

Analysis of B-endorphin/ChAT IHC Count Data

C-T Berezin

11/21/2021

```
library(knitr)
library(tidyverse)
library(dplyr)
library(WebPower)

counts <- read_csv("../data/B-endo-totals.csv")

## Rows: 12 Columns: 12
## -- Column specification -----
## Delimiter: ","
## chr (3): Light, Time, Sex
## dbl (9): Mouse, ChAT, Bendo, Percentage, Age, GCLchat, GCLbendo, INLchat, IN...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
counts <- counts %>%
  dplyr::select(-ChAT, -Bendo, -Percentage) %>%
  #make columns lowercase
  rename(mouse = Mouse,
         light = Light,
         time = Time,
         sex = Sex,
         age = Age,
         GCL_chat = GCLchat,
         GCL_bendo = GCLbendo,
         INL_chat = INLchat,
         INL_bendo = INLbendo) %>%
  #update the classes of some columns
  mutate(mouse = as.character(mouse),
         light = factor(as.factor(light), levels=c("light", "dark")),
  #reinroduce total counts
         total_chat = GCL_chat+INL_chat,
         total_bendo = GCL_bendo+INL_bendo) %>%
  #calculate percentages
  mutate(GCL_perc=GCL_bendo/GCL_chat*100,
         INL_perc=INL_bendo/INL_chat*100,
         total_perc=total_bendo/total_chat*100)

#checking if the cell counts are normally distributed
shapiro.test(counts$INL_chat)
```

```

##
## Shapiro-Wilk normality test
##
## data: counts$INL_chat
## W = 0.89571, p-value = 0.1396
shapiro.test(counts$INL_bendo)

##
## Shapiro-Wilk normality test
##
## data: counts$INL_bendo
## W = 0.91934, p-value = 0.2805
shapiro.test(counts$GCL_chat)

##
## Shapiro-Wilk normality test
##
## data: counts$GCL_chat
## W = 0.88355, p-value = 0.09733
shapiro.test(counts$GCL_bendo)

##
## Shapiro-Wilk normality test
##
## data: counts$GCL_bendo
## W = 0.86341, p-value = 0.05396
shapiro.test(counts$total_chat)

##
## Shapiro-Wilk normality test
##
## data: counts$total_chat
## W = 0.88056, p-value = 0.08911
shapiro.test(counts$total_bendo)

##
## Shapiro-Wilk normality test
##
## data: counts$total_bendo
## W = 0.90417, p-value = 0.1795
percs <- counts %>%
  #give one row for each GCL, INL, and total
  #retain only the percentages, not the raw counts
  dplyr::select(-c(GCL_chat:total_bendo)) %>%
  pivot_longer(GCL_perc:total_perc,
               names_to="type",
               values_to="perc") %>%
  separate(type, sep="_", into=c("layer", "protein")) %>%
  dplyr::select(-protein) %>%
  mutate(layer = as.factor(layer),
         mouse = fct_reorder(as.factor(mouse), perc,
                             .fun=max, .desc=FALSE),

```

```
time = as.factor(time))
head(percs, 3)
```

```
## # A tibble: 3 x 7
##   mouse light time sex age layer perc
##   <fct> <fct> <fct> <chr> <dbl> <fct> <dbl>
## 1 11 light day F 95 GCL 3.14
## 2 11 light day F 95 INL 7.44
## 3 11 light day F 95 total 5.54
```

```
#checking if the percentages are normally distributed
gc <- percs %>% filter(layer == "GCL")
shapiro.test(gc$perc)
```

```
##
## Shapiro-Wilk normality test
##
## data: gc$perc
## W = 0.94973, p-value = 0.633
```

```
ic <- percs %>% filter(layer == "INL")
shapiro.test(ic$perc)
```

```
##
## Shapiro-Wilk normality test
##
## data: ic$perc
## W = 0.94583, p-value = 0.5771
```

```
tc <- percs %>% filter(layer == "total")
shapiro.test(tc$perc)
```

```
##
## Shapiro-Wilk normality test
##
## data: tc$perc
## W = 0.96135, p-value = 0.8029
```

```
layer_percs <- percs %>% filter(layer != "total")
aov <- aov(perc ~ light * time * layer, layer_percs)
summary(aov)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## light          1  139.1    139.1    6.110 0.025056 *
## time           1    9.0     9.0    0.397 0.537298
## layer          1  417.5    417.5   18.337 0.000571 ***
## light:time      1  258.2    258.2   11.341 0.003919 **
## light:layer     1   51.4     51.4    2.258 0.152427
## time:layer      1   12.9     12.9    0.565 0.463282
## light:time:layer 1   30.7     30.7    1.351 0.262231
## Residuals     16  364.3     22.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aov)
```

```
## Tukey multiple comparisons of means
```

```

##      95% family-wise confidence level
##
## Fit: aov(formula = perc ~ light * time * layer, data = layer_percs)
##
## $light
##           diff           lwr           upr           p adj
## dark-light 4.815019 0.6854227 8.944616 0.025056
##
## $time
##           diff           lwr           upr           p adj
## night-day 1.228134 -2.901462 5.357731 0.5372978
##
## $layer
##           diff           lwr           upr           p adj
## INL-GCL 8.341742 4.212145 12.47134 0.0005715
##
## $`light:time`
##           diff           lwr           upr           p adj
## dark:day-light:day 11.375105 3.49327511 19.256934 0.0039392
## light:night-light:day 7.788220 -0.09360982 15.670049 0.0533578
## dark:night-light:day 6.043154 -1.83867592 13.924983 0.1672187
## light:night-dark:day -3.586885 -11.46871446 4.294945 0.5748113
## dark:night-dark:day -5.331951 -13.21378056 2.549879 0.2528778
## dark:night-light:night -1.745066 -9.62689563 6.136763 0.9197070
##
## $`light:layer`
##           diff           lwr           upr           p adj
## dark:GCL-light:GCL 1.887984 -5.9938453 9.769814 0.9011799
## light:INL-light:GCL 5.414707 -2.4671227 13.296536 0.2414734
## dark:INL-light:GCL 13.156761 5.2749316 21.038591 0.0010656
## light:INL-dark:GCL 3.526723 -4.3551070 11.408552 0.5877993
## dark:INL-dark:GCL 11.268777 3.3869473 19.150606 0.0042612
## dark:INL-light:INL 7.742054 -0.1397752 15.623884 0.0550899
##
## $`time:layer`
##           diff           lwr           upr           p adj
## night:GCL-day:GCL -0.2357322 -8.117562 7.646097 0.9997654
## day:INL-day:GCL 6.8778753 -1.003954 14.759705 0.0986522
## night:INL-day:GCL 9.5698762 1.688047 17.451706 0.0149110
## day:INL-night:GCL 7.1136075 -0.768222 14.995437 0.0844234
## night:INL-night:GCL 9.8056084 1.923779 17.687438 0.0125448
## night:INL-day:INL 2.6920009 -5.189829 10.573830 0.7641311
##
## $`light:time:layer`
##           diff           lwr           upr           p adj
## dark:day:GCL-light:day:GCL 6.18425591 -7.3043521 19.672864 0.7510493
## light:night:GCL-light:day:GCL 4.06053948 -9.4280686 17.549148 0.9601054
## dark:night:GCL-light:day:GCL 1.65225203 -11.8363560 15.140860 0.9998280
## light:day:INL-light:day:GCL 1.68702656 -11.8015815 15.175635 0.9998025
## dark:day:INL-light:day:GCL 18.25297992 4.7643719 31.741588 0.0047085
## light:night:INL-light:day:GCL 13.20292650 -0.2856816 26.691535 0.0573262
## dark:night:INL-light:day:GCL 12.12108175 -1.3675263 25.609690 0.0950936
## light:night:GCL-dark:day:GCL -2.12371644 -15.6123245 11.364892 0.9991136
## dark:night:GCL-dark:day:GCL -4.53200388 -18.0206119 8.956604 0.9313412

```

```
## light:day:INL-dark:day:GCL      -4.49722936 -17.9858374  8.991379 0.9338179
## dark:day:INL-dark:day:GCL      12.06872400  -1.4198841 25.557332 0.0973977
## light:night:INL-dark:day:GCL    7.01867059  -6.4699375 20.507279 0.6278714
## dark:night:INL-dark:day:GCL    5.93682583  -7.5517822 19.425434 0.7848034
## dark:night:GCL-light:night:GCL -2.40828744 -15.8968955 11.080321 0.9980338
## light:day:INL-light:night:GCL   -2.37351292 -15.8621210 11.115095 0.9982051
## dark:day:INL-light:night:GCL    14.19244044   0.7038324 27.681049 0.0355493
## light:night:INL-light:night:GCL  9.14238703  -4.3462210 22.630995 0.3281466
## dark:night:INL-light:night:GCL  8.06054227  -5.4280658 21.549150 0.4713661
## light:day:INL-dark:night:GCL    0.03477453 -13.4538335 13.523383 1.0000000
## dark:day:INL-dark:night:GCL    16.60072789   3.1121198 30.089336 0.0107646
## light:night:INL-dark:night:GCL  11.55067447  -1.9379336 25.039283 0.1230287
## dark:night:INL-dark:night:GCL  10.46882971  -3.0197783 23.957438 0.1959052
## dark:day:INL-light:day:INL      16.56595336   3.0773453 30.054561 0.0109536
## light:night:INL-light:day:INL    11.51589995  -1.9727081 25.004508 0.1249448
## dark:night:INL-light:day:INL    10.43405519  -3.0545529 23.922663 0.1987393
## light:night:INL-dark:day:INL    -5.05005341 -18.5386615  8.438555 0.8875683
## dark:night:INL-dark:day:INL    -6.13189817 -19.6205062  7.356710 0.7583366
## dark:night:INL-light:night:INL -1.08184476 -14.5704528 12.406763 0.9999901
```

#set up a table of p-values for labeling the graphs below

```
label <- tibble(mouse = c("97", "93", "57"),
                 perc = c(Inf, Inf, Inf),
                 time = c("day", "night", "night"),
                 light = c("dark", "light", "dark"),
                 label = c("p=0.00009", "p=0.0058", "p=0.037"))
```

#1 row facet wrap, small

```
bendo_ihc <- percs %>% ggplot(aes(x=mouse, y=perc)) +
  geom_bar(aes(fill=layer), stat="identity", position="dodge") +
  facet_wrap(time ~ factor(light, c("light", "dark")), scales="free_x", nrow=1, as.table=FALSE) +
  labs(y="% of B-endorphin+ ChAT cells",
       col="Light Condition", shape="Cell Layer",
       title = "Figure 2") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        text = element_text(size=4),
        plot.title = element_text(size=4, margin=margin(0,0,0,0), face = "bold"),
        legend.key.size = unit(0.25, 'cm'),
        strip.text.x = element_text(size = 3, margin = margin(0, 0, 0.02, 0, "cm")),
        axis.ticks = element_blank(),
        legend.margin=margin(0,0,0,0)) +
  scale_fill_manual(values = c("yellow", "cornflowerblue", "grey50")) +
  geom_text(aes(label=label), data=label, vjust="top", hjust="left", size=1)

ggsave(filename="../figures/bendo_ihc.png", plot=bendo_ihc, height=1, width=2.5)
```

#1 row facet wrap, large

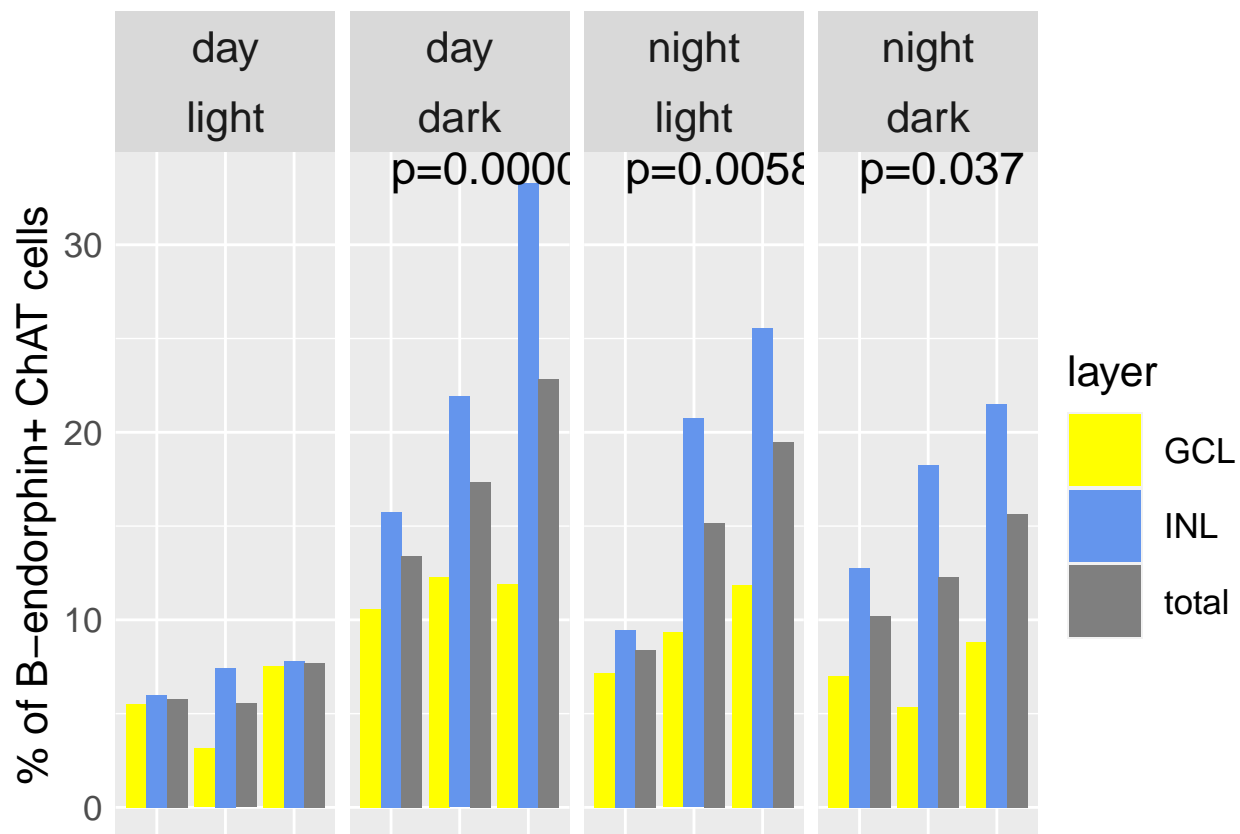
```
bendo_ihc_large <- percs %>% ggplot(aes(x=mouse, y=perc)) +
  geom_bar(aes(fill=layer), stat="identity", position="dodge") +
  facet_wrap(time ~ factor(light, c("light", "dark")), scales="free_x", nrow=1, as.table=FALSE) +
  labs(y="% of B-endorphin+ ChAT cells",
       col="Light Condition", shape="Cell Layer") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
```

```

text = element_text(size=16),
legend.key.size = unit(1, 'cm'),
strip.text.x = element_text(size = 16, margin = margin(0.2, 0, 0.2, 0, "cm")),
axis.ticks = element_blank(),
legend.margin=margin(0,0,0,0)) +
scale_fill_manual(values = c("yellow", "cornflowerblue", "grey50")) +
geom_text(aes(label=label), data=label, vjust="top", hjust="left", size=6)

```

bendo_ihc_large



```
ggsave(filename="../figures/bendo_ihc_large.png", plot=bendo_ihc_large, height=5, width=12)
```

```

#2 BY 2 FACET WRAP
bendo_ihc_boxxy <- percs %>% ggplot(aes(x=mouse, y=perc)) +
  geom_bar(aes(fill=layer), stat="identity", position="dodge") +
  facet_wrap(time ~ factor(light, c("light", "dark")), scales="free_x", nrow=2, as.table=TRUE) +
  labs(y="% of B-endorphin+ ChAT cells",
       col="Light Condition", shape="Cell Layer") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        text = element_text(size=4),
        plot.title = element_text(size=4, margin=margin(0,0,0,0), face = "bold"),
        legend.key.size = unit(0.25, 'cm'),
        strip.text.x = element_text(size = 3, margin = margin(0, 0, 0.02, 0, "cm")),
        axis.ticks = element_blank(),
        legend.margin=margin(0,0,0,0)) +

```

```
scale_fill_manual(values = c("yellow", "cornflowerblue", "grey50")) +  
geom_text(aes(label=label), data=label, vjust="top", hjust="left", size=1)  
  
ggsave(filename="../figures/bendo_ihc_boxxy.png", plot=bendo_ihc_boxxy, height=4, width=4)
```