```r
#To run trajectory inference, monocle3 package is needed.
#To convert the seurat object wppgl.rds into monocle3-supportive format, seurat wrapper package is needed
#Installation instruction of monocle 3 package is in this link: https://cole-trapnell-lab.github.io/monocle3/docs/installation/
#Installation instruction of seurat wrapper is in this link: https://rdrr.io/github/satijalab/seurat-wrappers/f/README.md
#Viridis is used to make customized colormap. Instruction of installation is here https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html

#Load these packages using the following commands:
library(monocle3)
library(SeuratWrappers)
library(viridis)


#load the packages we used in integration and markers file as well
library(ggplot2)
library(tidyverse)
library(gridExtra)
library(Seurat)

#Here to utilize the non-integrated numbers to do monocle 3 analysis in integrated umap, the default assay type of wppgl is switched to RNA
#Integrated gene expression are scaled because of data integration, non-integrated RNA numbers are more representing what amount of genes
#these cells originally express without scaling up because the another condition had more expression.
#Integrated UMAP visualization was used however to show that these cells are not different because of batch effects because of two seperate sequencing experiments.

DefaultAssay(wppgl)<- "RNA"
#We run wild type and gl mutant seperately, so we split the wppgl object into two new objects, wpp_integrated and gl_integrated
wt_gl_list <-SplitObject(wppgl, split.by = "condition")
wpp_integrated <-wt_gl_list$wt
gl_integrated <-wt_gl_list$`gl mutant`

#Double check that these split file's default assay is RNA
DefaultAssay(wpp_integrated)
DefaultAssay(gl_integrated)

#Now we are ready to convert these seurat objects into monocle 3 supported forms using the seurat wrapper package
wpp_integrated_monocle3.cds <- as.cell_data_set(wpp_integrated)
gl_integrated_monocle3.cds <-as.cell_data_set(gl_integrated)

#This study is centered around photoreceptors, so we select photoreceptors manually using this command
#Click "done" when you done selecting to come out of the window.
PR_wt_cds <-choose_cells(wpp_integrated_monocle3.cds)
PR_gl_cds <-choose_cells(gl_integrated_monocle3.cds)

#choose the approporiate k number here to show seperation of different photoreceptor cell types. k number is set to 75 here
PR_wt_cds <- cluster_cells(PR_wt_cds, reduction_method = "UMAP", cluster_method = c("louvain"), random_seed =2, k=75)
#learn graph function does the trajectory inference. The wild type parameter is set as default. Learn graph may take a min to run.
PR_wt_cds <- learn_graph(PR_wt_cds, use_partition = TRUE)
#Choose the cells of origin here to designate the root nodes. Please refer to  already published raja.et al https://doi.org/10.1038/s41467-023-43037-0 figure 1D for inferring cell of origin. They should be located on the left
#if you used our seeds. We also included pictures of this step to help.
PR_wt_cds <-order_cells(PR_wt_cds)
#Load a customed colormap using package viridis
cpal1 <-c("#CFCFCF","#4662D7FF" ,"#36AAf9FF", "#1AE4B6FF", "#72FE5EFF" ,"#C7EF34FF",
          "#FABA39FF" ,"#F66B19FF" ,"#CB2A04FF" ,"#7A0403FF")
#Plot the photoreceptor trajectory in wild type condition
plot_cells(PR_wt_cds, reduction_method = "UMAP", color_cells_by = "pseudotime", cell_size = 0.6, trajectory_graph_segment_size = 1.5, label_branch_points = FALSE, label_leaves = FALSE) & scale_colour_gradientn(colours = cpal1, na.value = "lightgrey") & theme(legend.position = "right")& theme(axis.text = element_text(size = 20),axis.title.x.top = element_text(size = 50), axis.title = element_text(size = 20), plot.title = element_text(size = 20),legend.title = element_text(size =20),legend.text = element_text(size = 15), text = element_text(size =30, face = "bold"))

#Now we need to do the gl mutant trajectory
#choose the approporiate k number here to show seperation of different photoreceptor cell types. k number is set to 10 here
PR_gl_cds <-cluster_cells(PR_gl_cds, reduction_method = "UMAP", cluster_method = c("louvain"), k = 10, random_seed =1)
#Same as in wild type, we used default parameters for trajectory inference with learn graph. This may take a min.
PR_gl_cds <- learn_graph(PR_gl_cds, use_partition = TRUE)
#Choose the cells of origin here to designate the root nodes. Click done to finish and click clear to redo.
PR_gl_cds <-order_cells(PR_gl_cds)
#Plot the trajectory of gl mutant using the same color pallet in the wild type, which is the cpal1 that's already loaded
```

```
plot_cells(PR_gl_cds, reduction_method = "UMAP", color_cells_by = "pseudotime", cell_size = 1, trajectory_graph_segment_size =
1.5, label_branch_points = FALSE, label_leaves = FALSE) & scale_colour_gradientn(colours = cpal1, na.value = "lightgrey") &
theme(legend.position = "right")& theme(axis.text = element_text(size = 20),axis.title.x.top = element_text(size = 50),
axis.title = element_text(size = 20), plot.title = element_text(size = 20),legend.title = element_text(size = 20),legend.text
= element_text(size = 15), text = element_text(size =30, face = "bold"))
```