

Analysis of B-endorphin/ChAT IHC Count Data

C-T Berezin

11/21/2021

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

```
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")),
         time = as.factor(time),
         #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
```

```
#data setup for previous anova analysis
#also used in following shapiro tests for ease
anovpercs <- 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)) %>%
  filter(layer != "total")
head(anovpercs, 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 5 light day M 141 GCL 7.53
```

```
manset <- counts %>% dplyr::select(light:time, GCL_perc:INL_perc)

#testing manova assumptions
#multivariate normality
#normal distribution for each combo of indep and dependent variables
anovpercs %>% filter(light == "light" & layer == "GCL") %>% pull(perc) %>% shapiro.test(.)
```

```
##
## Shapiro-Wilk normality test
##
## data: .
## W = 0.99193, p-value = 0.9934
```

```
anovpercs %>% filter(light == "dark" & layer == "GCL") %>% pull(perc) %>% shapiro.test(.)
```

```
##
## Shapiro-Wilk normality test
##
## data: .
## W = 0.93389, p-value = 0.6104
```

```
anovpercs %>% filter(light == "light" & layer == "INL") %>% pull(perc) %>% shapiro.test(.)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: .  
## W = 0.80407, p-value = 0.06392
```

```
anovpercs %>% filter(light == "dark" & layer == "INL") %>% pull(perc) %>% shapiro.test(.)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: .  
## W = 0.91084, p-value = 0.442
```

```
anovpercs %>% filter(time == "day" & layer == "GCL") %>% pull(perc) %>% shapiro.test(.)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: .  
## W = 0.9171, p-value = 0.4847
```

```
anovpercs %>% filter(time == "night" & layer == "GCL") %>% pull(perc) %>% shapiro.test(.)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: .  
## W = 0.96858, p-value = 0.8828
```

```
anovpercs %>% filter(time == "day" & layer == "INL") %>% pull(perc) %>% shapiro.test(.)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: .  
## W = 0.86996, p-value = 0.226
```

```
anovpercs %>% filter(time == "night" & layer == "INL") %>% pull(perc) %>% shapiro.test(.)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: .  
## W = 0.95699, p-value = 0.7962
```

```
#all pass
```

```
anovpercs %>% filter(light == "light" & layer == "GCL") %>% pull(perc) %>% shapiro.test(.)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: .  
## W = 0.99193, p-value = 0.9934
```

```
#checking homogenous variance in each combo  
#null hyp is that variance is equal  
car::leveneTest(manset$GCL_perc, group=manset$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##      Df F value Pr(>F)  
## group 1  0.0151 0.9047  
##      10
```

```
car::leveneTest(manset$INL_perc, group=manset$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##      Df F value Pr(>F)  
## group 1  0.0504 0.8268  
##      10
```

```
car::leveneTest(manset$GCL_perc, group=manset$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##      Df F value Pr(>F)  
## group 1  2.5789 0.1394  
##      10
```

```
car::leveneTest(manset$INL_perc, group=manset$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)  
##      Df F value Pr(>F)  
## group 1  1.3768 0.2678  
##      10
```

```
#checking for multicollinearity  
#low values are good  
inllm <- lm(INL_perc ~ light * time, manset)  
car::vif(inllm)
```

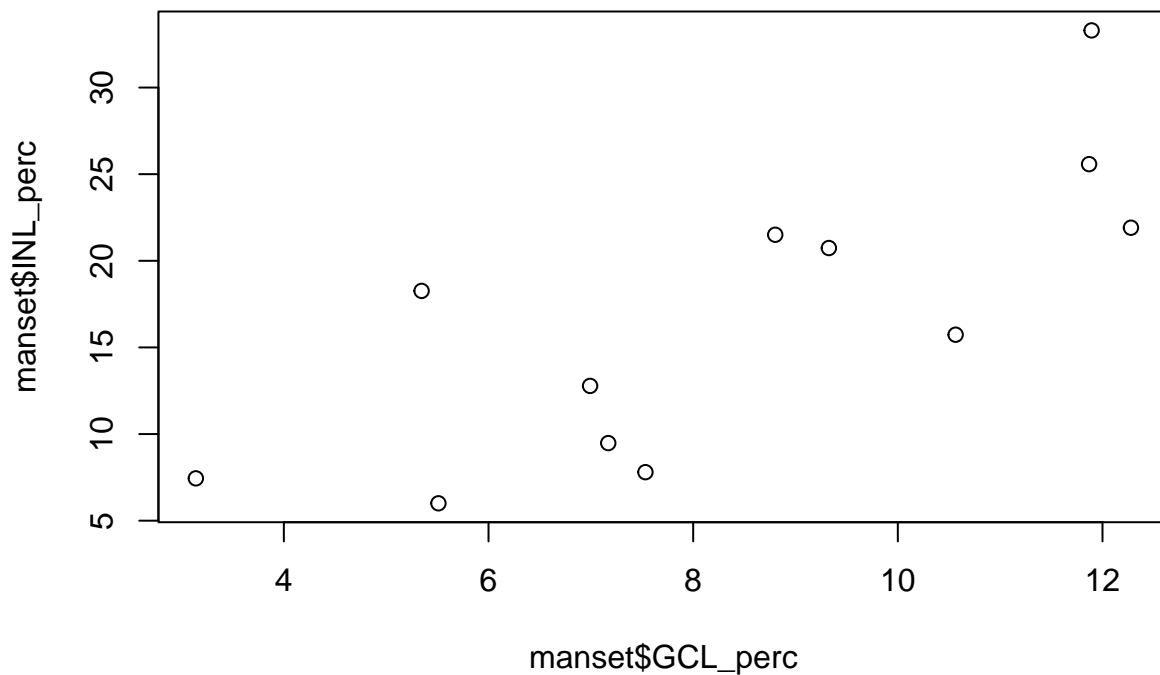
```
##      light      time light:time  
##         2         2           3
```

```
gcllm <- lm(GCL_perc ~ light * time, manset)
car::vif(gcllm)
```

```
##      light      time light:time
##      2        2        3
```

```
#linearity of dep variables
```

```
plot(manset$GCL_perc, manset$INL_perc)
```



```
#good to go on manova!
```

```
#manova
```

```
mantest <- manova(cbind(GCL_perc, INL_perc) ~ light * time, data=manset)
summary(mantest)
```

```
##           Df  Pillai approx F num Df den Df  Pr(>F)
## light      1  0.37461    2.0965     2     7  0.19343
## time       1  0.11699    0.4637     2     7  0.64697
## light:time  1  0.66218    6.8605     2     7  0.02241 *
## Residuals   8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.aov(mantest)
```

```
## Response GCL_perc :
##           Df Sum Sq Mean Sq F value    Pr(>F)
## light      1 10.693   10.693   3.0196 0.120464
## time       1  0.167    0.167   0.0471 0.833664
## light:time  1 55.374   55.374 15.6365 0.004211 **
## Residuals   8 28.331    3.541
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Response INL_perc :
##           Df Sum Sq Mean Sq F value    Pr(>F)
## light      1 179.82 179.818   4.2818 0.07231 .
## time       1  21.74   21.741   0.5177 0.49231
## light:time  1 233.58 233.584   5.5621 0.04607 *
## Residuals   8 335.96   41.996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#light:time interaction is significant overall and for GCL and INL
#look at comparisons through univariate models for each dep variable
summary(gcllm)
```

```
##
## Call:
## lm(formula = GCL_perc ~ light * time, data = manset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.28500 -1.18579  0.03093  0.96322  2.41406
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.395      1.086   4.965 0.00110 **
## lightdark         6.184      1.537   4.025 0.00382 **
## timenight         4.061      1.537   2.643 0.02959 *
## lightdark:timenight -8.593      2.173  -3.954 0.00421 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.882 on 8 degrees of freedom
## Multiple R-squared:  0.7004, Adjusted R-squared:  0.5881
## F-statistic: 6.234 on 3 and 8 DF,  p-value: 0.01728
```

```
summary(inllm)
```

```
##
## Call:
## lm(formula = INL_perc ~ light * time, data = manset)
##
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -9.1204 -2.4874  0.5396  2.6011  9.6505
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)         7.082      3.741   1.893   0.0950 .
## lightdark           16.566      5.291   3.131   0.0140 *
## timenight           11.516      5.291   2.176   0.0612 .
## lightdark:timenight -17.648      7.483  -2.358   0.0461 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.48 on 8 degrees of freedom
## Multiple R-squared:  0.5643, Adjusted R-squared:  0.4009
## F-statistic: 3.454 on 3 and 8 DF,  p-value: 0.07147
```

```
plotting_manset <- counts %>%
  dplyr::select(mouse, light:time, GCL_perc:INL_perc)

gclplotmanset <- plotting_manset %>%
  mutate(mouse = fct_reorder(as.factor(mouse), GCL_perc,
                             .fun=max, .desc=FALSE))

gclplot <- gclplotmanset %>% ggplot(aes(x=mouse, y=GCL_perc)) +
  geom_bar(stat="identity") +
  facet_wrap(time ~ factor(light, c("light", "dark")), scales="free_x", nrow=1, as.table=FALSE) +
  expand_limits(y=c(0,20)) +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x = element_blank()) +
  labs(y="Ganglion cell layer") +
  scale_y_continuous(labels=function(x) paste0(x,"%"))

inlplotmanset <- plotting_manset %>%
  mutate(mouse = fct_reorder(as.factor(mouse), INL_perc,
                             .fun=max, .desc=FALSE))

inlplot <- inlplotmanset %>% ggplot(aes(x=mouse, y=INL_perc)) +
  geom_bar(stat="identity") +
  facet_wrap(time ~ factor(light, c("light", "dark")), scales="free_x", nrow=1, as.table=FALSE) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.3)),
                    labels=function(x) paste0(x,"%")) +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x = element_blank()) +
  labs(y="Inner nuclear layer")

bothplots <- ggarrange(inlplot, gclplot, nrow=2)

bothplots <- annotate_figure(bothplots, top="% of ChAT cells expressing \U03B22-endorphin")

ggsave("../figures/manova_bendo_ihc.png", plot=bothplots, width=6, height=6)
```



```
aov <- aov(perc ~ light * time * layer, anovpercs)
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 = anovpercs)
##
## $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 <- anovpercs %>% 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)

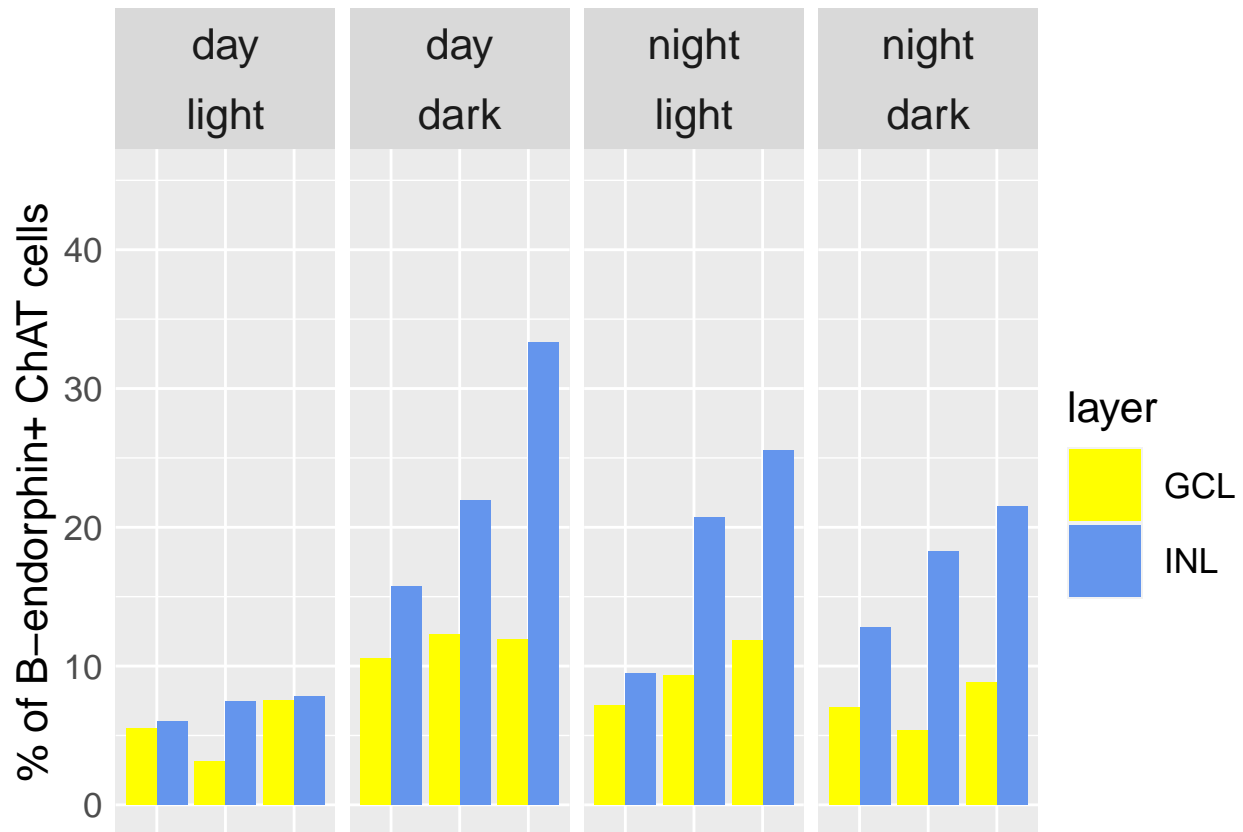
```

```

#1 row facet wrap, large
bendo_ihc_large <- anovpercs %>% 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")) +
  expand_limits(y=c(0,45))
#geom_text(aes(label=label), data=label, vjust="top", hjust="left", size=6)

bendo_ihc_large

```



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

```
#2 BY 2 FACET WRAP
bendo_ihc_boxy <- anovpercs %>% 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_boxy.png", plot=bendo_ihc_boxy, height=4, width=4)
```