This journal is © The Royal Society of Chemistry 2014

Supplementary Information

Box S1 Procedure of fugacity-based food web bioaccumulation model

```
function   [Cf BCF BAF f Dm FLXup FLXlt] = Food web(DieMat,BioProp,ChemConc,lgKow,DIR,fIR,MR)
% Return to biological concentrations (Cf, ng/g), fugacities (f), bioconcentration
% factors (BCF), bioaccumulation factors (BAF), biotransformation D-values (Dm)
% for an individual chemical in the aquatic compartment based on the fugacity-based
% food web models. Additionally, the chemical fluxes for the food web will be
% also optionally returned, including the uptake flux (FLXup) and loss flux (FLXlt),
% and the former includes the flux via respiration, diet and the biodegradation
% via respective maternal chemical, while the latter via respiration, egestion,
% biotransformation, and growth dilution, repectively.
% For more information pertaining to this program or any others in this paper,
% please contact with the corresponding author via rusg@ouc.edu.cn

error(nargchk(4,7,nargin))
if nargin==5
      error('The argument of "fIR" are required when "DIR" was inputted ');
end
if nargin==6
      MR = 1;
end

ln2 = log(2);
%---(1)------- Reading parameters
Vfp   = ChemConc(2,2);    %% Lipid volume fraction of SS (V/V,m/m)
Vfs   = ChemConc(3,2);    %% Lipid volume fraction of sediment solids (V/V,m/m)
Dnp   = ChemConc(2,3);    %% Solids density of SS (kg/m3)
Dns   = ChemConc(3,3);    %% Solids density of sediment (kg/m3)
TOCp = ChemConc(2,4);    %% Total organic carbon in suspended particles (g/g,dwt)
TOCs = ChemConc(3,4);    %% Total organic carbon in sediment solids (g/g,dwt)
Cwb   = ChemConc(1,5);    %% Chemical concentration in bulk water (ng/L)
Cs    = ChemConc(3,5);    %% Chemical concentration in sediment solids (ng/g,dwt)

V   = BioProp(:,1);    %% Organism volume (cm3)
L   = BioProp(:,2);    %% Lipid volume fraction (V/V)
HL = BioProp(:,3);    %% Chemical Biotransformation/metabolic half-life (d)
GR = BioProp(:,4);    %% Growth rate (1/d)
Fd = BioProp(:,5);    %% Feeding rate (1/d)
X   = BioProp(:,[6 7]); %% Fractional respiration from overlying water and porewater
Aw = BioProp(:,8);    %% Gut absorption efficiency for water
Ao = BioProp(:,9);    %% Gut absorption efficiency for lipid
clear   ChemProp ChemConc BioProp

%%----(2)----Calculate Z-values for Type 1 chemicals
Kow = 10^lgKow;      %% Calculation of the octanol/water partition coeffent
```

```
Koc = 0.41*Kow ;      %% Calculation of the organic carbon water partition coefficient
Zw   = 1 ;                   %% Calculation of the Z-value for water
Zo   = Zw * Kow;        %% Calculation of the Z-value for octanol
Zp   = Zw * TOCp * Koc * Dnp/1000; %% Z-value for particles in the water column
Zs   = Zw * TOCs * Koc * Dns/1000; %% Z-value for solid sediment particles
Zwb = Zw * (1 - Vfp) + Vfp*Zp;       %% Bulk Z-value for water
Zsb = Zw * (1 - Vfs) + Vfs*Zs;       %% Bulk Z-value for the sediment

%%----(3)---Calculate the concentration and fugacity in water and sediment
Cwb = Cwb /1E+6;   %% Chemical concentration in the water column (g/m3)
fw   = Cwb /Zwb;     %% Fugacity of the chemical in water              (g/m3)
Cw   = fw * Zw;       %% Dissolved concentration of the chemical in water (g/m3)
Cs   = (Cs*Dns)/1E+6;%% Concentration in sediment solid particles (g/m3)
fs   = Cs /Zs;          %% Sediment fugacity

%%----(4)----Calculate biological parameters for the food web
V   = V/1E+6;          %% Organism volume (m3)
HL = HL*24;            %% Organism Biotransformation half-life (h)
km = ln2./HL;        %% Organism Biotransformation rate constant (1/h)
GR = GR / 24;        %% Growth rate constant (1/h)
Ga = Fd.*V /24;    %% Food intake rate (m3)
Qw = 88.3 * (V*1000).^0.6; %% Water transport parameter
QL = 0.001 * Qw;                    %% Lipid transport parameter
k1D = 1./ ((V*1000./Qw) + (V*1000./QL)/Kow); %% The water uptake rate constant (1/d)
k1   = k1D / 24;              %% Water uptake rate constant (1/h)
Zf   = Zw * Kow * L;        %% Z-value for the organism
BCF = Zf / Zw;               %% Equilibrium bioconcentration factor (BCF)

%%----(5)----Calculate D-values
Dw   = V.*Zw.*k1;             %% Gill D-value
Dm   = V.*Zf.*km;             %% Biotransformation D-value
Dg   = V.*Zf.*GR;             %% Growth D-value
Ea   = (Aw*Kow + Ao).^(-1); %% Gut chemical absorbance efficiency
Da   = diag(Ea)*diag(Ga)*DieMat*diag(L)*Zo;%% Net uptake D-value i.e. including efficiency
Dat = sum(Da,2);               %% D-value for the total gross uptake from all species
Q     = 3;                           %% Digestion Factor
De   = Dat./Q;                    %% Loss by egestion
Dt   = Dw + Dm + Dg + De;%% Total Output

%%----(6)----Calculate food web fugacity, concentration, and acute BCF value
W   = Dw./Dt;                  %% Fugacity Multiplier for water and porewater respiration
A   = eye(length(V)) - diag(Dt)\Da;     %% Feeding matrix
E   = W.* (X*[fw fs]');%% Exposure via respiration
if nargin>=6
       ER = MR*DIR./Dt.*fIR;%% Exposure via the biodegradation of other chemical
       E   = E + ER;           %% Multiple total exposure
```

```
end
f   = inv(A)*E;              %% Food web fugacity
Cf = f.* Zf;                 %% Fish concentraiton (g/m3)
Cf = 1000 * Cf;              %% Fish concentration (ng/g)
BAF = Cf./(Cw*X*[1 fs/fw]')/1000;%% Bioaccumulate factor, donation from both water column and porewater are taken in consideration

%%----(7)----Calculate the flux in the food web
if nargout>5
    fa = zeros(length(Dat),1);%% Total prey fugacities
    n   = find(~~Dat);                %% Get the list numbers of the preys in the food web
    fa(n) = Da(n,:)*f./Dat(n);
    Uw = X*[fw fs]'.*Dw;      %% Uptake flux by respiration                  (g/h)
    Ua = fa.* Dat;                %% Uptake flux from food
    FLXup = [Uw Ua];
    if nargin>=6
        Ur    = DIR.*fIR*MR; %% Uptake flux by the biodegradation of other chemical
        FLXup = [FLXup Ur];
    end
    FLXlt = f*ones(1,4).*[Dw De Dm Dg];%% Flux lost by respiration, egestion, biotransformation, and growth dilution
end
```

Box S2 The procedure to compute the Standard Regression Coefficients (*SRC*)

```
function    [SRC H P Stat] = Parregress(X, Y, testway, alf)
% Partial regression for variable Y matching a multiple-independent-variable matrix of X. This function return to the
% standard regression coefficients (SRC) with concerned significant test parameters of H, P, and F/t-value based on a
% F-test or t-test. The test approach should be stated by assigning the parameter of 'testway' with an 'F' for F-test or 't'
% for t-test, and then the parameter of Stat stands for F-value or t-value respectively. If it's not assigned, F-test would
% be performed as a default value. X must be a 2-D array with each row standing for an independent-variable vector
% while Y should be a row vector.

if nargin==2||nargin==3
    alf = 0.05;
end
if nargin==2
    testway = 'F';
end
[b R2 H P F SS MS] = Lineregress(X, Y, alf);
clear R2 H P F SS
b(1) = [];
MSe    = MS(2);
SRC    = b.*std(X)/std(Y);    %% Partial coefficient
```

```
[n k] = size(X);
if testway=='F'; %% F-test for partial coefficient
    SSr = SubSSr(X,Y);
    for i = 1:k
        x = X;
        x(:,i) = [];
        ssr(i) = SubSSr(x,Y);
    end
    SSP = SSr - ssr;
    F   = SSP/MSe;
    P   = fcdf(F,k-1,n-k-1);
    P   = min(P,1-P);
    H   = P<alf/2;
    Stat = F;
elseif testway=='t'; %% t-test for partial coefficient
    X = [ones(n,1) X];
    SE = sqrt(MSe*diag(inv(X'*X)))';
    SE(1) = [];
    t   = b./SE;
    P   = tcdf(t,n-k-1);
    P   = min(P,1-P);
    H   = P<alf/2;
    Stat = t;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function SSr = SubSSr(X,Y)
%   Subfunction for SSr calculation
b     = Lineregress(X,Y);
b(1) = [];
Sy    = (X'*Y)' - sum(X)*mean(Y);
SSr   = b*Sy';


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [B R2 H P F SS MS] = Lineregress(X, Y, alf)
% Subfunction for multiple liner regulation in the form of Y = X·β+ε
% and ANOVA of b based on F-test
% X must be a 2-D array with each row for different independent vectors, as follows:
%     X = [1 X] = [1 X1 X2 ...Xn]
% Y and E(b') =βshould be row vectors, as follows:
%      Y = [y1; y2; ...; yn]
%      B = [ a b1 b2 ... bn]
```

```
%----Parameters checking
if det(X'*X)==0
       error('Bad values of X, Please Check whether if a column of X with all the values of Zero')
end
%-----Lineregress
[n k] = size(X);
X    = [ones(n,1) X];
B    = (X'*X)\(X'*Y);
B    = B';

%----ANOVA of the regression coefficient b based on F-test
if nargin==2
       alf = 0.05;
end
b      = B(2:end);
X(:,1) = [];
Sy    = (X'*Y)' - sum(X)*mean(Y);
SSr = b*Sy';
V1 = k;
V2 = n - k - 1;
Syy = (Y'-mean(Y))*(Y-mean(Y));
SSe = Syy - SSr;
SSt = SSr + SSe;
R2    = SSr/SSt;
MSr = SSr/V1;
MSe = SSe/V2;
F      = MSr/MSe;
P      = fcdf(F,V1,V2);
P      = min(P,1-P);
H      = P<alf;
SS    = [SSr; SSe; SSt];
MS     = [MSr; MSe];
```