

fishR Vignette - Von Bertalanffy Growth Model - Extra

Dr. Derek Ogle, Northland College

June 21, 2013

XXX THIS IS A WORK IN PROGRESS XXX

The von Bertalanffy growth model (VBGM) was introduced in a separate vignette. This vignette builds on the previous vignette but contains some extensions from the more “traditional” use of the VBGM.

The functions required to perform growth analyses in R are contained in the packages loaded below,

```
> library(FSA)
> library(FSAdata)
> library(nlstools)
```

1 Methods for Tag-Recapture Data

1.1 Fabens (1965)

In tag-recapture data the length at two times in the fish’s life is known, the amount of time between those two periods is known, but the age of the fish at those times is not known. [Fabens \(1965\)](#) provided a modification of the traditional VBGM (see [Appendix 6.1](#)) for the derivation) that can be used to estimate L_∞ and K (but not t_0) with this type of data. As with the traditional VBGM there are many parameterizations of Fabens modified model. The two most popular parameterizations appear to be,

$$L_r = L_m + (L_\infty - L_m) (1 - e^{-K\delta t}) \quad (1)$$

and

$$L_r - L_m = (L_\infty - L_m) (1 - e^{-K\delta t}) \quad (2)$$

where L_m is the length of the fish at the time of marking (t_m), L_r is the length of the fish at the time of recapture (t_r), δt is the “time-at-large” (i.e., $t_r - t_m$), and the other parameters are as defined previously (but see [Francis \(1988\)](#)). Model (1) is preferred over (2) for fitting because the left-hand-side contains only the response variable. Note also that this is a two-parameter (L_∞ and K) model.

Fabens’ method requires the length-at-marking, length-at-recapture, and the length of time between marking and recapture. As an example of this method, [Baker et al. \(1991\)](#) recorded these three variables for rainbow trout from the Kenai River. These data are loaded and viewed with,

```
> data(RBTroutKenai)
> str(RBTroutKenai)

'data.frame': 102 obs. of 3 variables:
 $ Lr: int  249 272 279 295 305 312 320 324 330 334 ...
 $ Lm: int  239 259 244 281 300 241 218 234 321 245 ...
 $ dt: num  0.175 0.06 0.211 0.093 0.06 ...

> head(RBTroutKenai)

  Lr Lm  dt
1 249 239 0.175
```

```

2 272 259 0.060
3 279 244 0.211
4 295 281 0.093
5 305 300 0.060
6 312 241 0.999

```

Fabens' model is a non-linear model; therefore, initial values of L_∞ and K must be supplied to `nls()`. Graphical procedures for finding initial values have not been developed for Faben's model. However, a simple starting value for L_∞ is the maximum length in the data frame,

```

> max(RBTroutKenai$Lr)
[1] 731

```

A reasonable starting value for K can be found with,

$$K_{init} = \frac{\log(\bar{L}_r) - \log(\bar{L}_m)}{t_r - t_m}$$

which is made more robust by averaging across all fish in the sample,

```

> with(RBTroutKenai, mean((log(Lr)-log(Lm))/dt))
[1] 0.2447

```

These initial values are then entered into a list as shown for the traditional VBGM. For example,

```

> Fabens.sv <- list(Linf=730,K=0.25)

```

The model is declared with

```

> fvb <- vbFuns("Fabens")

```

and fit and summarized with

```

> FVB1 <- nls(Lr ~ fvb(Lm,dt,Linf,K),start=Fabens.sv,data=RBTroutKenai)
> summary(FVB1,correlation=TRUE)

```

Formula: $L_r \sim \text{fvb}(L_m, dt, L_{inf}, K)$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
Linf	552.2012	20.3894	27.1	< 2e-16
K	0.4677	0.0584	8.0	2.2e-12

Residual standard error: 35.2 on 100 degrees of freedom

Correlation of Parameter Estimates:

Linf	
K	-0.84

Number of iterations to convergence: 11

Achieved convergence tolerance: 6.28e-06

The second version – using length increments – can be fit by first constructing the length increments,

```
> RBTroutKenai$dL <- RBTroutKenai$Lr-RBTroutKenai$Lm
```

and then fitting the model,

```
> fvb2 <- vbFuns("Fabens2")
> FVB2 <- nls(dL ~ fvb2(Lm,dt,Linf,K),start=Fabens.sv,data=RBTroutKenai)
> summary(FVB2,correlation=TRUE)
```

Formula: $dL \sim fvb2(Lm, dt, Linf, K)$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
Linf	552.2012	20.3894	27.1	< 2e-16
K	0.4677	0.0584	8.0	2.2e-12

Residual standard error: 35.2 on 100 degrees of freedom

Correlation of Parameter Estimates:

Linf	
K	-0.84

Number of iterations to convergence: 11

Achieved convergence tolerance: 6.27e-06

Fabens' model has been criticized or modified by many authors (see following sections).

2 Wang(1998)

Wang *et al.* (1995) noted that Fabens' method assumes that each fish follows the same growth curve and does not take into account individual growth variability and further noted that this characteristic caused Fabens' method to produce biased and inconsistent parameter estimates. Wang (1998) provided a solution to this problem. Specifically, he defined the model

$$L_r - L_m = [l_\infty + \beta (L_m - \bar{L}_m) - L_m] (1 - e^{-K\delta t})$$

where l_∞ is the mean L_∞ "over the recapture population" and β is a new parameter that XXXX. Wang (1998) proposed a slightly modified model of

$$L_r - L_m = (a + bL_m) (1 - e^{-K\delta t})$$

The first model is fit with,

```
> wvb <- vbFuns("Wang")
> Wang.sv <- list(Linf=730,K=0.25,b=0)
> WVB1 <- nls(dL ~ wvb(Lm,dt,Linf,K,b),start=Wang.sv,data=RBTroutKenai)
> summary(WVB1,correlation=TRUE)
```

Formula: $dL \sim wvb(Lm, dt, Linf, K, b)$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
Linf	458.587	14.349	32.0	<2e-16
K	1.713	0.660	2.6	0.011
b	0.664	0.064	10.4	<2e-16

Residual standard error: 30.9 on 99 degrees of freedom

Correlation of Parameter Estimates:

	Linf	K
K	-0.95	
b	-0.67	0.73

Number of iterations to convergence: 8

Achieved convergence tolerance: 3.89e-06

The second model is fit with,

```
> wvb2 <- vbFuns("Wang2")
> Wang2.sv <- list(K=0.25,a=200,d=1)
> WVB2 <- nls(dL ~ wvb2(Lm,dt,K,a,d),start=Wang2.sv,data=RBTroutKenai)
> summary(WVB2,correlation=TRUE)
```

Formula: dL ~ wvb2(Lm, dt, K, a, d)

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
K	1.713	0.660	2.60	0.011
a	213.910	34.898	6.13	1.8e-08
d	-0.336	0.064	-5.24	9.1e-07

Residual standard error: 30.9 on 99 degrees of freedom

Correlation of Parameter Estimates:

	K	a
a	-0.88	
d	0.73	-0.95

Number of iterations to convergence: 8

Achieved convergence tolerance: 3.23e-06

XXX I have not yet been able to verify these calculations XXX

2.1 Francis(1988)

Francis (1988) noted that L_∞ and K do not have the same meanings in Fabens' and the traditional VBGM. In addition, he noted the very high correlation between these two parameters in Fabens' model. To address these issues he suggested the following parameterization,

$$L_r = L_m + \frac{\frac{\beta g_\alpha - \alpha g_\beta}{(g_\alpha - g_\beta) - L_m}}{1 - \left(1 + \frac{g_\alpha - g_\beta}{\alpha - \beta}\right) \delta t}$$

where g_α and g_β are parameters that represent the mean growth rate at the arbitrary ages α and β .

XXX MORE HERE XXX

2.2 Baker *et al.* (1991)

Finally, [Baker *et al.* \(1991\)](#) provide modifications of Fabens method that produce analogues to Schnute's four-parameter growth model.

XXX MORE HERE XXX

3 Modeling Seasonal Growth Oscillations

3.1 Background

Numerous models to describe the seasonal growth of organisms have been proposed. Nearly all these models are a modification of the traditional VBGM that allows for seasonal oscillations in length within each growth year. The model originally proposed by Hoenig and Hanumara in an unpublished report (Hoenig and Hanumara 1990; Pauly *et al.* 1992), but most often cited from its description in Somers (1988), is highly popular. Following other authors (e.g. Pauly (1988)), this model will be referred to as Somers' model throughout this vignette.

Pitcher and MacDonald (1973) and Pauly and Gaschutz (1979) modified the traditional VBGM to incorporate seasonal growth oscillations by including a sine function with a period of one year. However, Somers (1988) showed that these modifications produced a model that only fulfilled the definition of t_0 (i.e., when $t = t_0$, $L = 0$) under a very strict circumstance (i.e., the seasonal oscillations began at $t = t_0$). Thus, Somers (1988) proposed the following formula for modeling seasonal growth that rectified this situation¹:

$$E[L|t] = L_\infty \left(1 - e^{-K(t-t_0)-S(t)+S(t_0)} \right) \quad (3)$$

with

$$S(t) = \frac{CK}{2\pi} \sin(2\pi(t - t_s))$$

where $E[L|t]$ is the expected or average length at time (or age) t ; C modulates the amplitude of the growth oscillations and corresponds to the proportion of decrease in growth at the depth of the oscillation (i.e., “winter”); t_s is the time between time 0 and the start of the convex portion of the first sinusoidal growth oscillation, and L_∞ , K , and t_0 are as defined above.

This model has two more parameters than the traditional VBGM. These parameters warrant further discussion. If $C = 0$, then there is no seasonal oscillation and the model reduces to the traditional VBGM (Figure 1). If $C = 1$, then growth completely stops once a year at the “winter-point” (WP ; Figure 1). Values of $0 < C < 1$ result in reduced, but not stopped, growth during the winter (Figure 1). Values of $C > 1$ (or 0) allow seasonal decreases in average length-at-age, and might not seem realistic in organisms whose skeletons largely preclude shrinkage (Pauly *et al.* 1992), but could result from size-dependent overwinter mortality.

The t_s is harder to visualize because it is the very start of a convex oscillation on a curve (the traditional von Bertalanffy growth model) that is already convex (Figure 2). Thus, many authors refer to the “winter point”, where growth is slowest and is the “bottom” of the seasonal oscillation (Figure 2). The “winter point” is half-way between the starts of two consecutive seasonal oscillations; thus, $WP = t_s + 0.5$ (Figure 2). If interest is fully in WP and not t_s , then the model of Somers (1988) can be modified by substituting $t_s = WP - 0.5$ into S_t , or

$$E[L|t] = L_\infty \left(1 - e^{-K(t-t_0)-R(t)+R(t_0)} \right) \quad (4)$$

with

$$R(t) = \frac{CK}{2\pi} \sin(2\pi(t - WP + 0.5))$$

¹Note the warning expressed in Garcia-Berthou *et al.* (2012) about assuring that the correct formula for this equation is used in all analyses.

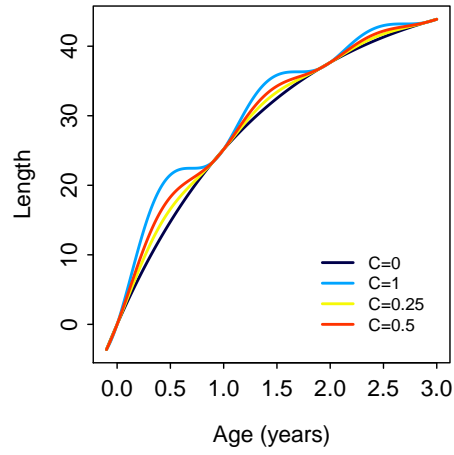


Figure 1. Hypothetical growth trajectories from age -0.1 to 3 for Somers' model using $L_\infty=50$, $K=0.7$, $t_0=0$, $t_s=0.2$, and various values of C .

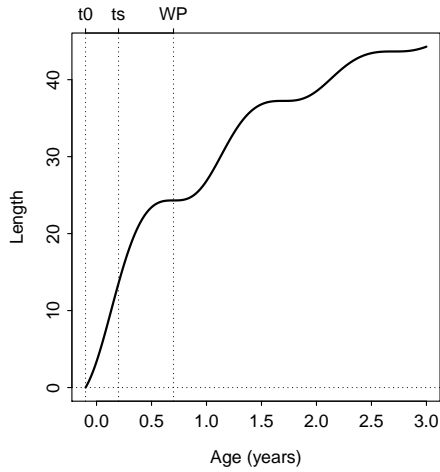


Figure 2. Hypothetical growth trajectory from age -0.1 to 3 for Somers' model using $L_\infty=50$, $K=0.7$, $t_0=-0.1$, $t_s=0.2$, and $C = 0$. Vertical dotted lines are placed at t_0 , t_s , and $WP=t_s + 0.5$ and a horizontal dotted line at $L_t = 0$ to aid understanding.

3.2 Fitting Models in R

Fitting Somer's model will be illustrated with length-at-age data for Chilean anchoveta (*Engraulis ringens*) extracted approximately from Figure 9 in Cubillos *et al.* (2001). The data are loaded and their structured is viewed with

```
> data(AnchovetaChile)
> str(AnchovetaChile)
'data.frame': 207 obs. of 3 variables:
 $ age.mon: int  42 44 51 52 53 42 43 44 45 46 ...
 $ tl.cm  : num  17.1 18 18 18.2 18.3 ...
 $ cohort : int  1987 1987 1987 1987 1987 1988 1988 1988 1988 1988 ...
> head(AnchovetaChile)
  age.mon tl.cm cohort
1     42 17.12  1987
2     44 18.01  1987
3     51 17.98  1987
4     52 18.15  1987
5     53 18.31  1987
6     42 16.85  1988
```

The age is recorded in months. A new variable of age in years is created with

```
> AnchovetaChile$age.yrs <- AnchovetaChile$age.mon/12
```

As with the traditional VBGM, Somers' model requires starting values for each parameter. Starting values for L_∞ , K , and t_0 can be obtained as described in the vignette for the VBGM. The range of possible values for both t_s and C is from 0 to 1. Starting values near 0.5 for t_s and nearer to 1 for C are likely adequate for most species. Starting values can be found with `vbStarts` using a formula of the form `length~age` as the first argument, the data frame where the two variables are found in the `data=` argument, and `type="Somers"`. Starting values for the Chilean anchoveta are found with

```
> svbso <- vbStarts(tl.cm~age.yrs,data=AnchovetaChile,type="Somers")
> unlist(svbso) # unlist used for display purposes only
      Linf      K      t0      C      ts
24.27855 0.02256 -0.55775 0.90000 0.10000
```

This Somers' model is a more complicated formula than the traditional VBGM function. Thus, the simplest way to enter the Somers' model function is to use `vbFuns()` with `"Somers"` as the only argument. For example,

```
> ( vbso <- vbFuns("Somers") )
function (t, Linf, K, t0, C, ts)
{
  if (length(Linf) > 1) {
    K <- Linf[2]
    t0 <- Linf[3]
    C <- Linf[4]
    ts <- Linf[5]
    Linf <- Linf[1]
  }
  St <- (C * K)/(2 * pi) * sin(2 * pi * (t - ts))
  Sto <- (C * K)/(2 * pi) * sin(2 * pi * (t0 - ts))
}
```



```

      Linf * (1 - exp(-K * (t - t0) - St + Sto))
}
<environment: 0x0a2cdea0>

```

The starting values and model function are then used with `nls()` to find the best fit model. The results of `nls()` must be saved to an object. That object is then submitted to `overview()` from the `nlstools` package to extract model parameters and other summary information. For example,

```

> vbso1 <- nls(tl.cm~vbso(age.yrs,Linf,K,t0,C,ts),data=AnchovetaChile,start=svbso)
> overview(vbso1)

```

```

-----
Formula: tl.cm ~ vbso(age.yrs, Linf, K, t0, C, ts)

Parameters:
      Estimate Std. Error t value Pr(>|t|)
Linf  21.9591     1.0055   21.84 < 2e-16
K      0.3673     0.0426    8.63 1.8e-15
t0    -0.6044     0.0561  -10.77 < 2e-16
C      0.9549     0.1573    6.07 6.2e-09
ts     0.2898     0.0261   11.09 < 2e-16

Residual standard error: 0.862 on 202 degrees of freedom

Number of iterations to convergence: 18
Achieved convergence tolerance: 6.19e-06

-----
Residual sum of squares: 150

-----
Asymptotic confidence interval:
      2.5%  97.5%
Linf 19.9765 23.9418
K     0.2834 0.4512
t0   -0.7150 -0.4937
C     0.6448 1.2651
ts    0.2383 0.3413

-----
Correlation matrix:
      Linf      K      t0      C      ts
Linf  1.00000 -0.97997 -0.8089 -0.05828 0.06224
K     -0.97997 1.00000  0.8914  0.03324 -0.07673
t0    -0.80891 0.89137  1.0000 -0.17804 0.10822
C     -0.05828 0.03324 -0.1780 1.00000  0.01997
ts     0.06224 -0.07673 0.1082  0.01997 1.00000

```

A fitted line plot (Figure 3) can be obtained (manually) with

```

> plot(tl.cm~age.yrs,data=AnchovetaChile,pch=19,xlab="Age (years)",ylab="Total Length (cm)")
> curve(vbso(x,coef(vbso1)),from=0,to=5,col="red",lwd=2,add=TRUE)

```

The modified version of Somers' model used to estimate the "winter point" directly can be fit very similarly as shown with (note use of "Somers2" and *WP* in the model function)

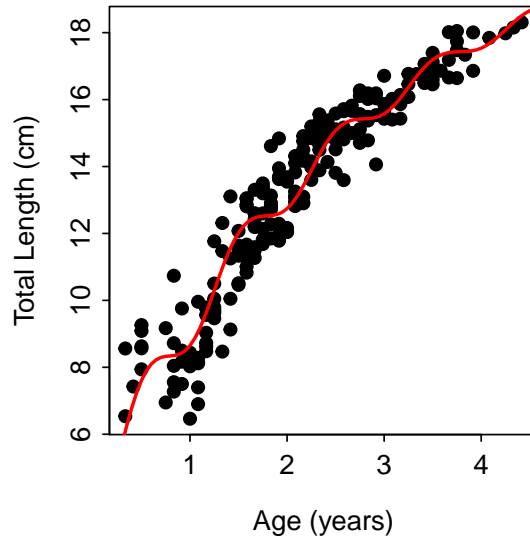


Figure 3. Fitted line plot from the seasonally oscillating VBGM fit to the Chilean anchoveta data.

```

> svbso2 <- vbStarts(tl.cm~age.yrs,data=AnchovetaChile,type="Somers2")
> unlist(svbso2) # unlist used for display purposes only
      Linf      K      t0      C      WP
24.27855 0.02256 -0.55775 0.90000 0.90000
> ( vbso2 <- vbFuns("Somers2") )
function (t, Linf, K, t0, C, WP)
{
  if (length(Linf) > 1) {
    K <- Linf[2]
    t0 <- Linf[3]
    C <- Linf[4]
    WP <- Linf[5]
    Linf <- Linf[1]
  }
  Rt <- (C * K)/(2 * pi) * sin(2 * pi * (t - WP + 0.5))
  Rto <- (C * K)/(2 * pi) * sin(2 * pi * (t0 - WP + 0.5))
  Linf * (1 - exp(-K * (t - t0) - Rt + Rto))
}
<environment: 0x09ce3308>
> vbso2 <- nls(tl.cm~vbso2(age.yrs,Linf,K,t0,C,WP),data=AnchovetaChile,start=svbso2)
> overview(vbso2)

-----
Formula: tl.cm ~ vbso2(age.yrs, Linf, K, t0, C, WP)

Parameters:
      Estimate Std. Error t value Pr(>|t|)
Linf  21.9592     1.0055   21.84 < 2e-16
K      0.3673     0.0426    8.63 1.8e-15
t0    -0.6044     0.0561  -10.77 < 2e-16
C      0.9549     0.1573    6.07 6.2e-09
WP     0.7898     0.0261   30.24 < 2e-16

```

Residual standard error: 0.862 on 202 degrees of freedom

Number of iterations to convergence: 19
Achieved convergence tolerance: 7.67e-06

Residual sum of squares: 150

Asymptotic confidence interval:

	2.5%	97.5%
Linf	19.9765	23.9419
K	0.2834	0.4512
t0	-0.7150	-0.4937
C	0.6448	1.2651
WP	0.7383	0.8413

Correlation matrix:

	Linf	K	t0	C	WP
Linf	1.00000	-0.97997	-0.8089	-0.05828	0.06224
K	-0.97997	1.00000	0.8914	0.03324	-0.07673
t0	-0.80891	0.89138	1.0000	-0.17804	0.10823
C	-0.05828	0.03324	-0.1780	1.00000	0.01997
WP	0.06224	-0.07673	0.1082	0.01997	1.00000

4 Double von Bertalanffy

Vaughan and Helser (1990) introduced and Porch *et al.* (2002) explored the so-called “double” VBGM that allows the fitting of one Beverton-Holt VBGM to fish less than some “critical” age (t_c) and a different Beverton-Holt VBGM to fish greater than that critical age. The model is parameterized as follows,

$$E[L|t] = \begin{cases} L_\infty * (1 - e^{-K_1(t-t_1)}) & \text{if } t < t_c \\ L_\infty * (1 - e^{-K_2(t-t_2)}) & \text{if } t \geq t_c \end{cases} \quad (5)$$

where L , t , and L_∞ are as defined previously, K_1 and K_2 are Brody growth rate coefficients for the two groups, t_1 and t_2 are the ages when the average size is zero for each group, and

$$t_c = \frac{K_2 t_2 - K_1 t_1}{K_2 - K_1}$$

This model is difficult to fit because it is essentially a piecewise model where the critical point of switching between the two pieces depends on parameters in the model. Thus, the model effectively contains an “if” statement. Fitting this form of model is difficult with `nls()` because `nls()` depends on derivatives of the function, which cannot be taken with an “if” component to the model. Fortunately, other algorithms in R – specifically, `optim()` – can be used to find the minimum of a function without requiring the calculation of derivatives. Unfortunately, `optim()` requires a manual setup of the function to be minimized.

The process of using `optim()` to fit a VBGM is illustrated below, first by using `optim()` to fit the typical VBGM and then showing how to use it for fitting the double VBGM. Before showing these methods, note that hypothetical data that follows a the double VBGM with $L_\infty=40$, $K_1=0.412$, $K_2=0.114$, $t_1=0.053$, $t_2=-8.41$ parameters, resulting in , $t_c=3.291$, was created and stored in the `df` data frame. A view of these data is shown below².

```
> head(df)
      len age
1  9.739  1
2 12.913  1
3 14.542  1
4 10.923  1
5 17.705  1
6 12.495  1
```

The traditional Beverton-Holt VBGM is fit to these data with `nls()`³

```
> svbs <- list(Linf=40,K=0.20,to=0)
> mdl <- len~Linf*(1-exp(-K*(age-to)))
> fit.nls <- nls(mdl,data=df,start=svbs)
> ( c.nls <- coef(fit.nls) )
      Linf      K      to
37.3595  0.3630 -0.3064
```

Then, for comparative purposes, the same model is fit to the same data using `optim()`. To use `optim()`, a function must first be created that returns the predicted lengths from the model given a current set of parameters (i.e., these parameters will change during the iterations required for fitting the model) and the observed ages. Such a function is shown below where the first argument is a vector of parameter values⁴ and the second argument is vector of the observed ages.

²The algorithm for creating these data can be seen by examining the R code script found in last section of this vignette.

³Note that the starting values were found graphically with `growthModelSim("vbTypical",len age,data=df)`.

⁴Note that this function is hard-wired such that the first parameter must be L_∞ , the second must be K , and the third must be t_0 .

```
> sVonB <- function(params,age) {
  Linf <- params[1]; K <- params[2]; to <- params[3]
  Linf*(1-exp(-K*(age-to)))
}
```

The `optim()` function also requires a function that computes the residual sum-of-squares between the observed lengths and the predicted lengths. Such a function is shown below where the first argument is again the vector of parameter estimates and the second and third arguments are the vectors of observed lengths and ages, respectively. Note how this function uses the function previously constructed to find predicted lengths.

```
> sRSS <- function(params,len,age) { sum((len-sVonB(params,age))^2) }
```

Finally, `optim()` can be run by including the initial parameter guesses in the first argument, the RSS function that should be minimized as the second argument, the vectors of observed lengths and ages as the third and fourth arguments, and two arguments (`hessian=TRUE` and `method="BFGS"`) that control how the `optim()` routine will behave). The results should be assigned to an object as illustrated below.

```
> fit.o <- optim(svbs,sRSS,len=df$len,age=df$age,hessian=TRUE,method="BFGS")
```

The estimated parameters are extracted from the `par` portion of the object returned by `optim()` as follows

```
> fit.o$par
  Linf      K      to
37.3593 0.3630 -0.3063
```

From this, it is evident that the parameter estimates are slightly, but imperceptibly, different between the `optim()` and `nls()` results (Figure 4).

Using a similar process the double VBGM can fit be fit to the hypothetical data using `optim()`. So that the value of t_c can be easily computed later it is useful to first construct a function to compute t_c . The function below performs this calculation using only a vector of parameter values as an argument⁵.

```
> tc <- function(params) {
  K1 <- params[2]; t1 <- params[3]; K2 <- params[4]; t2 <- params[5]
  (K2*t2-K1*t1)/(K2-K1)
}
```

The function to constructed predicted lengths from double VBGM equation is then constructed as shown below. In this function, note the use of `ifelse()` to calculate the two “pieces” of the double VBGM.

```
> dVonB <- function(params,age) {
  Linf <- params[1]; K1 <- params[2]; t1 <- params[3]; K2 <- params[4]; t2 <- params[5]
  ifelse(age<tc(params),Linf*(1-exp(-K1*(age-t1))), Linf*(1-exp(-K2*(age-t2))))
}
```

The function to compute the sum of squared residuals is a simple modification of the previously used function, i.e.,

⁵Note that this function is hard-wired such that the parameter values must be in the order of L_∞ , K_1 , K_2 , t_1 , and t_2 .

```
> dRSS <- function(params,len,age) { sum((len-dVonB(params,age))^2) }
```

Finally, `optim()` is used as described previously and shown below⁶ The final model fit is shown in (Figure 4).

```
> dvbs <- c(37,0.36,-0.3,0.26,-3.3)
> fit.d <- optim(dvbs,dRSS,len=df$len,age=df$age,hessian=TRUE,method="BFGS")
> fit.d$par
[1] 40.1346  0.4017  0.0198  0.1114 -8.6753
> tc(fit.d$par)
[1] 3.358
```

⁶Note that there is not an automatic way to construct the starting values.

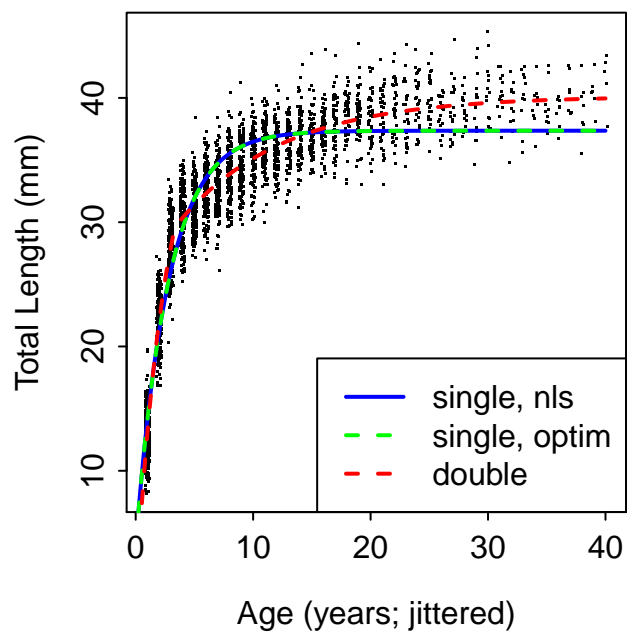


Figure 4. Example of the fit of a double VBGM, (5), fit to RANDOM data.

5 Model of Porch et al. (2002)

WORKING ON THIS

```
> df$age1 <- df$age + runif(length(df$age),0.3,07)
>
>
> beta1 <- function(params,age) {
>   K1 <- params[4]; lambda1 <- params[5]; to <- params[3]
>   K1/lambda1*(exp(-lambda1*age)-exp(-lambda1*to))
> }
>
> beta2 <- function(params,age) {
>   K1 <- params[4]; lambda1 <- params[5]; to <- params[3]
>   K2 <- params[6]; lambda2 <- params[7]; ts <- params[8]
>   (K2/(4*pi^2+lambda2^2))*(exp(-lambda2*age)*(2*pi*cos(2*pi*(ts-age))-lambda2*sin(2*pi*(ts-age)))-
> }
>
> porchVB <- function(params,age) {
>   Linf <- params[1]; K0 <- params[2]; to <- params[3]
>   Linf*(1-exp(beta1(params,age)+beta2(params,age)-K0*(age-to)))
> }
>
> porchRSS <- function(params,age,len) { sum((len-porchVB(params,age))^2) }
>
> pvbs <- c(43.4,0.0475,0.443,0.695,0.476,0.301,0.344,0.439)
> fit.p <- optim(pvbs,porchRSS,len=df$len,age=df$age1,hessian=TRUE,method="BFGS")
> fit.p$par
>
> pred.p <- porchVB(fit.p$par,p.ages)
> plot(len~age1,data=df,xlab="Age (years)",ylab="Total Length (mm)",pch=".")
> lines(p.ages,pred.p,col="blue",lwd=2,lty=1)
```


References

- Baker, T., R. Lafferty, and T. Q. II. 1991. A general growth model for mark-recapture data. *Fisheries Research* 11:257–281. [1](#), [5](#)
- Cubillos, L., D. Arcos, D. Bucarey, and M. Canales. 2001. Seasonal growth of small pelagic fish off Talcahuano, Chile (37°S, 73°W): A consequence of their reproductive strategy to seasonal upwelling? *Aquatic Living Resources* 14:115–124. [8](#)
- Fabens, A. 1965. Properties and fitting of the von Bertalanffy growth curve. *Growth* 29:265–289. [1](#)
- Francis, R. 1988. Are growth parameters estimated from tagging and age-length data comparable? *Canadian Journal of Fisheries and Aquatic Sciences* 45:936–942. [1](#), [4](#)
- Garcia-Berthou, E., G. Carmona-Catot, R. Merciai, and D. H. Ogle. 2012. A technical note on seasonal growth models. *Reviews in Fish Biology and Fisheries* 22:635–640. [6](#)
- Hoenig, N. and R. Hanumara. 1990. An empirical comparison of seasonal growth models. *Fishbyte* 8:32–34. [6](#)
- Pauly, D. 1988. Beyond our original horizons: The tropicalization of Beverton and Holt. *Reviews in Fish Biology and Fisheries* 8:307–334. [6](#)
- Pauly, D. and G. Gaschutz. 1979. A simple method for fitting oscillating length growth data, with a program for pocket calculators, ices c.m. I.C.E.S. CM . Demersal Fish Committee 1979/6:24:1–26. [6](#)
- Pauly, D., M. Soriano-Bartz, J. Moreau, and A. Jarre-Teichmann. 1992. A new model accounting for seasonal cessation of growth in fishes. *Marine and Freshwater Research* 43:1151–1156. [6](#)
- Pitcher, T. and P. MacDonald. 1973. Two models for seasonal growth in fishes. *Journal of Applied Ecology* 10:599–606. [6](#)
- Porch, C. E., C. A. Wilson, and D. L. Nieland. 2002. A new growth model for red drum (*Sciaenops ocellatus*) that accomodates seasonal and ontogenic changes in growth rates. *Fisheries Bulletin* 100:149–152. [12](#)
- Somers, I. F. 1988. On a seasonally oscillating growth function. *Fishbyte* 6(1):8–11. [6](#)
- Vaughan, D. S. and T. E. Helser. 1990. Status of the red drum stock of the atlantic coast: stock assessment report for 1989. NOAA Technical Memorandum NMFS-SEFC-263, U.S. Department of Commerce. [12](#)
- Wang, Y.-G. 1998. An improved Fabens method for estimation of growth parameters in the von Bertalanffy model with individual asymptotes. *Canadian Journal of Fisheries and Aquatic Sciences* 55:397–400. [3](#)
- Wang, Y.-G., M. Thomas, and I. Somers. 1995. A maximum likelihood approach for estimating growth from tag-recapture data. *Canadian Journal of Fisheries and Aquatic Sciences* 52:252–259. [3](#)

6 Appendix

6.1 Derivation of Fabens (1965) Increment Model

First Version

Define the length at time of marking (t_m) as

$$L_m = L_\infty \left(1 - e^{-K(t_m - t_0)}\right)$$

which is immediately rewritten as,

$$\begin{aligned}\frac{L_m}{L_\infty} &= 1 - e^{-K(t_m - t_0)} \\ \frac{L_m}{L_\infty} - 1 &= -e^{-K(t_m - t_0)} \\ 1 - \frac{L_m}{L_\infty} &= e^{-K(t_m - t_0)} \\ \frac{L_\infty - L_m}{L_\infty} &= e^{-K(t_m - t_0)}\end{aligned}\tag{6}$$

Similarly, the length at time of recapture (t_r) is

$$L_r = L_\infty \left(1 - e^{-K(t_r - t_0)}\right)\tag{7}$$

and define δt as “time-at-large” such that

$$t_r = t_m + \delta t\tag{8}$$

Substitute (8) into (7) for t_r and simplify,

$$\begin{aligned}L_r &= L_\infty \left(1 - e^{-K(t_m + \delta t - t_0)}\right) \\ L_r &= L_\infty \left(1 - e^{-K(t_m - t_0)} e^{-K\delta t}\right)\end{aligned}$$

and now substitute in the last expression in (6) and simplify to get,

$$\begin{aligned}L_r &= L_\infty \left(1 - \frac{L_\infty - L_m}{L_\infty} e^{-K\delta t}\right) \\ L_r &= L_\infty - (L_\infty - L_m) e^{-K\delta t}\end{aligned}$$

Some authors will further modify this (note the addition of $L_m - L_m$) to get,

$$\begin{aligned}L_r &= L_m + L_\infty - L_m - (L_\infty - L_m) e^{-K\delta t} \\ L_r &= L_m + (L_\infty - L_m) - (L_\infty - L_m) e^{-K\delta t} \\ L_r &= L_m + (L_\infty - L_m) (1 - e^{-K\delta t})\end{aligned}$$

Finally, other authors will present this model as a change in length,

$$L_r - L_m = (L_\infty - L_m) (1 - e^{-K\delta t})$$

Second Version

Alternatively, one can start the derivation by thinking of the increment in lengths

$$\begin{aligned} L_r - L_m &= L_\infty \left(1 - e^{-K(t_m + \delta t - t_0)}\right) - L_\infty \left(1 - e^{-K(t_m - t_0)}\right) \\ L_r - L_m &= L_\infty - L_\infty e^{-K(t_m + \delta t - t_0)} - L_\infty + L_\infty e^{-K(t_m - t_0)} \\ L_r - L_m &= L_\infty e^{-K(t_m - t_0)} - L_\infty e^{-K(t_m + \delta t - t_0)} \\ L_r - L_m &= L_\infty e^{-K(t_m - t_0)} (1 - e^{-K\delta t}) \end{aligned}$$

Now, note that $L_m - L_\infty = -L_\infty e^{-K(t_m - t_0)}$ or, more usefully, $L_\infty - L_m = L_\infty e^{-K(t_m - t_0)}$ which can be substituted into the above to get

$$L_r - L_m = (L_\infty - L_m) (1 - e^{-K\delta t})$$

6.2 More Algebra Related to the Fabens (1965) Model

Understanding how the Faben's model "works" is sometimes difficult to visualize. However, some algebra on the increment model may help in this regard. Begin with,

$$L_r - L_m = (L_\infty - L_m) (1 - e^{-K\delta t})$$

which is easily modified to

$$\frac{L_r - L_m}{L_\infty - L_m} = 1 - e^{-K\delta t}$$

and then to

$$\frac{L_\infty - L_r}{L_\infty - L_m} = e^{-K\delta t} \tag{9}$$

which shows that Fabens model examines the relative "distances" of L_r and L_m from L_∞ and models that ratio as an exponential decay model. This can be explored with hypothetical data. The predicted lengths at six ages were computed supposing that $L_\infty=400$, $K=0.5$, and $t_0=0$. All combinations of two of these lengths were then taken as a mark and recapture pair, such that the "time-at-large" was computed as the difference in ages and the growth increment was computed as the difference in predicted lengths. The results are shown in Table 1 and Figure 5.

Thus, for example if the "time-at-large" (δt) is "small" (near zero) then the length-at-marking and the length-at-recapture should be approximately equal and both the LHS and the RHS of (9) will be approximately equal to 1. However, as the "time-at-large" increases the RHS will be smaller and L_r will be closer to L_∞ than L_m making the LHS also smaller.

Table 1. Results of several scenarios computed for hypothetical mark-recapture data from hypothetical von Bertalanffy data.

Scenario	Age_r	Age_m	δt	L_r	L_m	δL	$\frac{L_\infty - L_r}{L_\infty - L_m}$
1	1.3	1.0	0.3	191.2	157.4	33.8	0.861
2	2.0	1.0	1.0	252.8	157.4	95.5	0.607
3	5.0	1.0	4.0	367.2	157.4	209.8	0.135
4	13.5	1.0	12.5	399.5	157.4	242.1	0.002
5	14.0	1.0	13.0	399.6	157.4	242.2	0.002
6	2.0	1.3	0.7	252.8	191.2	61.7	0.705
7	5.0	1.3	3.7	367.2	191.2	176.0	0.157
8	13.5	1.3	12.2	399.5	191.2	208.3	0.002
9	14.0	1.3	12.7	399.6	191.2	208.5	0.002
10	5.0	2.0	3.0	367.2	252.8	114.3	0.223
11	13.5	2.0	11.5	399.5	252.8	146.7	0.003
12	14.0	2.0	12.0	399.6	252.8	146.8	0.002
13	13.5	5.0	8.5	399.5	367.2	32.4	0.014
14	14.0	5.0	9.0	399.6	367.2	32.5	0.011
15	14.0	13.5	0.5	399.6	399.5	0.1	0.779

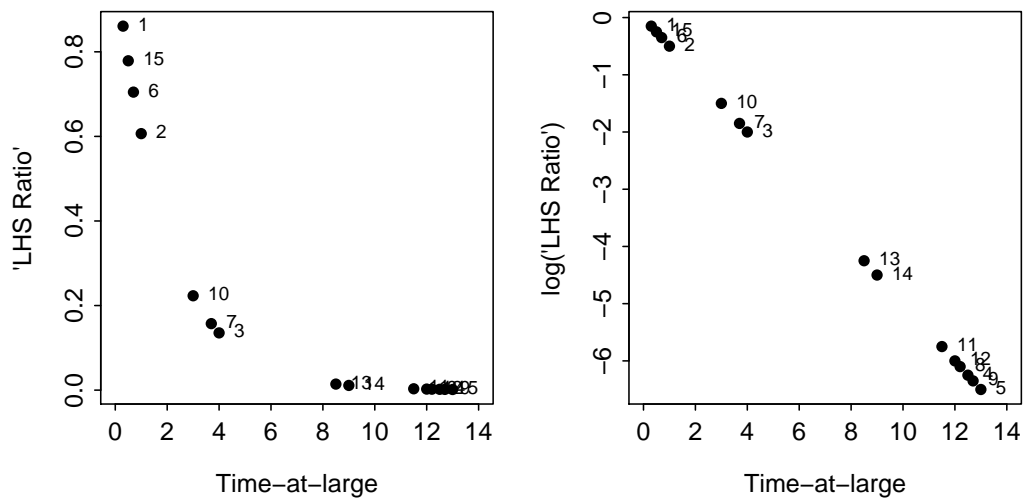


Figure 5. Plots of the ratio (LEFT) and log ratio (RIGHT) on the LHS of (9) versus time-at-large (δt) for data from a hypothetical von Bertalanffy growth model. The numbers correspond to the “scenario” shown in Table 1.

Reproducibility Information

Version Information

- **Compiled Date:** Fri Jun 21 2013
- **Compiled Time:** 6:57:06 PM
- **Code Execution Time:** 4 s

R Information

- **R Version:** R version 3.0.0 (2013-04-03)
- **System:** Windows, i386-w64-mingw32/i386 (32-bit)
- **Base Packages:** base, datasets, graphics, grDevices, methods, stats, tcltk, utils
- **Other Packages:** FSA_0.3.5, FSAdat0.1.3, gdata_2.12.0.2, knitr_1.2, nlstools_0.0-14, plyr_1.8, reshape_0.8.4, xtable_1.7-1
- **Loaded-Only Packages:** car_2.0-16, cluster_1.14.4, digest_0.6.3, evaluate_0.4.3, formatR_0.7, gplots_2.11.0.1, grid_3.0.0, gtools_2.7.1, Hmisc_3.10-1, lattice_0.20-15, multcomp_1.2-17, nlme_3.1-109, plotrix_3.4-7, quantreg_4.98, relax_1.3.13, sciplot_1.1-0, SparseM_0.99, stringr_0.6.2, TeachingDemos_2.9, tools_3.0.0
- **Required Packages:** FSA, FSAdat0, nlstools and their dependencies (car, gdata, gplots, Hmisc, knitr, multcomp, nlme, plotrix, quantreg, relax, reshape, sciplot, stats, tcltk, TeachingDemos)