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)
library(ggthemes)
library(viridis)
library(writexl)
```

Reading in and Tidying the Data

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

```
## Rows: 12 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Light, Time, Sex
## dbl (10): Mouse, ChAT, Bendo, Percentage, Age, GCLchat, GCLbendo, INLchat, I...
```

```
##
## 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.factor(mouse),
          light = factor(as.factor(light), levels=c("light", "dark")),
          time = as.factor(time),
  #reintroduce 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)
counts
```

```
## # A tibble: 12 x 15
##    mouse light time  sex     age GCL_chat GCL_bendo INL_chat INL_bendo  Area
##    <fct> <fct> <fct> <chr> <dbl>    <dbl>     <dbl>    <dbl>     <dbl> <dbl>
##  1 11    light day   F        95     4427       139     5589       416  5.22
##  2 5     light day   M       141     1646       124     2013       157  2.44
##  3 32    light day   M        65     2722       150     3232       194  3.34
##  4 57    dark  night F        54     2474       173     3083       394  3.31
##  5 47    dark  night F        97     4374       385     5134      1104  6.32
##  6 20    dark  night F       122     5088       272     5886      1075  5.64
##  7 31    dark  day   M        70     4515       537     4730      1575  4.43
##  8 34    dark  day   F        70     4113       505     4587      1005  4.54
##  9 97    dark  day   M        74     4373       462     5313       836  5.62
## 10 92    light night F        96     4221       501     5266      1347  5.17
## 11 44    light night M        41     4986       465     5242      1087  3.56
## 12 93    light night F        96     3138       225     3672       348  6.16
## # ... with 5 more variables: total_chat <dbl>, total_bendo <dbl>,
## #   GCL_perc <dbl>, INL_perc <dbl>, total_perc <dbl>
```

```
#manova for ChAT cells per mm2
counts <- counts %>% mutate(
  GCL_chat_mm = GCL_chat/Area,
  GCL_bendo_mm = GCL_bendo/Area,
  INL_chat_mm = INL_chat/Area,
  INL_bendo_mm = INL_bendo/Area
)

counts %>% group_by(light, time) %>%  summarise(
  min_INL_chat_mm = min(INL_chat_mm),
  mean_INL_chat_mm = mean(INL_chat_mm),
  max_INL_chat_mm = max(INL_chat_mm),
  min_GCL_chat_mm = min(GCL_chat_mm),
  mean_GCL_chat_mm = mean(GCL_chat_mm),
  max_GCL_chat_mm = max(GCL_chat_mm)
)
```

```
## `summarise()` has grouped output by 'light'. You can override using the `.groups` argument.
```

```
## # A tibble: 4 x 8
## # Groups:   light [2]
##   light time  min_INL_chat_mm mean_INL_chat_mm max_INL_chat_mm min_GCL_chat_mm
##   <fct> <fct>           <dbl>            <dbl>           <dbl>           <dbl>
## 1 light day              825              954.           1071.            675.
## 2 light night            596.            1029.           1472.            509.
## 3 dark  day              945.            1008.           1068.            778.
## 4 dark  night            812.             929.           1044.            692.
## # ... with 2 more variables: mean_GCL_chat_mm <dbl>, max_GCL_chat_mm <dbl>
```

```
shapiro.test(counts$GCL_chat_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts$GCL_chat_mm
## W = 0.88757, p-value = 0.1096
```

```
shapiro.test(counts$INL_chat_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts$INL_chat_mm
## W = 0.89834, p-value = 0.151
```

```
counts_long <- counts %>%
  #give one row for each GCL, INL, and total
  #retain only the percentages, not the raw counts
  dplyr::select(-c(GCL_chat:INL_bendo, total_chat:total_perc)) %>%
  pivot_longer(GCL_chat_mm:INL_bendo_mm,
               names_to="type",
               values_to="cells_mm") %>%
  separate(type, sep="_", into=c("layer", "protein", "size")) %>%
  dplyr::select(-size) %>%
  mutate(layer = as.factor(layer),
         protein = as.factor(protein))

#testing manova assumptions
#multivariate normality
#normal distribution for each combo of indep and dependent variables
counts_long %>% filter(layer =="GCL" & protein == "chat" & time == "day") %>% pull(cells_mm) %>% shapiro
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.99202, p-value = 0.9935
```

```
counts_long %>% filter(layer =="GCL" & protein == "chat" & time == "night") %>% pull(cells_mm) %>% shap
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.88836, p-value = 0.3097
```

```
counts_long %>% filter(layer =="INL" & protein == "chat" & time == "day") %>% pull(cells_mm) %>% shapiro
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.9109, p-value = 0.4424
```

```
counts_long %>% filter(layer =="INL" & protein == "chat" & time == "night") %>% pull(cells_mm) %>% shap
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  .
## W = 0.95001, p-value = 0.7404
```

```
counts_long %>% filter(layer =="GCL" & protein == "chat" & light == "light") %>% pull(cells_mm) %>% shap
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.85537, p-value = 0.1739
```

```
counts_long %>% filter(layer =="GCL" & protein == "chat" & light == "dark") %>% pull(cells_mm) %>% shapi
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.94687, p-value = 0.7149
```

```
counts_long %>% filter(layer =="INL" & protein == "chat" & light == "light") %>% pull(cells_mm) %>% shap
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.95916, p-value = 0.8133
```

```
counts_long %>% filter(layer =="INL" & protein == "chat" & light == "dark") %>% pull(cells_mm) %>% shapi
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.93201, p-value = 0.5957
```

```
#checking homogenous variance in each combo
car::leveneTest(counts$GCL_chat_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.6201 0.4493
##       10
```

```
car::leveneTest(counts$INL_chat_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  2.2117 0.1678
##       10
```

```
car::leveneTest(counts$GCL_chat_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  1.3156 0.2781
##       10
```

```
car::leveneTest(counts$INL_chat_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
```

```
##       Df F value Pr(>F)
## group  1  2.5293 0.1428
##       10
```

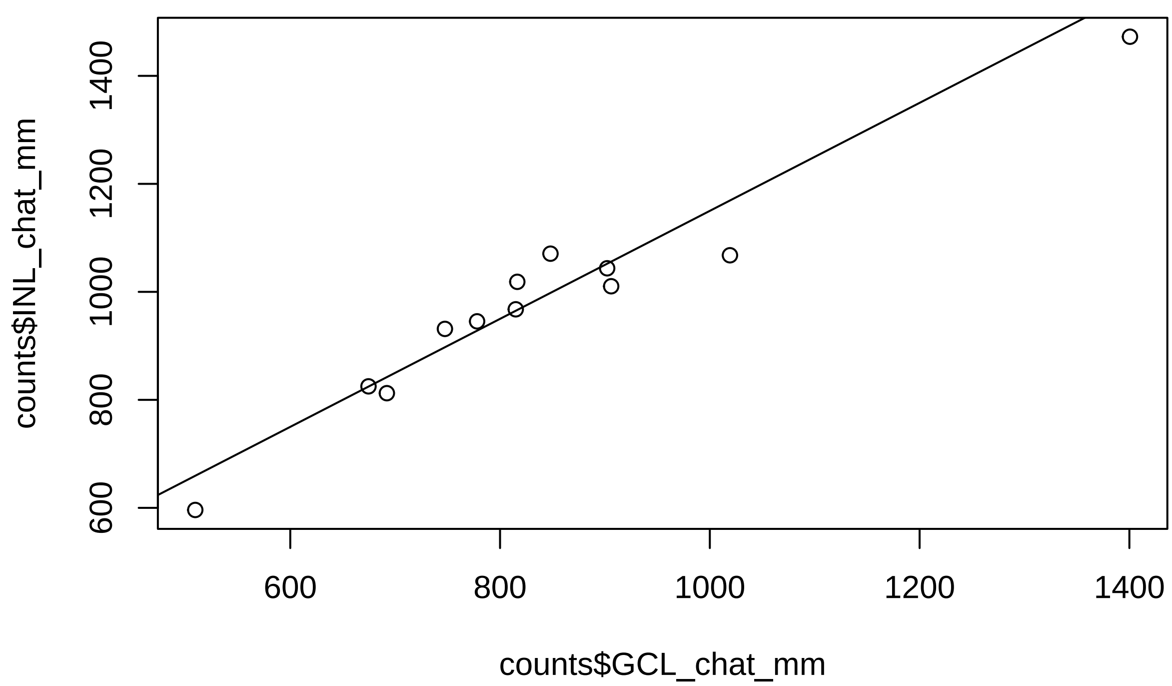
```
#checking for multicollinearity
#low values are good
inllm <- lm(INL_chat_mm ~ light * time, counts)
car::vif(inllm)
```

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

```
gcllm <- lm(GCL_chat_mm ~ light * time, counts)
car::vif(gcllm)
```

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

```
#linearity of dep variables
plot(counts$GCL_chat_mm, counts$INL_chat_mm)
abline(150,1)
```




```
#good to go on manova!

#manova
mantest <- manova(cbind(GCL_chat_mm, INL_chat_mm) ~ light * time, data=counts)
summary(mantest)
```

```
##            Df   Pillai approx F num Df den Df Pr(>F)
```

```
## light       1 0.054947  0.20350      2      7 0.8205
## time        1 0.005750  0.02024      2      7 0.9800
## light:time  1 0.254205  1.19298      2      7 0.3582
## Residuals   8
```

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

```
##  Response GCL_chat_mm :
##             Df Sum Sq Mean Sq F value Pr(>F)
## light        1     31      31  0.0005 0.9825
## time         1     62      62  0.0010 0.9752
## light:time   1  46922   46922  0.7826 0.4021
## Residuals    8 479633   59954
##
##  Response INL_chat_mm :
##             Df Sum Sq Mean Sq F value Pr(>F)
## light        1   1626    1626  0.0290 0.8691
## time         1     13      13  0.0002 0.9884
## light:time   1  17622   17622  0.3141 0.5905
## Residuals    8 448865   56108
```

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

```
## Analysis of Variance Table
##
## Response: GCL_chat_mm
##            Df Sum Sq Mean Sq F value Pr(>F)
## light       1     31      31  0.0005 0.9825
## time        1     62      62  0.0010 0.9752
## light:time  1  46922   46922  0.7826 0.4021
## Residuals   8 479633   59954
```

```r
anova(inllm)
```

```
## Analysis of Variance Table
##
## Response: INL_chat_mm
##            Df Sum Sq Mean Sq F value Pr(>F)
## light       1   1626    1626  0.0290 0.8691
## time        1     13      13  0.0002 0.9884
## light:time  1  17622   17622  0.3141 0.5905
## Residuals   8 448865   56108
```

```r
#manova for bendo cells per mm2
counts %>% group_by(light, time) %>% summarise(
  min_INL_bendo_mm = min(INL_bendo_mm),
  mean_INL_bendo_mm = mean(INL_bendo_mm),
  max_INL_bendo_mm = max(INL_bendo_mm),
  min_GCL_bendo_mm = min(GCL_bendo_mm),
  mean_GCL_bendo_mm = mean(GCL_bendo_mm),
  max_GCL_bendo_mm = max(GCL_bendo_mm)
)
```

```
## `summarise()` has grouped output by 'light'. You can override using the `.groups` argument.

## # A tibble: 4 x 8
```

```
## # Groups:   light [2]
##   light time  min_INL_bendo_mm mean_INL_bendo_mm max_INL_bendo_mm
##   <fct> <fct>            <dbl>             <dbl>            <dbl>
## 1 light day               58.1              67.4             79.7
## 2 light night             56.5             207.             305.
## 3 dark  day              149.              242.             356.
## 4 dark  night            119.              161.             191.
## # ... with 3 more variables: min_GCL_bendo_mm <dbl>, mean_GCL_bendo_mm <dbl>,
## #   max_GCL_bendo_mm <dbl>
```

```
shapiro.test(counts$GCL_bendo_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts$GCL_bendo_mm
## W = 0.90829, p-value = 0.2028
```

```
shapiro.test(counts$INL_bendo_mm)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  counts$INL_bendo_mm
## W = 0.92663, p-value = 0.3458
```

```
#testing manova assumptions
#multivariate normality
#normal distribution for each combo of indep and dependent variables
counts_long %>% filter(layer =="GCL" & protein == "bendo" & time == "day") %>% pull(cells_mm) %>% shapi
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.92539, p-value = 0.545
```

```
counts_long %>% filter(layer =="GCL" & protein == "bendo" & time == "night") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.87924, p-value = 0.2656
```

```
counts_long %>% filter(layer =="INL" & protein == "bendo" & time == "day") %>% pull(cells_mm) %>% shapi
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.85792, p-value = 0.1821
```

```
counts_long %>% filter(layer =="INL" & protein == "bendo" & time == "night") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  .
## W = 0.98211, p-value = 0.9616
```

```
counts_long %>% filter(layer =="GCL" & protein == "bendo" & light == "light") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.86634, p-value = 0.212
```

```
counts_long %>% filter(layer =="GCL" & protein == "bendo" & light == "dark") %>% pull(cells_mm) %>% shap
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.88645, p-value = 0.3
```

```
counts_long %>% filter(layer =="INL" & protein == "bendo" & light == "dark") %>% pull(cells_mm) %>% shap
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.87501, p-value = 0.2469
```

```
#did not pass, but transformations don't either
counts_long %>% filter(layer =="INL" & protein == "bendo" & light == "light") %>% pull(cells_mm) %>% sha
```

```
##
##  Shapiro-Wilk normality test
##
## data:  .
## W = 0.73385, p-value = 0.01378
```

```
#checking homogenous variance in each combo
car::leveneTest(counts$GCL_bendo_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.0384 0.8485
##       10
```

```
car::leveneTest(counts$INL_bendo_mm, group=counts$light)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2279 0.6434
##       10
```

```
car::leveneTest(counts$GCL_bendo_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2692 0.6152
##       10
```

```
car::leveneTest(counts$INL_bendo_mm, group=counts$time)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2504 0.6276
##       10
```

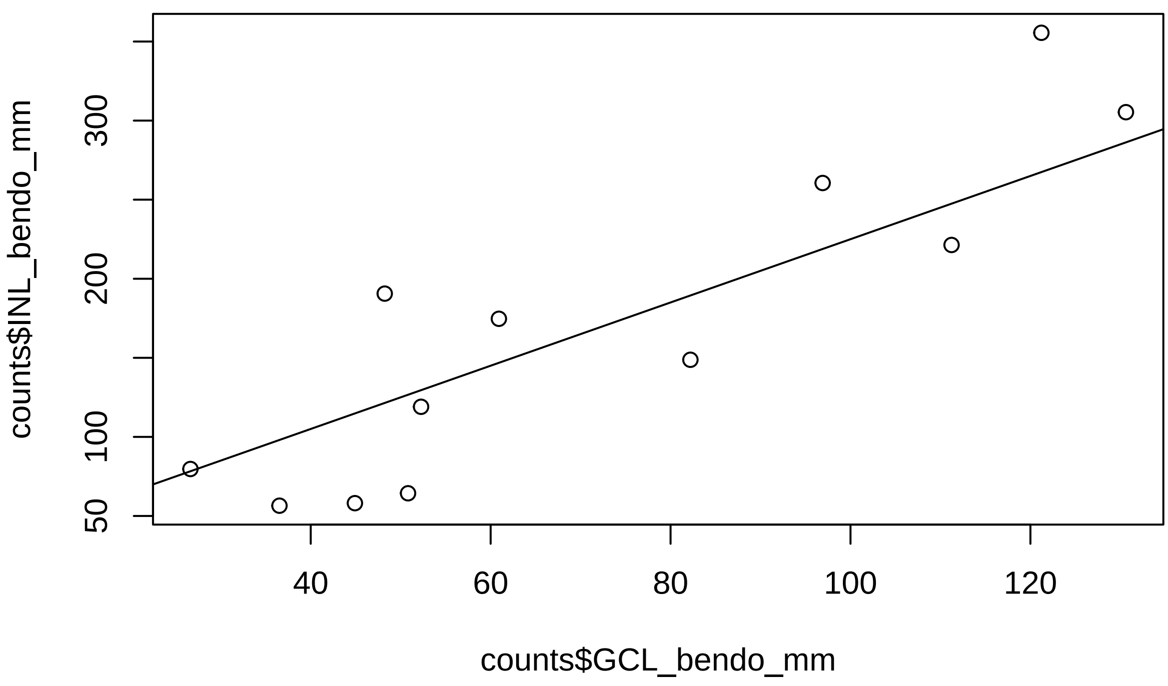
```r
#checking for multicollinearity
#low values are good
inl_bendo_lm <- lm(INL_bendo_mm ~ light * time, counts)
car::vif(inl_bendo_lm)
```

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

```r
gcl_bendo_lm <- lm(GCL_bendo_mm ~ light * time, counts)
car::vif(gcl_bendo_lm)
```

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

```r
#linearity of dep variables
plot(counts$GCL_bendo_mm, counts$INL_bendo_mm)
abline(25,2)
```




```r
#good to go on manova!

#manova
man_bendo_test <- manova(cbind(GCL_bendo_mm, INL_bendo_mm) ~ light * time, data=counts)
summary(man_bendo_test)
```

```
##            Df  Pillai approx F num Df den Df  Pr(>F)
## light       1 0.17650   0.7502      2      7 0.50678
## time        1 0.19087   0.8256      2      7 0.47651
## light:time  1 0.58085   4.8503      2      7 0.04767 *
## Residuals   8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.aov(man_bendo_test)
```

```
##  Response GCL_bendo_mm :
##             Df Sum Sq Mean Sq F value  Pr(>F)
## light        1  669.9   669.9  0.9291 0.36334
## time         1   11.1    11.1  0.0154 0.90419
## light:time   1 7249.1  7249.1 10.0529 0.01318 *
## Residuals    8 5768.8   721.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response INL_bendo_mm :
##             Df Sum Sq Mean Sq F value  Pr(>F)
## light        1  12383   12383  1.6436 0.23573
## time         1   2668    2668  0.3541 0.56825
## light:time   1  36474   36474  4.8415 0.05896 .
## Residuals    8  60270    7534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
## Analysis of Variance Table
##
## Response: GCL_bendo_mm
##            Df Sum Sq Mean Sq F value  Pr(>F)
## light       1  669.9   669.9  0.9291 0.36334
## time        1   11.1    11.1  0.0154 0.90419
## light:time  1 7249.1  7249.1 10.0529 0.01318 *
## Residuals   8 5768.8   721.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(gcl_bendo_lm)
```

```
##
## Call:
## lm(formula = GCL_bendo_mm ~ light * time, data = counts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.490  -7.722   5.236   9.175  42.602
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)             40.79      15.50   2.631   0.0301 *
```

```
## lightdark              64.10      21.93   2.924   0.0192 *
## timenight              47.23      21.93   2.154   0.0634 .
## lightdark:timenight   -98.31      31.01  -3.171   0.0132 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.85 on 8 degrees of freedom
## Multiple R-squared:  0.5789, Adjusted R-squared:  0.421
## F-statistic: 3.666 on 3 and 8 DF,  p-value: 0.06294
```

```
anova(inl_bendo_lm)
```

```
## Analysis of Variance Table
##
## Response: INL_bendo_mm
##            Df Sum Sq Mean Sq F value  Pr(>F)
## light       1  12383   12383  1.6436 0.23573
## time        1   2668    2668  0.3541 0.56825
## light:time  1  36474   36474  4.8415 0.05896 .
## Residuals   8  60270    7534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(inl_bendo_lm)
```

```
##
## Call:
## lm(formula = INL_bendo_mm ~ light * time, data = counts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -150.964  -25.990    4.645   35.143  113.647
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)            67.37      50.11   1.344   0.2157
## lightdark             174.51      70.87   2.462   0.0392 *
## timenight             140.08      70.87   1.977   0.0835 .
## lightdark:timenight  -220.53     100.22  -2.200   0.0590 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86.8 on 8 degrees of freedom
## Multiple R-squared:  0.4609, Adjusted R-squared:  0.2587
## F-statistic:  2.28 on 3 and 8 DF,  p-value: 0.1563
```

```
emmeans::emmeans(gcl_bendo_lm, pairwise ~ light * time, adjust="none")$contrasts
```

```
##  contrast                 estimate   SE df t.ratio p.value
##  light day - dark day        -64.1 21.9  8  -2.924  0.0192
##  light day - light night     -47.2 21.9  8  -2.154  0.0634
##  light day - dark night      -13.0 21.9  8  -0.594  0.5691
##  dark day - light night       16.9 21.9  8   0.769  0.4638
##  dark day - dark night        51.1 21.9  8   2.330  0.0482
##  light night - dark night     34.2 21.9  8   1.560  0.1573
```

```r
emmeans::emmeans(inl_bendo_lm, pairwise ~ light * time, adjust="none")$contrasts
```

```
##  contrast                  estimate   SE df t.ratio p.value
##  light day - dark day        -174.5 70.9  8  -2.462  0.0392
##  light day - light night     -140.1 70.9  8  -1.977  0.0835
##  light day - dark night       -94.1 70.9  8  -1.327  0.2210
##  dark day - light night        34.4 70.9  8   0.486  0.6402
##  dark day - dark night         80.4 70.9  8   1.135  0.2892
##  light night - dark night      46.0 70.9  8   0.649  0.5343
```

```r
#plotting bendo/mm2
sumstats <- counts %>% group_by(light, time) %>%
            summarise(n = n(),
                      mean_inl = mean(INL_bendo_mm),
                      sd_inl = sd(INL_bendo_mm),
                      mean_gcl = mean(GCL_bendo_mm),
                      sd_gcl = sd(GCL_bendo_mm))
```

```
## `summarise()` has grouped output by 'light'. You can override using the `.groups` argument.
```

```r
gcl_mm_plot <- ggplot() +
  geom_bar(data=sumstats, aes(x=light, y=mean_gcl, fill=light), stat="identity") +
  geom_errorbar(data=sumstats, aes(x=light,
                               ymin=mean_gcl-sd_gcl,
                               ymax=mean_gcl+sd_gcl), width=0.2) +
  geom_point(data=counts, aes(x=light, y=GCL_bendo_mm)) +
  facet_wrap(~time) +
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        panel.spacing = unit(0, "pt"),
        legend.position="none") +
  labs(y=expression(paste("# ", beta, "-endorphin+ cells in GCL /" ~mm^2))) +
  scale_fill_manual(values=c("yellow", "gray70")) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.4)))

inl_mm_plot <- ggplot() +
  geom_bar(data=sumstats, aes(x=light, y=mean_inl, fill=light), stat="identity") +
  geom_errorbar(data=sumstats, aes(x=light,
                               ymin=mean_inl-sd_inl,
                               ymax=mean_inl+sd_inl), width=0.2) +
  geom_point(data=counts, aes(x=light, y=INL_bendo_mm)) +
  facet_wrap(~time) +
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.ticks.x = element_blank(),
        panel.spacing = unit(0, "pt"),
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        legend.position="none") +
  labs(y=expression(paste("# ", beta, "-endorphin+ cells in INL /" ~mm^2))) +
  scale_fill_manual(values=c("yellow", "gray70")) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.3)))
```

```
#,
#        plot.background = element_rect(fill = "transparent",colour = NA),
#        panel.background = element_rect(fill = "transparent",colour = NA)

both_mm_plots <- ggarrange(gcl_mm_plot, inl_mm_plot, nrow=2)

ggsave("../figures/both_mm_plots.png", plot=both_mm_plots, height=6, width=6, bg = "transparent")

ctrl_counts <- counts %>% filter(light == "light" & time == "day")

#between var
sum(3*(mean(counts$GCL_bendo_mm - mean(ctrl_counts$GCL_bendo_mm)))^2)
```

```
## [1] 2899.204
```

```
#within.var
anova(gcl_bendo_lm)["Residuals", "Mean Sq"]
```

```
## [1] 721.0983
```

```
power.anova.test(groups=4, n=3, between.var=2899.204,
                 within.var=721.0983, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 3
##     between.var = 2899.204
##      within.var = 721.0983
##       sig.level = 0.05
##           power = 0.9820665
##
## NOTE: n is number in each group
```

```
#between var
sum(3*(mean(counts$INL_bendo_mm - mean(ctrl_counts$INL_bendo_mm)))^2)
```

```
## [1] 31312.94
```

```
#within.var
anova(inl_bendo_lm)["Residuals", "Mean Sq"]
```

```
## [1] 7533.694
```

```
power.anova.test(groups=4, n=3, between.var=31312.94,
                 within.var=7533.694, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 3
##     between.var = 31312.94
##      within.var = 7533.694
##       sig.level = 0.05
##           power = 0.9849714
##
```

```
## NOTE: n is number in each group
```

```r
#percentage / mm2
counts <- counts %>% mutate(inl_perc_mm = INL_bendo_mm/INL_chat_mm*100,
                            gcl_perc_mm = GCL_bendo_mm/GCL_chat_mm*100)

perc_sumstats <- counts %>% group_by(light, time) %>%
            summarise(n = n(),
                      mean_inl_perc = mean(inl_perc_mm),
                      sd_inl_perc = sd(inl_perc_mm),
                      mean_gcl_perc = mean(gcl_perc_mm),
                      sd_gcl_perc = sd(gcl_perc_mm))
```

```
## `summarise()` has grouped output by 'light'. You can override using the `.groups` argument.
```

```r
shapiro.test(counts$gcl_perc_mm)
```

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

```r
shapiro.test(counts$inl_perc_mm)
```

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

```r
counts_perc_long <- counts %>%
  #give one row for each GCL, INL, and total
  #retain only the percentages, not the raw counts
  dplyr::select(-c(GCL_chat:INL_bendo, total_chat:INL_bendo_mm)) %>%
  pivot_longer(inl_perc_mm:gcl_perc_mm,
               names_to="type",
               values_to="perc_mm") %>%
  separate(type, sep="_", into=c("layer", "na", "size")) %>%
  dplyr::select(-c(na,size)) %>%
  mutate(layer = as.factor(layer))

#testing manova assumptions
#multivariate normality
#normal distribution for each combo of indep and dependent variables
counts_perc_long %>% filter(layer =="gcl" & time == "day") %>% pull(perc_mm) %>% shapiro.test(.)
```

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

```r
counts_perc_long %>% filter(layer =="gcl" & time == "night") %>% pull(perc_mm) %>% shapiro.test(.)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  .
## W = 0.96858, p-value = 0.8828
```

```r
counts_perc_long %>% filter(layer =="gcl" & time == "day") %>% pull(perc_mm) %>% shapiro.test(.)
```

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

```r
counts_perc_long %>% filter(layer =="gcl" & time == "night") %>% pull(perc_mm) %>% shapiro.test(.)
```

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

```r
counts_perc_long %>% filter(layer =="gcl" & light == "light") %>% pull(perc_mm) %>% shapiro.test(.)
```

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

```r
counts_perc_long %>% filter(layer =="gcl" & light == "dark") %>% pull(perc_mm) %>% shapiro.test(.)
```

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

```r
counts_perc_long %>% filter(layer =="inl" & light == "dark") %>% pull(perc_mm) %>% shapiro.test(.)
```

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

```r
counts_perc_long %>% filter(layer =="inl" & light == "light") %>% pull(perc_mm) %>% shapiro.test(.)
```

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

```r
#checking homogenous variance in each combo
car::leveneTest(counts$gcl_perc_mm, group=counts$light)
```

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

```r
car::leveneTest(counts$inl_perc_mm, group=counts$light)
```

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

```r
car::leveneTest(counts$gcl_perc_mm, group=counts$time)
```

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

```r
car::leveneTest(counts$inl_perc_mm, group=counts$time)
```

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

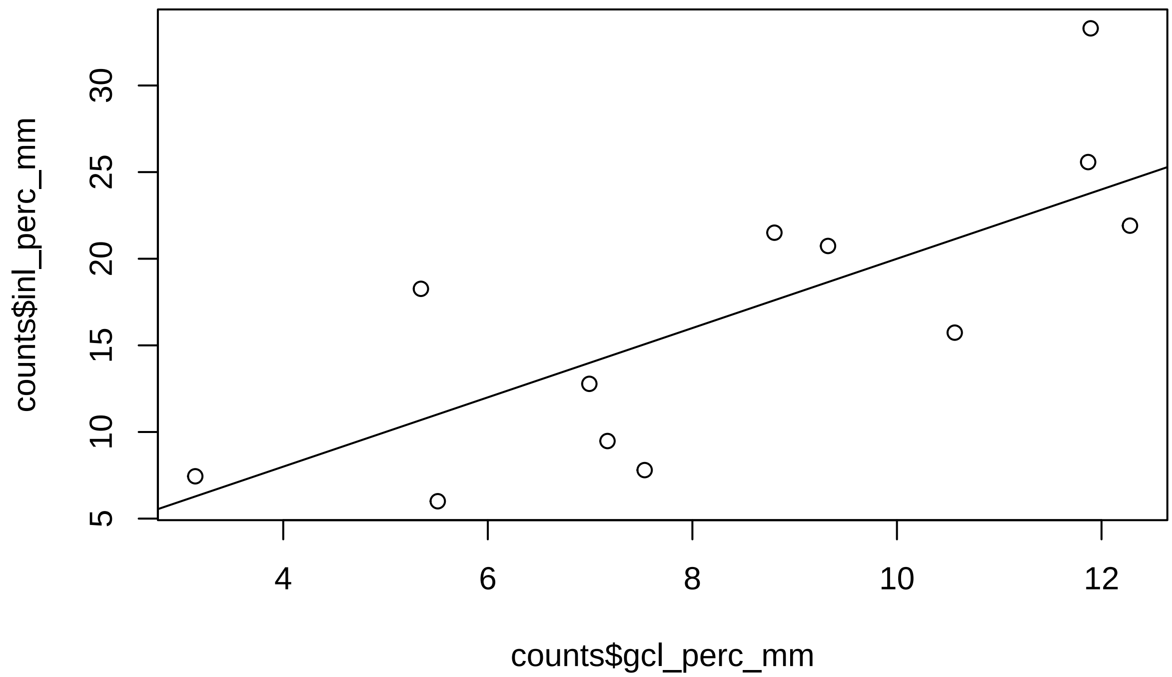
```r
#checking for multicollinearity
#low values are good
inl_perc_lm <- lm(inl_perc_mm ~ light * time, counts)
car::vif(inl_perc_lm)
```

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

```r
gcl_perc_lm <- lm(gcl_perc_mm ~ light * time, counts)
car::vif(gcl_perc_lm)
```

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

```r
#linearity of dep variables
plot(counts$gcl_perc_mm, counts$inl_perc_mm)
abline(0,2)
```




```
#good to go on manova!

#manova
man_perc_test <- manova(cbind(gcl_perc_mm, inl_perc_mm) ~ light * time, data=counts)
summary(man_perc_test)
```

```
##             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(man_perc_test)
```

```
##  Response gcl_perc_mm :
##             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_mm :
##             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
```

```r
#light:time interaction is just about significant overall and for GCL and INL
#look at comparisons through univariate models for each dep variable
anova(gcl_perc_lm)
```

```
## Analysis of Variance Table
##
## Response: gcl_perc_mm
##            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
```

```r
summary(gcl_perc_lm)
```

```
##
## Call:
## lm(formula = gcl_perc_mm ~ light * time, data = counts)
##
## 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
```

```r
anova(inl_perc_lm)
```

```
## Analysis of Variance Table
##
## Response: inl_perc_mm
##            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
```

```r
summary(inl_perc_lm)
```

```
##
## Call:
## lm(formula = inl_perc_mm ~ light * time, data = counts)
##
## 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
```

```r
emmeans::emmeans(gcl_perc_lm, pairwise ~ light * time, adjust="none")$contrasts
```

```
##  contrast                estimate   SE df t.ratio p.value
##  light day - dark day       -6.18 1.54  8  -4.025  0.0038
##  light day - light night    -4.06 1.54  8  -2.643  0.0296
##  light day - dark night     -1.65 1.54  8  -1.075  0.3136
##  dark day - light night      2.12 1.54  8   1.382  0.2043
##  dark day - dark night       4.53 1.54  8   2.950  0.0184
##  light night - dark night    2.41 1.54  8   1.567  0.1557
```

```r
emmeans::emmeans(inl_perc_lm, pairwise ~ light * time, adjust="none")$contrasts
```

```
##  contrast                estimate   SE df t.ratio p.value
##  light day - dark day      -16.57 5.29  8  -3.131  0.0140
##  light day - light night   -11.52 5.29  8  -2.176  0.0612
##  light day - dark night    -10.43 5.29  8  -1.972  0.0841
##  dark day - light night      5.05 5.29  8   0.954  0.3678
##  dark day - dark night       6.13 5.29  8   1.159  0.2799
##  light night - dark night    1.08 5.29  8   0.204  0.8431
```

```r
gcl_perc_plot <- ggplot() +
  geom_bar(data=perc_sumstats, aes(x=light, y=mean_gcl_perc, fill=light), stat="identity") +
  geom_errorbar(data=perc_sumstats, aes(x=light,
                              ymin=mean_gcl_perc-sd_gcl_perc,
                              ymax=mean_gcl_perc+sd_gcl_perc), width=0.2) +
  geom_point(data=counts, aes(x=light, y=gcl_perc_mm)) +
  facet_wrap(~time) +
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
```

```r
        panel.spacing = unit(0, "pt"),
        legend.position="none") +
  labs(y=expression(paste("% ", beta, "-endorphin+/ChAT+ cells in GCL /" ~mm^2))) +
  scale_fill_manual(values=c("yellow", "gray70")) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.4)),
                     labels=function(x) paste0(x,"%"))

inl_perc_plot <- ggplot() +
  geom_bar(data=perc_sumstats, aes(x=light, y=mean_inl_perc, fill=light), stat="identity") +
  geom_errorbar(data=perc_sumstats, aes(x=light,
                              ymin=mean_inl_perc-sd_inl_perc,
                              ymax=mean_inl_perc+sd_inl_perc), width=0.2) +
  geom_point(data=counts, aes(x=light, y=inl_perc_mm)) +
  facet_wrap(~time) +
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.ticks.x = element_blank(),
        panel.spacing = unit(0, "pt"),
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        legend.position="none") +
  labs(y=expression(paste("% ", beta, "-endorphin+/ChAT+ cells in INL /" ~mm^2))) +
  scale_fill_manual(values=c("yellow", "gray70")) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.3)),
                     labels=function(x) paste0(x,"%"))

#,
#        plot.background = element_rect(fill = "transparent",colour = NA),
#        panel.background = element_rect(fill = "transparent",colour = NA)

both_perc_plots <- ggarrange(gcl_perc_plot, inl_perc_plot, nrow=2)

ggsave("../figures/both_perc_plots.png", plot=both_perc_plots, height=7, width=6, bg = "transparent")

ctrl_counts <- counts %>% filter(light == "light" & time == "day")

#between var
sum(3*(mean(counts$gcl_perc_mm - mean(ctrl_counts$gcl_perc_mm)))^2)
```

```
## [1] 26.5387
```

```r
#within.var
anova(gcl_perc_lm)["Residuals", "Mean Sq"]
```

```
## [1] 3.541329
```

```r
power.anova.test(groups=4, n=3, between.var=26.5387,
                 within.var=3.541329, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 3
##     between.var = 26.5387
```

```
##      within.var = 3.541329
##       sig.level = 0.05
##           power = 0.9998598
##
## NOTE: n is number in each group
```

```r
#between var
sum(3*(mean(counts$inl_perc_mm - mean(ctrl_counts$inl_perc_mm)))^2)
```

```
## [1] 278.1516
```

```r
#within.var
anova(inl_perc_lm)["Residuals", "Mean Sq"]
```

```
## [1] 41.99559
```

```r
power.anova.test(groups=4, n=3, between.var=278.1516,
                 within.var=41.99559, sig.level=0.05)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 3
##     between.var = 278.1516
##      within.var = 41.99559
##       sig.level = 0.05
##           power = 0.9995018
##
## NOTE: n is number in each group
```

```r
summary_table <- counts %>% group_by(light, time) %>%
  summarise(n = n(),
            "Mean # ChAT+ counted in INL" = mean(INL_chat),
            "Total # ChAT+ counted in INL" = sum(INL_chat),
            "Mean # B-endo+ counted in INL" = mean(INL_bendo),
            "Total # B-endo+ counted in INL" = sum(INL_bendo),
            "Mean ChAT+ cells in INL/mm2" = mean(INL_chat_mm),
            "Range ChAT+ cells in INL/mm2" = paste(round(min(INL_chat_mm), 0),
                                                   round(max(INL_chat_mm), 0),
                                                   sep=" - "),
            "Mean B-endo+ cells in INL/mm2" = mean(INL_bendo_mm),
            "Range B-endo+ cells in INL/mm2" = paste(round(min(INL_bendo_mm), 0),
                                                     round(max(INL_bendo_mm), 0),
                                                     sep=" - "),
            "Mean % B-endo+/ChAT+ in INL/mm2" = mean(inl_perc_mm),
            "Range % B-endo+/ChAT+ in INL/mm2" = paste(round(min(inl_perc_mm), 0),
                                                       round(max(inl_perc_mm), 0),
                                                       sep=" - "),
            "Mean # ChAT+ counted in GCL" = mean(GCL_chat),
            "Total # ChAT+ counted in GCL" = sum(GCL_chat),
            "Mean # B-endo+ counted in GCL" = mean(GCL_bendo),
            "Total # B-endo+ counted in GCL" = sum(GCL_bendo),
            "Mean ChAT+ cells in GCL/mm2" = mean(GCL_chat_mm),
            "Range ChAT+ cells in GCL/mm2" = paste(round(min(GCL_chat_mm), 0),
                                                  round(max(GCL_chat_mm), 0),
                                                  sep=" - "),
```

```
            "Mean B-endo+ cells in GCL/mm2" = mean(GCL_bendo_mm),
            "Range B-endo+ cells in GCL/mm2" = paste(round(min(GCL_bendo_mm), 0),
                                                     round(max(GCL_bendo_mm), 0),
                                                  sep=" - "),
            "Mean % B-endo+/ChAT+ in GCL/mm2" = mean(gcl_perc_mm),
            "Range % B-endo+/ChAT+ in GCL/mm2" = paste(round(min(gcl_perc_mm), 0),
                                                       round(max(gcl_perc_mm), 0),
                                                       sep=" - ")) %>%
  mutate(across(where(is.numeric), round, 0))
```

```
## `summarise()` has grouped output by 'light'. You can override using the `.groups` argument.
```

```
summary_table
```

```
## # A tibble: 4 x 23
## # Groups:   light [2]
##   light time      n `Mean # ChAT+ count~ `Total # ChAT+ cou~ `Mean # B-endo+ co~
##   <fct> <fct> <dbl>                <dbl>               <dbl>               <dbl>
## 1 light day       3                 3611               10834                 256
## 2 light night     3                 4727               14180                 927
## 3 dark  day       3                 4877               14630                1139
## 4 dark  night     3                 4701               14103                 858
## # ... with 17 more variables: Total # B-endo+ counted in INL <dbl>,
## #   Mean ChAT+ cells in INL/mm2 <dbl>, Range ChAT+ cells in INL/mm2 <chr>,
## #   Mean B-endo+ cells in INL/mm2 <dbl>, Range B-endo+ cells in INL/mm2 <chr>,
## #   Mean % B-endo+/ChAT+ in INL/mm2 <dbl>,
## #   Range % B-endo+/ChAT+ in INL/mm2 <chr>, Mean # ChAT+ counted in GCL <dbl>,
## #   Total # ChAT+ counted in GCL <dbl>, Mean # B-endo+ counted in GCL <dbl>,
## #   Total # B-endo+ counted in GCL <dbl>, ...
```

```
#write_xlsx(summary_table, "../data/bendo_summary_table_temp.xlsx")
```

OLD

```
#data setup for previous anova analysis
#also used in manova 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, 2)
```

```
## # A tibble: 2 x 13
##   mouse light time  sex     age GCL_chat_mm GCL_bendo_mm INL_chat_mm
##   <fct> <fct> <fct> <chr> <dbl>       <dbl>        <dbl>       <dbl>
```

```
## 1 11    light day   F        95        848.         26.6       1071.
## 2 11    light day   F        95        848.         26.6       1071.
## # ... with 5 more variables: INL_bendo_mm <dbl>, inl_perc_mm <dbl>,
## #   gcl_perc_mm <dbl>, layer <fct>, perc <dbl>
```

Checking Normality

```
#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
```

Checking MANOVA Assumptions

```
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


#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
```

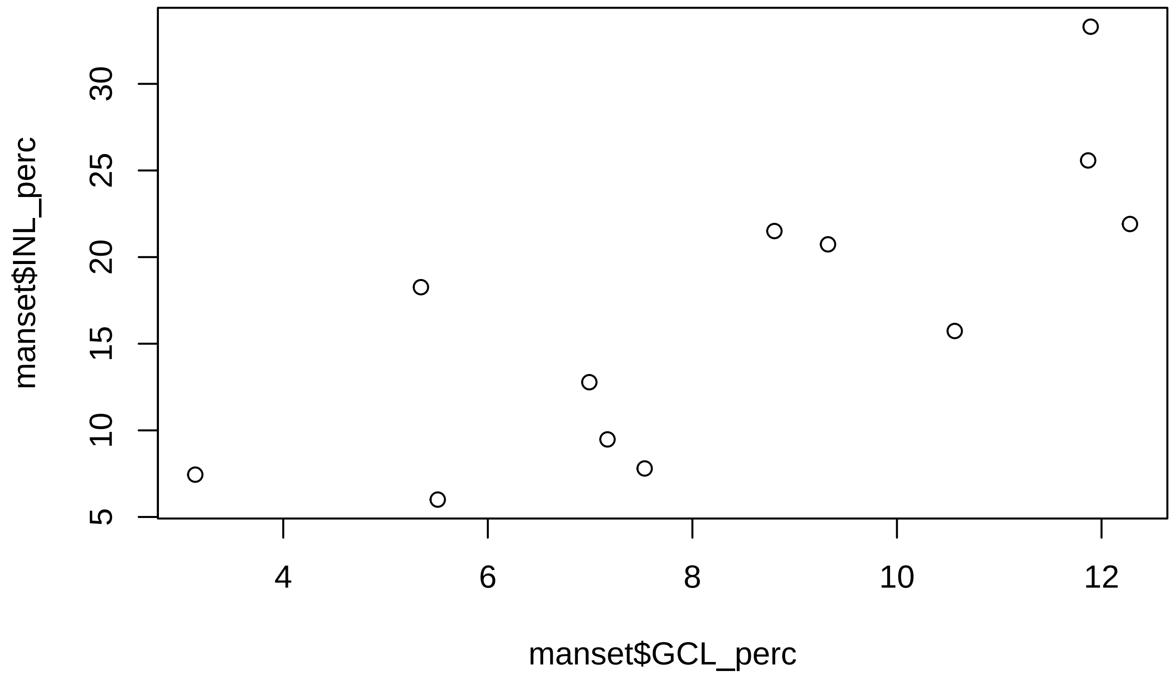
```
#checking for multicollinearity
#low values are good
inllm <- lm(INL_perc ~ light * time, manset)
car::vif(inllm)
```

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

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

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

```r
#linearity of dep variables
plot(manset$GCL_perc, manset$INL_perc)
```




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

MANOVA

```r
#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
```

```r
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
```

```
emmeans::emmeans(gcllm, pairwise ~ light * time, adjust="none")$contrasts
```

```
##  contrast                 estimate   SE df t.ratio p.value
##  light day - dark day        -6.18 1.54  8  -4.025  0.0038
##  light day - light night     -4.06 1.54  8  -2.643  0.0296
##  light day - dark night      -1.65 1.54  8  -1.075  0.3136
##  dark day - light night       2.12 1.54  8   1.382  0.2043
##  dark day - dark night        4.53 1.54  8   2.950  0.0184
##  light night - dark night     2.41 1.54  8   1.567  0.1557
```

```
emmeans::emmeans(inllm, pairwise ~ light * time, adjust="none")$contrasts
```

```
##  contrast                 estimate   SE df t.ratio p.value
##  light day - dark day       -16.57 5.29  8  -3.131  0.0140
##  light day - light night    -11.52 5.29  8  -2.176  0.0612
##  light day - dark night     -10.43 5.29  8  -1.972  0.0841
##  dark day - light night       5.05 5.29  8   0.954  0.3678
##  dark day - dark night        6.13 5.29  8   1.159  0.2799
##  light night - dark night     1.08 5.29  8   0.204  0.8431
```

```
emmeans::emmeans(gcllm, pairwise ~ light | time, adjust="none")$contrasts
```

```
## time = day:
##  contrast     estimate   SE df t.ratio p.value
##  light - dark    -6.18 1.54  8  -4.025  0.0038
##
## time = night:
##  contrast     estimate   SE df t.ratio p.value
##  light - dark     2.41 1.54  8   1.567  0.1557
```

```
emmeans::emmeans(gcllm, pairwise ~ time | light, adjust="none")$contrasts
```

```
## light = light:
##  contrast    estimate   SE df t.ratio p.value
##  day - night    -4.06 1.54  8  -2.643  0.0296
##
## light = dark:
##  contrast    estimate   SE df t.ratio p.value
##  day - night     4.53 1.54  8   2.950  0.0184
```

```
emmeans::emmeans(inllm, pairwise ~ light | time, adjust="none")$contrasts
```

```
## time = day:
##  contrast     estimate   SE df t.ratio p.value
##  light - dark   -16.57 5.29  8  -3.131  0.0140
##
## time = night:
##  contrast     estimate   SE df t.ratio p.value
##  light - dark     1.08 5.29  8   0.204  0.8431
```

```
emmeans::emmeans(inllm, pairwise ~ time | light, adjust="none")$contrasts
```

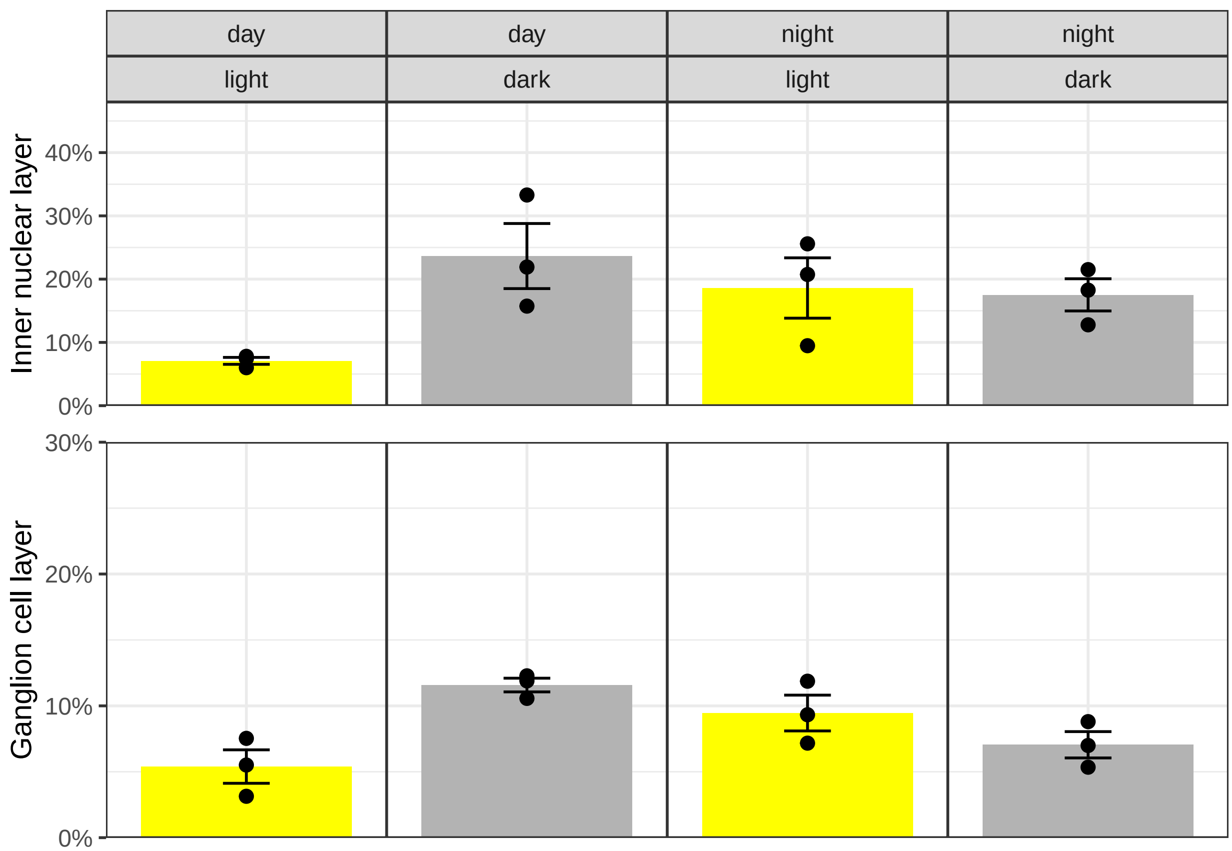
```
## light = light:
```

```
##   contrast    estimate   SE df t.ratio p.value
##   day - night   -11.52 5.29  8  -2.176  0.0612
##
## light = dark:
##   contrast    estimate   SE df t.ratio p.value
##   day - night     6.13 5.29  8   1.159  0.2799
```

Plotting

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


plotting_manset$group <- c("1", "1", "1", "2", "2", "2", "3", "3", "3", "4", "4", "4")


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

inlpointplot <- inlpointmanset %>% ggplot(aes(x=group, y=INL_perc, fill=light)) +
  stat_summary(fun = 'mean', geom="bar") +
  stat_summary(fun.data = mean_se, geom = "errorbar", width=.2) +
  geom_point(size =2) +
  facet_wrap(time ~ factor(light, c("light", "dark")),
             scales="free_x",
             nrow=1,
             as.table=FALSE) +
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x = element_blank(),
        legend.position = "none",
        panel.spacing = unit(0,"pt"),
        plot.background = element_rect(fill = "transparent",colour = NA),
        panel.background = element_rect(fill = "transparent",colour = NA)) +
  labs(y="Inner nuclear layer") +
  scale_y_continuous(labels=function(x) paste0(x,"%"),
                     limits=c(0,40),
                     expand = expansion(mult = c(0, 0.2))) +
  scale_fill_manual(values=c("yellow", "gray70"))


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

gclpointplot <- gclpointmanset %>% ggplot(aes(x=group, y=GCL_perc, fill=light)) +
  stat_summary(fun = 'mean', geom="bar") +
  stat_summary(fun.data = mean_se, geom = "errorbar", width=.2) +
  geom_point(size=2) +
  facet_wrap(time ~ factor(light, c("light", "dark")),
             scales="free_x",
```

```
            nrow=1,
            as.table=FALSE) +
  theme_bw() %+replace%
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x = element_blank(),
        legend.position = "none",
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        panel.spacing = unit(0, "pt"),
        plot.background = element_rect(fill = "transparent",colour = NA),
        panel.background = element_rect(fill = "transparent",colour = NA)) +
  labs(y="Ganglion cell layer") +
  scale_y_continuous(labels=function(x) paste0(x,"%"),
                     limits=c(0,30),
                     expand = expansion(mult = c(0, 0))) +
  scale_fill_manual(values=c("yellow", "gray70"))

bothpointplots <- ggarrange(inlpointplot, gclpointplot, nrow=2)

bothpointplots
```




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

Old ANOVA analysis

```
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`
```

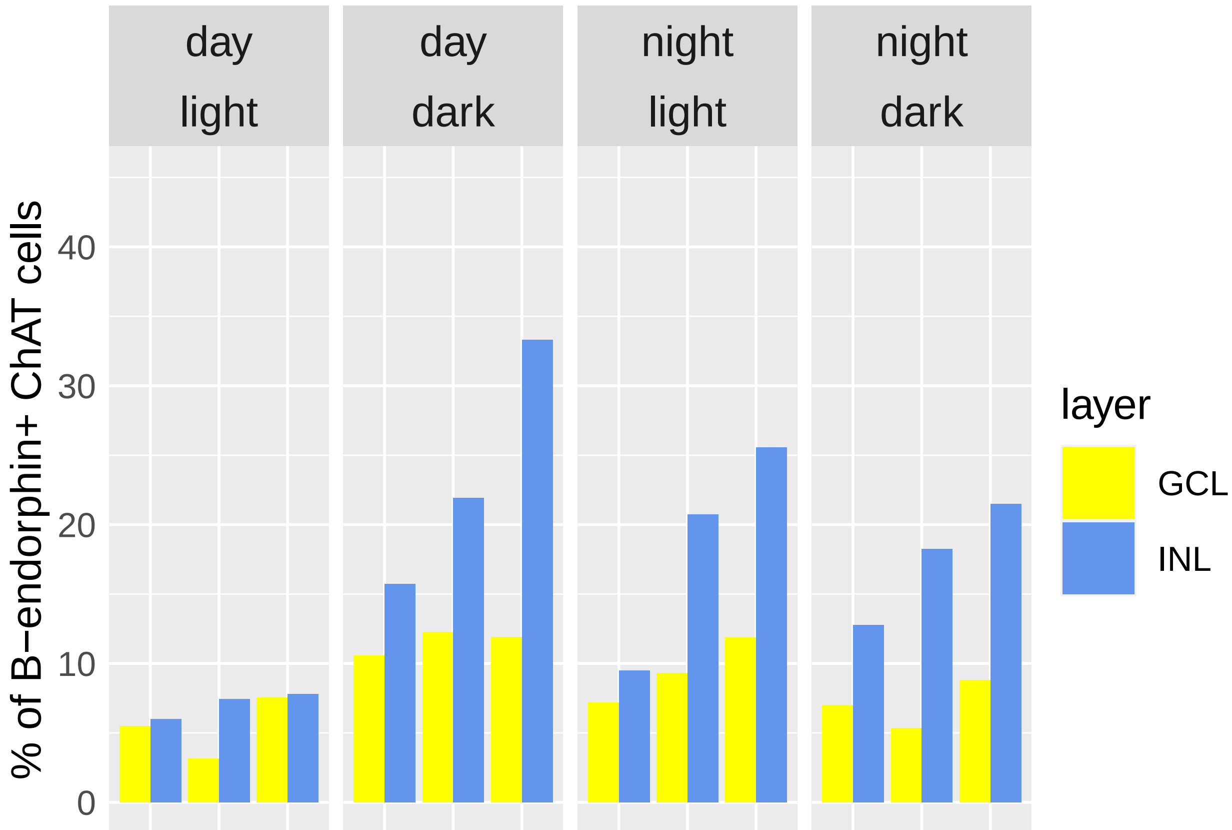
```
##                           diff       lwr       upr     p adj
## night:GCL-day:GCL   -0.2357322 -8.117562  7.646097 0.9997654
## day:INL-day:GCL      6.8778753 -1.003954 14.759705 0.0986522
## night:INL-day:GCL    9.5698762  1.688047 17.451706 0.0149110
## day:INL-night:GCL    7.1136075 -0.768222 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)
```