Simulation and Parameter Estimation for C_3 photosynthesis

Fernando E. Miguez
Energy Biosciences Institute
University of Illinois

April 20, 2010

Abstract

Simulation of C_3 photosynthesis and estimation of parameters.

1 Simulating Photosynthesis

The package has only two functions at the time. The first one of interest is c3photo. Let us see what the arguments for this function are

```
> args(c3photo)

function (Qp, Tl, RH, vcmax = 100, jmax = 180, Rd = 1.1, Catm = 380,
    O2 = 210, b0 = 0.08, b1 = 5, theta = 0.7)
NULL
```

Qp is the quantum flux, Tl is the temperature of the leaf, RH is the relative humidity, vcmax is the maximum rate of carboxylation, jmax is the maximum rate of electron transport, Rd is the dark respiration, Catm is the atmospheric CO_2 concentration, O2 is the atmospheric oxygen concentration, b0 is the intercept of the Ball-Berry model, b1 is the slope of the Ball-Berry model for stomatal conductance and theta is the curvature parameter for the light response. For more information see the function documentation (i.e. ?c3photo).

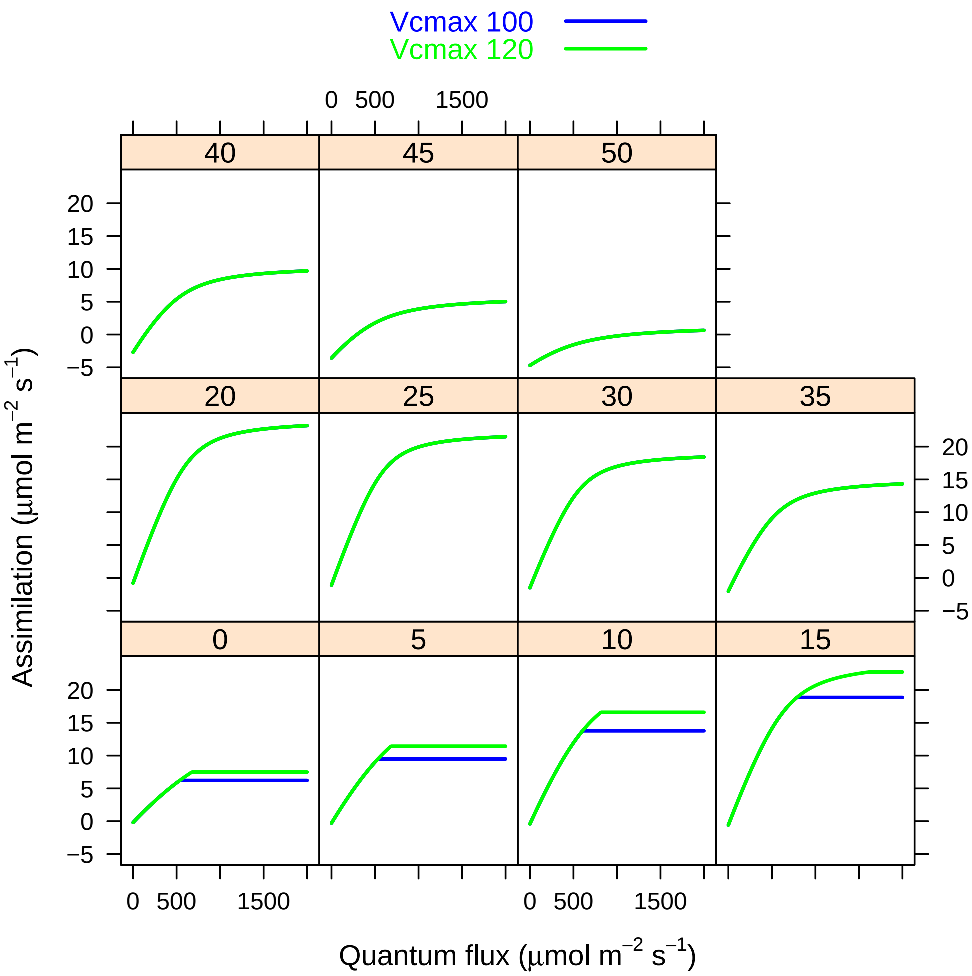

Figure 1: Assimilation response to light levels for different temperatures (in Celsius). Each panel is a different level of temperature. The two lines within a panel show different values for $Vcmax$.

2 Optimizing Parameters for a single A/Ci curve

The other function of interest is `Opc3photo`.

```
> args(Opc3photo)

function (data, ivcmax = 100, ijmax = 180, iRd = 1.1, Catm = 380,
    O2 = 210, ib0 = 0.08, ib1 = 9.58, itheta = 0.7, op.level = 1,
    op.method = c("optim", "nlminb"), response = c("Assim", "StomCond"),
    level = 0.95, hessian = TRUE, curve.kind = c("Ci", "Q"),
    op.ci = FALSE, ...)
NULL
```

The `data` argument should be the observed assimilation data. One example is the built-in dataset `simA100`.

```
> data(simA100)
> head(simA100)

         Assim   Qp Tl  RH        Ci    Vcmax     Jmax       Rd Catm
[1,] -0.6944929 1500 25 0.7  4.653977 90.80678 205.9745 2.311849    0
[2,] -0.6944929 1500 25 0.7  9.653877 90.80678 205.9745 2.311849    5
[3,] -0.6944929 1500 25 0.7 14.653877 90.80678 205.9745 2.311849   10
[4,] -0.6944929 1500 25 0.7 24.653877 90.80678 205.9745 2.311849   20
[5,] -0.6944929 1500 25 0.7 34.653877 90.80678 205.9745 2.311849   30
[6,]  0.4378667 1500 25 0.7 52.389158 90.80678 205.9745 2.311849   50
```

The dataset contains more than is needed to run `Opc3photo`. We know that this dataset was simulated and that the 'true' values for $Vcmax$, $Jmax$, and Rd are 90.8, 206, and 2.31 respectively. Can we recover them from the data alone?

```
> Opc3photo(simA100[, 1:5], Catm = simA100[, 9], curve.kind = "Ci",
+     op.level = 2)

Optimization of C3 photosynthesis

                95 %   Conf Int
         best  lower   upper
Vmax    90.66   90.1    91.2
Jmax   205.67  204.8   206.5
Rd       0.65    0.6     0.7

 Corr  Vmax and Jmax: 0.2635003

 Resid Sums Sq: 0.083576

Convergence:YES
```

This is fabricated data and the function works even if the variance seems to be close zero. We can try a slower, less accurate method first to get starting values as well. And suppress the computation of the hessian.

```
> op100 <- Opc3photo(simA100[, 1:5], Catm = simA100[, 9], method = "SANN",
+     hessian = FALSE, curve.kind = "Ci", op.level = 2)
```

now we can use this values as starting values. If we do not specify the optimization method it will use the default used by the `optim` function which is "Nelder-Mead" (see `?optim`).

```
> op100 <- Opc3photo(simA100[, 1:5], Catm = simA100[, 9], ivcmax = op100$bestVmax,
+     ijmax = op100$bestJmax, iRd = op100$bestRd, curve.kind = "Ci",
+     op.level = 2)
> op100
```

```
Optimization of C3 photosynthesis

                95 %   Conf Int
         best   lower   upper
Vmax    90.66    90.1    91.2
Jmax   205.67   204.8   206.5
Rd       0.65     0.6     0.7

 Corr  Vmax and Jmax: 0.2635042

 Resid Sums Sq: 0.08357593

Convergence:YES
```

The small confidence intervals are a result of using fabricated data. We can examine the quality of the fit by plotting the residuals. The option `resid` is used to plot 'raw' residuals as opposed to standardized.

```
> plot(op100, resid = "raw")
```

The residuals show one outlier, but the deviations are small. The option `plot.kind` is used to plot the observed vs. fitted.

```
> plot(op100, plot.kind = "OvsF")
> plot(op100, plot.kind = "OandF", type = "o")
```

This function can optimize photosynthesis considering assimilation and intercellular CO_2 both as outputs of the model, but this should be done only for 'slow-measured' curves. For A/C_i curves the values of atmospheric CO_2 should also be supplied. The fifth column with C_i values is optional. This allows this optimization function to at least attempt to optimize any type of photosynthesis data including diurnals, temperature response functions and A/Q curves as well. Not all data are suitable to estimate the three parameters shown here, so the optimization level could also be adjusted.

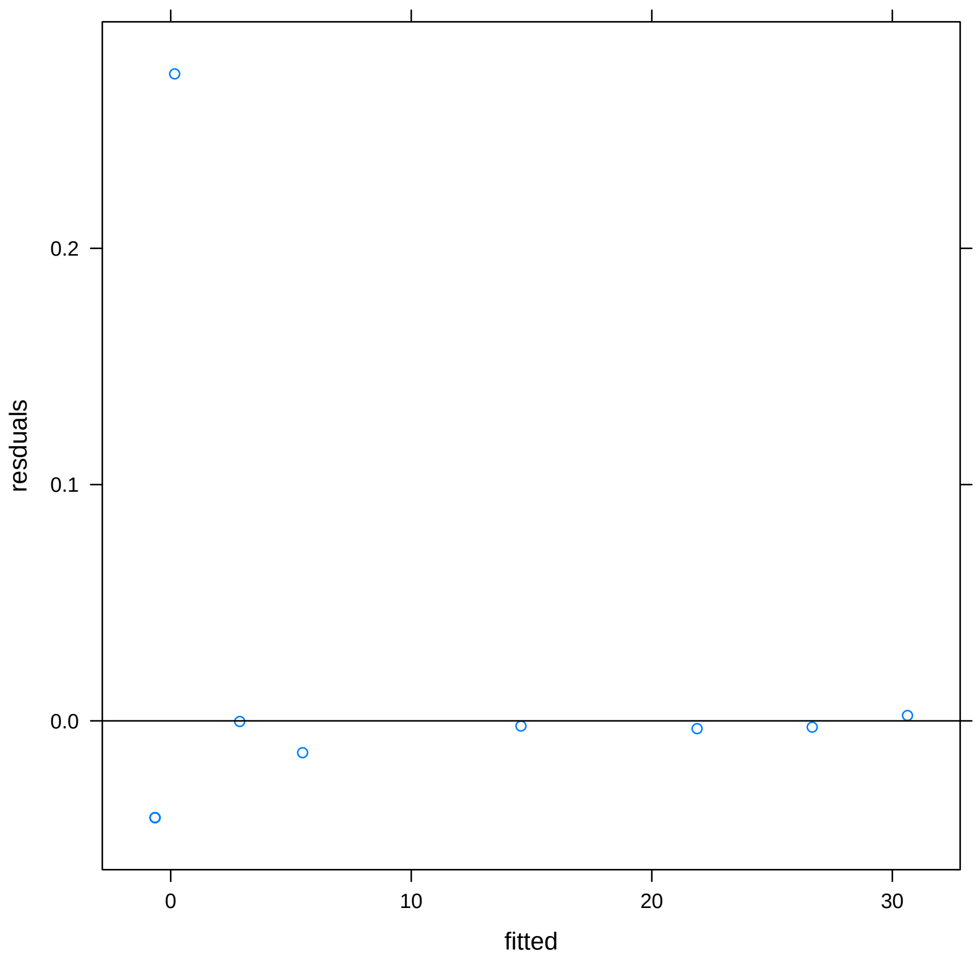

Figure 2: Raw residuals for op100

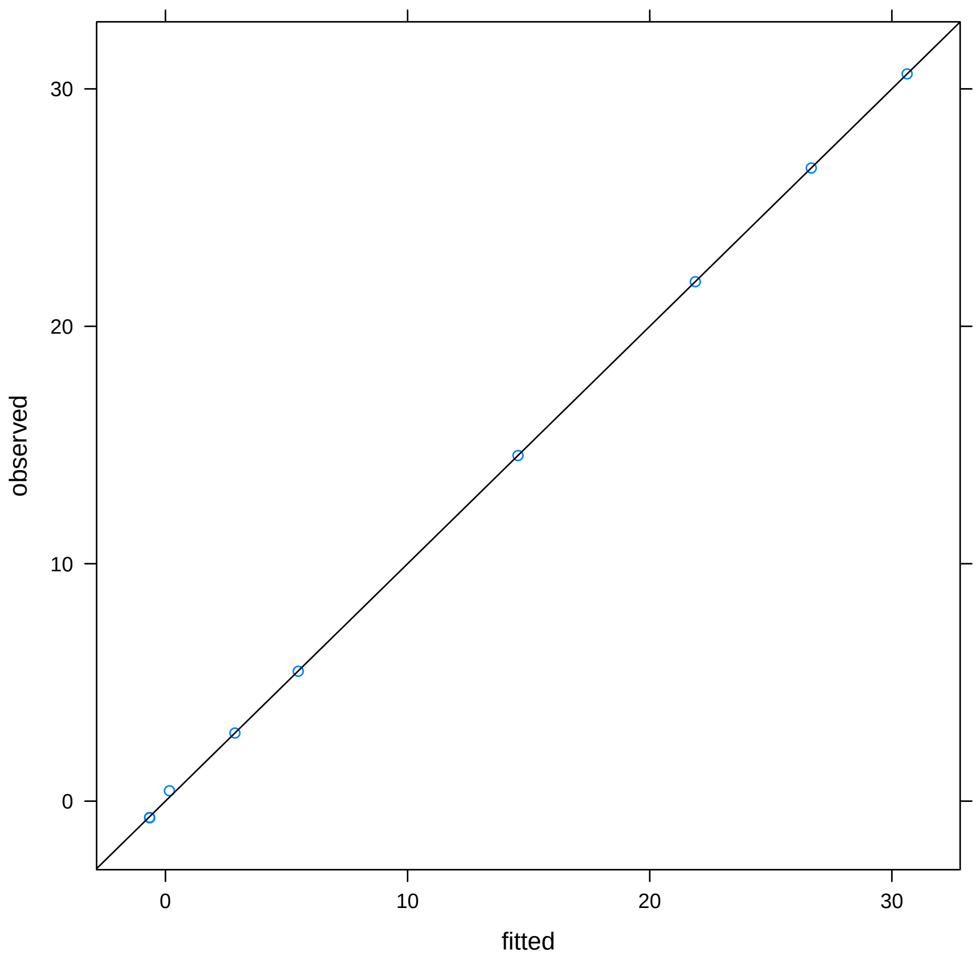

Figure 3: Observed vs. fitted for the optimization on the simulated data.

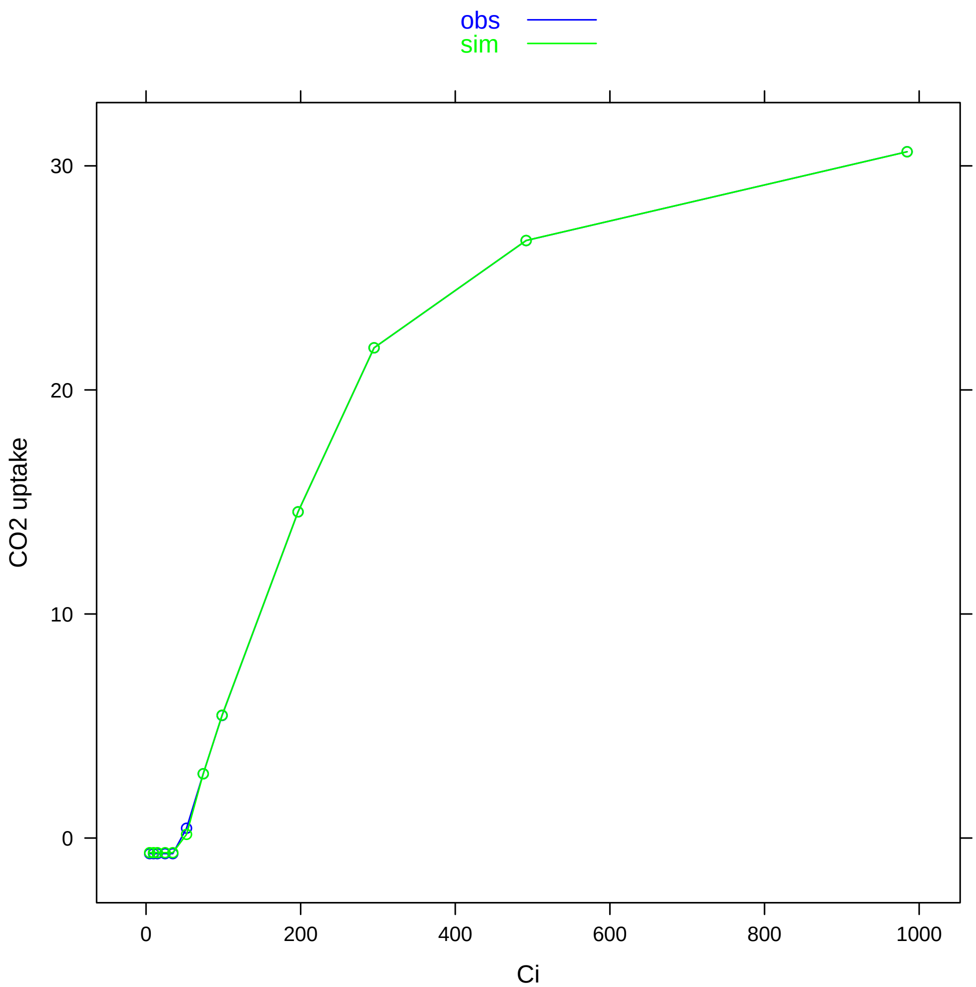

Figure 4: Observed and fitted for the optimization on the simulated data.

3 Optimizing Parameters for multiple A/C_i curves

A wrapper function for `Opc3photo` called `mOpc3photo` can be used to optimize multiple A/C_i curves which are considered multiple 'runs'. An example dataset is included.

```
> data(simAssim)
> head(simAssim)
```

```
     run    Assim   Qp Tl  RH        Ci   Vcmax     Jmax       Rd Catm
[1,]   1 -2.326939 1500 25 0.7  4.653977 105.8553 197.6750 2.311849    0
[2,]   1 -2.326939 1500 25 0.7  9.653877 105.8553 197.6750 2.311849    5
[3,]   1 -2.326939 1500 25 0.7 14.653877 105.8553 197.6750 2.311849   10
[4,]   1 -2.326939 1500 25 0.7 24.653877 105.8553 197.6750 2.311849   20
[5,]   1 -2.326939 1500 25 0.7 34.653877 105.8553 197.6750 2.311849   30
[6,]   1 -1.069772 1500 25 0.7 52.139543 105.8553 197.6750 2.311849   50
```

These has more than we need, but it contains the 'true' values used to generate the data so that we can later see if the optimization method can recover the 'true' values of the parameters. For the optimization we need this format.

```
> simAssim2 <- cbind(simAssim[, 1:6], Catm = simAssim[, 10])
> head(simAssim2)
```

```
     run    Assim   Qp Tl  RH        Ci Catm
[1,]   1 -2.326939 1500 25 0.7  4.653977    0
[2,]   1 -2.326939 1500 25 0.7  9.653877    5
[3,]   1 -2.326939 1500 25 0.7 14.653877   10
[4,]   1 -2.326939 1500 25 0.7 24.653877   20
[5,]   1 -2.326939 1500 25 0.7 34.653877   30
[6,]   1 -1.069772 1500 25 0.7 52.139543   50
```

```
> parms <- simAssim[seq(1, 3600, 12), 7:9]
```

The 'true' parameters were stored in the parms object. Now we can run the `mOpc3photo` function.

```
> op.all <- mOpc3photo(simAssim2, op.level = 2)
> op.all
```

```
Number of runs: 300
Number converged: 299

           mean         min         max
vmax  98.779988   81.811349  107.647065
jmax 205.681674  182.209486  236.321901
Rd     2.206139    1.732995    2.580131
```

For the initial run we know that 1 runs did not converge, but this is expected as what we want with this first optimization is to get good starting values. If some of them did not converge we could replace missing values with the mean of each parameter, but this is not needed here.

```
> colm <- apply(op.all$mat[, 2:4], 2, mean, na.rm = TRUE)
> ival <- op.all$mat[, 2:4]
> ival[is.na(ival[, 1]), 1] <- colm[1]
> ival[is.na(ival[, 2]), 2] <- colm[2]
> ival[is.na(ival[, 3]), 3] <- colm[3]
```

Now we can run it again.

```
> op.all2 <- mOpc3photo(simAssim2, iVcmax = ival[, 1], iJmax = ival[,
+     2], iRd = ival[, 3], op.level = 2)
> op.all2
```

```
Number of runs: 300
Number converged: 300

            mean         min         max
vmax    98.780335   81.812743  107.649342
jmax   205.681979  182.211809  236.325520
Rd       2.206140    1.732954    2.580021
```

Some of them might not converge, in this case all of them did. We can examine if each optimization was able to recover the true values.

```
> plot(parms[, 1], op.all2$mat[, 2], ylim = c(70, 110), xlim = c(70,
+     110), xlab = "Obs (true)", ylab = "Sim (est)", main = "Vcmax")
> abline(0, 1)
```

```
> plot(parms[, 2], op.all2$mat[, 3], xlab = "Obs (true)", ylab = "Sim (est)",
+     main = "Jmax")
> abline(0, 1)
```

```
> plot(parms[, 3], op.all2$mat[, 4], xlab = "Obs (true)", ylab = "Sim (est)",
+     main = "Rd")
> abline(0, 1)
```

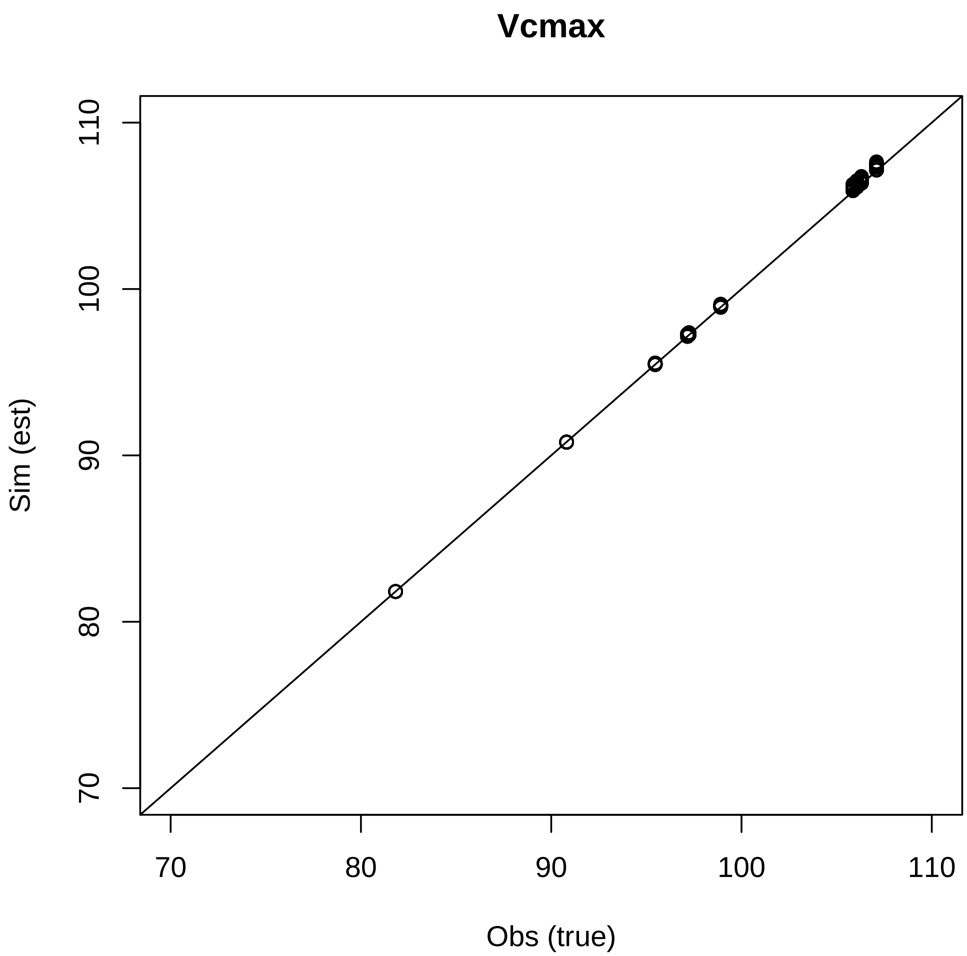

Figure 5: Agreement between true and estimated values for Vcmax

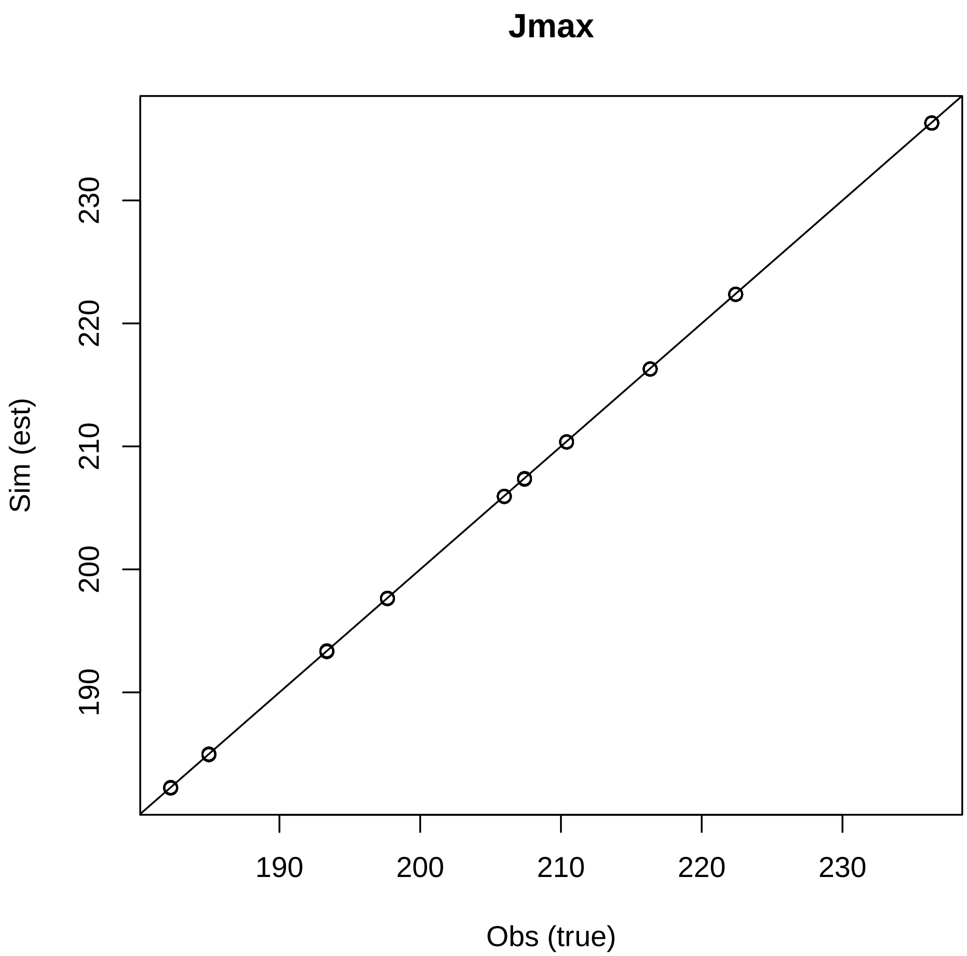

Figure 6: Agreement between true and estimated values for Jmax

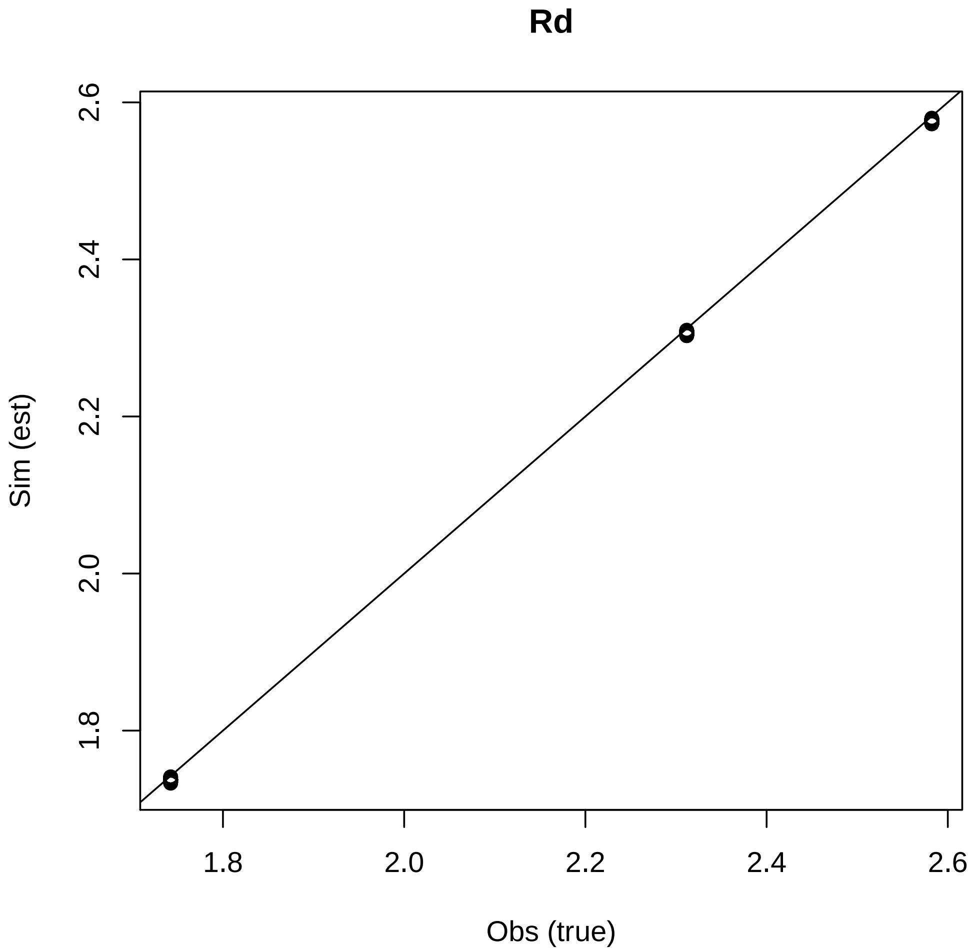

Figure 7: Agreement between true and estimated values for Rd