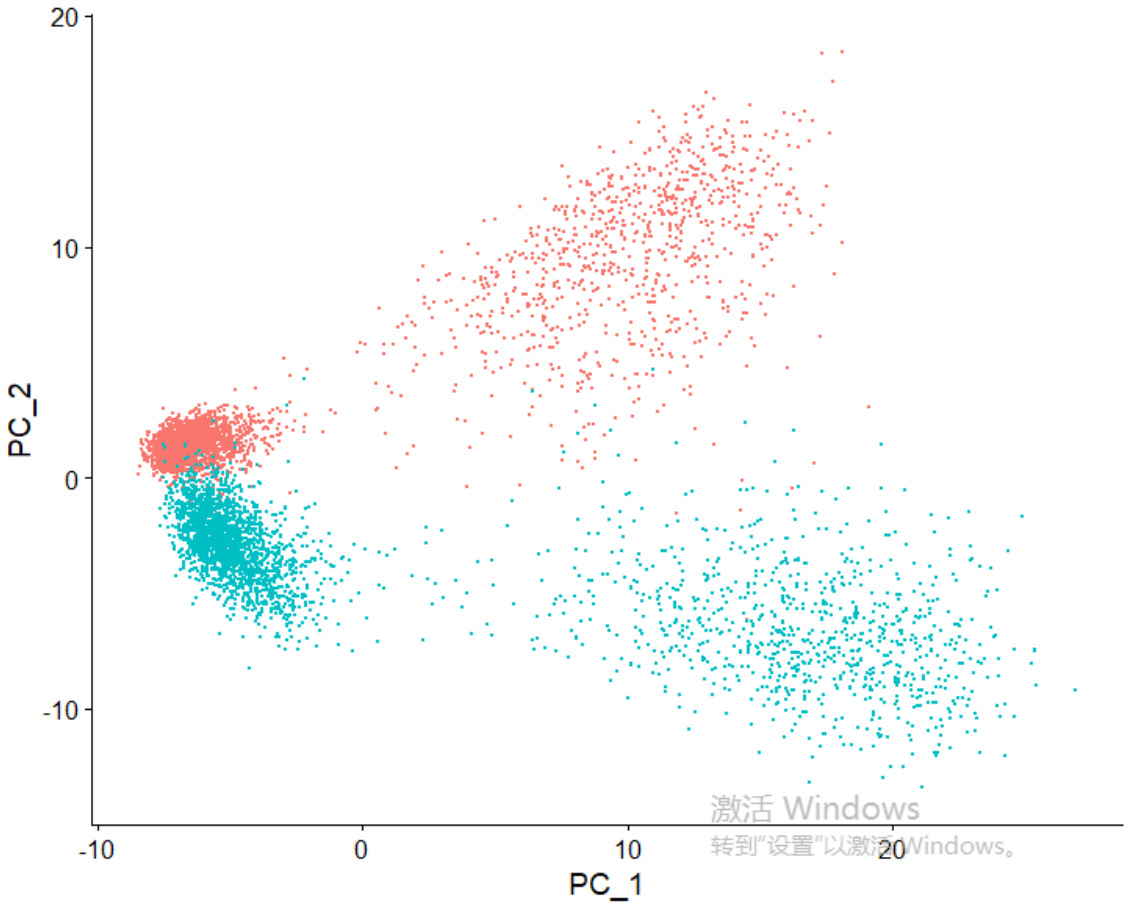
检查 PCA 分群结果， 这里只展示前 12 个 PC,每个 PC 只显示 3 个基因;

print(seurat_object[["pca"]], dims = 1:12, nfeatures = 3)

```
> print(seurat_object[["pca"]], dims = 1:12, nfeatures = 3)
PC_ 1
Positive:  C15orf48, TYROBP, CST3
Negative:  CCR7, LTB, ITM2A
PC_ 2
Positive:  IL8, CD14, CLEC5A
Negative:  ISG15, ISG20, IFIT3
PC_ 3
Positive:  CCR7, HLA-DQA1, CD79A
Negative:  NKG7, GNLY, GZMB
PC_ 4
Positive:  GZMB, NKG7, HLA-DQA1
Negative:  LTB, IL7R, TRAT1
PC_ 5
Positive:  CCL7, CCL2, PLA2G7
Negative:  VMO1, FCGR3A, MS4A4A
PC_ 6
Positive:  CD79A, MS4A1, CD74
Negative:  CACYBP, RSRC2, HSPH1
PC_ 7
Positive:  IDO1, IL27, IL1RN
Negative:  IFI6, IFIT3, MX1
PC_ 8
Positive:  CD14, CD69, NFKBIA
Negative:  PPBP, GNG11, SDPR
PC_ 9
Positive:  PKIB, CALCRL, CCL22
Negative:  CD79A, MS4A1, CCL4
```

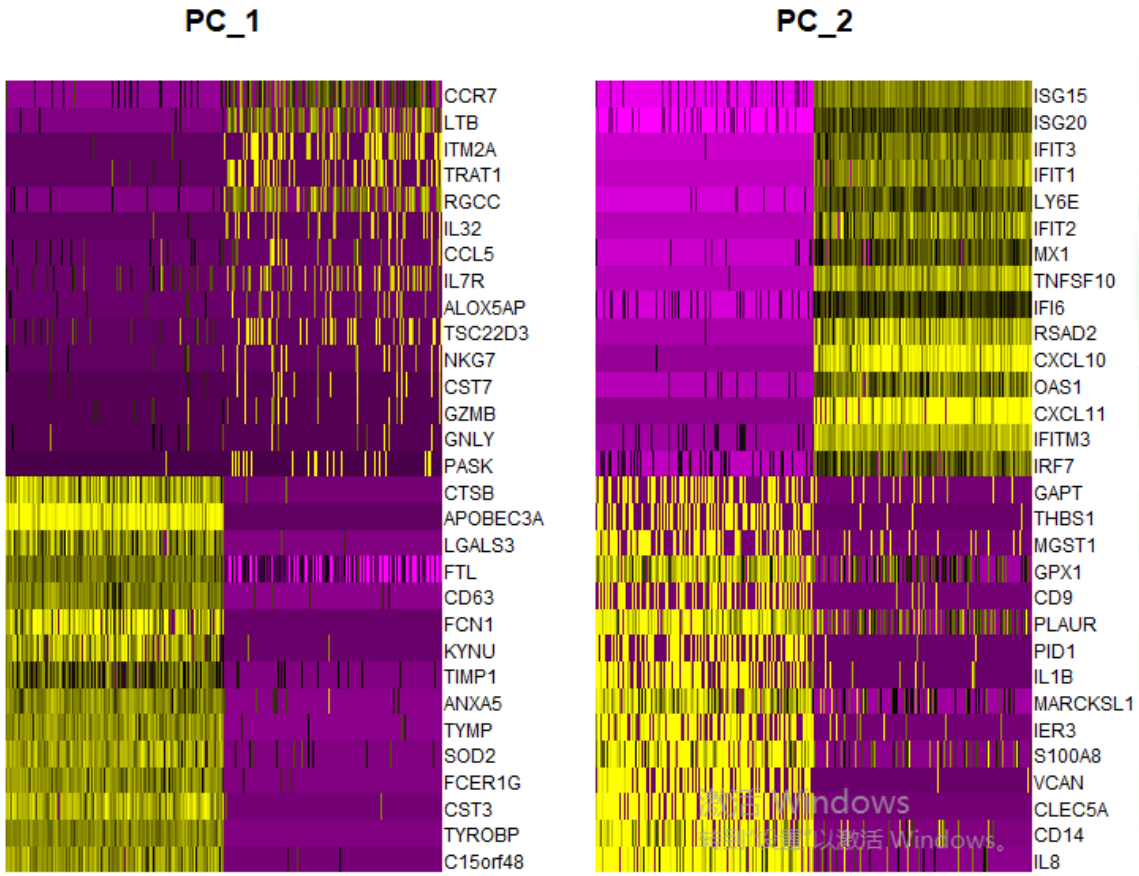
#绘制 pca 散点图； 去除图例

```
DimPlot(seurat_object, reduction = "pca")+ NoLegend()
```



#画前 2 个主成分的热图；

DimHeatmap(seurat_object, dims = 1:2, cells = 500, balanced = TRUE)

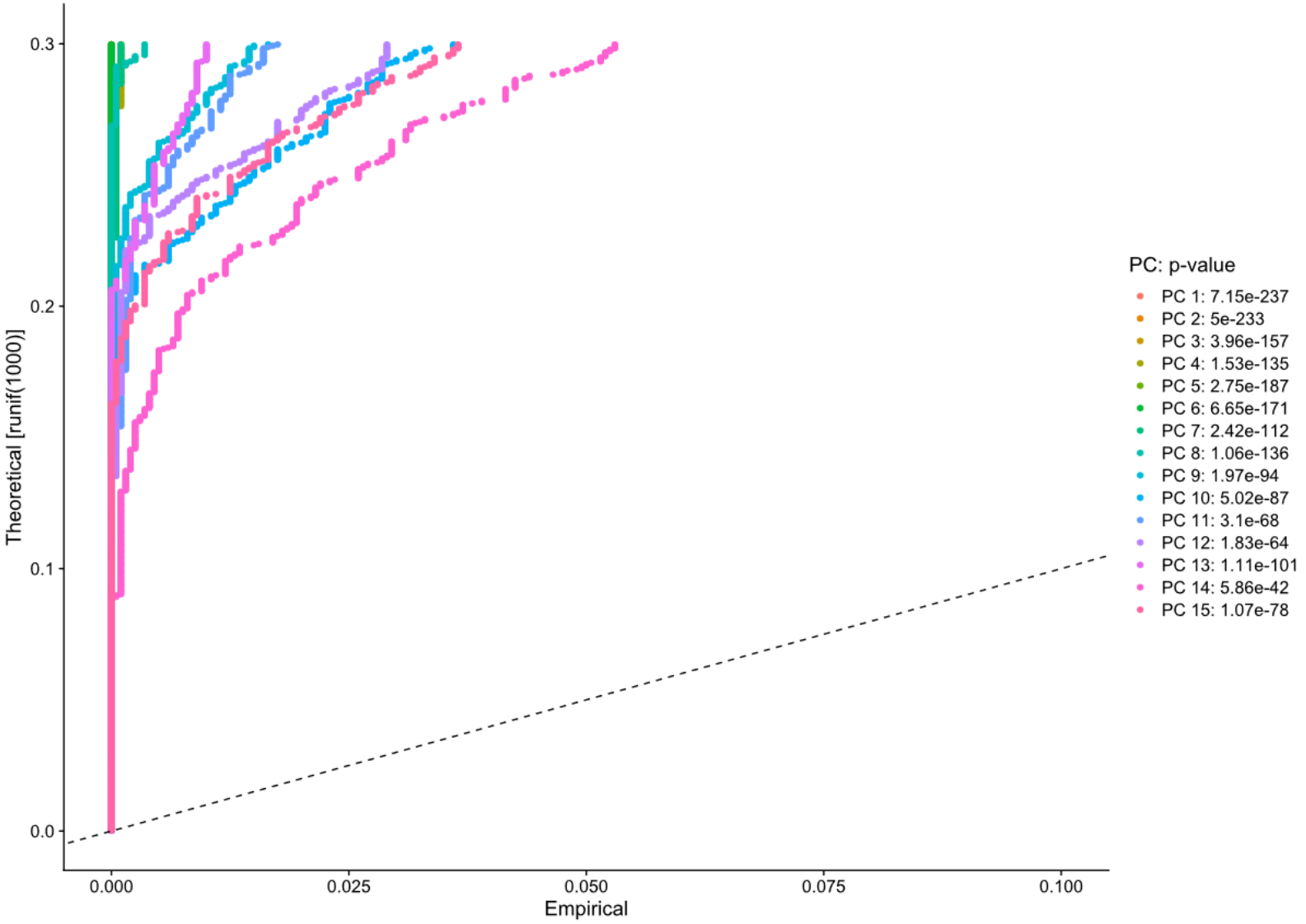


4.2 定义数据集的有效主成分

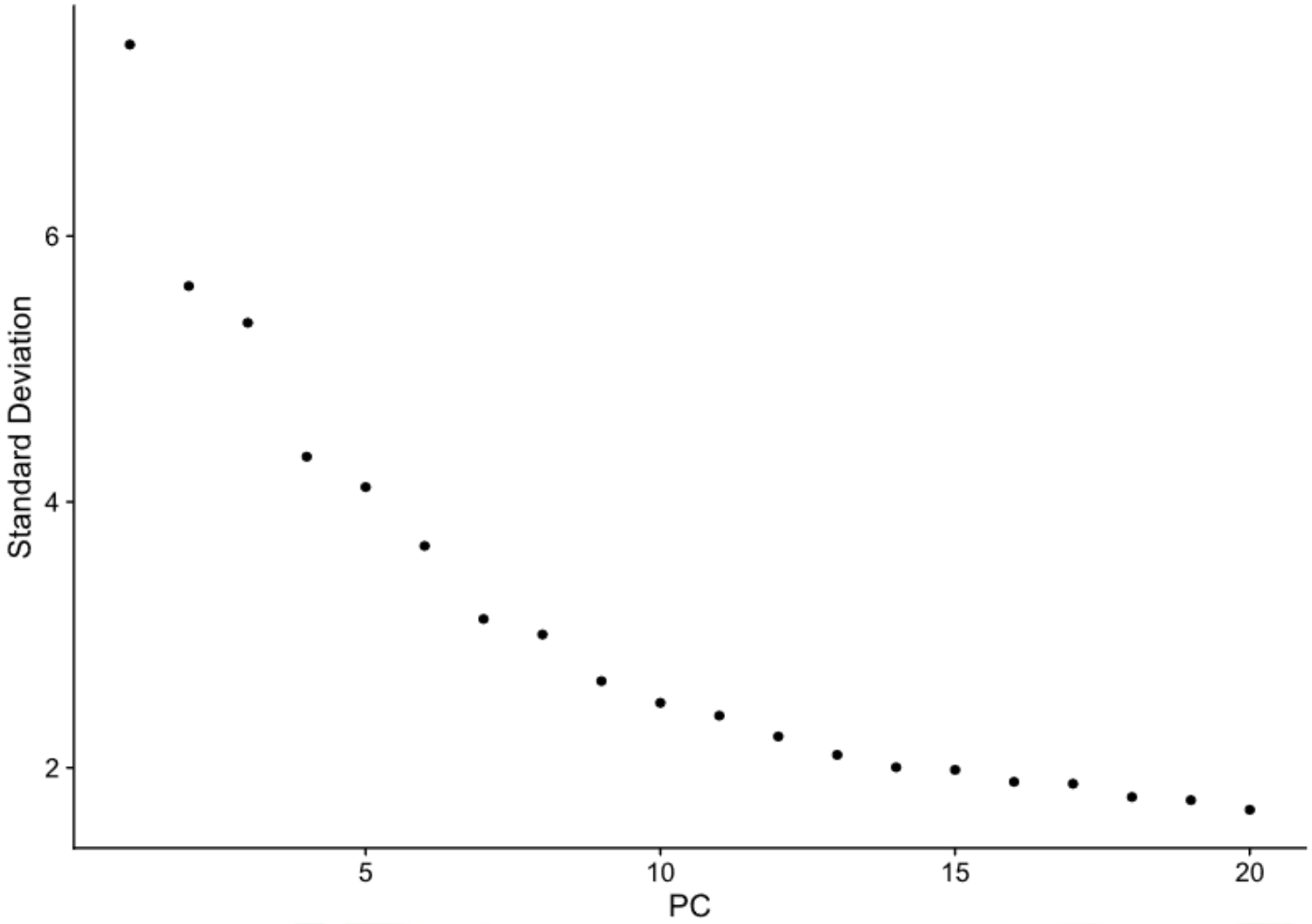
```
##方法 1：Jackstraw 置换检验算法；重复取样（原数据的 1%），重跑PCA,鉴定p-value较小的PC；计算‘null distribution’(即零假设成立时)时的基因 scores;
seurat_object <- JackStraw(seurat_object, num.replicate = 100) ##耗时3min
seurat_object <- ScoreJackStraw(seurat_object, dims = 1:20)
JackStrawPlot(seurat_object, dims = 1:15)

#方法 2：肘部图（碎石图），基于每个主成分对方差解释率的排名；
ElbowPlot(seurat_object)
```

Jackstraw 置换检验算法



肘部图（碎石图）



4.3 细胞分群

#基于PCA空间中的欧氏距离计算 nearest neighbor graph，优化任意两个细胞间的距离权重（输入上一步得到的 PC 维数）；

```
seurat_object <- FindNeighbors(seurat_object, dims = 1:10)
```

#接着优化模型，resolution 参数决定下游聚类分析得到的分群数，对于3K左右的细胞，设为0.4-1.2能得到较好的结果(官方说明)；如果数据量增大，该参数也应该适当增大；

```
seurat_object <- FindClusters(seurat_object, resolution = 0.5)
```

#使用 Idents（）函数可查看不同细胞的分群； 查看前8个细胞的分群ID

head(Idents(seurat_object), 5)

```
> head(Idents(seurat_object), 5)
.con_con_CAGCAATGGCCATA-1 .con_con_CTCGAGCTCATTCT-1 .con_con_CTAGGTGACCCAAA-1 .con_con_CGCACTACTCCTAT-1 .con_con_TGAGACACGGGACA-1
                        0                         6                         0                         0                         6
Levels: 0 1 2 3 4 5 6 7 8 9 10 11 12
```

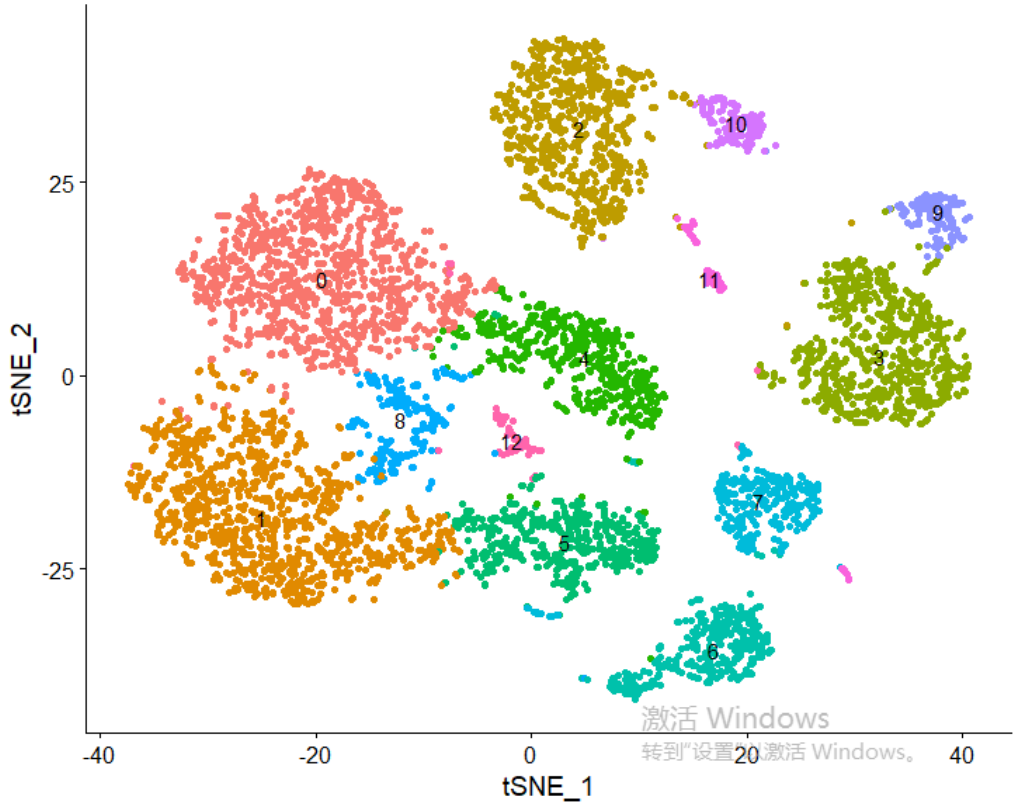
5. 非线性降维可视化

5.1 tSNE非线性降维

5.2 UMAP非线性降维

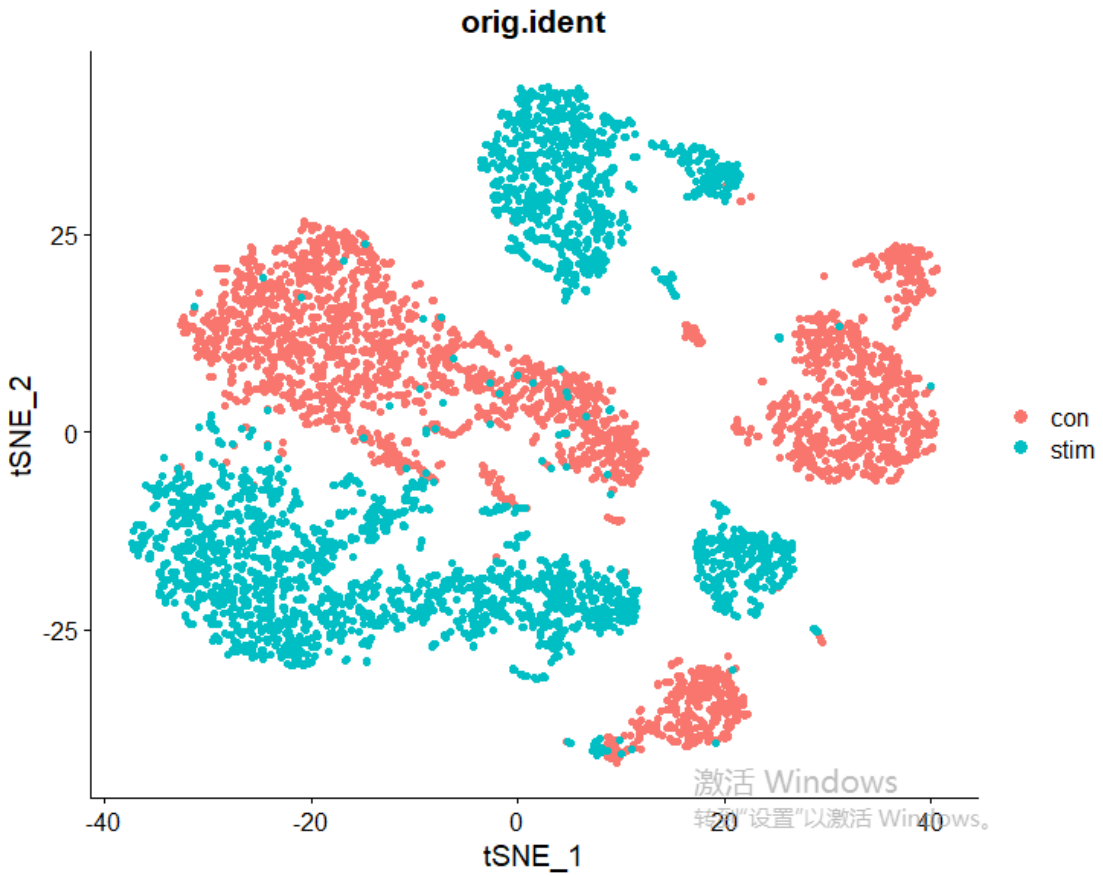
5.1 tSNE非线性降维

```
#tsne非线性降维
seurat_object <- RunTSNE(seurat_object, dims = 1:10)
#用TSNEPlot函数绘制tsne图
tsneplot<-TSNEPlot(seurat_object,label = TRUE, pt.size = 1.5)+ NoLegend()
tsneplot
```



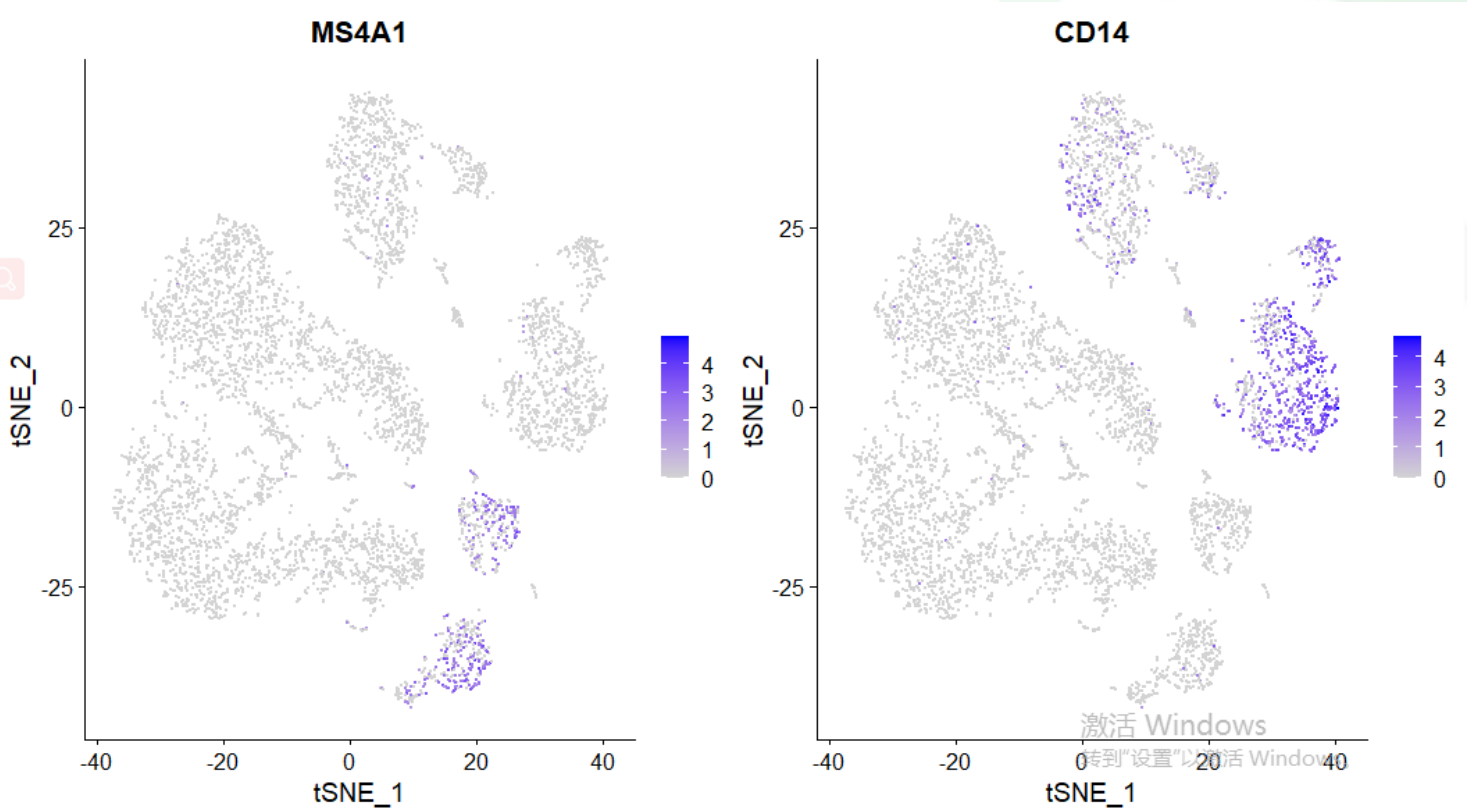
#用DimPlot函数绘制tsne图

```
tsneplot1 <- DimPlot(seurat_object, reduction = "tsne", group.by = "orig.ident", pt.size = 1.5)
tsneplot1
```



#绘制 Marker 基因的 tsne 图;

FeaturePlot(seurat_object, features = c("MS4A1", "CD14"))



5.2 UMAP非线性降维

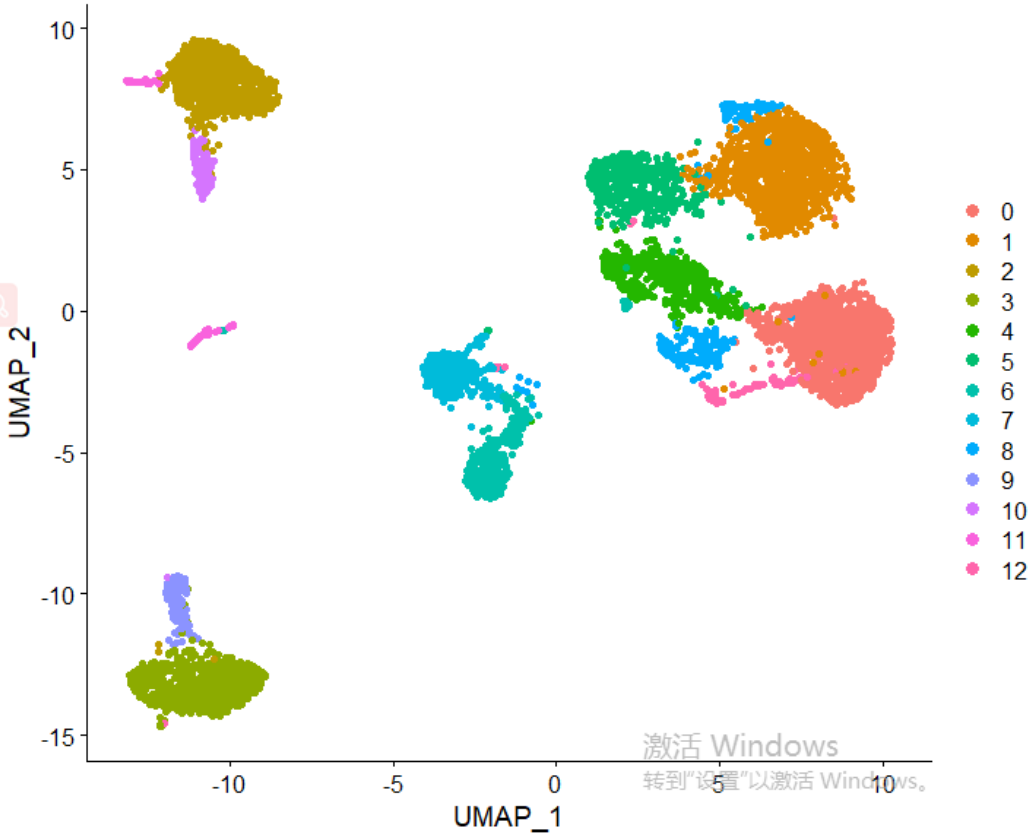
#UMAP非线性降维

```
seurat_object <- RunUMAP(seurat_object, dims = 1:10, label = T)
# 提取UMAP坐标值
head(seurat_object@reductions$umap@cell.embeddings)
```

```
> head(seurat_object@reductions$umap@cell.embeddings) # 提取UMAP坐标值。
                              UMAP_1     UMAP_2
.con_con_CAGCAATGGCCATA-1   8.136338 -0.6434816
.con_con_CTCGAGCTCATTCT-1  -1.785729 -5.9512848
.con_con_CTAGGTGACCCAAA-1   7.871753 -2.4003078
.con_con_CGCACTACTCCTAT-1   6.997654 -0.7175507
.con_con_TGAGACACGGGACA-1  -2.280376 -5.2749070
.con_con_TGATCGGATGTTTC-1   7.378167 -0.8028702
```

#用DimPlot函数绘制UMAP图

```
umapplot <- DimPlot(seurat_object, reduction = "umap",pt.size = 1.5)
umapplot
```



6. 为分群重新指定细胞类型

#为分群重新指定细胞类型

new.cluster.ids <- c("Naive CD4 T", "Memory CD4 T", "CD14+ Mono", "B", "CD8 T", "FCGR3A+ Mono", "NK", "DC", "Platelet","T","Eryth","Mk","HSPC")

#自定义名称

names(new.cluster.ids)

```
> names(new.cluster.ids)
NULL
```

levels(seurat_object)

```
> levels(seurat_object)
 [1] "0"  "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12"
```

#将seurat_object的水平属性赋值给new.cluster.ids的names属性;

```
names(new.cluster.ids) <- levels(seurat_object)
names(new.cluster.ids)
```

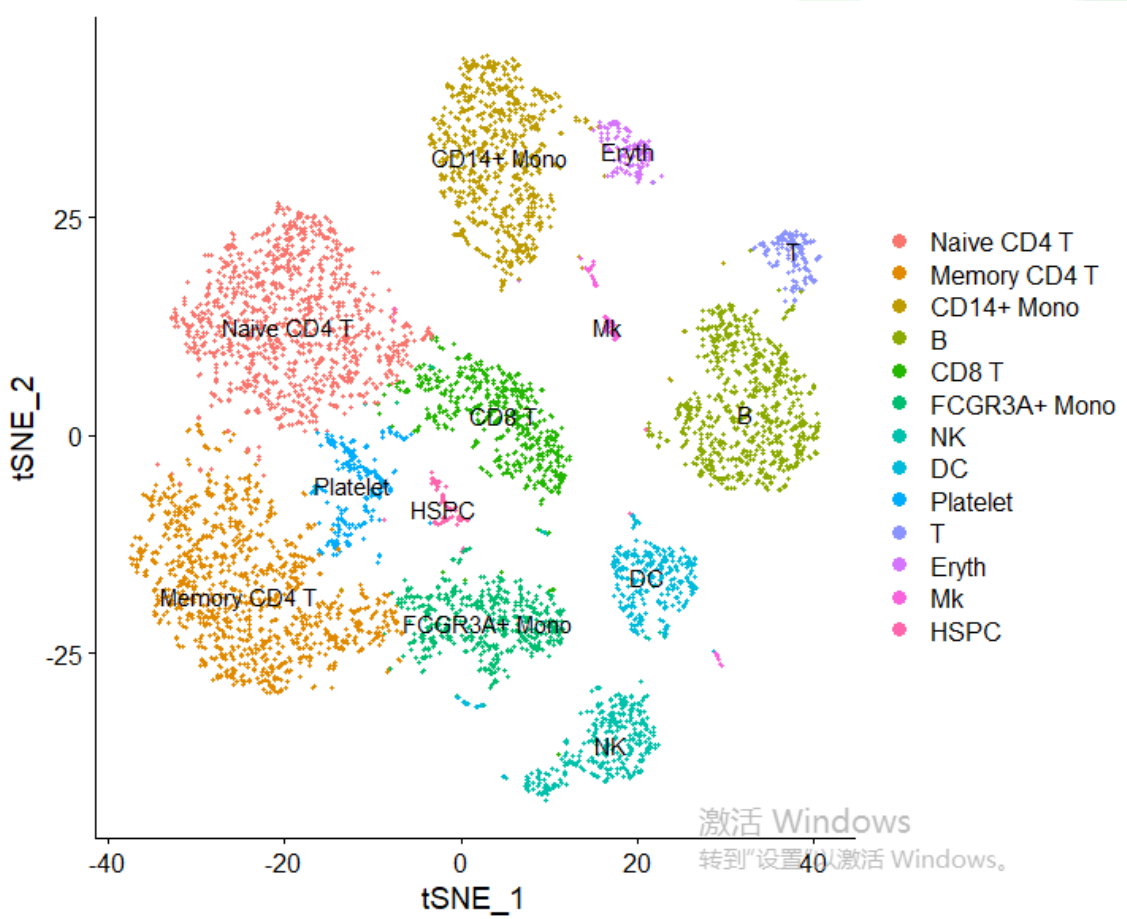
```
> names(new.cluster.ids)
 [1] "0"  "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12"
```

```
seurat_object <- RenameIdents(seurat_object, new.cluster.ids)
```

#绘制 tsne 图(修改标签后的)；

tsneplot2<-TSNEPlot(seurat_object,label = TRUE, pt.size = 0.8)

tsneplot2



7. 保存工作空间

```
#保存工作空间
save(seurat_object,file = "obj.Rda")
#查看当前目录，在改目录下查找obj.Rda文件
getwd()
```

目录

批次效应矫正

1. CCA矫正
2. harmony矫正

1. CCA矫正

1.1 数据导入（与标准流程相同）

```
#系统性红斑狼疮患者干扰素治疗前PBMC样本的读取
data_dir <- "E:/单细胞培训班/课件ppt/NO.1 10X单细胞概述及R语言入门/流程分析/数据及脚本/control/"  ##指定数据所在目录
list.files(data_dir)  ##列出文件名
con_expression_matrix <- Read10X(data.dir = data_dir) ##读取数据
dim(con_expression_matrix)   #查看维度，即基因数和细胞数
#系统性红斑狼疮患者干扰素治疗前PBMC样本的读取
data_dir <- "E:/单细胞培训班/课件ppt/NO.1 10X单细胞概述及R语言入门/流程分析/数据及脚本/control/"
list.files(data_dir)
stim_expression_matrix <- Read10X(data.dir = data_dir)
dim(stim_expression_matrix)
```

1.2 创建seurat对象与数据过滤（两个样本分别过滤）

```
#con样本创建seurat对象并过滤
con <- CreateSeuratObject(counts = con, project = "control", min.cells = 3, min.features = 200)
con[["percent.mt"]] <- PercentageFeatureSet(con, pattern = "^MT-")
con <- subset(con, subset = nFeature_RNA > 200 & nFeature_RNA < 1500 & percent.mt < 5)

#stim样本创建seurat对象并过滤
stim <- CreateSeuratObject(counts = stim, project = "stimulus", min.cells = 3, min.features = 200)
stim[["percent.mt"]] <- PercentageFeatureSet(stim, pattern = "^MT-")
VlnPlot(stim, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
stim <- subset(stim, subset = nFeature_RNA > 200 & nFeature_RNA < 1500 & percent.mt < 5)
```

1.3 数据归一化与高变基因筛选

```
#数据归一化
con <- NormalizeData(con, normalization.method = "LogNormalize",
scale.factor = 10000)
stim <- NormalizeData(stim, normalization.method = "LogNormalize",
scale.factor = 10000)
#高变基因筛选
con <- FindVariableFeatures(con, selection.method = "vst", nfeatures =
2000)
stim <- FindVariableFeatures(stim, selection.method = "vst", nfeatures =
2000)
```

1.4 数据合并

#筛选两组数据中共有的高变基因
features <- SelectIntegrationFeatures(object.list = list(con,stim))

#CCA识别两组数据连接锚
con_stim.anchors <- FindIntegrationAnchors(object.list = list(con,stim), anchor.features = features)

```
> con_stim.anchors <- FindIntegrationAnchors(object.list = list(con,stim), anchor.features = features)
Scaling features for provided objects
  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=01s
Finding all pairwise anchors
  |                                                  | 0 % ~calculating  Running CCA
Merging objects
Finding neighborhoods
Finding anchors
        Found 8143 anchors
Filtering anchors
        Retained 4145 anchors
  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=50s
```

#数据合并
con_stim.combined <- IntegrateData(anchorset = con_stim.anchors)

```
> con_stim.combined <- IntegrateData(anchorset = con_stim.anchors)
Merging dataset 1 into 2
Extracting anchors for merged samples
Finding integration vectors
Finding integration vector weights
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Integrating data
```

1.5 细胞分类

```
#数据合并之前已经做过归一化和高变基因筛选，这里直接进行标准化和聚类
seurat_object <- ScaleData(con_stim.combined, verbose = FALSE)
seurat_object <- RunPCA(seurat_object, npcs = 50, verbose = FALSE)
seurat_object <- RunTSNE(seurat_object, reduction = "pca", dims = 1:30)
seurat_object <- FindNeighbors(seurat_object, reduction = "pca", dims
= 1:30)
seurat_object <- FindClusters(seurat_object, resolution = 0.5)
```

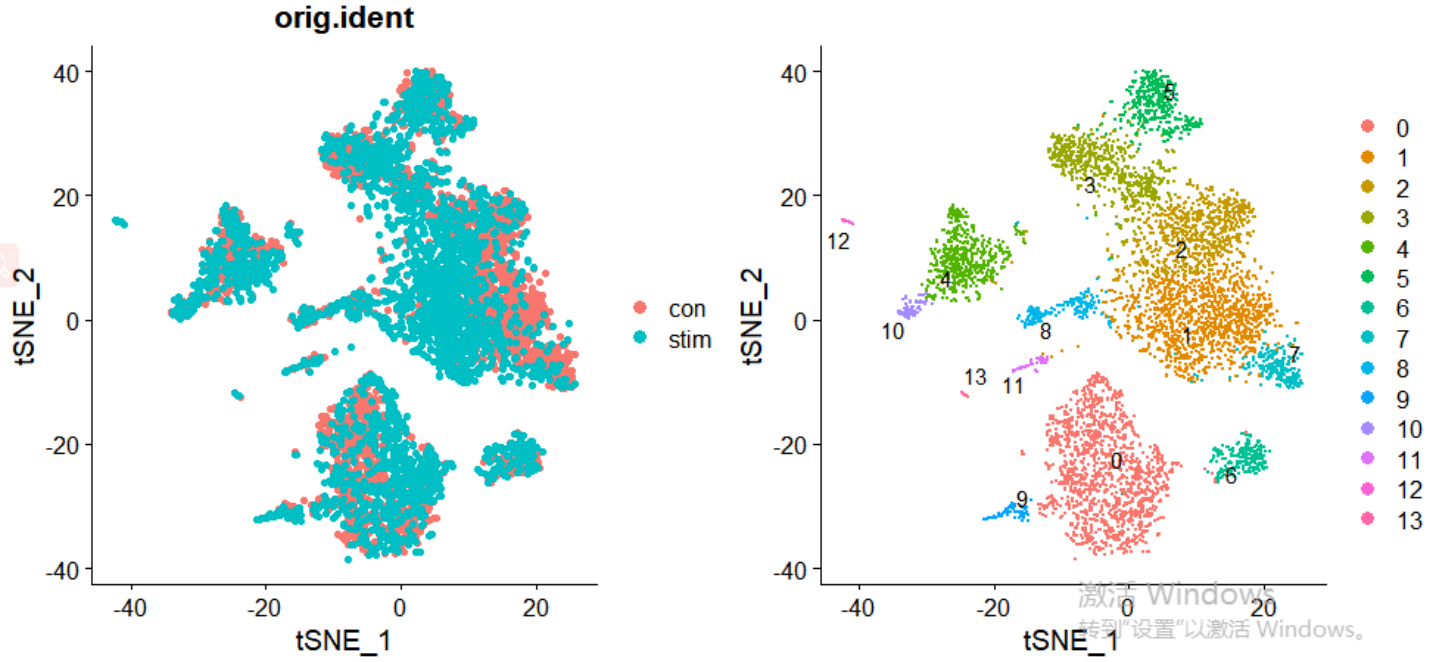
```
> seurat_object <- FindClusters(seurat_object, resolution = 0.5)
Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck

Number of nodes: 5942
Number of edges: 280415

Running Louvain algorithm...
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Maximum modularity in 10 random starts: 0.8975
Number of communities: 14
Elapsed time: 0 seconds
```

1.6 可视化

```
#降维可视化
p1 <- DimPlot(seurat_object, reduction = "tsne", group.by = "orig.ident", pt.size = 1.5)
p2 <- DimPlot(seurat_object, reduction = "tsne", label = TRUE, repel = TRUE)
p1 + p2
```



2. harmony矫正

2.1 数据导入（与标准流程相同）

```
#系统性红斑狼疮患者干扰素治疗前PBMC样本的读取
data_dir <- "E:/单细胞培训班/课件ppt/NO.1 10X单细胞概述及R语言入门/流程分析/数据及脚本/control/"  ##指定数据所在目录
list.files(data_dir)  ##列出文件名
con_expression_matrix <- Read10X(data.dir = data_dir) ##读取数据
dim(con_expression_matrix)   #查看维度，即基因数和细胞数
#系统性红斑狼疮患者干扰素治疗前PBMC样本的读取
data_dir <- "E:/单细胞培训班/课件ppt/NO.1 10X单细胞概述及R语言入门/流程分析/数据及脚本/control/"
list.files(data_dir)
stim_expression_matrix <- Read10X(data.dir = data_dir)
dim(stim_expression_matrix)
```

2.2 表达量矩阵合并

```
#矩阵合并
con_stim = cbind(con,stim)
```

2.3 创建seurat对象与数据过滤（与标准流程相同）

```
#创建seurat对象
seurat_object <- CreateSeuratObject(counts = con_stim)
#计算线粒体比例
seurat_object[["percent.mt"]] <- PercentageFeatureSet(seurat_object, pattern = "^MT-")
#细胞过滤
seurat_object <- subset(seurat_object, subset = nFeature_RNA > 200 & nFeature_RNA < 1500 & percent.mt < 5)
```

2.4 归一化、高边基因筛选、标准化

#归一化、高变基因筛选、标准化

#%>%就是把左边的值发送给右边的表达式，并作为右边表达式函数的第一个参数，就是管道函数。

```
seurat_object <- NormalizeData(seurat_object, normalization.method =
"LogNormalize", scale.factor = 10000) %>%
  FindVariableFeatures(selection.method = "vst", nfeatures = 2000) %>%
  ScaleData()
```

```
> seurat_object <- NormalizeData(seurat_object, normalization.method = "LogNormalize", scale.factor = 10000) %>%
+   FindVariableFeatures(selection.method = "vst", nfeatures = 2000) %>%
+   ScaleData()
Performing log-normalization
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Calculating gene variances
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Calculating feature variances of standardized and clipped values
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Centering and scaling data matrix
  |=================================================================================================| 100%
```

2.5 harmony整合

#PCA降维

```
seurat_object <- RunPCA(seurat_object, npcs = 50, verbose = FALSE)
```

#harmony矫正

```
seurat_object = seurat_object %>% RunHarmony("orig.ident",
plot_convergence = TRUE)#耗时1min
```

```
> seurat_object = seurat_object %>% RunHarmony("orig.ident", plot_convergence = TRUE)#耗时1min
Harmony 1/10
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Harmony 2/10
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Harmony 3/10
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Harmony 4/10
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Harmony 5/10
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
```

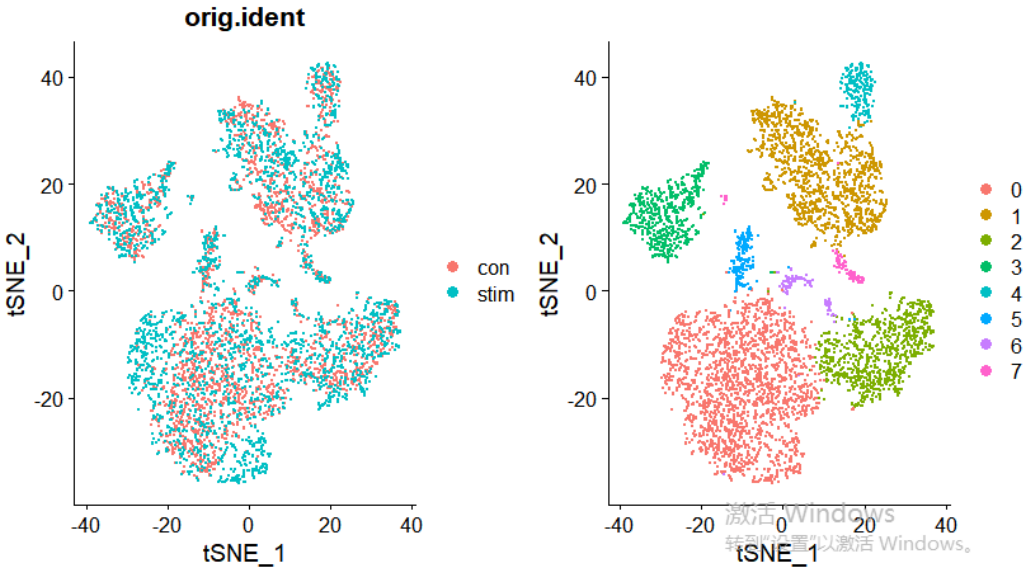
2.6 细胞聚类降维及可视化

#细胞聚类

```
seurat_object <- seurat_object %>%
  RunTSNE(reduction = "harmony", dims = 1:10) %>%
  FindNeighbors(reduction = "harmony", dims = 1:10) %>%
  FindClusters(resolution = 0.5) %>%
  identity()
```

#降维可视化

```
p1<- DimPlot(seurat_object, reduction = "tsne", group.by = "orig.ident",
pt.size = 0.5)
p2<- DimPlot(seurat_object, reduction = "tsne", pt.size = 0.5)
P1+P2
```



目录

细胞周期评估

1. 细胞周期评分
2. 周期基因回归

1. 细胞周期评分

seurat自带周期评分函数CellCycleScoring，以及人的S期和G2/M期marker基因集，可以完成细胞周期评分。

1.1 数据导入（与流程分析相同）

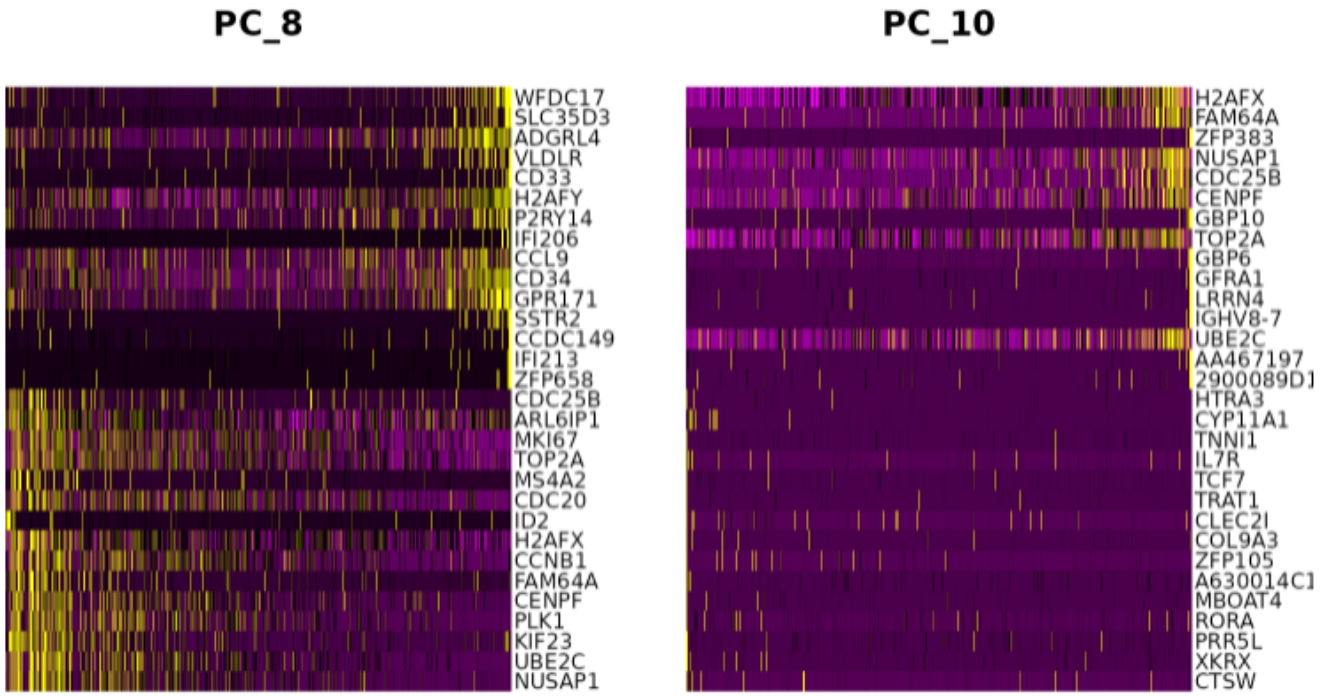
1.2 seurat对象创建与细胞过滤（与流程分析相同）

1.3 标准化（与流程分析相同）

1.4 细胞周期评分

#PCA线性降维，查看周期marker基因

```
seurat_object <- RunPCA(seurat_object, features =
VariableFeatures(seurat_object), ndims.print = 6:10, nfeatures.print =
10)
DimHeatmap(seurat_object, dims = c(8, 10))
```



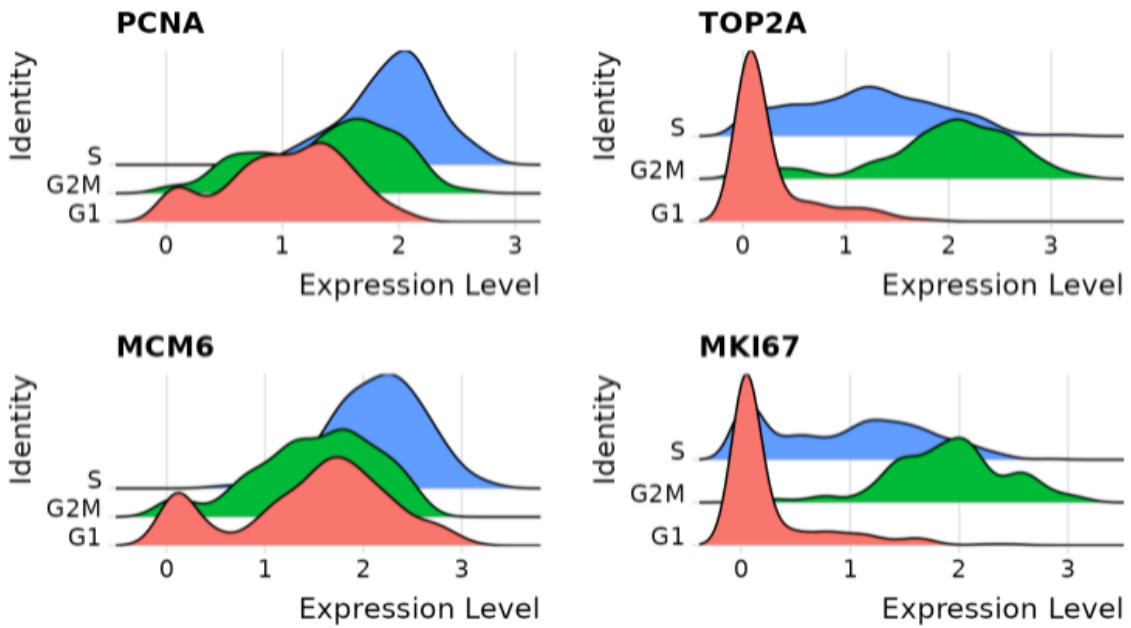
#细胞周期marker基因加载

```
s.genes <- cc.genes$s.genes
g2m.genes <- cc.genes$g2m.genes
#计算细胞周期分数
seurat_object <- CellCycleScoring(seurat_object, s.features = s.genes,
g2m.features = g2m.genes, set.ident = TRUE)
head(seurat_object[[]])
```

```
> head(seurat_object[[]])
                    orig.ident nCount_RNA nFeature_RNA percent.mt      S.Score   G2M.Score Phase
con_CGAAGTACTAGAAG-1        con       3278          921          0 -0.007607564  0.01223682   G2M
con_AAGTGGCTGGTGTT-1        con       1055          522          0  0.084781388  0.05984147     S
con_GGTGATACAGAGTA-1        con       1516          599          0 -0.055471721 -0.03748255    G1
con_TCTAGTTGAGGGTG-1        con        885          398          0 -0.029102943  0.01636687   G2M
con_GTCTGAGATTTGTC-1        con       6426         1324          0 -0.020160192 -0.04948001    G1
con_AACCCAGACATACG-1        con       1163          467          0  0.021177666 -0.04083677     S
                    old.ident
con_CGAAGTACTAGAAG-1       con
con_AAGTGGCTGGTGTT-1       con
con_GGTGATACAGAGTA-1       con
con_TCTAGTTGAGGGTG-1       con
con_GTCTGAGATTTGTC-1       con
con_AACCCAGACATACG-1       con
```

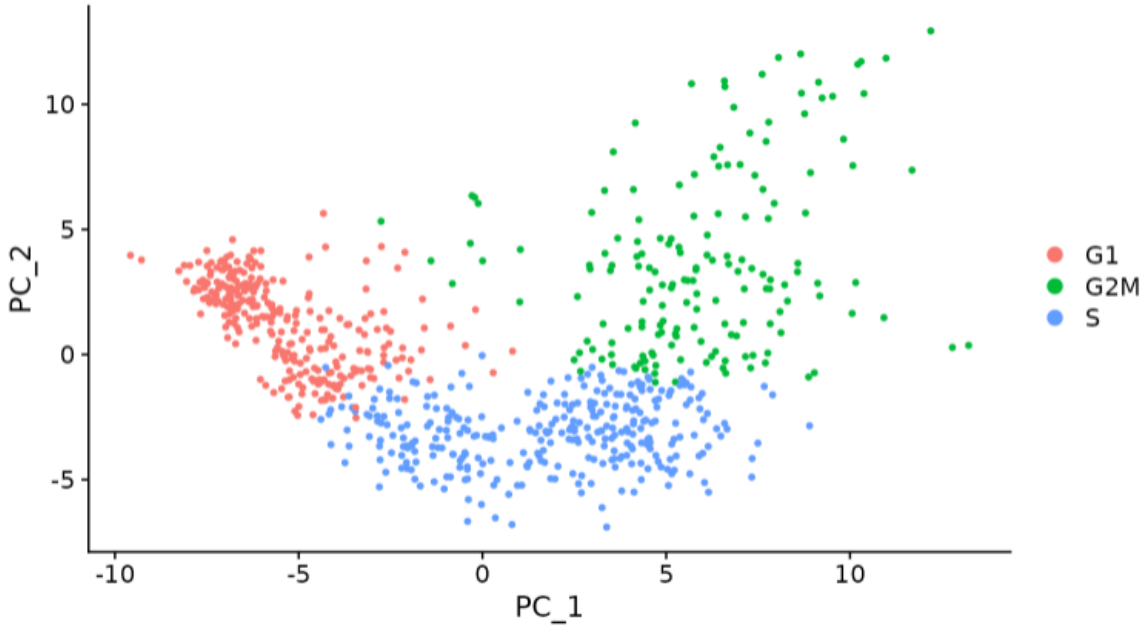
#周期marker基因可视化

RidgePlot(seurat_object, features = c("PCNA", "TOP2A", "MCM6", "MKI67"), ncol = 2)



#以周期marker基因进行PCA降维

```
seurat_object <- RunPCA(seurat_object, features = c(s.genes, g2m.genes))
DimPlot(seurat_object)
```



2. 周期基因回归

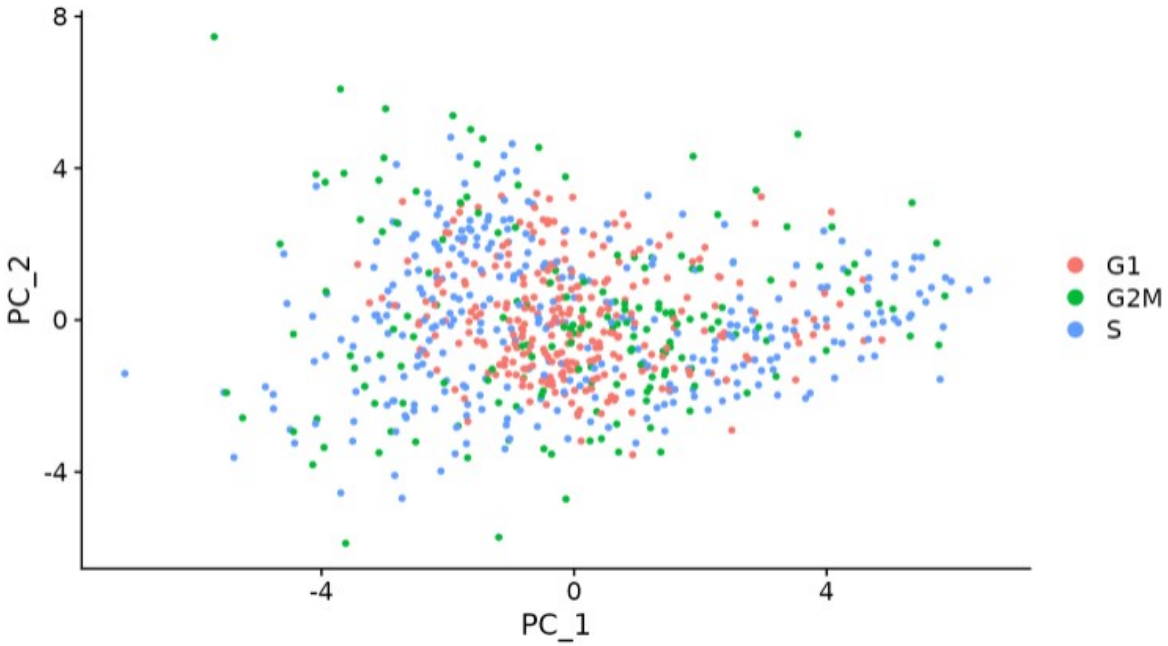
2.1 去除细胞周期对分群差异的影响

```
#接续上一步
#去除细胞周期对分群差异的影响
seurat_object <- ScaleData(seurat_object, vars.to.regress = c("S.Score", "G2M.Score"), features = rownames(seurat_object))
```

```
> #去除细胞周期对细胞异质性影响
> seurat_object <- ScaleData(seurat_object, vars.to.regress = c("S.Score", "G2M.Score"), features = rownames
(seurat_object))
Regressing out S.Score, G2M.Score
  |=========================================================================================| 100%
Centering and scaling data matrix
  |=========================================================================================| 100%
```

#以周期marker基因进行PCA降维

```
seurat_object <- RunPCA(seurat_object, features =
VariableFeatures(seurat_object), nfeatures.print = 10)
seurat_object <- RunPCA(seurat_object, features = c(s.genes, g2m.genes))
DimPlot(seurat_object)
```



2.2 保持非周期细胞和周期细胞的组分差异

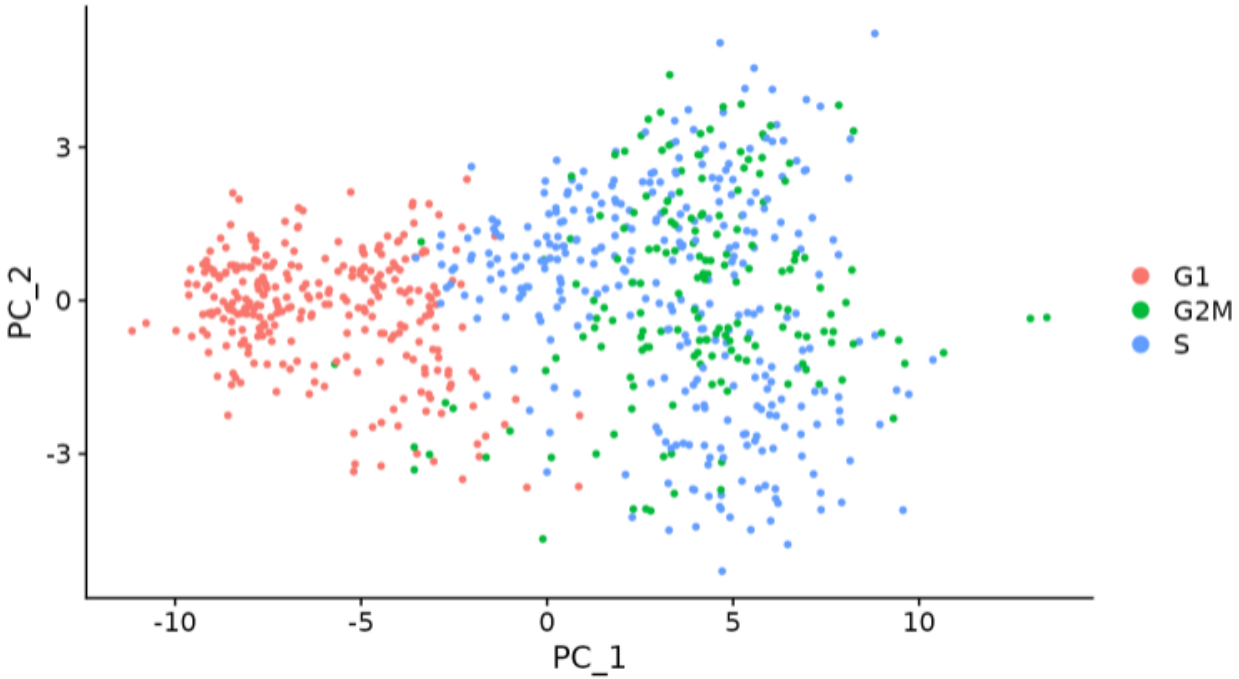
#去除G2M和S阶段分数之间的差异
seurat_object$CC.Difference <- seurat_object$S.Score - seurat_object$G2M.Score
head(seurat_object[[]])

```
> head(seurat_object[[]])
                    orig.ident nCount_RNA nFeature_RNA percent.mt      S.Score   G2M.Score Phase
con_CGAAGTACTAGAAG-1        con       3278          921          0 -0.007607564  0.01223682   G2M
con_AAGTGGCTGGTGTT-1        con       1055          522          0  0.084781388  0.05984147     S
con_GGTGATACAGAGTA-1        con       1516          599          0 -0.055471721 -0.03748255    G1
con_TCTAGTTGAGGGTG-1        con        885          398          0 -0.029102943  0.01636687   G2M
con_GTCTGAGATTTGTC-1        con       6426         1324          0 -0.020160192 -0.04948001    G1
con_AACCCAGACATACG-1        con       1163          467          0  0.021177666 -0.04083677     S
                    old.ident CC.Difference
con_CGAAGTACTAGAAG-1       con   -0.01984438
con_AAGTGGCTGGTGTT-1       con    0.02493992
con_GGTGATACAGAGTA-1       con   -0.01798917
con_TCTAGTTGAGGGTG-1       con   -0.04546981
con_GTCTGAGATTTGTC-1       con    0.02931982
con_AACCCAGACATACG-1       con    0.06201444
```

seurat_object <- ScaleData(seurat_object, vars.to.regress = "CC.Difference", features = rownames(seurat_object))

#以周期marker基因进行PCA降维

```
seurat_object <- RunPCA(seurat_object, features =
VariableFeatures(seurat_object), nfeatures.print = 10)
seurat_object <- RunPCA(seurat_object, features = c(s.genes, g2m.genes))
DimPlot(seurat_object)
```



基迪奥生物
GENE DENOVO

Thanks