

Butt et al: scRNASeq analysis - QC

Agnes A. Steixner-Kumar

12 11 2020

Contents

1 Libraries and load data	1
1.1 Load library	1
1.2 Load 10X data	1
2 Prepare Seurat objects	2
3 Calculate and plot basic features	2
4 Filter cells	9
5 Normalize data	10
6 Define variable genes	10
7 Scale data and regress out nUMI and percent.mito	12
8 Save objects	12

1 Libraries and load data

1.1 Load library

```
library(Seurat) #v2.3.0
```

1.2 Load 10X data

```
hypoxia_1<-Read10X("../10X_files/Hypoxia_1")
hypoxia_2<-Read10X("../10X_files/Hypoxia_2")
normoxia_1<-Read10X("../10X_files/Normoxia_1")
normoxia_2<-Read10X("../10X_files/Normoxia_2")
```

2 Prepare Seurat objects

```
# Paste sample identifier in front of cell ID
colnames(hypoxia_1) <- paste0('hyp1_', colnames(hypoxia_1))
colnames(hypoxia_2) <- paste0('hyp2_', colnames(hypoxia_2))
colnames(normoxia_1) <- paste0('nor1_', colnames(normoxia_1))
colnames(normoxia_2) <- paste0('nor2_', colnames(normoxia_2))

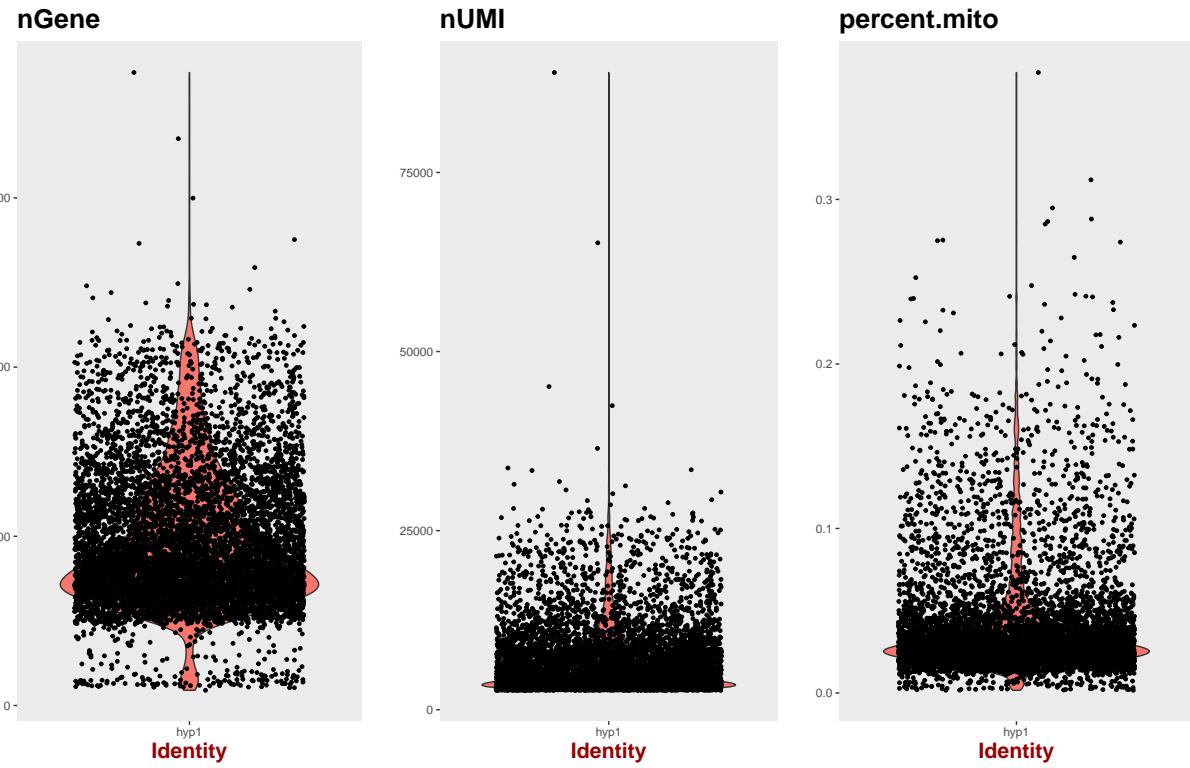
# create Seurat objects
hypoxia_1 <- CreateSeuratObject(hypoxia_1, min.cells=3, min.genes=200, project="hypoxia_1")
hypoxia_2 <- CreateSeuratObject(hypoxia_2, min.cells=3, min.genes=200, project="hypoxia_2")
normoxia_1 <- CreateSeuratObject(normoxia_1, min.cells=3, min.genes=200, project="normoxia_1")
normoxia_2 <- CreateSeuratObject(normoxia_2, min.cells=3, min.genes=200, project="normoxia_2")

# make list of Seurat objects for easier handling
obj <- list(hypoxia_1=hypoxia_1, hypoxia_2=hypoxia_2, normoxia_1=normoxia_1, normoxia_2=normoxia_2)
# create list of sample names
samnam <- c('hypoxia_1', 'hypoxia_2', 'normoxia_1', 'normoxia_2')
# set name for objects in list
names(obj) <- samnam
```

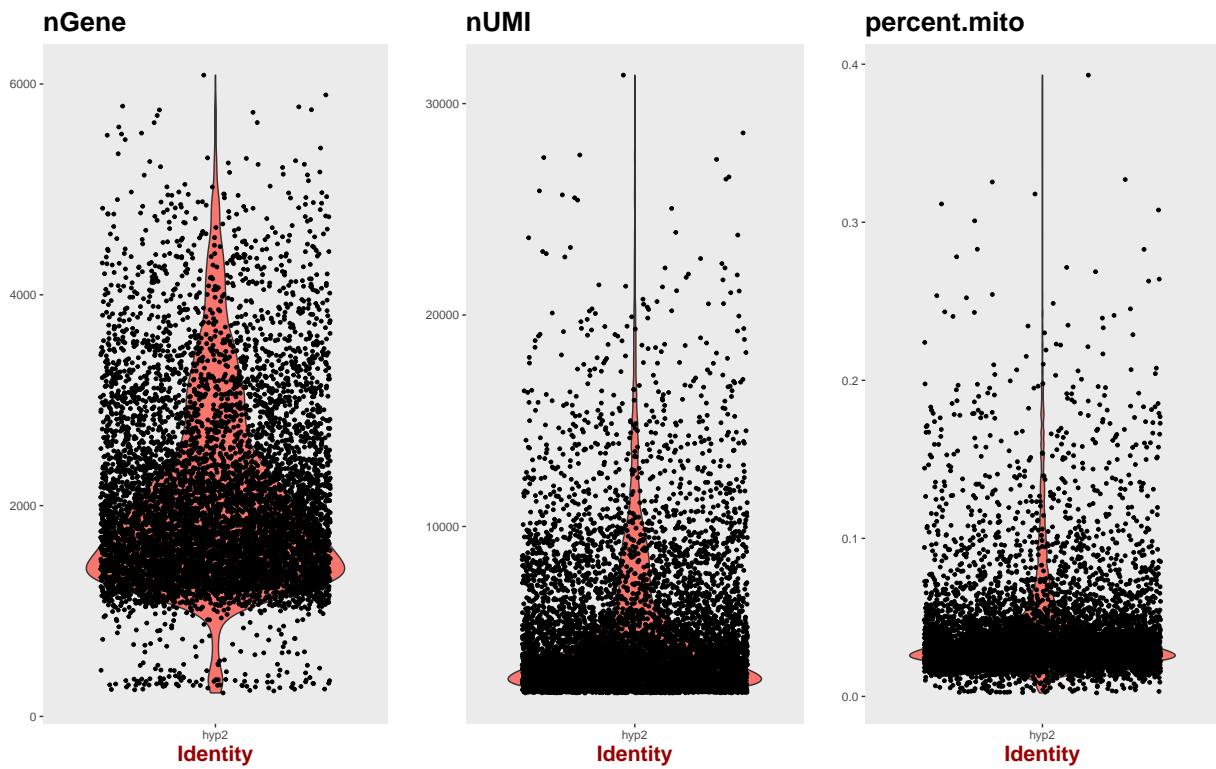
3 Calculate and plot basic features

```
# select all mitochondrial gene names (13 genes, same in all samples)
mito <- lapply(obj, function(x) grep("^mt", rownames(x@data), value=T))
# calculate percentage of mitochondrial gene per cell and sample
percent.mito <- lapply(obj, function(x) Matrix::colSums(x@raw.data[mito$hypoxia_1, ]) /
  Matrix::colSums(x@raw.data))
# add mito percentage to metadata
obj <- lapply(seq_along(samnam), function(x) AddMetaData(obj[[x]], metadata=percent.mito[[x]],
  col.name="percent.mito"))
# plot basic featuers nGene, nUMI and percent.mito
mitoplot <- lapply(obj, function(x) VlnPlot(x, features.plot=c("nGene", "nUMI", "percent.mito"),
  nCol = 3))
print(mitoplot)

## [[1]]
```

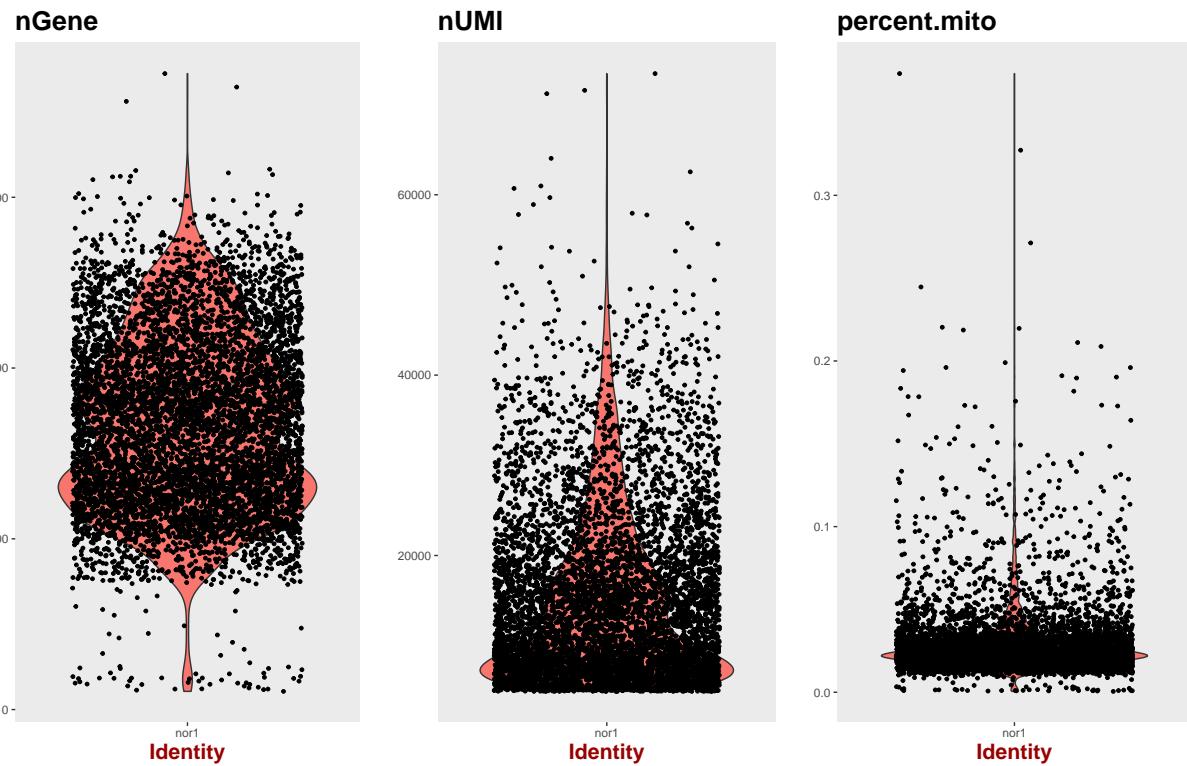


```
##  
## [[2]]
```

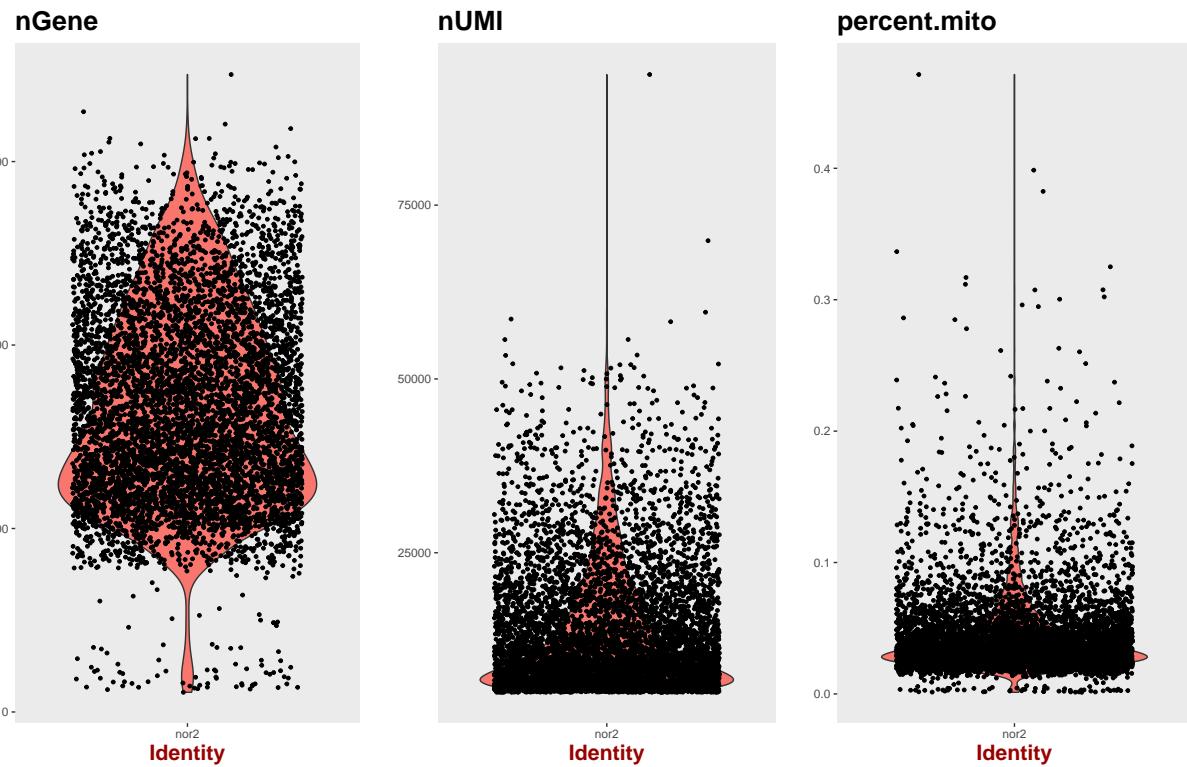


```
##
```

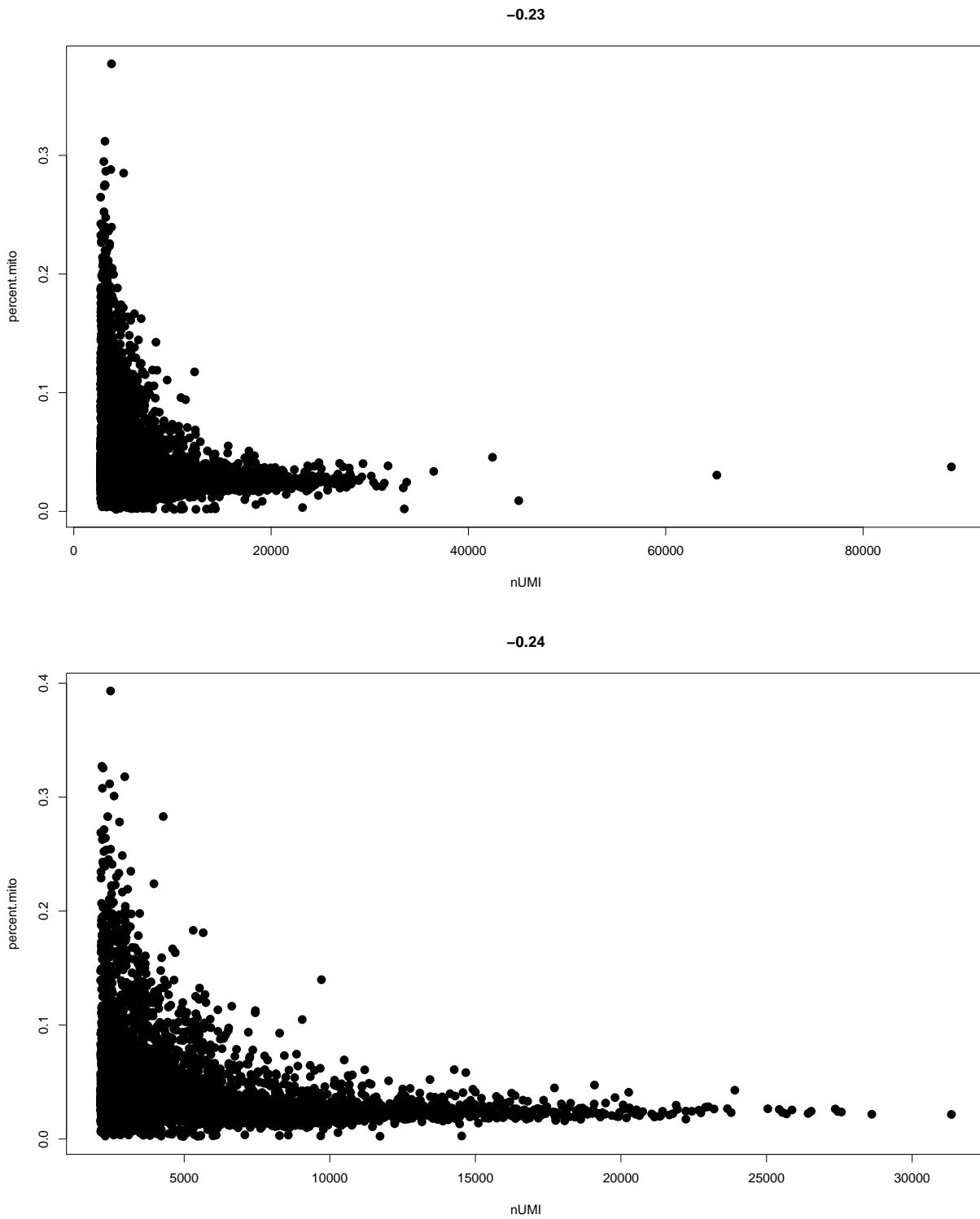
```
## [[3]]
```



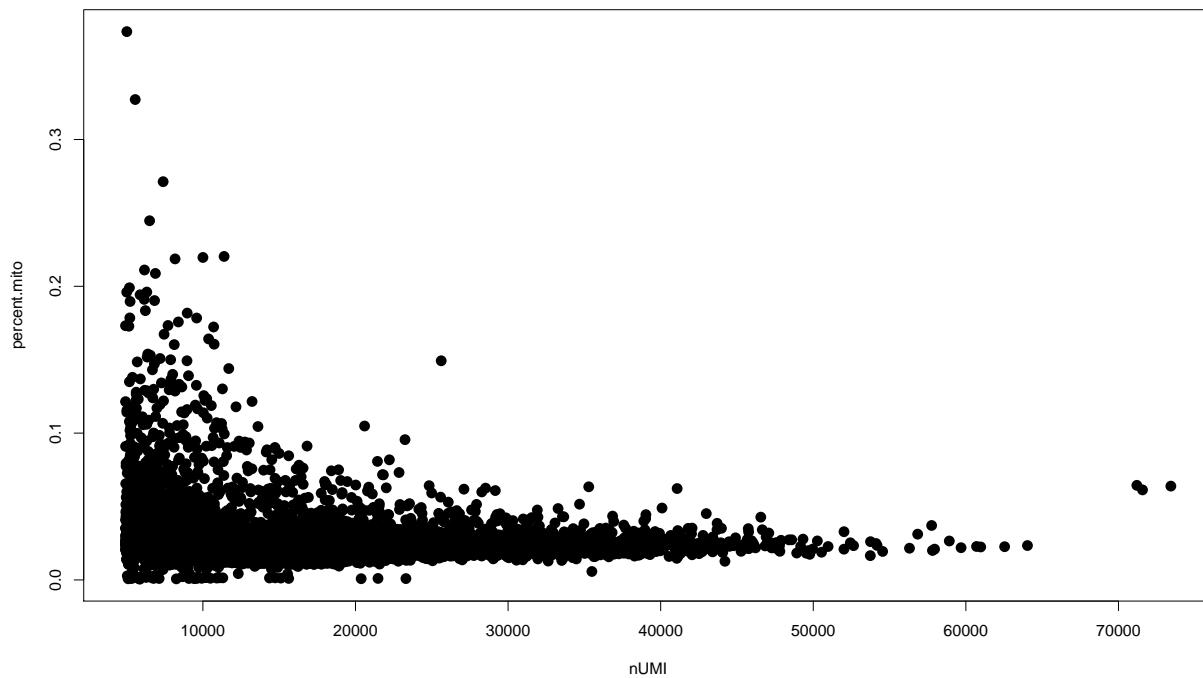
```
##  
## [[4]]
```



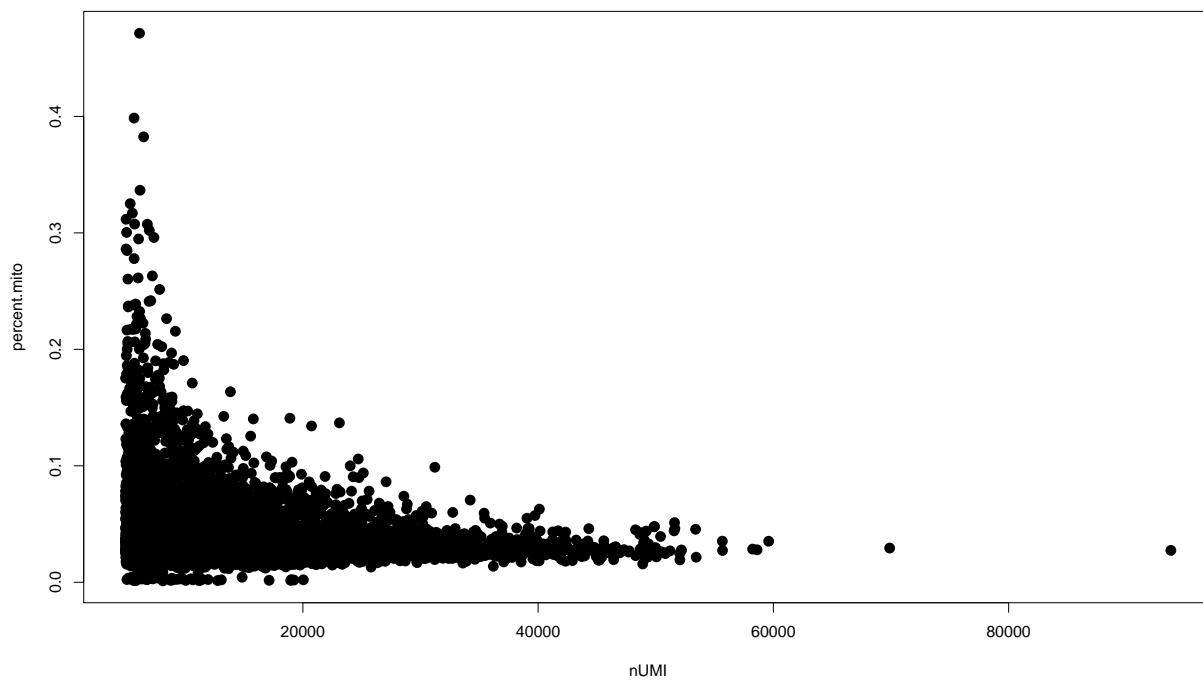
```
# plot nUMI by percent.mito/nGene to see distribution and decide on filtering cutoffs
lapply(obj, function(x) GenePlot(x, gene1 = "nUMI", gene2 = "percent.mito"))
```



-0.16

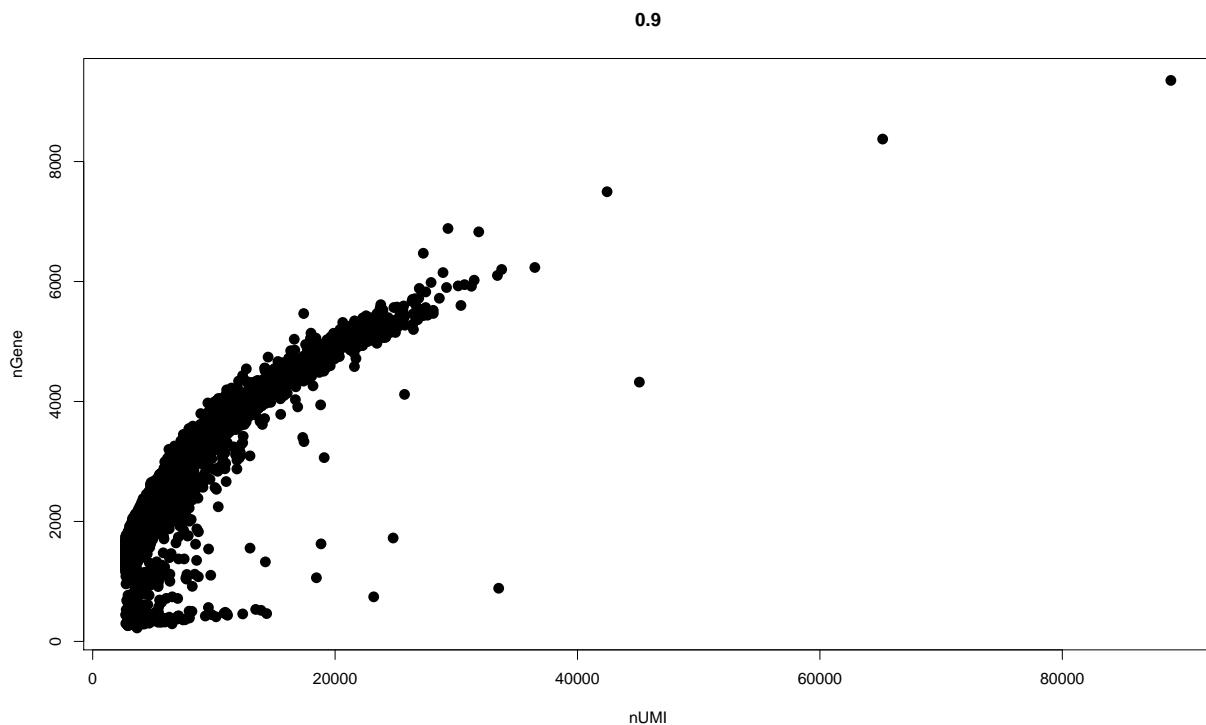


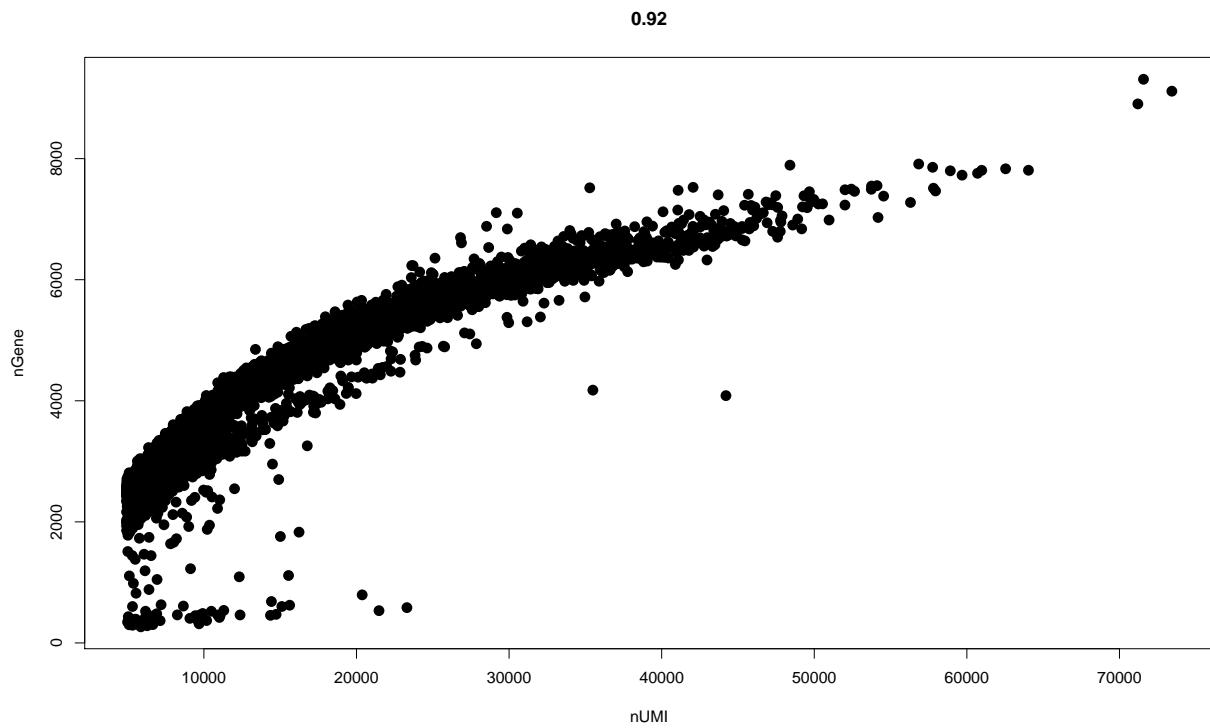
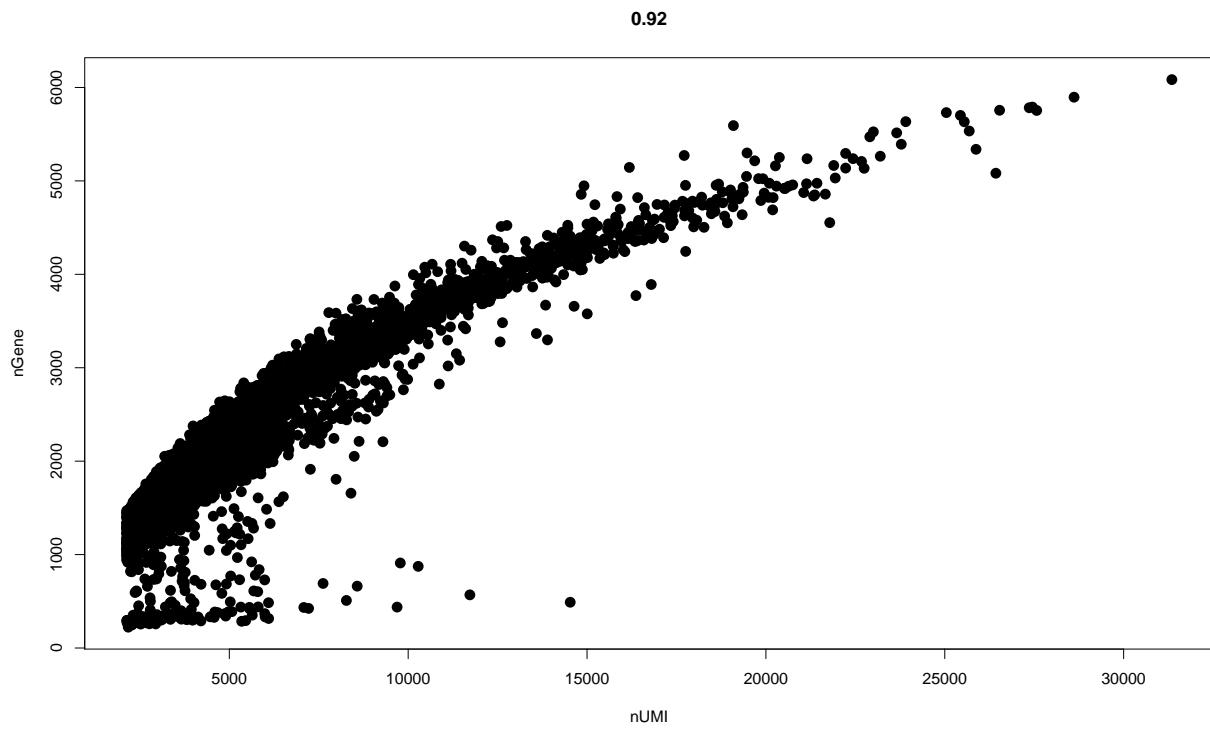
-0.22

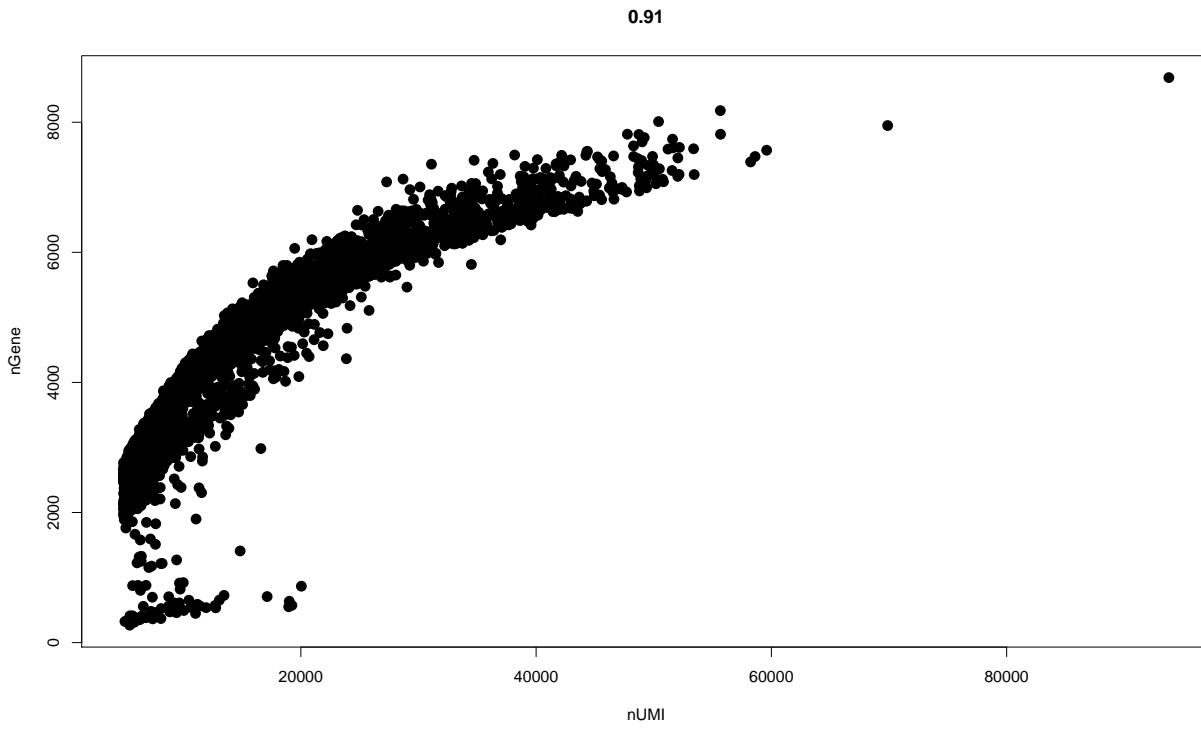


```
## [[1]]  
## NULL  
##  
## [[2]]  
## NULL
```

```
##  
## [[3]]  
## NULL  
##  
## [[4]]  
## NULL  
  
lapply(obj, function(x) GenePlot(x, gene1 = "nUMI", gene2 = "nGene"))
```







```
## [[1]]
## NULL
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
```

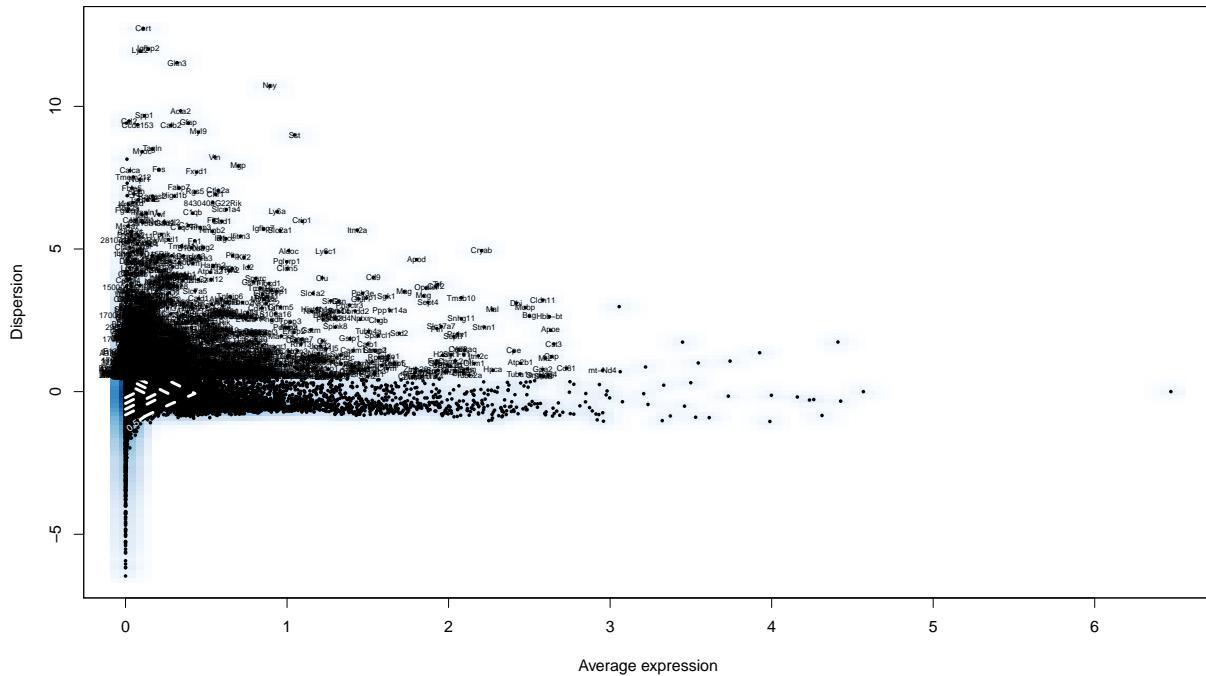
4 Filter cells

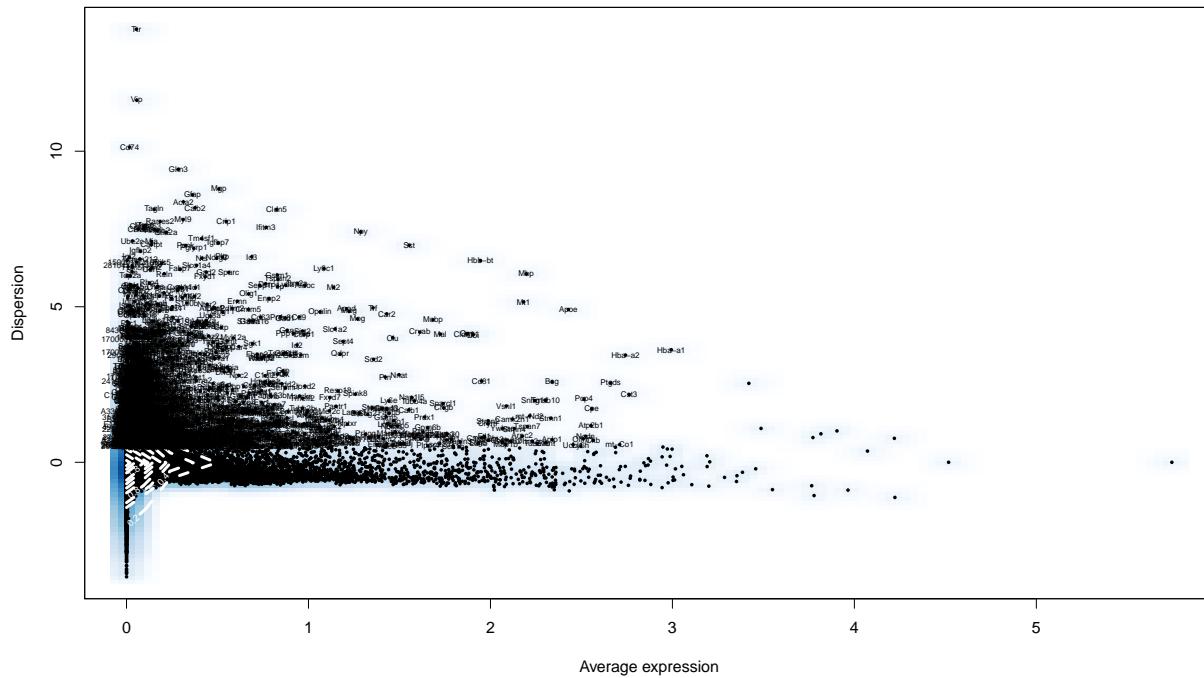
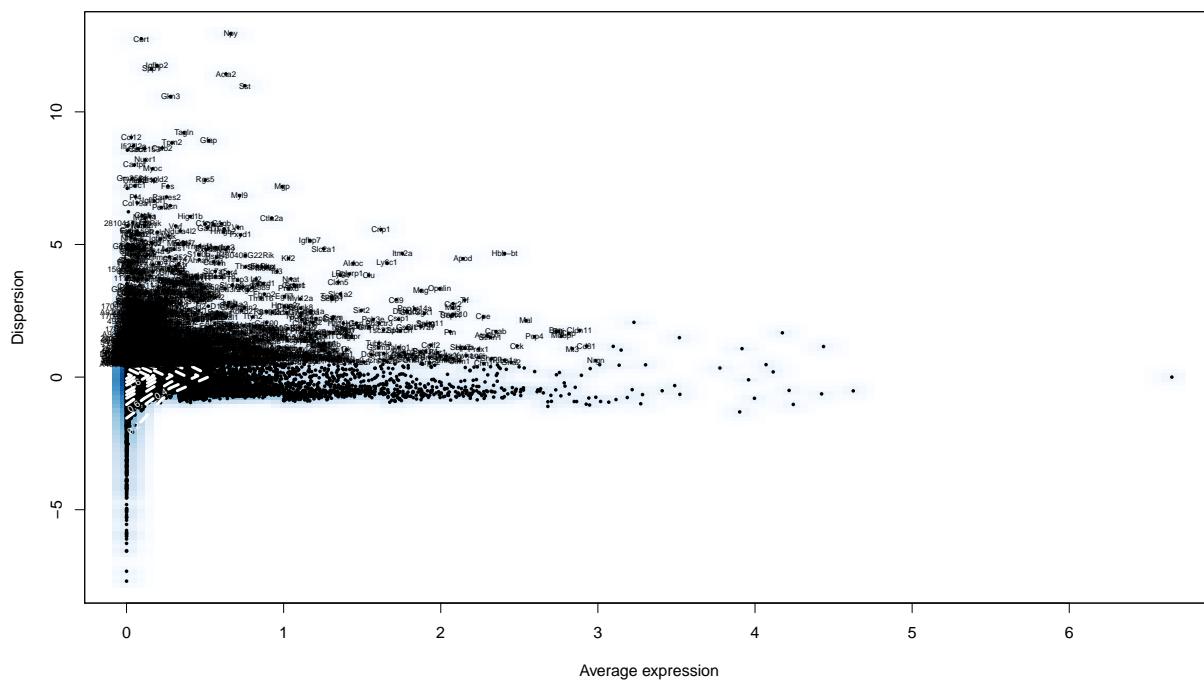
```
obj[[1]] <- FilterCells(obj[[1]], subset.names = c("nGene", "percent.mito"),
                         low.thresholds = c(500, -Inf), high.thresholds = c(6000, 0.2))
obj[[2]] <- FilterCells(obj[[2]], subset.names = c("nGene", "percent.mito"),
                         low.thresholds = c(500, -Inf), high.thresholds = c(6000, 0.2))
obj[[3]] <- FilterCells(obj[[3]], subset.names = c("nGene", "percent.mito"),
                         low.thresholds = c(500, -Inf), high.thresholds = c(7500, 0.2))
obj[[4]] <- FilterCells(obj[[4]], subset.names = c("nGene", "percent.mito"),
                         low.thresholds = c(500, -Inf), high.thresholds = c(7500, 0.2))
```

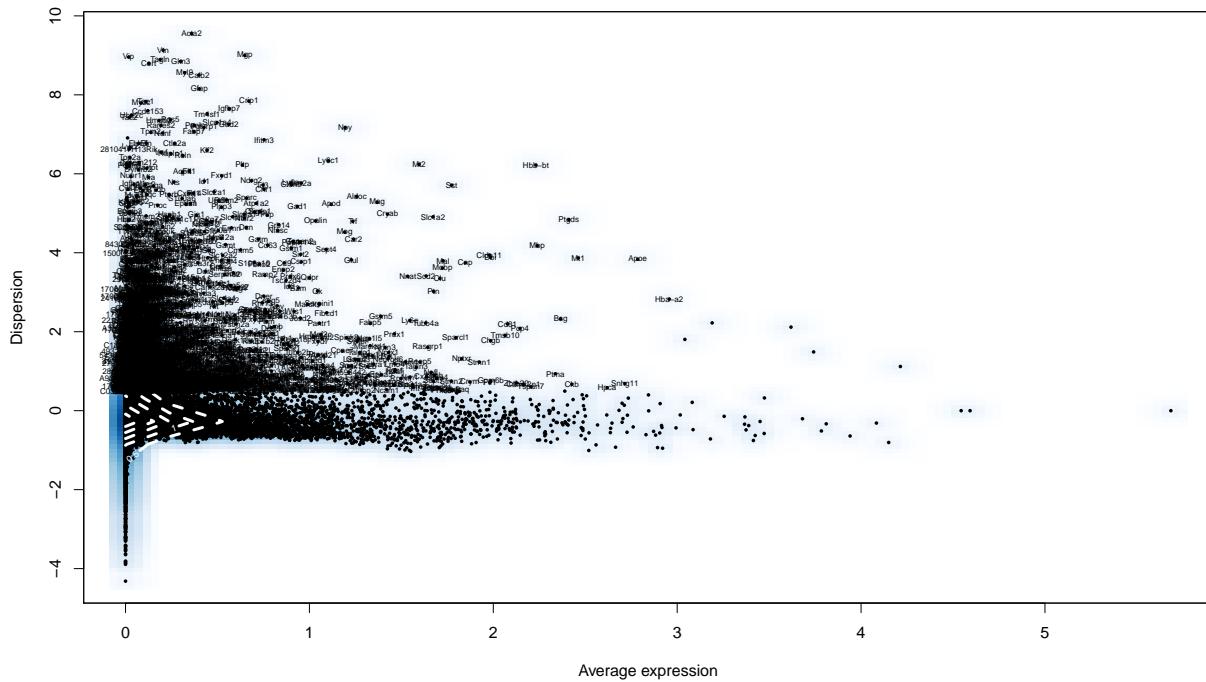
5 Normalize data

6 Define variable genes

```
obj<- lapply(obj, function(x) FindVariableGenes(object = x, mean.function = ExpMean,
                                                dispersion.function = LogVMR,
                                                x.low.cutoff = 0.0125, x.high.cutoff = 3, y.cutoff = 0.5))
```







7 Scale data and regress out nUMI and percent.mito

```
obj<- lapply(obj, function(x) ScaleData(object = x, vars.to.regress = c("nUMI", "percent.mito")))
```

8 Save objects

```
lapply(seq_along(samnam), function(x) saveRDS(obj[[x]], file=paste0(samnam[x], ".rds")))
```

sessionInfo()

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=German_Germany.1252  LC_CTYPE=German_Germany.1252
## [3] LC_MONETARY=German_Germany.1252 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.1252
##
## attached base packages:
```

```

## [1] stats      graphics   grDevices utils      datasets   methods    base
##
## other attached packages:
## [1] Seurat_2.3.0  Matrix_1.2-18 cowplot_1.0.0 ggplot2_3.2.1
##
## loaded via a namespace (and not attached):
##  [1] snow_0.4-3           backports_1.1.5     Hmisc_4.3-1
##  [4] VGAM_1.1-2          sn_1.5-5            plyr_1.8.5
##  [7] igraph_1.2.5         lazyeval_0.2.2     splines_3.5.2
## [10] TH.data_1.0-10      digest_0.6.24      foreach_1.4.8
## [13] htmltools_0.4.0      lars_1.2            gdata_2.18.0
## [16] magrittr_1.5         checkmate_2.0.0   cluster_2.1.0
## [19] mixtools_1.2.0       ROCR_1.0-7          recipes_0.1.9
## [22] gower_0.2.1         R.utils_2.9.2     sandwich_2.5-1
## [25] colorspace_1.4-1    xfun_0.12          dplyr_0.8.4
## [28] crayon_1.3.4        survival_3.1-8   zoo_1.8-7
## [31] iterators_1.0.12    ape_5.3            glue_1.3.1
## [34] gtable_0.3.0         ipred_0.9-9        kernlab_0.9-29
## [37] prabclus_2.3-2     BiocGenerics_0.28.0 DEoptimR_1.0-8
## [40] scales_1.1.0         mvtnorm_1.0-12    bibtex_0.4.2.2
## [43] Rcpp_1.0.3           metap_1.3          dtw_1.21-3
## [46] plotrix_3.7-7       htmlTable_1.13.3  tclust_1.4-1
## [49] foreign_0.8-75      proxy_0.4-23      mclust_5.4.5
## [52] SDMTools_1.1-221.1 Formula_1.2-3    tsne_0.1-3
## [55] stats4_3.5.2         lava_1.6.6         prodlim_2019.11.13
## [58] htmlwidgets_1.5.1    FNN_1.1.3          gplots_3.0.1.2
## [61] RColorBrewer_1.1-2  fpc_2.2-5          acepack_1.4.1
## [64] TFisher_0.2.0        modeltools_0.2-22  ica_1.0-2
## [67] farver_2.0.3         pkgconfig_2.0.3    R.methodsS3_1.8.0
## [70] flexmix_2.3-15      nnet_7.3-12        caret_6.0-85
## [73] labeling_0.3         tidyselect_1.0.0   rlang_0.4.4
## [76] reshape2_1.4.3       munsell_0.5.0     tools_3.5.2
## [79] generics_0.0.2        ranger_0.12.1    ggridges_0.5.2
## [82] evaluate_0.14        stringr_1.4.0     yaml_2.2.1
## [85] npsurv_0.4-0         ModelMetrics_1.2.2.1 knitr_1.28
## [88] fitdistrplus_1.0-14 robustbase_0.93-5  caTools_1.17.1.2
## [91] purrr_0.3.3          RANN_2.6.1          pbapply_1.4-3
## [94] nlme_3.1-137         formatR_1.7        R.oo_1.23.0
## [97] compiler_3.5.2       rstudioapi_0.11   png_0.1-7
## [100] lsei_1.2-0          tibble_2.1.3       stringi_1.4.6
## [103] lattice_0.20-40     vctrs_0.2.3        multtest_2.38.0
## [106] mutoss_0.1-12       diffusionMap_1.2.0 pillar_1.4.3
## [109] lifecycle_0.1.0      Rdpack_0.11-1     lmtest_0.9-37
## [112] data.table_1.12.8   bitops_1.0-6       irlba_2.3.3
## [115] gbRd_0.4-11         R6_2.4.1          latticeExtra_0.6-28
## [118] KernSmooth_2.23-16 gridExtra_2.3     codetools_0.2-16
## [121] MASS_7.3-51.5       gtools_3.8.1      assertthat_0.2.1
## [124] withr_2.1.2         mnormt_1.5-6     multcomp_1.4-12
## [127] diptest_0.75-7      parallel_3.5.2   doSNOW_1.0.18
## [130] grid_3.5.2           rpart_4.1-15     timeDate_3043.102
## [133] tidyverse_1.0.2       class_7.3-15     rmarkdown_2.1
## [136] segmented_1.1-0      Rtsne_0.15        pROC_1.16.1
## [139] numDeriv_2016.8-1.1 scatterplot3d_0.3-41 Biobase_2.42.0
## [142] lubridate_1.7.4      base64enc_0.1-3

```