

# Larval performance and substrate pH

Stijn Schreven

1 June 2021

## Contents

<b>Load packages</b>	<b>1</b>
<b>Input files</b>	<b>2</b>
<b>1. Prepare data</b>	<b>2</b>
1.2. Plot presets . . . . .	3
<b>2. Larval performance</b>	<b>3</b>
2.0. Model validation . . . . .	3
2.1. Survival . . . . .	4
2.2. Prepupae . . . . .	5
2.3. Individual larval weight . . . . .	6
<b>3. Substrate pH and moisture content on day 15</b>	<b>7</b>
3.1. pH . . . . .	7
3.2. Moisture content . . . . .	8
<b>4. Plots</b>	<b>9</b>
4.1. Collect estimates . . . . .	10
4.2. Plots . . . . .	10
<b>5. pH of starting diets</b>	<b>15</b>
5.1. Prepare data . . . . .	16
5.2. LMM regression . . . . .	16

## Load packages

```
library(nlme)
library(lme4)
library(plyr)
library(emmeans)
library(sciplot)
library(ggplot2)
library(ggpubr)
library(viridis)
```

## Input files

```
# performance and substrate pH on day 15
harvest <- read.delim("./input_data/Schreven_harvest_data.txt", header = T)

# substrate pH of starting diets
dietph <- read.delim("./input_data/Schreven_pHdiets_fresh.txt", header=T)
```

## 1. Prepare data

Total larval biomass is excluded from analysis, because this is affected by timepoint sampling.

```
# factors
harvest$Density <- as.factor(harvest$Density)
harvest$Day <- as.factor(harvest$Day)
harvest$Diet <- factor(harvest$Diet, levels(harvest$Diet)[c(1,3,2)])

# rescale units
harvest$SurvivalRate <- 100 * harvest$SurvivalRate # in %
harvest$pPrepupae <- 100 * harvest$pPrepupae # in %
harvest$pDMRes <- 100 * harvest$pDMRes # DM content of residue (% FM)
harvest$pMoist <- 100 - harvest$pDMRes # moisture content of residue (% FM)

# subset to treatments with larvae
harvL <- subset(harvest, Density != 0)
harvL$Density <- droplevels(harvL$Density)

# melt dataframe
harv.m <- reshape2::melt(harvest)

## Using Diet, Density, Day as id variables

# summarise
harv.sum <- ddply(harv.m, ~ Diet + Density + variable, summarise,
  mean = mean(value, na.rm = T), median = median(value, na.rm = T),
  sd = sd(value, na.rm = T), se = se(value, na.rm = T))
```

## 1.2. Plot presets

```
labs_perf <- as_labeller(c(
  SurvivalRate = "survival rate (%)",
  pPrepupae = "prepupae (%)",
  pHres = "pH",
  dmLarvInd = "ind. larval weight (g DM)"))

theme_perf <- theme_classic() + theme(
  plot.margin = margin(0, .7, .5, .5, "cm"),
  panel.grid.major.y = element_line(colour = "grey80"),
  panel.spacing = unit(.5, "lines"),
  panel.border = element_rect(color = "black", fill = NA, size = .5),
  strip.background = element_blank(),
  strip.placement = "outside",
  text = element_text(size = 20),
  axis.title = element_text(size = 16),
  legend.title = element_text(size = 16),
  axis.text.x = element_text(hjust = .5, vjust = 1))

pd <- position_dodge(width=5)
```

## 2. Larval performance

Linear regression models with mixed model selection to assess inclusion of:

- batch effect (random intercept for Day);
- variance structure for treatments.

We used gls/lme models. If residuals did not meet assumptions of normality and homoskedasticity, we compared LM with GLM (Gamma distribution). Only for individual larval weight the GLM was a better fit (lower AIC) than the LM, and therefore used instead. For the other larval performance parameters, GLM was a worse fit and not shown below.

### 2.0. Model validation

The R code chunk below was used to validate models by creating QQ plots and plotting residuals against fitted values and independent variables. It is not evaluated in this R markdown file, but can be adapted to validate each model in the following R code chunks.

```
# mod = model
# data = dataframe with original data

# GLS/LMM model validation

## create dataframe with data and model residuals
res.df <- data.frame(data, res = residuals(mod, type = "normalized"), fit = fitted(mod))

## check normality of residuals
```

```

hist(res.df$res)
ggplot(res.df, aes(sample = res)) + stat_qq() + stat_qq_line()

## if random term in model: check normality of random effects
qqnorm(mod, ~ ranef())

## check homoskedasticity of residuals
plot(mod)
plot(res ~ fit, res.df); abline(0,0)
plot(res ~ Diet, res.df, las = 2); abline(0,0)
plot(res ~ Density, res.df); abline(0,0)
plot(res ~ Day, res.df); abline(0,0)

# GLM model validation

## create dataframe with data and model residuals
res.df <- data.frame(data, res = residuals(mod, type = "pearson"), fit = fitted(mod))

plot(mod)

## check normality
hist(res.df$res)
ggplot(res.df, aes(sample = res)) + stat_qq() + stat_qq_line()

## check homoskedasticity
plot(res ~ fit, res.df); abline(0,0)
plot(res ~ Diet, res.df, las = 2); abline(0,0)
plot(res ~ Density, res.df); abline(0,0)
plot(res ~ Day, res.df); abline(0,0)

```

## 2.1. Survival

```

# GLS/LMM model selection
S.m0 <- gls(SurvivalRate ~ Diet * Density, data = harvL, method = "REML")
S.m1 <- lme(SurvivalRate ~ Diet * Density, data = harvL, method = "REML", random = ~1|Day)
S.m2 <- update(S.m0, weights = varIdent(form = ~1|Diet))
S.m3 <- update(S.m0, weights = varIdent(form = ~1|Density))
S.m4 <- update(S.m0, weights = varIdent(form = ~1|Diet*Density))
S.m5 <- update(S.m1, weights = varIdent(form = ~1|Diet))
S.m6 <- update(S.m1, weights = varIdent(form = ~1|Density))
S.m7 <- update(S.m1, weights = varIdent(form = ~1|Diet*Density))
AIC(S.m0, S.m1, S.m2, S.m3, S.m4, S.m5, S.m6, S.m7)

##      df      AIC
## S.m0 10 235.3481
## S.m1 11 237.3481
## S.m2 12 223.0588
## S.m3 12 237.2338
## S.m4 18 224.9100
## S.m5 13 225.0588

```

```
## S.m6 13 239.2338
## S.m7 19 226.9100
```

```
# model S.m2 is best: variance structure for Diet, no random term.
```

```
# model output
anova(S.m2)
```

```
## Denom. DF: 27
##          numDF    F-value p-value
## (Intercept)      1 10559.510 <.0001
## Diet              2   25.710 <.0001
## Density           2    1.276 0.2955
## Diet:Density      4    1.864 0.1457
```

```
CLD(emmeans(S.m2, ~ Diet * Density, method = "tukey"), Letters = letters)
```

```
## Diet Density emmean    SE    df lower.CL upper.CL .group
## CF    200      60.4 7.50 9.01     43.4     77.3   ab
## CF     50      65.9 7.50 9.01     48.9     82.8  abc
## CF    100      66.5 7.50 9.01     49.5     83.4  abc
## CM    100      69.2 4.65 9.01     58.7     79.8    a
## CM     50      82.3 4.65 9.01     71.8     92.8  abc
## CM    200      84.2 4.65 9.01     73.6     94.7  abc
## CS     50      88.4 1.61 9.00     84.8     92.1   bc
## CS    100      91.8 1.61 9.00     88.1     95.4    c
## CS    200      92.4 1.61 9.00     88.7     96.0    c
##
## Degrees-of-freedom method: satterthwaite
## Confidence level used: 0.95
## P value adjustment: tukey method for comparing a family of 9 estimates
## significance level used: alpha = 0.05
```

## 2.2. Prepupae

```
# GLS/LMM model selection
P.m0 <- gls(pPrepupae ~ Diet * Density, data = harvL, method = "REML")
P.m1 <- lme(pPrepupae ~ Diet * Density, data = harvL, method = "REML", random = ~1|Day)
P.m2 <- update(P.m0, weights = varIdent(form = ~1|Diet))
P.m3 <- update(P.m0, weights = varIdent(form = ~1|Density))
# P.m4 <- update(P.m0, weights = varIdent(form = ~1|Diet*Density))
## P.m4: error false convergence - issue with zeroes
P.m5 <- update(P.m1, weights = varIdent(form = ~1|Diet))
P.m6 <- update(P.m1, weights = varIdent(form = ~1|Density))
AIC(P.m0, P.m1, P.m2, P.m3, P.m5, P.m6)
```

```
##      df      AIC
## P.m0 10 225.2573
## P.m1 11 227.2573
## P.m2 12 215.5747
```

```
## P.m3 12 228.9144
## P.m5 13 217.2817
## P.m6 13 230.9144
```

```
# model P.m2 is best: variance structure for Diet, no random term.
```

```
# model output
anova(P.m2)
```

```
## Denom. DF: 27
##          numDF  F-value p-value
## (Intercept)      1 106.2460 <.0001
## Diet            2 449.7745 <.0001
## Density         2   0.0456 0.9555
## Diet:Density     4   1.2277 0.3225
```

```
CLD(emmeans(P.m2, ~ Diet * Density, method = "tukey"), Letters = letters)
```

```
## Diet Density emmean SE df lower.CL upper.CL .group
## CS 200      0.00 1.39 9.00   -3.14    3.14 a
## CS 100      1.30 1.39 9.00   -1.85    4.44 a
## CS 50       2.03 1.39 9.00   -1.12    5.17 a
## CM 50       8.53 5.31 7.10   -3.98   21.04 a
## CM 100      9.06 5.31 7.10   -3.46   21.57 a
## CM 200     15.80 5.31 7.10    3.29   28.32 a
## CF 50      85.97 5.03 7.06   74.09   97.84 b
## CF 100     90.62 5.03 7.06   78.75  102.49 b
## CF 200     97.67 5.03 7.06   85.79  109.54 b
##
## Degrees-of-freedom method: satterthwaite
## Confidence level used: 0.95
## P value adjustment: tukey method for comparing a family of 9 estimates
## significance level used: alpha = 0.05
```

## 2.3. Individual larval weight

```
# GLM regression, compared to LM regression
I.lm <- glm(dmLarvInd ~ Diet * Density, harvL, family = gaussian)
I.glm <- glm(dmLarvInd ~ Diet * Density, harvL, family = Gamma)
AIC(I.lm, I.glm)
```

```
##      df      AIC
## I.lm 10 -231.7398
## I.glm 10 -232.0516
```

```
# GLM is better than LM.
```

```
# model output
car::Anova(I.glm)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: dmLarvInd
##           LR Chisq Df Pr(>Chisq)
## Diet           59.622  2  1.130e-13 ***
## Density         59.131  2  1.445e-13 ***
## Diet:Density    73.429  4  4.281e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

CLD(emmeans(ref_grid(lm, transform = "response"), ~ Diet * Density,
             method = "tukey"), Letters = letters)

## Diet Density response      SE df asymp.LCL asymp.UCL .group
## CM    200      0.0245 0.00169 Inf    0.0212    0.0278    a
## CM    100      0.0440 0.00303 Inf    0.0381    0.0499    b
## CF    200      0.0549 0.00378 Inf    0.0475    0.0623   bc
## CS    200      0.0617 0.00425 Inf    0.0534    0.0700   cd
## CS     50      0.0666 0.00459 Inf    0.0576    0.0756   cd
## CM     50      0.0704 0.00485 Inf    0.0609    0.0799   cd
## CS    100      0.0722 0.00498 Inf    0.0625    0.0820   cd
## CF    100      0.0728 0.00501 Inf    0.0629    0.0826   cd
## CF     50      0.0813 0.00560 Inf    0.0704    0.0923    d
##
## Confidence level used: 0.95
## P value adjustment: tukey method for comparing a family of 9 estimates
## significance level used: alpha = 0.05
```

### 3. Substrate pH and moisture content on day 15

Linear regression models with mixed model selection to assess inclusion of:

- batch effect (random intercept for Day);
- variance structure for treatments.

#### 3.1. pH

```
# GLS/LMM model selection
ph.m0 <- gls(pHres ~ Diet * Density, data = harvest, method = "REML")
ph.m1 <- lme(pHres ~ Diet * Density, data = harvest, method = "REML", random = ~1|Day)
ph.m2 <- update(ph.m0, weights = varIdent(form = ~1|Diet))
ph.m3 <- update(ph.m0, weights = varIdent(form = ~1|Density))
ph.m4 <- update(ph.m0, weights = varIdent(form = ~1|Diet*Density))
ph.m5 <- update(ph.m1, weights = varIdent(form = ~1|Diet))
ph.m6 <- update(ph.m1, weights = varIdent(form = ~1|Density))
ph.m7 <- update(ph.m1, weights = varIdent(form = ~1|Diet*Density))
AIC(ph.m0, ph.m1, ph.m2, ph.m3, ph.m4, ph.m5, ph.m6, ph.m7)

##           df           AIC
```

```
## ph.m0 13 90.05863
## ph.m1 14 92.05735
## ph.m2 15 75.59863
## ph.m3 16 89.59654
## ph.m4 24 78.23093
## ph.m5 16 75.31010
## ph.m6 17 90.76028
## ph.m7 25 79.95471
```

```
# model ph.m5 is best: random intercept for batch and variance structure for Diet.
```

```
# model output
```

```
anova(ph.m5)
```

```
##           numDF denDF   F-value p-value
## (Intercept)      1    35 14229.053 <.0001
## Diet             2    35   177.430 <.0001
## Density          3    35    14.733 <.0001
## Diet:Density     6    35    12.255 <.0001
```

```
CLD(emmeans(ph.m5, ~ Diet * Density, method = "tukey"), Letters = letters)
```

```
## Diet Density emmean      SE df lower.CL upper.CL .group
## CS    50      5.17 0.2761  1     1.66     8.68    a
## CS   100      5.54 0.2761  1     2.04     9.05   ab
## CS    0      6.55 0.2761  1     3.04    10.06   bc
## CF    0      7.20 0.2973  1     3.42    10.97   cd
## CF   50      7.49 0.2973  1     3.71    11.26   cd
## CF  100      8.04 0.2973  1     4.27    11.82   def
## CF  200      8.22 0.2973  1     4.45    12.00   def
## CS  200      8.44 0.2761  1     4.93    11.95   def
## CM   50      8.67 0.0922  1     7.50     9.84    e
## CM    0      8.79 0.0922  1     7.62     9.96   ef
## CM  100      8.89 0.0922  1     7.72    10.06   ef
## CM  200      9.07 0.0922  1     7.90    10.24    f
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
## P value adjustment: tukey method for comparing a family of 12 estimates
## significance level used: alpha = 0.05
```

### 3.2. Moisture content

```
# GLS/LMM model selection
```

```
moist.m0 <- gls(pMoist ~ Diet * Density, data = harvest, method = "REML")
```

```
moist.m1 <- lme(pMoist ~ Diet * Density, data = harvest, method = "REML",
  random = ~1|Day)
```

```
moist.m2 <- update(moist.m0, weights = varIdent(form = ~1|Diet))
```

```
moist.m3 <- update(moist.m0, weights = varIdent(form = ~1|Density))
```

```
moist.m4 <- update(moist.m0, weights = varIdent(form = ~1|Diet*Density))
```



```
moist.m5 <- update(moist.m1, weights = varIdent(form = ~1|Diet))
moist.m6 <- update(moist.m1, weights = varIdent(form = ~1|Density))
moist.m7 <- update(moist.m1, weights = varIdent(form = ~1|Diet*Density))
AIC(moist.m0, moist.m1, moist.m2, moist.m3, moist.m4, moist.m5, moist.m6, moist.m7)
```

```
##          df      AIC
## moist.m0 13 218.0341
## moist.m1 14 220.0341
## moist.m2 15 196.3069
## moist.m3 16 205.2984
## moist.m4 24 199.2265
## moist.m5 16 198.3069
## moist.m6 17 207.2982
## moist.m7 25 201.2265
```

```
# model m2 is best: variance structure for Diet, no random term.
```

```
# final model
anova(moist.m2)
```

```
## Denom. DF: 36
##          numDF    F-value p-value
## (Intercept)      1 149689.59 <.0001
## Diet              2   118.27 <.0001
## Density           3    42.99 <.0001
## Diet:Density      6     4.22 0.0026
```

```
CLD(emmeans(moist.m2, ~ Diet * Density, method = "tukey"), Letters = letters)
```

```
## Diet Density emmean    SE df lower.CL upper.CL .group
## CF      0      56.7 2.143 12    52.0    61.4    a
## CF     200      61.0 2.143 12    56.4    65.7    a
## CF     100      62.3 2.143 12    57.6    66.9    a
## CF      50      64.4 2.143 12    59.7    69.0   ab
## CM     100      71.4 0.977 12    69.3    73.6   bc
## CM      0      71.7 0.977 12    69.6    73.9   bc
## CM      50      72.7 0.977 12    70.6    74.8   bc
## CS      0      73.1 0.432 12    72.1    74.0    c
## CM     200      73.1 0.977 12    71.0    75.2  bcd
## CS      50      75.1 0.432 12    74.2    76.1   cd
## CS     100      76.9 0.432 12    75.9    77.8    d
## CS     200      80.2 0.432 12    79.2    81.1    e
##
```

```
## Degrees-of-freedom method: satterthwaite
## Confidence level used: 0.95
## P value adjustment: tukey method for comparing a family of 12 estimates
## significance level used: alpha = 0.05
```

## 4. Plots

Figure 2 in manuscript Chapter 3 in PhD thesis and Table 1 in submitted manuscript to *Applied and Environmental Microbiology*.

## 4.1. Collect estimates

```
# estimates per variable
S.emm <- CLD(emmeans(S.m2, ~Diet*Density, method = "tukey"), Letters = letters)
P.emm <- CLD(emmeans(P.m2, ~Diet*Density, method = "tukey"), Letters = letters)
I.emm <- CLD(emmeans(ref_grid(I.gm, transform = "response"), ~ Diet * Density,
                     method = "tukey"), Letters = letters)
colnames(I.emm) <- colnames(S.emm) # match colnames
ph.emm <- CLD(emmeans(ph.m5, ~Diet*Density, method = "tukey"), Letters = letters)
moist.emm <- CLD(emmeans(moist.m2, ~Diet*Density, method = "tukey"), Letters = letters)
S.emm$variable <- "SurvivalRate"
P.emm$variable <- "pPrepupae"
I.emm$variable <- "dmLarvInd"
ph.emm$variable <- "pHres"
moist.emm$variable <- "pMoist"

# combine estimates in one dataframe (rbind)
harv.emm <- rbind(S.emm, P.emm, I.emm, ph.emm, moist.emm)

# rename/ order factor levels
harv.emm$Diet <- revalue(harv.emm$Diet, c(
  "CF" = "chicken\nfeed", "CS" = "camelina",
  "CM" = "chicken\nmanure"))
harv.emm$Density <- factor(harv.emm$Density,
                          levels(harv.emm$Density)[c(4,1,2,3)])
levels(harv.emm$Density)

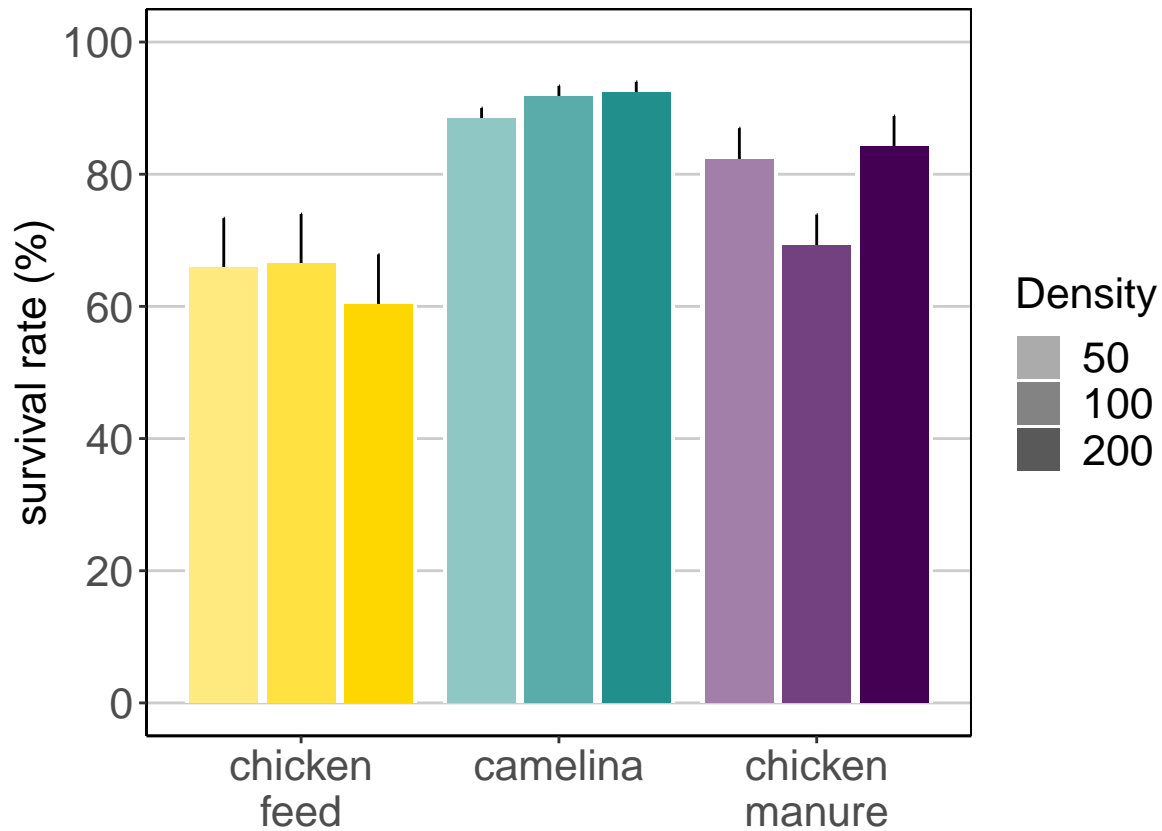
## [1] "0"   "50"  "100" "200"

write.table(harv.emm, "./tables/Summary_Performance_pH.txt", sep = "\t")
```

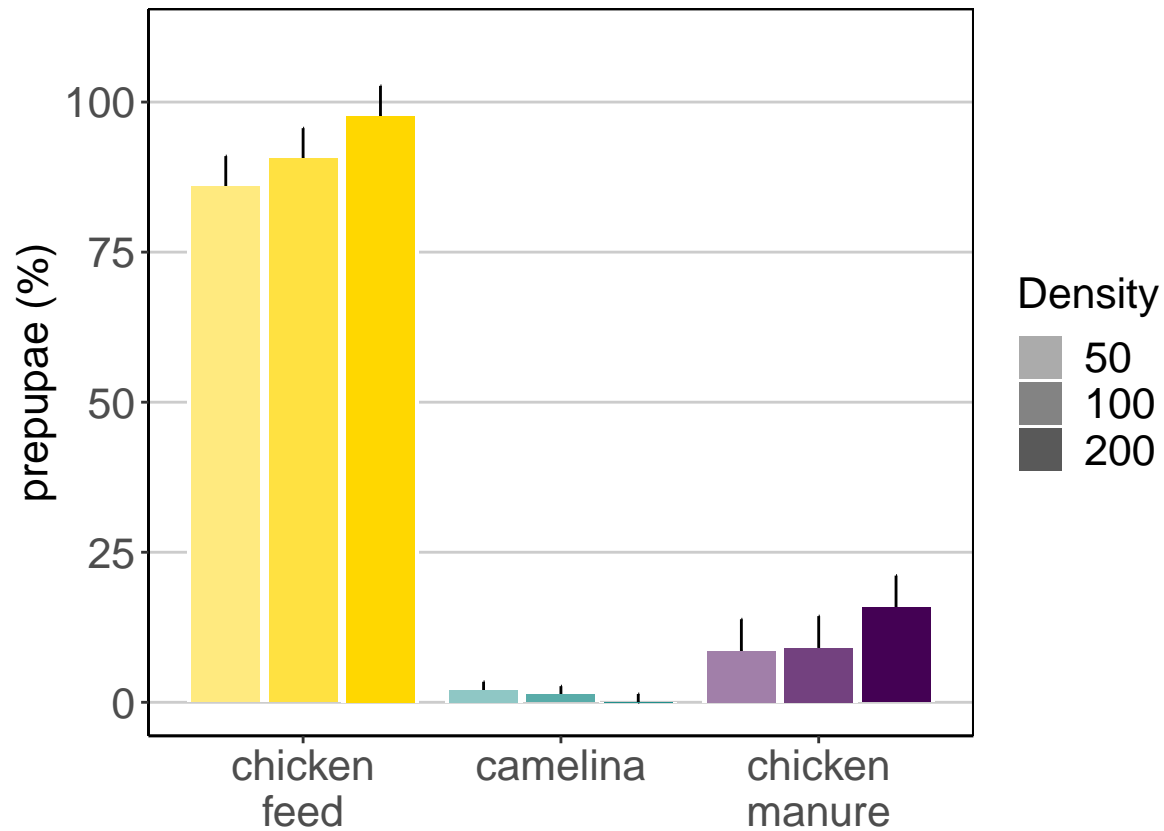
## 4.2. Plots

Errorbar plots based on EMM and SE from models.

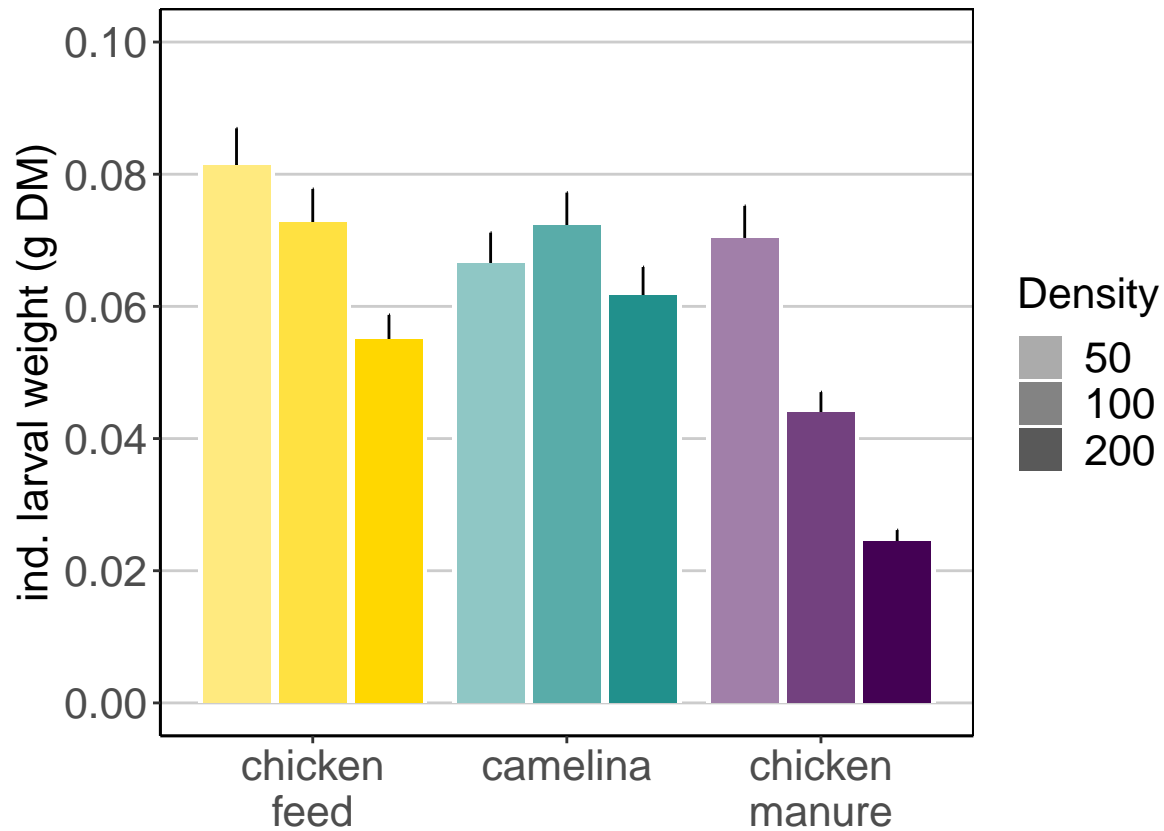
```
# separate plots, to set y axis limits and n.breaks (not possible in facet)
pSurv <- ggplot(subset(harv.emm, variable == "SurvivalRate"),
               aes(x=Diet, y=emmean, alpha = Density,
                   group=Density)) +
  geom_errorbar(aes(ymin=emmean, ymax=emmean+SE), alpha = 1,
               width=0, position= position_dodge(.9)) +
  geom_col(fill = "white", alpha = 1, position = position_dodge(.9)) +
  geom_col(aes(fill = Diet), width = .8, position = position_dodge(.9)) +
  labs(y="survival rate (%)", x = NULL) +
  scale_y_continuous(limits=c(0,100), n.breaks=6) +
  scale_fill_manual(values = c("gold", "#21908CFF", "#440154FF")) +
  scale_alpha_ordinal(range = c(0.5, 1)) +
  guides(fill = "none") +
  theme_perf
pSurv
```



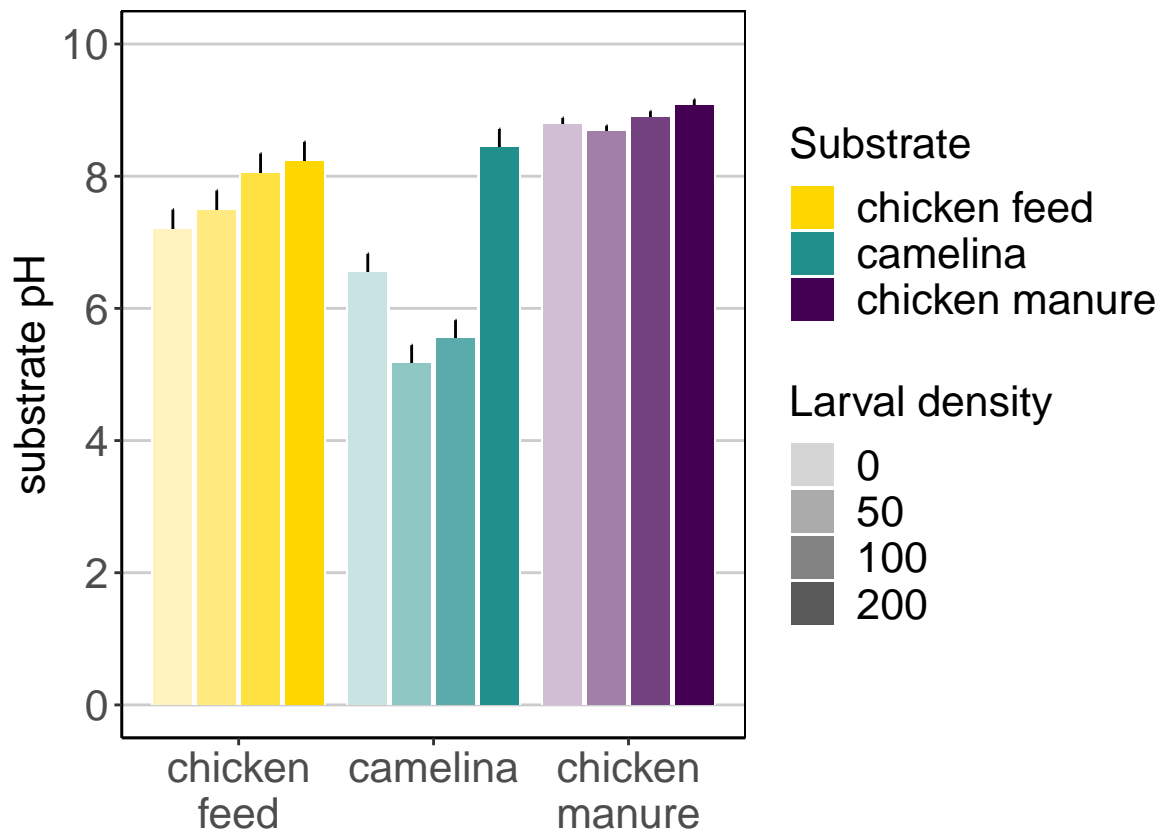
```
pPrep <- ggplot(subset(harv.emm,variable == "pPrepupae"),
  aes(x=Diet, y=emmean, alpha = Density,
    group=Density)) +
  geom_errorbar(aes(ymin=emmean, ymax=emmean+SE), alpha = 1,
    width=0, position= position_dodge(.9)) +
  geom_col(fill = "white", alpha = 1, position = position_dodge(.9)) +
  geom_col(aes(fill = Diet), width = .8, position = position_dodge(.9)) +
  labs(y="prepupae (%)", x=NULL) +
  scale_y_continuous(limits=c(-.1,110), n.breaks=6) +
  scale_fill_manual(values = c("gold", "#21908CFF", "#440154FF")) +
  scale_alpha_ordinal(range = c(0.5, 1)) +
  guides(fill = "none") +
  theme_perf
pPrep
```



```
pInd <- ggplot(subset(harv.emm,variable == "dmLarvInd"),
  aes(x=Diet, y=emmean, alpha = Density,
    group=Density)) +
  geom_errorbar(aes(ymin=emmean, ymax=emmean+SE), alpha = 1,
    width=0, position= position_dodge(.9)) +
  geom_col(fill = "white", alpha = 1, position = position_dodge(.9)) +
  geom_col(aes(fill = Diet), width = .8, position = position_dodge(.9)) +
  labs(y="ind. larval weight (g DM)", x=NULL) +
  scale_y_continuous(limits=c(0,.1), n.breaks=6) +
  scale_fill_manual(values = c("gold", "#21908CFF", "#440154FF")) +
  scale_alpha_ordinal(range = c(0.5, 1)) +
  guides(fill = "none") +
  theme_perf
pInd
```



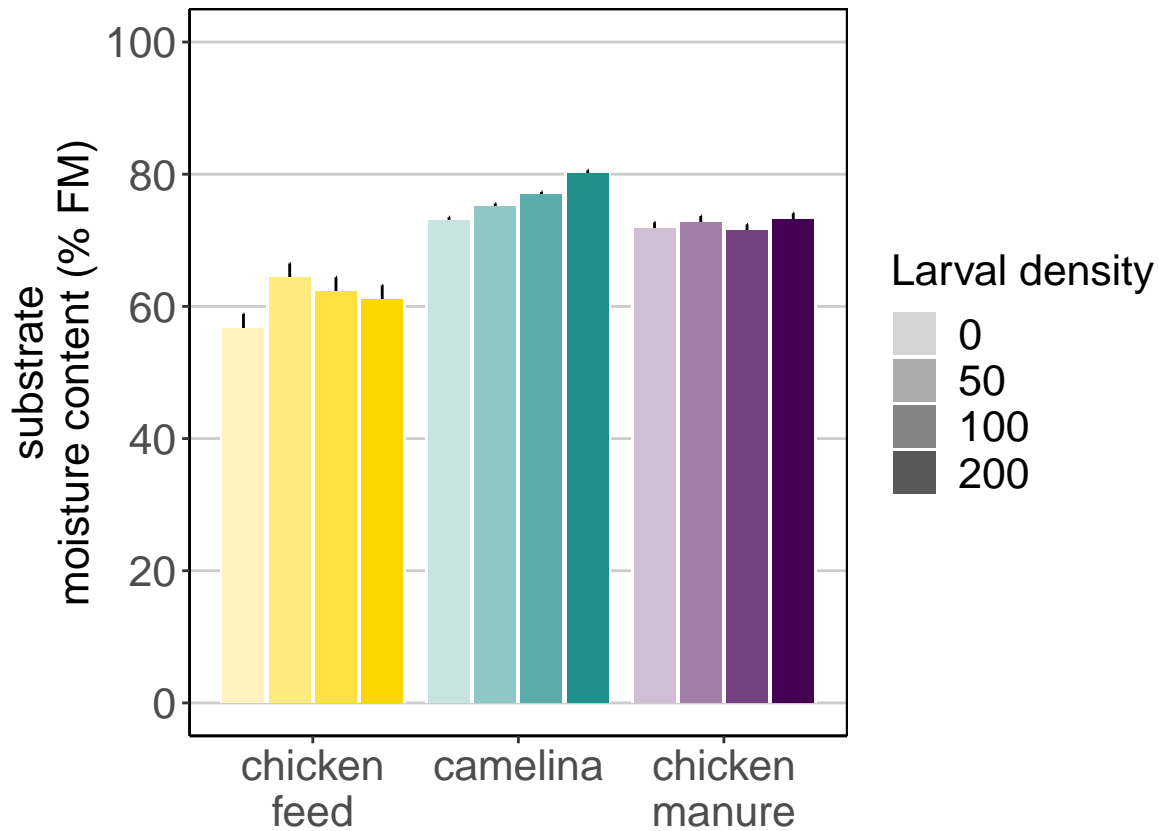
```
ppH <- ggplot(subset(harv.emm, variable == "pHres"),
  aes(x=Diet, y=emmean, alpha = Density,
    group=Density)) +
  geom_errorbar(aes(ymin=emmean, ymax=emmean+SE), alpha = 1,
    width=0, position= position_dodge(.9)) +
  geom_col(fill = "white", alpha = 1, position = position_dodge(.9)) +
  geom_col(aes(fill = Diet), width = .8, position = position_dodge(.9)) +
  labs(y="substrate pH", x=NULL) +
  scale_y_continuous(limits=c(0,10), n.breaks=6) +
  scale_fill_manual("Substrate",
    values = c("gold", "#21908CFF", "#440154FF"),
    labels = c("chicken feed", "camelina", "chicken manure")) +
  scale_alpha_ordinal("Larval density", range = c(0.25, 1)) +
  guides(fill = guide_legend(order = 1),
    alpha = guide_legend(order = 2)) +
  theme_perf
ppH
```



```

pmoist <- ggplot(subset(harv.emm,variable == "pMoist"),
  aes(x=Diet, y=emmean, alpha = Density,
    group=Density)) +
  geom_errorbar(aes(ymin=emmean, ymax=emmean+SE), alpha = 1,
    width=0, position= position_dodge(.9)) +
  geom_col(fill = "white", alpha = 1, position = position_dodge(.9)) +
  geom_col(aes(fill = Diet), width = .8, position = position_dodge(.9)) +
  labs(y="substrate\nmoisture content (% FM)", x=NULL) +
  scale_y_continuous(limits=c(0,100), n.breaks=6) +
  scale_fill_manual(values = c("gold", "#21908CFF", "#440154FF")) +
  scale_alpha_ordinal("Larval density", range = c(0.25, 1)) +
  guides(fill = "none") +
  theme_perf
pmoist

```



```
# plot only legend
perf.legend <- get_legend(ppH)
perf.leg <- as_ggplot(perf.legend)
ggsave(plot=perf.leg, "./figures/Performance_leg.png", h = 3, w = 2)
ggsave(plot=perf.leg, "./figures/Performance_leg.pdf", h = 75, w = 50, u = "mm")

# arrange all into one plot
pPerf <- ggarrange(pSurv, pPrep, pInd, ppH,
  nrow=2, ncol=2, align="hv",
  legend="right", common.legend=T,
  labels = "auto", label.x = .21, label.y = 1,
  font.label = list(size = 20, face = "plain"))
ggsave(plot = pPerf, "./figures/Performance_SE.png", w = 10, h = 7)
ggsave(plot = pPerf, "./figures/Performance_SE.pdf", w = 250, h = 175, u = "mm")

ggsave(plot = pmoist, "./figures/MoistureContent_SE.png", w = 8, h = 6)
ggsave(plot = pmoist, "./figures/MoistureContent_SE.pdf", w = 200, h = 150, u = "mm")
```

## 5. pH of starting diets

We prepared fresh diets of CF and CS and measured pH; for CM, we used subsamples of frozen reference samples of the four batches from the experiment.

## 5.1. Prepare data

```
# factors
dietph$Diet <- factor(dietph$Diet, levels(dietph$Diet)[c(1,3,2)])

# for CM, we have 4 batches
dietph$Batch <- as.factor(dietph$Batch)
# NB: batch 1 in CF has nothing to do with batch 1 in CS or CM!
# recode Batch: for CF and CS need different level than the four of CM
dietph$Batch2 <- ifelse(dietph$Diet == "CF", 5,
                       ifelse(dietph$Diet == "CS", 6, dietph$Batch))
dietph$Batch2 <- as.factor(dietph$Batch2)
```

## 5.2. LMM regression

Linear mixed model regression because we wanted to account for Batch effect (random intercept).

```
# test random intercept and varIdent structure
phd.m0 <- gls(pH ~ Diet, dietph, method = "REML")
phd.m1 <- lme(pH ~ Diet, dietph, method = "REML", random = ~1|Batch2)
phd.m2 <- update(phd.m1, weights = varIdent(form = ~1|Diet))
AIC(phd.m0, phd.m1, phd.m2)
```

```
##          df          AIC
## phd.m0   4 15.370778
## phd.m1   5 14.908220
## phd.m2   7  8.615939
```

*# model m2 is best: random term and variance structure for Diet.*

```
# model output
anova(phd.m2)
```

```
##          numDF denDF  F-value p-value
## (Intercept)     1    18 8570.699 <.0001
## Diet           2     3   84.277  0.0023
```

```
CLD(emmeans(phd.m2, ~Diet, method = "tukey"), Letters = letters)
```

```
## Diet emmean    SE df lower.CL upper.CL .group
## CS      5.44 0.153  3     4.95     5.93    a
## CF      6.26 0.157  5     5.85     6.66    a
## CM      7.68 0.100  3     7.36     8.00    b
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
## P value adjustment: tukey method for comparing a family of 3 estimates
## significance level used: alpha = 0.05
```