**Домашнее задание №3**

**Анализ single-cell RNA-seq в R**

0. Создаем новый проект в RStudio.

Скачиваем файл из папки R\_single-cell\_RNA-seq > hw > pbmc\_hw\_sub.RData

Файл содержит матрицы каунтов scRNA-seq для образца PBMC, человек.

Вам понадобится скрипт R\_scRNA-seq.R.

1. Загружаем данные в проект, это уже готовый seurat object для образца.

pbmc <- readRDS("~/path\_to/pbmc\_hw\_sub.RData")

2. Фильтрация.

Тщательно фильтруем данные, аккуратно выбираем фильтры на mt-контент, nFeature; рисуем картинки до-после фильтрации. Сколько клеток было, сколько стало после фильтрации?

*Количество клеток: 2700*

> dim(meta) # shows number of cells and number of metadata columns for cells

[1] 2700 3

> head(meta)

orig.ident nCount\_RNA nFeature\_RNA

AAACATACAACCAC-1 pbmc3k 2419 779

AAACATTGAGCTAC-1 pbmc3k 4903 1352

AAACATTGATCAGC-1 pbmc3k 3147 1129

AAACCGTGCTTCCG-1 pbmc3k 2639 960

AAACCGTGTATGCG-1 pbmc3k 980 521

AAACGCACTGGTAC-1 pbmc3k 2163 781

> summary(meta$nCount\_RNA)

Min. 1st Qu. Median Mean 3rd Qu. Max.

546 1756 2196 2365 2762 15818

> summary(meta$nFeature\_RNA)

Min. 1st Qu. Median Mean 3rd Qu. Max.

212.0 690.0 816.0 845.5 952.0 3400.0

> head(pbmc[[]])

orig.ident nCount\_RNA nFeature\_RNA percent.mt

AAACATACAACCAC-1 pbmc3k 2419 779 3.0177759

AAACATTGAGCTAC-1 pbmc3k 4903 1352 3.7935958

AAACATTGATCAGC-1 pbmc3k 3147 1129 0.8897363

AAACCGTGCTTCCG-1 pbmc3k 2639 960 1.7430845

AAACCGTGTATGCG-1 pbmc3k 980 521 1.2244898

AAACGCACTGGTAC-1 pbmc3k 2163 781 1.6643551

percent.rb

AAACATACAACCAC-1 43.69574

AAACATTGAGCTAC-1 42.40261

AAACATTGATCAGC-1 31.68097

AAACCGTGCTTCCG-1 24.25161

AAACCGTGTATGCG-1 14.89796

AAACGCACTGGTAC-1 36.19972

*Количество клеток до фильтрации:*

> #Number of cells before filtration

> dim(pbmc)[2]

*Количество клеток после фильтрации:*

[1] 2700

> #Number of cells after filtration

> dim(pbmc)[2]

[1] 2635

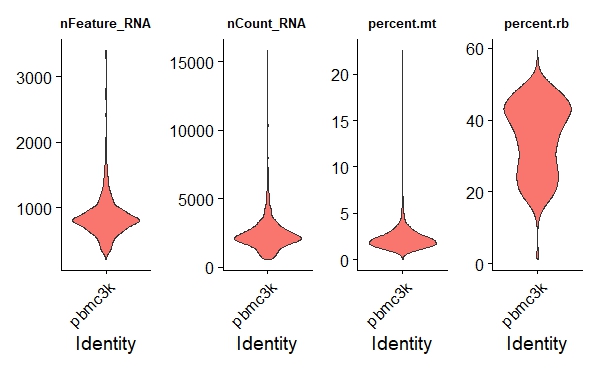
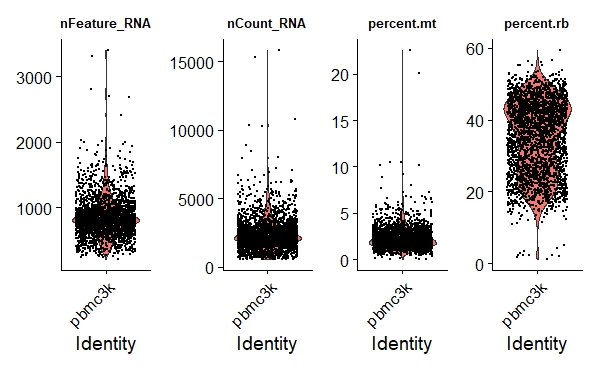


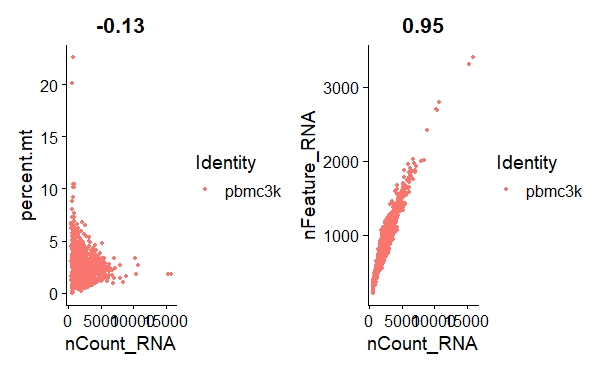
График корреляции между числом каунтов и процентом mt-контента и каунтов и числом генов в образце:

График корреляции между числом каунтов и процентом rb-контента:

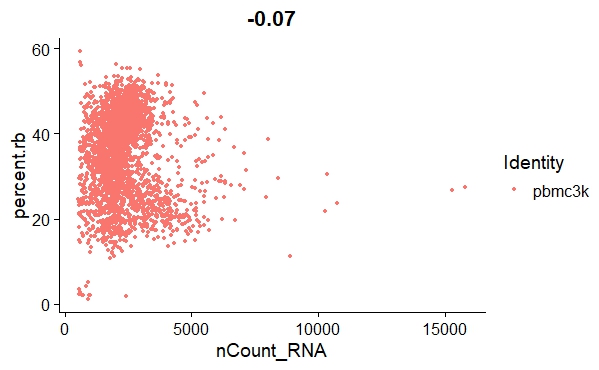


График корреляции между процентом rb-контента и процентом mt-контента:

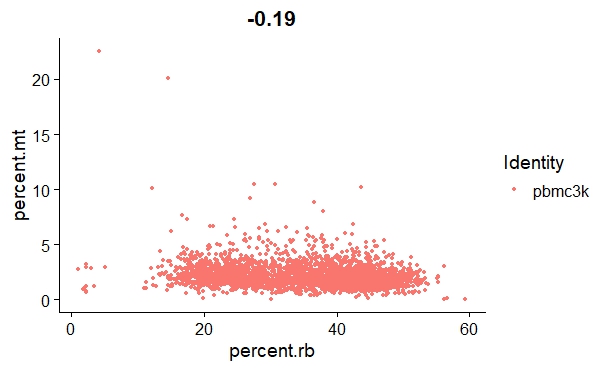
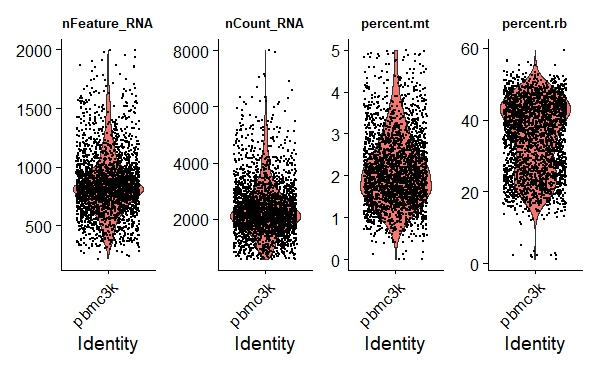


График отфильтрованных данных:



Изображение выглядит как диаграмма, текст

Автоматически созданное описание

3. Делаем нормализацию, variable features, scale, RunPCA, рисуем ElbowPlot. Сколько PC (главных компонент) описывает разницу? 10 или больше, меньше? Выбираем количество PC для следующего шага

> pbmc <- NormalizeData(pbmc) #log normalization

Normalizing layer: counts

Performing log-normalization

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

[----|----|----|----|----|----|----|----|----|----|

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|

> #Next step discovers the most variable features (genes) - these are usually most interesting for downstream analysis.

> pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)

Finding variable features for layer counts

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%

[----|----|----|----|----|----|----|----|----|----|

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|

> top10 <- head(VariableFeatures(pbmc), 10)

> top10

[1] "PPBP" "LYZ" "S100A9" "IGLL5" "GNLY" "FTL" "PF4"

[8] "FTH1" "S100A8" "FCER1A"

# ScaleData converts normalized gene expression to Z-score (values centered at 0 and with variance of 1).

> # It’s stored in pbmc[['RNA']]@scale.data and used in following PCA. Default is to run scaling only on variable genes.

> all.genes <- rownames(pbmc)

> pbmc <- ScaleData(pbmc, features = all.genes) # optionally you can add here vars.to.regress = "percent.mt"

Centering and scaling data matrix

|==========================================================| 100%

> # Run PCA

> pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))

PC\_ 1

Positive: CST3, TYROBP, LST1, AIF1, FTL, LYZ, FCN1, FTH1, S100A9, TYMP

FCER1G, CFD, LGALS1, S100A8, LGALS2, CTSS, SERPINA1, IFITM3, SPI1, CFP

PSAP, IFI30, SAT1, COTL1, S100A11, NPC2, GRN, LGALS3, GSTP1, PYCARD

Negative: MALAT1, LTB, IL32, IL7R, CD2, B2M, ACAP1, CD27, STK17A, CTSW

CD247, GIMAP5, AQP3, CCL5, SELL, GZMA, TRAF3IP3, CST7, MAL, ITM2A

HOPX, GIMAP7, MYC, BEX2, LDLRAP1, GZMK, ZAP70, ETS1, TNFAIP8, RIC3

PC\_ 2

Positive: CD79A, MS4A1, TCL1A, HLA-DQA1, HLA-DQB1, HLA-DRA, LINC00926, CD79B, HLA-DRB1, CD74

HLA-DPB1, HLA-DMA, HLA-DQA2, CD37, HLA-DRB5, HLA-DPA1, HLA-DMB, LTB, FCRLA, HVCN1

BLNK, P2RX5, IGLL5, IRF8, SWAP70, ARHGAP24, FCGR2B, SMIM14, PPP1R14A, C16orf74

Negative: NKG7, PRF1, CST7, GZMB, GZMA, FGFBP2, CTSW, GNLY, B2M, SPON2

CCL4, GZMH, FCGR3A, CCL5, CD247, XCL2, CLIC3, AKR1C3, SRGN, HOPX

TTC38, APMAP, CTSC, S100A4, IGFBP7, ID2, ANXA1, IL32, XCL1, TPST2

PC\_ 3

Positive: HLA-DQA1, CD79A, CD79B, HLA-DQB1, MS4A1, CD74, HLA-DPB1, HLA-DPA1, HLA-DRB1, TCL1A

HLA-DQA2, HLA-DRA, HLA-DRB5, LINC00926, HLA-DMA, HLA-DMB, HVCN1, FCRLA, CD37, GZMB

PLAC8, IRF8, BLNK, FGFBP2, FCGR3A, IGLL5, SWAP70, SMIM14, P2RX5, PRF1

Negative: IL7R, TMSB4X, VIM, IL32, S100A8, S100A6, FYB, GIMAP7, S100A4, MAL

AQP3, S100A9, CD2, S100A10, CD14, GIMAP4, LDLRAP1, RBP7, CD27, ANXA1

LGALS2, S100A12, PPBP, GIMAP5, NDFIP1, NRGN, FOLR3, LYZ, SPARC, GPX1

PC\_ 4

Positive: PPBP, PF4, SDPR, SPARC, GNG11, HIST1H2AC, NRGN, GP9, CLU, CD9

AP001189.4, TUBB1, ITGA2B, PTCRA, CA2, TMEM40, TREML1, MYL9, ACRBP, MMD

F13A1, SEPT5, MPP1, TSC22D1, CMTM5, RP11-367G6.3, GP1BA, LY6G6F, CLEC1B, MAP3K7CL

Negative: MALAT1, VIM, LTB, IL7R, GIMAP7, IL32, EIF3H, S100A10, AQP3, MAL

CD2, CD27, GIMAP4, TRAF3IP3, PPA1, S100A6, S100A4, GIMAP5, S100A11, ANXA1

CCDC109B, CYTIP, KLF6, TRADD, ATP5H, UBXN1, ANXA5, RBM3, TRABD2A, PTGES3

PC\_ 5

Positive: LTB, CKB, MS4A7, IL7R, SIGLEC10, RP11-290F20.3, CYTIP, AQP3, HMOX1, VIM

MPP1, LILRB2, SDPR, HN1, GDI2, CTD-2006K23.1, PF4, PTGES3, CORO1B, TIMP1

VMO1, HIST1H2AC, ATP1A1, ANXA5, GNG11, CA2, CLU, WARS, IFITM2, SPARC

Negative: S100A8, FGFBP2, NKG7, GZMB, GNLY, CCL4, CST7, LGALS2, S100A9, GZMA

PRF1, SPON2, CD14, CCL3, S100A12, RBP7, GZMH, MS4A6A, FOLR3, CTSW

GSTP1, XCL2, CLIC3, TYROBP, IGFBP7, TTC38, XCL1, AKR1C3, ASGR1, LYZ

*Наиболее дифференциально экспрессируемые гены:*

> # Some ways to investigate PCA results and explore the heterogeneity of the data

> print(pbmc[["pca"]], dims = 1:5, nfeatures = 5) # Top 5 genes explaining the difference

PC\_ 1

Positive: CST3, TYROBP, LST1, AIF1, FTL

Negative: MALAT1, LTB, IL32, IL7R, CD2

PC\_ 2

Positive: CD79A, MS4A1, TCL1A, HLA-DQA1, HLA-DQB1

Negative: NKG7, PRF1, CST7, GZMB, GZMA

PC\_ 3

Positive: HLA-DQA1, CD79A, CD79B, HLA-DQB1, MS4A1

Negative: IL7R, TMSB4X, VIM, IL32, S100A8

PC\_ 4

Positive: PPBP, PF4, SDPR, SPARC, GNG11

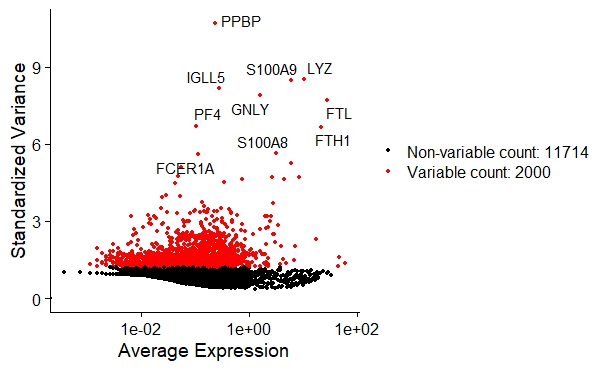
Negative: MALAT1, VIM, LTB, IL7R, GIMAP7

PC\_ 5

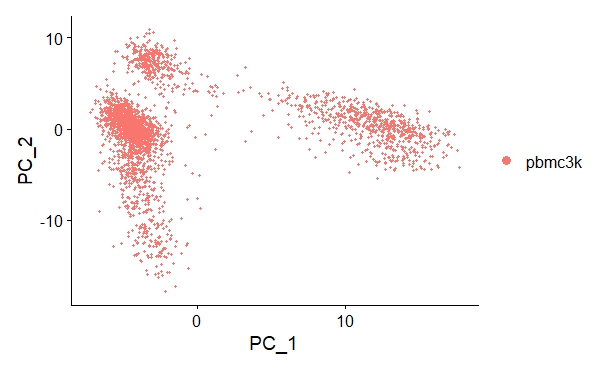
Positive: LTB, CKB, MS4A7, IL7R, SIGLEC10

Negative: S100A8, FGFBP2, NKG7, GZMB, GNLY

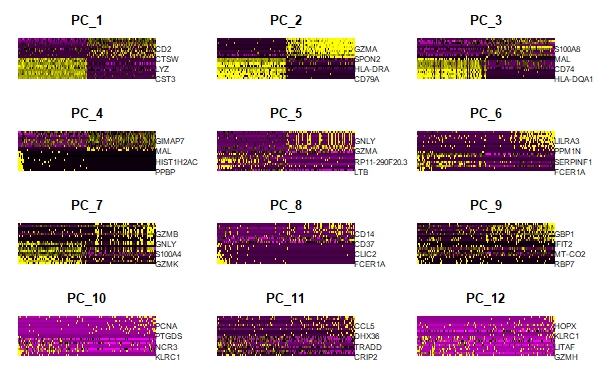
*топ 2000 генов с дифференциальной экспрессией:*



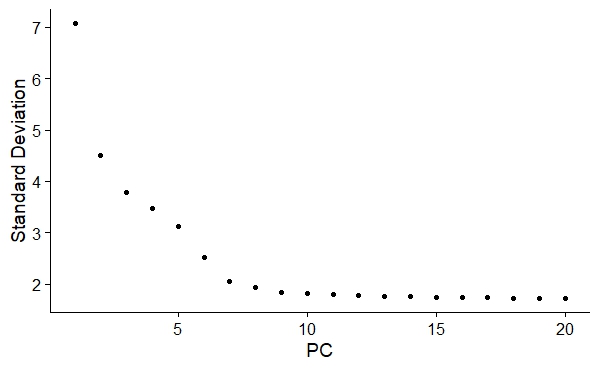
*График PCA:*



*Хитмап дифференциально экспрессируемых генов:*



*ElbowPlot:*



4. UMAP и кластеризация.

Подставляем в функцию RunUMAP параметр dims = 1:n\_PC

Как выглядит UMAP? Сколько получилось кластеров? Какой resolution выбрали (0.3, 0.4, 0.5, 0.6)?

> # Let's run UMAP

> pbmc <- RunUMAP(pbmc, dims = 1:8, verbose = F)

> # Now let's make clustering

> pbmc <- FindNeighbors(pbmc, dims = 1:8) #1:10)

Computing nearest neighbor graph

Computing SNN

> pbmc <- FindClusters(pbmc, resolution = 0.5) # Resolution may vary ~0.4-1.2, depending on how well (biologically) it describes clusters

Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck

Number of nodes: 2635

Number of edges: 91573

Running Louvain algorithm...

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

[----|----|----|----|----|----|----|----|----|----|

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*|

Maximum modularity in 10 random starts: 0.8789

Number of communities: 10

Elapsed time: 0 seconds

> # Look at cluster IDs of the first 5 cells

> head(Idents(pbmc), 5)

AAACATACAACCAC-1 AAACATTGAGCTAC-1 AAACATTGATCAGC-1 AAACCGTGCTTCCG-1

1 3 1 5

AAACCGTGTATGCG-1

6

Levels: 0 1 2 3 4 5 6 7 8 9

> # Table of the clusters composition

> table(pbmc@meta.data$seurat\_clusters)

0 1 2 3 4 5 6 7 8 9

543 512 475 344 277 164 146 127 35 12

> cc.genes.updated.2019

$s.genes

[1] "MCM5" "PCNA" "TYMS" "FEN1" "MCM7"

[6] "MCM4" "RRM1" "UNG" "GINS2" "MCM6"

[11] "CDCA7" "DTL" "PRIM1" "UHRF1" "CENPU"

[16] "HELLS" "RFC2" "POLR1B" "NASP" "RAD51AP1"

[21] "GMNN" "WDR76" "SLBP" "CCNE2" "UBR7"

[26] "POLD3" "MSH2" "ATAD2" "RAD51" "RRM2"

[31] "CDC45" "CDC6" "EXO1" "TIPIN" "DSCC1"

[36] "BLM" "CASP8AP2" "USP1" "CLSPN" "POLA1"

[41] "CHAF1B" "MRPL36" "E2F8"

$g2m.genes

[1] "HMGB2" "CDK1" "NUSAP1" "UBE2C" "BIRC5" "TPX2"

[7] "TOP2A" "NDC80" "CKS2" "NUF2" "CKS1B" "MKI67"

[13] "TMPO" "CENPF" "TACC3" "PIMREG" "SMC4" "CCNB2"

[19] "CKAP2L" "CKAP2" "AURKB" "BUB1" "KIF11" "ANP32E"

[25] "TUBB4B" "GTSE1" "KIF20B" "HJURP" "CDCA3" "JPT1"

[31] "CDC20" "TTK" "CDC25C" "KIF2C" "RANGAP1" "NCAPD2"

[37] "DLGAP5" "CDCA2" "CDCA8" "ECT2" "KIF23" "HMMR"

[43] "AURKA" "PSRC1" "ANLN" "LBR" "CKAP5" "CENPE"

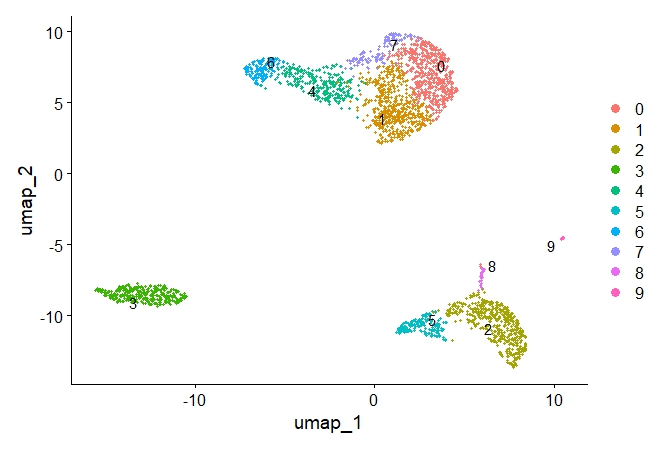
[49] "CTCF" "NEK2" "G2E3" "GAS2L3" "CBX5" "CENPA"

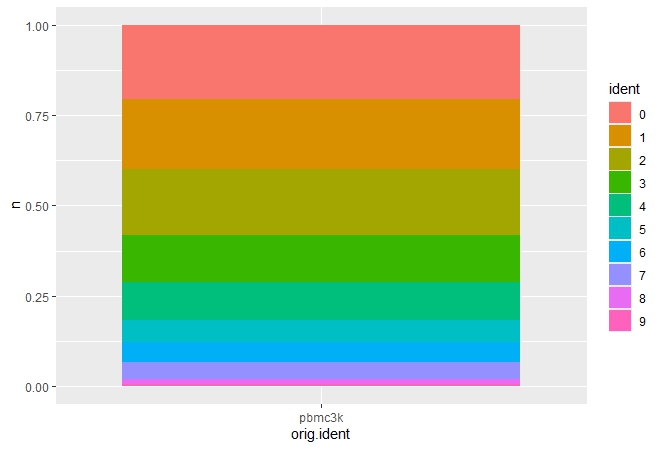
> table(pbmc[[]]$Phase)

G1 G2M S

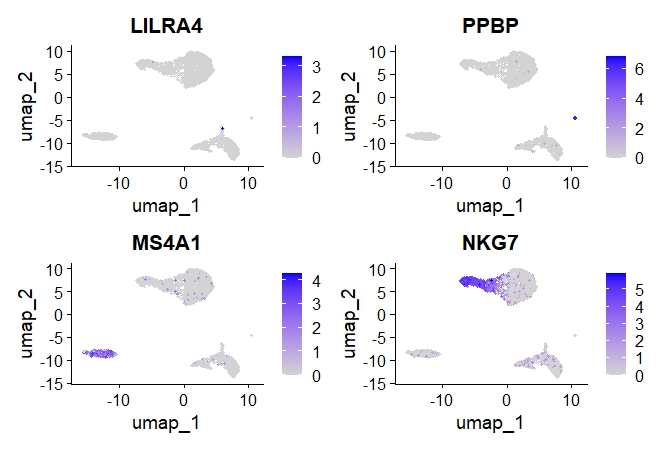
1165 408 1062

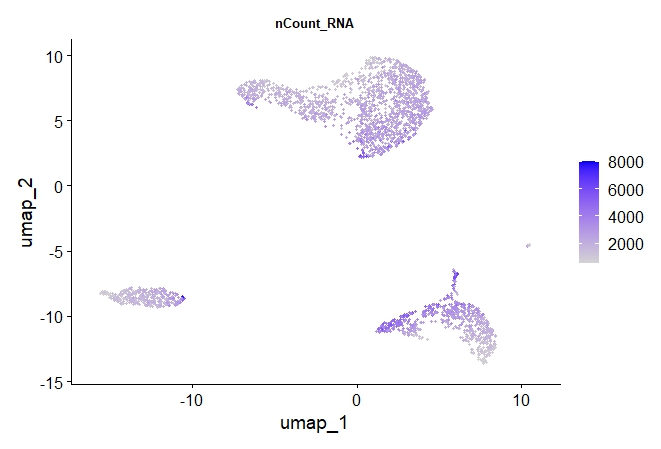
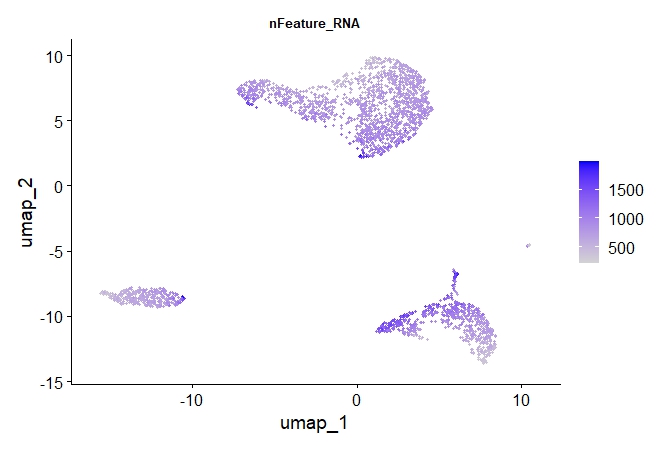
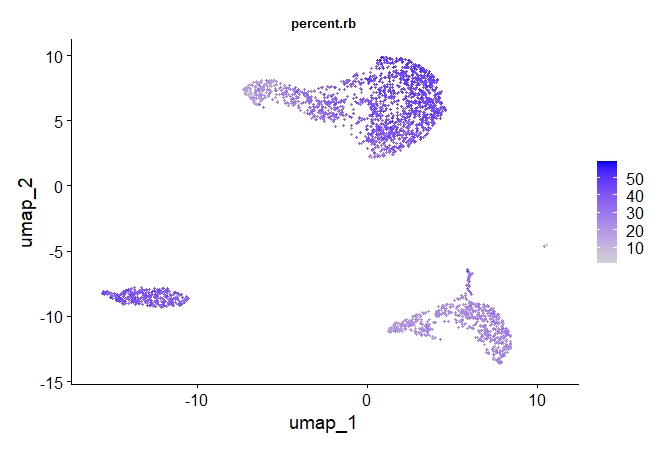
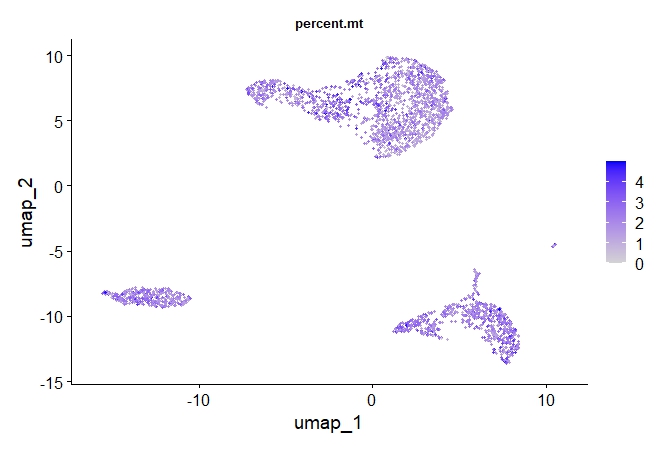
*UMAP-график (получилось 10 кластеров клеток):*

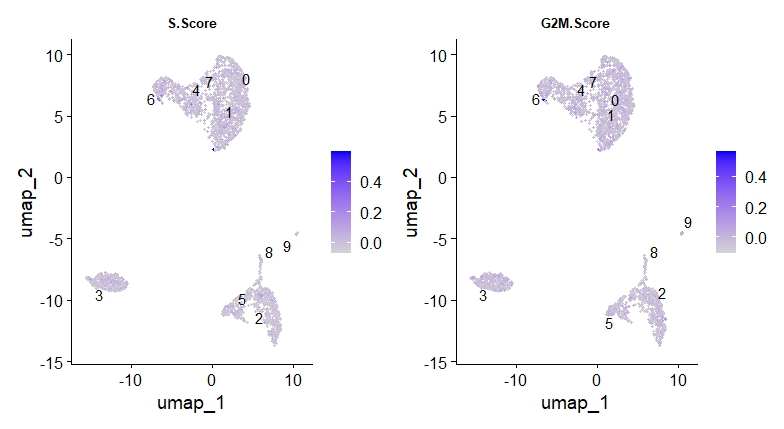
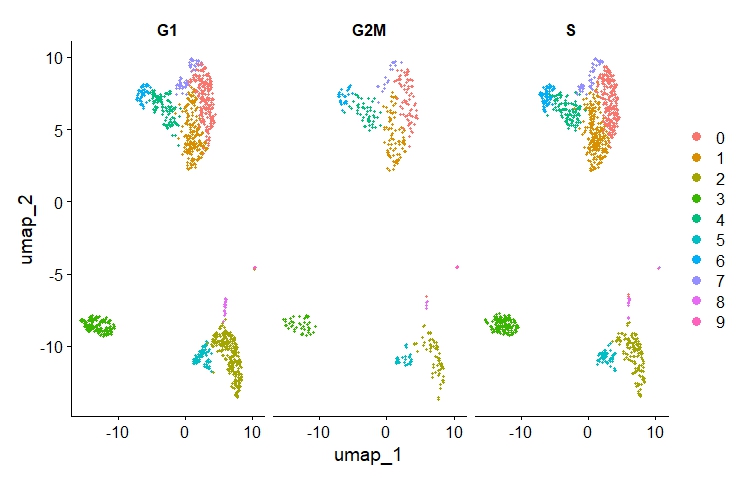


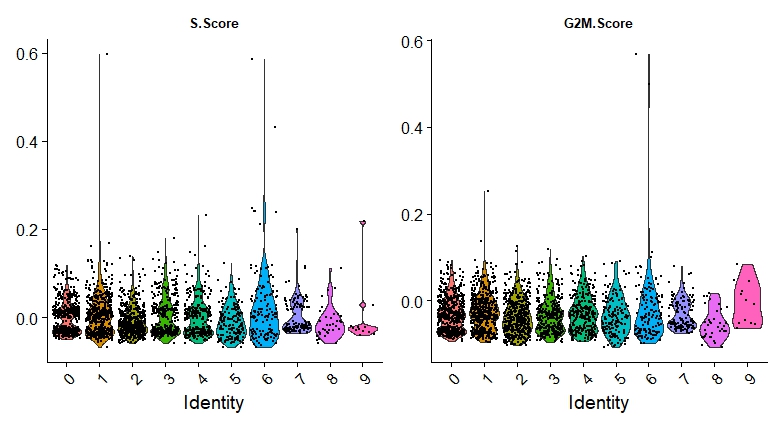


Проверка QC:









5. Аннотируем некоторые кластеры.

Используйте функцию FindAllMarkers() для идентификации дифференциально ап-регулированных генов для каждого кластера. По представленной таблице маркеров попробуйте определить тип клеток кластеров. Bизуализируйте некоторые маркеры с помощью функций FeaturePlot(), VlnPlot().



> # find markers for every cluster compared to all remaining cells, report only the positive

> # ones

> pbmc.markers <- FindAllMarkers(pbmc, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

Calculating cluster 0

For a (much!) faster implementation of the Wilcoxon Rank Sum Test,

(default method for FindMarkers) please install the presto package

--------------------------------------------

install.packages('devtools')

devtools::install\_github('immunogenomics/presto')

--------------------------------------------

After installation of presto, Seurat will automatically use the more

efficient implementation (no further action necessary).

This message will be shown once per session

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=02s

Calculating cluster 1

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=03s

Calculating cluster 2

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=05s

Calculating cluster 3

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=02s

Calculating cluster 4

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=03s

Calculating cluster 5

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=06s

Calculating cluster 6

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=04s

Calculating cluster 7

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=01s

Calculating cluster 8

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=06s

Calculating cluster 9

|++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=03s

> pbmc.markers %>%

+ group\_by(cluster) %>%

+ slice\_max(n = 2, order\_by = avg\_log2FC)

# A tibble: 20 × 7

# Groups: cluster [10]

p\_val avg\_log2FC pct.1 pct.2 p\_val\_adj cluster gene

*<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<fct>* *<chr>*

1 2.27e- 92 2.32 0.501 0.118 3.11e- 88 0 CCR7

2 1.64e- 57 2.10 0.387 0.108 2.25e- 53 0 LEF1

3 5.95e- 60 2.16 0.41 0.108 8.16e- 56 1 AQP3

4 2.47e- 42 2.07 0.273 0.064 3.38e- 38 1 CD40LG

5 7.33e-141 7.30 0.303 0.004 1.01e-136 2 FOLR3

6 5.87e-123 6.76 0.28 0.006 8.05e-119 2 S100A12

7 5.18e-272 7.38 0.564 0.009 7.10e-268 3 LINC00926

8 5.41e-237 7.13 0.488 0.007 7.42e-233 3 VPREB3

9 5.63e-166 4.32 0.592 0.056 7.72e-162 4 GZMK

10 2.89e- 92 3.60 0.437 0.061 3.97e- 88 4 GZMH

11 2.12e-165 5.86 0.366 0.005 2.91e-161 5 CKB

12 1.22e-211 5.45 0.5 0.009 1.67e-207 5 CDKN1C

13 3.97e-185 6.21 0.493 0.013 5.45e-181 6 AKR1C3

14 2.05e-269 5.98 0.986 0.07 2.80e-265 6 GZMB

15 1.99e- 4 1.95 0.094 0.266 1 e+ 0 7 NDUFA2

16 5.38e- 3 1.47 0.157 0.314 1 e+ 0 7 TBCB

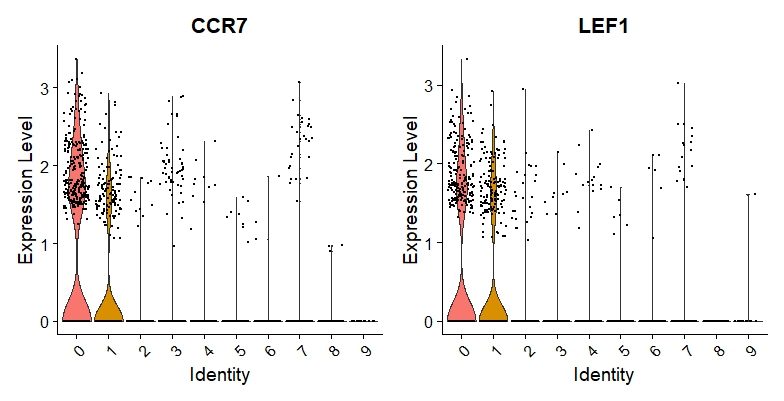
17 1.65e-198 8.06 0.457 0.002 2.27e-194 8 SERPINF1

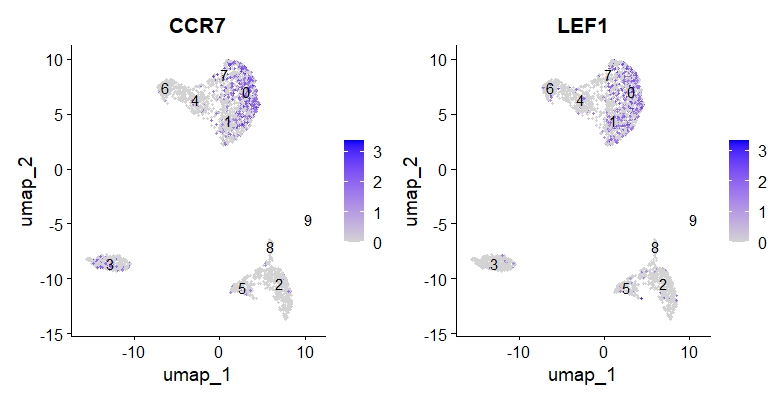
18 7.97e-269 8.05 0.857 0.01 1.09e-264 8 FCER1A

19 0 14.4 0.583 0 0 9 LY6G6F

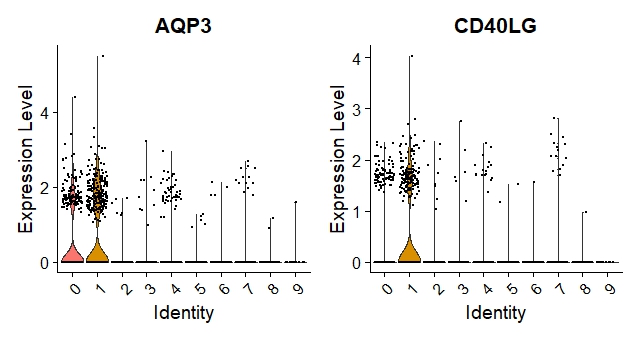
20 2.46e-192 14.0 0.333 0 3.38e-188 9 RP11-879F14.2

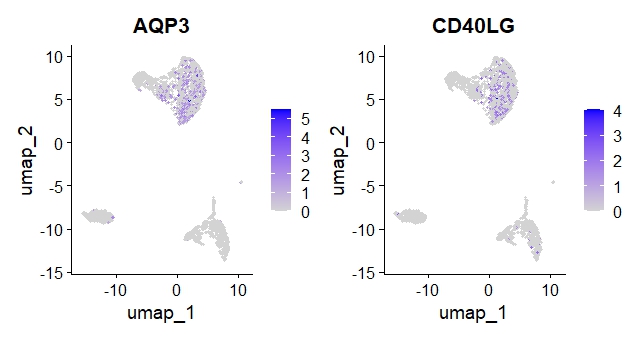
*кластер 0:*



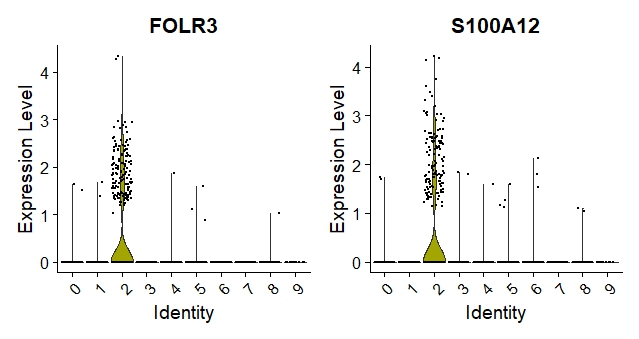


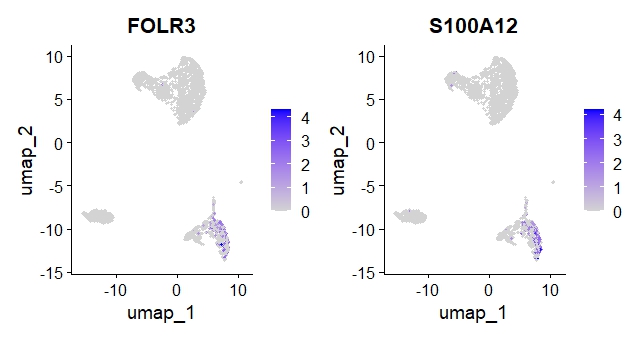
*кластер 1:*

**

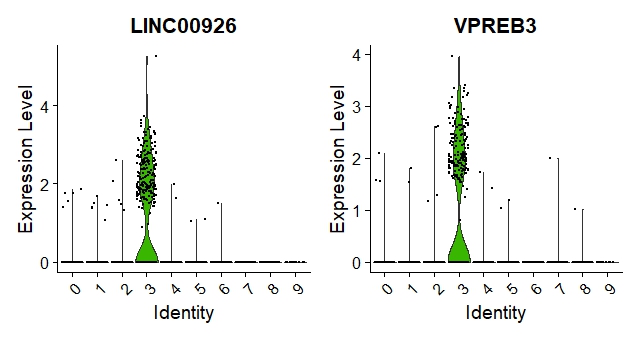


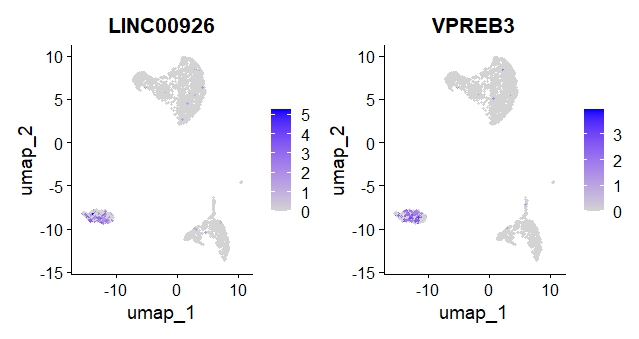
*кластер 2:*

**

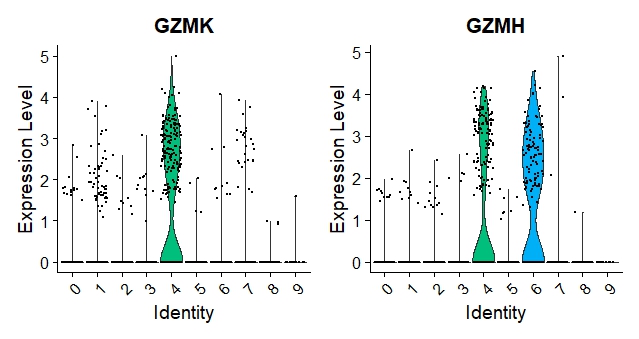


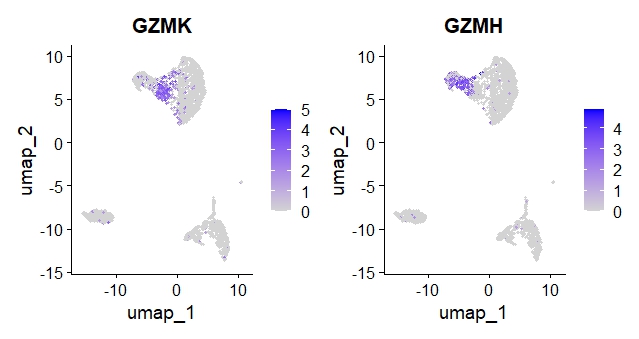
*кластер 3:*

**

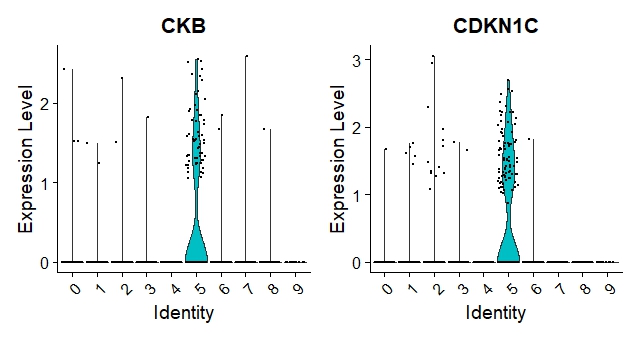


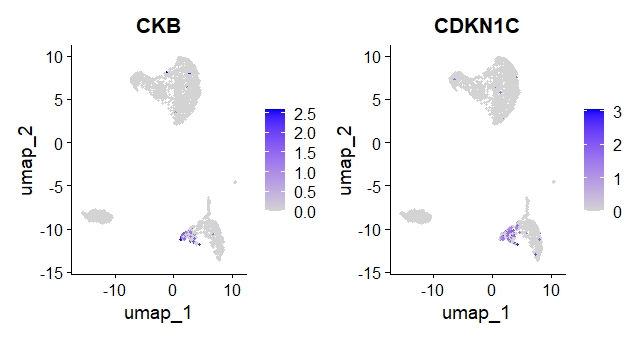
*кластер 4:*

**

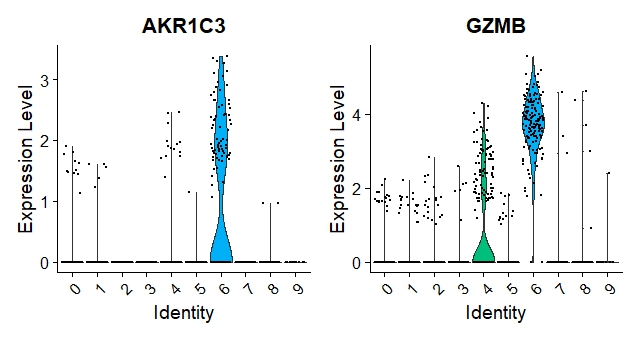


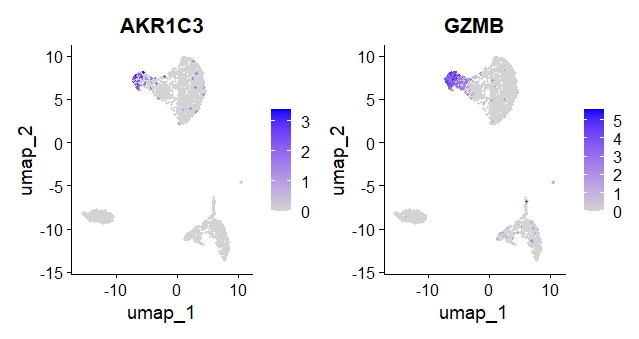
*кластер 5:*

**

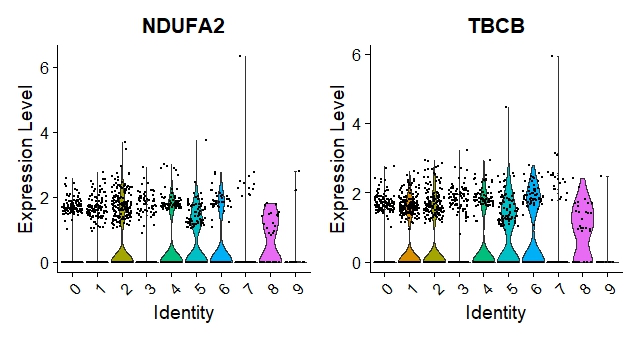


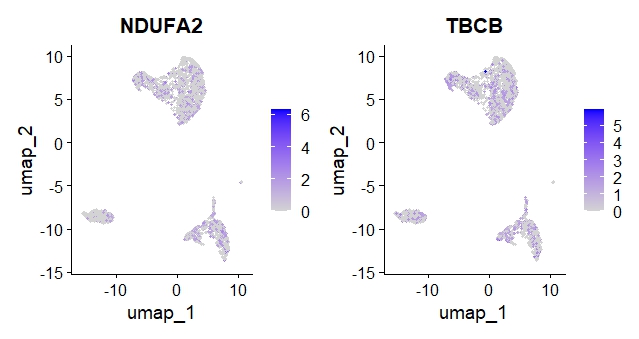
*кластер 6:*

**

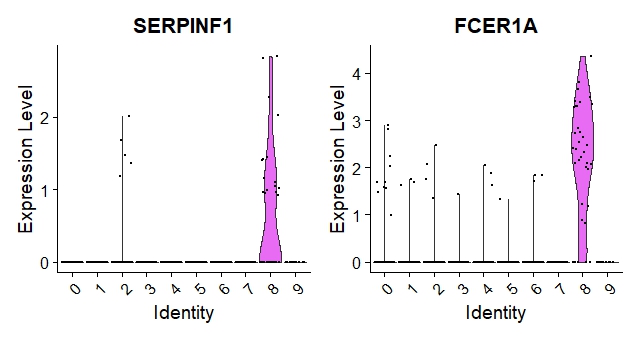


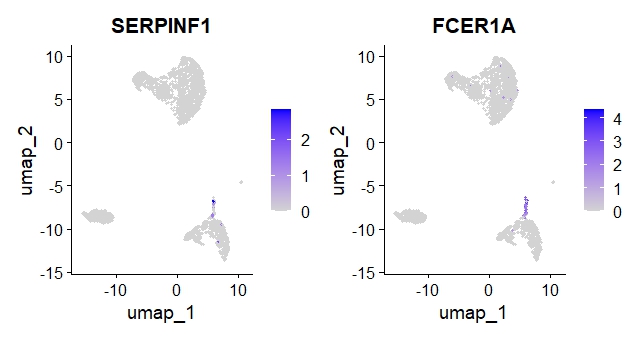
*кластер 7:*

**

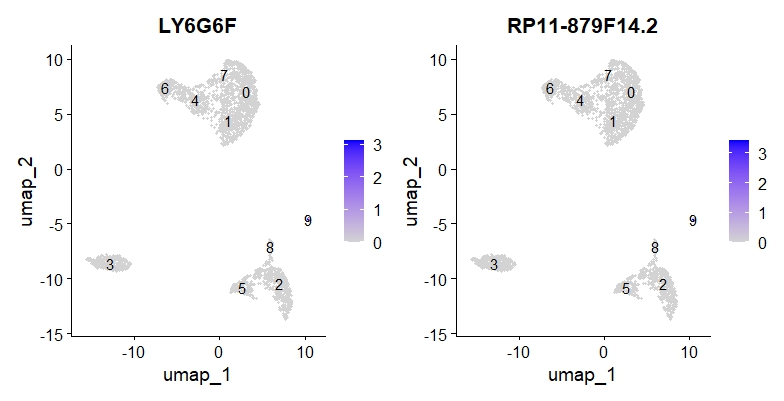
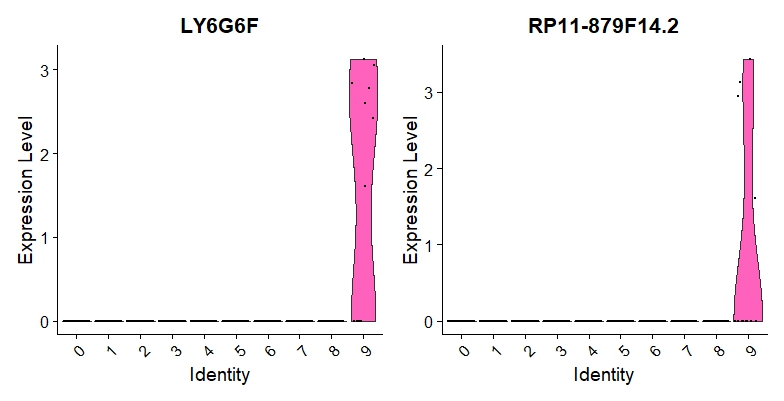


*кластер 8:*

**



*Кластер 9:*



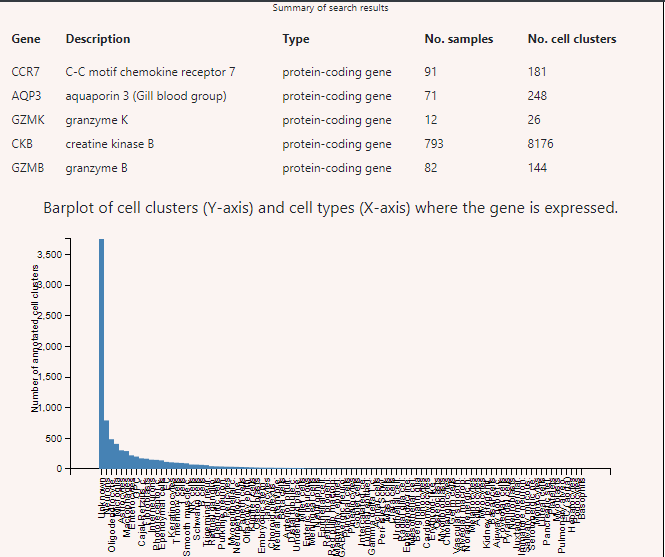
\*Можно использовать базу <https://panglaodb.se/search.html> - по гену подскажет в каком типе клеток обычно экспрессируется этот ген.

В результате проаннотируйте хотя бы какое-то количество кластеров; можно присвоить одинаковую аннотацию нескольким кластерам, если они похожи.

Результат: UMAP с (частично) проаннотированными кластерами.

*CCR7, LEF1, AQP3, CD40LG, FOLR3, S100A12, LINC00926, VPREB3, GZMK, GZMH, CKB, CDKN1C, AKR1C3, GZMB, NDUFA2, TBCB, SERPINF1, FCER1A, LY6G6F, RP11-879F14.2*

*CCR7, AQP3, GZMK, CKB, GZMB*



*library(Seurat)*

*library(ggplot2)*

*seurat\_data <- readRDS("C:/Users/nasty/Desktop/Магистратура/ДЗ RNA/hw3/pbmc\_norm.rds")*

*# Определяем типы клеток для каждого кластера на основе экспрессии генов*

*# кластер 0 представляет T-хелперы,*

*# кластер 1 представляет эпителиальные клетки, кластер 4 представляет T-киллеры,*

*# кластер 5 представляет мышечные клетки, а кластер 6 представляет B-клетки.*

*cluster\_annotations <- list(*

*T\_helpers = "0",*

*Epithelial\_cells = "1",*

*T\_killers = "4",*

*Muscle\_cells = "5",*

*B\_cells = "6"*

*)*

*# Преобразуем список аннотаций кластеров в столбец данных*

*cluster\_annotations\_df <- data.frame(*

*cluster = as.numeric(unlist(cluster\_annotations)),*

*annotation = rep(names(cluster\_annotations), lengths(cluster\_annotations)),*

*stringsAsFactors = FALSE*

*)*

*seurat\_data[["cluster\_annotations"]] <- as.character(cluster\_annotations\_df$annotation)*

*seurat\_data$cluster\_annotations <- as.character(cluster\_annotations\_df$annotation)*

*# Визуализируем аннотированные кластеры с помощью UMAP*

*DimPlot(seurat\_data, reduction = "umap", label = TRUE, group.by = "cluster\_annotations", repel = TRUE) +*

*scale\_color\_manual(values = c(*

*T\_helpers = "blue",*

*Epithelial\_cells = "green",*

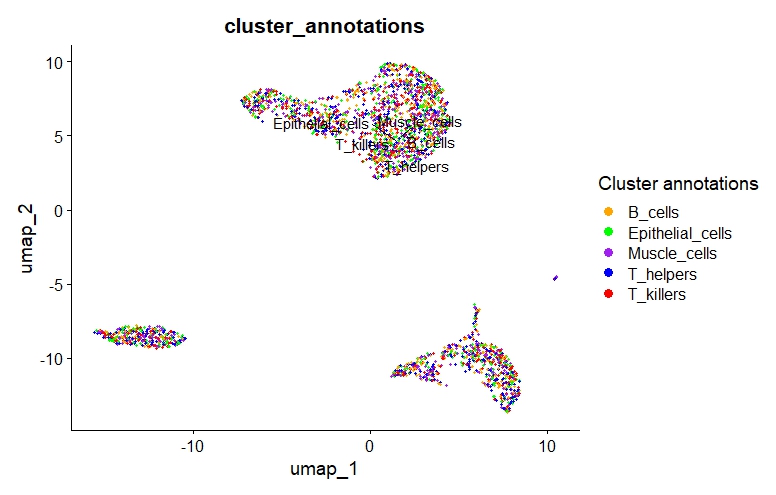
*T\_killers = "red",*

*Muscle\_cells = "purple",*

*B\_cells = "orange"*

*)) +*

*labs(color = "Cluster annotations")*



Single-cell RNA-seq typical Analysis

Изображение выглядит как диаграмма

Автоматически созданное описание