Supplementary Material

SAS programs for estimation of the FvCB model, with instructions

The following programs are written for the SAS System for Windows, version 9.1, but other software is equally suitable. Programming statements are denoted by the font `Courier New`, with *`italics`* reserved for input that would be dependent on each analysis. Copying only the programming passages into a program editor will produce a complete program. Brief instructions are included in the program itself, and more extensive ones in the text.

Prior to estimating the model for a large number of A/C_i sets contained in separate data files outputted by measuring instruments, it is advisable to consolidate all data in a single file. A SAS program suitable to process all files contained in a computer directory (folder), into a single file ready for recursive fitting of all included A/C_i sets, without the need to open the individual files, is available by contacting the correspondence author.

Example of the steps in processing A/C_i data.

1. Download data files from instrument.
2. Consolidate into a single file, and ascertain that the instrument and measuring procedure functioned adequately for every A/C_i set.
3. Run the program.
4. Examine all output for: failure to converge, singular Hessians, missing statistics, bounds that were reached, biologically absurd values, and $C_{i\,tr}$ outside of the range of observed values. In each problem case, determine whether or not two segments were sampled.

5. Fit only the problematic A/Ci sets: use a one-segment model for one-segment data, add a partial constraint on $C_{i\,tr}$ for the others.

Program for simultaneous estimation, with output of estimates to a data set, and graphics.

The following example of a SAS program used in estimating the FvCB model assumes a data set named 'dataset.csv', formatted as comma–separated text starting with a single line of headings, and containing A/C_i sets recorded at several levels of two factors (denoted 'factor1' and 'factor2'), and at several times ('days'). Those details are only for illustrative purposes, and can be changed easily. The model is fitted recursively to each A/C_i set defined by a single combination of 'factor 1', 'factor 2', and 'days'. In the SAS system, this is accomplished through 'by–processing'. Such recursive estimation is not an integral part of segmented regression, but segmented regression makes it possible, and it is essential to more efficient analysis of large amounts of A/C_i data.

It is assumed that every observation (line) in the data set includes the relevant value of the variables (columns) *factor1*, *factor2*, *days*, C_i, A, and leaf temperature in degrees centigrade ('`Tleaf`'). The program reads the data, adds temperature–adjusted values of K_c, K_o, and Γ_* ('`Gstar`'), estimates the two–segment model for every A/C_i set, and outputs the estimates and their standard errors. Each A/C_i set is then plotted, along with modeled A. Some examples of additional exploratory graphs are given, along with graphs of residuals. Maximum likelihood can be substituted to NLOLS in other SAS procedures, and robust methods such as Iteratively Reweighted Least Squares can be used in NLIN and other procedures. The validity

of estimates being closely dependent on the reliability of fixed parameters, the methods through which those inherited values were originally derived should be carefully reviewed.

```
/*+------------------------------------------------------------+
  |     SAS PROGRAM FOR SIMULTANEOUS ESTIMATION OF THREE       |
  |     PARAMETERS OF THE FARQUHAR-VON CAEMMERER-BERRY         |
  |     MODEL OF PHOTOSYNTHETIC CARBON ASSIMILATION            |
  |                                                            |
  |    Reference: Dubois et al. 2007                           |
  |    Optimizing the statistical estimation of the            |
  |    parameters of the Farquhar–von Caemmerer–Berry          |
  |    model of photosynthesis. The New Phytologist.           |
  |                                                            |
  |    Programmed by J.-J.B. Dubois                            |
  |                  USDA/ARS Plant Science Research Unit      |
  |                  3908 Inwood Road                          |
  |                  Raleigh NC 27603                          |
  |                  United States of America                  |
  |                                          March 2007        |
  +------------------------------------------------------------+*/
options ls=80 ps=1000 formdlim='_' nodate nonumber;

data ACI;
infile 'dataset.csv' dlm=',' firstobs=2 dsd missover;
input factor1 factor2 days Ci A Tleaf;
/* The following temperature adjusted coefficients are taken from
   Bernacchi et al. 2001. Plant, Cell and Environment. 24, 253-259   */
   R=0.008314472;
   Kc=exp(38.05-79.43/(R*(Tleaf+273.15)));
   Ko=exp(20.30-36.38/(R*(Tleaf+273.15)));
   Gstar=exp(19.02-38.83/(R*(Tleaf+273.15)));
run;
```

Additional details: Algebraic computation of temperature corrections to K_c, K_o, and $\Gamma*$, and pressure correction to O, if required, should precede fitting. They can be accomplished either within the NLIN procedure, or in a DATA step preceding it, as illustrated. Temperature standardization of the estimates of $V_{c\,max}$, J, and R_d can likewise be accomplished within the NLIN procedure, or in a DATA step following it. In the interest of clarity, performing all such operations in steps separate from the estimation procedure is strongly preferable. When gas exchange measurements are recorded at non–saturating PPFD, computation of J_{max} from

outputted values of J is best performed in a separate step as well. The set values for K_c, K_o, and Γ_* and their temperature adjustments in this example program were computed from experimental data using transformed tobacco (*Nicotiana tabacum*) with low Rubisco concentration (Bernacchi *et al.* 2001). Changes in these values have a strong effect on estimates of $V_{c\,max}$, J, and R_d, and taxon–appropriate values and adjustments should be substituted if available. Any adjustment to O, the partial pressure of O_2, should be performed in this first DATA step. In that eventuality, the line '`control O=210;`', whose function is to set the value of O in the estimation procedure, must be deleted in the following statements.

```
/*****************************************************************
  Fit the FvCB model to each A/Ci set
*****************************************************************/
proc sort data=ACI;
   by factor1 factor2 days;
run;
ods output ParameterEstimates=estimates;
proc NLIN data=ACI  method=Marquardt outest=status noitprint;
   by factor1 factor2 days;
   rubisco= ((Vcmax*(Ci-Gstar))/(Ci+(Kc*(1+(O/Ko)))));
   RuBP= ((J*(Ci-Gstar))/((4*Ci)+(8*Gstar)));
   control O=210;   *fixes the value of O;
   bounds -3<Rd<50;
   parameters Vcmax= 5 to 330 by 25
                  J= 15 to 505 by 35
                 Rd= 1E-8 to 10.1 by .5;
   model A=min(rubisco,RuBP)-Rd;
output out=ACiOUT parms=Vcmax J Rd predicted=Ahat student=student;
run;
ods listing;
data temporary;
   set status;
   where(_TYPE_ ne'GRID' and _TYPE_ ne'COVB');
run;
data estimates;
   merge estimates temporary(keep=factor1 factor2 days _status_ _iter_);
   by factor1 factor2 days;
run;
```

Additional details: The output data set '`ACiOUT`' contains all the original observations, the 'by' variables, estimates for $V_{c\,max}$, J, and R_d, predicted values for every observation in the

'ACI' data set, and studentized residuals. The data set 'estimates' presents, in tabular form, a summary of all estimates, standard errors, tests of significance, and whether the procedure converged, or was terminated without convergence. The full output should be reviewed, for such problems as biased estimates, and singular Hessians. As indicated in the text, special care should also be taken to inspect data for which estimation reaches set bounds. In the above programs, bounds are specified for R_d, through the statement "bounds 3<Rd<50;"

The following statements produce data sets suitable for graphing the results, and various graphs.

```
/*********************************************************************
  Create data for modeled curves
*********************************************************************/
*1) collect parameters, using average temperature from each A/Ci set;
proc means data=ACiOUT noprint;
   by factor1 factor2 days;
   var Tleaf Kc Ko Gstar Vcmax J Rd;
output out=parms(drop=_type_ _freq_)  mean=;
data parms;
   set parms;
   Citr=(-(Vcmax*8*Ko*Gstar)+(J*Kc*Ko)+(Kc*J*210))/(Ko*((Vcmax*4)-J));
run;
*2) create high density modeled points using estimated parameters;
data modeled; set parms;
   do Ci=0 to 1400 by 10;
/*adjust the range and density of modeled points by modifying the
  statement 'do Ci=0 to 1400 by 10;'                              */

 rubisco=((Vcmax*(Ci-Gstar))/(Ci+(Kc*(1+(210/Ko)))))-Rd;
   RuBP =((J*(Ci-Gstar))/((4*Ci)+(8*Gstar)))-Rd;
   if Ci<Gstar then FvCB=rubisco;
   else FvCB=min(rubisco,RuBP);
   output;
   end;
run;
*3) combine observed and modeled;
data graphset;
   set ACiOUT modeled;
   rename A=observed;
run;
```

Additional details: If each observed temperature is retained, instead of using mean temperature over all observations within each A/C_i set, the graph of modeled assimilation will reflect the transitory effects of temperature changes. When data were collected under sub–saturating light, and J_{max} cannot be assumed to equate J, computation of J_{max} can be carried out using the 'PARMS' data set, as can temperature standardization of $V_{c\,max}$, J, and R_d to 25°C.

```
/*****************************************************************
  Set graphing environment
*****************************************************************/

goptions reset=global gunit=pct
noborder htitle=5 ftitle="Times New Roman" htext=4 ftext="Times New Roman";
legend1 frame cframe=CXF1F1F1 cborder=black mode=protect label=none
       value=(justify=right) position=(bottom inside right) offset=(-
.60,.725);
symbol1 value=dot height=5 color=navy;
symbol2 line=1 width=5 value=none color=CXCF0202 interpol=splines;
symbol3 line=4 width=2 value=none color=lime interpol=splines;
symbol4 line=4 width=2 value=none color=lightseagreen interpol=splines;
axis1 order=-10 to 90 by 10 label=(angle=90 rotate=0)
      value=(font="Times New Roman" height=3) minor=none;
axis2 order=0 to 1400 by 100 value=(font="Times New Roman" height=3)
minor=none;

/* the scale of the axes can be adjusted by modifying
   the 'order=<value>' option of the axis statements                */


/*****************************************************************
  Graph observations and modeled curve on same plot
*****************************************************************/
DM "graph;cancel;";
proc datasets library=work mt=cat nolist;
delete gseg;
run;
/*the preceding statements prevent the accumulation of large numbers
  of graphs in the catalog when the program is run repeatedly in a
  single session.                                                   */

proc sort data=graphset;
   by factor1 factor2 days;
```

```
run;
proc gplot data=graphset;
   by factor1 factor2 days;
   where Ci>Gstar; *prevents drawing of points extrapolated below Ci=Gstar;
   plot observed*Ci FvCB*Ci
   /cframe=CXEFEFEF legend=legend1 vaxis=axis1 haxis=axis2 vref=0 overlay;
run;
/*predicted points for Rubisco limited and rubp-regeneration limited
 functions are both available in data graphset, and can thus be overlayed
*/


/*************************************************************************
  Graph studentized residuals
*************************************************************************/
symbol1 value=circle height=3 interpol=needle width=3 color=black;
axis1 order=-15 to 15 by 1
      value=(font="Times New Roman" height=3) minor=none;*adjust axis range
as needed;
Title 'Residuals vs. Ci';
proc gplot data=ACiOUT;
   by factor1 factor2 days;
   plot student*Ci
   /cframe=CXEFEFEF legend=legend1 vaxis=axis1 haxis=axis2;
Title 'Residuals vs. predicted A';
proc gplot data=ACiOUT;
   by factor1 factor2 days;
   plot student*Ahat
   /cframe=CXEFEFEF legend=legend1 vaxis=axis1;
run;


/*************************************************************************
  Examples of exploratory plots
*************************************************************************/
Title 'Check for strange values';
symbol1 font="Wingdings" value="¤" height=4 color=blue interpol=none;
symbol2 font="Wingdings" value="¤" height=4 color=firebrick interpol=none;
symbol3 font="Wingdings" value="¤" height=4 color=green interpol=none;
symbol4 font="Wingdings" value="¤" height=4 color=goldenrod interpol=none;
proc gplot data=parms;
      plot J*Vcmax=factor1;
      plot Citr*factor1;
      plot Citr*days=factor1;
      plot Vcmax*days=factor1;
run;
quit;
```

Using an explicit expression for $C_{i\,tr}$

Modify the relevant section of the above program using :

```
...
```

```
proc NLIN data= dataset;
parameters Vcmax= values J= values Rd= values;
    Citr= (Kc*J*(Ko+O)-(8*Ko*Gstar*Vcmax))/(Ko*(4*Vcmax-J));
    rubisco= ((Vcmax*(Ci-Gstar))/(Ci+(Kc*(1+(O/Ko)))));
    RuBP= ((J*(Ci-Gstar))/((4*Ci)+(8*Gstar)));
if Ci<Citr then model A= rubisco-Rd;
else model A= RuBP-Rd;
run;
...
```

This syntax requires unchanging fixed parameters. Since the value of K_c, K_o, and $\Gamma*$ vary with temperature, temperature must be assumed to be invariant within each A/C_i set. If temperature fluctuations were recorded, using their mean over each A/C_i set is a pragmatic way to satisfy this assumption. To that effect, a MEANS procedure can be added between the starting DATA step, and the NLIN procedure.

Note that the two syntaxes also have different implications for points of A that are below the ($\Gamma*$, $-R_d$) point. The expression [$A = \min\{A_c, A_j\}$] implies that the modeled function would be extended below [$C_i = \Gamma*$] by A_j, not by A_c. The segmented function A would thus have an additional change point at $\Gamma*$. That is what the syntax …`model=min(rubisco,RuBP)-Rd`… accomplishes: where [$C_i < \Gamma*$], A is modeled as A_j, just as where [$C_i \geq C_{i\,tr}$]. By contrast, an explicit $C_{i\,tr}$ results in the equation '`rubisco`' being used for all points where [$C_i < C_{i\,tr}$], *including* points where [$C_i < \Gamma*$], and the equation '`RuBP`' being used where [$C_i \geq C_{i\,tr}$]. This implies that below the first intersection of A_c and A_j, i.e. below ($\Gamma*$, $-R_d$), the assimilation function A corresponds to A_c, the Rubisco–saturation limited function. Both syntaxes produce the same parameter estimates, however.

Avoiding local minima through a grid search

The statement "`parameters Vcmax= 5 to 330 by 25 J= 15 to 505 by 35 Rd= 1E-8 to 10.1 by .5;`" generates 4410 starting combinations of $V_{c\,max}$, J and R_d, based on which

4410 SSEs are computed and compared for each A/C_i set. As explained in the text, using a large, high density grid search for starting values ensures that minimization will start in sufficient proximity to the global minimum to avoid convergence to a local one. It should be noted, first, that narrower ranges, or different ones, can be used when prior knowledge of the plants and environment under consideration suggests it. Second, that starting values may also be chosen based on the logic of the model: as an effect of the hyperbolic form of A_j, and provided that a few points have been sampled in sufficient proximity to its asymptotic region, J can be approximated by multiplying by 4 the sum of the largest A observed and of approximated R_d. The relationship between $V_{c\,max}$ and J can then be used to deduce a potential range of starting values for $V_{c\,max}$. Third, that the combination selected by a grid search being but a starting point for fitting, eventual estimates are not restricted to these ranges.

Estimating g_i.

Although, as reviewed in the text, estimating g_i through regression on gas exchange data is likely to be less reliable than through other means, the first program may be modified as follows:

```
...
proc NLIN data=ACI  method=Marquardt outest=status noitprint;
   by factor1 factor2 days;
   Cc=Ci-(A*giinv); *because gi may approach 0, its inverse is used;
   rubisco= ((Vcmax*(Cc-Gstar))/(Cc+(Kc*(1+(O/Ko)))));
   RuBP= ((J*(Cc-Gstar))/((4*Cc)+(8*Gstar)));
   control O=210;   *fixes the value of O;
   bounds -3<Rd<50;
   parameters giinv=0 to 10 by 1;
              Vcmax= 5 to 330 by 25
              J= 15 to 505 by 35
              Rd= 1E-8 to 10.1 by .5;
   model A=min(rubisco,RuBP)-Rd;
output out=ACiOUT parms=giinv Vcmax J Rd predicted=Ahat student=student;
run;
...
```

Fitting only one segment

When a single segment model has been determined to be more appropriate, the following can be used to fit either A_c, or A_j only.

A_c only:

```
...
proc NLIN data= dataset;
parameters Vcmax= values Rd= values;
    rubisco= ((Vcmax*(Ci-Gstar))/(Ci+(Kc*(1+(O/Ko)))));
model A= rubisco-Rd;
run;
...
```

A_j only:

```
...
proc NLIN data= dataset;
parameters Vcmax= values Rd= values;
    RuBP= ((J*(Ci-Gstar))/((4*Ci)+(8*Gstar)));
model A= RuBP-Rd;
run;
...
```

Placing a partial constraint on $C_{i\,tr}$

As explained in the text, a single segment will fit some rare A/C_i sets that contain data from both segments, better than the two–segment model. For other infrequent sets, excessive correlation among parameters may lead to convergence failure. In such rare cases, constraining the range of $C_{i\,tr}$ may be called upon, and is best accomplished by setting a boundary C_i value for fitting the two segments, rather than by setting $C_{i\,tr}$ itself. Contrary to disjunct methods, this maintains estimation from the entire A/C_i set. Constraining of $C_{i\,tr}$ is achieved through adjustments to all three parameters being estimated, and retains some flexibility in $C_{i\,tr}$:

```
...
proc NLIN data= dataset;
parameters Vcmax= values J= values Rd= values;
    rubisco= ((Vcmax*(Ci-Gstar))/(Ci+(Kc*(1+(O/Ko)))));
```

```
    RuBP= ((J*(Ci-Gstar))/((4*Ci)+(8*Gstar)));
if Ci< value then model A= rubisco-Rd;
else model A= RuBP-Rd;
run;
...
```

The most appropriate value for the condition '`if Ci< value`' is the mean of $C_{i\,tr}$ for related sets. This syntax does restrict the range of $C_{i\,tr}$, but does not require that a fixed value be set for it. Some software, such as the MODEL procedure of the SAS System, permit explicit bounds for quantities that are calculated from estimated parameters, such as $C_{i\,tr}$.

Benchmarks.

The following processing times were recorded on an ordinary low processing power desktop computer, equipped with a 1.8 GHz Pentium 4 CPU, 768 MB of RAM, and using the Windows XP operating system. The model was estimated for 200 separate A/C_i sets in approximately 2.3 seconds, when a 100 point grid was used for starting values of the three parameters. Using the 4410 point grid shown in the above program, execution took 14 seconds. This is considerably faster than manual subsetting and fitting of two segments for each of the 200 A/C_i sets. The effects of factors of interest, such as for example taxon, or temperature, on $V_{c\,max}$, J, and R_d, can be analyzed in the immediate next step, using the data set to which the parameter estimates are outputted.