Figure 4

EmptyNN - Figure 4

The following code reproduces the Figure 4 in our EmptyNN manuscript.

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

Load libraries

```
library("Seurat")
library("Matrix")
library("ggplot2")
library("pheatmap")
load("./../data/BlueYellowColormaps_V1.RData")
# R version 3.6.3, Seurat_3.2.3, Matrix_1.3-2, ggplot_2_3.3.3, pheatmap_1.0.12
```

Load seurat objects containing barcodes retained by EmptyNN or CellRanger 2.0

```
pbmc8k_retained <- readRDS("./../data/pbmc_8k_retained.rds")
neuron900_retained <- readRDS("./../data/neuron900_retained.rds")
```

Define functions: runSeurat() and plot_heatmap()

```
runSeurat <- function(seu,RNA.thres,mt.thres,resolution,verbose){
        seu[["percent.mt"]] <- PercentageFeatureSet(seu, pattern = "^MT|^mt")
        seu <- subset(seu, subset = nFeature_RNA > RNA.thres &
                              percent.mt < mt.thres)
        seu <- NormalizeData(seu,verbose = verbose)
        seu <- FindVariableFeatures(seu,verbose = verbose)
        seu <- ScaleData(seu, features = VariableFeatures(seu),verbose = verbose)
        seu <- RunPCA(seu,features=VariableFeatures(seu),verbose = verbose)
        seu <- FindNeighbors(seu, dims = 1:10,verbose = verbose)
        seu <- FindClusters(seu, resolution = resolution,verbose = verbose)
        seu <- RunTSNE(seu, dims = 1:10, check_duplicates = verbose,verbose = verbose)
        return(seu)
}
```

```
plot_heatmap <- function(seu,n_top,title){
        seu <- seu[-grep("^RPL|^RPS|^MT|^Rps|^Rpl|^mt", rownames(seu)),]
        des <- FindAllMarkers(seu, only.pos = TRUE, min.pct = 0.25,
                              logfc.threshold = 0.25,verbose=FALSE)
        asplit_genes <- split(1:nrow(des), des$cluster)
        # take top n genes
        genes <- unlist(lapply(asplit_genes, function(x) des[x[1:n_top], "gene"]))
        # Average cells within each cluster
        asplit_cells <- split(rownames(seu@meta.data), seu@active.ident)
        means <- do.call(cbind, lapply(asplit_cells, function(x){
        s1 <- Matrix::rowMeans(seu@assays$RNA@data[genes, sample(unlist(x), 10)])
        s2 <- Matrix::rowMeans(seu@assays$RNA@data[genes, sample(unlist(x), 10)])
        s3 <- Matrix::rowMeans(seu@assays$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",
                 breaks = seq(-2, 2, length = length(yellow2blue) + 1), col = yellow2blue,
                 annotation_col = anno_col,show_colnames = F,main=title)
}
```

Figure 4A

```
cellranger <- pbmc8k_retained[,pbmc8k_retained$cellranger]
cellranger_cutoff <- min(cellranger$nCount_RNA)
recovered_bcs <- pbmc8k_retained$nCount_RNA < cellranger_cutoff & pbmc8k_retained$emptynn
recover <- pbmc8k_retained[,recovered_bcs]
recover <- runSeurat(recover,RNA.thres=200,mt.thres=10,
                     resolution=0.1,verbose=FALSE)
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -2.4953
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.32208
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 2.8276e-28
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.090619
```

```
new.cluster <- c("CD4","CD14 Mono","Platelet")
names(new.cluster) <- levels(recover)
recover <- RenameIdents(recover,new.cluster)
DimPlot(recover,label=T)+NoLegend()+labs(title="PBMC_8k_dataset_recovered")
```

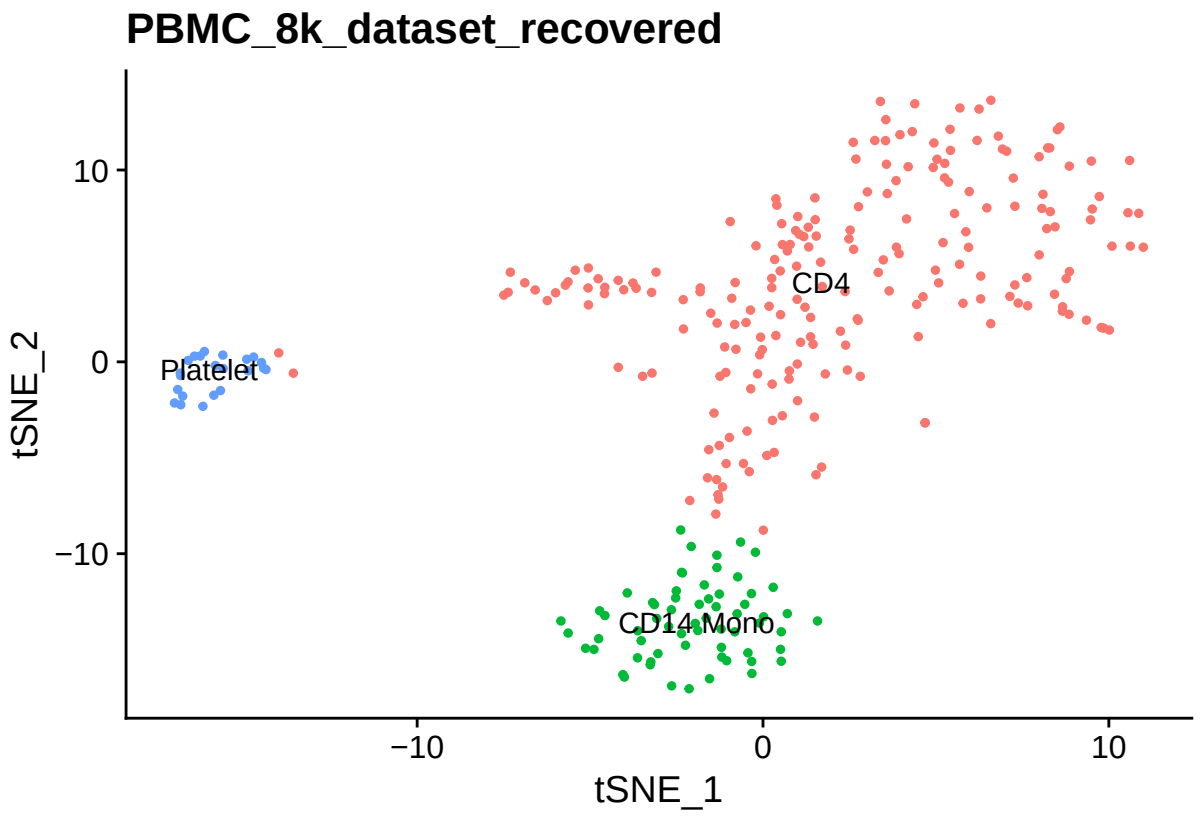


Figure 4B

```
plot_heatmap(recover,n_top=20,"PBMC_8k_dataset_recovered")
```

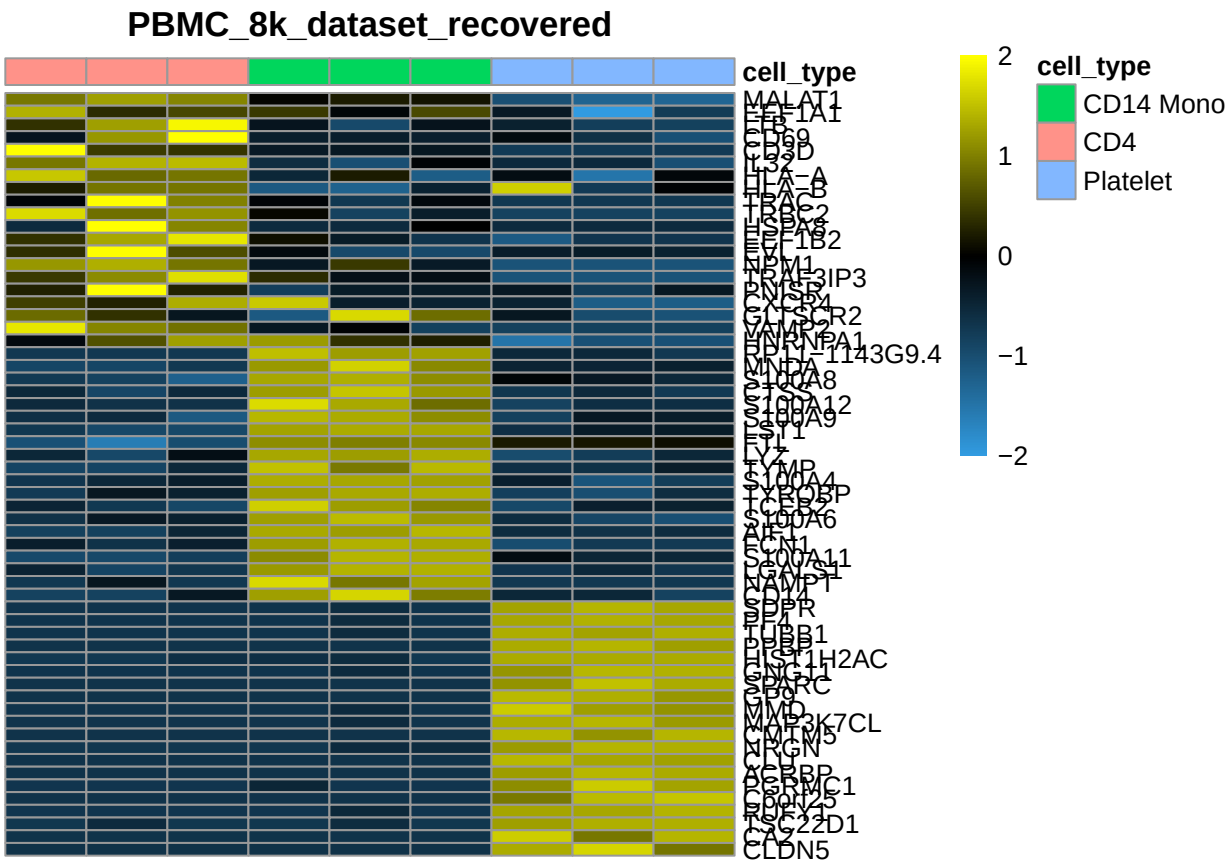


Figure 4C

```
cellranger <- neuron900_retained[,neuron900_retained$cellranger]
cellranger_cutoff <- min(cellranger$nCount_RNA)
recovered_bcs <- neuron900_retained$nCount_RNA < cellranger_cutoff & neuron900_retained$emptynn
recover <- neuron900_retained[,recovered_bcs]
recover <- runSeurat(recover,RNA.thres=200,mt.thres=20,
                     resolution=0.1,verbose=FALSE)
new.cluster <- c("Non-Neuronal","Non-Neuronal","Glutamatergic","GABAergic","GABAergic")
names(new.cluster) <- levels(recover)
recover <- RenameIdents(recover,new.cluster)
DimPlot(recover,label=T)+NoLegend()+labs(title="Neuron_900_dataset_recovered")
```

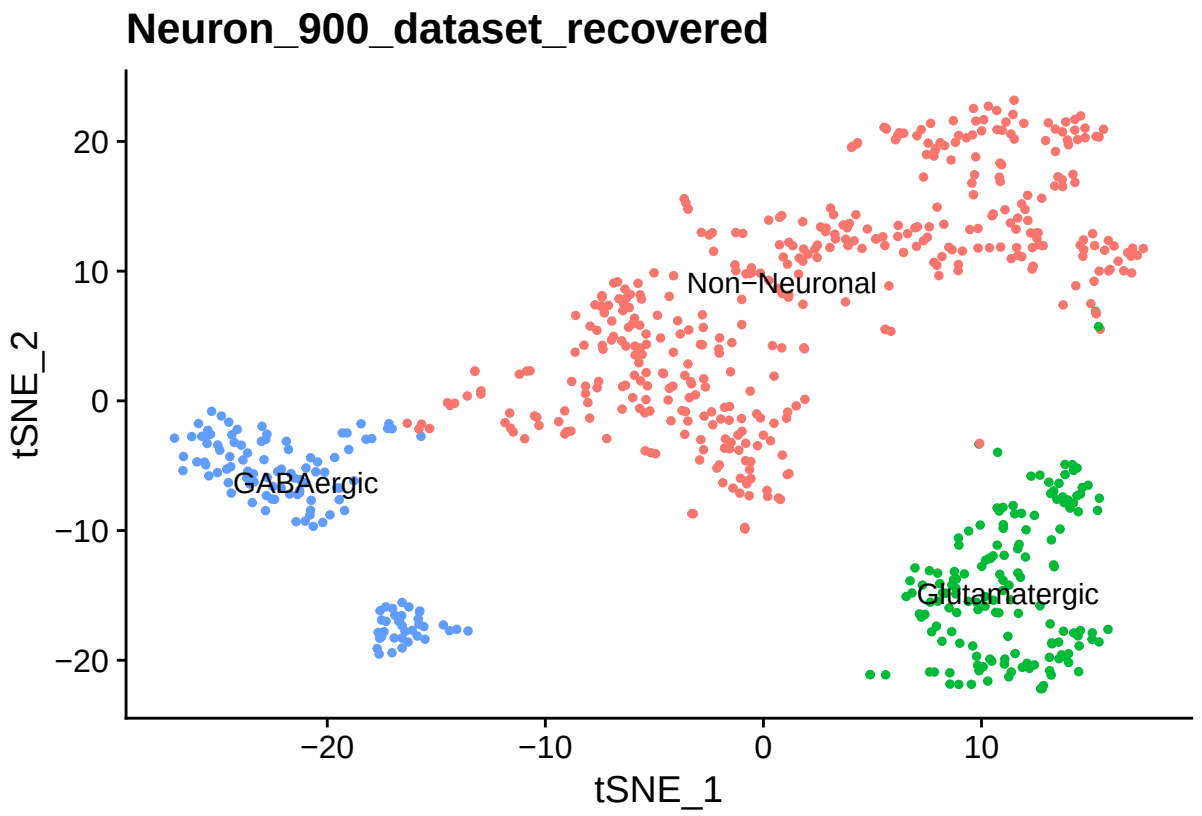
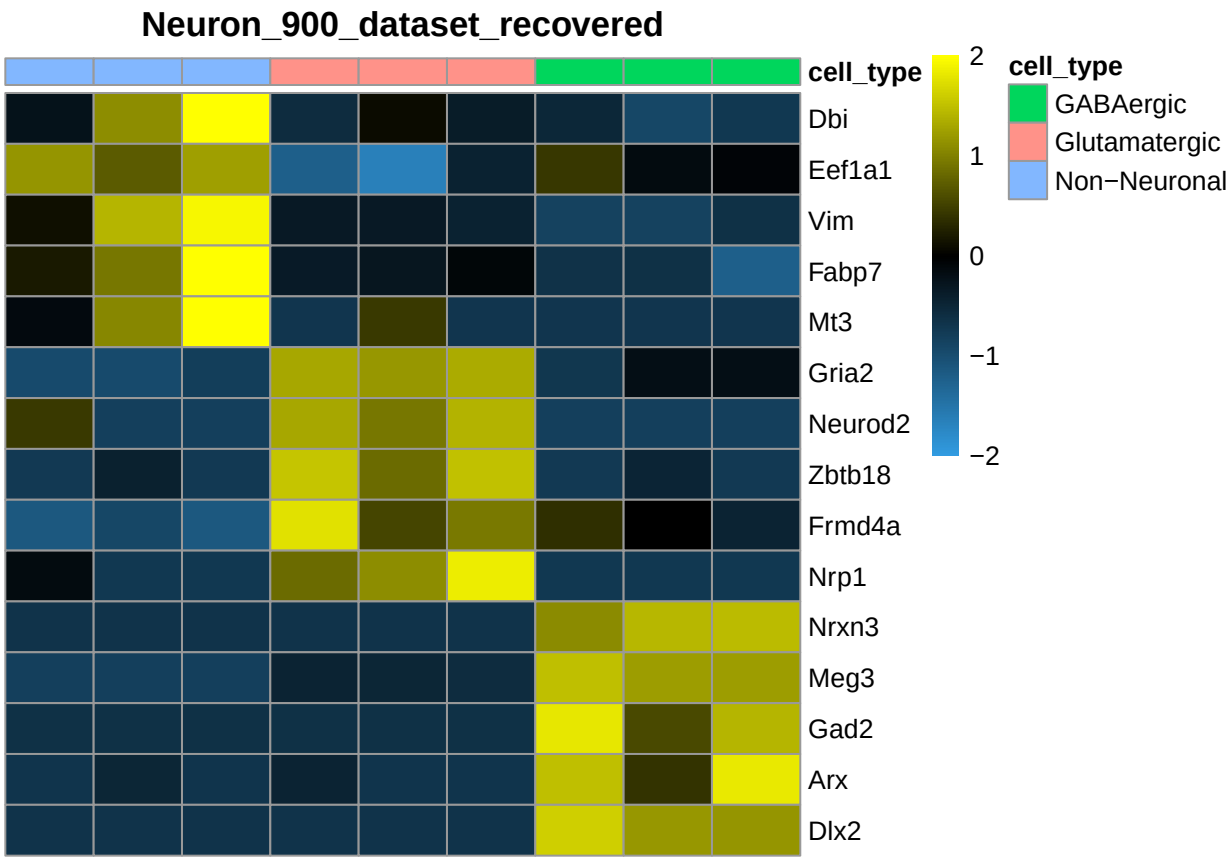


Figure 4D

```
plot_heatmap(recover,n_top=5,"Neuron_900_dataset_recovered")
```


Neuron_900_dataset_recovered
cell_type
Dbi
Eef1a1
Vim
Fabp7
Mt3
Gria2
Neurod2
Zbtb18
Frmd4a
Nrp1
Nrxn3
Meg3
Gad2
Arx
Dlx2
2
1
0
−1
−2
cell_type
GABAergic
Glutamatergic
Non−Neuronal