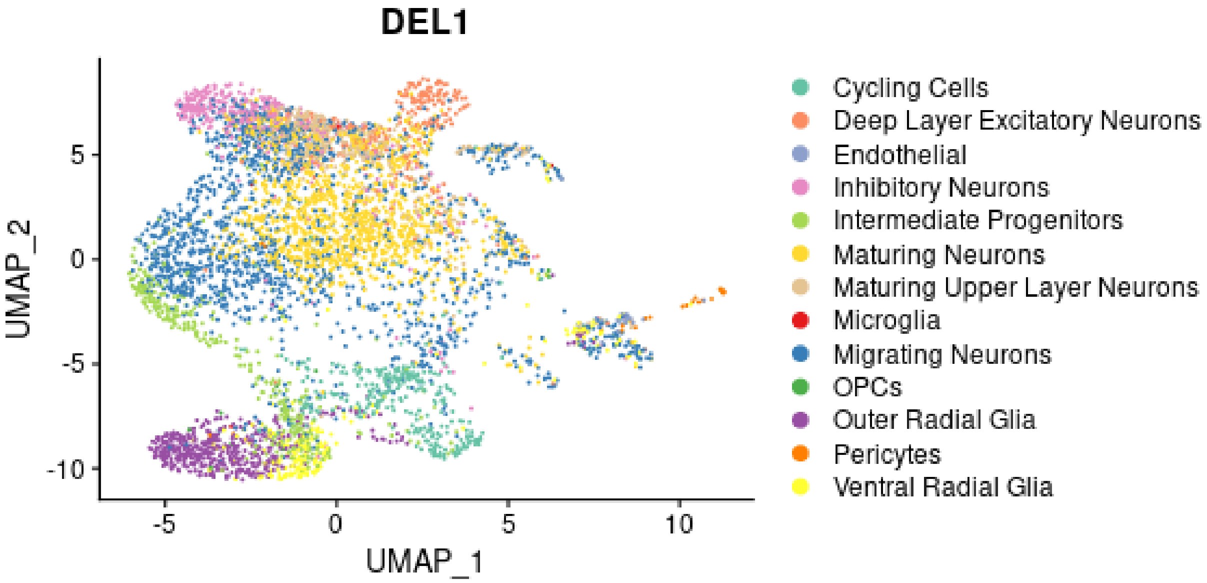
Figure 3

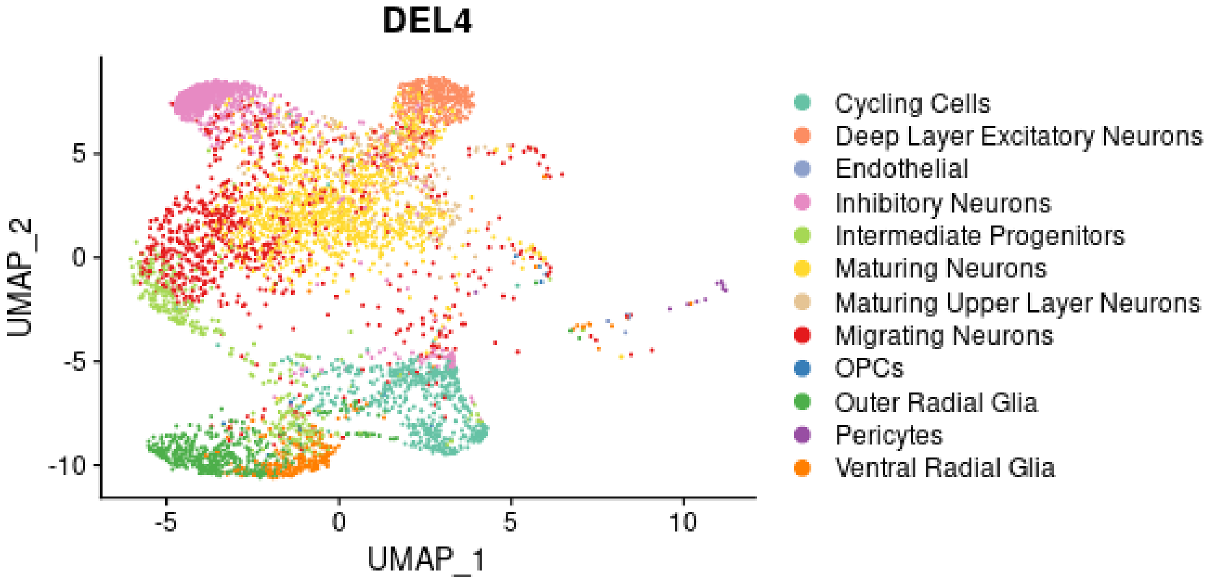
Joe Raymond

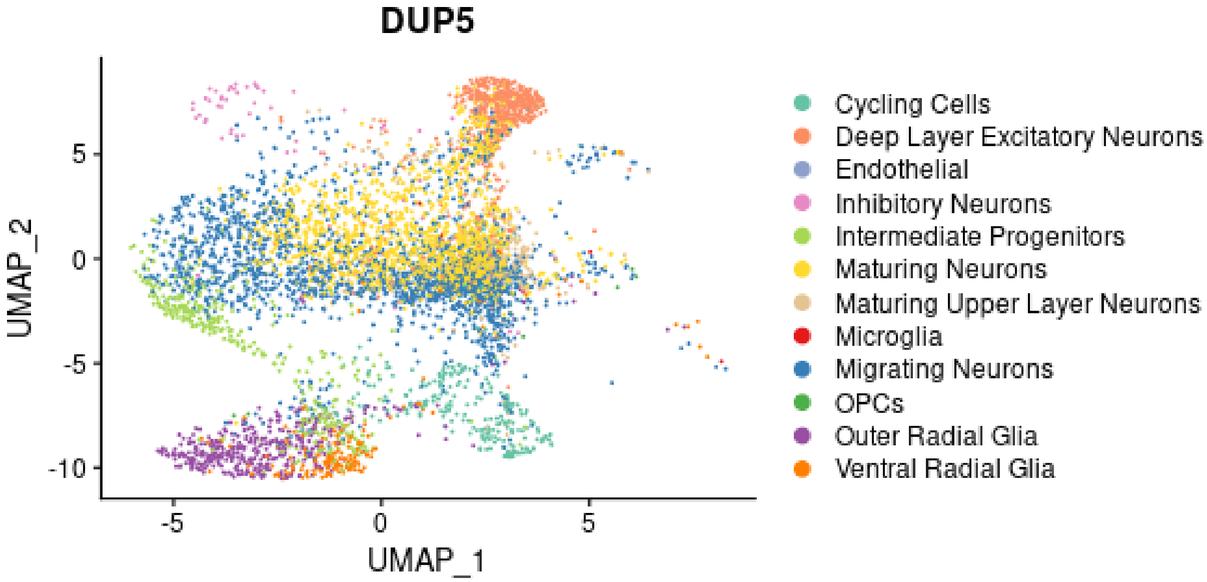
April 5, 2021

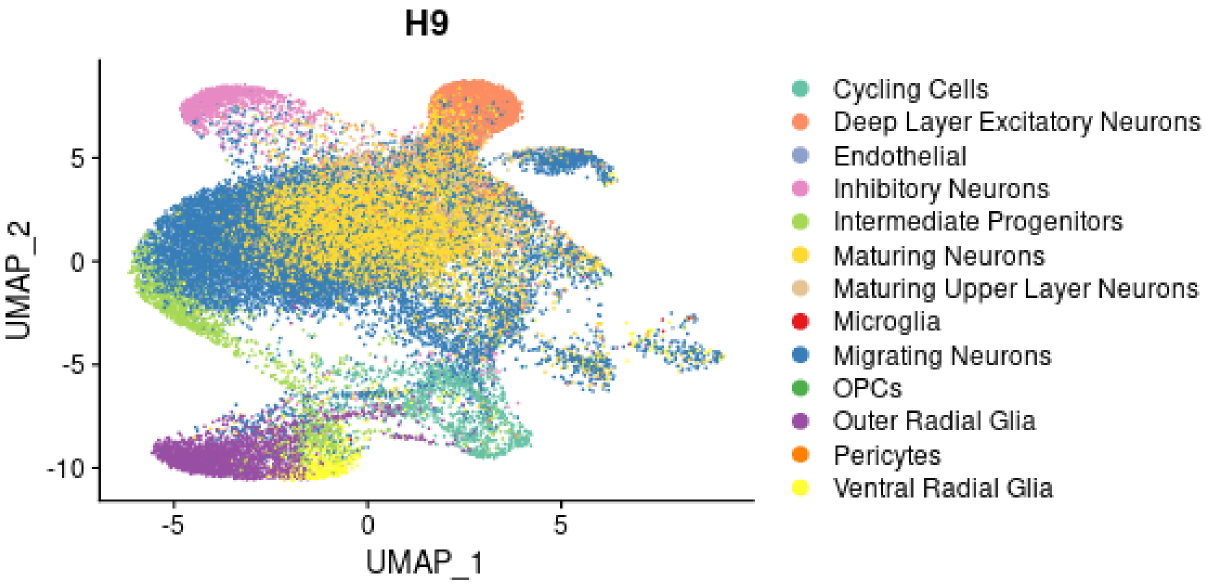
Figure 3B

```r
for(i in unique(dataset$cell_line)){
  cells <- JR.which.cells(dataset, meta.col = 'cell_line', which = i)
  gg <- UMAPPlot(dataset, group.by = 'predicted.id',
                 cols = colors, cells = cells) +
    ggtitle(i)
  print(gg)
}
```

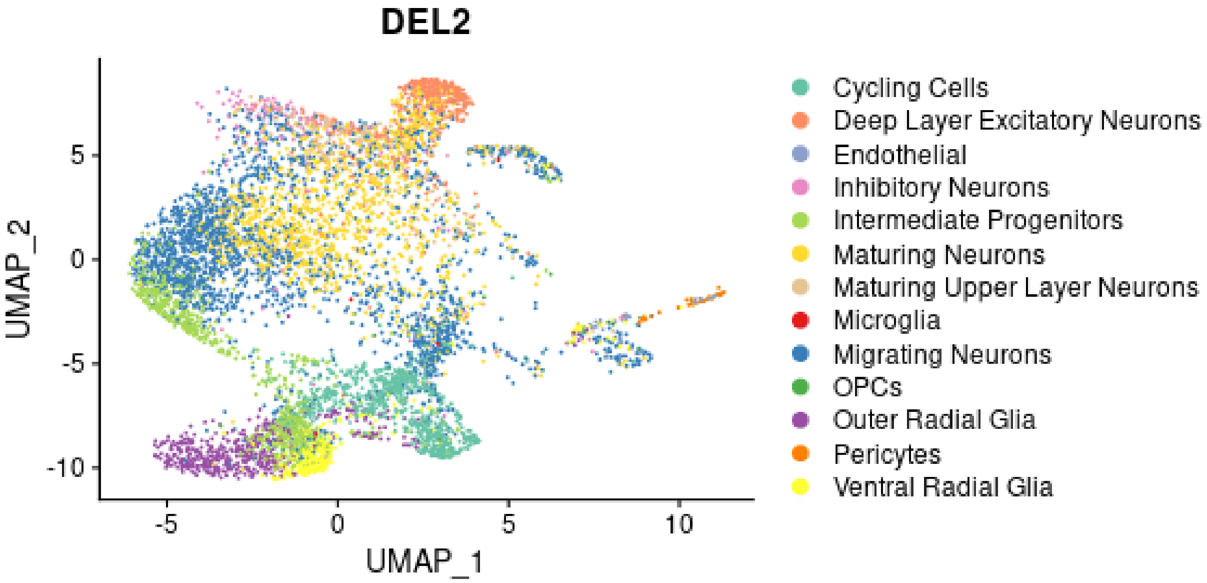



DEL4
UMAP_2
UMAP_1
5
0
-5
-10
-5
0
5
10
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Migrating Neurons
OPCs
Outer Radial Glia
Pericytes
Ventral Radial Glia

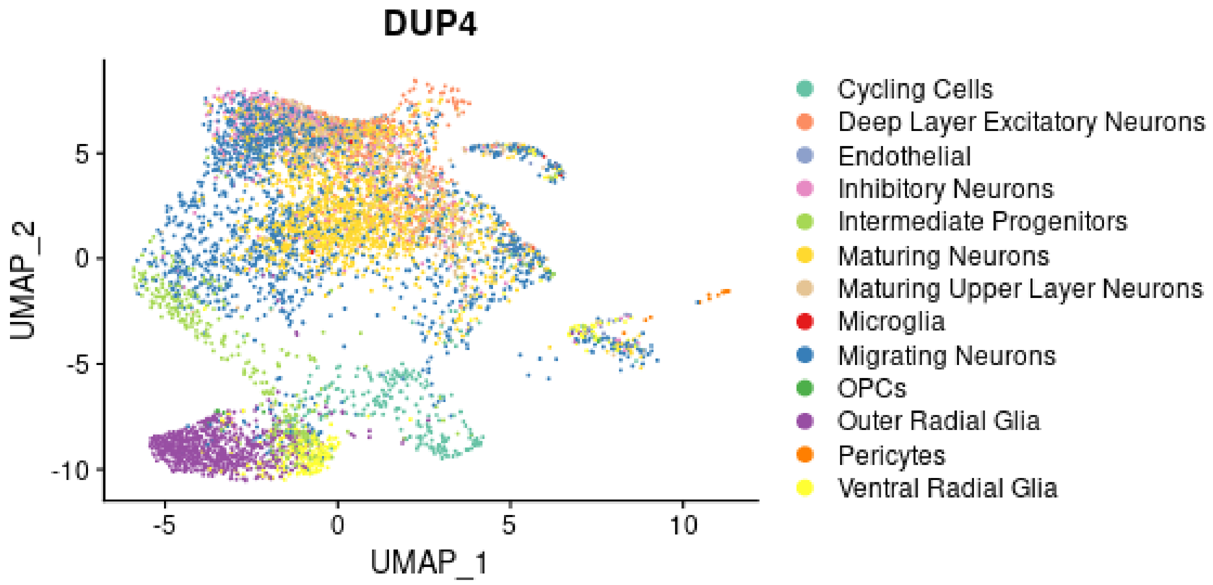

DUP5
UMAP_2
UMAP_1
5
0
-5
-10
-5
0
5
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Microglia
Migrating Neurons
OPCs
Outer Radial Glia
Ventral Radial Glia


H9

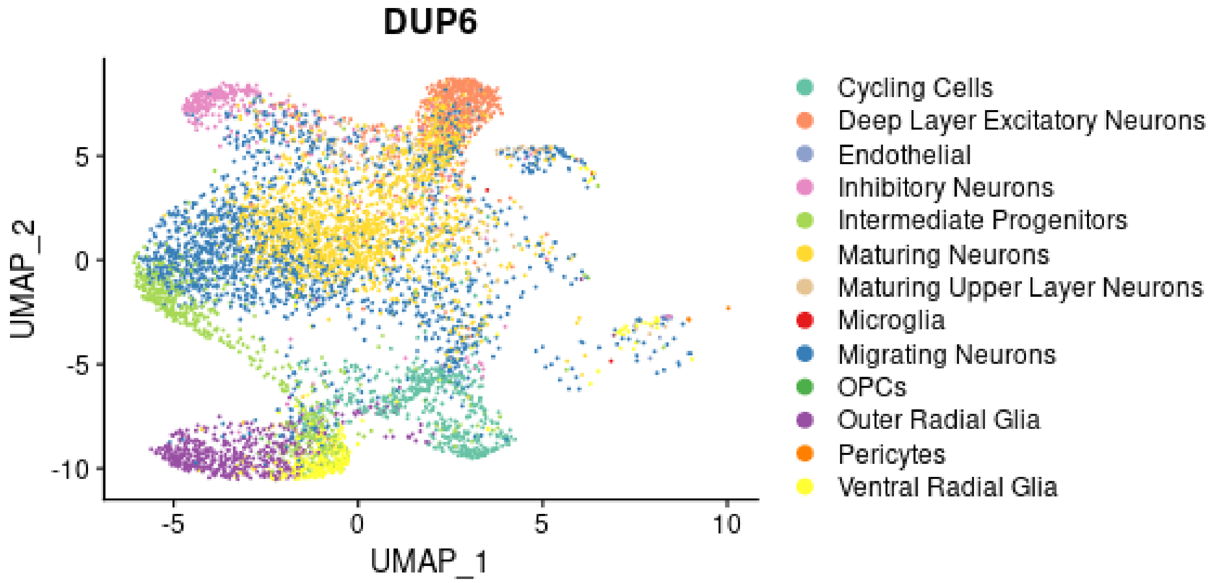
UMAP_2
5
0
-5
-10
-5
0
5
UMAP_1
● Cycling Cells
● Deep Layer Excitatory Neurons
● Endothelial
● Inhibitory Neurons
● Intermediate Progenitors
● Maturing Neurons
● Maturing Upper Layer Neurons
● Microglia
● Migrating Neurons
● OPCs
● Outer Radial Glia
● Pericytes
● Ventral Radial Glia


DEL2

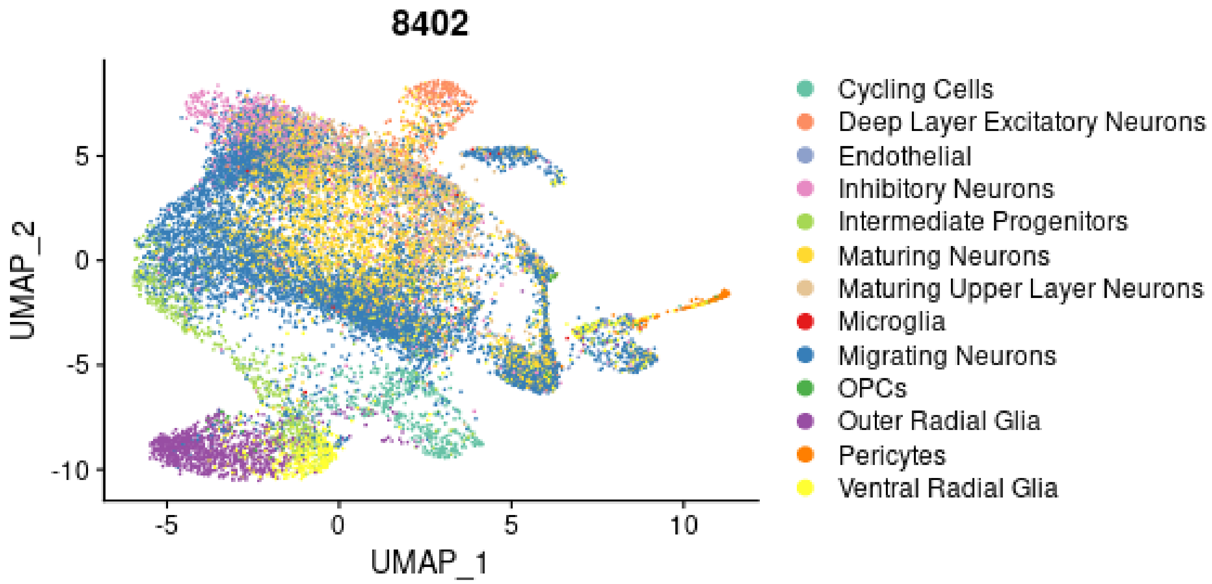
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
● Cycling Cells
● Deep Layer Excitatory Neurons
● Endothelial
● Inhibitory Neurons
● Intermediate Progenitors
● Maturing Neurons
● Maturing Upper Layer Neurons
● Microglia
● Migrating Neurons
● OPCs
● Outer Radial Glia
● Pericytes
● Ventral Radial Glia

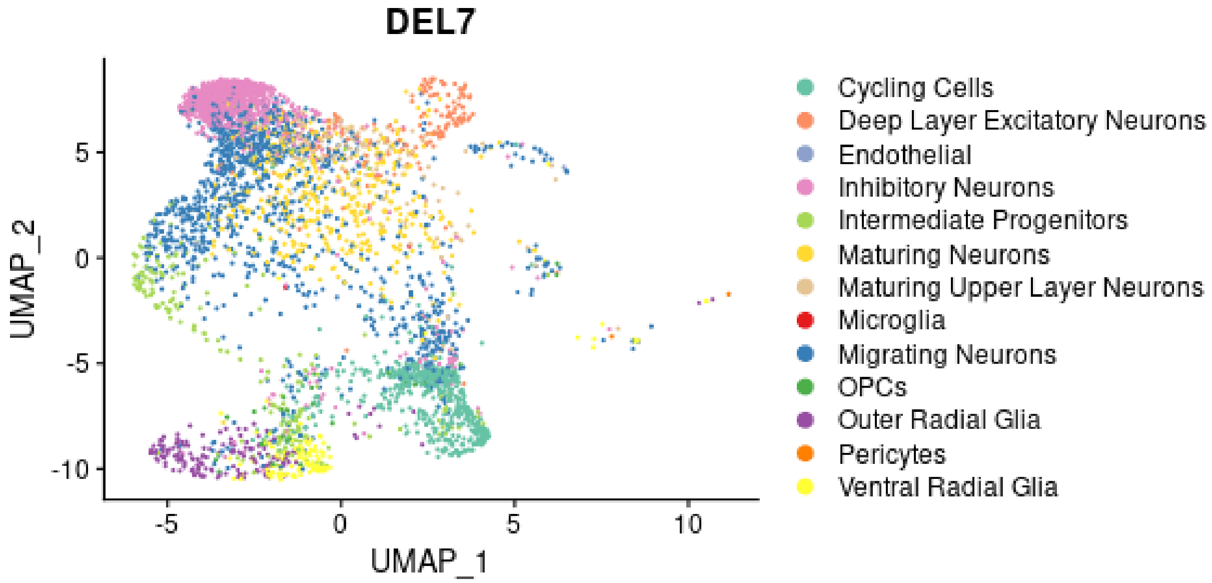

DUP4

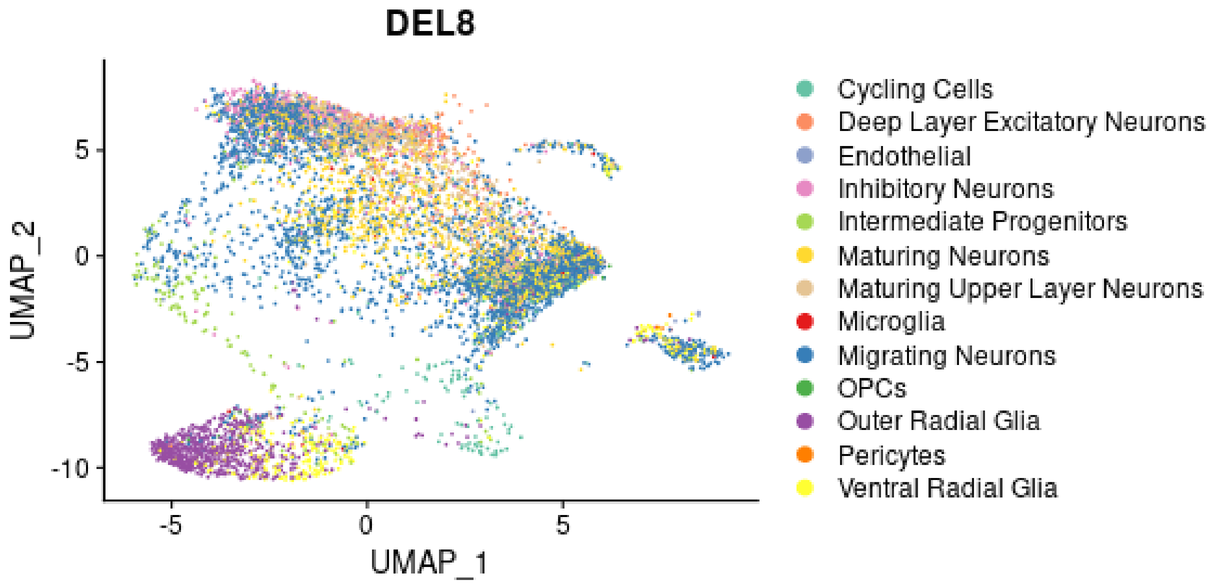
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
● Cycling Cells
● Deep Layer Excitatory Neurons
● Endothelial
● Inhibitory Neurons
● Intermediate Progenitors
● Maturing Neurons
● Maturing Upper Layer Neurons
● Microglia
● Migrating Neurons
● OPCs
● Outer Radial Glia
● Pericytes
● Ventral Radial Glia

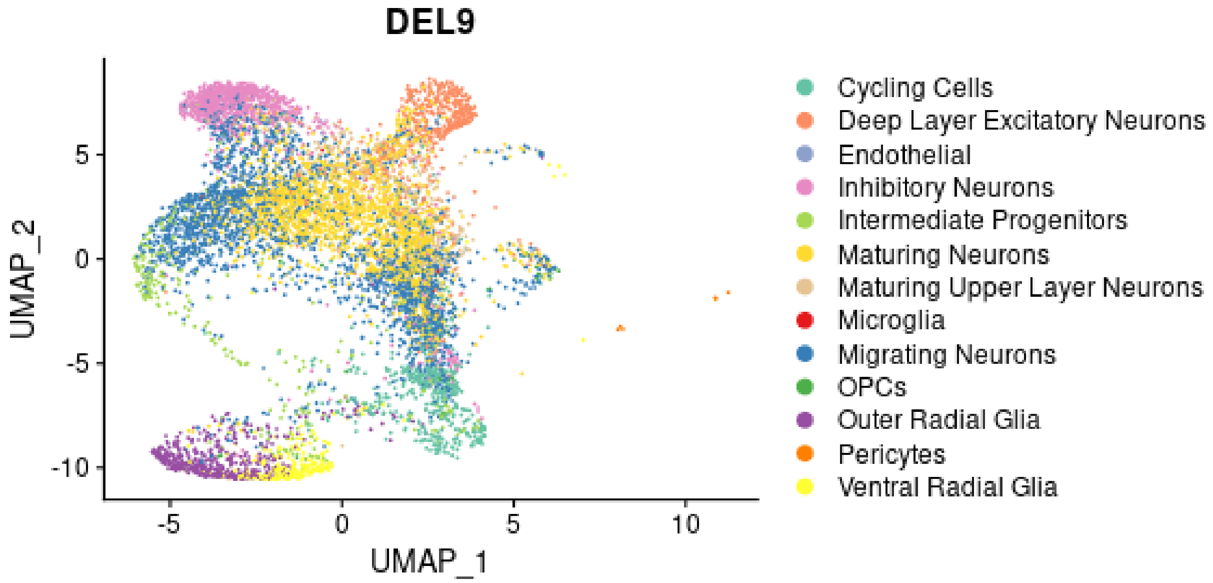

DUP6

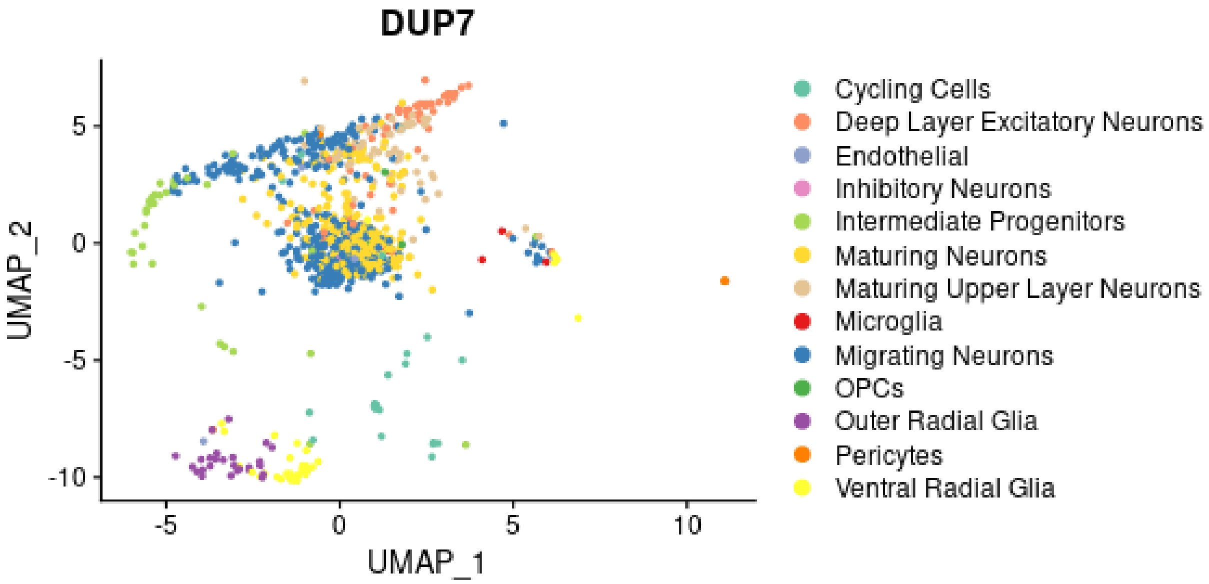
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
● Cycling Cells
● Deep Layer Excitatory Neurons
● Endothelial
● Inhibitory Neurons
● Intermediate Progenitors
● Maturing Neurons
● Maturing Upper Layer Neurons
● Microglia
● Migrating Neurons
● OPCs
● Outer Radial Glia
● Pericytes
● Ventral Radial Glia

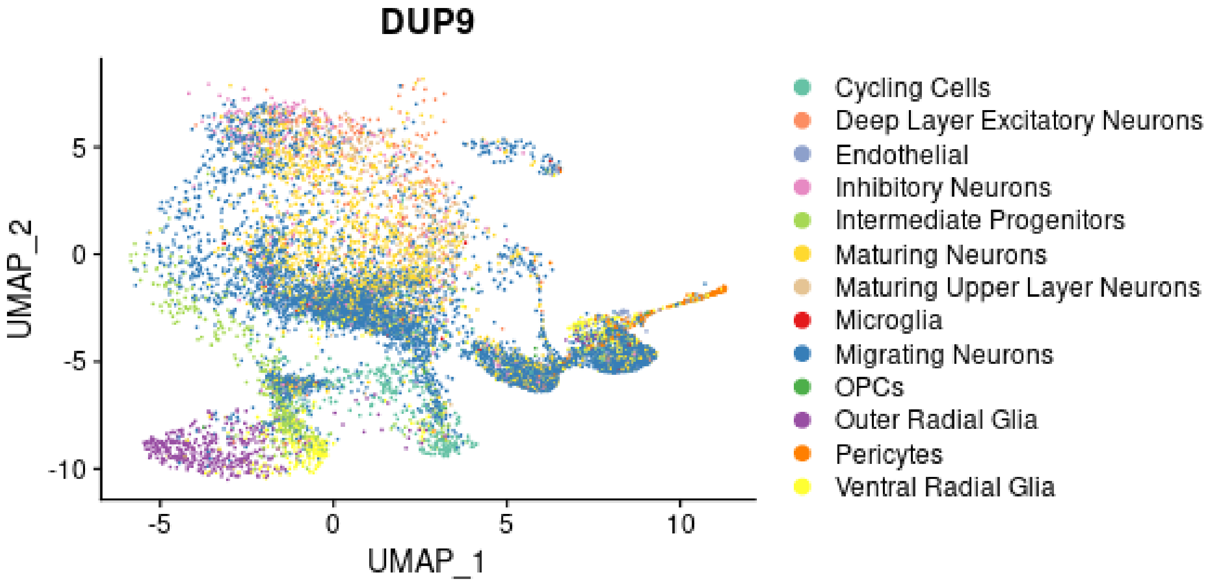

8402
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Microglia
Migrating Neurons
OPCs
Outer Radial Glia
Pericytes
Ventral Radial Glia

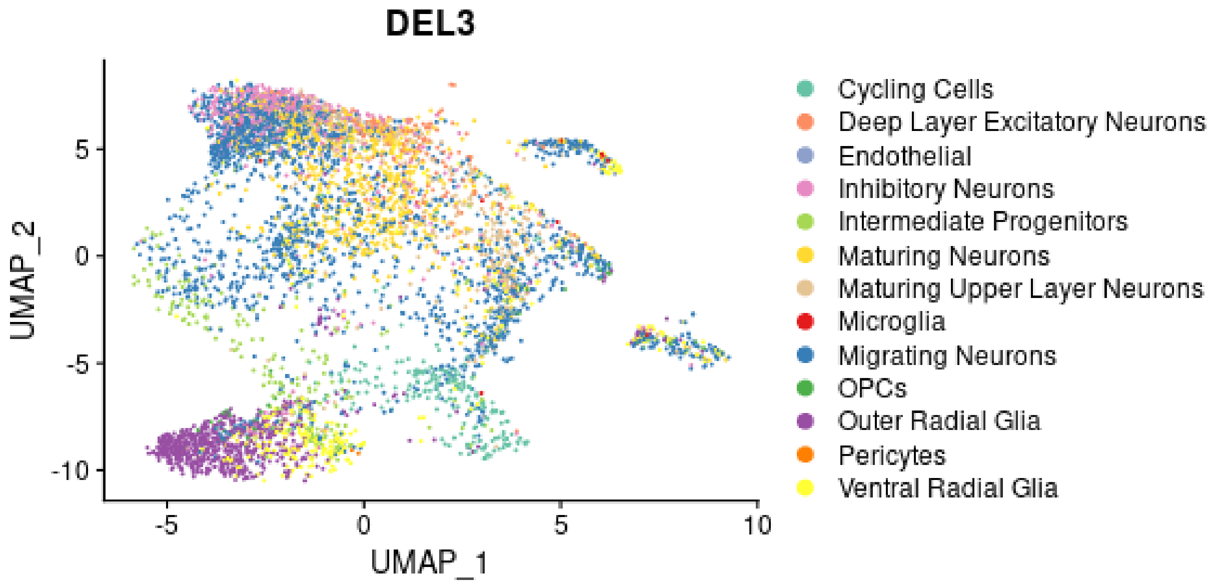

DEL7
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Microglia
Migrating Neurons
OPCs
Outer Radial Glia
Pericytes
Ventral Radial Glia

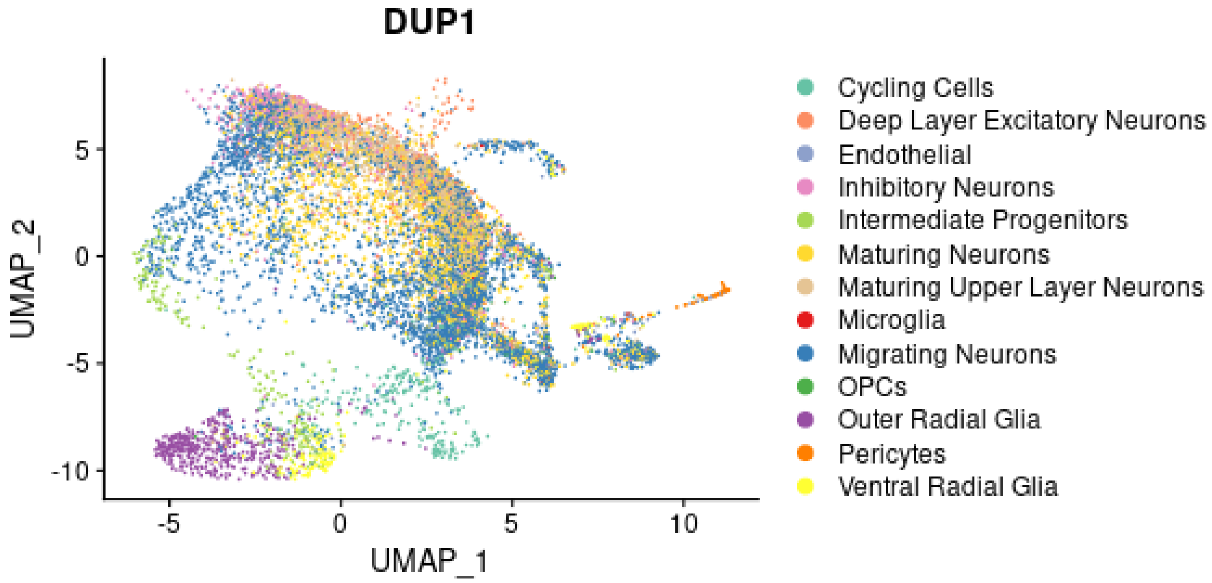

DEL8
UMAP_2
5
0
-5
-10
-5
0
5
UMAP_1
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Microglia
Migrating Neurons
OPCs
Outer Radial Glia
Pericytes
Ventral Radial Glia

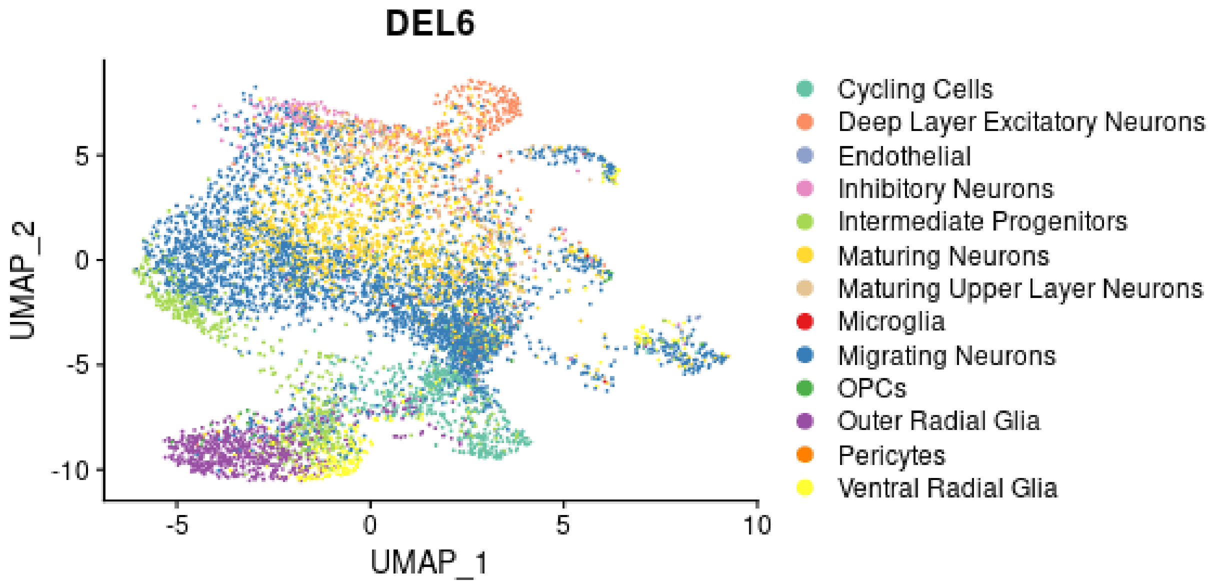

DEL9
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Microglia
Migrating Neurons
OPCs
Outer Radial Glia
Pericytes
Ventral Radial Glia


DUP7
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
● Cycling Cells
● Deep Layer Excitatory Neurons
● Endothelial
● Inhibitory Neurons
● Intermediate Progenitors
● Maturing Neurons
● Maturing Upper Layer Neurons
● Microglia
● Migrating Neurons
● OPCs
● Outer Radial Glia
● Pericytes
● Ventral Radial Glia

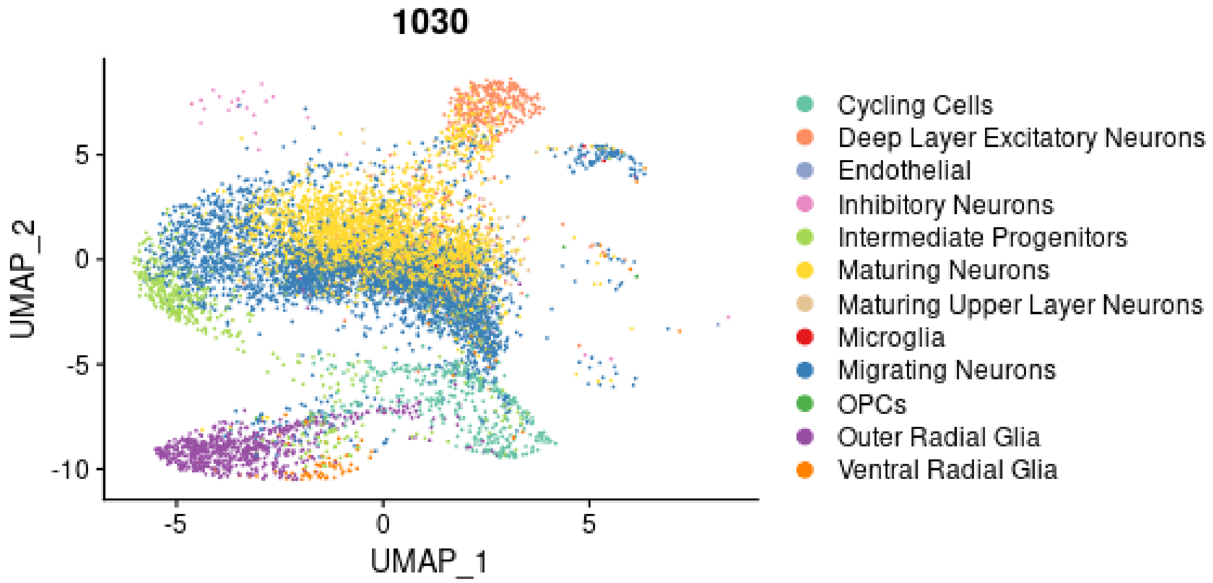

DUP9
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
● Cycling Cells
● Deep Layer Excitatory Neurons
● Endothelial
● Inhibitory Neurons
● Intermediate Progenitors
● Maturing Neurons
● Maturing Upper Layer Neurons
● Microglia
● Migrating Neurons
● OPCs
● Outer Radial Glia
● Pericytes
● Ventral Radial Glia


DEL3
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
● Cycling Cells
● Deep Layer Excitatory Neurons
● Endothelial
● Inhibitory Neurons
● Intermediate Progenitors
● Maturing Neurons
● Maturing Upper Layer Neurons
● Microglia
● Migrating Neurons
● OPCs
● Outer Radial Glia
● Pericytes
● Ventral Radial Glia

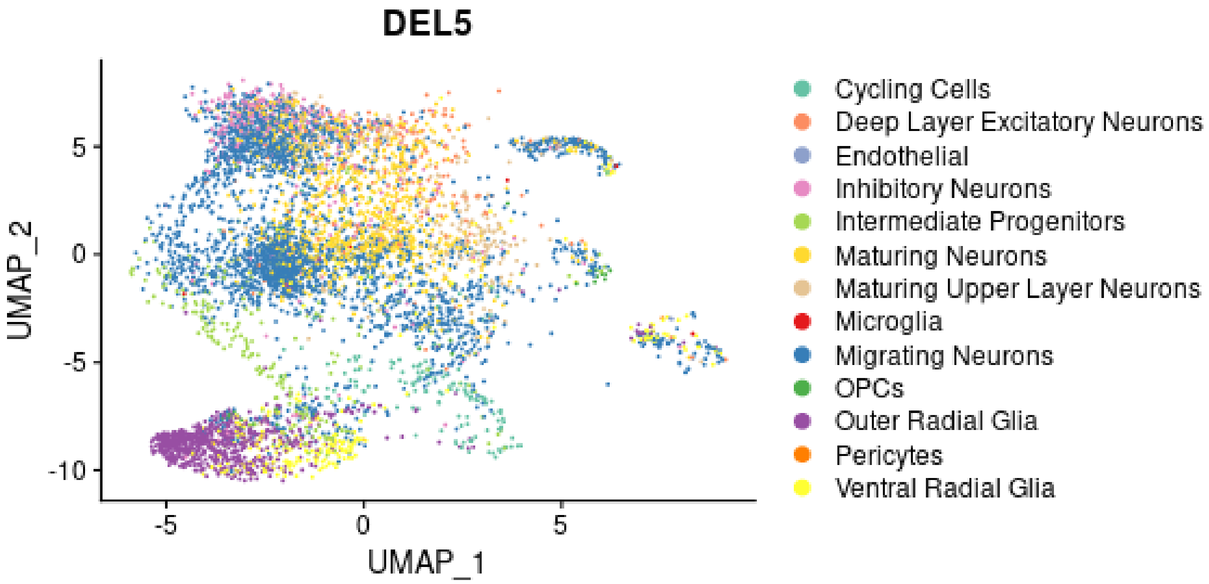

DUP1
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
● Cycling Cells
● Deep Layer Excitatory Neurons
● Endothelial
● Inhibitory Neurons
● Intermediate Progenitors
● Maturing Neurons
● Maturing Upper Layer Neurons
● Microglia
● Migrating Neurons
● OPCs
● Outer Radial Glia
● Pericytes
● Ventral Radial Glia

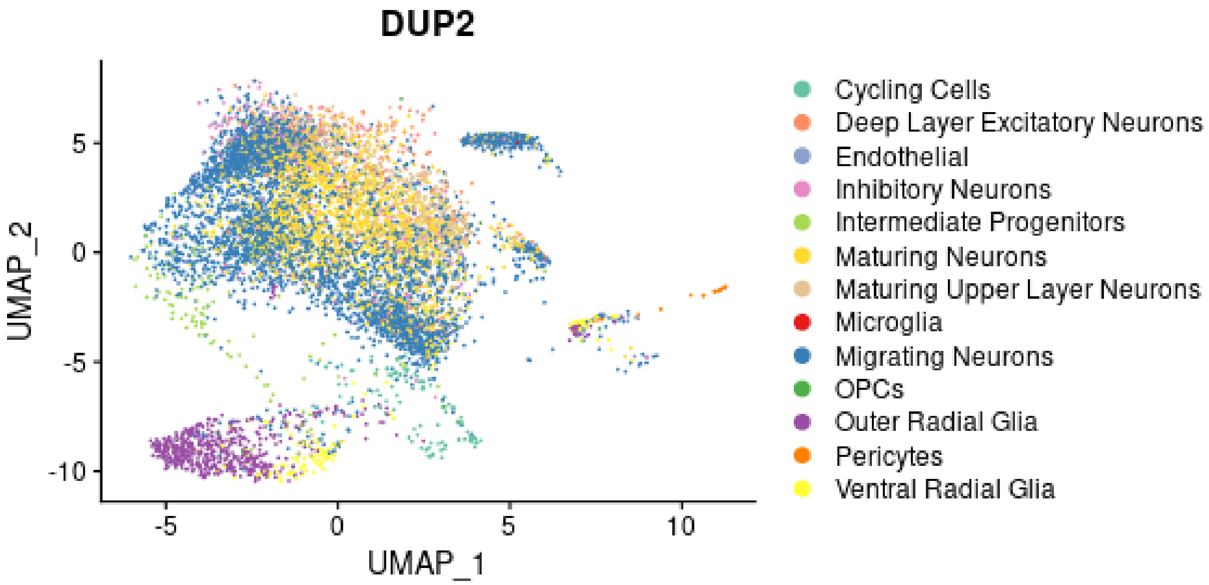

DEL6

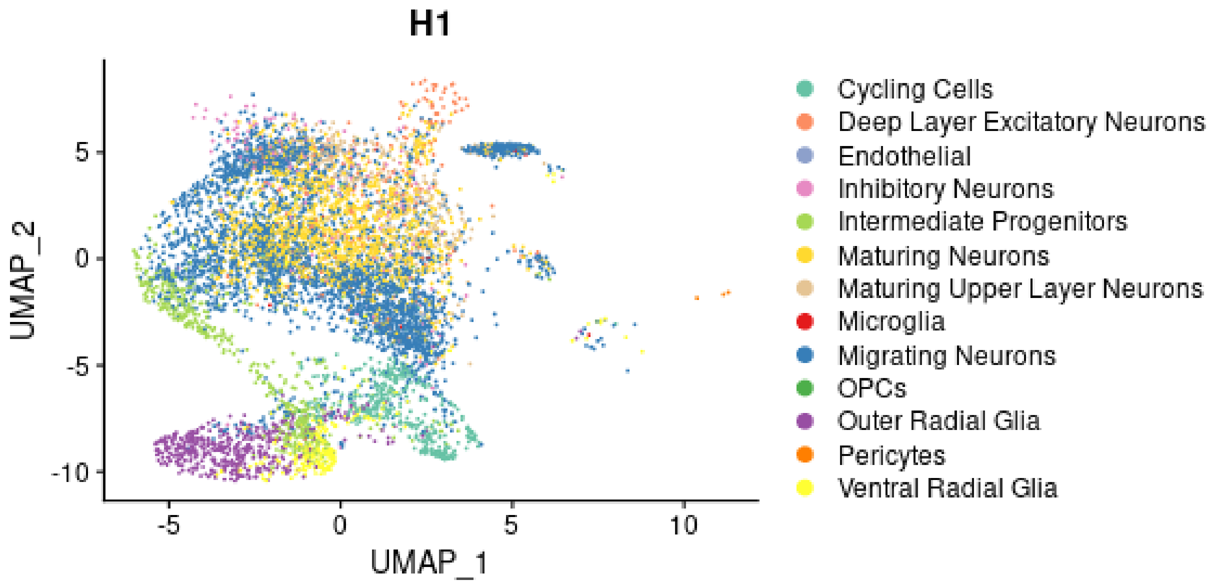
UMAP_2
5
0
-5
-10
-5
0
5
10
UMAP_1
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Microglia
Migrating Neurons
OPCs
Outer Radial Glia
Pericytes
Ventral Radial Glia


1030

UMAP_2
5
0
-5
-10
-5
0
5
UMAP_1
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Microglia
Migrating Neurons
OPCs
Outer Radial Glia
Ventral Radial Glia


DEL5
UMAP_2
UMAP_1
5
0
-5
-10
-5
0
5
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Microglia
Migrating Neurons
OPCs
Outer Radial Glia
Pericytes
Ventral Radial Glia


DUP2
UMAP_2
UMAP_1
5
0
-5
-10
-5
0
5
10
Cycling Cells
Deep Layer Excitatory Neurons
Endothelial
Inhibitory Neurons
Intermediate Progenitors
Maturing Neurons
Maturing Upper Layer Neurons
Microglia
Migrating Neurons
OPCs
Outer Radial Glia
Pericytes
Ventral Radial Glia



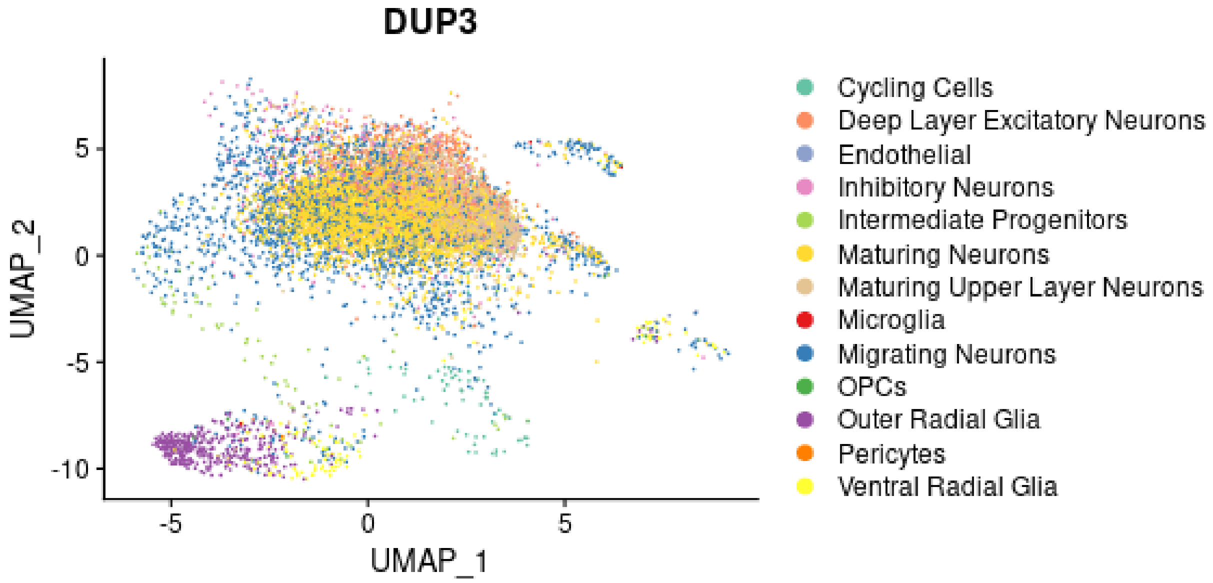
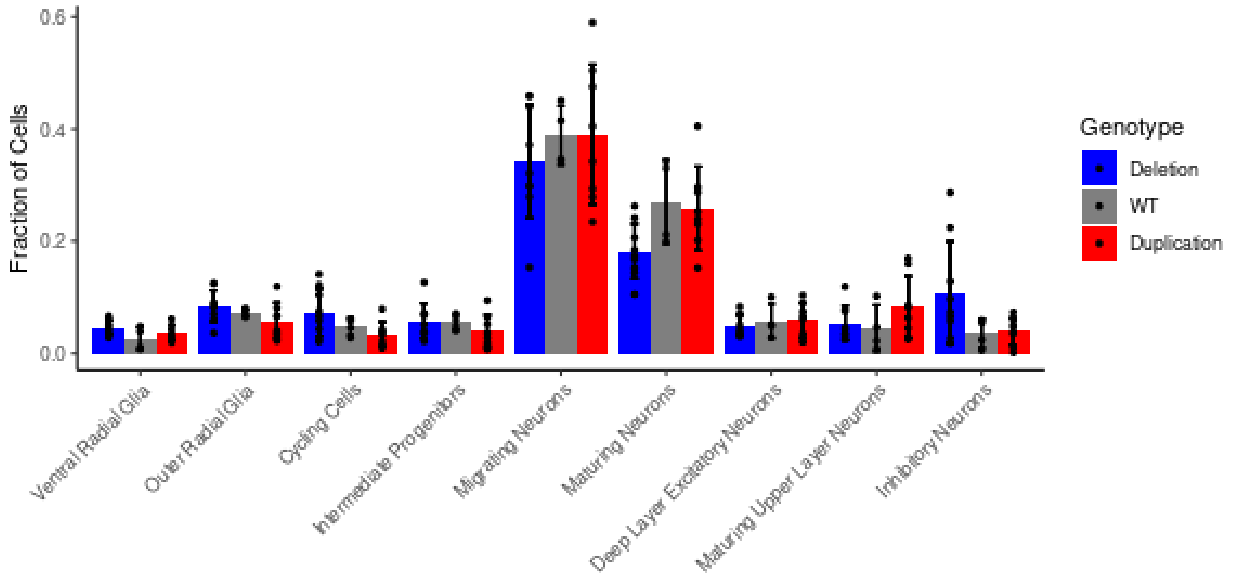


Figure 3C

```
# omitting cell types we aren't interested in and the duplicate samples from
# cell lines that were run more than once
cells.to.ignore <- c(JR.which.cells(dataset,
                                    meta.col = 'predicted.id', which = 'OPCs'),
        JR.which.cells(dataset, meta.col = 'predicted.id', which = 'Microglia'),
     JR.which.cells(dataset, meta.col = 'predicted.id', which = 'Endothelial'),
        JR.which.cells(dataset, meta.col = 'predicted.id', which = 'Pericytes'),
        JR.which.cells(dataset, meta.col = 'sample', which = 'H9_WT_90_2_35'),
        JR.which.cells(dataset, meta.col = 'sample', which = 'H9_WT_90_1_32'),
        JR.which.cells(dataset, meta.col = 'sample', which = 'H9_WT_90_2_34'),
        JR.which.cells(dataset, meta.col = 'sample', which = 'H9_WT_90_1_34'),
        JR.which.cells(dataset, meta.col = 'sample', which = 'H9_WT_90_1_35'),
        JR.which.cells(dataset, meta.col = 'sample', which = '8402_WT_90_2_35'))
cells.of.int <- colnames(dataset)[colnames(dataset) %in% cells.to.ignore == F]
meta <- dataset@meta.data[cells.of.int,]
# preferred order of samples for plot
stack.order <- c('Ventral Radial Glia', 'Outer Radial Glia', 'Cycling Cells',
            'Intermediate Progenitors', 'Migrating Neurons', 'Maturing Neurons',
               'Deep Layer Excitatory Neurons', 'Maturing Upper Layer Neurons',
         'Inhibitory Neurons', 'Microglia', 'Pericytes', 'OPCs', 'Endothelial')

#Getting cell counts for each genotype
meta.cast <- dcast(meta, sample + genotype ~ predicted.id)
del.cast <- meta.cast[meta.cast$genotype == 'DEL',]
rownames(del.cast) <- del.cast$sample
del.cast <- as.matrix(del.cast[,3:ncol(del.cast)])
dup.cast <- meta.cast[meta.cast$genotype == 'DUP',]
rownames(dup.cast) <- dup.cast$sample
dup.cast <- as.matrix(dup.cast[,3:ncol(dup.cast)])
wt.cast <- meta.cast[meta.cast$genotype == 'WT',]
rownames(wt.cast) <- wt.cast$sample
wt.cast <- as.matrix(wt.cast[,3:ncol(wt.cast)])
#converting to percentage
del.cast <- del.cast/rowSums(del.cast)
dup.cast <- dup.cast/rowSums(dup.cast)
wt.cast <- wt.cast/rowSums(wt.cast)

del.melt <- cbind(melt(del.cast), 'Genotype' = 'Deletion')
dup.melt <- cbind(melt(dup.cast), 'Genotype' = 'Duplication')
wt.melt <- cbind(melt(wt.cast), 'Genotype' = 'WT')
gg.points <- rbind(del.melt, dup.melt, wt.melt)
colnames(gg.points) <- c('Sample', 'Celltype', 'Cell Fraction', 'Genotype')

del.means <- colMeans(del.cast)
del.sds <- colSds(del.cast)
names(del.sds) <- names(del.means)
dup.means <- colMeans(dup.cast)
dup.sds <- colSds(dup.cast)
names(dup.sds) <- names(dup.means)
wt.means <- colMeans(wt.cast)
wt.sds <- colSds(wt.cast)
names(wt.sds) <- names(wt.means)

gg.inp1 <- cbind('Deletion' = del.means,
                 'Duplication' = dup.means, 'WT' = wt.means) %>% melt()
gg.inp2 <- cbind('Deletion' = del.sds,
                 'Duplication' = dup.sds, 'WT' = wt.sds) %>% melt()
gg.inp <- cbind(gg.inp1, gg.inp2$value)
colnames(gg.inp) <- c('Celltype', 'Genotype', 'Mean Cell Fraction',
                      'Cell Number SD')
gg.inp$join <- paste0(gg.inp$Celltype, gg.inp$Genotype)
gg.inp <- dplyr::select(gg.inp, 'join', 'Mean Cell Fraction', 'Cell Number SD')
gg.points$join <- paste0(gg.points$Celltype, gg.points$Genotype)
```

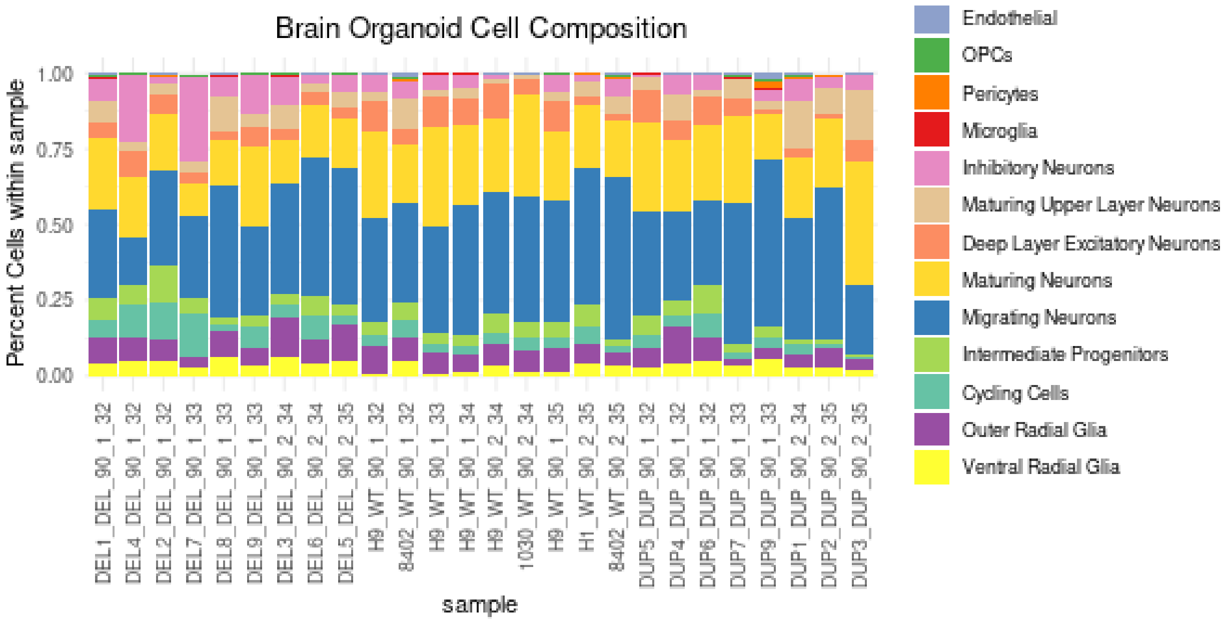
```r
gg.inp <- inner_join(x = gg.points, y = gg.inp, by = c('join'))
gg.inp <- dplyr::mutate(gg.inp, 'Genotype' = factor(Genotype,
                              levels = c('Deletion', 'WT', 'Duplication'))) %>%
  dplyr::mutate('Celltype' = factor(Celltype, levels = stack.order))
rm(gg.inp1, gg.inp2, del.means, del.sds, dup.means, dup.sds, wt.means,
   wt.sds, wt.cast, dup.cast, del.cast, gg.points)

gg <- ggplot(gg.inp, aes(x = as.factor(Celltype), y = `Mean Cell Fraction`,
                         fill = as.factor(Genotype))) +
    geom_col(aes(x = as.factor(Celltype), y = `Mean Cell Fraction`,
                 fill = as.factor(Genotype)),
             position = position_dodge()) +
    geom_errorbar(aes(ymin = (`Mean Cell Fraction` - `Cell Number SD`),
                      ymax = (`Mean Cell Fraction` + `Cell Number SD`),
                  fill = as.factor(Genotype)),
                  position= position_dodge(.9),
                  width = .2) +
    geom_jitter(position = position_dodge(width = .9),
               aes(x = Celltype, y = `Cell Fraction`), color = 'black',
               alpha = 1, size = 1) +
    theme_classic() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    xlab('') +
    ylab('Fraction of Cells') +
    labs(fill = 'Genotype') +
    scale_fill_manual(values = c('blue', 'gray50', 'red'))
gg
```



```r
to.write <- dplyr::select(gg.inp, 'Sample', 'Celltype', 'Cell Fraction',
                          'Genotype', 'Mean Cell Fraction', 'Cell Number SD')

rm(gg.inp, meta, meta.cast, stack.order, cells.of.int, cells.to.ignore)
```

Figure 3D

```
name.order <- c('Cycling Cells', 'Deep Layer Excitatory Neurons', 'Endothelial',
                'Inhibitory Neurons', 'Intermediate Progenitors', 'Maturing Neurons',
                'Maturing Upper Layer Neurons', 'Microglia', 'Migrating Neurons',
                'OPCs', 'Outer Radial Glia', 'Pericytes', 'Ventral Radial Glia')
new.order <- c('Ventral Radial Glia', 'Outer Radial Glia', 'Cycling Cells',
               'Intermediate Progenitors', 'Migrating Neurons', 'Maturing Neurons',
               'Deep Layer Excitatory Neurons', 'Maturing Upper Layer Neurons',
               'Inhibitory Neurons', 'Microglia', 'Pericytes', 'OPCs', 'Endothelial')
new.colors <- c()
for(i in new.order){
  cols <- match(i, name.order)
  new.colors <- c(new.colors, colors[cols])
}
stack.order <- new.order
dup.samples <- grep(pattern = 'DUP', x = unique(dataset$sample), value = T)
wt.samples <- grep(pattern = 'WT', x = unique(dataset$sample), value = T)
del.samples <- grep(pattern = 'DEL', x = unique(dataset$sample), value = T)

gg <- JR.bar.plot(meta = dataset@meta.data, x.axis = 'sample',
                  grouping = 'predicted.id', cols = rev(new.colors),
                  grouping_style = 'stack',
                  x.axis.order = c(del.samples, wt.samples, dup.samples),
                  grouping.order = rev(stack.order))
gg + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  ggtitle('Brain Organoid Cell Composition') +
  theme(plot.title = element_text(hjust = 0.5))
```



```
rm(stack.order, dup.samples, wt.samples, del.samples)
```

Figure 3E

```
# subsetting to only include one sample from each cell line
cells.to.ignore <- c(JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_2_35'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_1_32'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_2_34'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_1_34'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_1_35'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = '8402_WT_90_2_35'))
cells.of.int <- colnames(dataset)[colnames(dataset) %in% cells.to.ignore == F]
dataset.sub <- subset(dataset, cells = cells.of.int)
# cell types we were interested in looking at
cell.types.of.int <- c('Cycling Cells', 'Deep Layer Excitatory Neurons',
                       'Inhibitory Neurons', 'Intermediate Progenitors',
                       'Maturing Neurons', 'Maturing Upper Layer Neurons',
                       'Migrating Neurons', 'Outer Radial Glia',
                       'Ventral Radial Glia')
# matching order with rest of figure
new.order <- c('Ventral Radial Glia', 'Outer Radial Glia', 'Cycling Cells',
          'Intermediate Progenitors', 'Migrating Neurons', 'Maturing Neurons',
          'Deep Layer Excitatory Neurons', 'Maturing Upper Layer Neurons',
          'Inhibitory Neurons', 'Microglia', 'Pericytes', 'OPCs', 'Endothelial')
new.order <- new.order[new.order %in% cell.types.of.int]
# calculating prediction score distribution for each cell type within each
# genotype from the metadata in the Seurat object.
meta <- dataset.sub@meta.data
meta <- meta %>% select(predicted.id, prediction.score.max, sample) %>%
  dplyr::filter(predicted.id %in% cell.types.of.int) %>%
  melt(value.name = 'Prediction Score') %>%
  dcast(sample ~ predicted.id, fun.aggregate = mean) %>%
  melt(value.name = 'Prediction Score') %>%
  mutate('Cell Type' = variable) %>%
  separate(col = 'sample', sep = '_', remove = F,
           into = c('cellline', 'Genotype', 'age', 'batch', 'exp')) %>%
  mutate(Genotype = factor(Genotype, levels = c('DEL', 'WT', 'DUP'))) %>%
  mutate('Cell Type' = factor(`Cell Type`, levels = new.order))
# plotting the result
gg <- ggplot(meta) +
  geom_boxplot(aes(x = `Cell Type`, y = `Prediction Score`, color = Genotype),
               alpha = 5, weight = .5, position = position_dodge(width = 1)) +
  geom_jitter(position = position_dodge(width = 1),
               aes(x = `Cell Type`, y = `Prediction Score`, color = Genotype),
               alpha = .6, size = 1) +
  ggtitle('') + theme_classic() +
  theme(axis.text.x = element_text(angle = 90, vjust = .5,
                                   hjust = .9, size = 11),
        axis.title.x = element_blank()) +
  ylim(0,1) +
  theme(legend.title = element_blank()) +
  theme(legend.text = element_text(size = 14)) +
  theme(plot.title = element_text(hjust = .5, size = 16)) +
  scale_fill_manual(values = c('blue', 'gray50', 'red')) +
  scale_color_manual(values = c('blue', 'gray50', 'red'))
gg
```

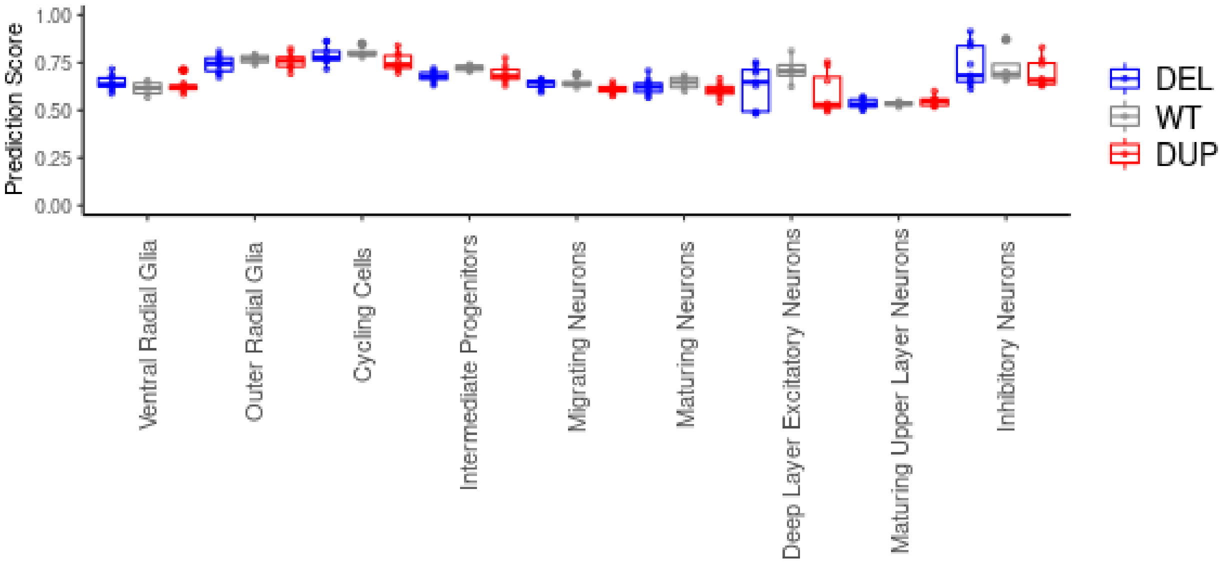
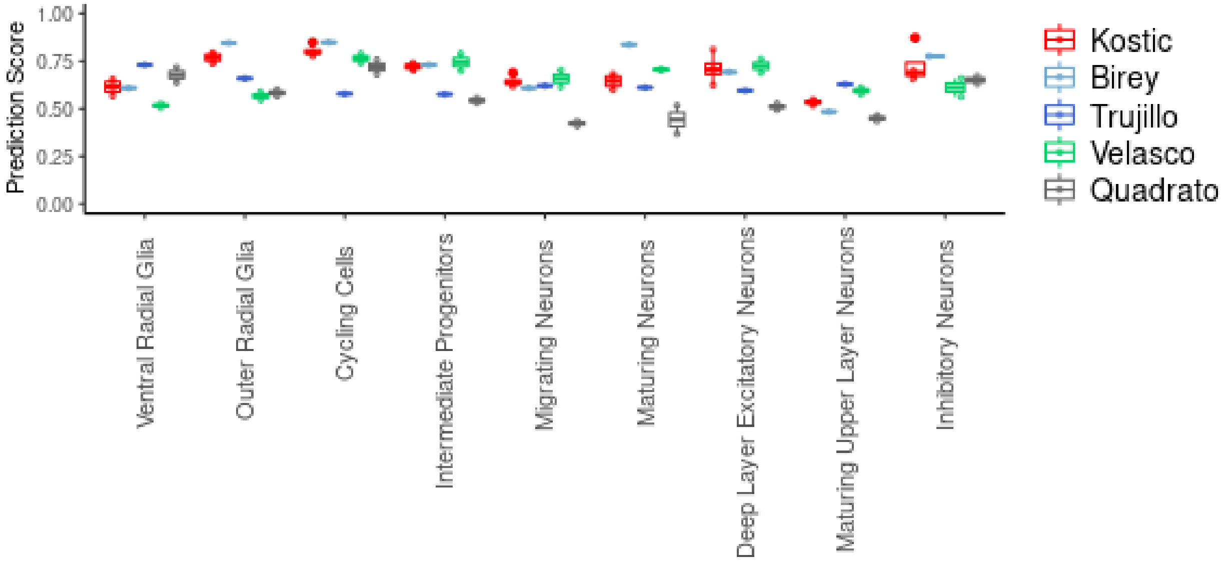


Figure 3F

```
# subsetting to one sample per cell line for analysis
cells.to.ignore <- c(JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_2_35'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_1_32'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_2_34'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_1_34'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = 'H9_WT_90_1_35'),
                     JR.which.cells(dataset, meta.col = 'sample',
                                    which = '8402_WT_90_2_35'))
cells.of.int <- colnames(dataset)[colnames(dataset) %in% cells.to.ignore == F]
dataset.sub <- subset(dataset, cells = cells.of.int)
# only looking at cell types of interest from previous figures
cell.types.of.int <- c('Cycling Cells', 'Deep Layer Excitatory Neurons',
                       'Inhibitory Neurons', 'Intermediate Progenitors',
                       'Maturing Neurons', 'Maturing Upper Layer Neurons',
                       'Migrating Neurons', 'Outer Radial Glia',
                       'Ventral Radial Glia')
```

```r
# matching cell type order from previous figures
new.order <- c('Ventral Radial Glia', 'Outer Radial Glia', 'Cycling Cells',
          'Intermediate Progenitors', 'Migrating Neurons', 'Maturing Neurons',
          'Deep Layer Excitatory Neurons', 'Maturing Upper Layer Neurons',
          'Inhibitory Neurons', 'Microglia', 'Pericytes', 'OPCs', 'Endothelial')
new.order <- new.order[new.order %in% cell.types.of.int]
# calculating prediction score distribution for each cell type within each
# genotype from the metadata in the Seurat object.
meta <- dataset.sub@meta.data
meta <- meta %>% dplyr::select(predicted.id, prediction.score.max, sample) %>%
  dplyr::filter(predicted.id %in% cell.types.of.int) %>%
  melt(value.name = 'Prediction Score') %>%
  dcast(sample ~ predicted.id, fun.aggregate = mean) %>%
  melt(value.name = 'Prediction Score') %>%
  mutate('Cell Type' = variable) %>%
  separate(col = 'sample', sep = '_', remove = F,
           into = c('cellline', 'Genotype', 'age', 'batch', 'exp')) %>%
  mutate(Genotype = factor(Genotype, levels = c('DEL', 'WT', 'DUP'))) %>%
  mutate('Cell Type' = factor(`Cell Type`, levels = new.order)) %>%
  dplyr::filter(Genotype == 'WT') %>%
  mutate('Study' = 'Kostic')
stash.meta <- meta

# repeating the above process for the external data
load(ext.dat.path)
cell.types.of.int <- c('Cycling Cells', 'Deep Layer Excitatory Neurons',
                       'Inhibitory Neurons', 'Intermediate Progenitors',
                       'Maturing Neurons', 'Maturing Upper Layer Neurons',
                       'Migrating Neurons', 'Outer Radial Glia',
                       'Ventral Radial Glia')
new.order <- c('Ventral Radial Glia', 'Outer Radial Glia', 'Cycling Cells',
          'Intermediate Progenitors', 'Migrating Neurons', 'Maturing Neurons',
          'Deep Layer Excitatory Neurons', 'Maturing Upper Layer Neurons',
          'Inhibitory Neurons', 'Microglia', 'Pericytes', 'OPCs', 'Endothelial')
new.order <- new.order[new.order %in% cell.types.of.int]
meta <- dataset@meta.data
meta <- meta %>% dplyr::select(predicted.id, prediction.score.max, Dataset) %>%
  dplyr::filter(predicted.id %in% cell.types.of.int) %>%
  melt(value.name = 'Prediction Score') %>%
  dcast(Dataset ~ predicted.id, fun.aggregate = mean) %>%
  melt(value.name = 'Prediction Score') %>%
  mutate('Cell Type' = variable) %>%
  mutate('Cell Type' = factor(`Cell Type`, levels = new.order)) %>%
  separate(col = 'Dataset', into = c('Study', 'age', 'sample'), remove = F) %>%
  dplyr::select(sample, Study, variable, `Prediction Score`, `Cell Type`)
meta[is.na(meta$sample),]$sample <- '1'

# merging data
stash.meta <- dplyr::select(stash.meta, sample, Study, variable,
                            `Prediction Score`, `Cell Type`)
meta <- rbind(stash.meta, meta)
study.order <- c('Kostic', 'Birey', 'Trujillo', 'Velasco', 'Quadrato')
meta <- dplyr::mutate(meta, 'Study' = factor(Study, levels = study.order))
```

```r
# plotting results
gg <- ggplot(meta) +
  geom_boxplot(aes(x = `Cell Type`, y = `Prediction Score`, color = Study),
               alpha = 5, weight = .5, position = position_dodge(width = .8)) +
  geom_jitter(position = position_dodge(width = .8),
               aes(x = `Cell Type`, y = `Prediction Score`, color = Study),
               alpha = .6, size = 1) +
  ggtitle('') + theme_classic() +
  theme(axis.text.x = element_text(angle = 90, vjust = .5,
                                   hjust = .9, size = 11),
        axis.title.x = element_blank()) +
  ylim(0,1) +
  theme(legend.title = element_blank()) +
  theme(legend.text = element_text(size = 14)) +
  theme(plot.title = element_text(hjust = .5, size = 16)) +
  scale_fill_manual(values = c('red', 'skyblue3', 'royalblue3',
                               'springgreen3', 'gray40')) +
  scale_color_manual(values = c('red', 'skyblue3', 'royalblue3',
                                'springgreen3', 'gray40'))
gg
```



```r
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: CentOS Linux 7 (Core)
##
## Matrix products: default
## BLAS/LAPACK: /usr/prog/OpenBLAS/0.2.20-GCC-6.4.0-2.28/lib/libopenblas_haswellp-r0.2.20.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] tidyr_1.1.3        matrixStats_0.58.0 RColorBrewer_1.1-2 dplyr_1.0.5
## [5] reshape2_1.4.4     ggplot2_3.3.3      SeuratObject_4.0.0 Seurat_4.0.0
##
## loaded via a namespace (and not attached):
##   [1] nlme_3.1-152         RcppAnnoy_0.0.18     httr_1.4.2
##   [4] sctransform_0.3.2    tools_4.0.2          utf8_1.1.4
##   [7] R6_2.5.0             irlba_2.3.3          rpart_4.1-15
##  [10] KernSmooth_2.23-18   uwot_0.1.10          mgcv_1.8-34
##  [13] DBI_1.1.1            lazyeval_0.2.2       colorspace_2.0-0
##  [16] withr_2.4.1          tidyselect_1.1.0     gridExtra_2.3
##  [19] compiler_4.0.2       plotly_4.9.3         labeling_0.4.2
##  [22] scales_1.1.1         spatstat.data_2.0-0  lmtest_0.9-38
##  [25] ggridges_0.5.3       pbapply_1.4-3        goftest_1.2-2
##  [28] spatstat_1.64-1      stringr_1.4.0        digest_0.6.27
##  [31] spatstat.utils_2.0-0 rmarkdown_2.7        pkgconfig_2.0.3
##  [34] htmltools_0.5.1.1    parallelly_1.23.0    highr_0.8
##  [37] fastmap_1.1.0        htmlwidgets_1.5.3    rlang_0.4.10
##  [40] shiny_1.6.0          farver_2.1.0         generics_0.1.0
##  [43] zoo_1.8-8            jsonlite_1.7.2       ica_1.0-2
##  [46] magrittr_2.0.1       patchwork_1.1.1      Matrix_1.3-2
##  [49] Rcpp_1.0.6           munsell_0.5.0        fansi_0.4.2
##  [52] abind_1.4-5          reticulate_1.18      lifecycle_1.0.0
##  [55] stringi_1.5.3        yaml_2.2.1           MASS_7.3-53.1
##  [58] Rtsne_0.15           plyr_1.8.6           grid_4.0.2
##  [61] parallel_4.0.2       listenv_0.8.0        promises_1.2.0.1
##  [64] ggrepel_0.9.1        crayon_1.4.1         deldir_0.2-10
##  [67] miniUI_0.1.1.1       lattice_0.20-41      cowplot_1.1.1
##  [70] splines_4.0.2        tensor_1.5           knitr_1.31
##  [73] pillar_1.5.1         igraph_1.2.6         future.apply_1.7.0
##  [76] codetools_0.2-18     leiden_0.3.7         glue_1.4.2
##  [79] evaluate_0.14        data.table_1.14.0    vctrs_0.3.6
##  [82] png_0.1-7            httpuv_1.5.5         polyclip_1.10-0
##  [85] gtable_0.3.0         RANN_2.6.1           purrr_0.3.4
##  [88] scattermore_0.7      future_1.21.0        assertthat_0.2.1
##  [91] xfun_0.21            mime_0.10            xtable_1.8-4
##  [94] later_1.1.0.1        survival_3.2-7       viridisLite_0.3.0
##  [97] tibble_3.1.0         cluster_2.1.1        globals_0.14.0
## [100] fitdistrplus_1.1-3   ellipsis_0.3.1       ROCR_1.0-11
```