Figure 2

The following code reproduces the Figure 2 in our EmptyNN manuscript.

Please download datasets and seurat objects before running this analysis (run download_datasets.sh in terminal)

Load libraries

Load (1) raw data

(2) filtering results of four cell-calling algorithms: EmptyNN (nn.res), CellRanger 2.0 (ranger.keep), CellRanger 3.0 (ranger3.keep), EmptyDrops (e.out, e.keep), CellBender (bender.keep)

```
load("../data/cell_hashing_raw.RData")
load("../data/cell_hashing_results.RData")
```

Figure 2A

```
total_counts <- Matrix::rowSums(Stoeckius.counts)
aframe <- data.frame(total_counts, rank = rank(-total_counts),
                     emptynn = nn.res$nn.keep)
twohundred <- unique(aframe$rank[which(aframe$total_counts == 200)])
fifty <- unique(aframe$rank[which(aframe$total_counts == 50)])
recovered <- aframe$rank[which(aframe$emptynn
                               & aframe$total_counts < 200
                               & aframe$total_counts > 50)]
ggplot(aframe, aes(rank, total_counts)) +
  scale_x_continuous(trans='log10') + scale_y_continuous(trans='log10') +
  geom_point() + theme_bw() +
  xlab("Sorted barcodes") + ylab("Total counts [log10]") +
  geom_vline(xintercept = twohundred, color = "dodgerblue", linetype = "dashed") +geom_vline(xintercept
```

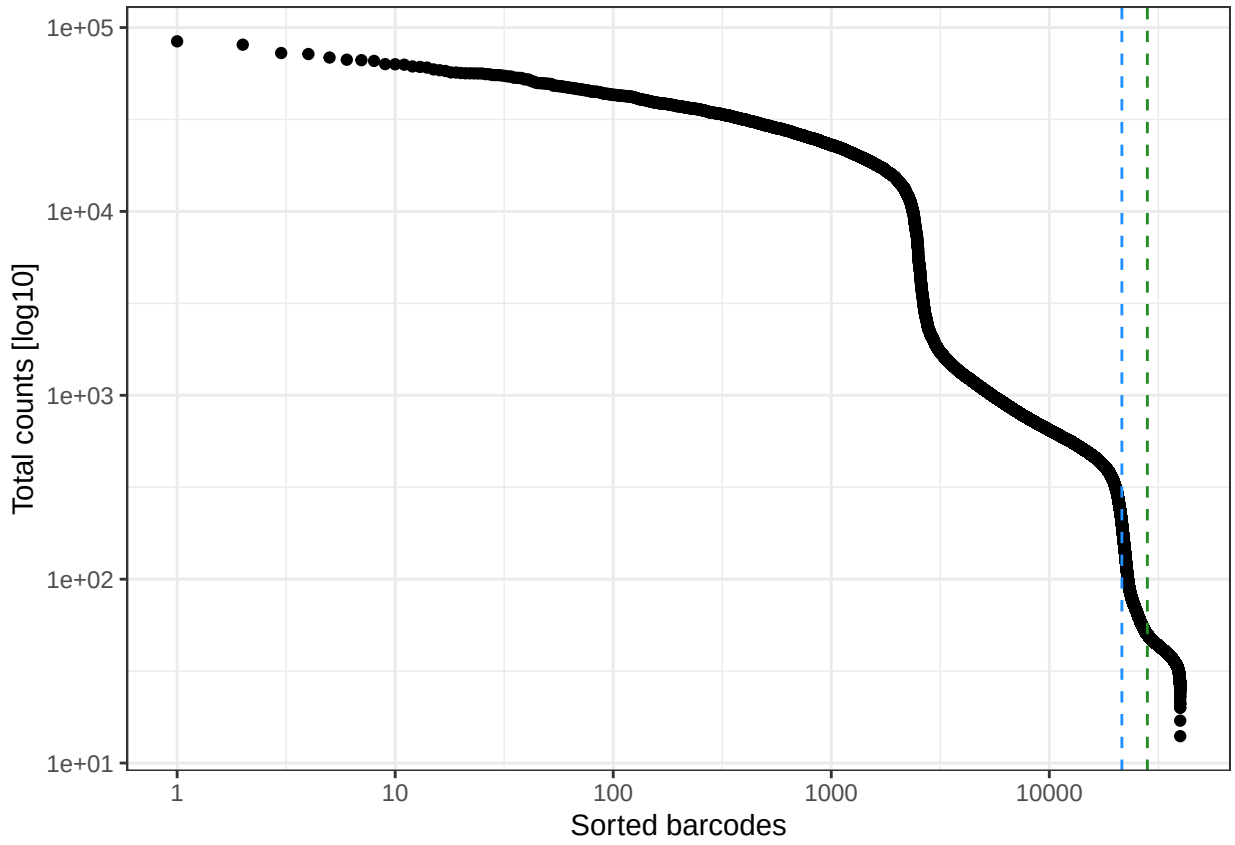


Figure 2B

```
ggplot(aframe[which(aframe$total_counts < 210 & aframe$total_counts > 40),], aes(rank,total_counts)) +
  xlab("Sorted barcodes") + ylab("Total counts") +
  geom_vline(xintercept = twohundred, color = "dodgerblue", linetype = "dashed") +
  geom_vline(xintercept = fifty, color = "forestgreen", linetype = "dashed") +
  geom_vline(xintercept = recovered, color = "darkred", alpha = 0.5) +
  geom_point() + theme_bw()
```

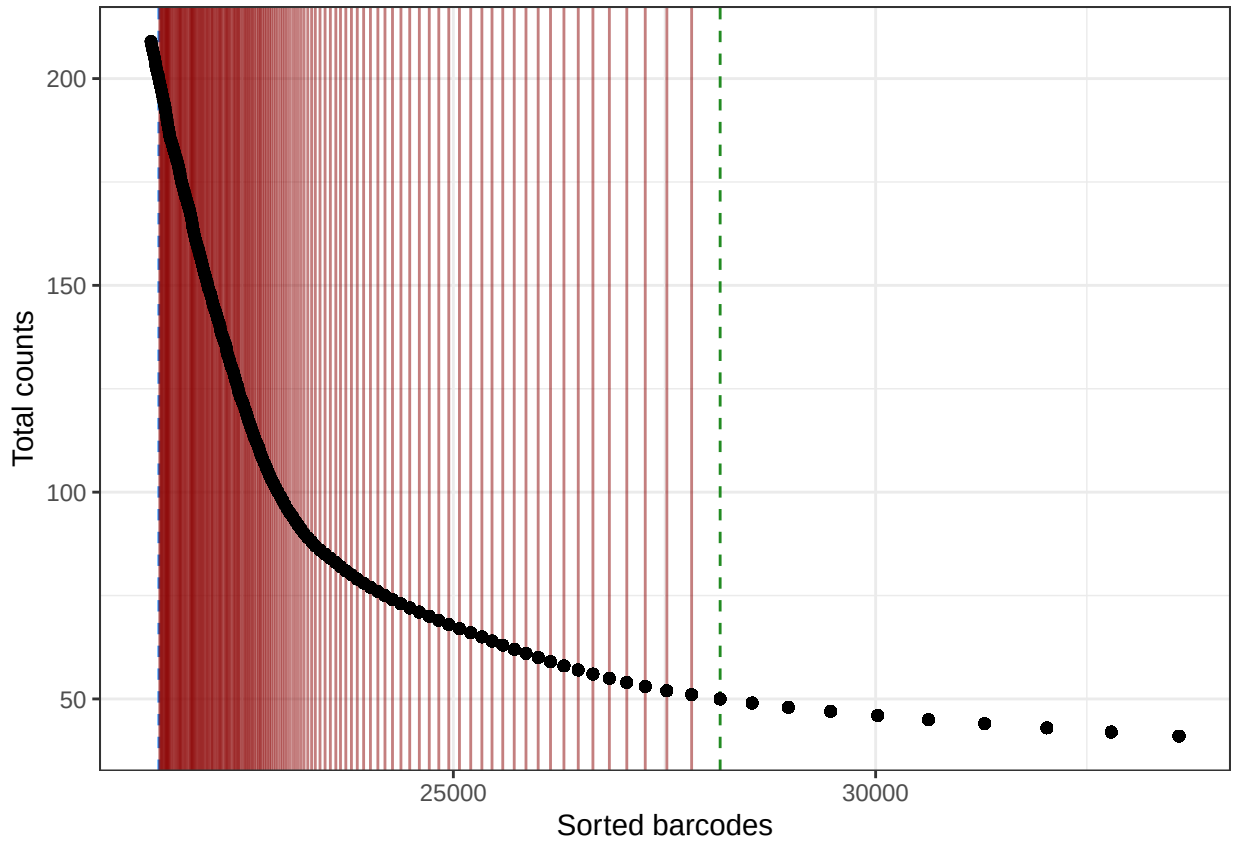


Figure 2C

```
names(nn.res$nn.keep) <- rownames(Stoeckius.counts)
recover.bcs <- intersect(names(total_counts[total_counts<200]),
                         names(nn.res$nn.keep[nn.res$nn.keep]))
recover <- CreateSeuratObject(counts = t(Stoeckius.counts[recover.bcs,]))
```

```
## Warning: The following arguments are not used: row.names
```

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

```
recover <- NormalizeData(recover,verbose = FALSE)
original_paper <- readRDS("./../data/cell_hashing_original_paper.rds")
genes.use <- VariableFeatures(original_paper)
recover <- ScaleData(recover, features = genes.use,verbose = FALSE)
sub.genes.use <- rownames(recover[["RNA"]]@scale.data)
recover <- RunPCA(recover,features=sub.genes.use,verbose = FALSE)
recover <- FindNeighbors(recover, dims = 1:10,verbose = FALSE)
recover <- FindClusters(recover, resolution = 0.3,verbose = FALSE)
recover <- RunTSNE(recover, dims = 1:10, check_duplicates = FALSE,verbose = FALSE)
new.cluster <- c("CD14 Mono","CD4","NK","B",'Platelet')
```

```
names(new.cluster) <- levels(recover)
recover <- RenameIdents(recover,new.cluster)
DimPlot(recover,label=T)+NoLegend()+labs(title="Cell type identities of recovered cells")
```

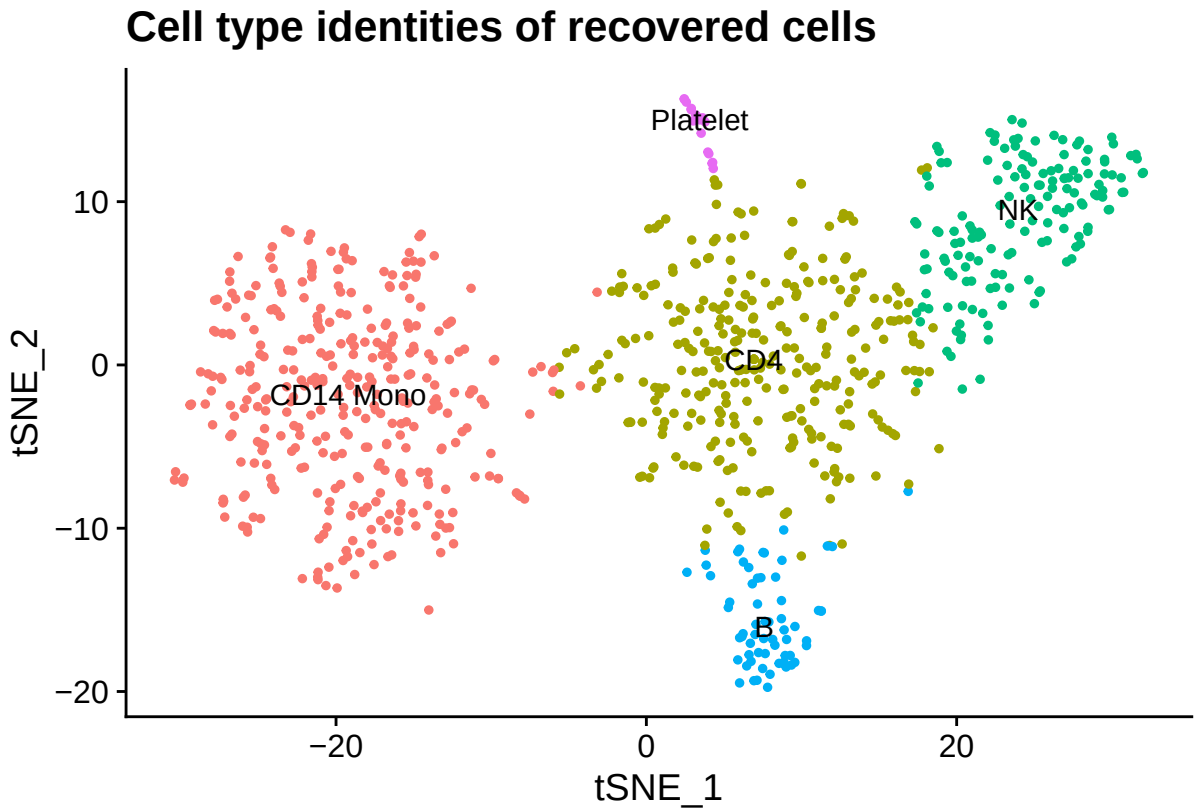


Figure 2D

```
des <- FindAllMarkers(recover, only.pos = TRUE, min.pct = 0.25,
                      logfc.threshold = 0.25,verbose=FALSE)
asplit_genes <- split(1:nrow(des), des$cluster)
genes <- unlist(lapply(asplit_genes, function(x) des[x[1:5], "gene"]))
genes <- genes[genes %in% rownames(recover@assays$RNA@data)]
# Average cells within each cluster
asplit_cells <- split(rownames(recover@meta.data), recover@active.ident)
means <- do.call(cbind, lapply(asplit_cells, function(x){
  s1 <- Matrix::rowMeans(recover@assays$RNA@data[genes, sample(unlist(x), 10)])
  s2 <- Matrix::rowMeans(recover@assays$RNA@data[genes, sample(unlist(x), 10)])
  s3 <- Matrix::rowMeans(recover@assays$RNA@data[genes, sample(unlist(x), 10)])
  cbind(s1, s2, s3)
}))
cell_type <- unlist(lapply(names(asplit_cells), function(x) rep(x, 3)))
# Create heatmap (sample 3 "replicates")
anno_col <- data.frame(cell_type)
```

```
rownames(anno_col) <- colnames(means) <- paste(colnames(means), cell_type)
pheatmap(means,cluster_rows = F, cluster_cols = F, scale = "row",show_rownames = F,
         breaks = seq(-2, 2, length = length(yellow2blue) + 1), col = yellow2blue,
         annotation_col = anno_col,show_colnames = F)
```

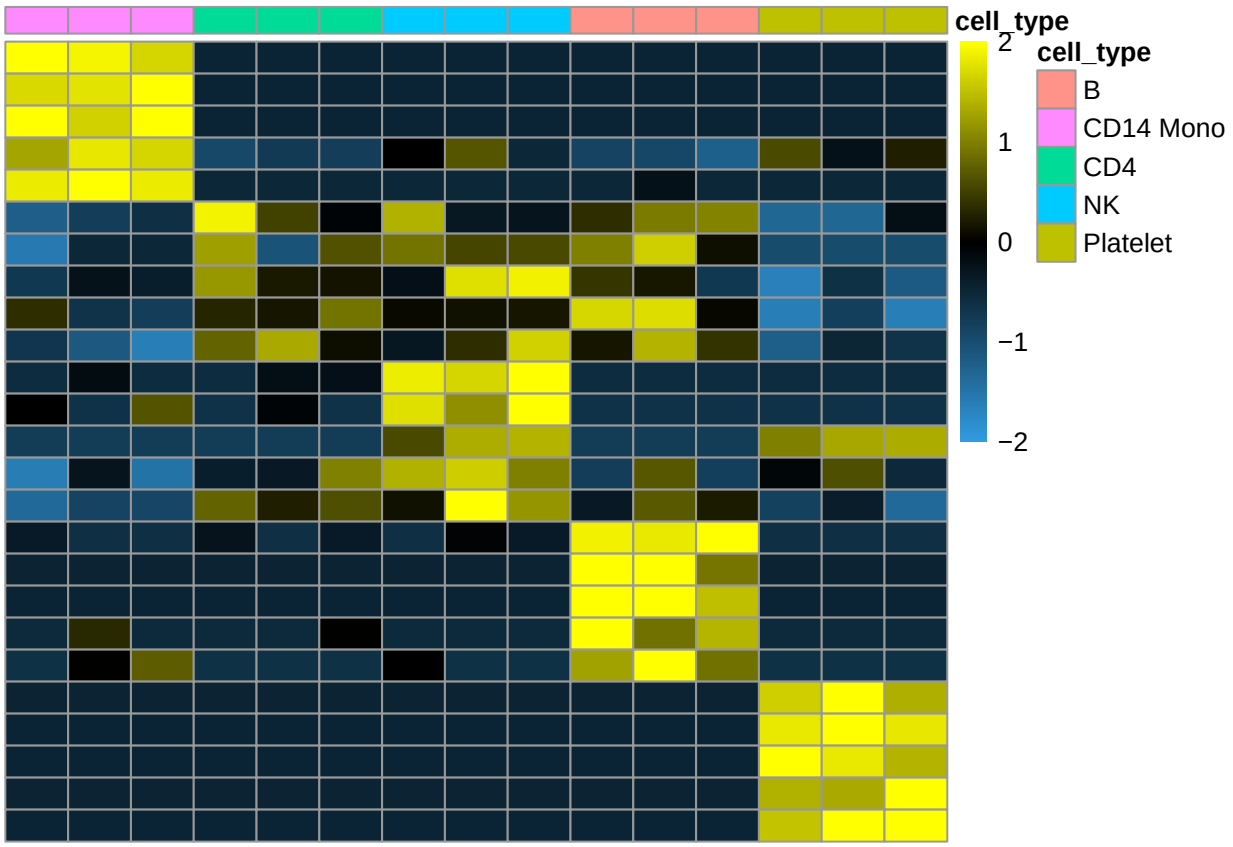


Figure 2E

```
pbmc <- readRDS("./../data/reference_pbmc_3k.rds")
pbmc$celltype <- as.character(Idents(pbmc))
pbmc$celltype[pbmc$celltype  %in% c("Naive CD4 T","Memory CD4 T")] <- "CD4"
pbmc$celltype[pbmc$celltype=="CD14+ Mono"] <- "CD14 Mono"
lst <- unique(as.character(Idents(recover)))
df <- do.call(rbind,lapply(lst,function(x){
  tmp <- pbmc[,pbmc$celltype==x]
  tmp.recover <- subset(recover,idents=x)
  c1 <- as.matrix(tmp.recover[['RNA']]@counts)
  c2 <- as.matrix(tmp[['RNA']]@counts)
  features <- intersect(rownames(c1),rownames(c2))
  m1 <- rowMeans(c1[features,])
  m2 <- rowMeans(c2[features,])
  cor(m1,m2)
}))
```

```
df <- data.frame(df)
colnames(df) <- "coefficient"
df$celltype <- lst
ggplot(df,aes(celltype,coefficient,fill=celltype))+
  geom_bar(stat="identity")+ylab("Correlation coefficient")+
  theme_bw()
```

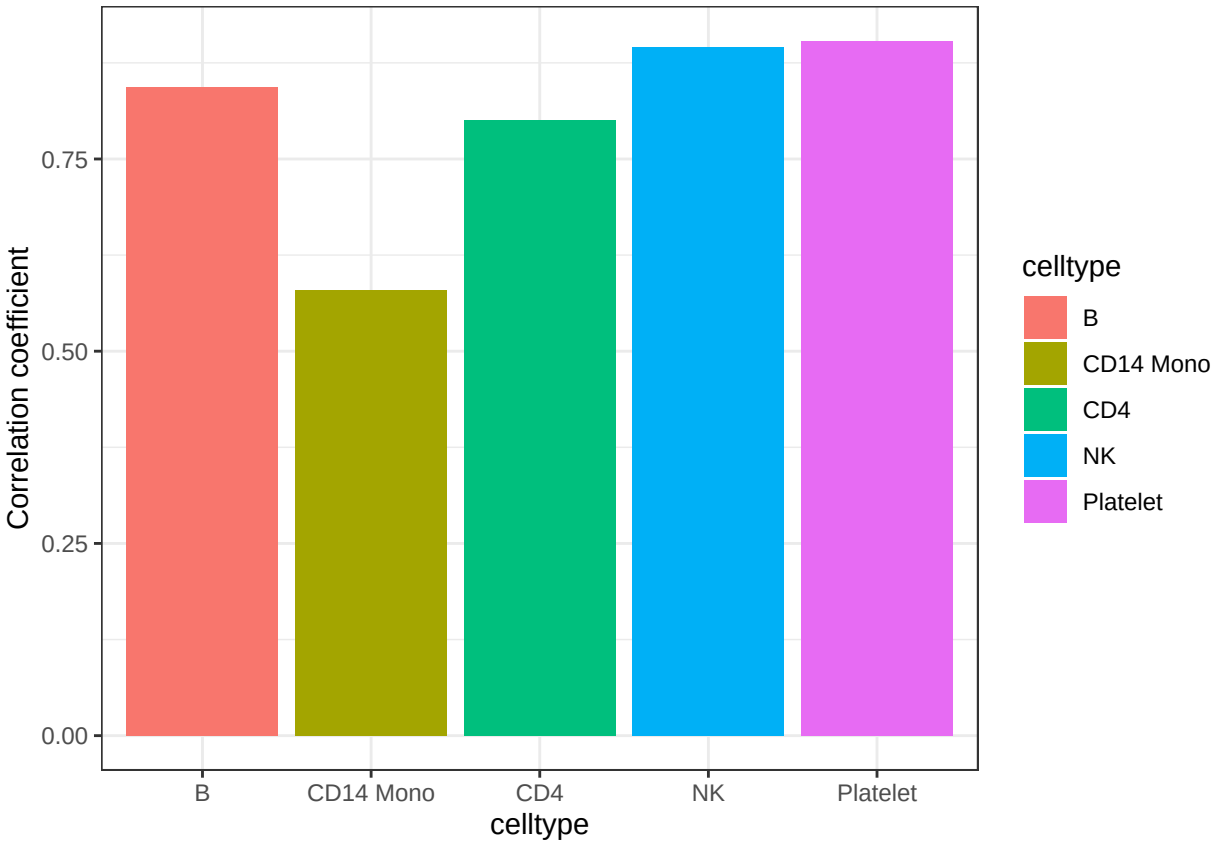
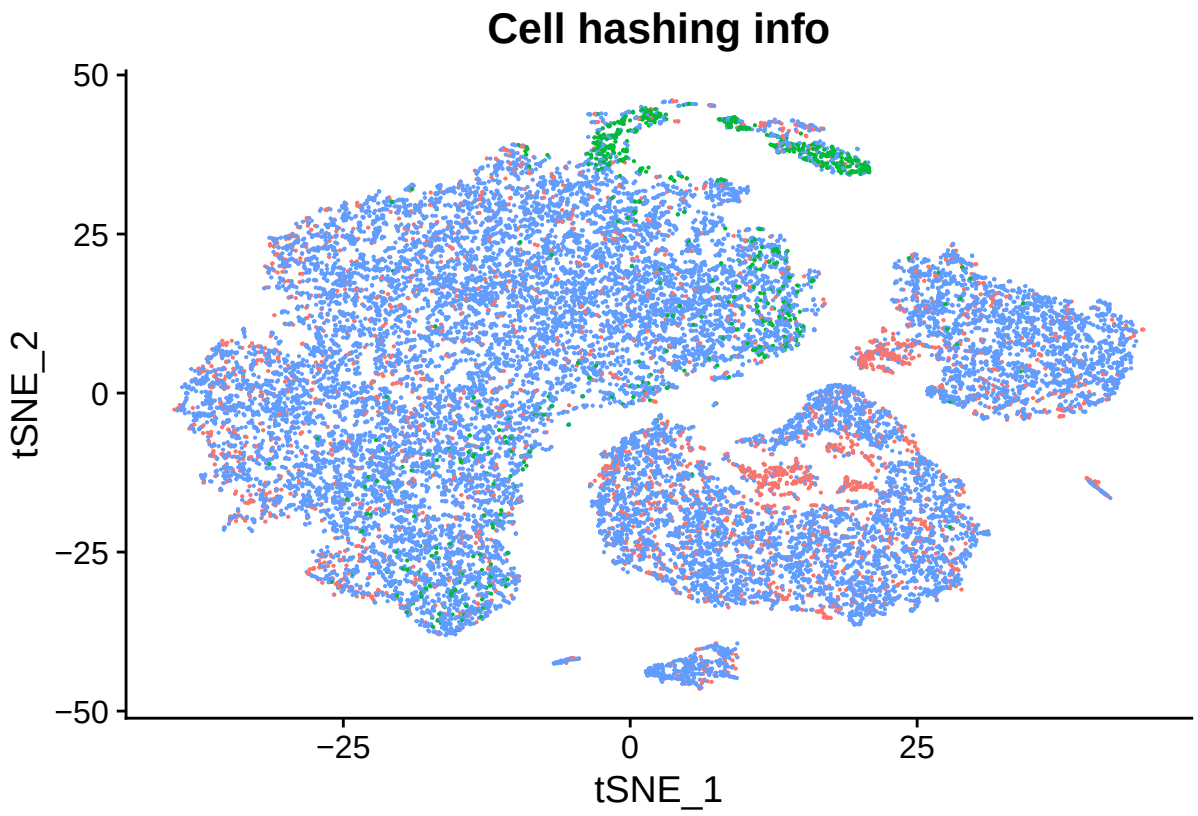


Figure S3A

```
retained <- readRDS("./../data/cell_hashing_retained.rds")
retained$label <- label[colnames(retained),1]
DimPlot(retained,reduction='tsne',group.by = 'label')+
  NoLegend()+labs(title="Cell hashing info")
```




```
DimPlot(retained,label=T,reduction='tsne')+NoLegend()+labs(title="Cell type identity")
```

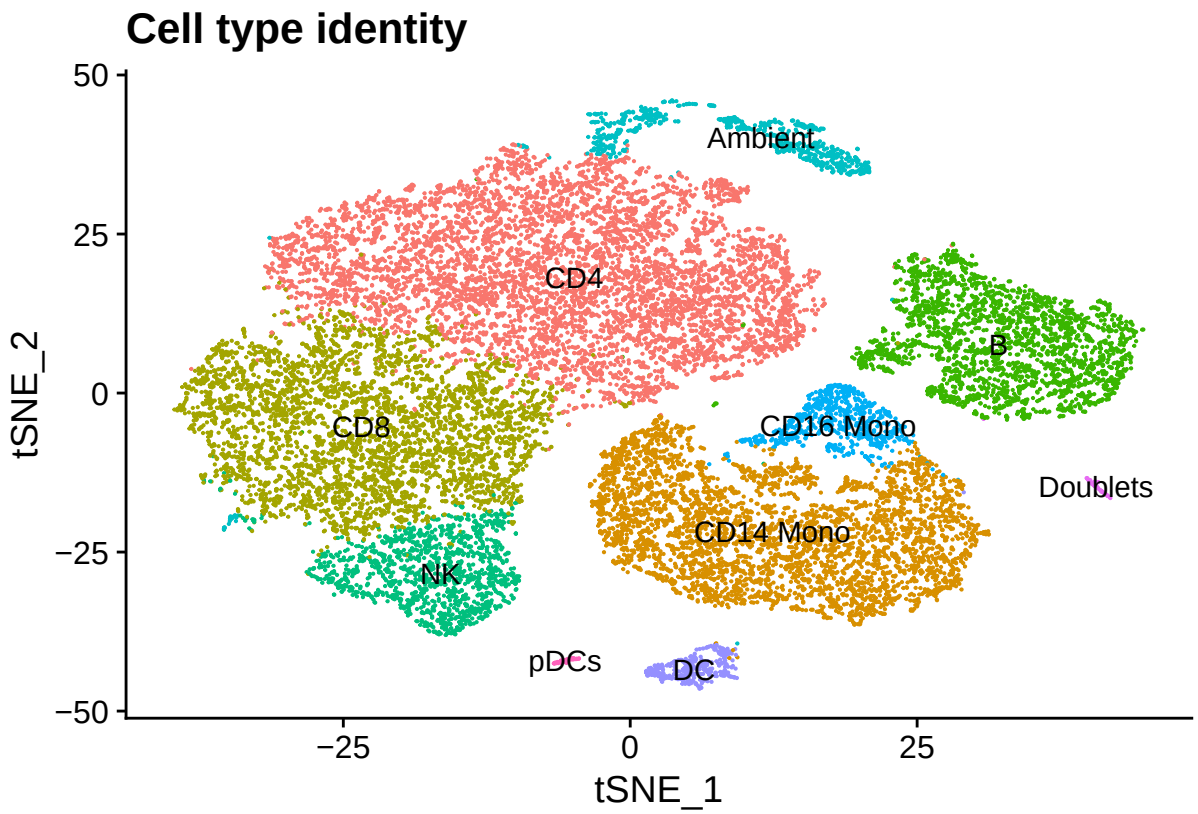


Figure 2F

```
# EmptyNN-retained, EmptyNN-recovered, All-retained
set1 <- colnames(retained)[retained$emptynn & !retained$emptydrops & !retained$cellranger & !retained$c
set2 <- colnames(retained)[!retained$emptynn & retained$emptydrops & retained$cellranger & retained$cell
set3 <- colnames(retained)[retained$emptynn & retained$emptydrops & retained$cellranger & retained$cell
tmp <- retained[,c(set1,set2,set3)]
tmp$detection <- c(rep("EmptyNN retained",length(set1)),
                   rep("EmptyNN removed",length(set2)),
                   rep("All",length(set3)))
list <- c("CD4","CD14 Mono","CD8","B")
res <- do.call(rbind,lapply(list,function(x){
  tmp <- tmp[,Idents(tmp)==x]
  Idents(tmp) <- tmp$detection
  m1 <- FindMarkers(tmp,ident.1="EmptyNN retained",ident.2="All")
  m1$method <- "EmptyNN retained vs All"
  a <- sum(m1$p_val_adj<0.05)
  m2 <- FindMarkers(tmp,ident.1="EmptyNN removed",ident.2="All")
  m2$method <- "EmptyNN removed vs All"
  b <- sum(m2$p_val_adj<0.05)
  return(c(a,b))
}))
res <- data.frame(res)
```

```
colnames(res) <- c("EmptyNN_retained vs All","EmptyNN_removed vs All")
res$celltype <- list
library(reshape2)
df <- melt(res,id.vars = 'celltype')
colnames(df)[1:2] <- c('Celltype',"comparison")
```

```
ggplot(df, aes(x=Celltype, y=value,fill=comparison)) +
  geom_bar(stat="identity", position=position_dodge())+
  scale_y_continuous(trans='log10',name = "# of differentially expressed genes")+
  theme_bw()
```

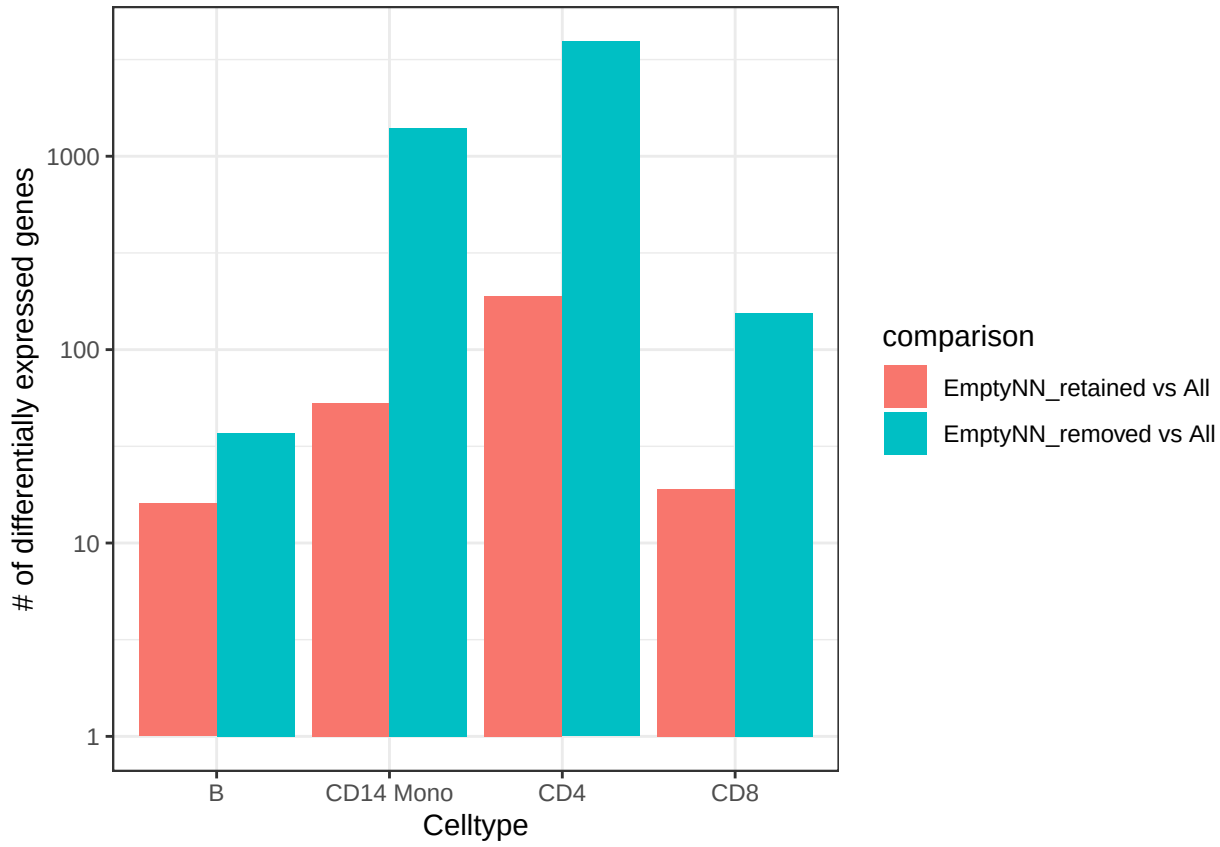


Figure 2G

```
tmp <- tmp[,Idents(tmp)=="CD4"]
Idents(tmp) <- tmp$detection
m1 <- FindMarkers(tmp,ident.1="EmptyNN retained",ident.2="All")
m1$method <- "EmptyNN retained vs All"
m2 <- FindMarkers(tmp,ident.1="EmptyNN removed",ident.2="All")
m2$method <- "EmptyNN removed vs All"
m <- rbind(m1,m2)
ggplot(m, aes(x = avg_log2FC, y = -log10(p_val_adj)))+
  geom_point() +geom_hex(bins = 50)+
```

```
  scale_fill_viridis_c(trans = 'log', breaks = c(1, 10, 100))+
  theme_bw()+
  facet_wrap(~method)+
  geom_vline(xintercept = 0, linetype = 'dashed') +
  ggtitle("CD4 T cells")+xlim(c(-10,10))
```

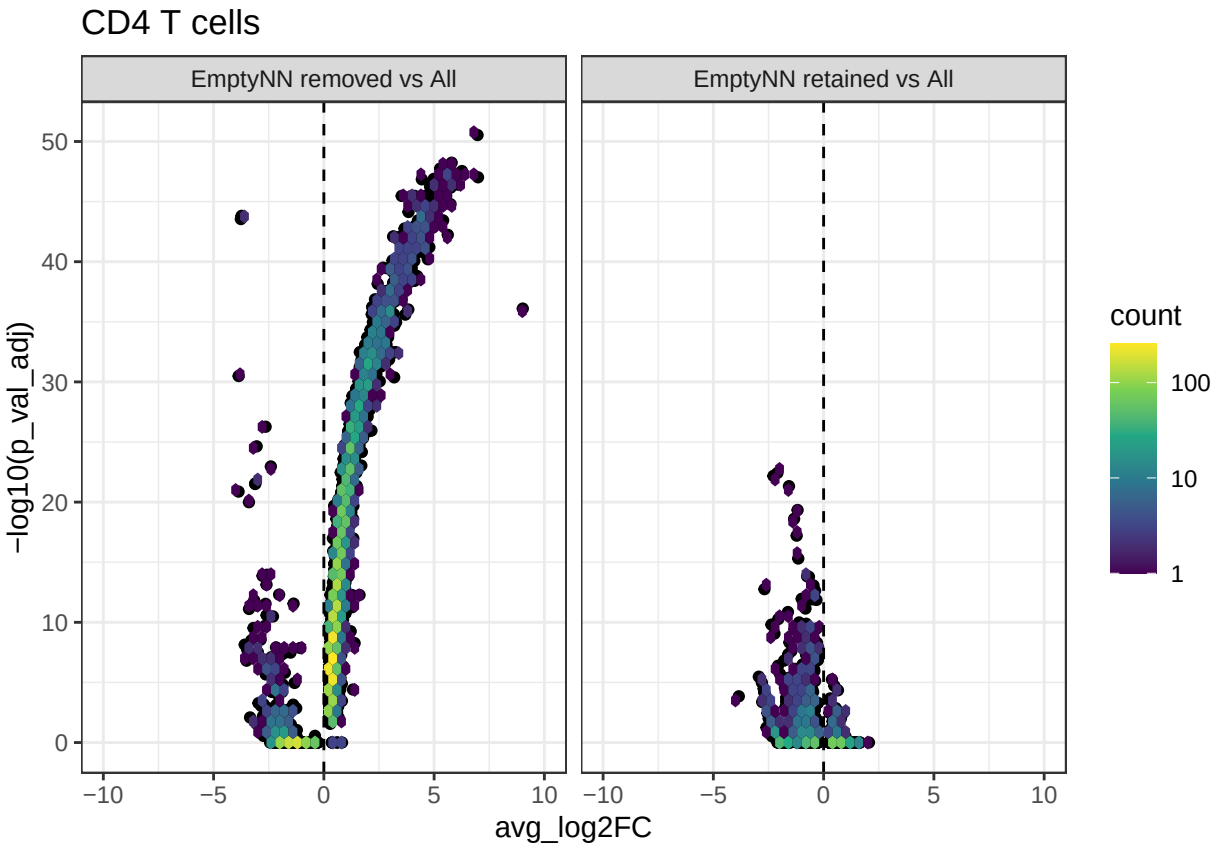
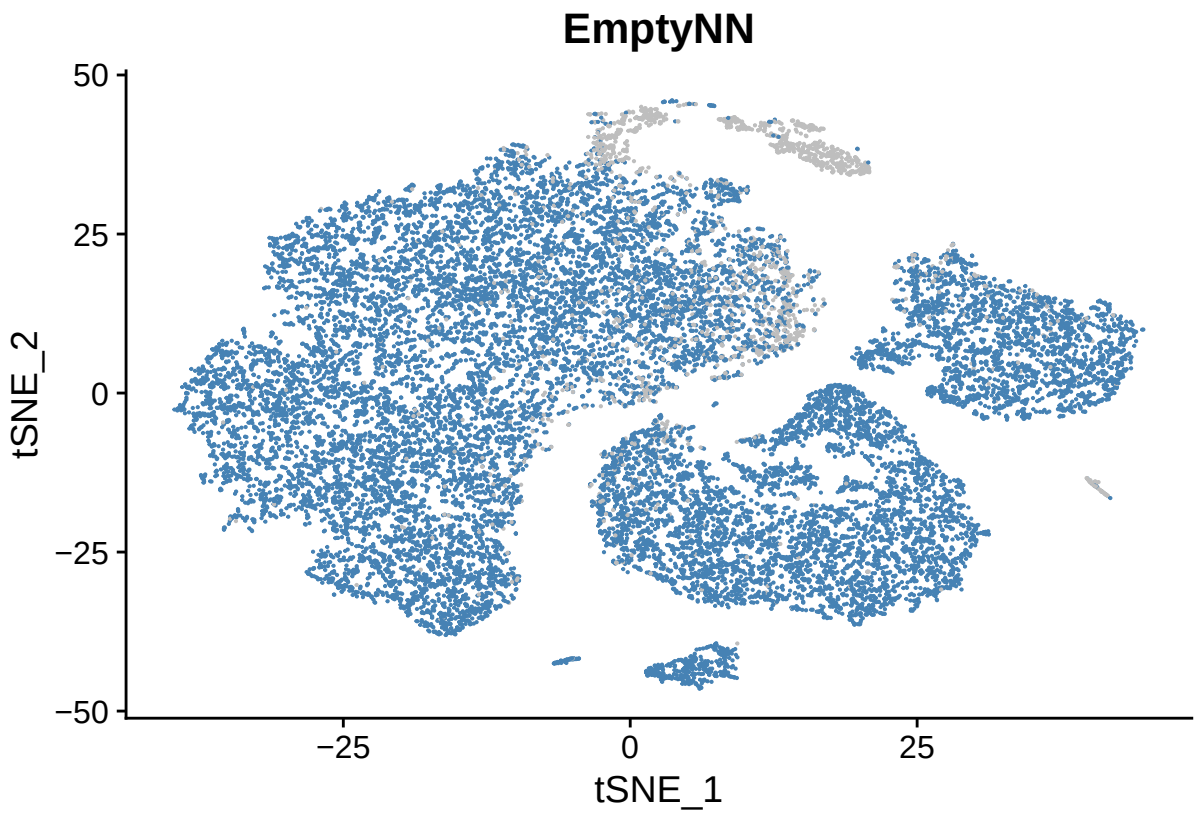
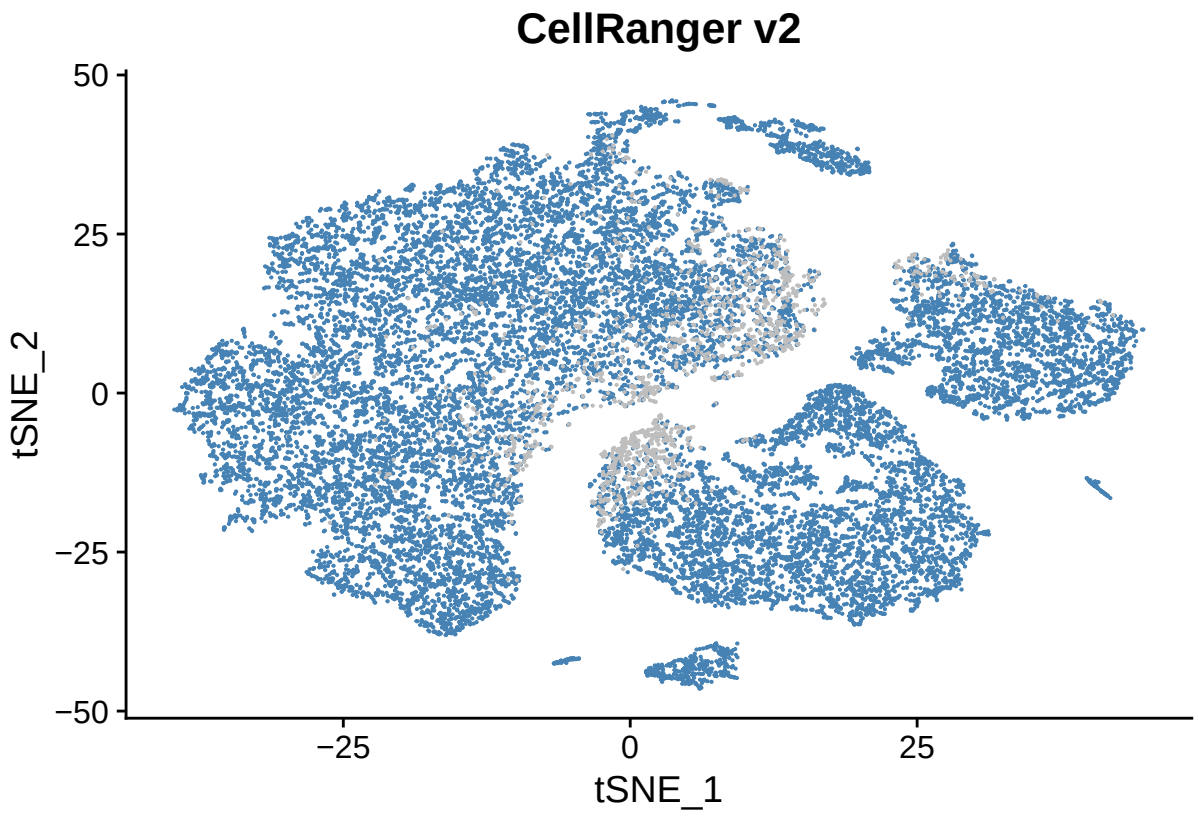


Figure S3C,D,E,F

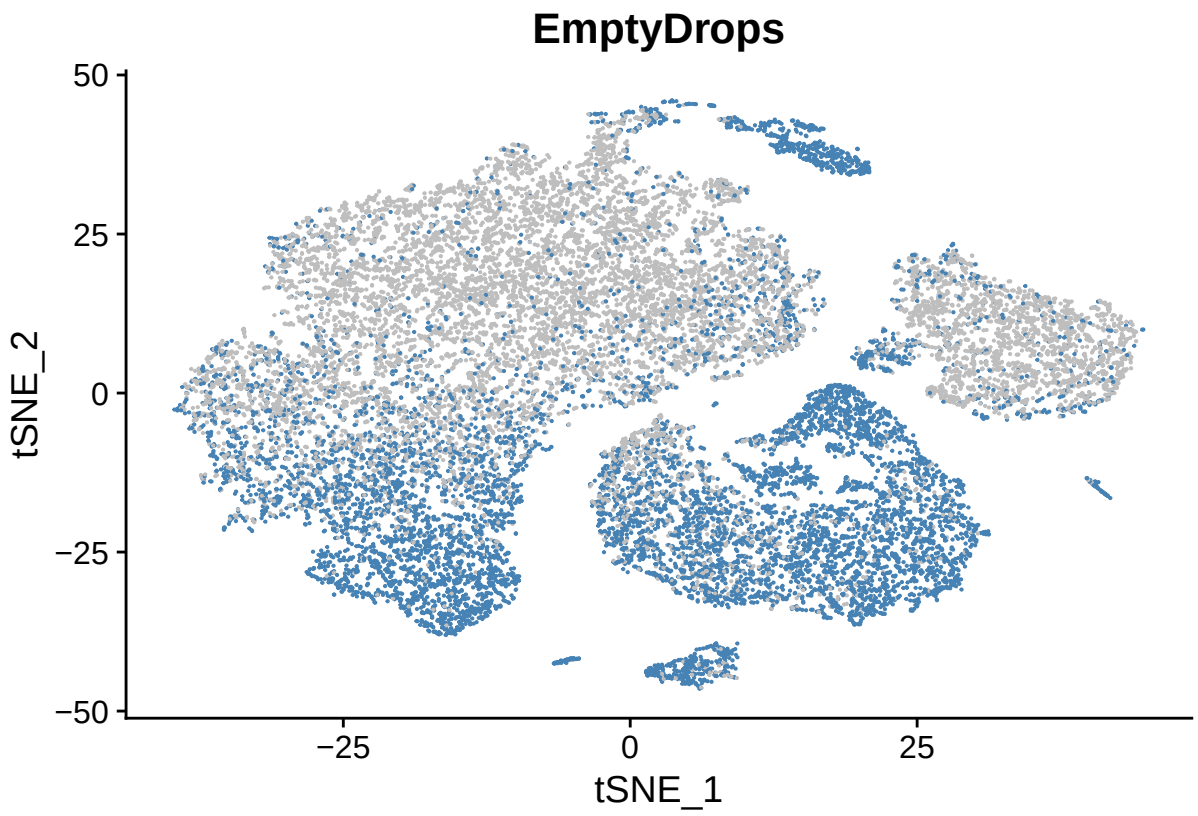
```
DimPlot(retained,reduction='tsne',group.by = 'emptynn',cols=c('grey','steelblue'))+
  NoLegend()+labs(title="EmptyNN")
```



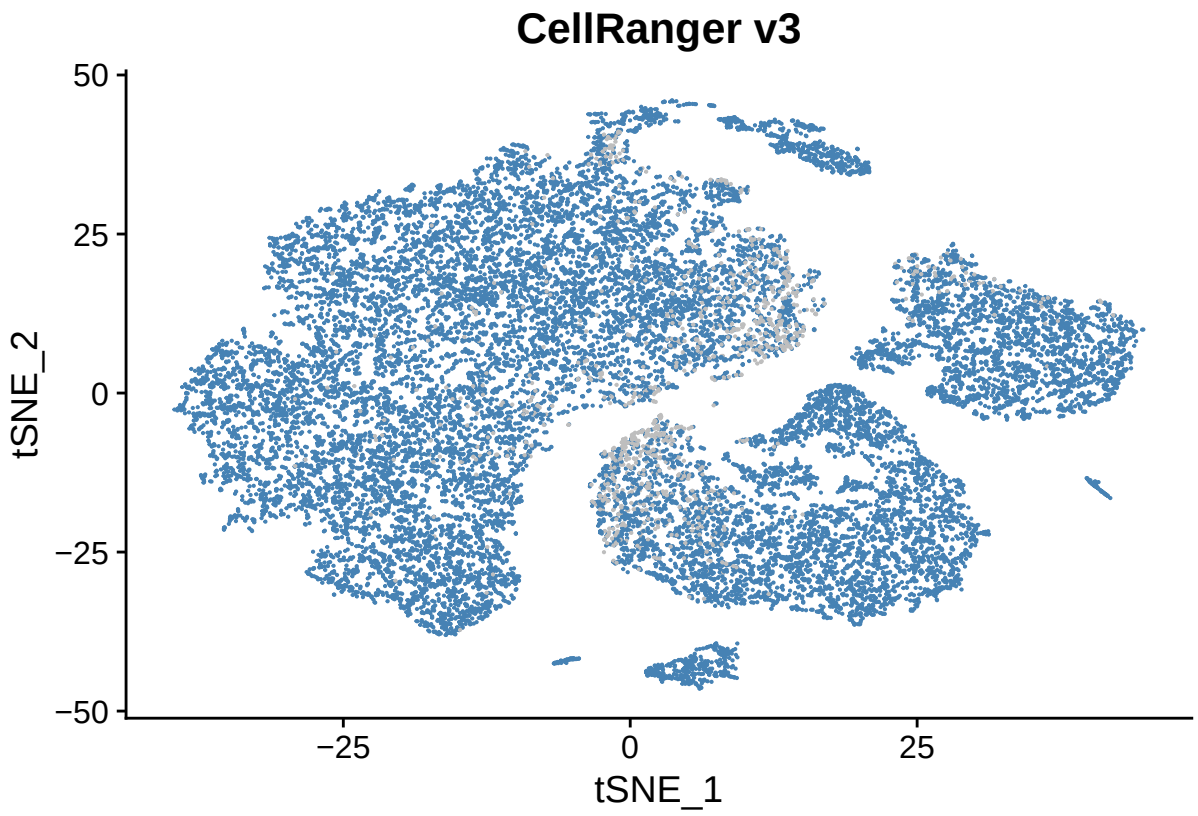

```
DimPlot(retained,reduction='tsne',group.by = 'cellranger',cols=c('grey','steelblue'))+
  NoLegend()+labs(title="CellRanger v2")
```




```
DimPlot(retained,reduction='tsne',group.by = 'emptydrops',cols=c('grey','steelblue'))+
  NoLegend()+labs(title="EmptyDrops")
```




```
DimPlot(retained,reduction='tsne',group.by = 'cellranger3.0',cols=c('grey','steelblue'))+
  NoLegend()+labs(title="CellRanger v3")
```




```
DimPlot(retained,reduction='tsne',group.by = 'cellbender',cols=c('grey','steelblue'))+
  NoLegend()+labs(title="CellBender")
```

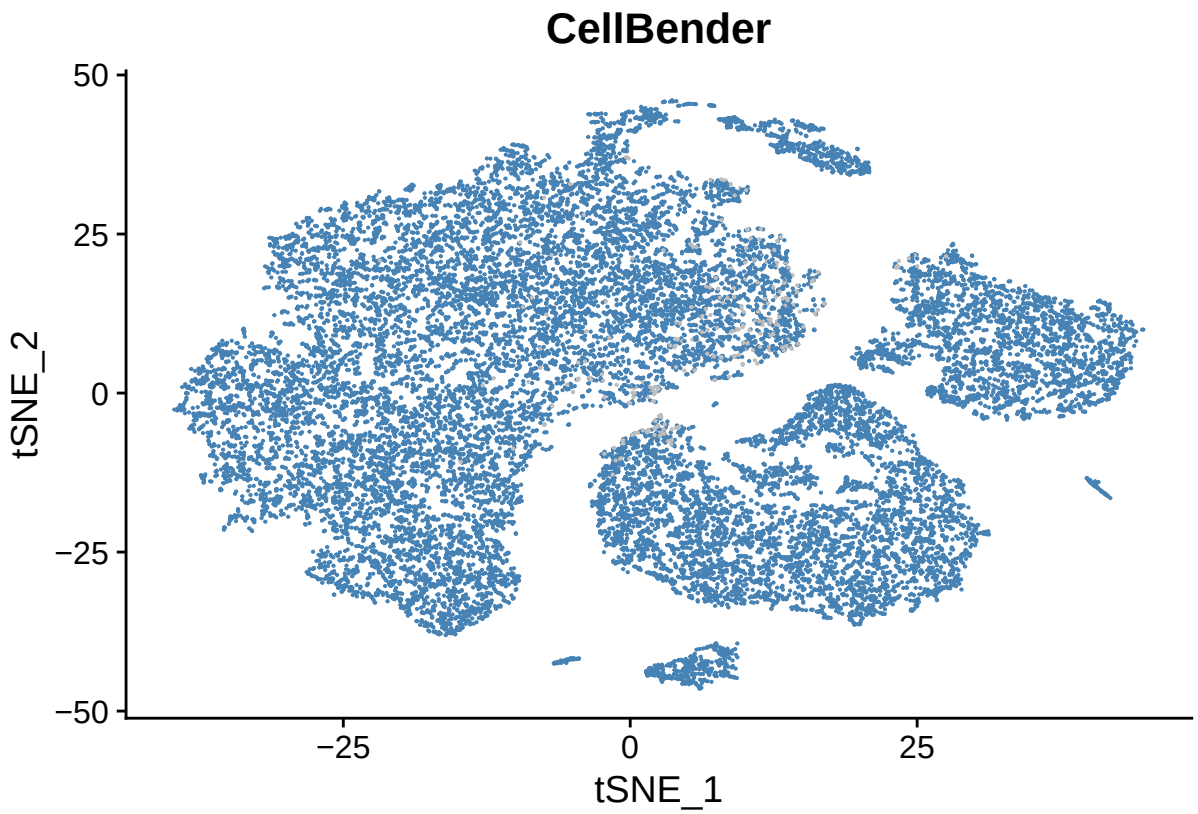


Figure 2H

```
label[,2] <- ifelse(label[,1]=="Negative","cell-free","cell-containing")
bcs <- rownames(nn.res$prediction)
rocobj1 <- roc(label[bcs,2], nn.res$prediction[bcs,'mean.crossval'])
```

```
## Setting levels: control = cell-containing, case = cell-free
```

```
## Setting direction: controls > cases
```

```
rocobj2 <- roc(label[names(ranger.keep),2], as.numeric(ranger.keep))
```

```
## Setting levels: control = cell-containing, case = cell-free
## Setting direction: controls > cases
```

```
rocobj2 <- roc(label[names(ranger.keep),2], as.numeric(ranger.keep))
```

```
## Setting levels: control = cell-containing, case = cell-free
## Setting direction: controls > cases
```

```r
bcs <- rownames(e.out[!is.na(e.out$FDR),])
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

## Warning: package 'BiocGenerics' was built under R version 4.0.5

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following object is masked from 'package:pROC':
##
##     var

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:plyr':
##
##     rename

## The following object is masked from 'package:Matrix':
##
##     expand

## The following object is masked from 'package:base':
##
##     expand.grid
```

```
rocobj3 <- roc(label[bcs,2], e.out[bcs,'FDR'])
```

```
## Setting levels: control = cell-containing, case = cell-free
```

```
## Setting direction: controls > cases
```

```
rocobj4 <- roc(label[names(bender.keep),2], as.numeric(bender.keep))
```

```
## Setting levels: control = cell-containing, case = cell-free
## Setting direction: controls > cases
```

```
ranger3.keep <- rownames(Stoeckius.counts) %in% colnames(retained)[retained$cellranger3.0]
names(ranger3.keep) <- rownames(Stoeckius.counts)
rocobj5 <- roc(label[names(ranger3.keep),2], as.numeric(ranger3.keep))
```

```
## Setting levels: control = cell-containing, case = cell-free
## Setting direction: controls > cases
```

```
ggroc(list("EmptyNN, 0.9473"=rocobj1,"CellRanger 2.0, 0.8688"=rocobj2,
           "EmptyDrops, 0.7967"=rocobj3,"CellBender, 0.8880"=rocobj4,
           "CellRanger 3.0, 0.9016"=rocobj5),linetype='dashed',
      legacy.axes = TRUE) + labs(x = "1-Specificity", y = "Sensitivity")+
  geom_segment(aes(x = 0, xend = 1, y = 0, yend = 1),
               color="darkgrey", linetype="dashed")+theme_bw()
```