sravg_example

Qingnan Liang

1/10/2022

Introduction

Here is a demonstration of how to use the package SRAVG to create a Seurat object with cells grouped and averaged.

Load libraries

```
library(Seurat)
library(SRAVG)
library(dplyr)
library(Matrix)
library(SeuratData)
```

Use the pbmc3k dataset from the SeuratData

```
data("pbmc3k")
pbmc3k
```

```
## An object of class Seurat
## 13714 features across 2700 samples within 1 assay
## Active assay: RNA (13714 features, 0 variable features)
```

The pbmc3k is a Seurat object while it is not processed. To run the SRAVG, we require the preprocessing of the object. Two things are needed: first, a dimension reduction (must be one of names(object@reductions)), for grouping nearest neighbors; second, a predefined classification of cells (must be one of colnames(object@meta.data)), because we don't want to group and average cells from different cluster/type/sample/donor etc.

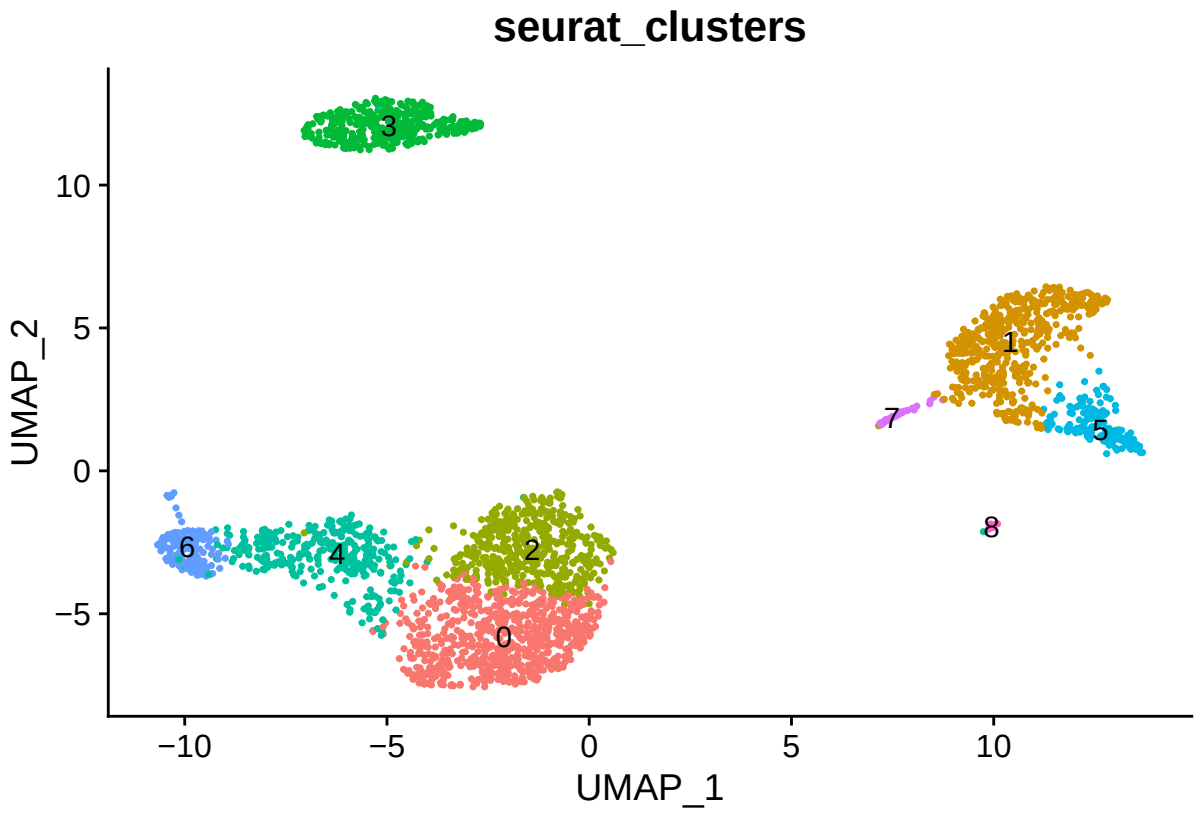
Regular Seurat workflow:

```
pbmc3k <- pbmc3k %>%
    NormalizeData() %>%
    FindVariableFeatures() %>%
    ScaleData() %>%
    RunPCA(verbose = FALSE) %>%
    FindNeighbors(dims = 1:10) %>%
    FindClusters(resolution = 0.5) %>%
    RunUMAP(dims = 1:10, verbose = FALSE)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 2700
## Number of edges: 97892
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8719
## Number of communities: 9
## Elapsed time: 0 seconds
```

Visualize the clustering:

```
DimPlot(pbmc3k, group.by = "seurat_clusters", label = TRUE) +
    NoLegend()
```



Run SRAVG

Here we use 'pca' as the dimension reduction to evaluate the distances between cells, and top 10 PCs are used (dr_dims). We try to form meta-cells by averaging 10 cells to 1. The group will be within each individual 'seurat_clusters'. We also average two other columns in the meta.data and transfer to the output seurat, which are 'nCount_RNA', and 'nFeature_RNA'. At this time we only support numerical columns for 'extra_meta'.

```
start_time <- Sys.time()

pbmc_avg <- sravg(object = pbmc3k, dr_key = "pca", dr_dims = 10,
    group_size = 10, group_within = "seurat_clusters", extra_meta = c("nCount_RNA",
        "nFeature_RNA"))
end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 1.352331 mins
```

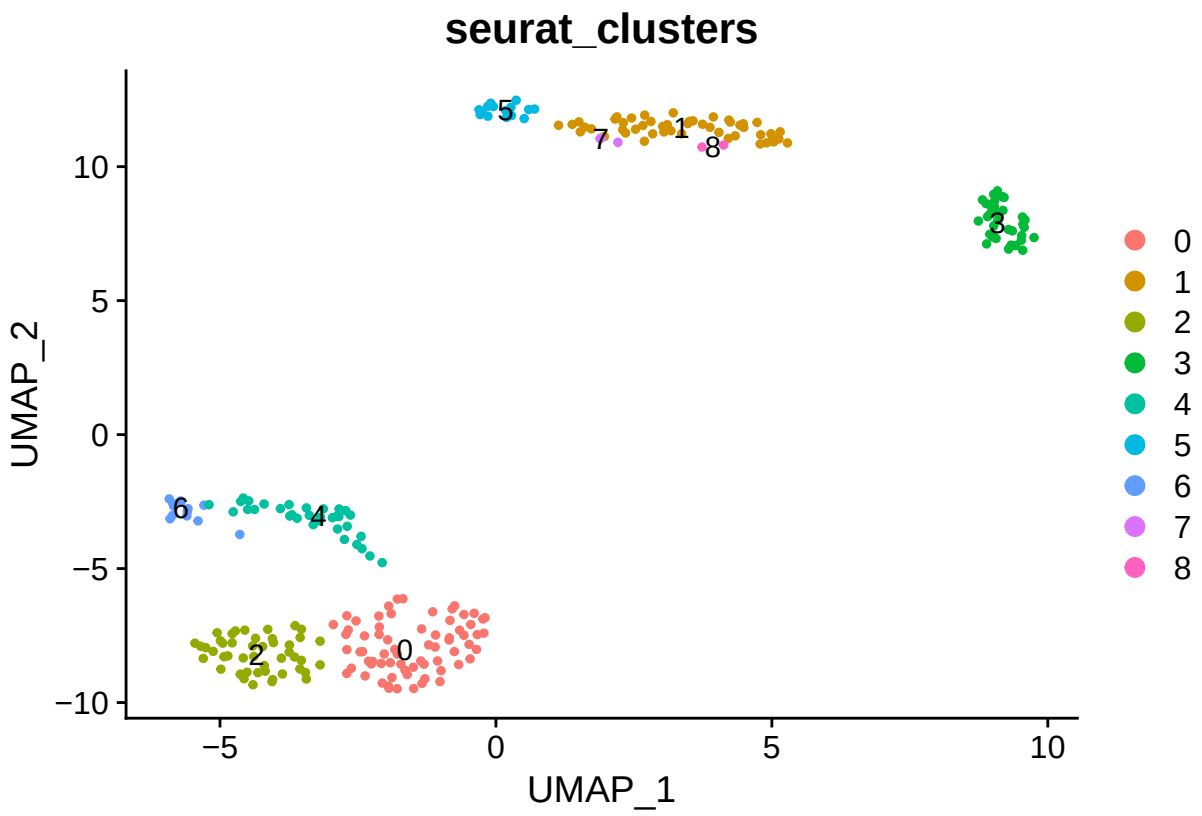
The output is a Seurat object

```
pbmc_avg
```

```
## An object of class Seurat
## 13714 features across 265 samples within 1 assay
## Active assay: RNA (13714 features, 0 variable features)
##  1 dimensional reduction calculated: pca
```

We can still run UMAP on the pbmc_avg object. The 'pca' in this averaged object is calculated with averaging the original 'pca' coordinates (for each meta-cell). Basically, the dimension reduction key used in the original object will be succeeded by the averaged object (if dr_key = 'umap', then the pbmc_avg will have a 'umap' in 'reductions'). Also, the 'group_within' column will be retained in the meta.data of the averaged object.

```
pbmc_avg <- RunUMAP(pbmc_avg, dims = 1:10, verbose = FALSE)
DimPlot(pbmc_avg, group.by = "seurat_clusters", label = T)
```



Compute the sparsity of matrix (proportion of zeros) before and after averaging

```
sparsity <- function(matrix) {
    sparsity <- sum(matrix == 0)/(dim(matrix)[1] * dim(matrix)[2])
    return(sparsity)
}

print(sparsity(pbmc3k@assays$RNA@counts))
```

```
## [1] 0.9383443
```

```
print(sparsity(pbmc_avg@assays$RNA@counts))
```

```
## [1] 0.727272
```