# Fernandez Garcia-Agudo et al: scRNAseq analysis

Agnes A. Steixner-Kumar

10 11 2020

## Contents

# 1 Libraries and setup

Uncomment if notebook location is different from project location to set working directory for the entire notebook

```
#wd<-"D:/your_path_to_folder"
#knitr::opts_knit$set(root.dir = normalizePath(wd))
```

Loading libraries required for analyses

```
#load required Seurat version,
#if you have only the version needed here installed,
#you can skip the lib.loc specification
library(Seurat, lib.loc = "../R/R-4.0.0/library/different_versions/")
library(ggplot2)
library(monocle)
```

# 2 Loading data

```
#load whole hippocampus data (hypoxia vs. normoxia) from Butt et al (2020): GSEXXX
HYP<-readRDS("Butt_et_al_hypoxia_hippocampus.RDS")
#load CA1 data (EPO vs. placebo) from Wakhloo et al (2020): GSE144444
EPO<-readRDS("Wakhloo_et_al_EPO_CA1.RDS")
#update object to latest Seurat version
EPO<-UpdateSeuratObject(EPO)
#load CA1 glutamatergic clusters from Wakhloo et al (2020)
#analyzed in pseudotime analysis (Monocle2)
#(https://github.com/AgnesSteixner/EPO_Wakhloo_et_al_NATCOMM/tree/master/trajectory)
mon<-readRDS("Wakhloo_et_al_monocle_pseudotime.RDS")
```

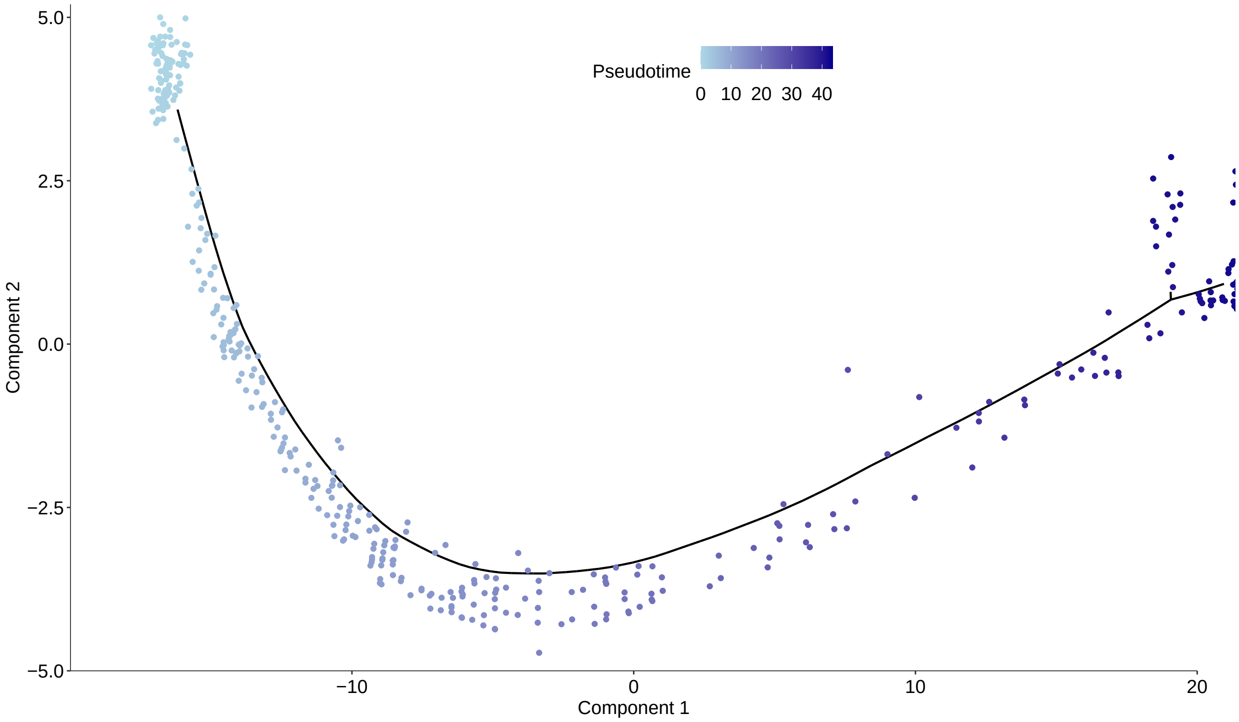
Create dataframe of hypoxia object for plotting

```
d_hyp<-data.frame(t(data.frame(GetAssayData(HYP, slot = "data", assay = "RNA"))))
d_hyp$group<-HYP$group
d_hyp$final_identity<-HYP$final_identity
```

# 3 Figure 2

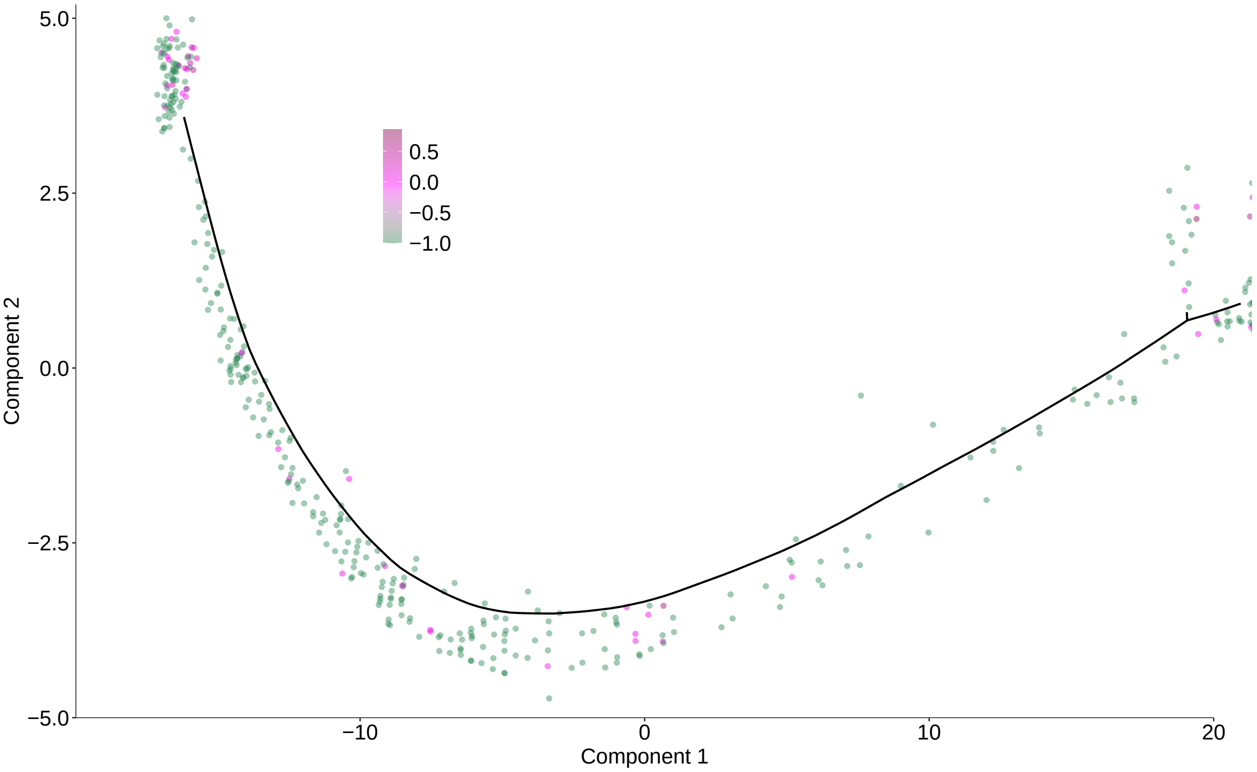
## 3.1 Figure 2G

```
plot_cell_trajectory(mon, color_by = "Pseudotime", show_branch_points = F)+
  theme(legend.title = element_text(size = 14))+
  theme(legend.text = element_text(size = 14, vjust = -1))+
  theme(axis.text.y = element_text(size = 14, colour = "black"))+
  theme(axis.text.x = element_text(size = 14, colour = "black",vjust = 0.5))+
  theme(axis.title.x = element_text(size = 14, colour = "black",vjust = 0.5))+
  theme(axis.title.y = element_text(size = 14, colour = "black"))+
  theme(legend.position = c(0.57,0.9))+
  coord_cartesian(xlim = c(-19.99,20), ylim= c(-5,5.2), clip = "off", expand = F)+
  theme(plot.margin = unit(c(1,1,0.5,0.5), "cm"))+
  scale_color_gradient(low = "lightblue",
  high = "darkblue")+
  theme(legend.key.size = unit(0.7, "cm"), legend.key.height = unit(0.6, "cm"),
        legend.direction = "horizontal")
```



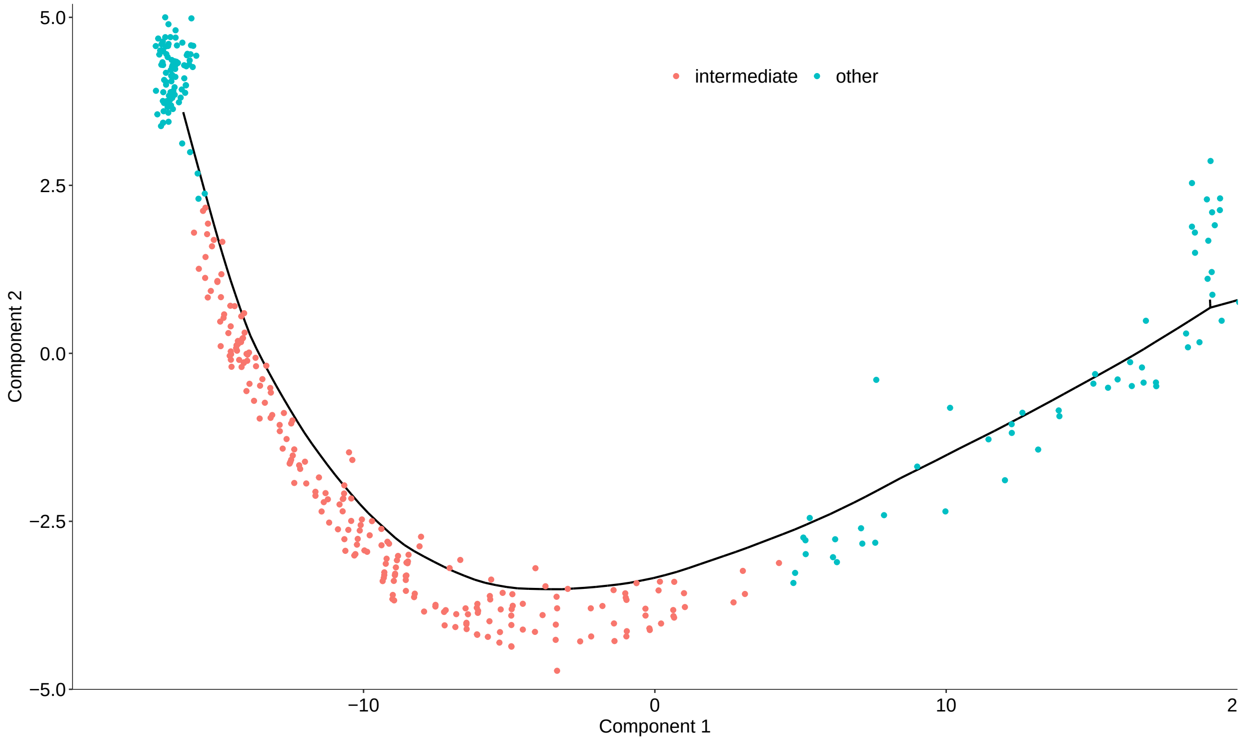
## 3.2 Figure 2H

```
plot_cell_trajectory(mon, markers = "Tbr1", use_color_gradient = T, cell_size = 1.5,
                     show_branch_points = F)+
  theme(axis.text = element_text(size = 16, colour = "black"))+
  theme(axis.title = element_text(size = 16, colour = "black"))+
  theme(legend.position = c(0.3, 0.75))+
  theme(legend.text = element_text(size = 16))+
  theme(legend.title = element_blank())+
  theme(legend.key.size = unit(0.5, "cm"), legend.key.height = unit(0.6, "cm"))+
  scale_color_gradientn(colors=alpha(c("seagreen", "magenta", "deeppink4"),0.45))+
  coord_cartesian(xlim = c(-19.99,20), ylim= c(-5,5.2), clip = "off", expand = F)+
  theme(plot.margin = unit(c(-0.48,1,0,0), "cm"))
```



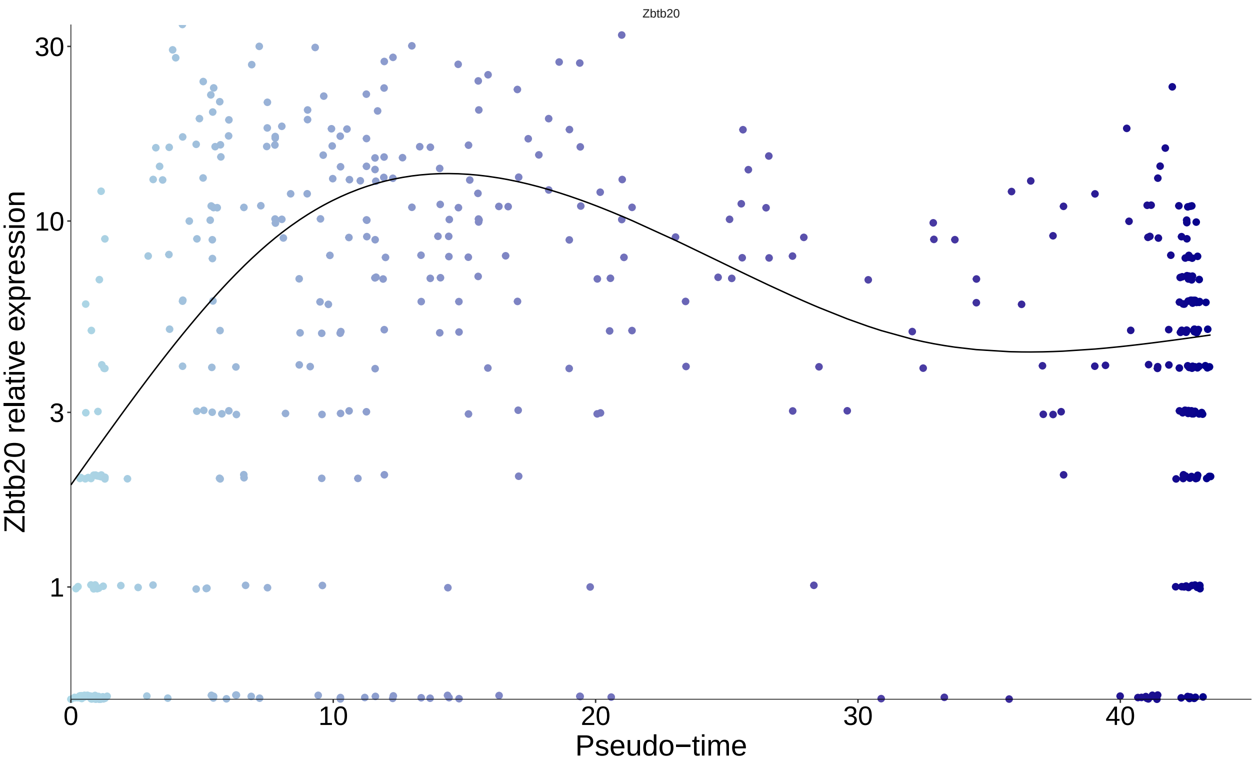
### 3.3 Figure 2I (left panel)

```
#define intermediate neuronal stage
mon$Pseudotime_interm<-"other"
mon$Pseudotime_interm[mon$Pseudotime>3&mon$Pseudotime<25]<-"intermediate"

#plot trajectory
plot_cell_trajectory(mon, color_by = "Pseudotime_interm", show_branch_points = F)+
  theme(legend.title = element_blank())+
  theme(legend.text = element_text(size = 14))+
  theme(axis.text.y = element_text(size = 14, colour = "black"))+
  theme(axis.text.x = element_text(size = 14, colour = "black",vjust = 0.5))+
  theme(axis.title.x = element_text(size = 14, colour = "black",vjust = 0.5))+
  theme(axis.title.y = element_text(size = 14, colour = "black"))+
  theme(legend.position = c(0.6,0.9))+
  guides(color=guide_legend(nrow = 1))+
  coord_cartesian(xlim = c(-19.99,20), ylim= c(-5,5.2), clip = "off", expand = F)+
  theme(plot.margin = unit(c(0.5,0,0.5,0.5), "cm"))
```



## 3.4 Figure 2I (right panel)

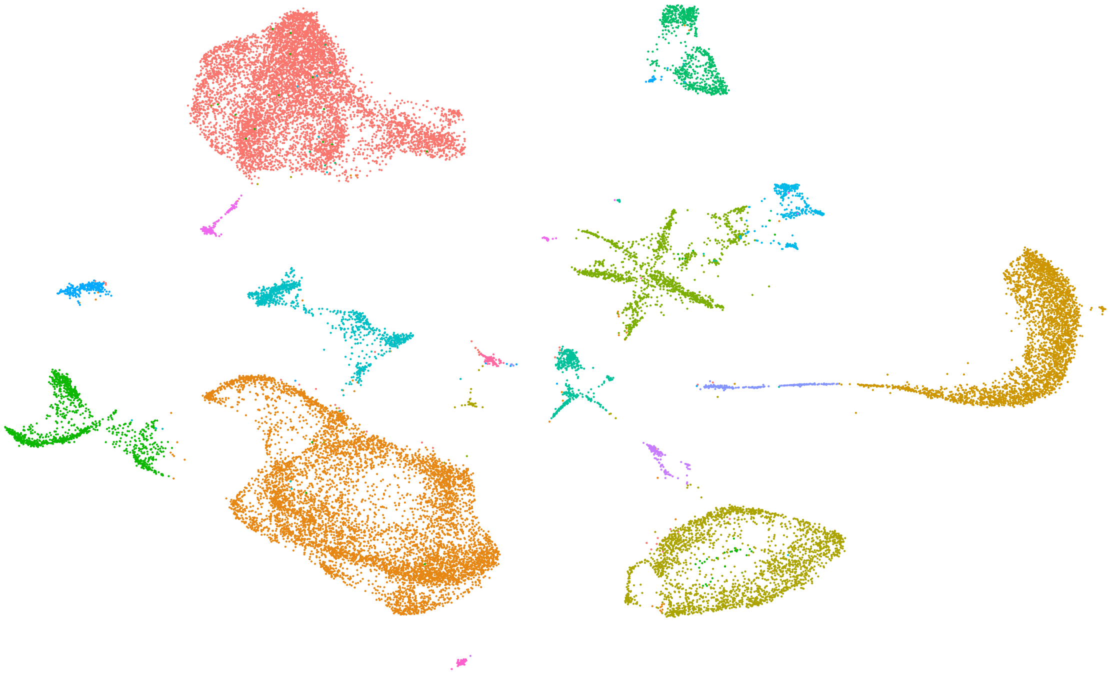
```
# subset monocle object
mon_zbtb<-mon["Zbtb20",]

# create pseudotime expression graph for Zbtb20
plot_genes_in_pseudotime(mon_zbtb, color_by = "Pseudotime", cell_size = 2)+
  theme(axis.text = element_text(size = 20, colour = "black"))+
  theme(axis.title = element_text(size = 22, colour = "black"))+
  theme(axis.line = element_line(colour = "black"))+
  theme(legend.position = "none")+
  scale_color_gradient(low = "lightblue",
  high = "darkblue")+
  coord_cartesian(xlim = c(0,45), clip = "off", expand = F)+
  theme(plot.margin = unit(c(0.2,0,0,0), "cm"))+
  ylab("Zbtb20 relative expression")
```



# 4 Figure 4

## 4.1 Figure 4B

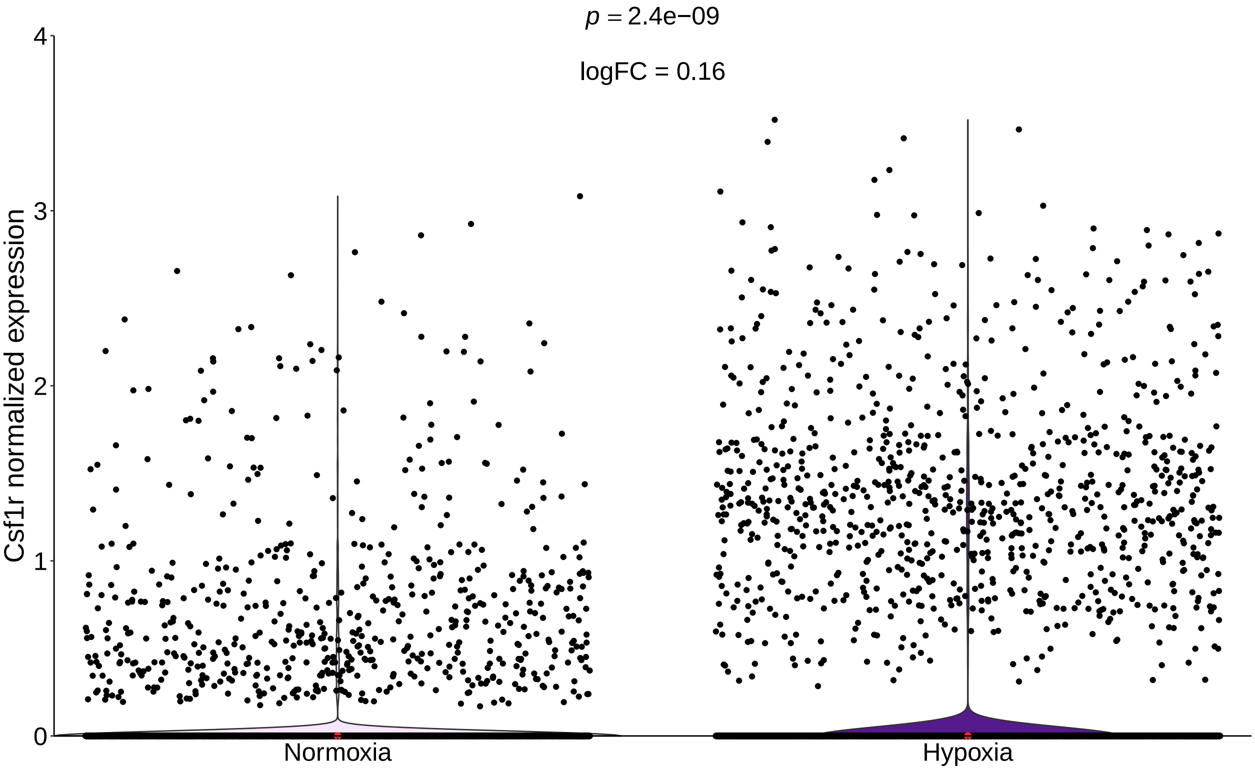
```
#shift ependymal cells upwards on dimension plot for easier visualization
#get UMAP embeddings
emb<-Embeddings(HYP, reduction = "umap")
#shift selected cells upwards on UMAP2
emb[,2][emb[,2]<(-15)]<-emb[,2][emb[,2]<(-15)]+12
#feed changed embeddings back into object
HYP@reductions$umap@cell.embeddings<-emb

# create dimension plot
DimPlot(HYP, reduction = "umap")+
theme(legend.position = "none")+
theme(axis.text = element_blank())+
theme(axis.title = element_blank())+
theme(axis.line = element_blank())+
theme(axis.ticks = element_blank())
```



## 4.2 Figure 4C - Csf1r

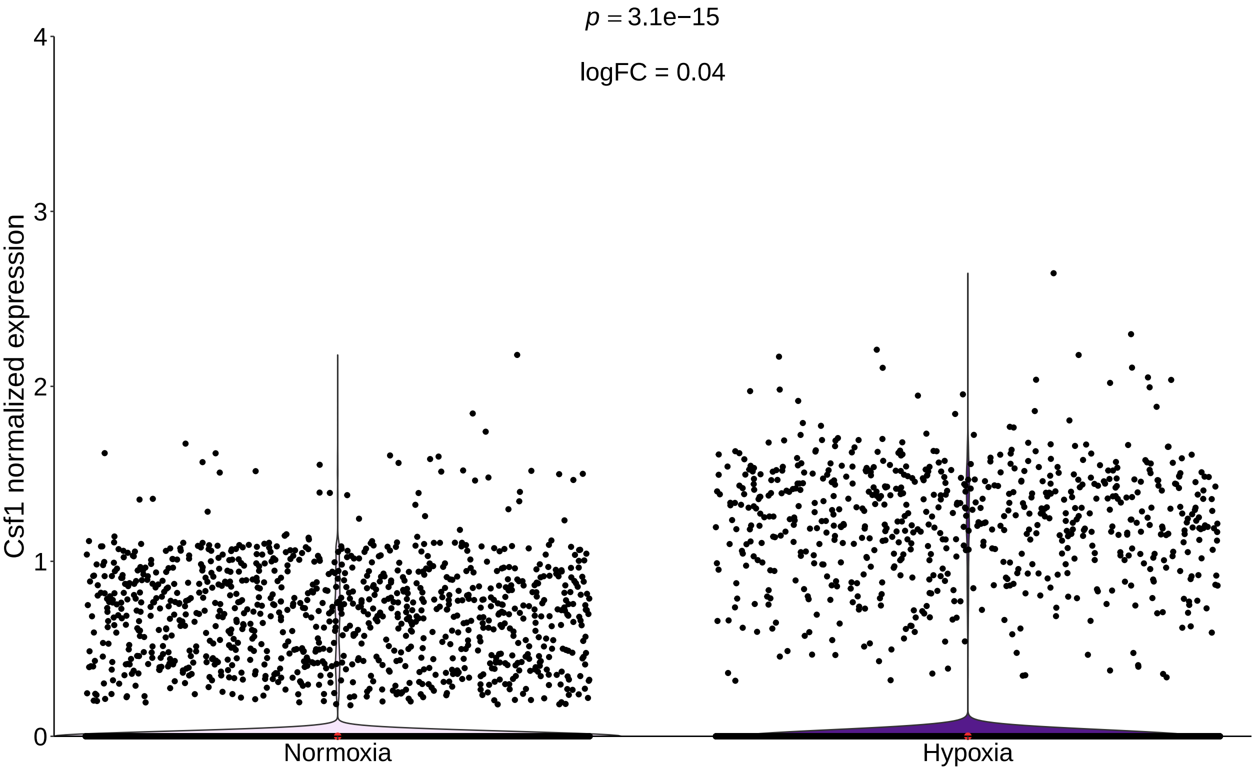
Note: Wilcox test was performed with limma as implemented in Seurat. This implementation corrects for correlation within groups (when correlation is positive, more conservative statistics) and continuity. Testing is done for both directions -> smaller p is multiplied by 2; Bonferroni adjusted p = p*number of genes in dataset.

```
# calculate p value
p<-formatC(min(limma::rankSumTestWithCorrelation(statistics = HYP@assays$RNA@data["Csf1r",],
                                                 index = which(HYP$group=="Hypoxia")))*
                                                          2*nrow(HYP),2)
# calculate log-fold change
logfc<-round(log(mean(expm1(HYP@assays$RNA@data["Csf1r",HYP$group=="Hypoxia"]))+1)-
               log(mean(expm1(HYP@assays$RNA@data["Csf1r",HYP$group=="Normoxia"]))+1),2)

# create violin plot
ggplot(d_hyp, aes(x=group, y=Csf1r, fill=group))+
  geom_violin()+ geom_jitter()+
  theme(panel.background = element_blank())+
  coord_cartesian(expand = F, ylim = c(0,4), clip = "off")+
  scale_fill_manual(values = c("#F7E7FC","#551A8B"))+
  theme(axis.title.x = element_blank())+ylab('Csf1r normalized expression')+
  theme(axis.text.x = element_text(size = 19, color = 'black'),
        axis.text.y = element_text(size = 19, color = 'black'),
        axis.title = element_text(size = 20.5),
        axis.line = element_line(colour = 'black'))+
  theme(legend.position = 'none')+
  annotate("text", x=1.5, y=4.1, label=paste0("italic('p')","==", p), size=6.7, parse=T)+
  annotate("text", x=1.5, y=3.8, label=paste0("logFC = ", logfc), size=6.7)+
  theme(plot.margin = unit(c(0.9,0,0.1,0),"cm"))+
  stat_summary(fun=median, geom="point", size=2, color="firebrick2")
```



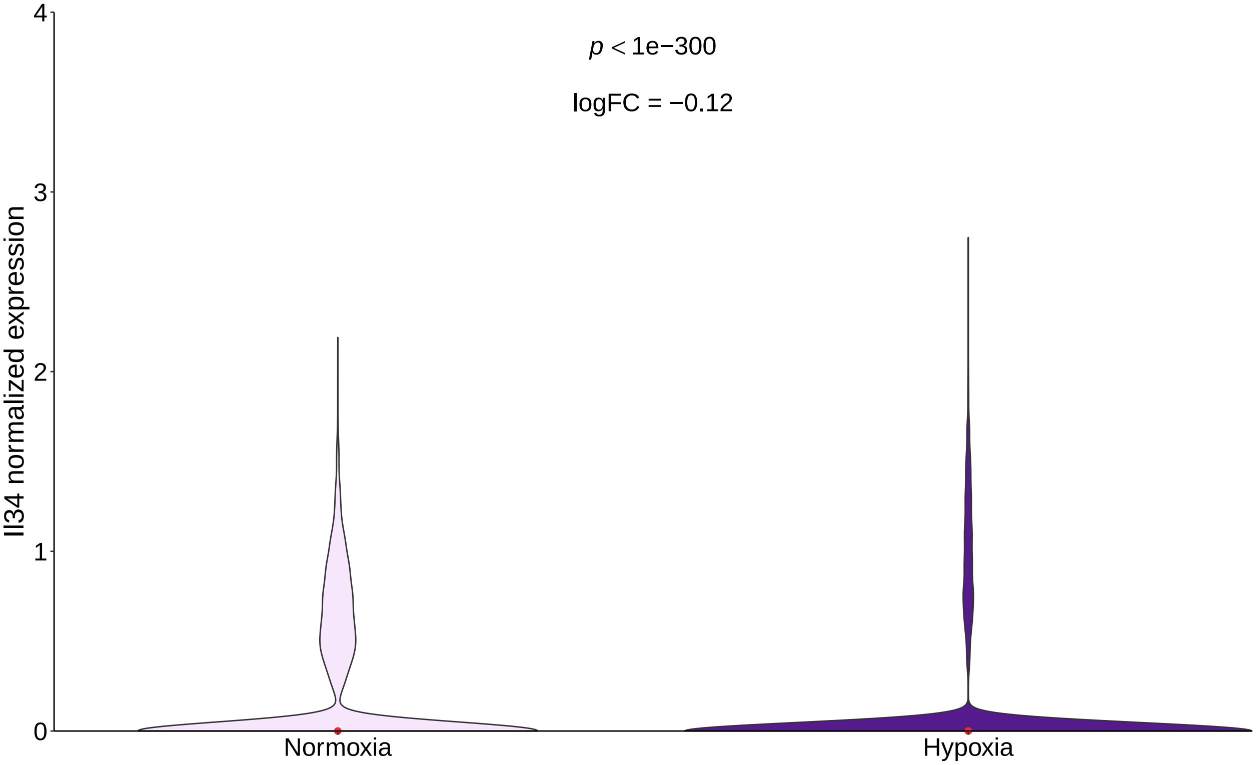
### 4.3 Figure 4C - Csf1

```
# calculate p value
p<-formatC(min(limma::rankSumTestWithCorrelation(statistics = HYP@assays$RNA@data["Csf1",],
                                                 index = which(HYP$group=="Hypoxia")))*2*nrow(HYP),2)
# calculate log-fold change
logfc<-round(log(mean(expm1(HYP@assays$RNA@data["Csf1",HYP$group=="Hypoxia"]))+1)-
               log(mean(expm1(HYP@assays$RNA@data["Csf1",HYP$group=="Normoxia"]))+1),2)

# create violin plot
ggplot(d_hyp, aes(x=group, y=Csf1, fill=group))+
  geom_violin()+ geom_jitter()+
  theme(panel.background = element_blank())+
  coord_cartesian(expand = F, ylim = c(0,4), clip = "off")+
  scale_fill_manual(values = c("#F7E7FC","#551A8B"))+
  theme(axis.title.x = element_blank())+ylab('Csf1 normalized expression')+
  theme(axis.text.x = element_text(size = 19, color = 'black'),
        axis.text.y = element_text(size = 19, color = 'black'),
        axis.title = element_text(size = 20.5),
        axis.line = element_line(colour = 'black'))+
  theme(legend.position = 'none')+
  annotate("text", x=1.5, y=4.1, label=paste0("italic('p')","==", p), size=6.7, parse=T)+
  annotate("text", x=1.5, y=3.8, label=paste0("logFC = ", logfc), size=6.7)+
  theme(plot.margin = unit(c(0.9,0,0.1,0),"cm"))+
  stat_summary(fun=median, geom="point", size=2, color="firebrick2")
```



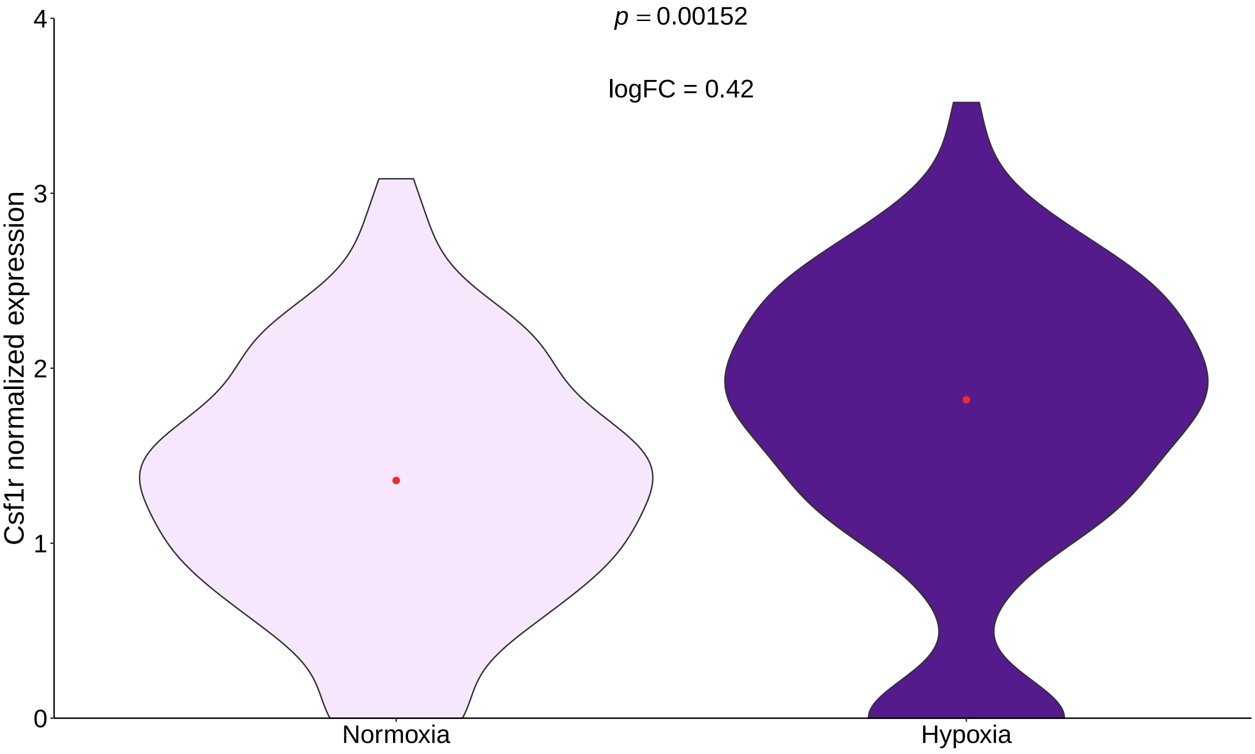
## 4.4 Figure 4C - Il34

```
# calculate p-value
# p-value is "0" due to R limitation -> show it as approximation
#p<-formatC(min(limma::rankSumTestWithCorrelation(statistics = HYP@assays$RNA@data["Il34",],
#index = which(HYP$group=="Hypoxia"))*2*nrow(HYP),3)
p<-1e-300
# calculate log-fold change
logfc<-round(log(mean(expm1(HYP@assays$RNA@data["Il34",HYP$group=="Hypoxia"]))+1)-
               log(mean(expm1(HYP@assays$RNA@data["Il34",HYP$group=="Normoxia"]))+1),2)

# create violin plot
ggplot(d_hyp, aes(x=group, y=Il34, fill=group))+
  geom_violin()+
  theme(panel.background = element_blank())+
  coord_cartesian(expand = F, ylim = c(0,4), clip = "off")+
  scale_fill_manual(values = c("#F7E7FC","#551A8B"))+
  theme(axis.title.x = element_blank())+ylab('Il34 normalized expression')+
  theme(axis.text.x = element_text(size = 19, color = 'black'),
        axis.text.y = element_text(size = 19, color = 'black'),
        axis.title = element_text(size = 20.5),
        axis.line = element_line(colour = 'black'))+
  theme(legend.position = 'none')+
  annotate("text", x=1.5, y=3.8, label=paste0("italic('p')","<", p), size=6.7, parse=T)+
  annotate("text", x=1.5, y=3.5, label=paste0("logFC = ", logfc), size=6.7)+
  theme(plot.margin = unit(c(0.5,0,0,0),"cm"))+
  stat_summary(fun=median, geom="point", size=2, color="firebrick2")
```



## 4.5 Figure 4D

```
#subset microglia
M<-subset(HYP, final_identity=="Microglia")
d_m<-data.frame(t(data.frame(GetAssayData(M, slot = "data", assay = "RNA"))))
d_m$group<-M$group
# calculate p-value
p<-formatC(min(limma::rankSumTestWithCorrelation(statistics = M@assays$RNA@data["Csf1r",],
                                                 index = which(M$group=="Hypoxia")))*2*nrow(M),3)
# calculate log-fold change
logfc<-round(log(mean(expm1(M@assays$RNA@data["Csf1r",M$group=="Hypoxia"]))+1)-
               log(mean(expm1(M@assays$RNA@data["Csf1r",M$group=="Normoxia"]))+1),2)

# create violin plot
ggplot(d_m, aes(x=group, y=Csf1r, fill=group))+
  geom_violin()+
  theme(panel.background = element_blank())+
  coord_cartesian(expand = F, xlim=c(0.4, 2.5), ylim = c(0,4), clip = "off")+
  scale_fill_manual(values = c("#F7E7FC","#551A8B"))+
  theme(axis.title.x = element_blank())+ylab('Csf1r normalized expression')+
  theme(axis.text.x = element_text(size = 19.3, color = 'black'),
        axis.text.y = element_text(size = 19, color = 'black'),
        axis.title = element_text(size = 20.5),
        axis.line = element_line(colour = 'black'))+
  theme(legend.position = 'none')+
  annotate("text", x=1.5, y=4, label=paste0("italic('p')","==", p), size=6.7, parse=T)+
  annotate("text", x=1.5, y=3.6, label=paste0("logFC = ", logfc), size=6.7)+
  theme(plot.margin = unit(c(0.9,0,0.1,0),"cm"))+
  stat_summary(fun=median, geom="point", size=2, color="firebrick2")
```

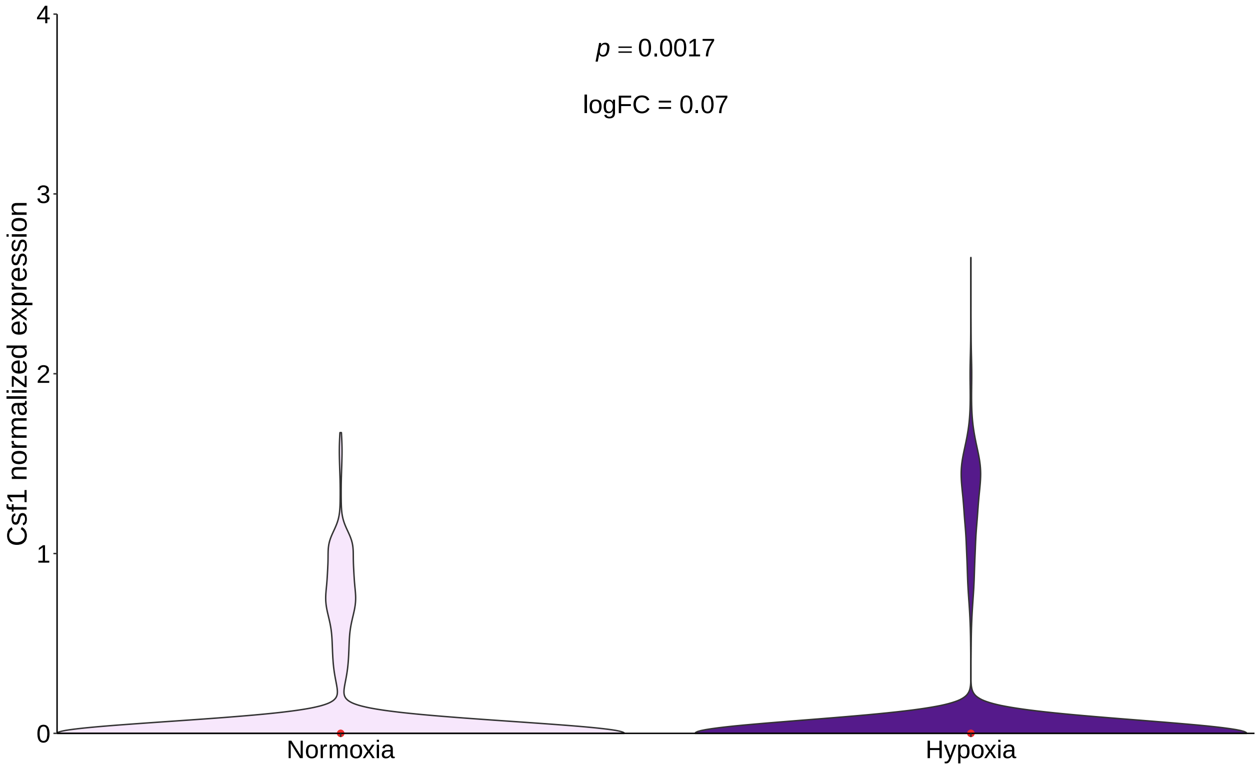


## 4.6 Figure 4E

```
# define glial identity
d_hyp$glia<-"NO"
d_hyp$glia[d_hyp$final_identity=="OPC"|d_hyp$final_identity=="Oligodendrocytes"|
             d_hyp$final_identity=="Astrocytes"|d_hyp$final_identity=="Microglia"|
             d_hyp$final_identity=="Ependymal_cells"]<-"YES"
# show table of glial numbers
table(d_hyp$glia)
```

```
##
##    NO   YES
## 20622  5228
```

```r
# set identity to glia
Idents(HYP)<-d_hyp$glia
# calculate p-value and log-fold change
csf1_glia_stats<-FindMarkers(HYP, features = "Csf1", group.by = "group",
                             ident.1 = "Hypoxia", logfc.threshold = 0.01,
                             subset.ident = "YES")
#round values
p<-round(csf1_glia_stats$p_val_adj,4)
logfc<-round(csf1_glia_stats$avg_logFC, 2)

# create violin plot
ggplot(d_hyp[d_hyp$glia=="YES",], aes(x=group, y=Csf1, fill=group))+
  geom_violin()+
  theme(panel.background = element_blank())+
  coord_cartesian(expand = F, ylim = c(0,4), clip = "off")+
  scale_fill_manual(values = c("#F7E7FC","#551A8B"))+
  theme(axis.title.x = element_blank())+ylab('Csf1 normalized expression')+
  theme(axis.text.x = element_text(size = 19, color = 'black'),
        axis.text.y = element_text(size = 19, color = 'black'),
        axis.title = element_text(size = 20.5),
        axis.line = element_line(colour = 'black'))+
  theme(legend.position = 'none')+
  annotate("text", x=1.5, y=3.8, label=paste0("italic('p')","==", p), size=6.7, parse=T)+
  annotate("text", x=1.5, y=3.5, label=paste0("logFC = ", logfc), size=6.7)+
  theme(plot.margin = unit(c(0.5,0,0,0),"cm"))+
  stat_summary(fun=median, geom="point", size=2, color="firebrick2")
```

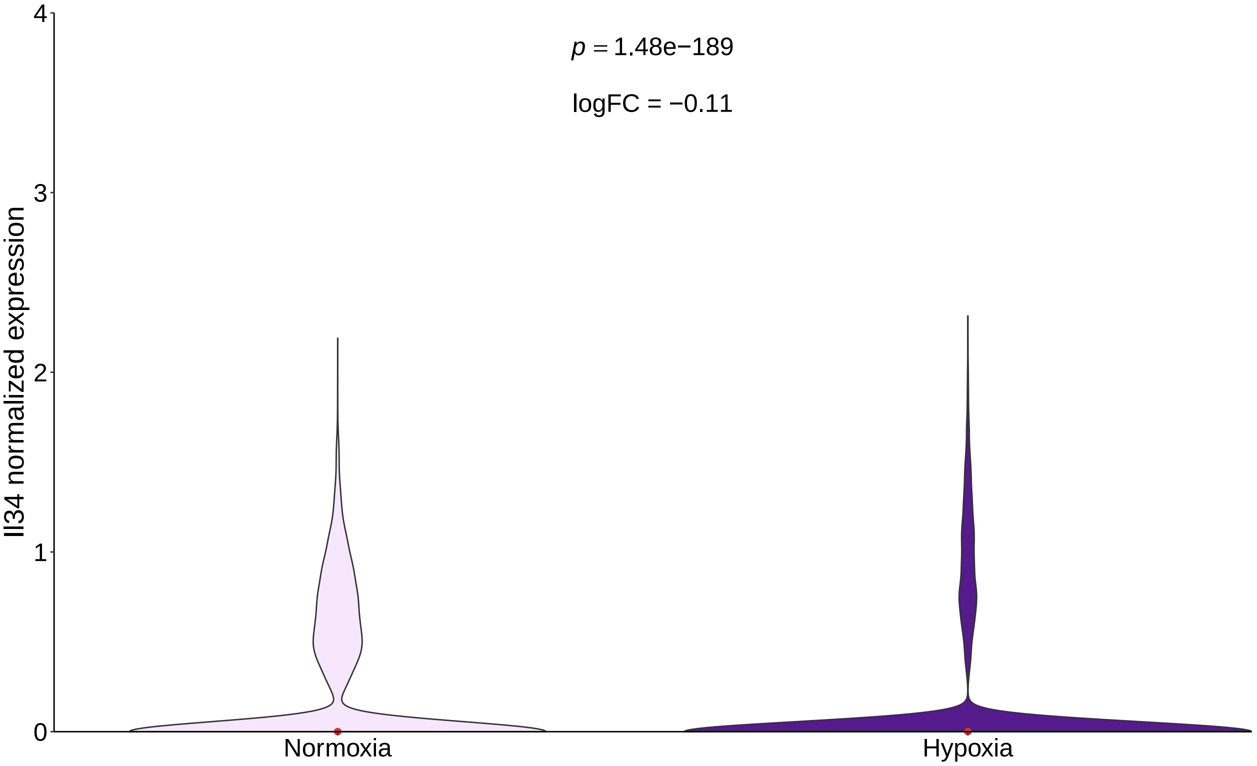


### 4.7 Figure 4F

```
# create identity of excitatory neurons
HYP$excitatory<-"NO"
HYP$excitatory[HYP$final_identity %in% c("Glutamatergic0_DG/CA2/CA3","Glutamatergic1_CA1",
                                          "Glutamatergic2","Glutamatergic3","Glutamatergic4",
                                          "Mossy_cells")]<-"YES"
# show table of excitatory neuron number
table(HYP$excitatory)
```

```
##
##    NO   YES
##  8039 17811
```

```
# fill info into dataframe
d_hyp$excitatory<-HYP$excitatory

# create violin plot for Il34 expression in glutamatergic neurons
# set identity to excitatory neuron
Idents(HYP)<-HYP$excitatory
# calculate stats for differential expression between hypoxia and normoxia
il34_glut_stats<-FindMarkers(HYP, features = "Il34", group.by = "group",
                             ident.1 = "Hypoxia", logfc.threshold = 0.01,
                             subset.ident = "YES")
p<-formatC(il34_glut_stats$p_val_adj,3)
logfc<-round(il34_glut_stats$avg_logFC, 2)

#create violin plot
ggplot(d_hyp[d_hyp$excitatory=="YES",], aes(x=group, y=Il34, fill=group))+
  geom_violin()+
  theme(panel.background = element_blank())+
  coord_cartesian(expand = F, ylim = c(0,4), clip = "off")+
  scale_fill_manual(values = c("#F7E7FC","#551A8B"))+
  theme(axis.title.x = element_blank())+
  ylab('Il34 normalized expression')+
  theme(axis.text.x = element_text(size = 19, color = 'black'),
        axis.text.y = element_text(size = 19, color = 'black'),
        axis.title =element_text(size = 20.5),
        axis.line = element_line(colour = 'black'))+
  theme(legend.position = 'none')+
  annotate("text", x=1.5, y=3.8, label=paste0("italic('p')","==", p), size=6.7, parse=T)+
  annotate("text", x=1.5, y=3.5, label=paste0("logFC = ", logfc), size=6.7)+
  theme(plot.margin = unit(c(0.5,0,0,0),"cm"))+
  stat_summary(fun=median, geom="point", size=2, color="firebrick2")
```



# 5 Session info

```
sessionInfo()
```

```
## R version 4.0.0 (2020-04-24)
## 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] splines   stats4    parallel  stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
## [1] monocle_2.16.0      DDRTree_0.1.5       irlba_2.3.3
## [4] VGAM_1.1-3          Biobase_2.48.0      BiocGenerics_0.34.0
## [7] Matrix_1.2-18       ggplot2_3.3.0       Seurat_3.1.5
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-147         tsne_0.1-3           matrixStats_0.56.0
##  [4] RcppAnnoy_0.0.16     RColorBrewer_1.1-2   httr_1.4.2
##  [7] docopt_0.6.1         sctransform_0.2.1    tools_4.0.0
## [10] R6_2.4.1             KernSmooth_2.23-16   uwot_0.1.8
## [13] lazyeval_0.2.2       colorspace_1.4-1     withr_2.3.0
## [16] tidyselect_1.1.0     gridExtra_2.3        compiler_4.0.0
## [19] formatR_1.7          plotly_4.9.2.1       labeling_0.3
## [22] slam_0.1-47          scales_1.1.1         lmtest_0.9-37
## [25] ggridges_0.5.2       pbapply_1.4-3        rappdirs_0.3.1
## [28] stringr_1.4.0        digest_0.6.26        sparsesvd_0.2
## [31] rmarkdown_2.5.3      pkgconfig_2.0.3      htmltools_0.5.0
## [34] limma_3.44.1         htmlwidgets_1.5.2    rlang_0.4.6
## [37] FNN_1.1.3            farver_2.0.3         combinat_0.0-8
## [40] zoo_1.8-8            jsonlite_1.6.1       ica_1.0-2
## [43] dplyr_0.8.5          magrittr_1.5         patchwork_1.0.0
## [46] Rcpp_1.0.4.6         munsell_0.5.0        viridis_0.5.1
## [49] ape_5.4              reticulate_1.16      lifecycle_0.2.0
## [52] stringi_1.4.6        yaml_2.2.1           MASS_7.3-51.5
## [55] Rtsne_0.15           plyr_1.8.6           grid_4.0.0
## [58] listenv_0.8.0        ggrepel_0.8.2        crayon_1.3.4
## [61] lattice_0.20-41      cowplot_1.1.0        knitr_1.30
## [64] pillar_1.4.6         igraph_1.2.5         future.apply_1.5.0
## [67] reshape2_1.4.4       codetools_0.2-16     leiden_0.3.3
## [70] glue_1.4.1           evaluate_0.14        data.table_1.12.8
## [73] png_0.1-7            vctrs_0.3.0          gtable_0.3.0
## [76] RANN_2.6.1           purrr_0.3.4          tidyr_1.0.3
## [79] future_1.17.0        assertthat_0.2.1     xfun_0.19
## [82] rsvd_1.0.3           qlcMatrix_0.9.7      survival_3.1-12
## [85] HSMMSingleCell_1.8.0 viridisLite_0.3.0    tibble_3.0.1
## [88] pheatmap_1.0.12      fastICA_1.2-2        densityClust_0.3
## [91] cluster_2.1.0        globals_0.12.5       fitdistrplus_1.1-1
## [94] ellipsis_0.3.1       ROCR_1.0-11
```