

# SCMarker.NK.GBM

The GBM data set is downloaded from GEO (GSE84465), including 3587 cells from four primary GBM patients. Darmanis S, Sloan SA, Croote D, Mignardi M et al. Single-Cell RNA-Seq Analysis of Infiltrating Neoplastic Cells at the Migrating Front of Human Glioblastoma. Cell Rep 2017 Oct 31;21(5):1399-1410. PMID: 29091775

```
GBM=read.csv('GSE84465_GBM_All_data.csv',sep=" ")
```

```
dim(GBM)
```

```
## [1] 23465 3589
```

Run Seurat to identify cluster enriched NK cell

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
##  
## 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(Seurat)  
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 3.5.2
```

```
##  
## *****
```

```
## Note: As of version 1.0.0, cowplot does not change the
```

```
## default ggplot2 theme anymore. To recover the previous
```

```
## behavior, execute:  
## theme_set(theme_cowplot())
```

```
## *****
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
GBMobj <- CreateSeuratObject(counts = GBM, project = "GBM")
```

```
## Warning: Feature names cannot have underscores ('_'), replacing with dashes  
## ('-')
```

```
GBMobj <- NormalizeData(object = GBMobj, verbose = FALSE)  
GBMobj <- FindVariableFeatures(object = GBMobj, selection.method = "vst", nfeatures =  
2000)  
GBMobj <- ScaleData(GBMobj)
```

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

```
GBMobj <- RunPCA(GBMobj)
```

```

## PC_ 1
## Positive: GPM6B, PTPRZ1, SCD5, GPRC5B, TUBA1A, TSPAN7, GPM6A, CLU, BCAN, NCAM1
##          ALDOC, SLC22A17, CPE, DNER, TSC22D1, S100B, SERPINE2, SPARCL1, NLGN3, FXYD6
##          Clorf61, BAALC, CNP, PLEKHB1, OLIG1, GPR37L1, PTN, SPOCK1, APOD, SLC1A2
## Negative: CD74, HLA-DRA, HLA-DRB1, C1QB, RGS1, HLA-DPA1, CD83, OLR1, HLA-DRB5, CC
L3
##          GPR183, A2M, CSF1R, HLA-DPB1, IFI30, FTL, C3, CTSB, VSIG4, FPR1
##          CCL4, SPP1, CD14, HLA-DQB1, PLEK, PLAUR, HLA-DMB, ACSL1, APOE, RHBDF2
## PC_ 2
## Positive: TF, DBNDD2, MOG, TUBB4A, ENPP2, ERMN, PPP1R14A, PLLP, KLK6, SPOCK3
##          CNDP1, CNTN2, EDIL3, PLP1, MAL, C11orf9, APOD, OPALIN, CARNS1, MBP
##          MAG, TMEM144, MOBP, UGT8, CNP, HAPLN2, FA2H, SPOCK1, SH3GL3, SEPT4
## Negative: IGFBP7, IGFBP2, TNC, CYR61, CSRP2, C1R, WWTR1, CLU, SERPINA3, AQP4
##          CHI3L1, GFAP, ID3, MT2A, ID4, C8orf4, ACTN1, GADD45A, MLC1, C1S
##          HSPB1, CA12, GAP43, EGFR, COL1A2, EMP1, CAV1, SPARC, CXCL14, GEM
## PC_ 3
## Positive: TNF, MEGF11, MEG3, ATCAY, PDGFRA, SMOC1, NKAIN4, GPR17, CA10, CSPG4
##          SULF2, PDZD2, EPN2, SEZ6L, OLIG1, EHD3, PCDH15, LHFPL3, DLL3, NLGN3
##          GFRA1, AFAP1L2, NXPH1, BCAN, VCAN, COL9A1, BAMBI, FLJ43390, HIP1R, FXYD6
## Negative: DBNDD2, ERMN, KLK6, CNDP1, TF, PLP1, CRYAB, CNTN2, TMEM144, MOG
##          MOBP, ENPP2, MAG, OPALIN, CARNS1, PPP1R14A, SEPT4, MAL, GPR37, CAPN3
##          EFHD1, ANLN, C11orf9, MBP, LPAR1, SEPP1, ABCA8, FOLH1, SH3GL3, CLDND1
## PC_ 4
## Positive: TIMP1, ANXA2, CAV1, GAP43, C8orf4, MAP1B, IGFBP2, TAGLN, FN1, AKAP12
##          TPM2, GADD45A, CALD1, WWTR1, SPOCD1, CA9, MGP, THY1, ACTN1, COL1A2
##          TM4SF1, IGFBP3, GRB10, RND3, CA12, CYR61, SCG2, MLLT11, IGFBP5, EMP1
## Negative: AGXT2L1, GJB6, GPR98, WIF1, PPP1R1B, NTSR2, SLC39A12, GABRA2, ACSBG1, S
LC14A1
##          CPNE6, PRODH, CLDN10, HHATL, GJA1, SPON1, ATP1A2, SLC1A3, SLC1A4, SLC1A2
##          AQP4, GABRB1, RYR3, FAM107A, AGT, SLC6A11, NDRG2, TPD52L1, GPC5, HSD17B6
## PC_ 5
## Positive: IFITM1, DCN, COL3A1, NID1, GGT5, COL1A1, FRZB, MYO1B, CD248, COL4A1
##          PDGFRB, APOLD1, COL1A2, CFH, ISLR, COL4A2, COL6A2, IGFBP4, UACA, IGFBP7
##          PCOLCE, FBLN2, COLEC12, LUM, FBLN1, OLFML3, MGLL, NDUFA4L2, CYP1B1, ITIH5
## Negative: KCNE1L, DBI, Clorf61, HES6, IL13RA2, HIST1H4C, CCND2, CREB5, EGFR, NNAT
##          TUBA1A, CD9, STMN1, NREP, CENPF, MEST, CDK4, MASP1, PMP2, UBE2T
##          CD24, AQP4, NMB, FABP7, MLLT11, BIRC5, PBK, RARRES2, MKI67, UBE2C

```

```
GBMobj <- FindNeighbors(GBMobj, dims = 1:25)
```

```
## Computing nearest neighbor graph
```

```
##Computing SNN
```

```
GBMobj <- FindClusters(GBMobj, resolution = 1.7)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 3589
## Number of edges: 114422
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8378
## Number of communities: 28
## Elapsed time: 0 seconds
```

```
GBMobj <- RunUMAP(GBMobj, dims = 1:25)
```

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

```
## 15:40:07 UMAP embedding parameters a = 0.9922 b = 1.112
```

```
## 15:40:07 Read 3589 rows and found 25 numeric columns
```

```
## 15:40:07 Using Annoy for neighbor search, n_neighbors = 30
```

```
## 15:40:07 Building Annoy index with metric = cosine, n_trees = 50
```

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

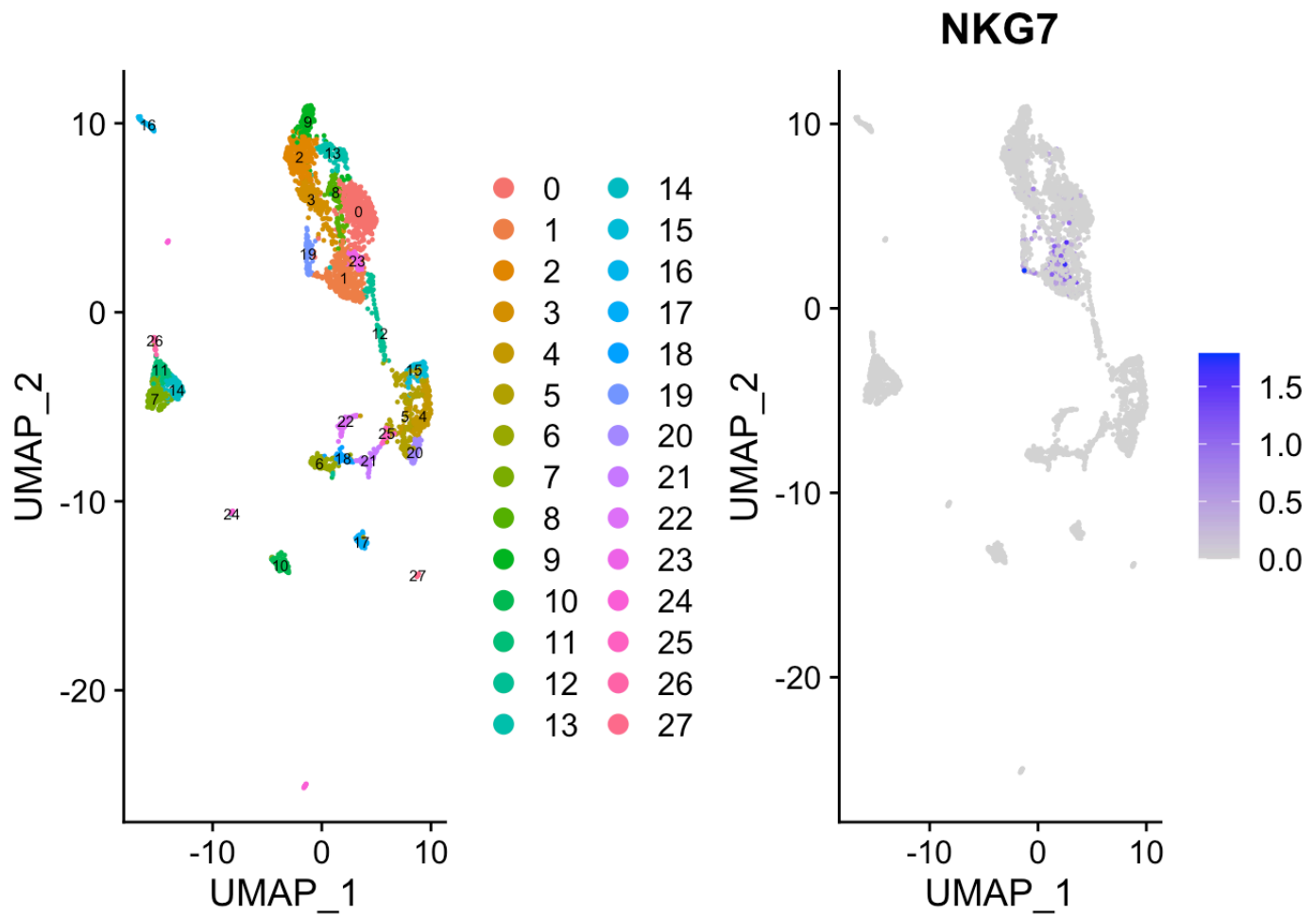
```
## [----|----|----|----|----|----|----|----|----|----|
```

```
## *****|
## 15:40:07 Writing NN index file to temp file /var/folders/gm/1x2dvvj50wvd8ncvyqm5z6
ssyh7m9q/T//RtmpXuw6Kd/filee28558c0d09d
## 15:40:07 Searching Annoy index using 1 thread, search_k = 3000
## 15:40:08 Annoy recall = 100%
## 15:40:08 Commencing smooth kNN distance calibration using 1 thread
## 15:40:08 Initializing from normalized Laplacian + noise
## 15:40:09 Commencing optimization for 500 epochs, with 140114 positive edges
## 15:40:13 Optimization finished
```

```
p1 <- DimPlot(GBMobj,label = TRUE,pt.size = 0.2,label.size=2)
```

```
## Warning: Using `as.character()` on a quosure is deprecated as of rlang 0.3.0.
## Please use `as_label()` or `as_name()` instead.
## This warning is displayed once per session.
```

```
p2 <- FeaturePlot(GBMobj, features = c("NKG7"),pt.size=0.2)
p1+p2
```

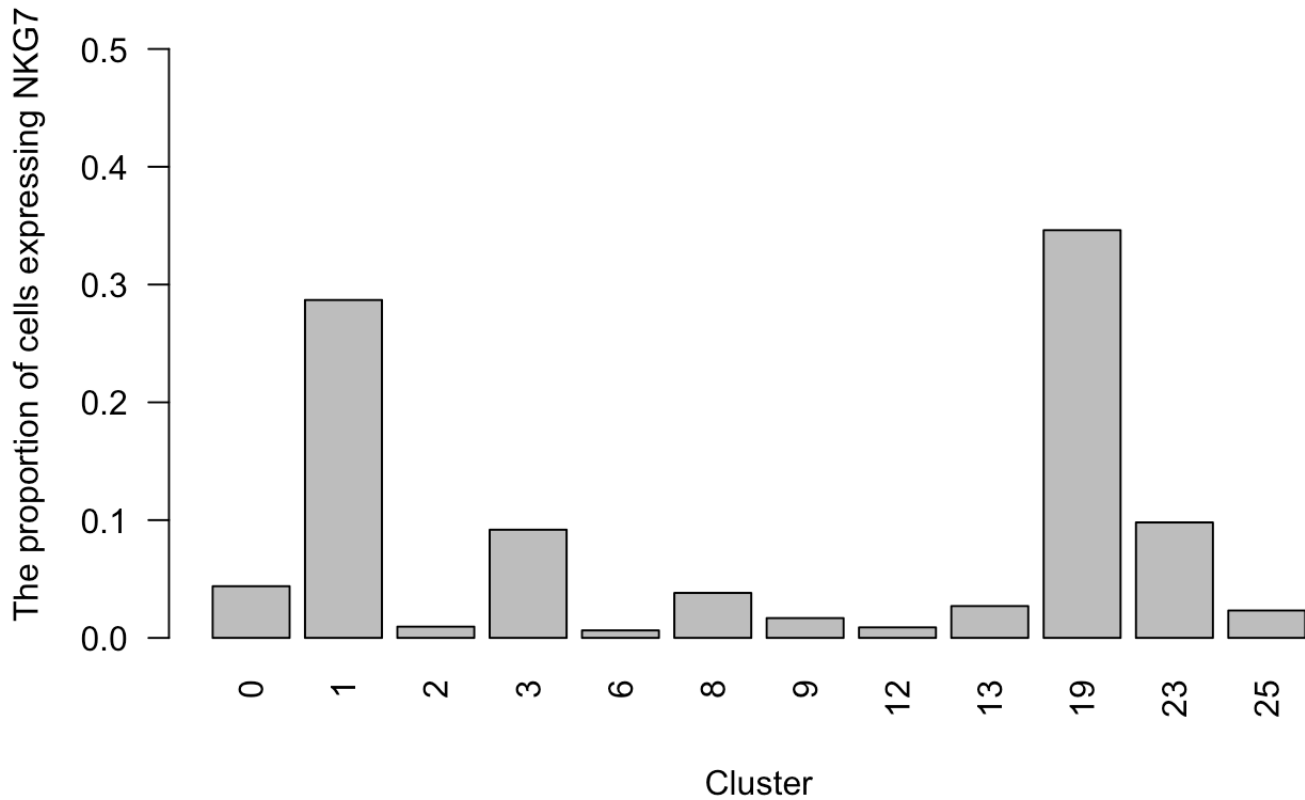


The proportion of cells expressing NK cell marker NKG7 in each cluster

```
subGBM=subset(GBMobj,NKG7>0)
index=match(names(table(Idsents(subGBM))),names(table(Idsents(GBMobj))))
seurat=table(Idsents(subGBM))/table(Idsents(GBMobj))[index]
```

```
barplot(seurat,ylim = c(0,0.55),las=2,main='Seurat',xlab="Cluster",ylab="The proportion of cells expressing NKG7")
```

## Seurat



### Integration of Seurat and SCMarker

```
library(SCMarker)
GBM1=log(GBM+1)
scMarker.res=ModalFilter(data=GBM1, geneK=10, cellK=10, width=2) # default width = 1 for UMI data, width =2 for TPM data.
scMarker.res=GeneFilter(obj=scMarker.res)
scMarker.res=getMarker(obj=scMarker.res, k=300, n=30)
```

### Extract markers identified by SCMarker

```
scmarker=scMarker.res$marker
length(scmarker)
```

```
## [1] 840
```

```
head(scmarker)
```

```
## [1] "ABL2" "ACSL1" "ADORA3" "AIF1" "AKR1B1" "ALOX5"
```

```
all.genes <- rownames(GBMobj)
GBMscmarker <- ScaleData(GBMobj, features = all.genes)
```

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

```
###Run PCA based on union of markers from SCMarker and Seurat
GBMscmarker <- RunPCA(GBMscmarker, features = union(scmaker, VariableFeatures(GBMobj))
))
```



```

## PC_ 1
## Positive: CD74, LAPTM5, SRGN, HLA-DRA, HLA-DRB1, C1QB, TYROBP, C1QA, FCER1G, RNAS
ET2
## SLC11A1, RGS1, HLA-DPA1, AIF1, CD83, OLR1, HLA-DRB5, CCL3, GPR183, HLA-DPB1
## CSF1R, A2M, IFI30, GPX1, ARHGDIB, FTL, HLA-DMA, C3, VSIG4, HCLS1
## Negative: PTPRZ1, GPM6B, TUBA1A, CLU, CNN3, GPM6A, NGFRAP1, SCD5, MARCKSL1, TSPAN
7
## Clorf61, BCAN, GPRC5B, SYT11, PTN, NRCAM, CPE, FEZ1, ITGB8, LDHB
## ALDOC, BAALC, FXYD6, MAP1B, SRI, PEBP1, TRIM9, PON2, SLC22A17, TSPAN3
## PC_ 2
## Positive: VIM, ANXA2, IGFBP2, LDHA, CHCHD2, CD44, IGFBP7, TPI1, CSRP2, TAGLN2
## LGALS3, ACTN1, WWTR1, TNC, ANXA1, S100A10, CALU, MT2A, GAPDH, C1R
## CYR61, CA12, CAV1, SOD2, TUBA1C, TIMP1, TUBA1B, CHI3L1, SERPINA3, CCT6A
## Negative: APOD, PLLP, SOX10, SPOCK1, TNF, CDR1, SNX22, MEGF11, SMOC1, EDIL3
## GPR17, ATCAY, OMG, PDGFRA, MEG3, NKAIN4, OLIG1, TUBB4A, EPN2, HIP1R
## CA10, BCAS1, PCDH15, BAMBI, EHD3, FAIM2, CNP, PDZD2, CSPG4, SOX8
## PC_ 3
## Positive: HTRA1, COTL1, CTSD, BIN1, CXCL16, GRN, CTSB, FPR1, CORO1A, GPX1
## TSPO, SERPINA1, YWHAH, CAPG, SPP1, UCP2, C1QB, TREM2, CREG1, FAM49B
## LAPTM5, CD14, CNP, CNDP2, CD81, RHOA, ITGB2, ARF1, C20orf3, SGK1
## Negative: ERCC-00131, GJA1, AQP4, ERCC-00071, ID4, ERCC-00044, RYR3, ERCC-00079,
HSPB1, GFAP
## ERCC-00112, ERCC-00116, GPR98, ERCC-00025, ERCC-00144, Clorf61, ID3, S1PR1, AG
T, AGXT2L1
## CYR61, EMP1, ERCC-00062, C8orf4, DTNA, CHI3L1, ABCA13, RARRES2, SULF1, CLU
## PC_ 4
## Positive: DBNDD2, ERMN, TF, KLK6, CNDP1, MOG, ENPP2, PLP1, CNTN2, PPP1R14A
## TMEM144, MOBP, OPALIN, CARNS1, MAG, MAL, SEPT4, CAPN3, C11orf9, MBP
## SH3GL3, ANLN, EFHD1, QDPR, FOLH1, APLP1, RAPGEF5, CRYAB, TPPP, ST18
## Negative: MEG3, TNF, ATCAY, MEGF11, PDGFRA, CSPG4, BCAN, PDZD2, LHFPL3, SULF2
## NKAIN4, SMOC1, CA10, SEZ6L, FXYD6, PTPRZ1, KCNQ2, CST3, GPR17, EHD3
## VCAN, PCDH15, C1QL1, TSPAN7, XYLT1, EPN2, CACNG4, SEMA5A, OLIG1, GPR56
## PC_ 5
## Positive: AGXT2L1, GJB6, GPR98, SLC1A3, WIF1, PPP1R1B, GJA1, NTSR2, NDRG2, SLC39A
12
## FAM107A, ACSBG1, GABRA2, MLC1, SLC14A1, PRODH, CLDN10, CPNE6, CPE, RYR3
## HHATL, DKK3, SPON1, GABRB1, TTYH1, TPD52L1, SPARCL1, AQP4, SLC6A11, SLC1A2
## Negative: STMN1, HIST1H4C, CREB5, NPM1, TM4SF1, ATP5O, S100A10, RAN, HINT1, GNB2L
1
## MLLT11, LDHA, LIMA1, IL13RA2, MDK, TGFBI, PSMA7, ANXA2, PDLIM1, MKI67
## VCAN, CENPF, TIMP1, FXYD6, CACYBP, SH3BGRL3, KCNE1L, EEF1G, BTF3, GSTO1

```

```
GBMscmarker <- FindNeighbors(GBMscmarker, dims = 1:25)
```

```
## Computing nearest neighbor graph
```

```
## Computing SNN
```

```
GBMscmarker <- FindClusters(GBMscmarker, resolution = 1.7)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 3589
## Number of edges: 112377
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8473
## Number of communities: 28
## Elapsed time: 0 seconds
```

```
GBMscmarker <- RunUMAP(GBMscmarker, dims = 1:25)
```

```
## 15:48:41 UMAP embedding parameters a = 0.9922 b = 1.112
```

```
## 15:48:41 Read 3589 rows and found 25 numeric columns
```

```
## 15:48:41 Using Annoy for neighbor search, n_neighbors = 30
```

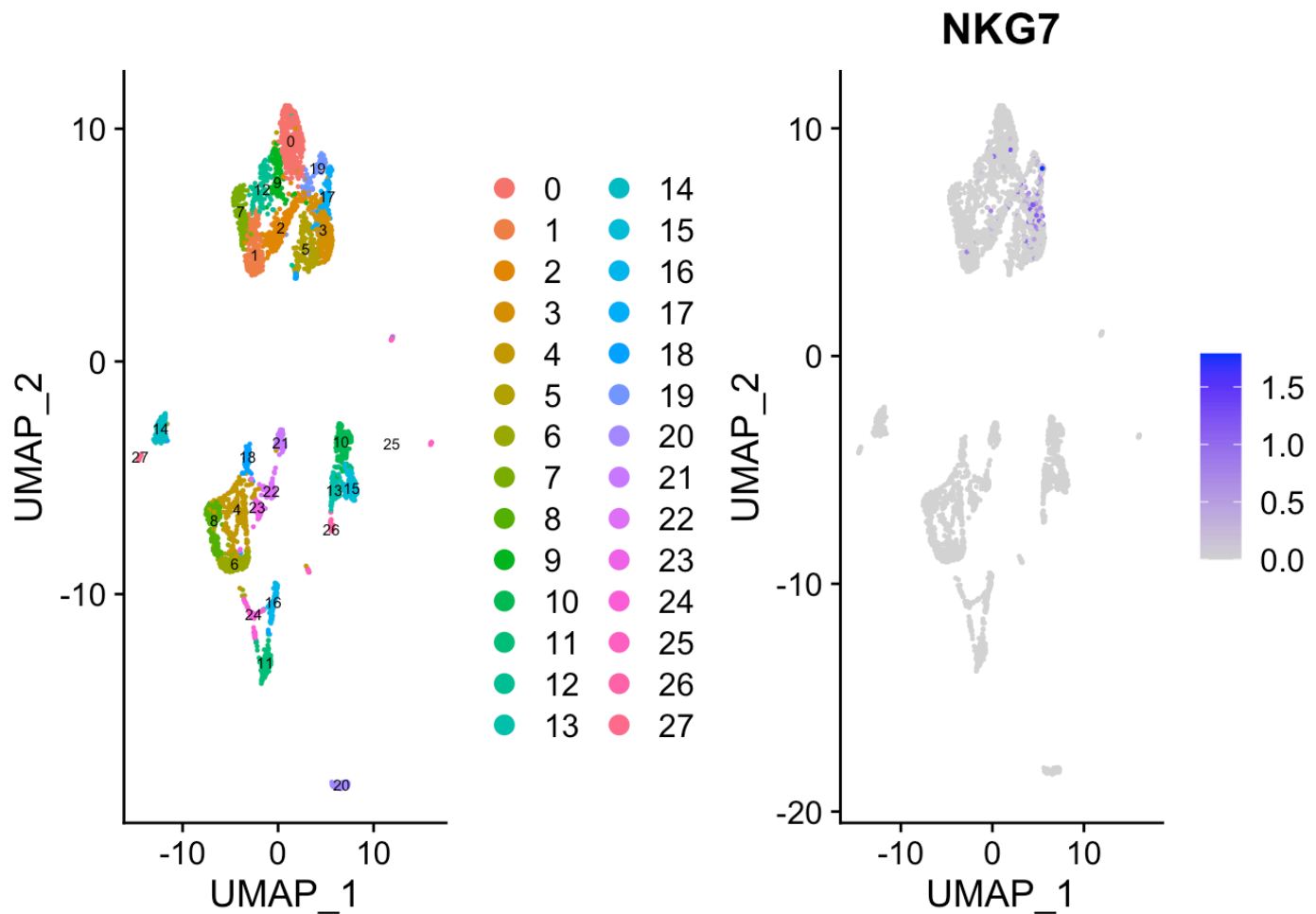
```
## 15:48:41 Building Annoy index with metric = cosine, n_trees = 50
```

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

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

```
## *****|
## 15:48:41 Writing NN index file to temp file /var/folders/gm/1x2dvvj50wvd8ncvyqm5z6
ssyh7m9q/T//RtmpXuw6Kd/filee2851ba4fed3
## 15:48:41 Searching Annoy index using 1 thread, search_k = 3000
## 15:48:42 Annoy recall = 100%
## 15:48:42 Commencing smooth kNN distance calibration using 1 thread
## 15:48:43 Initializing from normalized Laplacian + noise
## 15:48:43 Commencing optimization for 500 epochs, with 138724 positive edges
## 15:48:47 Optimization finished
```

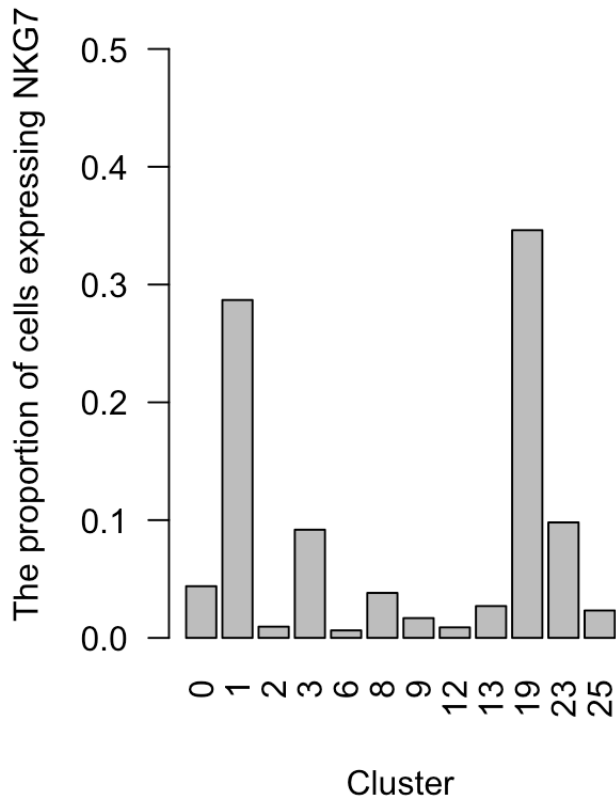
```
p1 <- DimPlot(GBMscmarker,label = TRUE,pt.size = 0.2,label.size=2)
p2 <- FeaturePlot(GBMscmarker, features = c("NKG7"),pt.size=0.2)
p1+p2
```



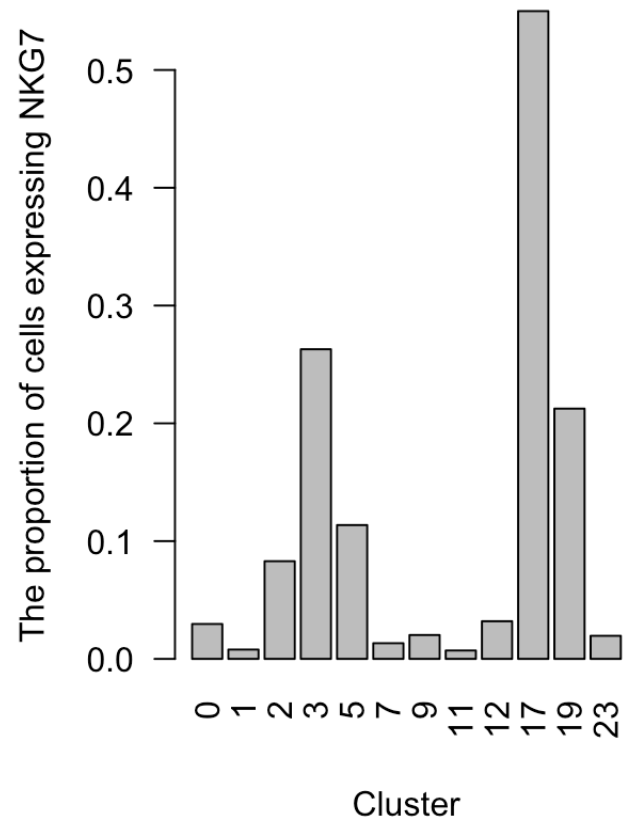
```
subGBMmarker=subset(GBMscmarker,NKG7>0)
index=match(names(table(Idsents(subGBMmarker))),names(table(Idsents(GBMscmarker))))
scmarkercluster=table(Idsents(subGBMmarker))/table(Idsents(GBMscmarker))[index]
```

```
par(mfrow=c(1,2))
barplot(seurat,ylim = c(0,0.55),las=2,main='Seurat',xlab="Cluster",ylab="The proportion of cells expressing NKG7")
barplot(scmarkercluster,ylim = c(0,0.55),las=2,main='Seurat+SCMarker',xlab="Cluster",ylab="The proportion of cells expressing NKG7")
```

## Seurat



## Seurat+SCMarker



FindMarker for each cluster

```
cluster=unique(Ids(GBMscmarker))
cluster=as.numeric(as.character(cluster[order(cluster)]))
gene=c()
for (i in cluster){
  cluster1.markers <- FindMarkers(GBMscmarker, only.pos = TRUE, ident.1 = i, min.pct = 0.25)
  if (length(grep("ERCC-", row.names(cluster1.markers)))!=0){
    cluster1.markers=cluster1.markers[-grep("ERCC-", row.names(cluster1.markers)),]
  }
  cluster1.markers=cluster1.markers[order(cluster1.markers$p_val_adj),]
  gene=union(gene[!is.na(gene)], row.names(cluster1.markers)[1:6])
}
```

```
"NKG7" %in% gene
```

```
## [1] TRUE
```

```
DoHeatmap(GBMscmarker, features = gene,size=3) + NoLegend()+ theme(axis.text.y = element_text(size = 3))
```

