Reconstruction

2023-03-10

Create a Seurat object for the reconstructed data

```
library(Seurat)
```

```
## Attaching SeuratObject

## Attaching sp
```

```
library(SeuratObject)
library(SeuratData)
```

```
## -- Installed datasets ------------------------------------- SeuratData v0.2.2 --

## v bmcite         0.3.0                  v panc8          3.0.2
## v cbmc           3.1.4                  v pancreasref    1.0.0
## v hcabm40k       3.0.0                  v pbmcref        1.0.0
## v ifnb           3.1.0                  v pbmcsca        3.0.0
## v lungref        2.0.0                  v thp1.eccite    3.1.5
## v mousecortexref 1.0.0

## -------------------------------------- Key -------------------------------------

## v Dataset loaded successfully
## > Dataset built with a newer version of Seurat than installed
## (?) Unknown version of Seurat installed
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(useful)
```

```
## Loading required package: ggplot2
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
LoadData('ifnb')
```

```
## An object of class Seurat
## 14053 features across 13999 samples within 1 assay
## Active assay: RNA (14053 features, 0 variable features)
```

```
ifnb$cell_type <- plyr::mapvalues(ifnb$seurat_annotations,
                                  from = c('CD8 T', 'CD4 Memory T', 'T activated', 'CD4 Naive T', 'B',
                                  to = c('T', 'T', 'T', 'T', 'B', 'B', 'Mono', 'Mono', 'DC'))
ifnb$cell_subtype <- ifnb$seurat_annotations


data <- fread('ifnb_celltype_denoised_expression.txt',data.table = FALSE)
rownames(data) <- data[,1]
data <- data[,-1]
data <- t(data)

counts <- expm1(data)


recon <- CreateSeuratObject(counts, meta.data = ifnb@meta.data[colnames(data), ])
recon <- SetAssayData(recon, slot = 'data', data)
recon <- FindVariableFeatures(recon, nfeatures = 5000)
recon <- ScaleData(recon)
```

```
## Centering and scaling data matrix
```

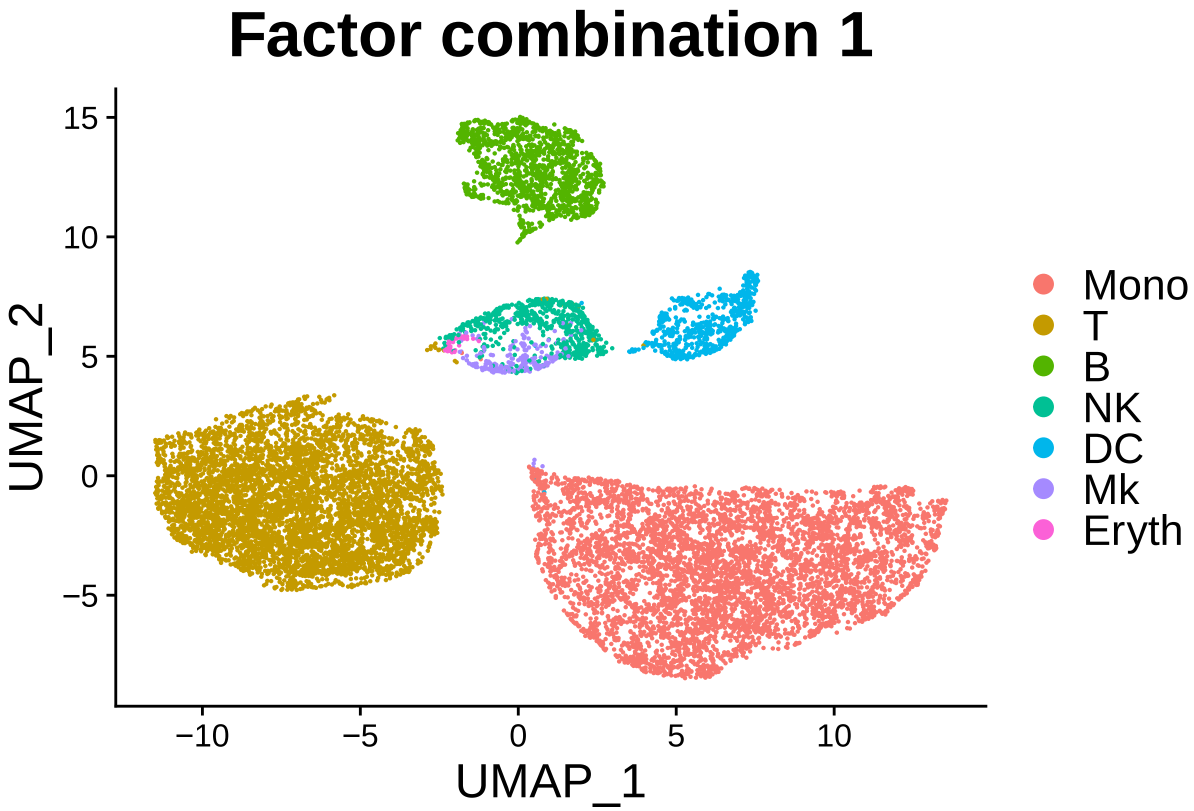
```
recon <- RunPCA(recon, verbose=FALSE)
recon <- RunUMAP(recon, dims=1:30)
```

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R-
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session
```

```
## 21:29:05 UMAP embedding parameters a = 0.9922 b = 1.112
```

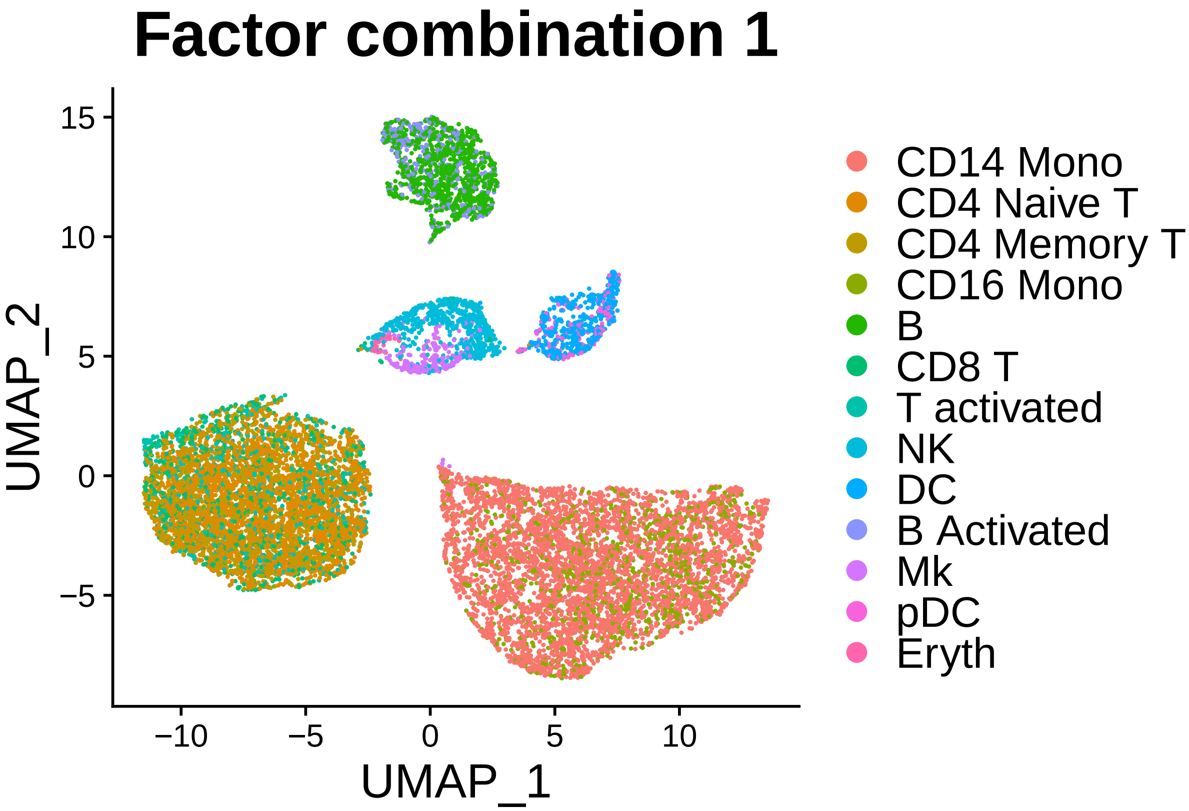
```
## 21:29:05 Read 13999 rows and found 30 numeric columns
## 21:29:05 Using Annoy for neighbor search, n_neighbors = 30

## 21:29:05 Building Annoy index with metric = cosine, n_trees = 50

## 0%   10   20   30   40   50   60   70   80   90   100%

## [----|----|----|----|----|----|----|----|----|----|

## **************************************************|
## 21:29:06 Writing NN index file to temp file /tmp/RtmpVSnhj6/file255e7334adc6b
## 21:29:06 Searching Annoy index using 1 thread, search_k = 3000
## 21:29:09 Annoy recall = 100%
## 21:29:10 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 21:29:10 Initializing from normalized Laplacian + noise (using irlba)
## 21:29:11 Commencing optimization for 200 epochs, with 495230 positive edges
## 21:29:15 Optimization finished
```

Plot cell type



Plot cell subtype

```
DimPlot(recon, reduction = 'umap', group.by = 'cell_subtype') +
  ggtitle('Factor combination 1') +
  theme(plot.title = element_text(size=24),
        axis.title = element_text(size=18),
        legend.text = element_text(size=16))
```


Factor combination 1
UMAP_2
UMAP_1
CD14 Mono
CD4 Naive T
CD4 Memory T
CD16 Mono
B
CD8 T
T activated
NK
DC
B Activated
Mk
pDC
Eryth