

Reproduction of results in George et al, PeerJ

Jean-Olivier Irisson

Thu May 24 11:20:02 2018

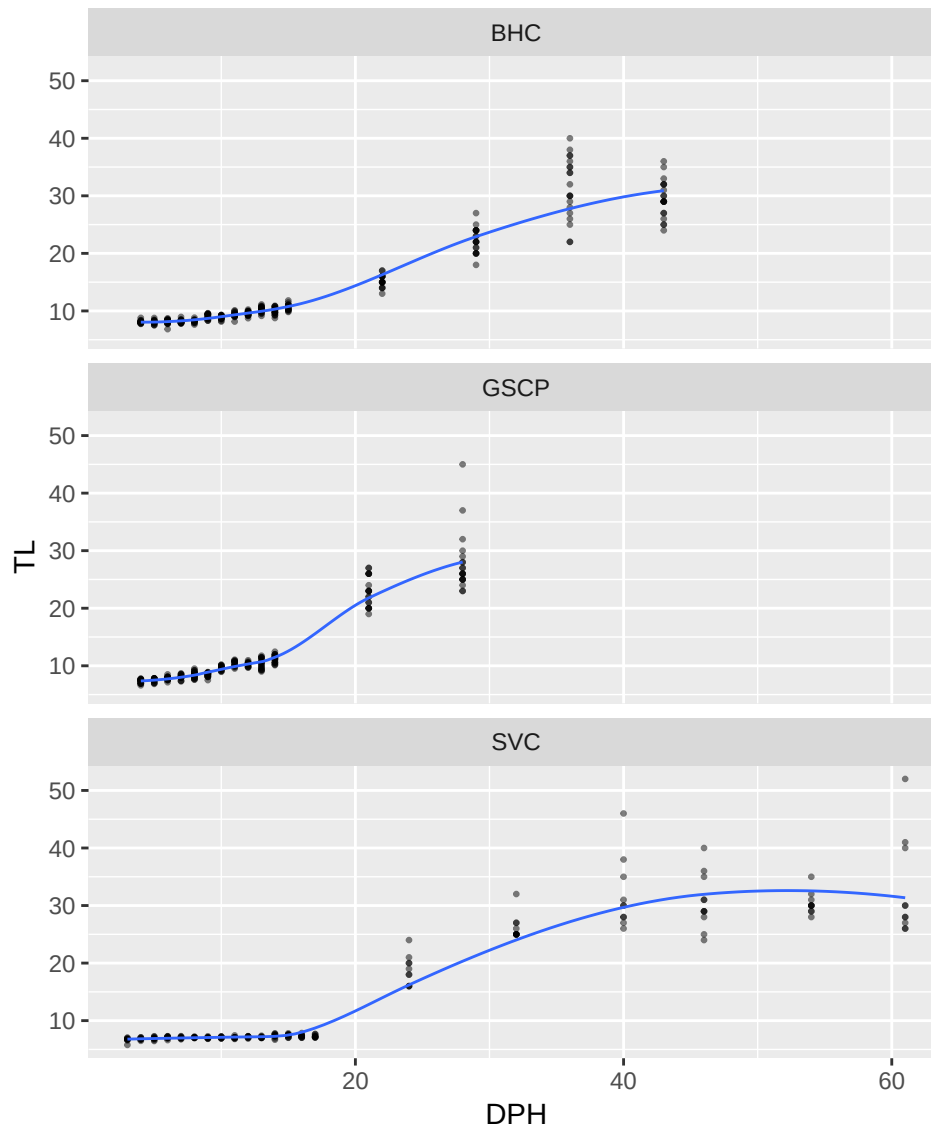
```
suppressMessages(library("tidyverse"))
suppressMessages(library("readxl"))
suppressMessages(library("gridExtra"))
suppressMessages(library("broom"))
suppressMessages(library("lmtest"))
suppressMessages(library("splines"))

# Predefine some plotting functions for homogeneity of aspect
# data points
gpnts <- geom_point(alpha=0.5, size=0.5, na.rm=TRUE)
# non-parametric polynomial regression
gsmooth <- geom_smooth(method="loess", se=FALSE, size=0.5, na.rm=TRUE)
# linear regression
glm <- geom_smooth(method="lm", se=TRUE, size=0.5, colour="red", na.rm=TRUE)
```

Growth

```
# read data
d <- read_excel("peerj-27628-Swimming_speeds_data_for_review-aeg.xlsx", sheet=1, na="") %>%
  select(Species, DPH, TL)
# summary(d)

# plot
ggplot(d, aes(x=DPH, y=TL)) +
  facet_wrap(~Species, ncol=1) +
  gpnts + gsmooth
```



Maximum speed

```
# read data
d <- read_excel("peerj-27628-Swimming_speeds_data_for_review-aeg.xlsx", sheet=1, na="") %>%
  select(Species, DPH, Hz, TL)
# summary(d)

# remove outlier
# NB: this should be fixed in the input data table
d <- filter(d, Hz != 60)

# compute speed in cm/s
d$Absolute <- (0.0218 * d$Hz - 0.0731) * 100 # in cm/s rather than m/s ?
d$Relative <- d$Absolute / d$TL
```

```

# Check how to compute mean relative speed
# keep only data rows where both information are available
df <- filter(d, !is.na(Absolute), !is.na(TL))
# per individual
mean(df$Absolute / df$TL)

## [1] 2.151139

# -> this would be the proper way of computing it

# divided by mean length (as the authors did)
mean(df$Absolute / mean(df$TL))

## [1] 1.993252

mean(df$Absolute) / mean(df$TL)

## [1] 1.993252

# -> both are equivalent (of course) but incorrect

# Regress absolute or relative speed on various quantities
# fit models and extract R2 and p-value
regress <- function(formula, data) {
  # fit model
  m <- lm(formula=formula, data=data)
  # extract diagnostics
  diag <- glance(m) %>% select(r.squared, adj.r.squared, p.value)
  # add tests of assumptions
  diag$norm.p.value <- shapiro.test(residuals(m))$p.value
  diag$homosc.p.value <- bptest(m)$p.value
  return(as.data.frame(diag))
}
(ma <- d %>% group_by(Species) %>% do(regress(Absolute ~ DPH, data=..)) %>% ungroup())

## # A tibble: 3 x 6
##   Species r.squared adj.r.squared p.value norm.p.value homosc.p.value
##   <chr>      <dbl>      <dbl>    <dbl>      <dbl>      <dbl>
## 1 BHC      0.592      0.591 7.26e-64    2.91e- 2    7.65e- 8
## 2 GSCP      0.676      0.675 4.22e-65    6.57e-10    3.63e- 9
## 3 SVC      0.839      0.838 2.62e-84    1.29e-12    1.18e-11

(mr <- d %>% group_by(Species) %>% do(regress(Relative ~ DPH, data=..)) %>% ungroup())

## # A tibble: 3 x 6
##   Species r.squared adj.r.squared p.value norm.p.value homosc.p.value
##   <chr>      <dbl>      <dbl>    <dbl>      <dbl>      <dbl>
## 1 BHC      0.244      0.241 3.23e-20    0.0265     0.000000397
## 2 GSCP      0.0116     0.00765 8.92e- 2    0.0322     0.105
## 3 SVC      0.00416    -0.000629 3.52e- 1    0.00000186 0.0000518

(ml <- d %>% group_by(Species) %>% do(regress(Absolute ~ TL, data=..)) %>% ungroup())

## # A tibble: 3 x 6
##   Species r.squared adj.r.squared p.value norm.p.value homosc.p.value
##   <chr>      <dbl>      <dbl>    <dbl>      <dbl>      <dbl>
## 1 BHC      0.554      0.552 3.44e-55    3.82e- 3    1.52e- 7

```

```
## 2 GSCP      0.555      0.554 2.30e-45      1.09e-11      2.70e-16
## 3 SVC       0.836      0.835 1.63e-83      4.00e- 6      1.05e-12
```

```
# NB: assumptions for the linear model are often NOT respected
```

```
# pick one model to examine the diagnostics
```

```
m <- lm(Absolute ~ DPH, data=filter(d, Species == "SVC"))
```

```
# plot(m, 2)
```

```
# -> not normal
```

```
# plot(m, 3)
```

```
# -> not homoscedastic
```

```
# plot absolute
```

```
pa <- ggplot(d, aes(x=DPH, y=Absolute)) +
```

```
  facet_wrap(~Species, ncol=1) +
```

```
  gpoints + gsmooth + glm +
```

```
  geom_label(aes(x=60, y=96, label=paste0(
```

```
    "R2=", round(r.squared, 2),
```

```
    ifelse(p.value<0.001, " p<0.001", paste0(" p=", round(p.value, 3)))
```

```
  )), data=ma, size=3.5, label.size=0, hjust="right", )
```

```
# plot relative
```

```
pr <- ggplot(d, aes(x=DPH, y=Relative)) +
```

```
  facet_wrap(~Species, ncol=1) +
```

```
  gpoints + gsmooth + glm +
```

```
  geom_label(aes(x=60, y=4.15, label=paste0(
```

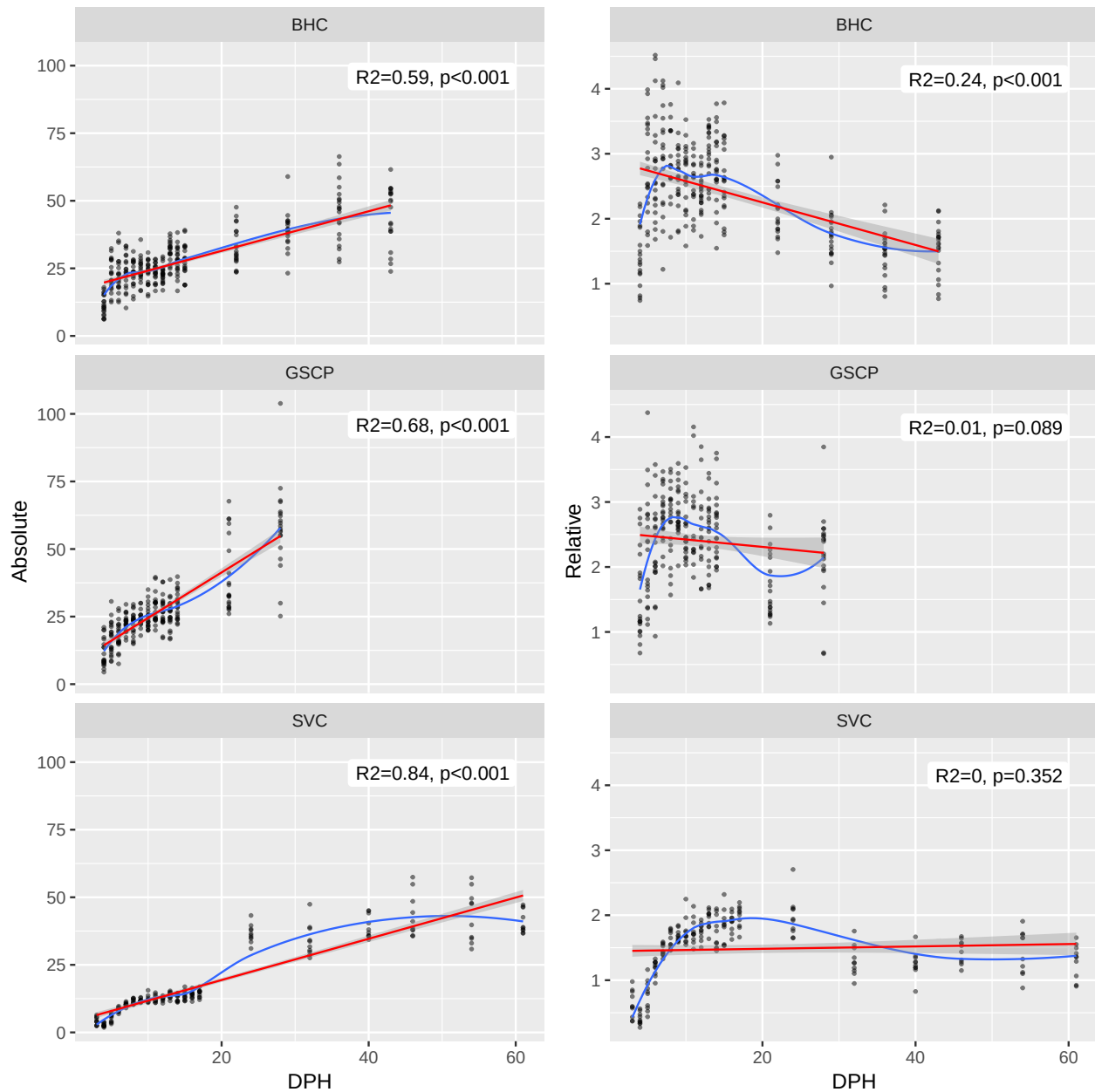
```
    "R2=", round(r.squared, 2),
```

```
    ifelse(p.value<0.001, " p<0.001", paste0(" p=", round(p.value, 3)))
```

```
  )), data=mr, size=3.5, label.size=0, hjust="right")
```

```
# combine plots
```

```
grid.arrange(pa, pr, nrow=1)
```



Routine speed

```
# read data
d <- read_excel("peerj-27628-Swimming_speeds_data_for_review-aeg.xlsx", sheet=2) %>%
  rename(Horizontal=Vx, Vertical=Vy)
# summary(d)

# keep all data until the same age
d <- filter(d, DPH < 15)

# reformat data
d <- gather(d, key="direction", value="Speed", Horizontal, Vertical)
```

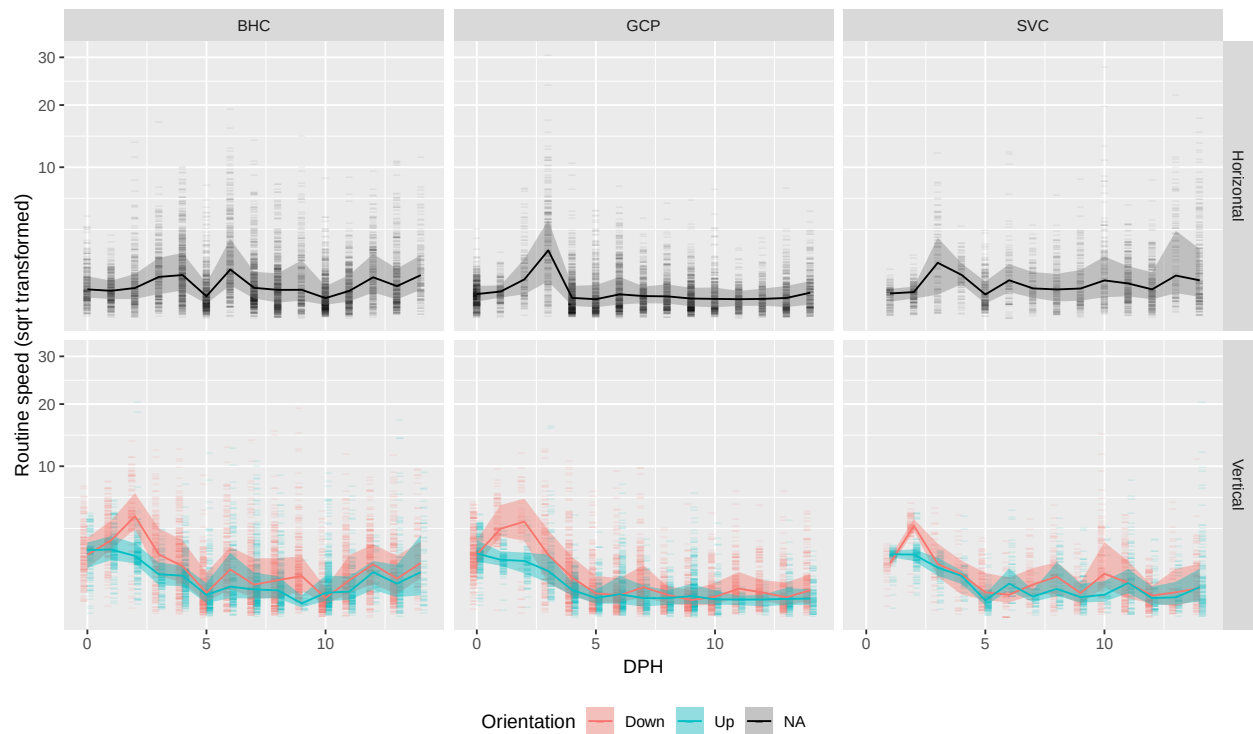
```

# define up/down or no direction
d$Orientation <- ifelse(d$Speed > 0, "Up", "Down")
d$Orientation[d$direction == "Horizontal"] <- NA
# compute absolute value of swimming speed (irrespective of direction)
d$Speed <- abs(d$Speed)

# compute summary values describing swimming speeds through time
ds <- d %>%
  # per species, horizontal/vertical, up/down, and day
  group_by(Species, direction, Orientation, DPH) %>%
  # compute descriptive statistics
  summarise(
    mean=mean(Speed), sd=sd(Speed),
    median=median(Speed), mad=mad(Speed),
    q25=quantile(Speed, 0.25), q75=quantile(Speed, 0.75)
  )

# plot speeds and summary statistics
ggplot(d, aes(x=DPH, colour=Orientation)) +
  facet_grid(direction~Species) +
  # data points (faint)
  geom_point(
    aes(y=Speed, alpha=direction), size=2, shape="_",
    position=position_dodge(width=0.5)
  ) +
  scale_alpha_manual(values=c(0.1, 0.2), guide="none") +
  # median and quartiles
  geom_ribbon(aes(ymin=q25, ymax=q75, fill=Orientation), data=ds, alpha=0.4, colour=NA) +
  geom_path(aes(y=median), data=ds) +
  # square-root transform y axis to compress a bit the very high values
  scale_y_continuous(trans="sqrt") +
  labs(y="Routine speed (sqrt transformed)") +
  # make horizontal speed black
  scale_colour_discrete(na.value="black") +
  theme(legend.position="bottom")

```



```
# plot speeds and smoothed summary statistics
ggplot(d, aes(x=DPH, colour=Orientation)) +
  facet_grid(direction~Species) +
  # data points (faint)
  geom_point(
    aes(y=Speed, alpha=direction), size=2, shape="_", na.rm=TRUE,
    position=position_dodge(width=0.5)
  ) +
  scale_alpha_manual(values=c(0.1, 0.2), guide="none") +
  # spline-smoothed quartiles and median
  geom_path(aes(y=q75), data=ds, stat="smooth", method=lm,
    formula=y ~ bs(x,9), linetype="21", lineend="round") +
  geom_path(aes(y=median), data=ds, stat="smooth", method=lm, formula=y ~ bs(x,9)) +
  geom_path(aes(y=q25), data=ds, stat="smooth", method=lm,
    formula=y ~ bs(x,9), linetype="21", lineend="round") +
  # square-root transform y axis to compress a bit the very high values
  scale_y_continuous(trans="sqrt") +
  labs(y="Routine speed (sqrt transformed)") +
  # make horizontal speed black
  scale_colour_discrete(na.value="black") +
  theme(legend.position="bottom")
```

