

scRNA-seq vein

Chris Doan

2022-09-15

```
mtx_obj_NC <- ReadMtx(mtx = 'scRNA_seq/NC/matrix.mtx.gz',
                       features = 'scRNA_seq/NC/features.tsv.gz',
                       cells = 'scRNA_seq/NC/barcodes.tsv.gz')

## as(<dgTMatrix>, "dgCMatrix") is deprecated since Matrix 1.4-2; do as(., "CsparseMatrix") instead

seurat_mtx_NC <- CreateSeuratObject(counts = mtx_obj_NC, project = 'CV', min.cells = 5)

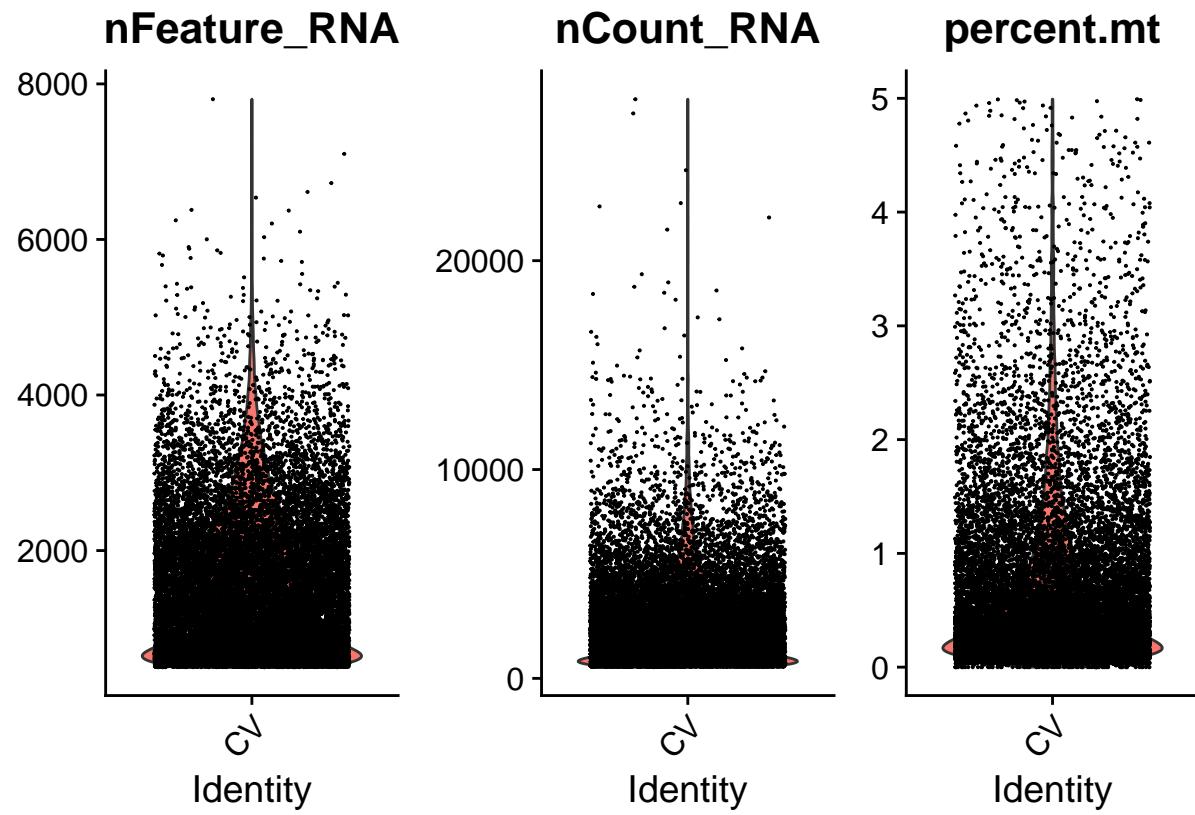
seurat_mtx_NC[['percent.mt']] <- PercentageFeatureSet(seurat_mtx_NC, pattern = '^MT-')

seurat_mtx_NC <- subset(seurat_mtx_NC, subset = nCount_RNA < 40000 &
                           nFeature_RNA > 500 &
                           percent.mt < 5)

seurat_mtx_NC <- NormalizeData(object = seurat_mtx_NC)

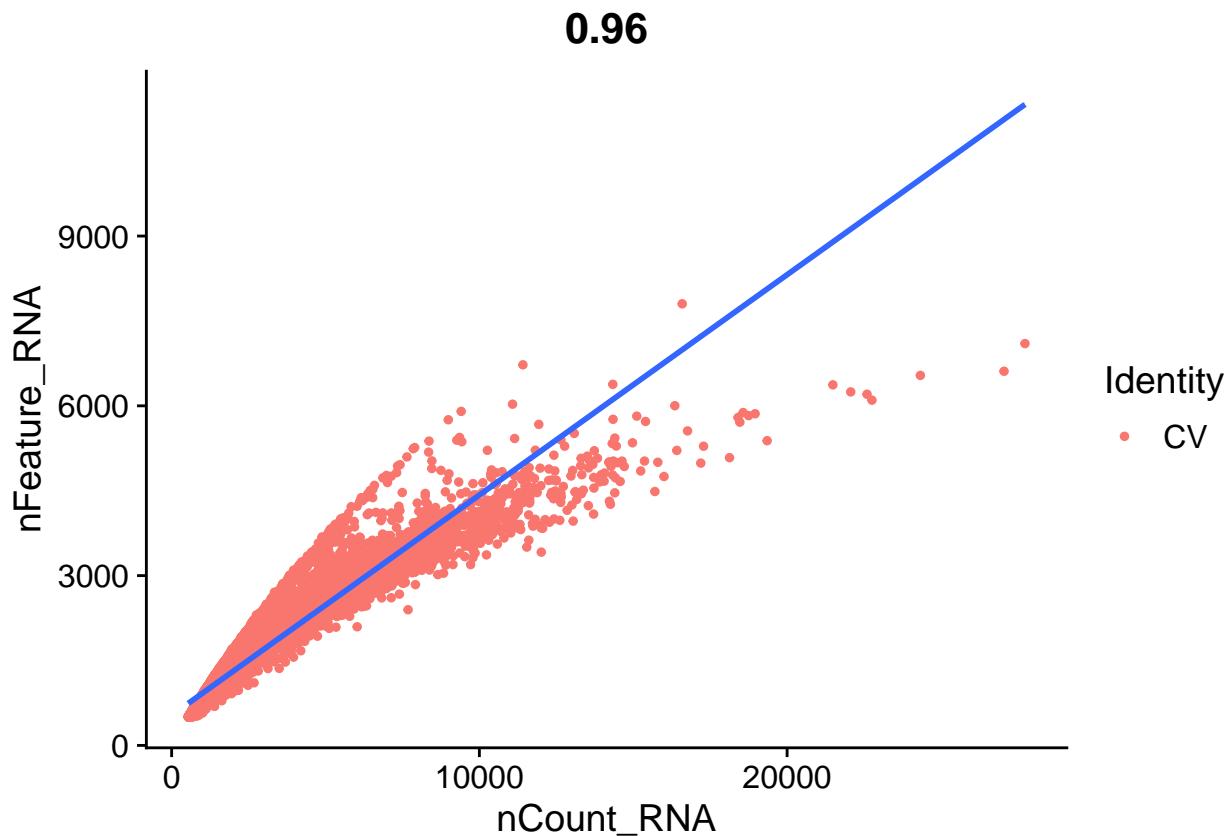
VlnPlot(seurat_mtx_NC, features = c('nFeature_RNA', 'nCount_RNA', 'percent.mt'), ncol = 3) +
  geom_smooth(method = 'lm')

## `geom_smooth()` using formula 'y ~ x'
```



```
FeatureScatter(seurat_mtx_NC, feature1 = 'nCount_RNA', feature2 = 'nFeature_RNA') +  
  geom_smooth(method = 'lm')
```

```
## `geom_smooth()` using formula 'y ~ x'
```

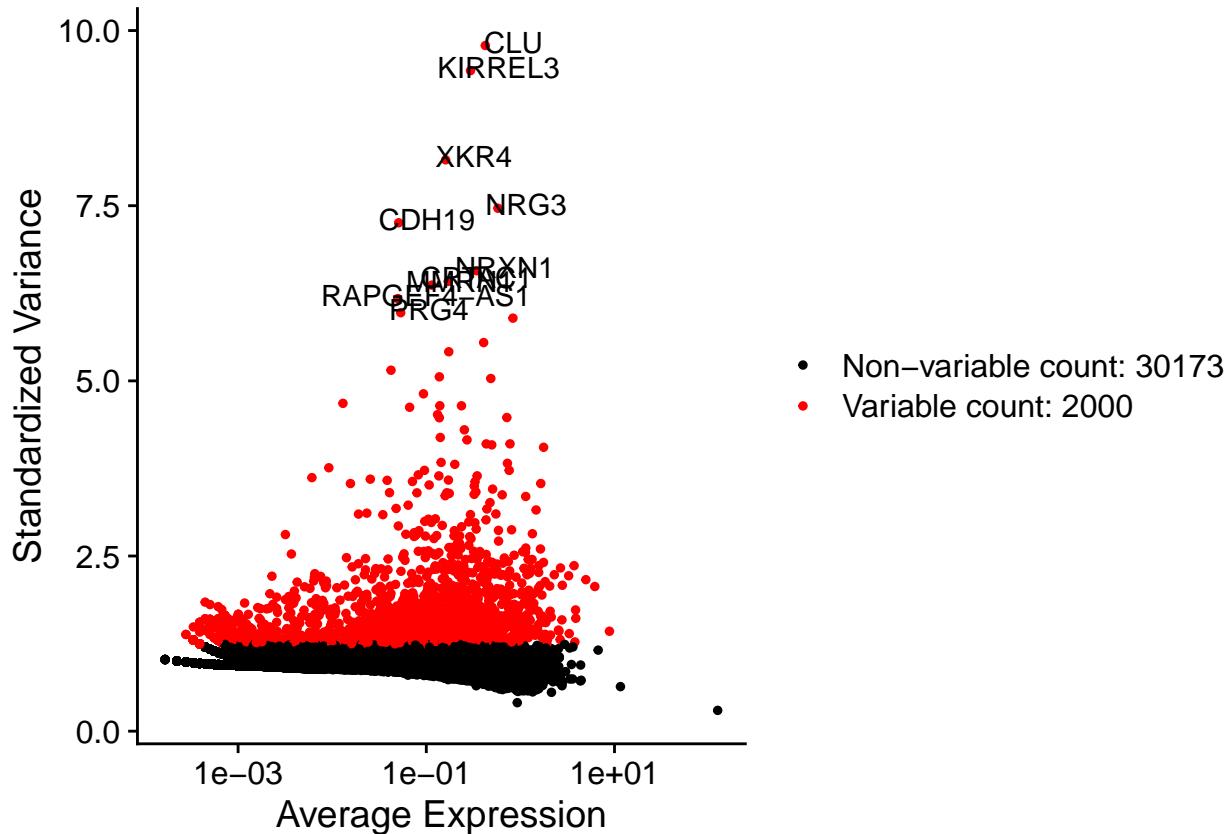


```
seurat_mtx_NC <- FindVariableFeatures(object = seurat_mtx_NC)
```

```
top10 <- head(VariableFeatures(seurat_mtx_NC), 10)
```

```
plot1 <- VariableFeaturePlot(seurat_mtx_NC)
LabelPoints(plot = plot1, points = top10, rebel = TRUE)
```

```
## Warning: Ignoring unknown parameters: rebel
```



```

mtx_obj_VV <- ReadMtx mtx = 'scRNA_seq/VV/matrix.mtx.gz',
  features = 'scRNA_seq/VV/features.tsv.gz',
  cells = 'scRNA_seq/VV/barcodes.tsv.gz')

seurat_mtx_VV <- CreateSeuratObject(counts = mtx_obj_VV, project = 'CV', min.cells = 5)

seurat_mtx_VV[['percent.mt']] <- PercentageFeatureSet(seurat_mtx_VV, pattern = '^MT-')

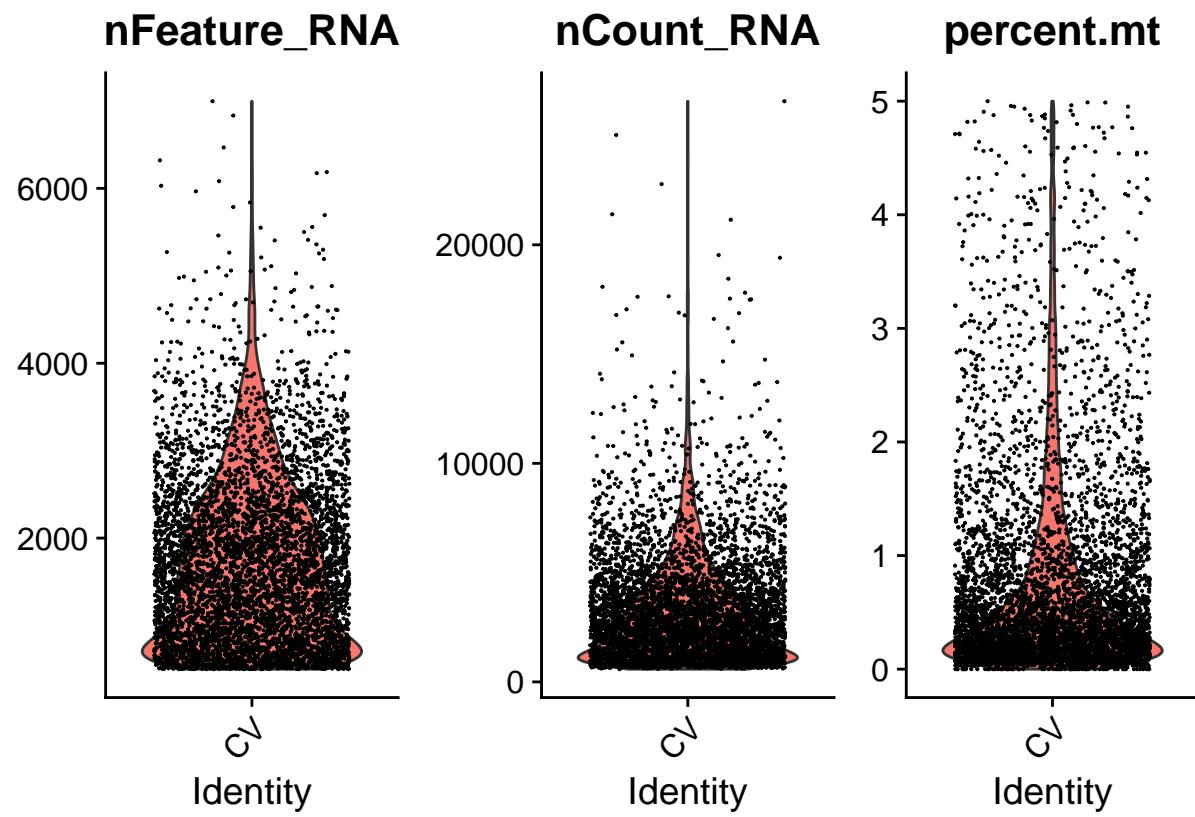
seurat_mtx_VV <- subset(seurat_mtx_VV, subset = nCount_RNA < 40000 &
  nFeature_RNA > 500 &
  percent.mt <5)

seurat_mtx_VV <- NormalizeData(object = seurat_mtx_VV)

VlnPlot(seurat_mtx_VV, features = c('nFeature_RNA', 'nCount_RNA', 'percent.mt'), ncol = 3) +
  geom_smooth(method = 'lm')

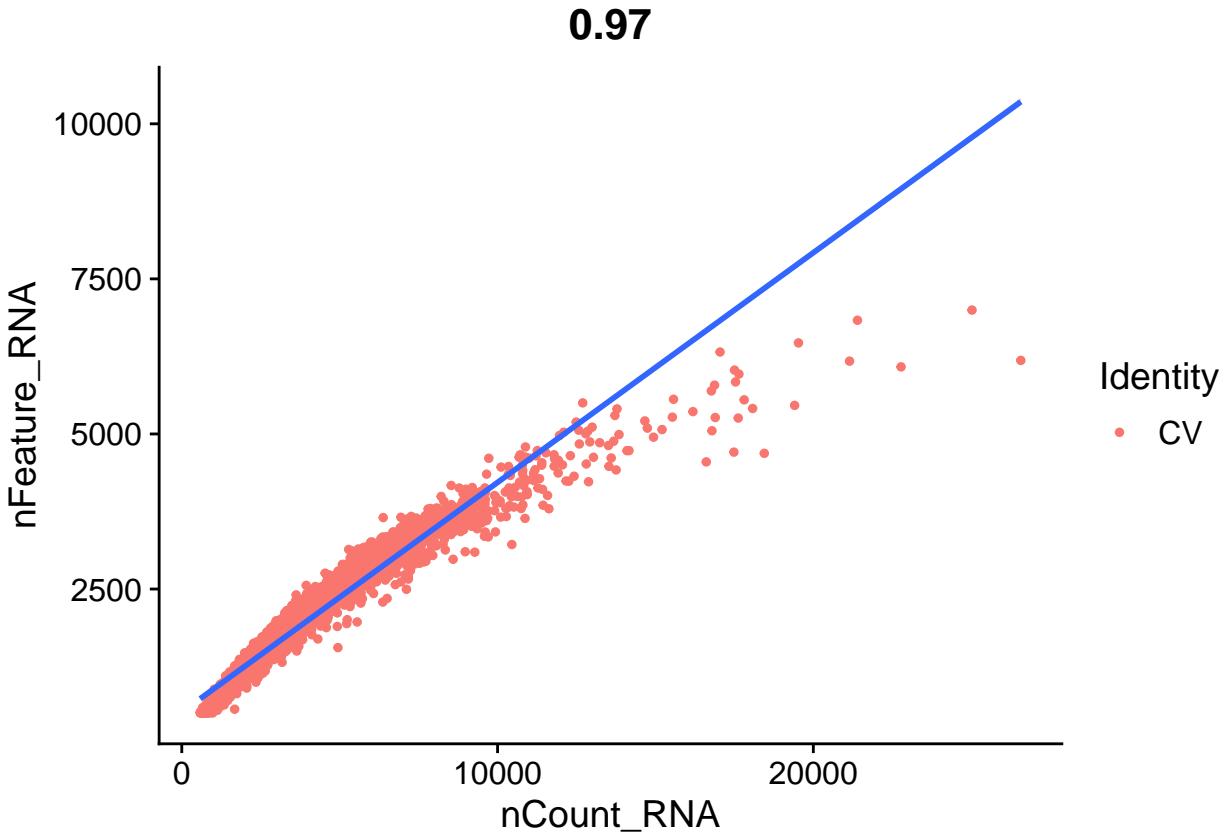
## `geom_smooth()` using formula 'y ~ x'

```



```
FeatureScatter(seurat_mtx_VV, feature1 = 'nCount_RNA', feature2 = 'nFeature_RNA') +  
  geom_smooth(method = 'lm')
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```

seurat_mtx_VV <- FindVariableFeatures(object = seurat_mtx_VV)

merged_seurat <- merge(seurat_mtx_NC, y = c(seurat_mtx_VV),
                        add.cell.ids = c('NC','VV'),
                        project = 'CV')

#merged_seurat@meta.data

merged_seurat$sample <- rownames(merged_seurat@meta.data)

merged_seurat@meta.data <- separate(merged_seurat@meta.data, col = 'sample', into = c('condition','Barco',
                                                                                     sep = '_'))

merged_seurat <- FindVariableFeatures(merged_seurat)

merged_seurat <- ScaleData(merged_seurat)

## Centering and scaling data matrix

merged_seurat <- RunPCA(merged_seurat)

## PC_

```

```

## Positive: PRKG1, CNTN4, COL1A2, LAMA2, ITGBL1, COL14A1, COL6A3, MIR100HG, DMD, SGCD
##      SLIT2, SORBS1, ANTXR1, SLIT3, COL1A1, COL3A1, CARMN, BICC1, FGF14, ABI3BP
##      EPHA3, ROBO2, AFF3, GPC6, LPAR1, PRUNE2, FBLN1, PDLIM3, KCNMA1, ADGRL3
## Negative: PECAM1, VWF, SHANK3, PTPRB, MECOM, MCTP1, ANO2, ST6GALNAC3, MYO5C, ELM01
##      EGFL7, ADGRL4, PRKCH, PITPN1, PIK3R3, FLT1, ITGA6, SNTG2, CYYR1, MCF2L
##      CMIP, TPO, FLI1, DOCK4, EMCN, AC105450.1, ABLIM1, RALGAPA2, PPP1R16B, DOCK9
## PC_ 2
## Positive: RYR2, CARMN, ADGRL3, DTNA, TNC, AKAP6, ITGA8, MRVI1, DEC1, CACNB2
##      P2RX1, SDK1, AC022325.2, MYOCD, PDE4D, LMOD1, INPP4B, ADCY5, FHOD3, PPP2R2B
##      GALNT17, NPNT, PRUNE2, LINCO1592, ACTG2, PCA3, JPH2, MPP7, PDZRN4, DGKG
## Negative: CFH, ABCA6, MEG3, IL16, ABCA10, FBLN1, EPHA3, BICC1, LAMA2, CDH11
##      GAS7, DCN, ABI3BP, FAP, MEG8, GREB1, TSHZ2, PRICKLE1, UST, ABCA8
##      ABCA9, TNXB, ADD3, FBLN2, NOX4, COL6A3, SVEP1, MIR100HG, PDGFRA, LTBP2
## PC_ 3
## Positive: CARMN, RBPMS, SORBS1, TNC, SULF1, DEC1, ITGA5, MRVI1, MYH10, STK38L
##      PRUNE2, CRIM1, STAMBPL1, ITGA8, PDLIM3, P2RX1, PCDH7, GALNT18, PRKG1, ENAH
##      MYOCD, MYO1E, LMOD1, PCA3, NPNT, PDZRN4, AC022325.2, AC005358.1, PLCB4, CNTN4
## Negative: MRC1, DOCK2, MSR1, TBXAS1, CD163, RBM47, IGSF21, PTPRC, MS4A6A, MS4A4E
##      MS4A4A, F13A1, SYK, FMN1, SLC02B1, PIK3R5, DOCK8, ADAM28, CLEC7A, IQGAP2
##      CSF1R, RUNX1, MS4A7, HLA-DQA1, CYTH4, CD83, LSAMP, AOA, CSF2RA, NCKAP5
## PC_ 4
## Positive: LINCO2055, AC105402.3, GPC5, NRXN1, CNTN5, LRRC4C, NRXN3, AC109466.1, CADM2, NRG1
##      KCNIP4, KAZN, PTPRT, ASIC2, CTNNA3, AC011287.1, ADAMSL1, CTNND2, RELN, DCC
##      GRM7, IL1RAPL1, LINCO1505, AC016766.1, LRRTM4, AC008415.1, MIR924HG, SG01-AS1, PCAT1, CCSER1
## Negative: MS4A6A, CD163, HLA-DQA1, CLEC7A, MS4A7, MS4A4E, CD74, HLA-DQB1, PTPRC, HLA-DPB1
##      HLA-DRB1, MRC1, CYBB, STAB1, ADAM28, CTSB, HLA-DRA, CD83, HLA-DMB, CYTH4
##      FCGR2A, CIITA, RGS1, MS4A4A, CSF1R, TLR2, SMAP2, EMB, LAPTM5, XIST
## PC_ 5
## Positive: FKBP5, NAV2, TTY14, SLC8A1, GPC6, DTNA, LINCO0278, HPSE2, DMD, RYR2
##      AC022325.2, MAMDC2, ARHGAP6, PDE4D, RCAN2, KCNMA1, SDK1, CACNB2, PRKG1, HDAC9
##      MACROD2, NTRK3, RBPMS, HSPH1, COL4A5, LINCO2388, SOX6, BMPR1B, ADCY2, PPP2R2B
## Negative: MT-CO3, MT-ATP6, MT-ND3, MT-CO1, XIST, MT-CYB, MT-ND4, MT-ND1, MTRNR2L12, MT-ND2
##      SPARC, MT-CO2, S100A6, VIM, CLU, BGN, CFD, COL1A1, CD63, TIMP1
##      MGP, RPS18, MT-ND5, CCDC80, COL3A1, RPS23, CST3, RPL19, LUM, SFRP4

merged_seurat <- RunUMAP(merged_seurat, reduction = "pca", dims = 1:20)

## 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

## 15:02:16 UMAP embedding parameters a = 0.9922 b = 1.112

## 15:02:16 Read 23479 rows and found 20 numeric columns

## 15:02:16 Using Annoy for neighbor search, n_neighbors = 30

## 15:02:16 Building Annoy index with metric = cosine, n_trees = 50

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

```

```

## ****|  

## 15:02:18 Writing NN index file to temp file /var/folders/07/b7qwj15d1zs41kh2whcqhh0c0000gp/T//RtmpKb  

## 15:02:18 Searching Annoy index using 1 thread, search_k = 3000  

## 15:02:23 Annoy recall = 100%  

## 15:02:24 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30  

## 15:02:25 Initializing from normalized Laplacian + noise (using irlba)  

## 15:02:26 Commencing optimization for 200 epochs, with 1047440 positive edges  

## 15:02:42 Optimization finished

merged_seurat <- FindNeighbors(merged_seurat, reduction = "pca", dims = 1:20)

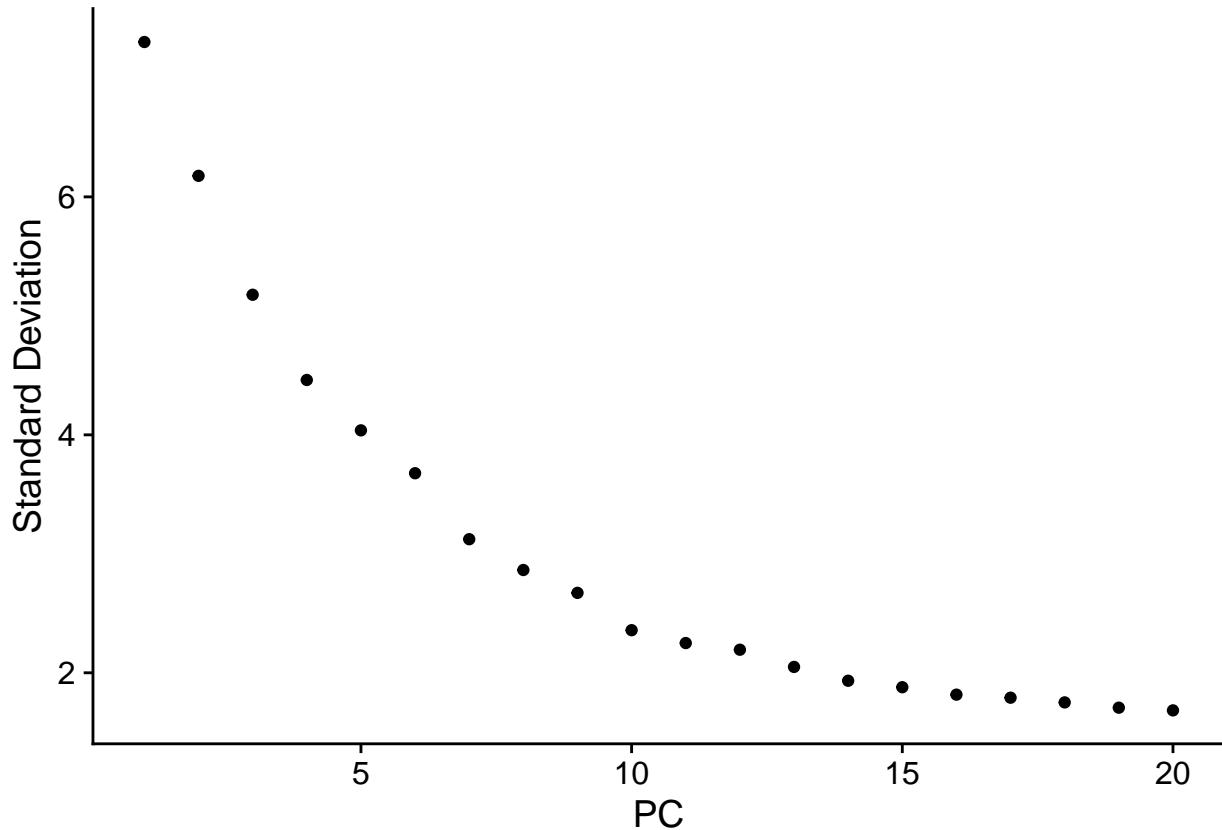
## Computing nearest neighbor graph
## Computing SNN

merged_seurat <- FindClusters(merged_seurat, resolution = 0.5)

## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 23479
## Number of edges: 787056
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9247
## Number of communities: 15
## Elapsed time: 3 seconds

ElbowPlot(merged_seurat)

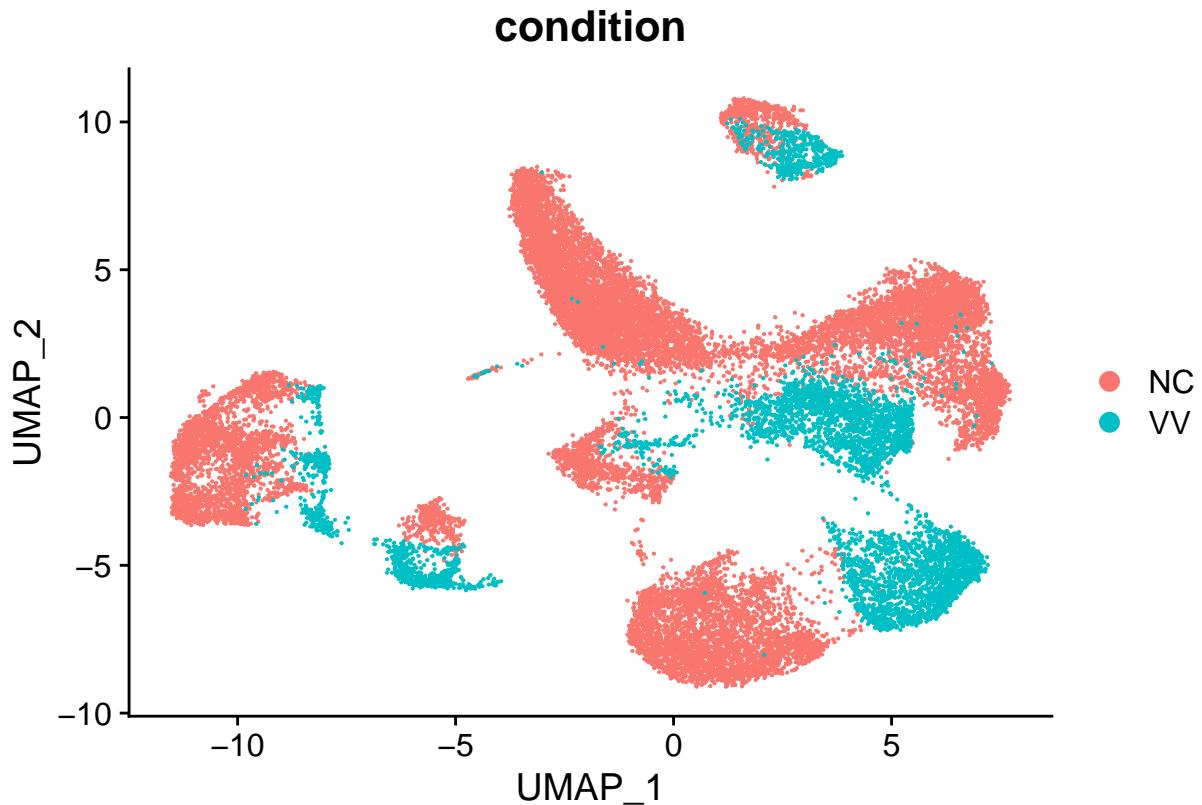
```



```
#merged_seurat@meta.data  
merged_seurat@meta.data$condition_1 <- merged_seurat@meta.data$condition
```

```
#merged_seurat@meta.data
```

```
before <- DimPlot(merged_seurat, reduction = 'umap', group.by = 'condition')  
before
```



```

CV.harmony <- merged_seurat %>%
  RunHarmony(group_by_vars = 'condition', plot_convergence = F)

## Harmony 1/10

## Harmony 2/10

## Harmony 3/10

## Harmony converged after 3 iterations

## Warning: Invalid name supplied, making object name syntactically valid. New
## object name is Seurat..ProjectDim.RNA.harmony; see ?make.names for more details
## on syntax validity

CV.harmony@meta.data <- unite(CV.harmony@meta.data, "condition_cluster", condition_1, seurat_clusters, )

#CV.harmony@meta.data

#compare_datasets <- FindMarkers(CV.harmony, ident.1 = 'NC_3', ident.2 = 'VV_3')
#head(compare_datasets)

```

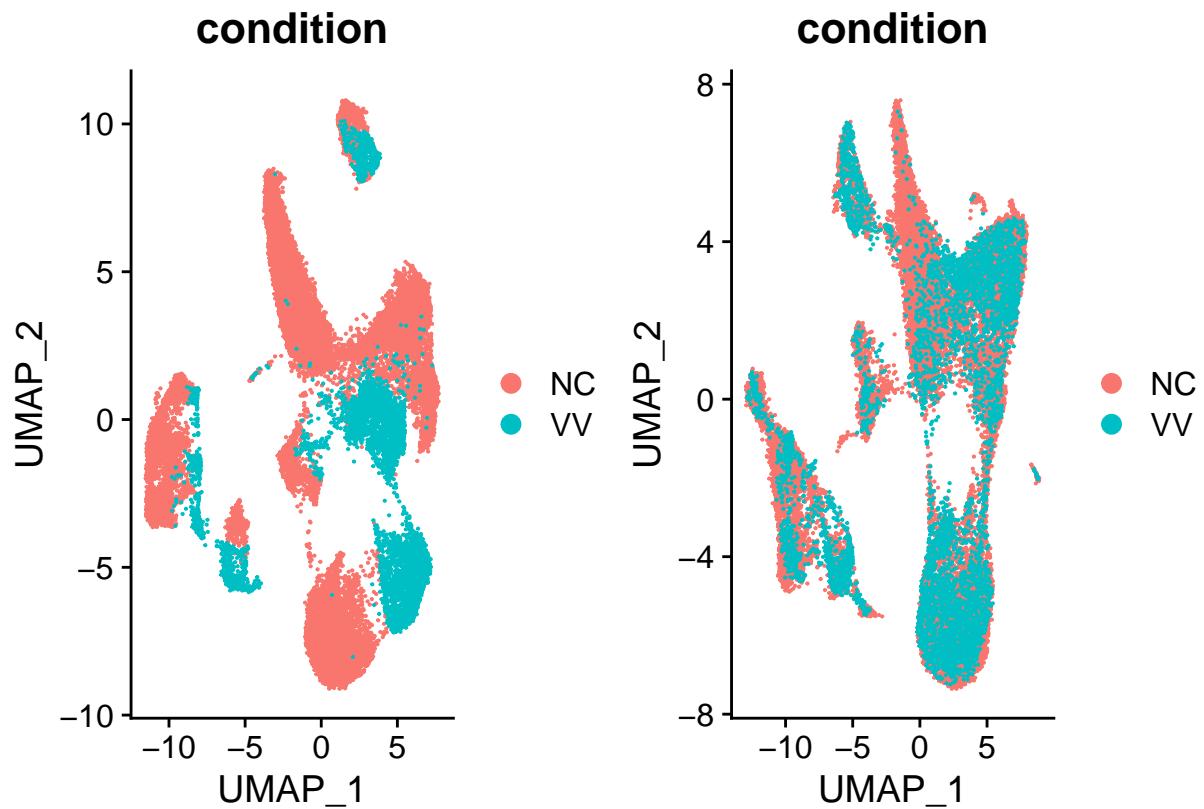
```
CV.harmony@reductions
```

```
## $pca
## A dimensional reduction object with key PC_
## Number of dimensions: 50
## Projected dimensional reduction calculated: FALSE
## Jackstraw run: FALSE
## Computed using assay: RNA
##
## $umap
## A dimensional reduction object with key UMAP_
## Number of dimensions: 2
## Projected dimensional reduction calculated: FALSE
## Jackstraw run: FALSE
## Computed using assay: RNA
##
## $harmony
## A dimensional reduction object with key harmony_
## Number of dimensions: 50
## Projected dimensional reduction calculated: TRUE
## Jackstraw run: FALSE
## Computed using assay: RNA
```

```
CV.harmony.embed <- Embeddings(CV.harmony, 'harmony')
CV.harmony.embed[1:10,1:10]
```

```
##                                     harmony_1   harmony_2   harmony_3   harmony_4   harmony_5
## NC_AAACCCACAAAGGGCT-1    0.4839524   2.862759  0.03109228  2.1669226 -4.4389233
## NC_AAACCCACAAGGAGTC-1   3.9281255   8.363760  4.91979620 -1.2951033 -0.7958067
## NC_AAACCCACAAGGTACG-1  -19.9054021  -1.242564  2.99977409  0.8918654  0.5950514
## NC_AAACCCACACCTCGTT-1   5.3013494   6.030749  6.03074749 -2.0536631  1.1980845
## NC_AAACCCACATCATCCC-1   5.2183506  -13.005694  2.43267337 -5.3182759 -10.8834659
## NC_AAACCCACATGCGGTC-1   1.7394328   3.909342 -3.59287702  2.9557919  0.1652221
## NC_AAACCCAGTAGGCAGT-1   7.0303968  -12.553329  0.59921043 -0.7694746  0.3558219
## NC_AAACCCAGTATATGGA-1  -13.0410851  -2.774753  3.36781720  0.1726799  2.9422314
## NC_AAACCCAGTCGCGGTT-1   6.6891994   9.212387  8.99612561 -4.0650930  1.7914426
## NC_AAACCCAGTGTTCGAT-1   2.9975849   2.837576 -0.71750541 -0.7769419 -2.7560187
##                                     harmony_6   harmony_7   harmony_8   harmony_9 harmony_10
## NC_AAACCCACAAAGGGCT-1  -0.1683862  1.6273861  0.7432752 -0.3236602 -0.5695922
## NC_AAACCCACAAGGAGTC-1  -0.4480959  1.8331491  0.6040126 -2.1958034  0.1431681
## NC_AAACCCACAAGGTACG-1   5.9479641   2.0720144 -3.2855690  1.3699383  0.9334324
## NC_AAACCCACACCTCGTT-1  -1.2710022  3.0930974 -1.6835841 -1.7365438 -0.7533242
## NC_AAACCCACATCATCCC-1  -1.7597746  0.9463009 -2.6956801 -5.9251363 10.7837267
## NC_AAACCCACATGCGGTC-1  -0.5852689  1.9523676 -1.1550953  0.6082552 -0.1779979
## NC_AAACCCAGTAGGCAGT-1  -1.0475638  4.5144771 -0.6799450 -1.2725906 -0.2244468
## NC_AAACCCAGTATATGGA-1   2.5632422 -11.7158317 -4.6816468 -3.7947900 -3.4318479
## NC_AAACCCAGTCGCGGTT-1  -1.0329019  3.5072905 -0.8869524 -1.3285815 -1.4083748
## NC_AAACCCAGTGTTCGAT-1   2.6566341 -3.6861606 -1.2233403 -0.9174170 -1.0173411
```

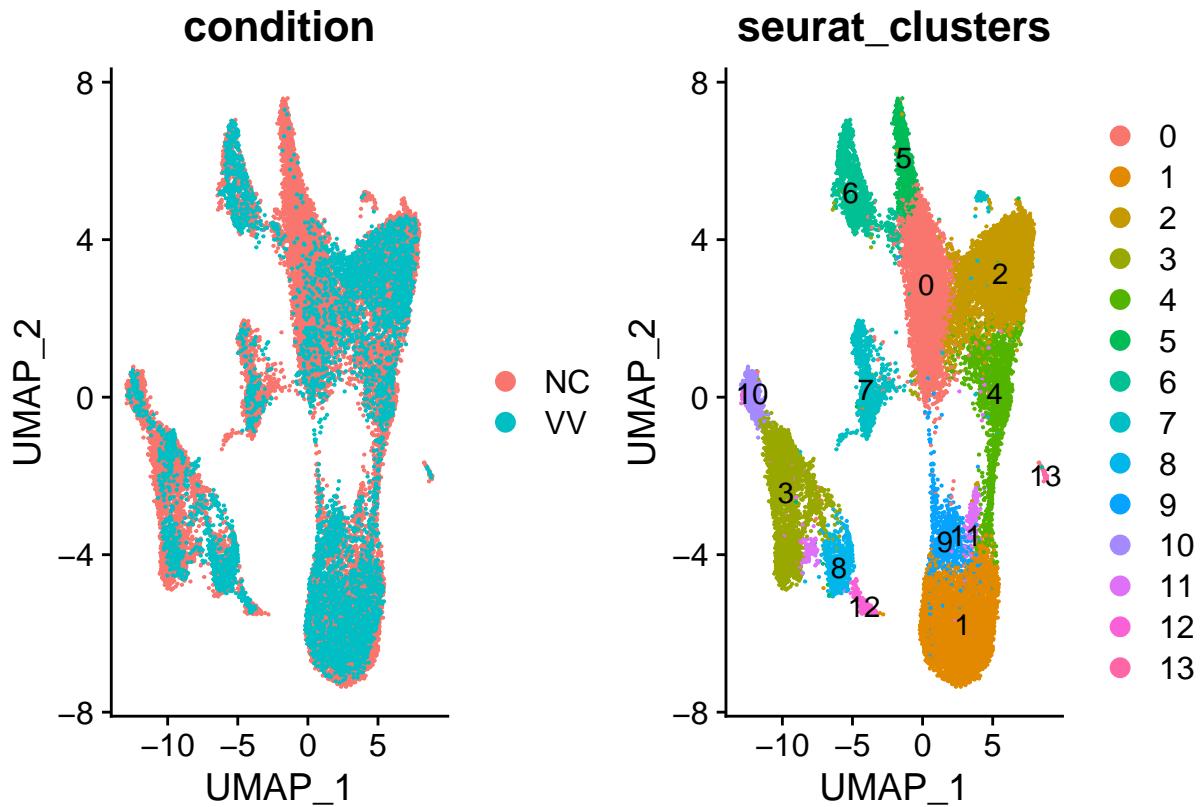
```
CV.harmony <- CV.harmony %>%
  RunUMAP(reduction = 'harmony', dims = 1:20) %>%
  FindNeighbors(reduction = 'harmony', dims = 1:20) %>%
  FindClusters(resolution = 0.5)
```

```

cluster <- DimPlot(CV.harmony, reduction = 'umap', group.by = 'seurat_clusters', label = T)
condition <- DimPlot(CV.harmony, reduction = 'umap', group.by = 'condition')
condition|cluster

```



```
Idents(CV.harmony) <- CV.harmony$condition_cluster
#Idents(CV.harmony)
```

```
CV.markers <- FindAllMarkers(CV.harmony,
  logfc.threshold = 0.25,
  min.pct = 0.1,
  only.pos = F)
```

```
## Calculating cluster NC_0
## Calculating cluster NC_2
## Calculating cluster NC_3
## Calculating cluster NC_5
## Calculating cluster NC_7
## Calculating cluster NC_1
## Calculating cluster NC_8
## Calculating cluster NC_6
```

```

## Calculating cluster NC_9

## Calculating cluster NC_10

## Calculating cluster NC_14

## Calculating cluster NC_12

## Calculating cluster NC_4

## Calculating cluster NC_11

## Calculating cluster NC_13

## Calculating cluster VV_4

## Calculating cluster VV_8

## Calculating cluster VV_5

## Calculating cluster VV_10

## Calculating cluster VV_13

## Calculating cluster VV_9

## Calculating cluster VV_6

## Calculating cluster VV_11

## Calculating cluster VV_0

## Calculating cluster VV_3

## Calculating cluster VV_2

## Calculating cluster VV_1

## Calculating cluster VV_14

## Calculating cluster VV_12

## Warning: The following tests were not performed:

## Warning: When testing NC_13 versus all:
## Cell group 1 has fewer than 3 cells

## Warning: When testing VV_12 versus all:
## Cell group 1 has fewer than 3 cells

```

```

#CV.markers
CV.markers %>%
  group_by(cluster) %>%
  slice_max(n = 2, order_by = avg_log2FC)

## # A tibble: 54 x 7
## # Groups:   cluster [27]
##      p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##      <dbl>     <dbl> <dbl> <dbl>    <dbl> <fct>  <chr>
## 1 0          1.78  0.498 0.303 0       NC_0   RBFOX1
## 2 3.27e-229 1.72  0.399 0.227 1.05e-224 NC_0   CNTNAP2
## 3 0          2.40  0.643 0.153 0       NC_2   AC022325.2
## 4 0          2.25  0.481 0.122 0       NC_2   SLC35F4
## 5 0          2.88  0.864 0.285 0       NC_3   MCTP1
## 6 0          2.85  0.671 0.076 0       NC_3   TPO
## 7 1.39e- 18 3.41  0.967 0.432 4.47e- 14 NC_5   DCN
## 8 1.81e- 29 3.27  0.333 0.024 5.85e- 25 NC_5   PRG4
## 9 0          1.74  0.776 0.23  0       NC_7   CNTNAP2
## 10 0         1.74  0.832 0.312 0       NC_7   RBFOX1
## # ... with 44 more rows

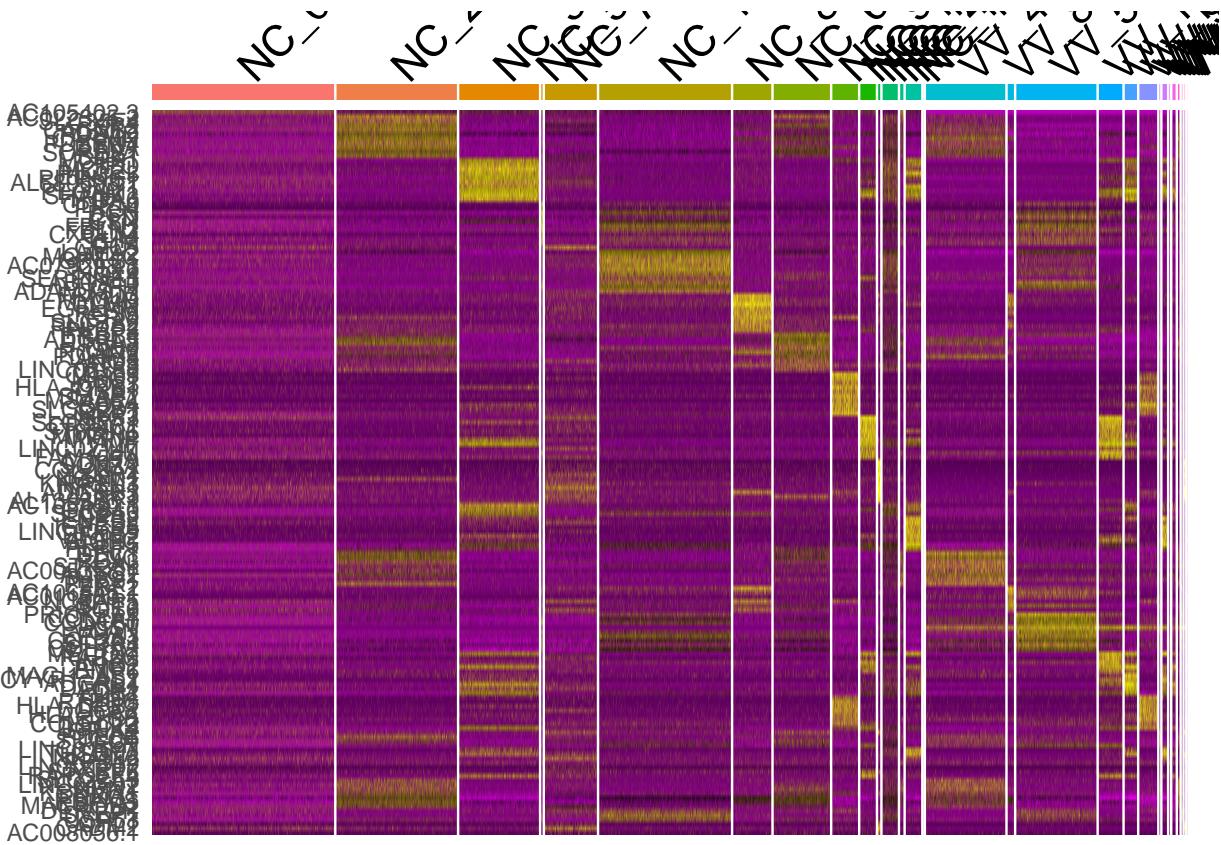
#CV.markers
CV.markers %>%
  group_by(cluster) %>%
  slice_min(n = 2, order_by = avg_log2FC)

## # A tibble: 54 x 7
## # Groups:   cluster [27]
##      p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##      <dbl>     <dbl> <dbl> <dbl>    <dbl> <fct>  <chr>
## 1 0          -1.82  1     1     0       NC_0   MALAT1
## 2 1.40e-224 -1.76  0.002 0.201 4.53e-220 NC_0   XIST
## 3 5.08e- 95 -1.82  0.156 0.322 1.64e- 90 NC_2   VWF
## 4 1.72e- 77 -1.75  0.206 0.35  5.56e- 73 NC_2   MCTP1
## 5 0          -2.60  0.5    0.761 0       NC_3   PRKG1
## 6 1.13e-102 -2.48  0.113 0.336 3.64e- 98 NC_3   DEC1
## 7 2.95e- 5   -2.45  0.267 0.63  9.52e-  1 NC_5   PDE4D
## 8 2.08e- 5   -2.34  0.067 0.476 6.72e-  1 NC_5   SLC8A1
## 9 0          -3.00  1     1     0       NC_7   MALAT1
## 10 0         -2.53  0.579 0.888 0       NC_7   NEAT1
## # ... with 44 more rows

CV.markers %>%
  group_by(cluster) %>%
  top_n(n = 10, wt = avg_log2FC) -> top10
DoHeatmap(CV.harmony, features = top10$gene) + NoLegend()

## Warning in DoHeatmap(CV.harmony, features = top10$gene): The following features
## were omitted as they were not found in the scale.data slot for the RNA assay:
## STARD13-AS, ITGB8, AAAS, ZBTB46, RTKN2, PTPN13, EGLN2, SYNPO2, PKP4, LAMA4,
## C16orf95, CCDC127, RBM22, SAP18, AC083864.5, MPZ, PIK3C2B, EDIL3, PDLIM5,
## LHFPL6, NCALD, MAGI1, DLC1, AC012409.2, COL18A1, NOTCH3, AC083870.1, DES, DSTN,
```

```
## CSRP1, TAGLN, C11orf96, ACTB, PPP1R14A, MYL9, FLNA, ARHGEF7, SGCG, PCDH15,  
## AC007402.1, LINC01320, CACNA1A, GALNTL6, LRP1B, CTNNA2, DPP10, CSMD1, CNTNAP2,  
## RBFOX1
```



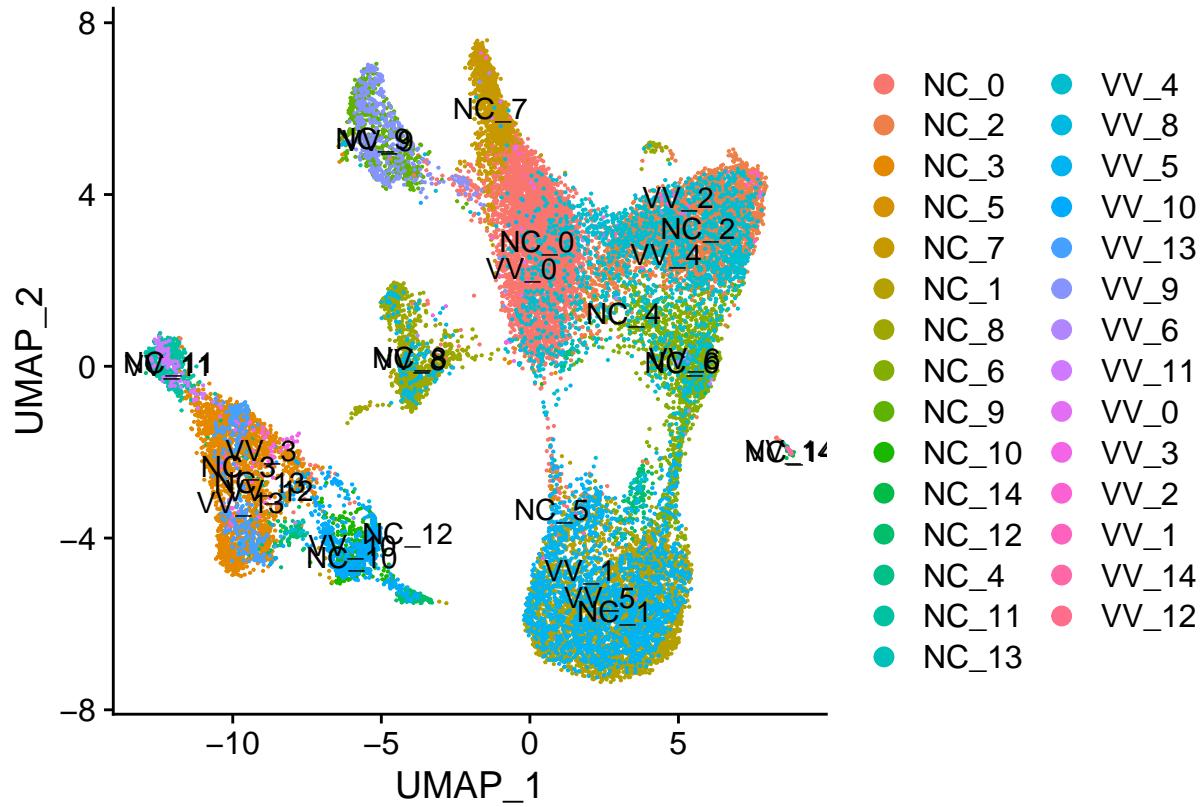
DefaultAssay(CV.harmony)

```
## [1] "RNA"
```

```
#head(markers cluster2)
```

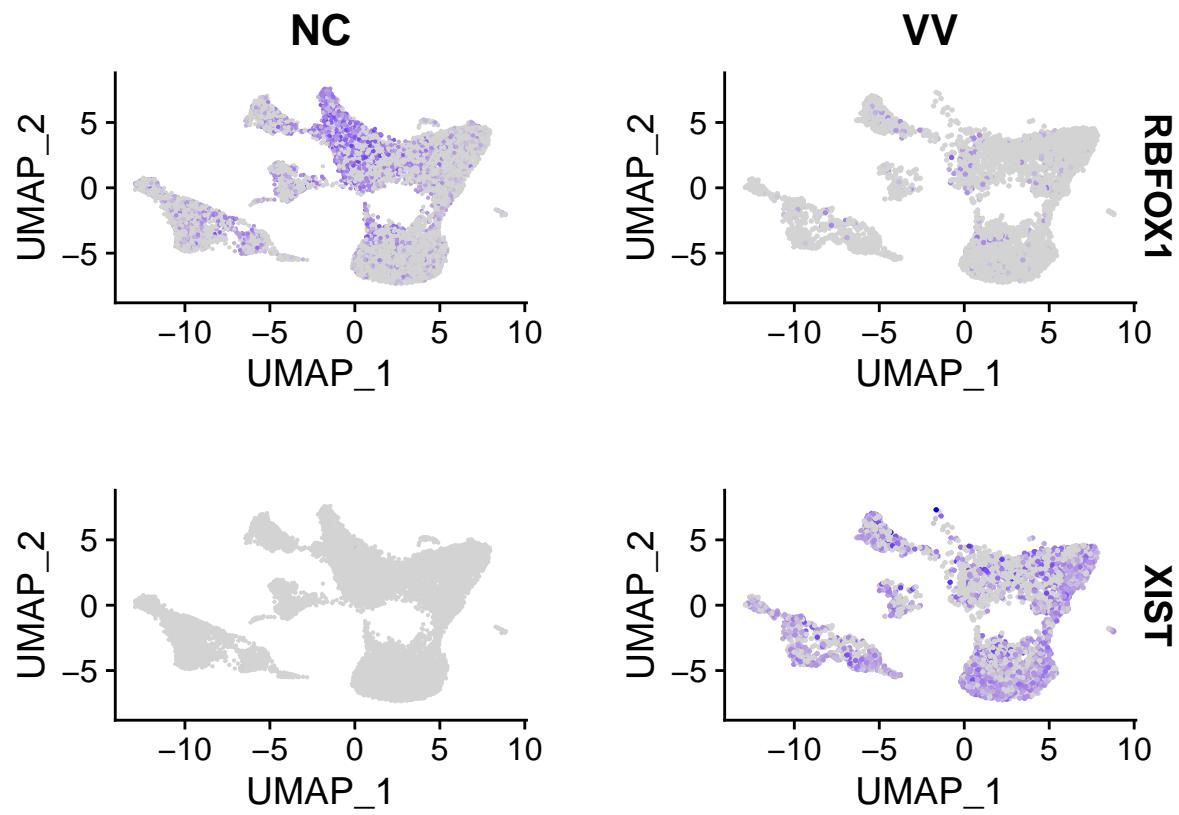
```
Idents(CV.harmony) <- CV.harmony@meta.data$condition_cluster  
#Idents(CV.harmony)
```

```
DimPlot(CV.harmony, reduction = 'umap', label = T)
```

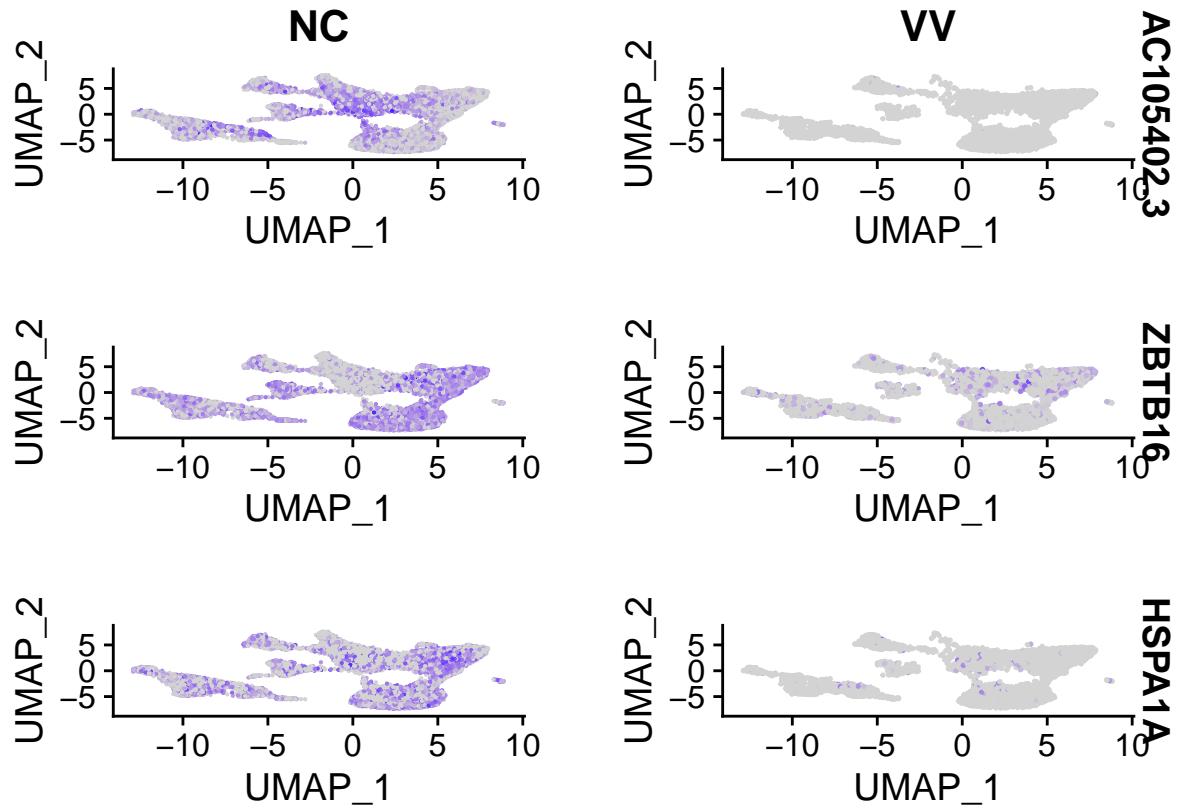


Genes differently express between 2 samples:

```
FeaturePlot(CV.harmony, features = c('RBFOX1','XIST'), split.by = 'condition', min.cutoff = 'q10')
```



```
FeaturePlot(CV.harmony, features = c('AC105402.3', 'ZBTB16', 'HSPA1A'), split.by = 'condition',
           min.cutoff = 'q10')
```



These clusters don't exist: NC_13, VV_12