

Butt et al: scRNASeq analysis - Final analyses

Agnes A. Steixner-Kumar

20 11 2020

Contents

1	Load libraries and data	1
1.1	Load libraries	1
1.2	Load data	1
2	Analysis	2
2.1	Figure 4A	2
2.2	Figure 4B	3
2.3	Figure 4C	5
2.4	Figure 4D	11
2.5	Differential expression testing	14
2.6	Supplementary info	14
2.7	Session info	18

1 Load libraries and data

1.1 Load libraries

```
library(reshape2)
library(RColorBrewer)
library(grid)
library(Seurat) #v3.0.0
library(cowplot)
library(dplyr)
library(ggplot2)
library(scales)
```

1.2 Load data

This is the final .RDS object as uploaded on GEO (GSExxx)

```
ODD<-readRDS("F:/GEO_submission_Butt_et_al/Butt_et_al_hypoxia_Seurat.RDS")
```

2 Analysis

2.1 Figure 4A

2.1.1 Define colors

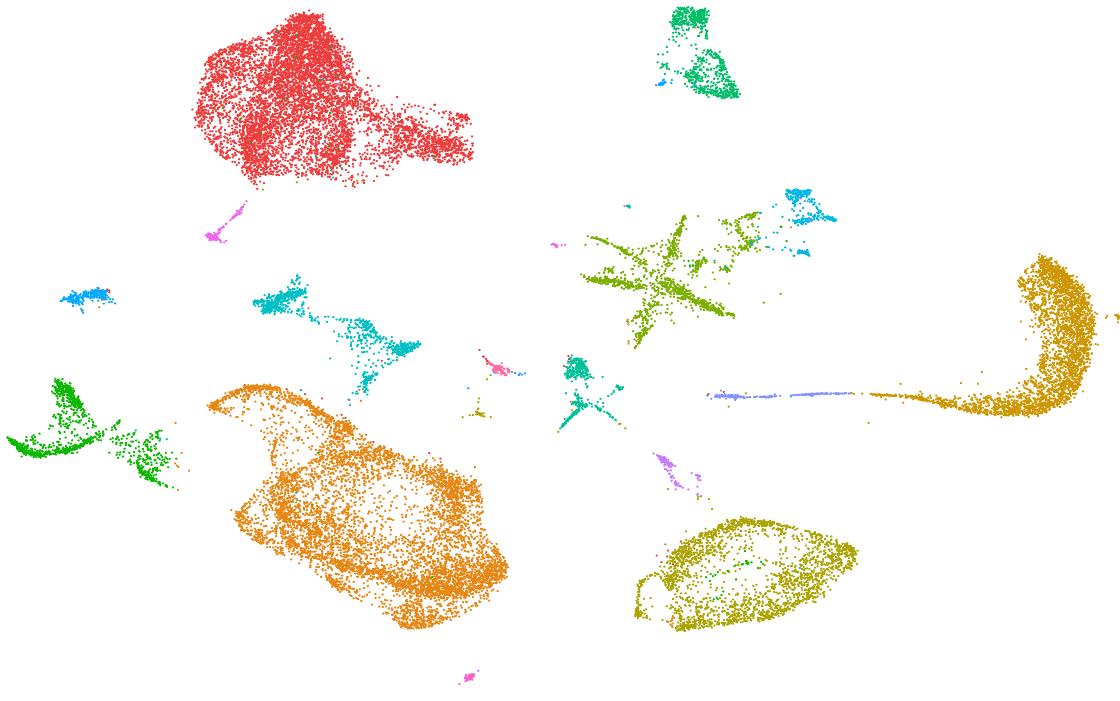
```
identities <- levels(Idents(ODD))
my_color_palette <- hue_pal()(length(identities))
my_color_palette <-c('brown2', my_color_palette[2:16])
```

2.1.2 Shift ependymal cluster upwards

```
emb<-Embeddings(ODD, reduction = "umap")
#shift selected cells upwards on UMAP2
emb[,2][emb[,2]<(-15)]<-emb[,2][emb[,2]<(-15)]+12
#feed changed embeddings back into object
ODD@reductions$umap@cell.embeddings<-emb
```

2.1.3 Make plot of UMAP embedding

```
DimPlot(ODD, reduction = 'umap', label = F, repel = T, cols = my_color_palette)+  
  NoLegend() +  
  NoAxes()
```



2.2 Figure 4B

2.2.1 Create dataframe of ODD &/or tdTomato+ cells

```

# prepare dataframe
ODD_tdTom_cluster<-data.frame(matrix(nrow = ncol(ODD), ncol = 5), row.names = colnames(ODD))
colnames(ODD_tdTom_cluster)=c("Cluster", "Group", "ODD", "tdTomato", "Expression")
# fill in metadata
ODD_tdTom_cluster$Cluster<-Idents(ODD)
ODD_tdTom_cluster$Sample<-ODD$Sample
ODD_tdTom_cluster$Group<-ODD$group
# get expression values
ODD_tdTom_cluster$ODD<-GetAssayData(ODD) [ "ODD",]
ODD_tdTom_cluster$tdTomato<-GetAssayData(ODD) [ "tdTomato",]
# recode
ODD_tdTom_cluster$Expression[ODD_tdTom_cluster$ODD==0 & ODD_tdTom_cluster$tdTomato==0]<- "none"
ODD_tdTom_cluster$Expression[ODD_tdTom_cluster$ODD>0 & ODD_tdTom_cluster$tdTomato==0]<- "ODD"
ODD_tdTom_cluster$Expression[ODD_tdTom_cluster$ODD==0 & ODD_tdTom_cluster$tdTomato>0]<- "tdTomato"
ODD_tdTom_cluster$Expression[ODD_tdTom_cluster$ODD>0 & ODD_tdTom_cluster$tdTomato>0]<- "both"

```

2.2.2 Count positive cells

```

# create count table
ODD_tdTom_cluster_counts<-ODD_tdTom_cluster %>%
  group_by(Cluster, Expression, Group) %>%

```

```

  count()
# order levels for display
ODD_tdTom_cluster_counts$Expression<-factor(ODD_tdTom_cluster_counts$Expression,
                                             levels=c("none", "both", "tdTomato", "ODD"))

```

2.2.3 Clean up cluster labels

```

cluster_labels<-as.character(unique(ODD_tdTom_cluster_counts$Cluster))
cluster_labels<-gsub('_', ' ', cluster_labels)
cluster_labels<-gsub('cells', '', cluster_labels)
cluster_labels<-gsub(' $', '', cluster_labels)
cluster_labels

## [1] "Glutamatergic0"    "Glutamatergic1"    "Oligodendrocytes" "Glutamatergic2"
## [5] "Endothelial"        "Astrocytes"       "Gabaergic"        "Microglia"
## [9] "Glutamatergic3"    "Mural"           "Glutamatergic4"   "OPC"
## [13] "Mossy"             "Red blood"        "Ependymal"        "Neuroblasts"

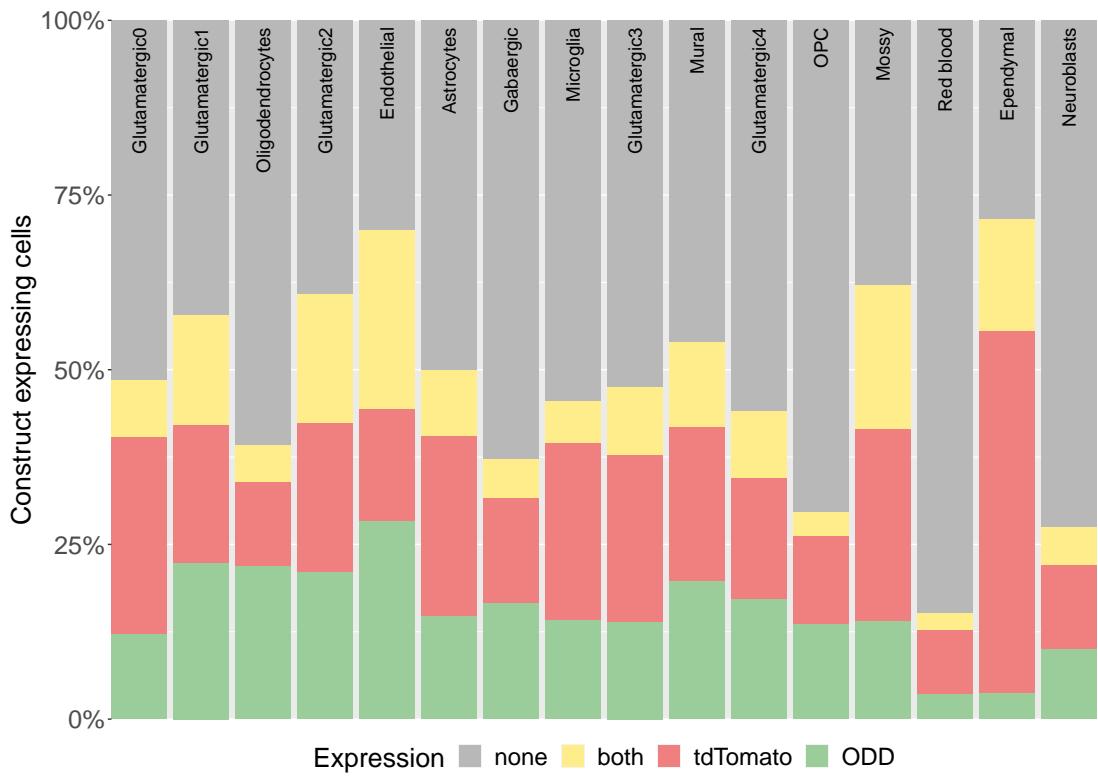
```

2.2.4 Plot proportions

```

ggplot(ODD_tdTom_cluster_counts, aes(y=n, x=Cluster, fill=Expression)) +
  geom_bar( stat="identity", position="fill")+
  scale_y_continuous(labels= scales::percent)+ 
  ylab("Construct expressing cells")+
  coord_cartesian(expand = FALSE)+ 
  scale_fill_manual(values=c("gray72", "lightgoldenrod1", "lightcoral", "darkseagreen3"))+
  annotate(geom = 'text',size=5, label=cluster_labels, y=0.99, x=seq(0:15),
           angle=90, hjust=1)+ 
  theme(axis.title.x=element_blank(),axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        axis.text.y=element_text(size=19), axis.title=element_text(size=20),
        legend.text=element_text(size=18),
        legend.title=element_text(size=20),legend.position = "bottom")+
  theme(plot.margin=unit(c(0.3,3.6,0,0.7), "cm"))

```



2.3 Figure 4C

2.3.1 Split objects by group

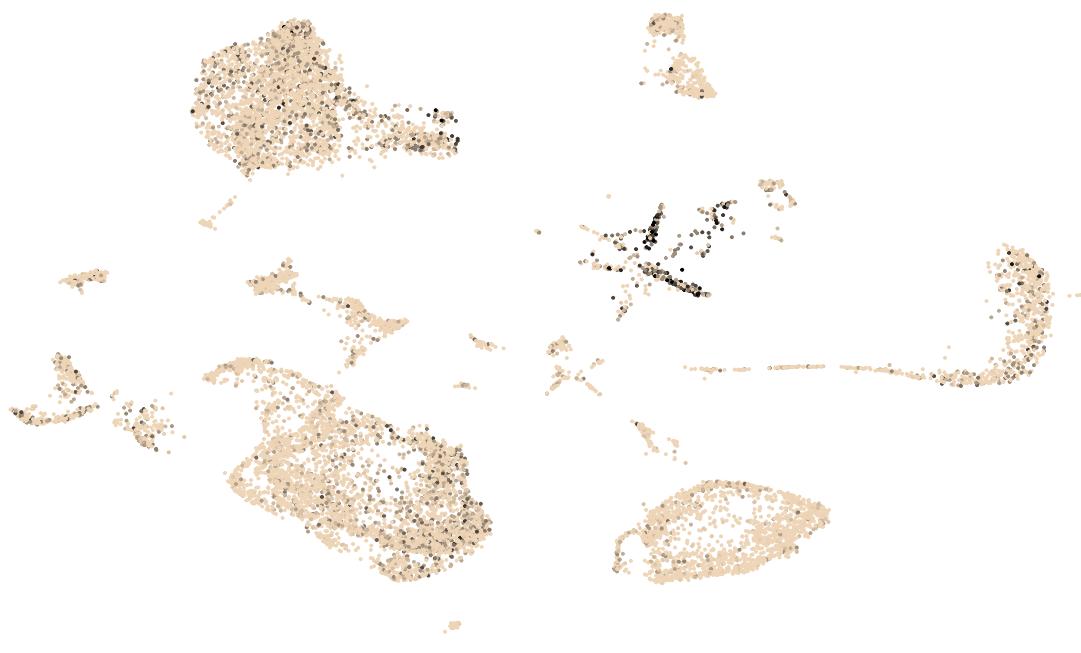
```
Idents(ODD)<-ODD$group
ODD_norm<-subset(ODD, idents = "Normoxia")
ODD_hyp<-subset(ODD, idents = "Hypoxia")
```

2.3.2 Make normoxia feature plots

```
lapply(c("ODD", "tdTomato", "Vegfa"), function(x) FeaturePlot(
  ODD_norm, features = x, pt.size = 0.7, order=F, min.cutoff = 0.5,
  max.cutoff = 2, cols = (c("bisque2", "black")))+
  NoAxes()+
  ggtitle(x)+
  NoLegend())
```

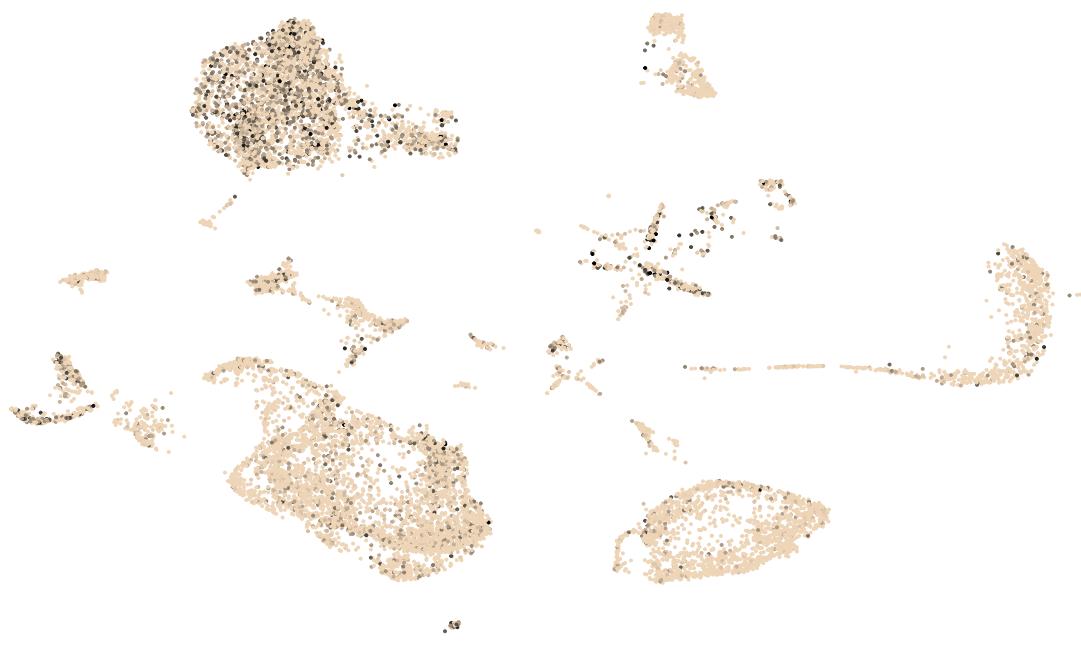
```
## [[1]]
```

ODD



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

tdTomato



```
##
```

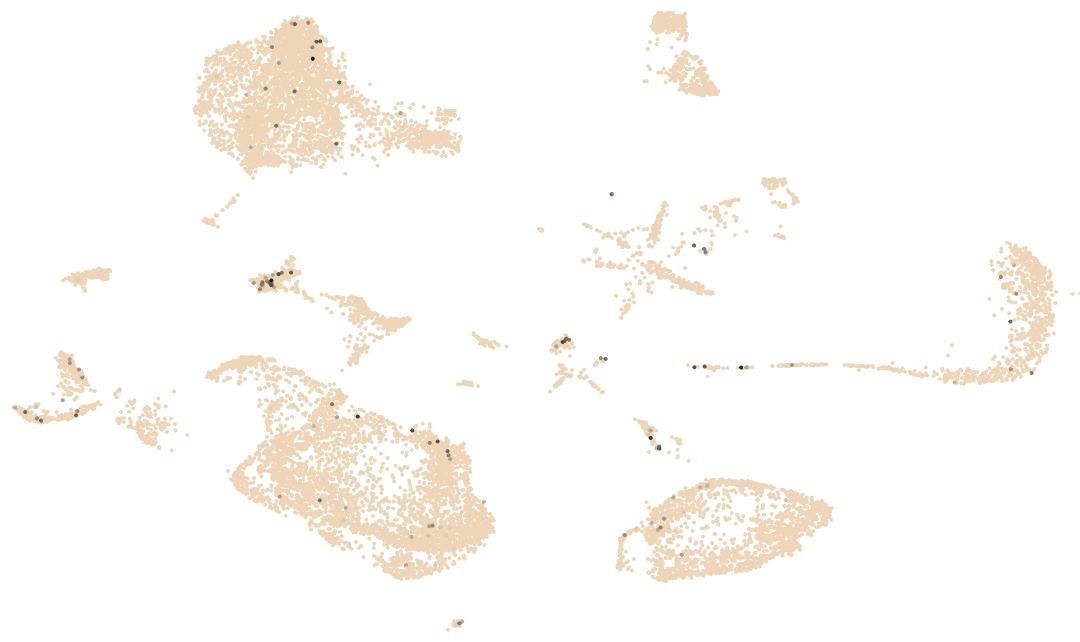
```
## [[3]]
```

```
Vegfa
```



```
# order cells with regard to expression to prevent masking
FeaturePlot(ODD_norm, features = "Hk2", pt.size = 0.7, order=T, min.cutoff = 0.5,
            max.cutoff = 2, cols = c("bisque2", "black"))+
  NoAxes()+
  ggtitle("Hk2")+
  NoLegend()
```

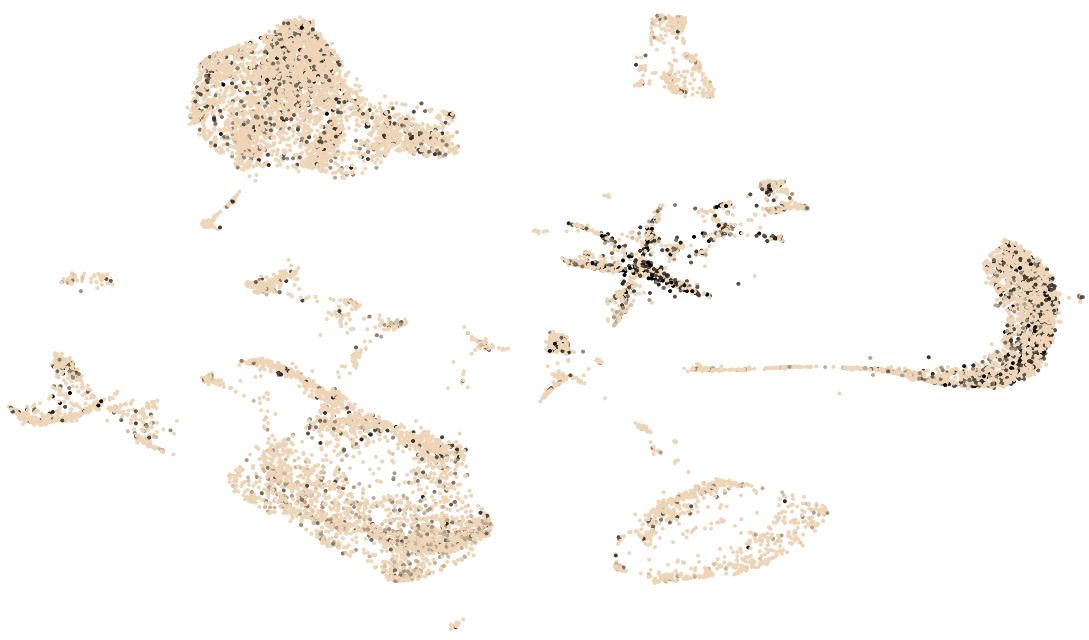
Hk2



2.3.3 Make hypoxia feature plots

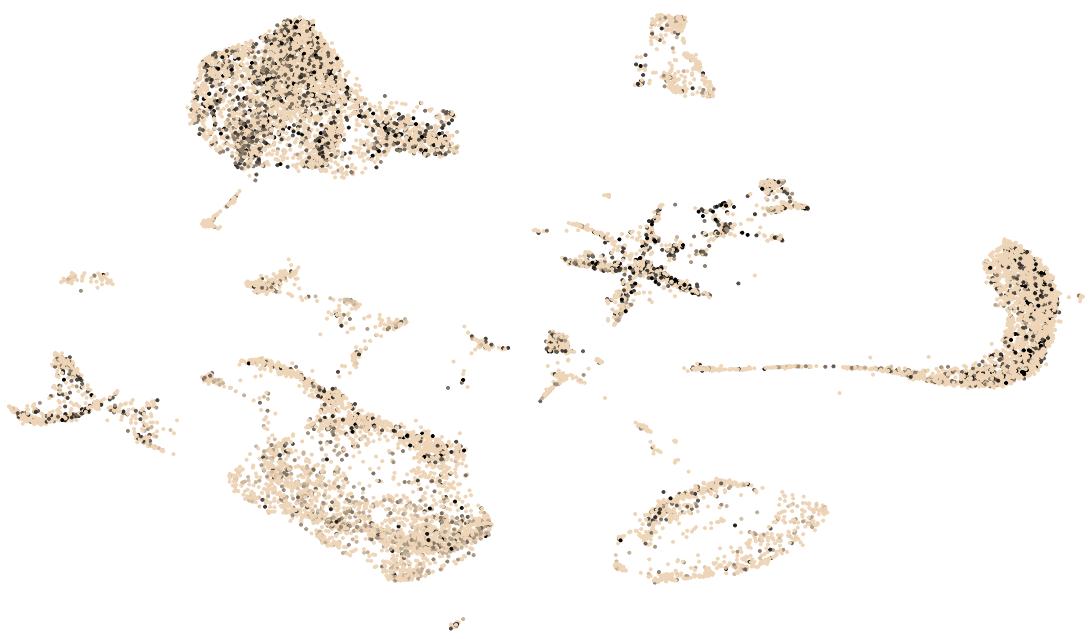
```
lapply(c("ODD", "tdTomato", "Vegfa"), function(x) FeaturePlot(  
  ODD_hyp, features = x, pt.size = 0.7, order=F, min.cutoff = 0.5,  
  max.cutoff = 2, cols = (c("bisque2", "black")))+  
  NoAxes() +  
  ggtitle(x) +  
  NoLegend())  
  
## [[1]]
```

ODD



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

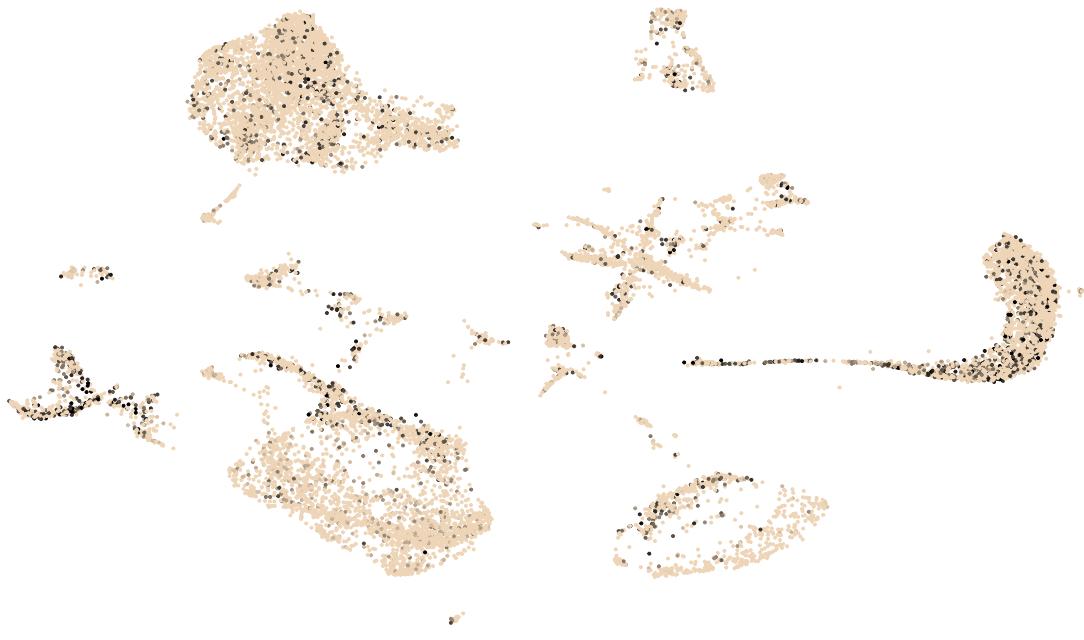
tdTomato



```
##
```

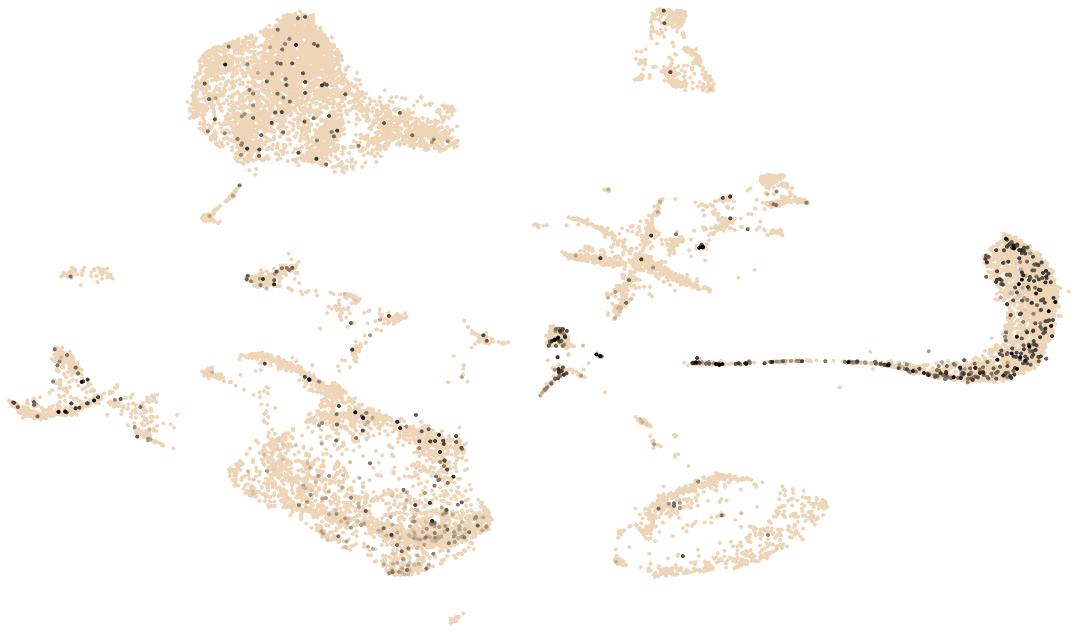
```
## [[3]]
```

Vegfa



```
# order cells with regard to expression to prevent masking
FeaturePlot(ODD_hyp, features = "Hk2", pt.size = 0.7, order=T, min.cutoff = 0.5,
            max.cutoff = 2, cols = c("bisque2", "black"))+
  NoAxes()+
  ggtitle("Hk2")+
  NoLegend()
```

Hk2



2.4 Figure 4D

2.4.1 Create rough identity clusters

These cell types correspond to cell types distinguished in immunofluorescence quantifications

```
ODD$rough_ident<-NA
# merge all neuronal types
ODD$rough_ident[ODD$final_identity=="Glutamatergic1" | ODD$final_identity=="Glutamatergic0" |
                 ODD$final_identity=="Glutamatergic2" | ODD$final_identity=="Glutamatergic3" |
                 ODD$final_identity=="Glutamatergic4" | ODD$final_identity=="Gabaergic" |
                 ODD$final_identity=="Mossy_cells"]<-"Neurons"
ODD$rough_ident[ODD$final_identity=="Astrocytes"]<-"Astrocytes"
ODD$rough_ident[ODD$final_identity=="Oligodendrocytes" | ODD$final_identity=="OPC"]<-"OPC, Oligo"
ODD$rough_ident[ODD$final_identity=="Microglia"]<-"Microglia"
ODD$rough_ident[ODD$final_identity=="Endothelial"]<-"Endothelial"
```

2.4.2 Create dataframe of binary Hk2 expression

```
# prepare dataframe
Hk2_cluster<-data.frame(matrix(nrow = sum(!is.na(ODD$rough_ident)), ncol = 4),
                           row.names = colnames(ODD)[!is.na(ODD$rough_ident)])
colnames(Hk2_cluster)=c("Cluster", "Group", "Hk2", "Hif1a")
# fill in metadata
Hk2_cluster$Cluster<-ODD$rough_ident[!is.na(ODD$rough_ident)]
Hk2_cluster$Sample<-ODD$Sample[!is.na(ODD$rough_ident)]
```

```

Hk2_cluster$Group<-ODD$group[!is.na(ODD$rough_ident)]
# fill in expression data and recode
Hk2_cluster$Hk2<-GetAssayData(ODD)[ "Hk2", !is.na(ODD$rough_ident)]
Hk2_cluster$Hk2_yn[Hk2_cluster$Hk2>0]<- "yes"
Hk2_cluster$Hk2_yn[Hk2_cluster$Hk2==0]<- "no"
# create count table
Hk2_cluster_prop<-Hk2_cluster %>%
  group_by(Cluster, Group, Hk2_yn) %>%
  summarise(n=n()) %>%
  mutate(Percent=(n/sum(n))*100)

```

2.4.3 Perform chi-square tests per cluster

```

lapply(unique(Hk2_cluster_prop$Cluster), function(x) {
  paste0(x, " : ",
    chisq.test(data.frame(c(Hk2_cluster_prop$n[Hk2_cluster_prop$Cluster==x][1],
                           Hk2_cluster_prop$n[Hk2_cluster_prop$Cluster==x][2]),
                           c(Hk2_cluster_prop$n[Hk2_cluster_prop$Cluster==x][3],
                           Hk2_cluster_prop$n[Hk2_cluster_prop$Cluster==x][4])),
                           correct = T)$p.value)
}
)

## [[1]]
## [1] "Astrocytes: 0.00453642535269136"
##
## [[2]]
## [1] "Endothelial: 0.114846624509192"
##
## [[3]]
## [1] "Microglia: 0.0291301299698948"
##
## [[4]]
## [1] "Neurons: 5.08634304732291e-20"
##
## [[5]]
## [1] "OPC, Oligo: 1.22973708434569e-20"

```

2.4.4 Perform chi-square tests for total sample

```

# count positive/negative cells per group
Hk2_cluster_prop_total<-Hk2_cluster %>%
  group_by(Group, Hk2_yn) %>%
  summarise(n=n()) %>%
  mutate(Percent=(n/sum(n))*100)
# perform chi-square test
chisq.test(data.frame(c(Hk2_cluster_prop_total$n[1],
                       Hk2_cluster_prop_total$n[2]),
                       c(Hk2_cluster_prop_total$n[3],
                       Hk2_cluster_prop_total$n[4])), correct = F)$p.value

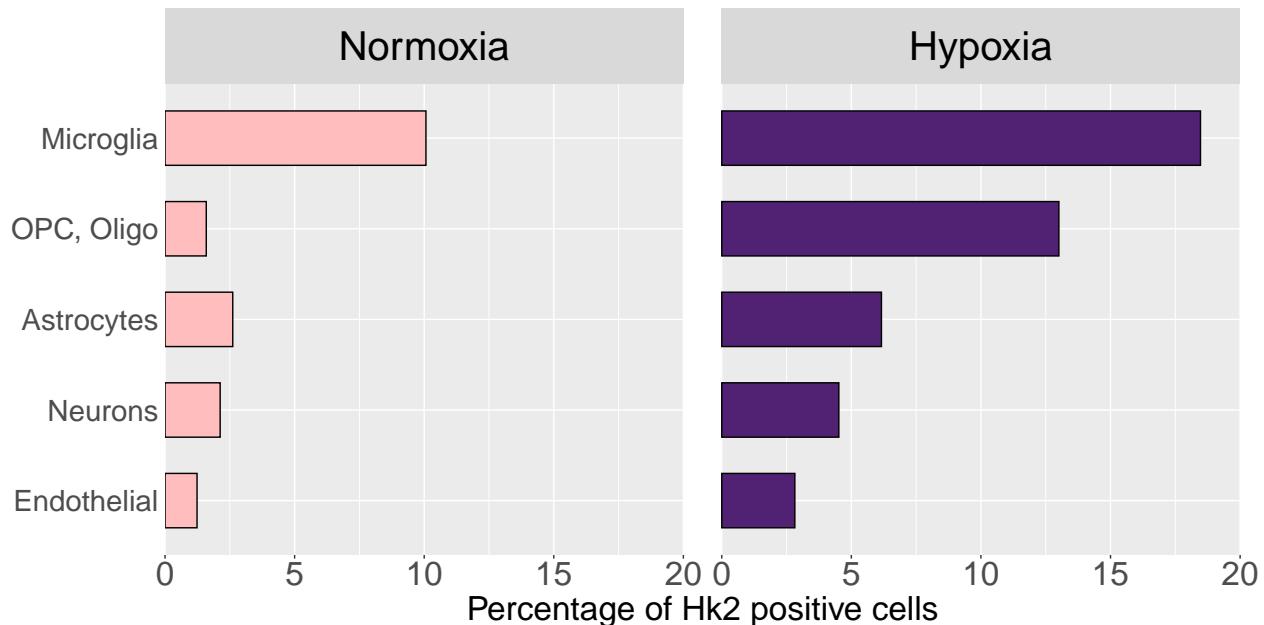
```

```
## [1] 5.790038e-63
```

2.4.5 Plot Hk2+ proportions per cluster

```
# make factor, order levels
Hk2_cluster_prop$Cluster<-factor(Hk2_cluster_prop$Cluster,
                                    levels=c("Endothelial", 'Neurons', 'Astrocytes',
                                            "OPC, Oligo", 'Microglia'))

# plot
ggplot(data = Hk2_cluster_prop[Hk2_cluster_prop$Hk2_yn=="yes",], aes(x = Cluster, y = Percent)) +
  geom_col(aes(fill = Group, width=0.6), col='black' , show.legend = FALSE) +
  coord_flip() + facet_wrap(~Group) +
  theme(axis.ticks.y=element_blank(), axis.text.y=element_text(size=22, angle = 0),
        axis.text.x=element_text(size=25), axis.title.y=element_blank(),
        axis.title.x=element_text(size=25),
        legend.text=element_text(size=16), legend.title=element_text(size=25)) +
  scale_fill_manual(values = c("#FFBDBD", "#512274")) +
  theme(strip.text.x = element_text(size = 30, colour = "black",
                                      margin = margin(0.5,0,0.5,0, "cm")))+
  theme(aspect.ratio = 1/1.1) +
  scale_y_continuous( limits = c(0,20), expand = c(0,0)) +
  theme(panel.spacing = unit(2, "lines")) +
  ylab('Percentage of Hk2 positive cells') +
  theme(plot.margin=unit(c(0,0.5,0,0), "cm"))
```



2.5 Differential expression testing

2.5.1 Vegfa

```
# Test for overall differential expression between Normoxia and Hypoxia
FindMarkers(ODD, ident.1 = 'Hypoxia', ident.2 = 'Normoxia', group.by = 'group', features = 'Vegfa')

##           p_val avg_logFC pct.1 pct.2   p_val_adj
## Vegfa 4.720659e-26 0.2654883 0.178 0.149 8.957922e-22

# Test for differential expression among clusters
Idents(ODD)<-ODD$final_identity
FindAllMarkers(ODD, features = "Vegfa")

##           p_val avg_logFC pct.1 pct.2   p_val_adj   cluster gene
## Vegfa  3.276771e-93 0.4583037 0.363 0.154 6.218001e-89 Astrocytes Vegfa
## Vegfa.1 1.479370e-15 0.3820837 0.354 0.162 2.807253e-11          OPC Vegfa
```

2.5.2 Hk2

```
Idents(ODD)<-ODD$group
# Microglial expression versus all other cells under normoxia
FindMarkers(ODD, ident.1 = 'Microglia', subset.ident = 'Normoxia',
            group.by = 'final_identity', features = 'Hk2', logfc.threshold = 0.01)

##           p_val avg_logFC pct.1 pct.2   p_val_adj
## Hk2 3.170487e-11 0.07440918 0.101 0.021 6.016317e-07

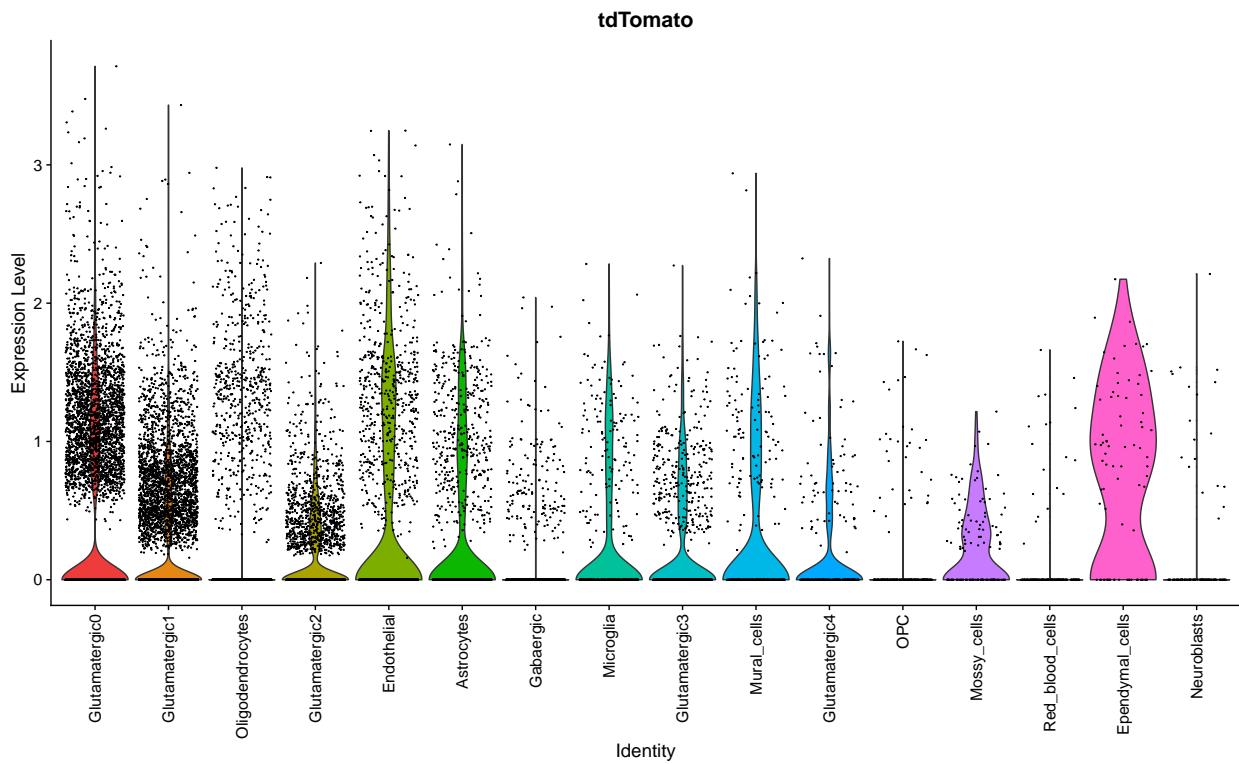
# Microglial expression versus all other cells under hypoxia
FindMarkers(ODD, ident.1 = 'Microglia', subset.ident = 'Hypoxia',
            group.by = 'final_identity', features = 'Hk2', logfc.threshold = 0.01)

##           p_val avg_logFC pct.1 pct.2   p_val_adj
## Hk2 3.610439e-19 0.2859878 0.185 0.061 6.85117e-15
```

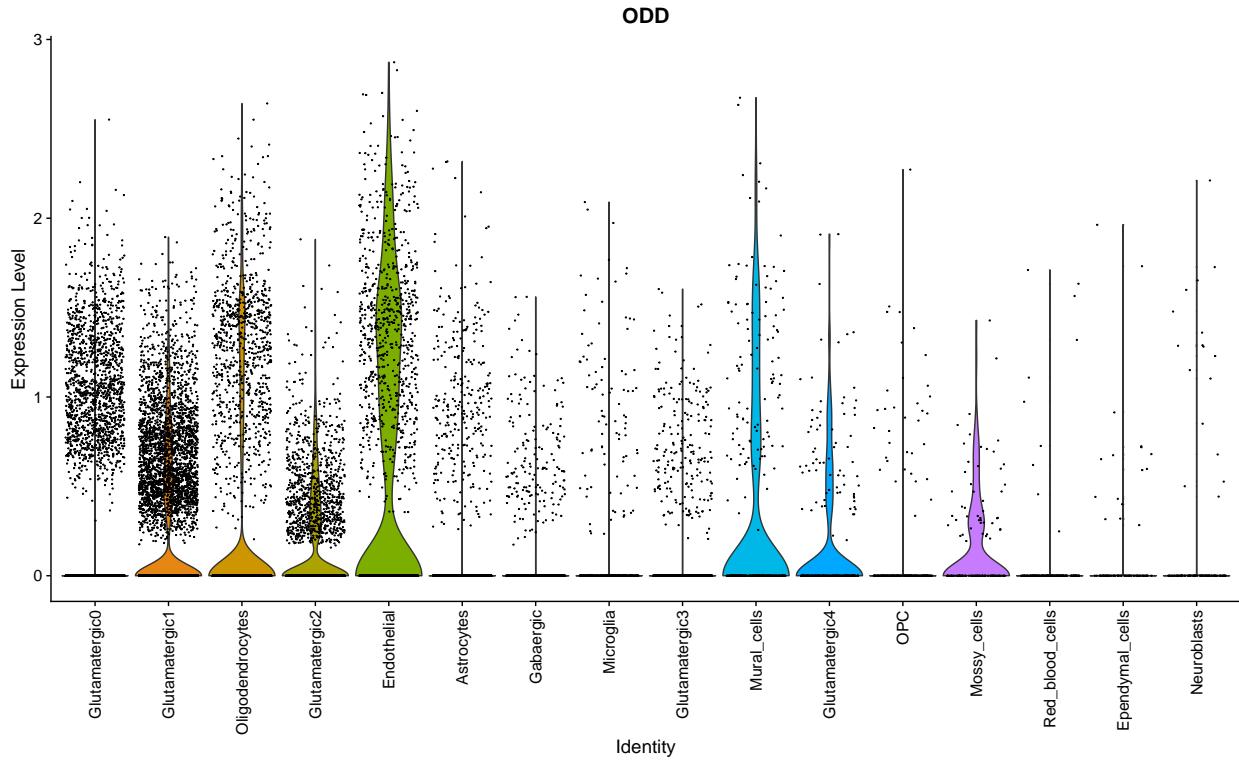
2.6 Supplementary info

2.6.1 Suppl. Figure 3

```
Idents(ODD)<-ODD$final_identity
# tdTomato violin plots
VlnPlot(ODD, features = "tdTomato", cols = my_color_palette, pt.size = 0.1) +
  NoLegend() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  theme(plot.title = element_text(hjust = 0.5))
```



```
# ODD violin plots
VlnPlot(ODD, features = "ODD", cols = my_color_palette, pt.size = 0.1) +
  NoLegend() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  theme(plot.title = element_text(hjust = 0.5))
```



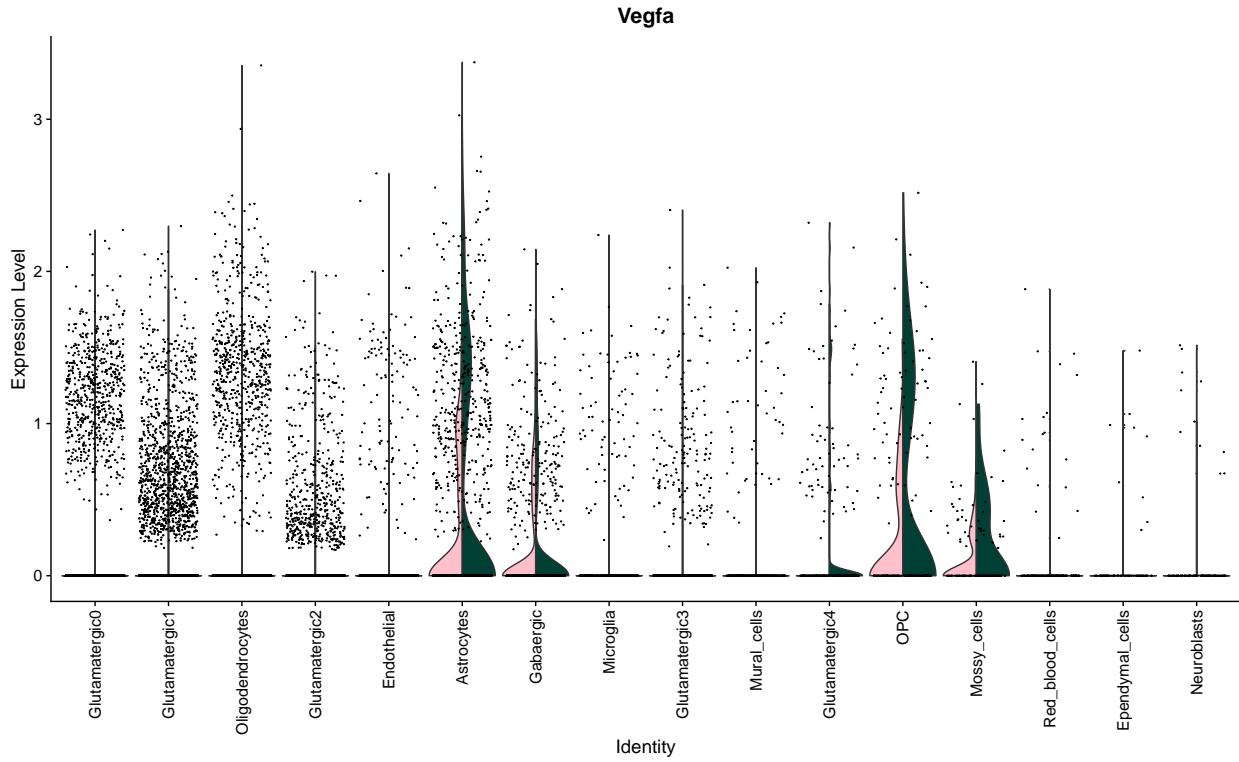
Test for expression differences in tdTomato and ODD between microglia and neurons

```
FindMarkers(ODD, features = c("tdTomato", "ODD"), logfc.threshold = 0.01,
           ident.1 = "Microglia", ident.2 = c("Glutamatergic0", "Glutamatergic1",
                                               "Glutamatergic2", "Glutamatergic3",
                                               "Glutamatergic4", "Gabaergic", "Mossy_cells"))

##          p_val    avg_logFC  pct.1  pct.2 p_val_adj
## tdTomato 0.2346904 -0.02371068 0.313 0.356      1
```

2.6.2 Suppl. Figure 4A

```
# Vegfa violin plots
VlnPlot(ODD, features = "Vegfa", cols = c("pink", "purple"), pt.size = 0.1, split.by = "group")+
  NoLegend()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))+
  theme(plot.title = element_text(hjust = 0.5))
```



2.6.3 Suppl. Figure 4B

```
pvals_Vegfa<-lapply(levels(ODD), function(x) FindMarkers(
  ODD, ident.1 = "Hypoxia", ident.2 = "Normoxia", subset.ident = x,
  group.by = "group", features = "Vegfa", logfc.threshold = 0.01))

names(pvals_Vegfa)<-levels(ODD)
dplyr::bind_rows(pvals_Vegfa, .id = "Cluster")
```

	Cluster	p_val	avg_logFC	pct.1	pct.2	p_val_adj
## Vegfa...1	Glutamatergic0	1.062049e-32	0.22140568	0.118	0.044	2.015344e-28
## Vegfa...2	Glutamatergic1	6.281274e-04	0.16903310	0.173	0.163	1.000000e+00
## Vegfa...3	Oligodendrocytes	4.840344e-20	0.46813190	0.228	0.092	9.185038e-16
## Vegfa...4	Glutamatergic2	1.272789e-01	0.20537892	0.222	0.225	1.000000e+00
## Vegfa...5	Endothelial	3.618140e-03	0.21182701	0.101	0.057	1.000000e+00
## Vegfa...6	Astrocytes	1.682084e-05	0.47005730	0.374	0.350	3.191923e-01
## Vegfa...7	Gabaergic	8.914883e-01	0.15443198	0.258	0.293	1.000000e+00
## Vegfa...8	Microglia	6.402041e-01	0.09817043	0.153	0.188	1.000000e+00
## Vegfa...9	Glutamatergic3	3.295643e-03	0.31391639	0.225	0.171	1.000000e+00
## Vegfa...10	Mural_cells	1.996432e-01	0.04942680	0.129	0.203	1.000000e+00
## Vegfa...11	Glutamatergic4	4.939329e-01	0.31782062	0.250	0.246	1.000000e+00
## Vegfa...12	OPC	6.774208e-03	0.53190615	0.389	0.293	1.000000e+00
## Vegfa...13	Mossy_cells	4.254754e-02	0.14138619	0.417	0.293	1.000000e+00
## Vegfa...14	Red_blood_cells	6.893902e-01	-0.05511010	0.108	0.129	1.000000e+00
## Vegfa...15	Ependymal_cells	1.917728e-01	0.25509083	0.174	0.086	1.000000e+00
## Vegfa...16	Neuroblasts	4.486206e-01	0.14135489	0.116	0.075	1.000000e+00

2.7 Session info

```
sessionInfo()

## R version 3.6.0 (2019-04-26)
## 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] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] scales_1.1.1       ggplot2_3.3.2       dplyr_1.0.2       cowplot_1.1.0
## [5] Seurat_3.0.0       RColorBrewer_1.1-2  reshape2_1.4.4
##
## loaded via a namespace (and not attached):
##  [1] tsne_0.1-3        nlme_3.1-150       matrixStats_0.57.0
##  [4] httr_1.4.2         numDeriv_2016.8-1.1 sctransform_0.3.1
##  [7] tools_3.6.0        R6_2.5.0          irlba_2.3.3
## [10] KernSmooth_2.23-18 lazyeval_0.2.2     BiocGenerics_0.32.0
## [13] colorspace_2.0-0   sn_1.6-2          withr_2.3.0
## [16] gridExtra_2.3      tidyselect_1.1.0   mnormt_2.0.2
## [19] compiler_3.6.0     Biobase_2.46.0    xml2_1.3.2
## [22] TFisher_0.2.0     plotly_4.9.2.1   sandwich_3.0-0
## [25] labeling_0.4.2    lmtest_0.9-38    mvtnorm_1.1-1
## [28] ggridges_0.5.2    pbapply_1.4-3    stringr_1.4.0
## [31] digest_0.6.27     rmarkdown_2.5    R.utils_2.10.1
## [34] pkgconfig_2.0.3    htmltools_0.5.0   parallelly_1.21.0
## [37] plotrix_3.7-8     htmlwidgets_1.5.2 rlang_0.4.8
## [40] farver_2.0.3      generics_0.1.0   jsonlite_1.7.1
## [43] zoo_1.8-8         ica_1.0-2        R.oo_1.24.0
## [46] magrittr_1.5       Matrix_1.2-18    Rcpp_1.0.5
## [49] munsell_0.5.0     reticulate_1.18  ape_5.4-1
## [52] lifecycle_0.2.0   R.methodsS3_1.8.1 stringi_1.5.3
## [55] multcomp_1.4-15  yaml_2.2.1       mathjaxr_1.0-1
## [58] gbRd_0.4-11     MASS_7.3-53     Rtsne_0.15
## [61] plyr_1.8.6        parallel_3.6.0   listenv_0.8.0
## [64] ggrepel_0.8.2    crayon_1.3.4    lattice_0.20-41
## [67] splines_3.6.0    multtest_2.42.0  SDMTools_1.1-221
## [70] tmvnsim_1.0-2    knitr_1.30     pillar_1.4.7
## [73] igraph_1.2.6     future.apply_1.6.0 codetools_0.2-18
## [76] stats4_3.6.0     mutoss_0.1-12   glue_1.4.2
## [79] packrat_0.5.0    evaluate_0.14  metap_1.4
## [82] data.table_1.13.2 png_0.1-7       vctrs_0.3.4
## [85] Rdpack_2.1       tidyrr_1.1.2   gtable_0.3.0
```

```
## [88] RANN_2.6.1           purrrr_0.3.4        future_1.20.1
## [91] xfun_0.19              rsvd_1.0.3          rbibutils_1.4
## [94] viridisLite_0.3.0      survival_3.2-7     tibble_3.0.4
## [97] cluster_2.1.0          globals_0.13.1     fitdistrplus_1.1-1
## [100] TH.data_1.0-10         ellipsis_0.3.1    ROOCR_1.0-11
```