

A unified-models analysis of the development of sexual size dimorphism in Damaraland Mole-rats

Jack Thorley (jackthorley1@gmail.com) & Tim Clutton-Brock

February 2019

Overview

This file provides accompanying ‘R’ code for our paper, “A unified-models analysis of the development of sexual size dimorphism in Damaraland Mole-rats”. Below we provide code to fit the Unified-sigmoid family of growth functions to mole-rat skeletal data within a mixed effects modelling framework. Like parameters estimated with the Unified functions can be directly compared, and each parameter only affects a single growth curve characteristic, which facilitates comparisons of the form of growth across datasets (as between the sexes). Before following this code, we recommend that readers are familiar with the following two papers:

- Tjorve, E. & Tjorve, K.M.C. (2010) *A unified approach to the Richards-model family for use in growth analyses: Why we need only two model forms*. *J. Theoretical Biology*, 267, 417-425.
- Tjorve, K.M.C. & Tjorve, E. (2017) *A proposed family of Unified models for sigmoidal growth*. *Ecological Modelling*, 359, 117-127.

The non-linear mixed effects models fitted throughout the paper, and in this tutorial, are fitted using the `nlme` package in the R environment. For more information on the use of `nlme`, and the fitting of mixed effects models in R see the `nlme` vignette, and <https://bbolker.github.io/mixedmodels-misc/glmmFAQ.html>. We also make use of the `tidyverse` package to manipulate data and model output. Details on the syntax of the `tidyverse`, which deviates somewhat from base R, can be found here: <http://r4ds.had.co.nz/>. Users need to have installed the following packages in order to run the code that follows:

- `nlme4`
- `tidyverse`
- `broom`
- `mgcv`
- `pracma`

The code below consists of two sections. In **Section 1**, we demonstrate how we modelled growth in captive mole-rats. For the purposes of this tutorial, we only do so for incisor width (skull size), but the same methods were employed for body length. Then, in **Section 2**, we model growth in wild Damaraland mole-rats. As we do not know the age of wild-caught individuals, we could not use conventional Unified models to do so. We instead used the ‘interval equations’ outlined in Schoener and Schoener (1978), which model size at time $t+1 \sim$ size at time t + the interval difference.

Section 1: Modelling sex differences in teeth width (skull size) in captive mole-rats

Load in the female teeth width dataset. We filter the data to only consider individuals when younger than 1200 days old, when asymptotic size had been reached (~ 3 years old)

```

library(nlme)
library(tidyverse)
library(broom)
library(mgcv)

female_teeth <- read.csv("DMR_femaletooth.csv") %>%
  filter(!is.na(teethwidth) & Age < 1200)

str(female_teeth) # look at the dataset to be familiar with the contents

```

Before starting, we first take a look at the raw data. We can fit a GAMM through the raw data to get an idea of the shape of growth.

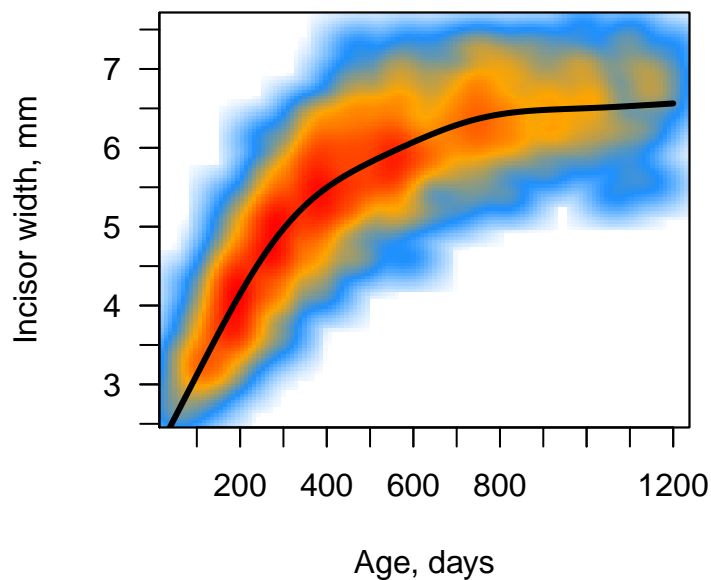
```

gammfem_teeth <- gamm(teethwidth ~ s(Age, k = 6), random = list(AnimalID = ~1),
  data = female_teeth)
plx <- data.frame(Age = seq(2, 1200, 1)) # females
plx <- cbind(plx, predict(gammfem_teeth$gam, newdata = plx, se.fit = TRUE))

# set the colour palette
clrs <- colorRampPalette(c("white", "dodgerblue", "orange", "red"), alpha = 0.8)

smoothScatter(female_teeth$Age, female_teeth$teeth, las = 1, xlab = "Age, days",
  ylab = "Incisor width, mm", nrpoints = 0, colramp = clrs)
axis(1, at = seq(100, 1200, 100), labels = FALSE) ; axis(2, at = seq(2, 8, 0.5),
  labels = FALSE) ; lines(plx$Age, plx$fit, lwd = 3)

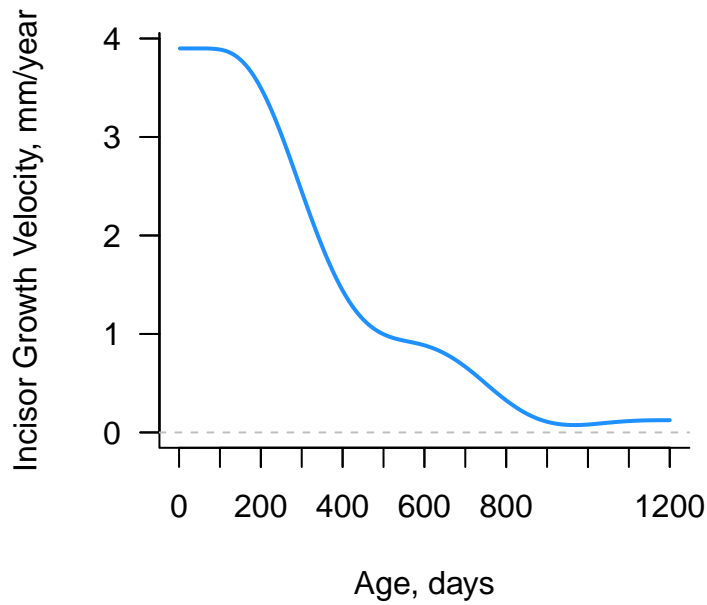
```



This plot suggest little evidence of an inflection point in growth. To get a better idea if this is the case, we

can also plot the rate of change in growth with age by calculating the gradient across the fitted slope using the *pracma* function.

```
# convert the curve to a rate with pracma (gradient across the fitted slope)
plx$teethvelocity <- pracma::gradient(as.vector(plx$fit))*365
plot(plx$Age, plx$teethvelocity, type = "l", las = 1, lwd = 2,
     ylab = "Incisor Growth Velocity, mm/year", xlab = "Age, days", col = "dodgerblue",
     ylim = c(0, max(plx$teethvelocity)), xlim = c(0, 1200), bty = "l")
axis(1, at = seq(100, 1200, 100), labels = FALSE)
axis(2, at = seq(0, 10, 1), labels = NA)
abline(h = 0, col = "grey", lty = 2)
```



This plot seems to confirm the general lack of inflection. But we can check this once we fit the candidate growth models.

Female teeth width in captivity: U-von Bertalanffy model

$$S_t = A. \left(1 - \frac{1}{3}.e^{-\frac{3}{4}.k.(t-T_i)} \right)^3$$

- where S_t is the size (incisor width) at age t ; a fitted value
- A is the upper asymptote (adult size value) to be estimated
- k is the relative maximum growth to be estimated
- T_i is the time of inflection to be estimated

Before running the non-linear mixed effects model, it is often advisable to fit a straightforward non-linear model using nonlinear least-squares. *nls* is more forgiving in locating sensible estimates than *lme*, which is more likely to face convergence issues. The *nls* estimates can then be plugged into *lme*.

```
nls.vb.fteeth <- nls(teethwidth ~ A*(1-((1/3)*exp(-(9/4)*k*(Age-Ti))))^3,
                    data=female_teeth,
                    start = c(A=6,k=0.003,Ti=5))
summary(nls.vb.fteeth) # A = 6.63, k = 0.00194; T= 3.90
```

We can use the output from this model to now inform the non-linear mixed effects model. In this model, we include random effects at the level of the individual on each of the three parameter values.

```
femteeth.vb.nlmm <- nlme(teethwidth ~ A*(1-((1/3)*exp(-(9/4)*k*(Age-Ti))))^3,
                        fixed=A+k+Ti~1, random= list(AnimalID = pdDiag(A + k + Ti~ 1)),
                        data=female_teeth,
                        weights = varPower(1),
                        start = c(A=6.63,k=0.002,Ti=3.75),
                        correlation = corAR1(form = ~1|AnimalID))
summary(femteeth.vb.nlmm) # A = 6.59, k = 0.00195; T= 4.63
```

- We also include a power function to account for the increased variance with increasing age, using nlme's 'varPower' function
- And an AR-1 auto-correlation structure, $correlation = corAR1(form = \sim 1|AnimalID)$, to estimate temporal autocorrelation on observations from individuals
- $pdDiag$ specifies that the random effects are uncorrelated.

In order to determine the appropriate parameters on which to estimate individual level variation, we also code a number of additional models that differ in the combination of the random effects.

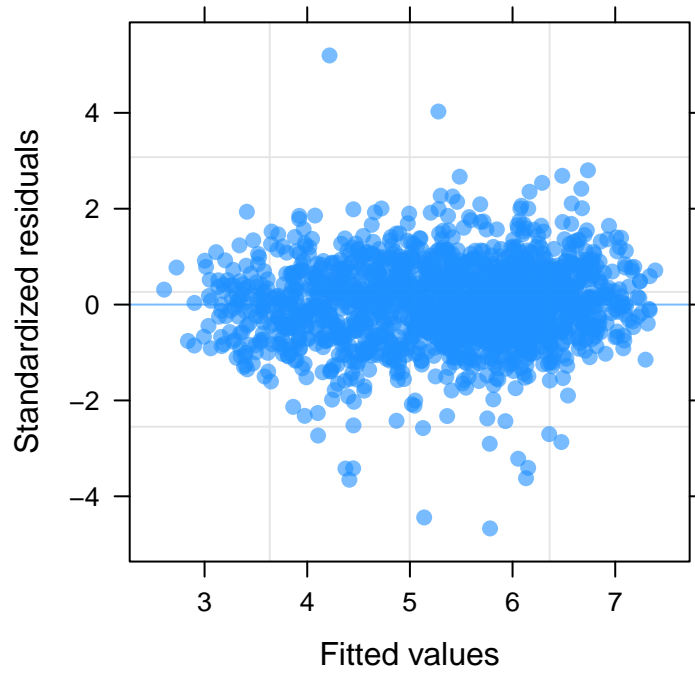
```
femteeth.vb.nlmm2 <- update(femteeth.vb.nlmm, random= list(AnimalID = pdDiag(k + Ti~ 1)))
femteeth.vb.nlmm3 <- update(femteeth.vb.nlmm, random= list(AnimalID = pdDiag(A + k~ 1)))
femteeth.vb.nlmm4 <- update(femteeth.vb.nlmm, random= list(AnimalID = pdDiag(A + Ti~ 1)))
# we also specified single combinations but not shown here
```

Comparing the models by AIC shows that for the U-von Bertalanffy model, the best fitting random effects structure includes an individual-level random effect for the Asymptote A and the growth rate constant k .

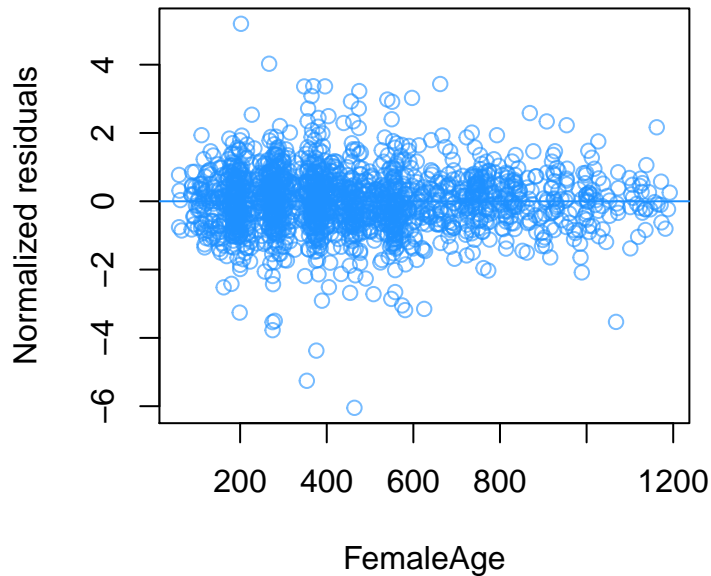
```
AIC(femteeth.vb.nlmm, femteeth.vb.nlmm2, femteeth.vb.nlmm3, femteeth.vb.nlmm4) %>%
  mutate(delta = AIC - min(AIC)) # compare models with Akaike's Information Criterion
summary(femteeth.vb.nlmm3) # 'Best' model as judged by lowest AIC (where -2 < delta < 2)
```

And check the fit of the best fitting model

```
# par(mfrow = c(2,1))
plot(femteeth.vb.nlmm3, pch = 16,
     col = adjustcolor("dodgerblue", alpha.f = 0.6))
```



```
plot(female_teeth$Age, resid(femteeth.vb.nlm3, type = c("normalized")),  
     ylab = "Normalized residuals", xlab = "FemaleAge",  
     col = adjustcolor("dodgerblue", alpha.f = 0.6))  
abline(h = 0,col = "dodgerblue")
```



We can replicate this procedure on the other three-parameter models.

Female teeth width in captivity: U-Gompertz model

$$S_t = A.e^{-e^{-e.k.(t-T_i)}}$$

```
femteeth.gomp.nlm1 <- nlme(teethwidth ~ A*(exp(-exp(-exp(1)*k*(Age-Ti)))),
  fixed=A+k+Ti~1, random= list(AnimalID = pdDiag(A + k + Ti ~ 1)),
  data=female_teeth,
  weights = varPower(1),
  start = c(A=6.59,k=0.002,Ti=33.4),
  correlation = corAR1(form = ~1|AnimalID))
summary(femteeth.gomp.nlm1) # A = 6.55, k = 0.00175, Ti = 34.49

femteeth.gomp.nlm2 <- update(femteeth.gomp.nlm1,
  random= list(AnimalID = pdDiag(k + Ti ~ 1)))
femteeth.gomp.nlm3 <- update(femteeth.gomp.nlm1,
  random= list(AnimalID = pdDiag(A + k ~ 1)))
femteeth.gomp.nlm4 <- update(femteeth.gomp.nlm1,
  random= list(AnimalID = pdDiag(A + Ti ~ 1)))
AIC(femteeth.gomp.nlm1, femteeth.gomp.nlm2, femteeth.gomp.nlm3, femteeth.gomp.nlm4)
# model checks carried out separately
```

Female teeth width in captivity: U-Logistic model

$$S_t = \frac{A}{1 + e^{-4.k.(t-T_i)}}$$

```
femteeth.log.nlm3 <- nlme(teethwidth ~ A/(1+exp(-4*k*(Age-Ti))),
  fixed=A+k+Ti~1, random= list(AnimalID = pdDiag(A + k + Ti ~ 1)),
  data=female_teeth,
  weights = varPower(1),
  start = c(A=6.51,k=0.001,Ti=102),
  correlation = corAR1(form = ~1|AnimalID))
summary(femteeth.log.nlm3) # A = 6.48, k = 0.00147, Ti = 101.79

femteeth.log.nlm2 <- update(femteeth.log.nlm3,
  random= list(AnimalID = pdDiag(k + Ti ~ 1)))
femteeth.log.nlm4 <- update(femteeth.log.nlm3,
  random= list(AnimalID = pdDiag(A + k ~ 1)))
femteeth.log.nlm5 <- update(femteeth.log.nlm3,
  random= list(AnimalID = pdDiag(A + Ti ~ 1)))
AIC(femteeth.log.nlm3, femteeth.log.nlm2, femteeth.log.nlm4, femteeth.log.nlm5)
```

Female teeth width in captivity: U4 model

$$S_t = A. \left(1 + 3.e^{-4\frac{4}{3}.k.(t - T_i)} \right)^{-\frac{1}{3}}$$

```
femteeth.U4.nlm3 <- nlme(teethwidth ~ A*(1+(3*exp(-(4^(4/3))*k*(Age-Ti))))^(-1/3),
  fixed=A+k+Ti~1, random= list(AnimalID = pdDiag(A + k + Ti ~ 1)),
  data=female_teeth,
  weights = varPower(1),
  start = c(A=6.42,k=0.001,Ti=187),
  correlation = corAR1(form = ~1|AnimalID))
summary(femteeth.U4.nlm3) # A = 6.45, k = 0.00121, Ti = 189.07

femteeth.U4.nlm2 <- update(femteeth.U4.nlm3,
  random= list(AnimalID = pdDiag(k + Ti ~ 1)))
femteeth.U4.nlm4 <- update(femteeth.U4.nlm3,
  random= list(AnimalID = pdDiag(A + k ~ 1)))
femteeth.U4.nlm5 <- update(femteeth.U4.nlm3,
  random= list(AnimalID = pdDiag(A + Ti ~ 1)))
AIC(femteeth.U4.nlm3, femteeth.U4.nlm2, femteeth.U4.nlm4, femteeth.U4.nlm5)
```

Female teeth width in captivity: U-Richards model

$$S_t = A. \left(1 + (d - 1).e^{\left(\frac{-k.(t - T_i)}{d^{1-d}} \right)} \right)^{\frac{1}{1-d}}$$

The standard version of the U-Richards model did not converge, irrespective of using parameter estimates from an initial *nls* model. To overcome this issue, we manually optimized the shape parameter *d*.

```
Richards.mod <- function(A,k,Ti,Age){
  temp <- Age*0
  temp <- A*(1+(d-1)*exp((-k*(Age-Ti))/(d^(d/(1-d))))^(1/(1-d))
  temp[d<0][is.na(temp[d<0])] <- 0
  return(temp)
```

```

}

richards.negLL = function(D) {
  d <- D
  modR <- nlme(teethwidth ~ Richards.mod(A=A,k=k,Ti=Ti,Age=Age),
    fixed=A+k+Ti~1, random= list(AnimalID = pdDiag(A + k + Ti ~ 1)),
    data = female_teeth,
    start =c(A=6.63,k=0.002,Ti=100),
    correlation = corAR1(form = ~1|AnimalID),
    na.action = na.pass, method = "ML",
    weights = varPower(1)
  )
  print(paste(d, "negLL:", -modR$logLik))
  return(-modR$logLik)
}

optimize(f=richards.negLL, interval=c(0.5,4)) # d = 0.82
femteeth.richards.nlm3 <- modR
summary(femteeth.richards.nlm3) # A = 6.49, k = 0.001, Ti = 92.51

femteeth.richards.nlm2 <- update(femteeth.richards.nlm3,
  random= list(AnimalID = pdDiag(k + Ti~ 1)))
femteeth.richards.nlm3 <- update(femteeth.richards.nlm3,
  random= list(AnimalID = pdDiag(A + k~ 1)))
femteeth.richards.nlm4 <- update(femteeth.richards.nlm3,
  random= list(AnimalID = pdDiag(A + Ti~ 1)))
AIC(femteeth.richards.nlm3, femteeth.richards.nlm2,
  femteeth.richards.nlm3, femteeth.richards.nlm4)

# Because of optimising manually, we need to correct for the estimation of the extra parameter in this l
2*10 - 2*logLik(femteeth.richards.nlm3) # AIC = -84.52115

```

We can now draw up a summary of the model outputs

```

femaleteethmodels <- data.frame(Model = c("vb", "gomp", "logistic", "U4", "Richards"),
  k = c(AIC(femteeth.vb.nlm3, femteeth.gomp.nlm3, femteeth.log.nlm3,
    femteeth.U4.nlm3)$df, 9),
  AIC = c(AIC(femteeth.vb.nlm3, femteeth.gomp.nlm3, femteeth.log.nlm3,
    femteeth.U4.nlm3)$AIC, 2*9 - 2*logLik(femteeth.richards.nlm3))) %>%
  mutate(DeltaAIC = AIC - min(AIC))

```

This suggests that the U-von Bertalanffy model is the best fitting model for female teeth (though the evidence that it is superior to the Gompertz is not strong given Delta AIC). We can now plot the predictions for each of the models in femaleteethmodels. To do so we simply write a function for each model, into which the output values can be plugged.

```

vonbertfunc <- function(A, k, Ti, x){
  Weight <- A*(1-((1/3)*exp(-(9/4)*k*(x-Ti))))^3
  return(Weight)
}

gompertzfunc <- function(A, k, Ti, x){
  Weight <- A*(exp(-exp(-exp(1)*k*(x-Ti))))

```



```

    return(Weight)
}

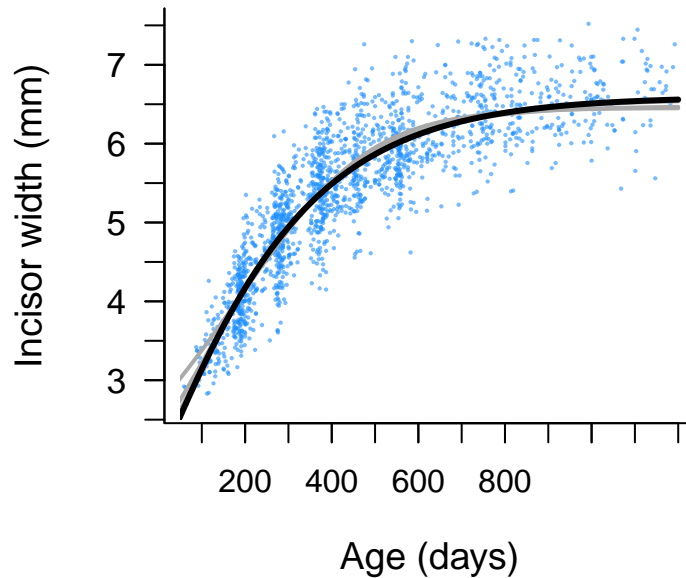
logisticfunc <- function(A, k, Ti, x){
  Weight <- A/(1+exp(-4*k*(x-Ti)))
  return(Weight)
}

U4func <- function(A, k, Ti, x){
  Weight <- A*(1+(3*exp(-(4^(4/3))*k*(x-Ti))))^(-1/3)
  return(Weight)
}

richardsfunc <- function(A, k, Ti, d, x){
  Weight <- A*(1+(d-1)*exp((-k*(x-Ti))/(d^(d/(1-d))))^(1/(1-d))
  return(Weight)
}

plot(teethwidth ~ Age, data = female_teeth, las = 1, xlab = "Age (days)",
     ylab = "Incisor width (mm)", pch = 16, cex = 0.3,
     col = adjustcolor("dodgerblue", alpha.f = 0.6), bty = "l", cex.axis = 1.1,
     cex.lab = 1.2, xaxt = "n")
axis(1, at = seq(100, 1200, 100), labels = FALSE)
axis(1, at = seq(200, 1000, 200), labels = seq(200, 1000, 200))
axis(2, at = seq(2, 8, 0.5), labels = FALSE)
clip(50, 100000, 2.5, 10000)
curve(gompertzfunc(x, A = 6.56546 , k = 0.00172, Ti = 32.04279),
      from = 0, to = 1200, add = TRUE, lwd = 2, lty = 1, col = "darkgrey")
curve(logisticfunc(x, A = 6.47652 , k = 0.00147, Ti = 101.78897),
      from = 0, to = 1200, add = TRUE, lwd = 2, lty = 1, col = "darkgrey")
curve(U4func(x, A = 6.45176 , k = 0.00121, Ti = 189.07690),
      from = 0, to = 1200, add = TRUE, lwd = 2, lty = 1, col = "darkgrey")
curve(richardsfunc(x, A = 6.580587, k = 0.001824, Ti = 17.407050, d = 0.82),
      from = 0, to = 1200, add = TRUE, lwd = 2, lty = 1, col = "darkgrey")
curve(vonbertfunc(x, A = 6.595021 , k = 0.001936, Ti = 3.645181),
      from = 0, to = 1200, add = TRUE, lwd = 3, lty = 1, col = "black")

```



In the code above, the best fitting U-von Bertalanffy is in black. The other U-forms are plotted as grey lines. Lastly, we can re-plot the growth curves as Instantaneous growth rates, by taking the the first derivative of the each predicted fit. Practically, we can just assess the gradient of the curve using `pracma`.

```
fem.IGR <- data.frame(Age = seq(0,1200, 1),
  vb.fit = vonbertfunc(x = seq(0, 1200, 1), A = 6.60 , k = 0.001936, Ti = 4))
fem.IGR$vb.IGR <- pracma::gradient(as.vector(fem.IGR$vb.fit))

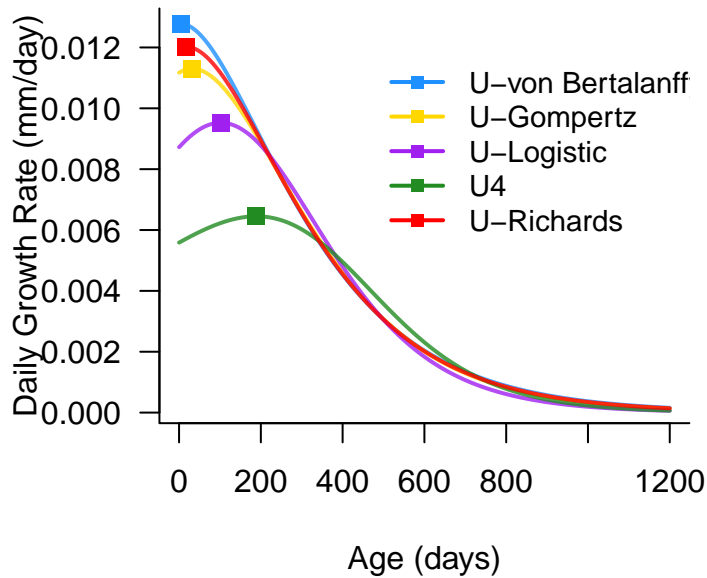
fem.IGR$gomp.fit <- gompertzfunc(x = seq(0, 1200, 1), A = 6.57, k = 0.00172, Ti = 32)
fem.IGR$gomp.IGR <- pracma::gradient(as.vector(fem.IGR$gomp.fit))

fem.IGR$log.fit <- logisticfunc(x = seq(0, 1200, 1), A = 6.48 , k = 0.00147, Ti = 102)
fem.IGR$log.IGR <- pracma::gradient(as.vector(fem.IGR$log.fit))

fem.IGR$U4.fit <- U4func(x = seq(0, 1200, 1), A = 6.45 , k = 0.001, Ti = 189)
fem.IGR$U4.IGR <- pracma::gradient(as.vector(fem.IGR$U4.fit))

fem.IGR$richards.fit <- richardsfunc(x = seq(0, 1200, 1),
  A = 6.580966 , k = 0.001827, Ti = 17.041730, d = 0.82)
fem.IGR$richards.IGR <- pracma::gradient(as.vector(fem.IGR$richards.fit))
```

Can then generate the plot:



Section 2: Modelling sex differences in incisor width growth in wild Damaraland mole-rats

As mole-rat captured in the wild are of unknown age, we cannot use conventional growth models. Instead, we comply an interval equation modelling approach, as outlined in: Schoener, T.W. & Schoener, A. (1978) *Estimating and Interpreting Body-Size Growth in Some Anolis Lizards*. Copeia, 1978, 390-405.

Load in the female teeth width dataset.

```
femteethwild <- read.csv("TeethWidthWild.csv") %>%
  filter(Sex == "F" & StatusCapt2 != "Q") %>%
  mutate(teethchange = teethwidth2 - teethwidth) %>%
  filter(!is.na(teethchange)) %>%
  arrange(teethwidth)
```

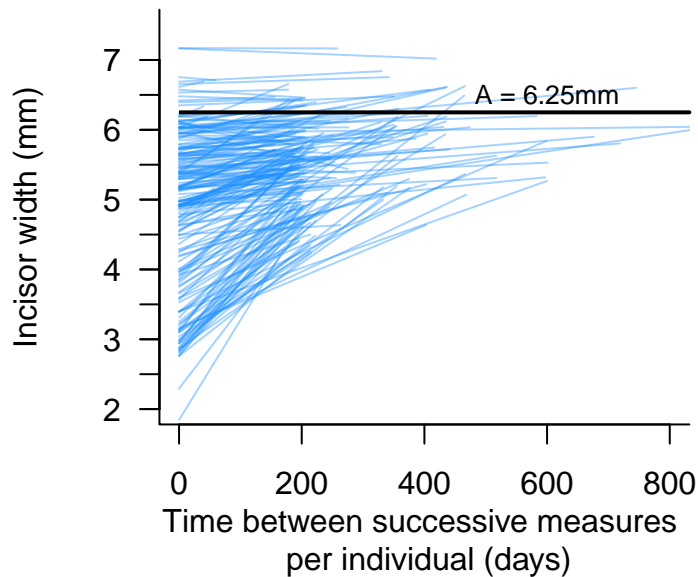
One way of visualising this data is to plot the time between recaptures. i.e. Tooth width for each capture (time t1) and recapture (time t2) within an individual is represented by a single slope. When represented this way, we notice a convergence of points towards the predicted asymptote, which is provided in the code that subsequently follows.

```
plot(teethwidth2 ~ timediff, data = femteethwild, type = "n", las = 1, bty = "l",
     ylim = c(2, 7.5), xlim = c(0, 800), ylab = "Incisor width (mm)",
     xlab = c("Time between successive measures \n per individual (days)"))
axis(2, at = seq(2,7, 0.5), labels = NA)
for (i in 1:nrow(femteethwild))
{
  with(femteethwild[i,],
       lines(x = c(0, timediff), y = c(teethwidth, teethwidth2),
```

```

    col = adjustcolor("dodgerblue", alpha.f = 0.4)
  )
}
clip(0, 2000, -1000, 1000)
abline(h = 6.25, lty = 1, lwd = 2, col = "black")
text(600, 6.5, "A = 6.25mm", cex = 0.8)

```



So how do we model data that looks this like this? Well, we can use the interval equations outlined in Schoener and Schener (1978). We take two specific interval equations from this paper, a **von Bertalanffy** and a **Logistic** interval equation. Note that in these models, the growth constant parameter k is not Unified in the same sense as in **Section 1**; the models are nonetheless useful to compare estimate asymptotic mass in the wild compared to captivity, and for making qualitative comparisons about differences in growth rate between males and females (.

Female teeth width in the wild: The von-Bertalanffy interval equation

$$S_2 = A - (A - S_1)e^{-k \cdot D}$$

- where S_1 and S_2 are the size (incisor width) at time 1 (capture 1) and time 2 (capture 2), respectively
- D is the time difference between the two capture events
- A is the upper asymptote to be estimated
- k is the growth rate constant to be estimated

Again we specify the model in `nlme`, setting uncorrelated random effects at the level of the individual for A and k

```

vonbertwild.teeth.female <- nlme(teethwidth2 ~ A - (A - teethwidth)*exp(-k*timediff),
                                fixed=A+k~1, random= list(AnimalID = pdDiag(A + k ~ 1)),
                                data= femteethwild,
                                start=c(A=6.35,k=0.004),
                                na.action = na.omit)
summary(vonbertwild.teeth.female) # Asymptote = 6.25; k = 0.004604

```

Female teeth width in the wild: The Logistic interval equation

$$S_2 = A.S_1/(S_1 + (A - S_1).e^{-k.D})$$

```

logisticwild.teeth.female <- nlme(teethwidth2~
(A*teethwidth)/(teethwidth + (A - teethwidth)*exp(-k*timediff)),
                                fixed=A+k~1, random= list(AnimalID = pdDiag(A + k ~ 1)),
                                data= femteethwild,
                                start=c(A=6,k=0.005),
                                na.action = na.omit)
summary(logisticwild.teeth.female) # Asymptote = 6.18; k = 0.00678
AIC(vonbertwild.teeth.female, logisticwild.teeth.female)
# AIC vonbert = 433.62; AIC logistic = 483.67.

```

Simple AIC comparisons show that the von-Bertalanffy intervals equation does a far better job of modelling incremental change in incisor width in Damaraland mole-rats in the wild. We can plot the predicted model in terms of the rates of change, given that we have estimated how much an individual of size S_1 will increase in size over a given time period D , e.g. 180 days; this output then just needs scaling to recover growth/day.

```

# and can also plot the rates of change
femteethwild <- femteethwild %>%
  mutate(rate = teethchange/timediff)

plot(rate ~ teethwidth, data = femteethwild, pch = 16,
      col = adjustcolor("dodgerblue", alpha.f = 0.6), xlab = "Incisor width (mm)",
      ylab = "Rate of change in incisor width (mm/day)", bty = "l", las = 1)
axis(1, at = seq(2, 7, 0.5), labels = NA) ; axis(2, at = seq(0, 0.015, 0.0025), labels = NA)
abline(h = 0, lty =2, col = adjustcolor("grey", alpha.f = 0.6))

# get the normal von bertalanffy model and predict with these parameters
summary(vonbertwild.teeth.female)

```

```

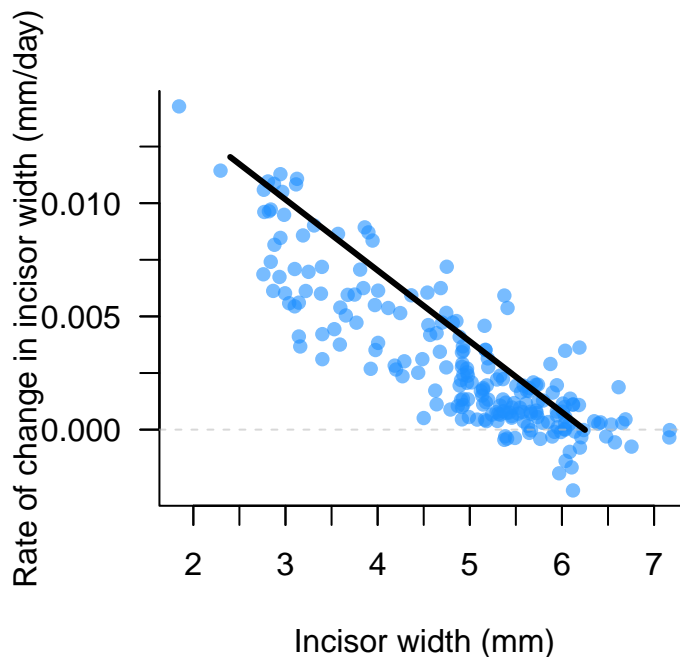
## Nonlinear mixed-effects model fit by maximum likelihood
## Model: teethwidth2 ~ A - (A - teethwidth) * exp(-k * timediff)
## Data: femteethwild
##      AIC      BIC    logLik
## 84.53913 101.2984 -37.26957
##
## Random effects:
## Formula: list(A ~ 1, k ~ 1)
## Level: AnimalID
## Structure: Diagonal
##           A           k Residual
## StdDev: 0.4494831 0.001132899 0.157768
##

```

```
## Fixed effects: A + k ~ 1
##      Value Std.Error DF   t-value p-value
## A 6.048223 0.05959104 86 101.49551    0
## k 0.004049 0.00024622 86  16.44526    0
## Correlation:
##   A
## k -0.559
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.55231176 -0.50162660 -0.04691431  0.45099647  1.93145739
##
## Number of Observations: 211
## Number of Groups: 124
```

```
vonbertratefunc <- function(x, A, k) {
  time2 = A - (A - x)*exp(-k*180)
  return(time2)
}

predfemteeth <- data.frame(time1 = seq(2.4, 6.25, 0.05),
  time2 = vonbertratefunc(x = seq(2.4, 6.25, 0.05), A = 6.246551, k = 0.004604)) %>%
  mutate(rate = (time2 - time1)/180)
points(rate ~ time1, data= predfemteeth, type = "l", lwd = 3, lty = 1)
```



We can see the the estimated asymptote in the wild is only slightly less than that in captivity. The same is true of the males. Thus one can use the models in the captive population and the wild to fully examine the ontogeny of size dimorphism in mole-rats.

All the above code was carried out in R v 3.5.2