Electronic Supplementary Material (ESI) for Physical Chemistry Chemical Physics.

Supplementary Materials

Competitive DNA binding of $Ru(bpy)_2dppz^{2+}$ enantiomers studied with isothermal titration calorimetry (ITC) using a direct and general binding isotherm algorithm

Anna K. F. Mårtensson and Per Lincoln

Contents: Page

Table S1

Baseline enthalpy values (kJ mol^{-1}) for the fit of (6+5) parameter Model 1-3

Titration	Model 1	Model 2	Model 3
1) Δ in AT	0.42315	-0.10353	0.46363
2) Λ in AT	0.20343	-0.084847	-0.21065
3) Δ in AT + Λ	-0.028274	-0.27608	0.089987
4) Λ in AT + Δ	-0.52783	-0.47468	-0.74829

Note that in the following programs, the total concentration of binding sites, denoted B_0 in the main text, is now denoted D0.

GeneralAlgorithm.m

```
function [conc,x,f,P,r]=GeneralAlgorithm(C0,M,T,Y,K,n,r)
%General algortihm for solving the mass balance for
%ligands interacting with a linear polymer of identical binding site units
%D with explicit nearest neighbor interaction.

%INPUT:
%CO=[D0 A0 B0...] 1x(1+L) vector with total concentration of
%binding site units D and L ligands A, B...
%M = mxL matrix of the stoichiometric coefficients for the m ligand-
containg elementary
%units.
%T = mxm non-singular matrix transforming M such that in the product T*M no
column is
%zero, and each row only contains one 1, the rest being zeros.
%Y =(m+1)x(m+1)matrix containg the cooperativity parameters for nearest
%neighbor interactions between the elementary units, the first row and
%column being the interactions with the ligand-free elementary unit.
%K = Intrinsic binding constants of the elementary units.
%n = Number of binding site units made inaccesible by formation of the
elementary unit.
%r = Guess for vector r

%OUTPUT
%conc = [[Afree; Bfree ...][ Abound; Bbound ...]]
%x = binding potentials of elementary units
%f = binding densities of elementary units
%P = matrix of conditional probabilities
%r = vector r that solves the mass balance equations

[m,l]=size(M);
lnK=log(K);
TMM=T*M*M';
L0=T*M*C0(2:end,1); %free ligand concentrations
D0=C0(1);
q=1;
    while norm(q)>1e-9;
       s=1./(Y*[1;r]);
       P=diag(s)*Y*diag([1;r]);
       v=eye(m)-P(2:end,2:end)+n'*P(1,2:end);
       f=inv(v')*P(1,2:end)';
       x=s(2:end).*r./(s(1).^n');
       etl=exp(T*(log(x)-lnK));
       q=etl+D0*TMM*f-L0;
       u=inv(v')*(diag(f)-P(:,2:end)'*diag([1-n*f;f])*P(:,2:end));
       dqdr=(D0*TMM*u+diag(etl)*T*v);
        dr=diag(r)*(dqdr\q);
       while min(r-dr)<=0    %ensures that r remains positive
           dr=dr/3;
       end
       r=r-dr;
    end
       conc=[etl D0*TMM*f];
```

ITCalgorithmModel3.m

```
function [err,sim,heat,conc]=ITCalgorithmModel3(par,totdata,inj)
%Calculates ITC curve for binding of two ligands La and Lb
%with a lattice binding model in which
%A forms symmetrical subunits A1 and A2 (here with identical properties),
when binding
%to an infinite chain of alternating binding sites 1 and 2, and B only
%binds to site 1.
%This code is (7+5) parameter Model 3; can easily be converted to Model 1
%or Model 2, see comments on line 32-33 and 40.

%INPUT:
%par=[1) Ka 2) Kb 3) yAA  4) yBB 5) yAB 6) na 7) nb]
%For (9+7) parameter Model 3, Ka2 and yA2A2 are included in par
%and inserted into the appropriate matrices below.
%totdata=[ITC-data D0 La0 Lb0] (4*20) X 4 matrix with data and total
%concentrations of basepairs and ligands, preferably with total
concentrations of Ln in increasing order
%inj=[inj1 inj2 inj3 inj4] injN = mol injectant/Vcell for titration N

%OUTPUT:
%err is the least square error
%sim = [ITC-data   simulated data]
%heat=[bl1 bl2 bl3 bl4 bl5 dHa dHb dHaa dHbb dHab]; bln is baseline value
%for titration n, dHa etc. enthalpy changes per mol

data=totdata([2:20 22:40 42:60 62:80],1);
o=ones([19,1]);
e4=eye([4,4]);
bl=[o*e4(1,:);o*e4(2,:);o*e4(3,:);o*e4(4,:)];
dc=[];
K=par([1 1 2])';
%K(2)=1e-9;  %Effectively removes A binding to site 2, converting Model 3
to Model 2 (when ff=0.5) or
%to Model 1 (when ff=1).
Y=[1 1 1 1  ;1 par(3) 0 par(5);1 1 par(3) 1 ;1 par(5) 0 par(4)];
n=par([6 6 7]);
C0=totdata(:,2:4);
conc=[];
M=[1 0;1 0;0 1];
T=eye(3);
ff=.5;  %Fraction of binding site unit per basepair, set to 1 for Model 1

if min(par(1:7))<=0 %Ensures the variables stay positive
    err=1e6;
else
dc=[];
C0=totdata(:,2:4);
conc=[];
    r=[.1 .1]'; % For titrations 1 and 2, the appropriate submatrices are
defined; here La only
    Kt=K(1:2);
    Mt=M(1:2,1);
    Tt=T(1:2,1:2);
    nt=n(1:2);
    Yt=Y(1:3,1:3);
 for t=C0(1:20,:)';
    t(1)=t(1)*ff;
```

```
    t=t(1:2);

    [C,x,f,P,r]=GeneralAlgorithm(t,Mt,Tt,Yt,Kt,nt,r);
    PP=t(1)*diag(f)*P(2:3,2:3);
    conc=[conc;(t(1)*[1 1]*f)' 0 PP(1,1)+PP(2,2) 0 0];
    %Note that the assumed identical La contributions are summed together

 end
    r=[.1 ]';   %Lb only

    Kt=K(3);
    Mt=M(3,2);
    Tt=T(3,3);
    nt=n(3);
    Yt=Y([1 4],[1 4]);
for t=C0(21:40,:)';
    t(1)=t(1)*ff;
    t=t([1 3]);
    [C,x,f,P,r]=GeneralAlgorithm(t,Mt,Tt,Yt,Kt,nt,r);
    PP=t(1)*diag(f)*P(2,2);
    conc=[conc;0  (t(1)*f)' 0  PP(1,1) 0 ];

end
r=[.1 .1 .1 ]';
for t=C0(41:60,:)'; %La into Lb + DNA
    t(1)=t(1)*ff;

    [C,x,f,P,r]=GeneralAlgorithm(t,M,T,Y,K,n,r);
    PP=t(1)*diag(f)*P(2:4,2:4);
    conc=[conc;(t(1)*[1 1 0;0 0 1]*f)' PP(1,1)+PP(2,2) PP(3,3) PP(1,3)*2 ];
end
r=[.1 .1 .1 ]';   %Lb into La + DNA
for t=C0(61:80,:)';
    t(1)=t(1)*ff;

    [C,x,f,P,r]=GeneralAlgorithm(t,M,T,Y,K,n,r);
    PP=t(1)*diag(f)*P(2:4,2:4);
    conc=[conc;(t(1)*[1 1 0;0 0 1]*f)' PP(1,1)+PP(2,2) PP(3,3) PP(1,3)*2 ];

end
          %Calculated concentration changes per mol injectant
          %corrected for dilution:
    dc=[dc;[conc(2:20,:)-conc(1:19,:)*204/206]/inj(1)];
    dc=[dc;[conc(22:40,:)-conc(21:39,:)*204/206]/inj(2)];
    dc=[dc;[conc(42:60,:)-conc(41:59,:)*204/206]/inj(3)];
    dc=[dc;[conc(62:80,:)-conc(61:79,:)*204/206]/inj(4)];

    dc=[bl dc];

    heat=pinv(dc)*data;
    sim=dc*heat;

    err=norm(data-sim,'fro');

    sim=[data sim];
end
```

ITCalgorithmIndependent.m

```
function [err,sim,heat]=ITCalgorithmIndependent(par,totdata,inj)
%Calculates ITC curve for competitive binding of two ligands La
%and Lb to two types of independent, non-overlapping binding sites X and Y.
%The fractions fX and fY can take independent values for each of the four
%titrations, or at wish be restricted to fewer independent values.
%
%INPUT:
%par=[1) Kax 2) Kay  3) Kbx 4) Kby 5) fraction;
%fraction can be one variable, as in the present case, or up to 8 variables
in the unrestricted case.
%totdata=[ITC-data D0 La0 Lb0] (4*20) X 4 matrix with data and total
%concentrations of basepairs and ligands, preferably with total
%concentrations of Ln in increasing order.
%The data are assumed to be ordered as only La, only Lb, La into Lb, Lb
%into La
%inj=[inj1 inj2 inj3 inj4] injN = mol injectant/Vcell for titration N

%OUTPUT:
%err is the least square error
%sim = [ITC-data   simulated data]
%heat=[bl1 bl2 bl3 bl4 dHax dHay dHbx dHby]; bli is baseline value for
%titration i, dHax etc. enthalpy changes per mol, same units as in ITC-data

data=totdata([2:20 22:40 42:60 62:80],1);   %The column of ITC-data
o=ones([19,1]);
C0=totdata(:,2:4);  %The 3 columns of total concentrations
e4=eye([4,4]);
bl=[o*e4(1,:);o*e4(2,:);o*e4(3,:);o*e4(4,:)];

fraction=par(5)*[[1 1 1 1];[1 1 1 1]];
%fraction=[par([5 7 9 11]);par([6 8 10 12])]; %Unrestricted case, first row
%X-fractions, second row Y-fractions
conc=[];
Kx=par([1 2]);
Ky=par([3 4]);
dc=[];
if min(par(1:5))<=0  %All variable values should be positive, here 5
variables are varied.
    err=1e6;
else

 f=fraction(:,1);
 L=[.1]; %Guess for free ligand concentration
 for t=C0(1:20,:)'; % La only titration
    D0=t(1);  %basepair concentration.
    L0=t(2)'; %Ligand total concentration
    q=1;
    kx=Kx(1);
    ky=Ky(1);
    while norm(q)>1e-9;
       q=(kx*L/(1+kx*L)*f(1)+ky*L/(1+ky*L)*f(2))*D0+L-L0;
       dqdL=(kx/(1+kx*L)*f(1)+ky/(1+ky*L)*f(2)-
(f(1)*L*kx^2/((1+kx*L)^2))+f(2)*L*ky^2/((1+ky*L)^2))*D0+1;
       dL=q*inv(dqdL);
       L=L-dL;
    end
    conc=[conc;[f(1)*kx*L/(1+kx*L) 0 f(2)*ky*L/(1+ky*L) 0]*D0];
```

```
 end
    f=fraction(:,2);
 L=[.1];
 for t=C0(21:40,:)';    %Only Lb titration
    D0=t(1);
    L0=t(3)';
    q=1;
    kx=Kx(2);
    ky=Ky(2);
    while norm(q)>1e-9;
       q=(kx*L/(1+kx*L)*f(1)+ky*L/(1+ky*L)*f(2))*D0+L-L0;
       dqdL=(kx/(1+kx*L)*f(1)+ky/(1+ky*L)*f(2)-
(f(1)*L*kx^2/((1+kx*L)^2))+f(2)*L*ky^2/((1+ky*L)^2))*D0+1;
       dL=q*inv(dqdL);
       L=L-dL;
    end
    conc=[conc;[0 f(1)*kx*L/(1+kx*L) 0 f(2)*ky*L/(1+ky*L)]*D0];

 end
 f=fraction(:,3);
 L=[.1 .1];
 for t=C0(41:60,:)';%lLa into Lb
    D0=t(1);
    L0=t(2:3)';
    q=1;
    while norm(q)>1e-9;
       q=(Kx.*L/(1+Kx*L')*f(1)+Ky.*L/(1+Ky*L')*f(2))*D0+L-L0;
       dqdL=(diag(Kx)/(1+Kx*L')*f(1)+diag(Ky)/(1+Ky*L')*f(2)-
(f(1)*(diag(L)*Kx'*Kx/((1+Kx*L')^2))+f(2)*(diag(L)*Ky'*Ky/((1+Ky*L')^2))))*
D0+eye(2);
       dL=q*inv(dqdL');
       L=L-dL;
    end
    conc=[conc;[f(1)*Kx.*L/(1+Kx*L') f(2)*Ky.*L/(1+Ky*L')]*D0];

 end

 f=fraction(:,4);
 L=[.1 .1];
for t=C0(61:80,:)'; %Lb into La
    D0=t(1);
    L0=t(2:3)';
    q=1;
    while norm(q)>1e-9;
       q=(Kx.*L/(1+Kx*L')*f(1)+Ky.*L/(1+Ky*L')*f(2))*D0+L-L0;
       dqdL=(diag(Kx)/(1+Kx*L')*f(1)+diag(Ky)/(1+Ky*L')*f(2)-
(f(1)*(diag(L)*Kx'*Kx/((1+Kx*L')^2))+f(2)*(diag(L)*Ky'*Ky/((1+Ky*L')^2))))*
D0+eye(2);
       dL=q*inv(dqdL');
       L=L-dL;
    end
    conc=[conc;[f(1)*Kx.*L/(1+Kx*L') f(2)*Ky.*L/(1+Ky*L')]*D0];

end

   dc=[dc;[conc(2:20,:)-conc(1:19,:)*204/206]/inj(1)]; %Calculated
concentration changes per mol injectant
    dc=[dc;[conc(22:40,:)-conc(21:39,:)*204/206]/inj(2)]; %corrected for
dilution
    dc=[dc;[conc(42:60,:)-conc(41:59,:)*204/206]/inj(3)];
```

```
    dc=[dc;[conc(62:80,:)-conc(61:79,:)*204/206]/inj(4)];
    dc=[bl dc];

    heat=pinv(dc)*data;
    sim=dc*heat;

   err=norm(data-sim,'fro');

    sim=[data sim];
end
```

ITCalgorithmModel4:

```
function [err,sim,heat]=ITCalgorithmModel4(par,totdata,inj)
%Uses the lattice model to calculates the ITC curve for binding of two
ligands La
%and Lb to two independent binding sites X and Y, which fractions are equal

%INPUT:
%par=[1) Ka1 2) Kb1 3) Ka2 4) Kb2 5) fraction ]

%totdata=[ITC-data D0 La0 Lb0] (4*20) X 4 matrix with data and total
%concentrations of basepairs and ligands, preferably with total
concentrations of Ln in increasing order

%inj=[inj1 inj2 inj3 inj4] injN = mol injectant/Vcell for titration N

%OUTPUT:
%err is the least square error

%sim = [ITC-data   simulated data]

%heat=[bl1 bl2 bl3 bl4 dHaX dHbX dHaY dHbY]; bln is baseline value for
%titration n, dHaX etc. enthalpy changes per mol, same units as in ITC-data

data=totdata([2:20 22:40 42:60 62:80],1);
o=ones([19,1]);
e4=eye([4,4]);
bl=[o*e4(1,:);o*e4(2,:);o*e4(3,:);o*e4(4,:)];
dc=[];
C0=totdata(:,2:4);
conc=[];

ff=par(5); %Fraction of binding site per base-pair.


Y=ones([9,9]); %No cooperativity in this case
K=[par(1:4) par(1)*par(3) par(2)*par(4) par(1)*par(4) par(2)*par(3)]'; %To
comply with the 2 independent site model

n=ones([1,8]);  %Binding site coverage is unity
mm=[eye(4);[1 0 1 0;0 1 0 1;1 0 0 1;0 1 1 0] ];     %Matrix of elementary
unit composition in terms of LaX, LbX, LaY and LbY
M=mm*[1 0 1 0;0 1 0 1]';
T=eye(8);
T(5,5)=.5;
T(6,6)=.5;
T(7,2)=-1;
T(8,1)=-1;

if min(par(1:5))<=0  %Ensures that the variables stays positive
    err=1e6;
else

    r=[.1 .1 .1]';  %For each of the 4 titrations, appropriate submatrices
are constructed. Here La only titration.
    Kt=K([1 3 5]);
    Mt=M([1 3 5],1);
    Tt=T([1 3 5],[1 3 5]);
```

```
    nt=n([1 3 5]);
    Yt=Y([1 2 4 6],[1 2 4 6]);
    mmt=mm([1 3 5],:);
 for t=C0(1:20,:)';
    t(1)=t(1)*ff;
    t=t(1:2);
    [C,x,f,P,r]=GeneralAlgorithm(t,Mt,Tt,Yt,Kt,nt,r);
    conc=[conc;t(1)*f'*mmt ];
 end

    r=[.1 .1 .1]';  %Lb only titration
    Kt=K([2 4 6]);
    Mt=M([2 4 6],2);
    Tt=T([2 4 6],[2 4 6]);
    nt=n([2 4 6]);
    Yt=Y([1 3 5 7],[1 3 5 7]);
    mmt=mm([2 4 6],:);
 for t=C0(21:40,:)';
    t(1)=t(1)*ff;
    t=t([1 3]);
    [C,x,f,P,r]=GeneralAlgorithm(t,Mt,Tt,Yt,Kt,nt,r);
    conc=[conc;t(1)*f'*mmt ];
 end

r=[.1 .1 .1 .1 .1 .1 .1 .1]';   %La into Lb, the full matrices are used.
for t=C0(41:60,:)';
    t(1)=t(1)*ff;
    [C,x,f,P,r]=GeneralAlgorithm(t,M,T,Y,K,n,r);
     conc=[conc;t(1)*f'*mm ];
end

r=[.1 .1 .1 .1 .1 .1 .1 .1]';   %Lb into La
for t=C0(61:80,:)';
    t(1)=t(1)*ff;
    [C,x,f,P,r]=GeneralAlgorithm(t,M,T,Y,K,n,r);
     conc=[conc;t(1)*f'*mm ];
end
    %Calculated concentration changes per mol injectant
    %corrected for dilution
    dc=[dc;[conc(2:20,:)-conc(1:19,:)*204/206]/inj(1)];
    dc=[dc;[conc(22:40,:)-conc(21:39,:)*204/206]/inj(2)];
    dc=[dc;[conc(42:60,:)-conc(41:59,:)*204/206]/inj(3)];
    dc=[dc;[conc(62:80,:)-conc(61:79,:)*204/206]/inj(4)];
    dc=[bl dc];

    heat=pinv(dc)*data;
    sim=dc*heat;

    err=norm(data-sim,'fro');

    sim=[data sim];
end
```

Analysis of the ITC data:

The total concentrations in *TotData* are calculated from the concentration of injectant stock solution, volume added (2 μL) and the initial content of the cell and its volume (206 μL). For each injection, the injectant is assumed to displace 2 μL of the content in the cell before mixing. *Inj* is the concentration resulting from one 2 μL addition of injectant into a cell filled with buffer, and used to normalize the species concentrations calculated to the ITC-data in the first column. All concentrations are given in μM. See further comments in the program 'ITCalgorithm' above.

Indata:

TotData =

3.5673	148.8010	6.9029	0.0000
2.8297	147.3563	11.4378	0.0000
2.2158	145.9257	15.9287	0.0000
1.5232	144.5089	20.3760	0.0000
0.8603	143.1059	24.7801	0.0000
0.2285	141.7165	29.1415	0.0000
-0.6821	140.3406	33.4605	0.0000
-1.6861	138.9781	37.7376	0.0000
-2.6287	137.6288	41.9732	0.0000
-3.0987	136.2926	46.1676	0.0000
-3.3115	134.9694	50.3213	0.0000
-3.1798	133.6590	54.4347	0.0000
-2.5041	132.3613	58.5081	0.0000
-2.0208	131.0763	62.5420	0.0000
-1.5544	129.8037	66.5368	0.0000
-1.0936	128.5435	70.4927	0.0000
-0.7567	127.2955	74.4103	0.0000
-0.5287	126.0596	78.2898	0.0000
-0.4940	124.8357	82.1316	0.0000
-0.4883	123.6237	85.9362	0.0000
0.3559	148.8010	0.0000	6.8738
-2.1600	147.3563	0.0000	11.3896
-4.5524	145.9257	0.0000	15.8615
-6.7236	144.5089	0.0000	20.2900
-8.1363	143.1059	0.0000	24.6756
-9.3413	141.7165	0.0000	29.0185
-10.7015	140.3406	0.0000	33.3193
-11.6217	138.9781	0.0000	37.5784
-12.7751	137.6288	0.0000	41.7961
-13.4433	136.2926	0.0000	45.9728
-13.2426	134.9694	0.0000	50.1090
-11.4619	133.6590	0.0000	54.2050
-8.2448	132.3613	0.0000	58.2613
-5.8635	131.0763	0.0000	62.2781
-3.6970	129.8037	0.0000	66.2560
-2.4371	128.5435	0.0000	70.1953

-1.7974	127.2955	0.0000	74.0963
-1.2967	126.0596	0.0000	77.9595
-1.1205	124.8357	0.0000	81.7851
-1.1004	123.6237	0.0000	85.5736
0.9546	121.8234	6.9029	84.3274
1.5000	120.6406	11.4378	83.5087
1.5933	119.4693	15.9287	82.6979
1.6437	118.3094	20.3760	81.8950
1.6524	117.1608	24.7801	81.0999
1.6227	116.0233	29.1415	80.3125
1.5730	114.8969	33.4605	79.5328
1.5407	113.7814	37.7376	78.7606
1.4867	112.6767	41.9732	77.9960
1.4239	111.5828	46.1676	77.2387
1.3592	110.4994	50.3213	76.4888
1.2688	109.4266	54.4347	75.7462
1.1883	108.3642	58.5081	75.0108
1.0863	107.3121	62.5420	74.2826
1.0232	106.2703	66.5368	73.5614
0.9363	105.2385	70.4927	72.8472
0.8703	104.2168	74.4103	72.1399
0.7674	103.2050	78.2898	71.4395
0.6834	102.2030	82.1316	70.7459
0.6225	101.2107	85.9362	70.0591
-9.5212	121.8234	84.6847	6.8738
-9.2077	120.6406	83.8625	11.3896
-8.0398	119.4693	83.0483	15.8615
-7.0590	118.3094	82.2420	20.2900
-6.0224	117.1608	81.4435	24.6756
-4.9818	116.0233	80.6528	29.0185
-4.2492	114.8969	79.8698	33.3193
-3.7254	113.7814	79.0944	37.5784
-3.2104	112.6767	78.3265	41.7961
-2.8004	111.5828	77.5660	45.9728
-2.4630	110.4994	76.8129	50.1090
-2.1631	109.4266	76.0672	54.2050
-1.9390	108.3642	75.3287	58.2613
-1.7970	107.3121	74.5973	62.2781
-1.7187	106.2703	73.8731	66.2560
-1.6006	105.2385	73.1559	70.1953
-1.5198	104.2168	72.4456	74.0963
-1.4685	103.2050	71.7422	77.9595
-1.4097	102.2030	71.0457	81.7851
-1.3651	101.2107	70.3560	85.5736

Inj =

4.6019	4.5825	4.6019	4.5825

MATLAB Command Window:

```
>> options=optimset('Display','Iter','MaxIter',4000,'MaxFunEvals',8000,'TolX',1e-6,'TolFun',1e-6);
```

(**7+5) parameter Model 3**

```
>> [Err,Sim,Heat]=ITCalgorithmModel3(Par,TotData,Inj)

>> Par=fminsearch('ITCalgorithmModel3',[ 1  0.1  1  5  5  1.1  1.3],options,TotData,Inj)
```

After 35 seconds (722 iterations / 1127 evaluations of 'ITCalgortithmModel3'), convergence criteria are satisfied and

```
Par =

    0.6478    0.0871    1.0918    5.0097    5.9443    1.1374    1.2942

>> [Err,Sim,Heat]=ITCalgorithm(Par,TotData,Inj)

Err =

    0.8155


Sim =

    2.8297    2.8322
    2.2158    2.2557
    1.5232    1.6102
    0.8603    0.8886
    0.2285    0.0881
   -0.6821   -0.7817
   -1.6861   -1.6821
   -2.6287   -2.5191
   -3.0987   -3.1242
   -3.3115   -3.3150
   -3.1798   -3.0614
   -2.5041   -2.5475
   -2.0208   -1.9960
   -1.5544   -1.5225
   -1.0936   -1.1504
   -0.7567   -0.8660
   -0.5287   -0.6487
   -0.4940   -0.4809
   -0.4883   -0.3495
   -2.1600   -2.2965
   -4.5524   -4.6329
   -6.7236   -6.4670
   -8.1363   -8.0184
   -9.3413   -9.3977
```

-10.7015	-10.6599
-11.6217	-11.8149
-12.7751	-12.8037
-13.4433	-13.4215
-13.2426	-13.1845
-11.4619	-11.4535
-8.2448	-8.4612
-5.8635	-5.6108
-3.6970	-3.6860
-2.4371	-2.5319
-1.7974	-1.8380
-1.2967	-1.4028
-1.1205	-1.1164
-1.1004	-0.9195
1.5000	1.3515
1.5933	1.5931
1.6437	1.7416
1.6524	1.7752
1.6227	1.7211
1.5730	1.6243
1.5407	1.5166
1.4867	1.4130
1.4239	1.3188
1.3592	1.2348
1.2688	1.1603
1.1883	1.0942
1.0863	1.0353
1.0232	0.9825
0.9363	0.9350
0.8703	0.8919
0.7674	0.8526
0.6834	0.8167
0.6225	0.7836
-9.2077	-9.1190
-8.0398	-8.1612
-7.0590	-7.0409
-6.0224	-5.9605
-4.9818	-5.0206
-4.2492	-4.2490
-3.7254	-3.6341
-3.2104	-3.1495
-2.8004	-2.7678
-2.4630	-2.4652
-2.1631	-2.2231
-1.9390	-2.0273
-1.7970	-1.8673
-1.7187	-1.7351
-1.6006	-1.6247
-1.5198	-1.5317
-1.4685	-1.4526
-1.4097	-1.3848
-1.3651	-1.3262

Heat =

0.4159
-0.2019
0.0503
-0.7214
3.4328
8.0063
-6.5616
-19.2850
-15.3134

(5+4) Parameter Model 4

>> ParInd=fminsearch('ITCalgorithmIndependent',[2 2 1 1 .2],options,TotData,inj)

After 13 seconds (428 iterations / 712 evaluations of 'ITCalgortithmIndependent'), convergence criteria are satisfied, the error is 2.1216 and

ParInd =

0.9017 1.4730 0.7276 1.3105 0.2079

(5+4) Parameter Model 4*

>> ParModel4=fminsearch('ITCalgorithmModel4',[2 2 1 1 .2],options,TotData,inj)

After 30 seconds (420 iterations / 703 evaluations of 'ITCalgortithmModel4'), convergence criteria are satisfied, the error is 2.1216 and

ParModel4 =

0.9017 1.4730 0.7276 1.3105 0.2079

[Err4,Sim4,Heat4]=ITCalgorithmModel4(ParModel4,TotData,Inj)

Err4 =

2.1216

Sim4 =

2.8297 3.2909
2.2158 2.4080
1.5232 1.5269
0.8603 0.6543

0.2285	-0.1992
-0.6821	-1.0166
-1.6861	-1.7699
-2.6287	-2.4126
-3.0987	-2.8717
-3.3115	-3.0540
-3.1798	-2.8981
-2.5041	-2.4675
-2.0208	-1.9469
-1.5544	-1.4952
-1.0936	-1.1621
-0.7567	-0.9317
-0.5287	-0.7739
-0.4940	-0.6644
-0.4883	-0.5867
-2.1600	-2.9630
-4.5524	-4.5572
-6.7236	-6.1438
-8.1363	-7.7100
-9.3413	-9.2349
-10.7015	-10.6822
-11.6217	-11.9861
-12.7751	-13.0214
-13.4433	-13.5500
-13.2426	-13.1635
-11.4619	-11.4283
-8.2448	-8.5654
-5.8635	-5.7166
-3.6970	-3.6968
-2.4371	-2.4671
-1.7974	-1.7358
-1.2967	-1.2875
-1.1205	-1.0000
-1.1004	-0.8075
1.5000	1.3615
1.5933	1.4045
1.6437	1.4241
1.6524	1.4272
1.6227	1.4185
1.5730	1.4014
1.5407	1.3782
1.4867	1.3508
1.4239	1.3205
1.3592	1.2881
1.2688	1.2546
1.1883	1.2204
1.0863	1.1859

1.0232	1.1514
0.9363	1.1172
0.8703	1.0835
0.7674	1.0503
0.6834	1.0178
0.6225	0.9861
-9.2077	-8.8329
-8.0398	-7.7102
-7.0590	-6.7332
-6.0224	-5.8861
-4.9818	-5.1535
-4.2492	-4.5206
-3.7254	-3.9739
-3.2104	-3.5016
-2.8004	-3.0929
-2.4630	-2.7389
-2.1631	-2.4315
-1.9390	-2.1641
-1.7970	-1.9309
-1.7187	-1.7269
-1.6006	-1.5482
-1.5198	-1.3910
-1.4685	-1.2525
-1.4097	-1.1301
-1.3651	-1.0217

Heat4 =

-0.3016
-0.1824
-0.0516
0.0726
56.3874
179.2164
-57.6632
-200.5562

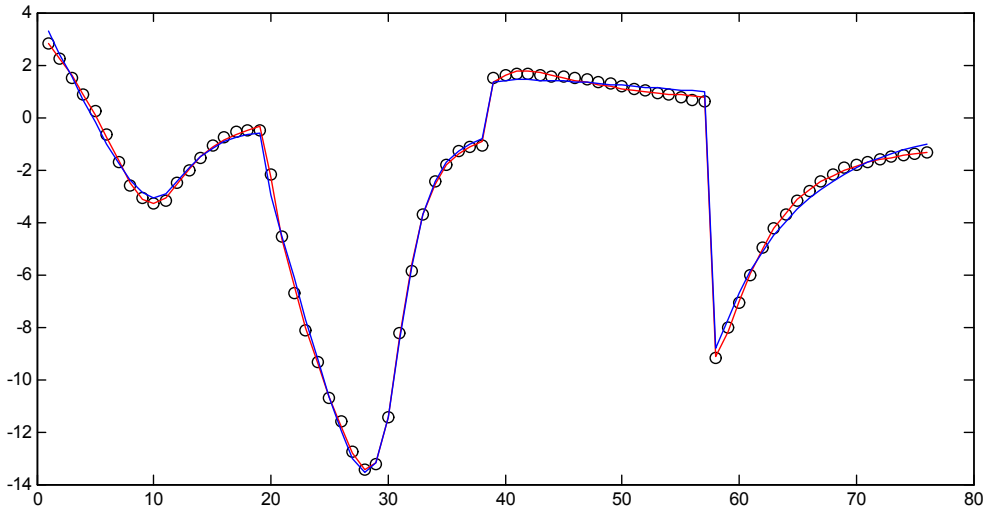

Figure S1: Fit of (7+5) parameter Model 3 (red) and (5+4) parameter model 4* (blue) to the ITC data. Note that in addition to an inferior fit, the enthalpy values obtained from Model 4 are of an unrealistically high magnitude.