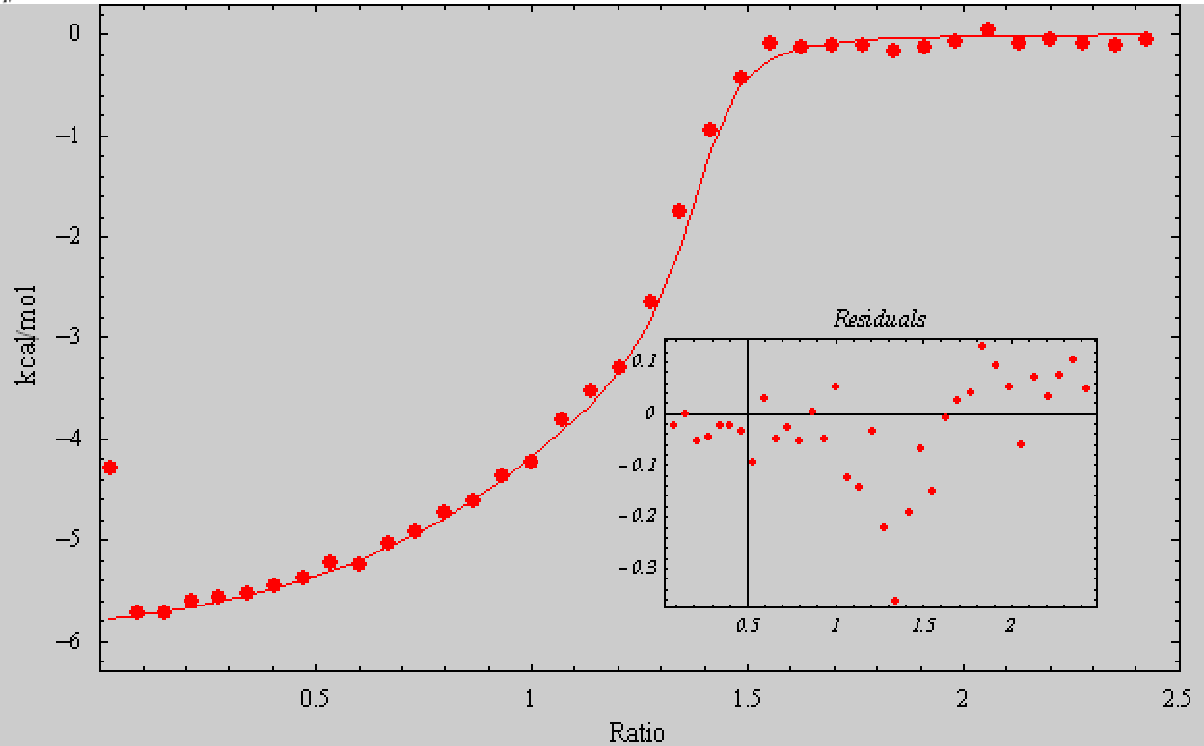
```
(* Created July 11, 2006 by Thomas Lundbäck *)

(* Load the required Mathematica Packages *)

Needs["Graphics`Graphics`"]
Needs["Graphics`Legend`"]
Needs["Statistics`NonlinearFit`"]

(* Import ITC titration data from Origin *)
(* A list of NDH values is created from the content of column "NDH" in the Microcal Origin software for data analysis *)
(* The list of values must be separated by commas and be placed between the {} brackets below *)
(* The imported data are divided by 1000 to convert values from cal/mol to kcal/mol *)
(* The parameters "NDHvaried" and "NDHerror" are used in the Monte Carlo simulations *)

NDH = NDHvaried = {-4279.7051, -5722.74979, -5706.34465, -5605.70374, -5561.38382, -5523.10812, -5455.46619, -5363.33982,
    -5214.25252, -5232.19545, -5030.77079, -4914.0371, -4729.54507, -4607.31788, -4355.41272, -4231.38595, -3808.40955, -3511.26552, -3287.29777, -2644.26606, -1748.06436,
    -942.57191, -430.95058, -89.26668, -126.17495, -109.16426, -96.53494, -169.42691, -121.138, -71.53377, 44.55209, -83.08268, -42.97158, -81.5833, -109.3263, -52.24775} / 1000;

NDHerror = 0.05; (* Estimated uncertainty in measured heats (kcal/mol) *)

(* Define starting values, titration setup and instrument parameters *)

Kd1 = 38.4;                 (* Starting value for Kd1 (μM) *)
Kd2 = 0.0844;               (* Starting value for Kd2 (μM) *)
H1 = -5.05;                 (* Starting value for ΔH1 (kcal/mol) *)
H2 = -6.52;                 (* Starting value for ΔH2 (kcal/mol) *)
n = 0.99;                   (* Starting value for stoichiometry or concentration correction factor n *)
f = 26.8;                   (* Starting value for the percentage of ligand X1 *)
Mcell = 113.;               (* Concentration in cell (μM) *)
Xsyr = 1.27 10^3;           (* Concentration in syringe (μM) *)
dV = 8.;                    (* Injection volume (μl) *)
dVstart = 3.;               (* Preliminary injection volume (μl) *)
NoInj = 35.;                (* Number of injections, not including the preliminary injection *)
V0 = 1440.9;                (* Calorimeter cell volume (μl) *)

(* Calculations of concentrations and definitions of results matrices for graphical display *)
(* Check that the calculated concentrations and molar ratios agree with those of the Microcal Origin software for data analysis *)

DVend = dVstart + NoInj dV;
Mtot = Table[Mcell (1 - v / (2 V0)) / (1 + v / (2 V0)), {v, dVstart, DVend, dV}];
Xtot = Table[v Xsyr (1 - v / (2 V0)) / V0, {v, dVstart, DVend, dV}];
Ratio = Xtot / Mtot;
RawData = Table[{Ratio[[i]], NDH[[i]]}, {i, Length[Ratio]}];
ExpRes = Table[{Mtot[[i]], Xtot[[i]], Mtot[[i - 1]], Xtot[[i - 1]], NDH[[i]]}, {i, 2, Length[Mtot]}];

(* Definition of a function for the calculation of heat exchange following an injection *)
(* The six input parameters are the dissociation constants (Kd1 and Kd2), the enthalpies (H1 and H2), the starting value of concentration correction factor (n), and the percentage of ligand X1 (f) *)
(* These calculations are based on: Wang (1995) An exact mathematical expression for describing competitive binding of two different ligands to a protein molecule. FEBS Letters 360, 111-114 *)

Calc[Kd1_, Kd2_, H1_, H2_, n_, f_] := (
c = -Kd1 Kd2 n Mtot;
b = Kd1 (Xtot (100 - f) / 100 - n Mtot) + Kd2 (Xtot f / 100 - n Mtot) + Kd1 Kd2;
a = Kd1 + Kd2 + Xtot - n Mtot;
CosTheta = (-2 a^3 + 9 a b - 27 c) / Sqrt[(a^2 - 3 b)^3] / 2;
Theta = ArcCos[CosTheta];
M = -a / 3 + 2 / 3 Cos[Theta / 3] Sqrt[a^2 - 3 b];
MX1 = Xtot M f / 100 / (Kd1 + M);
MX2 = Xtot M (100 - f) / 100 / (Kd2 + M);
Qtot = H1 V0 MX1 + H2 V0 MX2;
Qtot = Insert[Qtot, 0, 1];
DQ = Table[Qtot[[i]] - Qtot[[i - 1]] + dV (Qtot[[i]] + Qtot[[i - 1]]) / (2 V0), {i, 2, Length[Qtot], 1}];
Qfit = DQ / (dV Xsyr);
Qfit[[1]] = DQ[[1]] / (dVstart Xsyr);
Qfit)
```

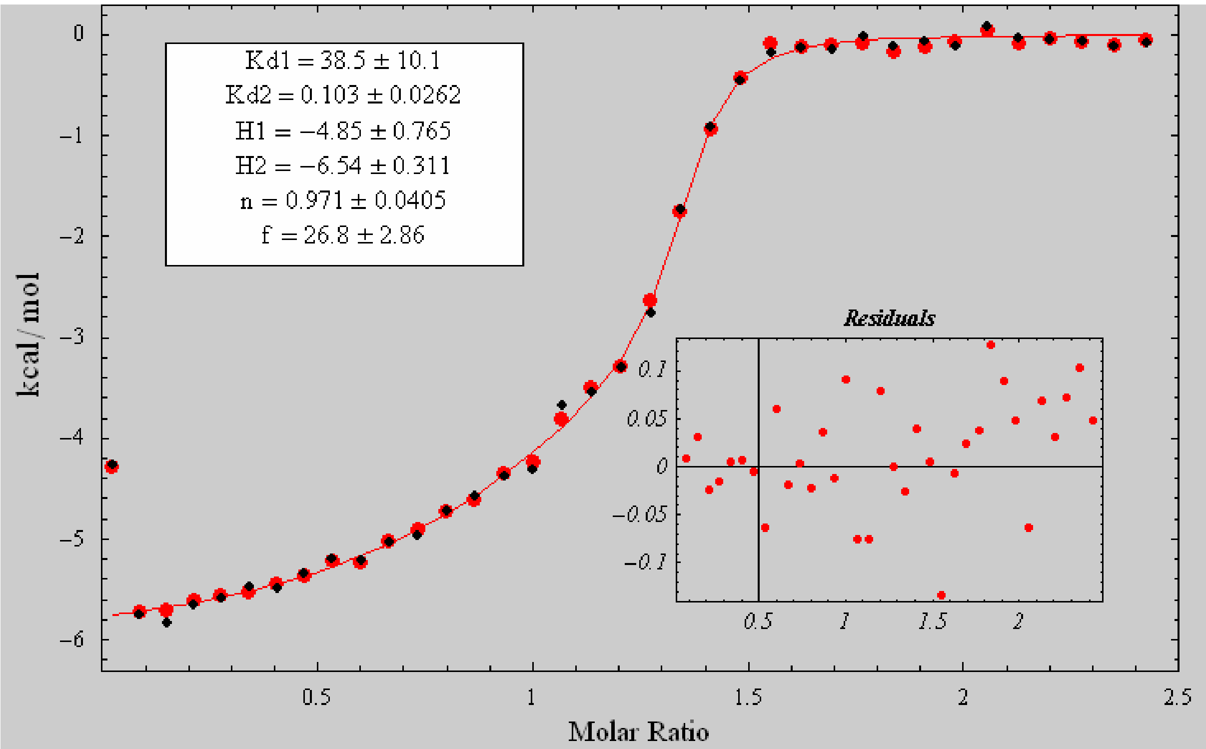
```
(* Calculations of heats based on the starting values for all parameters and graphical comparison of raw data and calculated data including an illustration of the residuals *)
(* A careful selection of appropriate starting parameters is required for a fit involving six unknowns *)
(* This is achieved by trial and error by going back to the cell in which the starting values are entered,
  new values are inserted and the cell is executed before going back to this cell to view the results of this process *)
(* Each user is responsible for the this process and it is strongly advised that control experiments are carried out to define as many as possible of these parameters before attempting to fit the data *)

Calc[Kd1, Kd2, H1, H2, n, f];
Res = Table[{Ratio[[i]], Qfit[[i]]}, {i, Length[Ratio]}];

Residuals = Table[{Ratio[[i]], Qfit[[i]] - NDH[[i]]}, {i, 2, Length[Ratio]}];
ResidualPlot = ListPlot[Residuals,
Prolog -> PointSize[0.02],
PlotStyle -> {RGBColor[1, 0, 0]},
Frame -> True,
PlotRange -> All,
DefaultFont -> {"Times-Italic", 10},
PlotLabel -> FontForm["Residuals", {"Times-BoldItalic", 12}],
DisplayFunction -> Identity];

DisplayTogether[
ListPlot[Res,
PlotJoined -> True,
PlotStyle -> {Thickness[0.002], RGBColor[1, 0, 0]},
Frame -> True,
Background -> GrayLevel[0.8],
DefaultFont -> {"Times", 12},
FrameLabel -> {FontForm["Ratio", {"Times", 14}], FontForm["kcal/mol", {"Times", 14}]},
PlotRange -> {{0, 2.5}, {-6.3, 0.3}},
Epilog -> {Rectangle[Scaled[{0.45, 0.05}], Scaled[{0.95, 0.55}], ResidualPlot]}],
ListPlot[RawData,
PlotStyle -> {PointSize[0.015], RGBColor[1, 0, 0]}]
];
```



```
(* Definition of a new function that is used by the Mathematica routine NonlinearRegress for the calculation of heat exchange following an injection *)

CalcRegress[Kd1_, Kd2_, H1_, H2_, n_, f_, M1_, X1_, M2_, X2_] := (

c = -Kd1 Kd2 n M1;
b = Kd1 (X1 (100 - f) / 100 - n M1) + Kd2 (X1 f / 100 - n M1) + Kd1 Kd2;
a = Kd1 + Kd2 + X1 - n M1;
CosTheta = (-2 a ^ 3 + 9 a b - 27 c) / Sqrt[(a ^ 2 - 3 b) ^ 3] / 2;
Theta = ArcCos[CosTheta];
M = -a / 3 + 2 / 3 Cos[Theta / 3] Sqrt[a ^ 2 - 3 b];
MX1 = X1 M f / 100 / (Kd1 + M);
MX2 = X1 M (100 - f) / 100 / (Kd2 + M);
Qtot1 = H1 V0 MX1 + H2 V0 MX2;

c = -Kd1 Kd2 n M2;
b = Kd1 (X2 (100 - f) / 100 - n M2) + Kd2 (X2 f / 100 - n M2) + Kd1 Kd2;
a = Kd1 + Kd2 + X2 - n M2;
CosTheta = (-2 a ^ 3 + 9 a b - 27 c) / Sqrt[(a ^ 2 - 3 b) ^ 3] / 2;
Theta = ArcCos[CosTheta];
M = -a / 3 + 2 / 3 Cos[Theta / 3] Sqrt[a ^ 2 - 3 b];
MX1 = X2 M f / 100 / (Kd1 + M);
MX2 = X2 M (100 - f) / 100 / (Kd2 + M);
Qtot2 = H1 V0 MX1 + H2 V0 MX2;

DQ = (Qtot1 - Qtot2 + dV (Qtot1 + Qtot2) / (2 V0)) / (dV Xsyr));

(* Performing the nonlinear least-square fit using the built-in function NonlinearRegress *)

Res = BestFitParameters /. NonlinearRegress[ExpRes,
CalcRegress[KKd1, KKd2, HH1, HH2, nn, ff, M1, X1, M2, X2],
{M1, X1, M2, X2},
{{KKd1, Kd1}, {KKd2, Kd2}, {HH1, H1}, {HH2, H2}, {nn, n}, {ff, f}},
ShowProgress -> True,
RegressionReport -> BestFitParameters];
Kd1fit = KKd1 /. Res;
Kd2fit = KKd2 /. Res;
H1fit = HH1 /. Res;
H2fit = HH2 /. Res;
nfit = nn /. Res;
ffit = ff /. Res;
Iteration:1   ChiSquared:0.367763   Parameters:{38.4, 0.0844, -5.05, -6.52, 0.99, 26.8}
Iteration:2   ChiSquared:0.115961   Parameters:{39.1911, 0.108455, -5.08699, -6.44181, 0.986165, 25.8555}
Iteration:3   ChiSquared:0.114402   Parameters:{39.0758, 0.104987, -4.91963, -6.51302, 0.974985, 26.4925}
Iteration:4   ChiSquared:0.114384   Parameters:{38.4384, 0.103412, -4.84533, -6.5404, 0.97105, 26.7803}
Iteration:5   ChiSquared:0.114384   Parameters:{38.4752, 0.103503, -4.85129, -6.53849, 0.971346, 26.7594}
Iteration:6   ChiSquared:0.114384   Parameters:{38.4696, 0.10349, -4.85057, -6.53875, 0.971307, 26.7621}
```

```
(* Estimation of standard deviations in the fitted parameters using a Monte Carlo simulation in which virtual data sets are created based on the estimated error in the measured heat exchange *)
(* This cell can be skipped if there is a need to save computer time *)
(* The number of virtual data sets can be modified using the "Iterations" variable
   below. Because of the required computation time it is recommended to keep this to a small number (<5) until appropriate estimates of the uncertainties are required *)
(* Note that these calculations are based on an estimated uncertainty that is the same for all data points throughout the titration
   (which is generally not the case). The calculations can easily be modified such that they are based on measured uncertainties when several titrations have been performed to define these *)

Iterations = 5;

Kd1res = Kd2res = H1res = H2res = nres = fres = {};
Do[
randerror = Table[Random[NormalDistribution[0, NDHerror]], {i, Length[NDH]}];
NDHvaried = NDH + randerror;
ExpRes = Table[{Mtot[[i]], Xtot[[i]], Mtot[[i - 1]], Xtot[[i - 1]], NDHvaried[[i]]}, {i, 2, Length[Mtot]}];
Res = BestFitParameters /. NonlinearRegress[ExpRes,
CalcRegress[KKd1, KKd2, HH1, HH2, nn, ff, M1, X1, M2, X2],
{M1, X1, M2, X2},
{{KKd1, Kd1}, {KKd2, Kd2}, {HH1, H1}, {HH2, H2}, {nn, n}, {ff, f}},
RegressionReport -> BestFitParameters];
Param1 = KKd1 /. Res;
Param2 = KKd2 /. Res;
Param3 = HH1 /. Res;
Param4 = HH2 /. Res;
Param5 = nn /. Res;
Param6 = ff /. Res;
Kd1res = Insert[Kd1res, Param1, 1];
Kd2res = Insert[Kd2res, Param2, 1];
H1res = Insert[H1res, Param3, 1];
H2res = Insert[H2res, Param4, 1];
nres = Insert[nres, Param5, 1];
fres = Insert[fres, Param6, 1],
{Iterations}];
stdKd1 = StandardDeviation[Kd1res];
stdKd2 = StandardDeviation[Kd2res];
stdH1 = StandardDeviation[H1res];
stdH2 = StandardDeviation[H2res];
stdn = StandardDeviation[nres];
stdf = StandardDeviation[fres];
```

```
(* Calculations of heats based on the best-fit values for all parameters and graphical comparison of raw data and calculated data including an illustration of the residuals *)

Calc[Kd1fit, Kd2fit, H1fit, H2fit, nfit, ffit];
 Res = Table[{Ratio[[i]], Qfit[[i]]}, {i, Length[Ratio]}];
ResIllustr = Table[{Ratio[[i]], NDHvaried[[i]]}, {i, Length[Ratio]}];

Residuals = Table[{Ratio[[i]], Qfit[[i]] - NDH[[i]]}, {i, 2, Length[Ratio]}];
ResidualPlot = ListPlot[Residuals,
Prolog -> PointSize[0.02],
PlotStyle -> {RGBColor[1, 0, 0]},
Frame -> True,
PlotRange -> All,
DefaultFont -> {"Times-Italic", 14},
PlotLabel -> FontForm["Residuals", {"Times-BoldItalic", 14}],
DisplayFunction -> Identity];

ResultPlot = Graphics[ShadowBox[{0, 0}, {1, 1},
ShadowOffset -> {0, 0}],
DefaultFont -> {"Times", 16},
Epilog -> {
Text[StringForm["Kd1 = `` ± ``", NumberForm[CForm[Kd1fit], 3], NumberForm[CForm[stdKd1], 3]], Scaled[{.5, 0.9}], {0, 0}],
Text[StringForm["Kd2 = `` ± ``", NumberForm[CForm[Kd2fit], 3], NumberForm[CForm[stdKd2], 3]], Scaled[{.5, 0.75}], {0, 0}],
Text[StringForm["H1 = `` ± ``", NumberForm[CForm[H1fit], 3], NumberForm[CForm[stdH1], 3]], Scaled[{.5, 0.6}], {0, 0}],
Text[StringForm["H2 = `` ± ``", NumberForm[CForm[H2fit], 3], NumberForm[CForm[stdH2], 3]], Scaled[{.5, 0.45}], {0, 0}],
Text[StringForm["n = `` ± ``", NumberForm[CForm[nfit], 3], NumberForm[CForm[stdn], 3]], Scaled[{.5, 0.3}], {0, 0}],
Text[StringForm["f = `` ± ``", NumberForm[CForm[ffit], 3], NumberForm[CForm[stdf], 3]], Scaled[{.5, 0.15}], {0, 0}]}];

DisplayTogether[
ListPlot[Res,
PlotJoined -> True,
PlotStyle -> {Thickness[0.002], RGBColor[1, 0, 0]},
Frame -> True,
Background -> GrayLevel[0.8],
DefaultFont -> {"Times", 14},
FrameLabel -> {FontForm["Molar Ratio", {"Times", 18}], FontForm["kcal / mol", {"Times", 18}]},
PlotRange -> {{0, 2.5}, {-6.3, 0.3}},
Epilog -> {Rectangle[Scaled[{0.45, 0.05}], Scaled[{0.95, 0.55}], ResidualPlot],
   Rectangle[Scaled[{0.05, 0.6}], Scaled[{0.4, 0.95}], ResultPlot]}],
ListPlot[RawData,
PlotStyle -> {PointSize[0.015], RGBColor[1, 0, 0]}],
ListPlot[ResIllustr,
PlotStyle -> {PointSize[0.01]}]
];
```


Kd1 = 38.5 ± 10.1
Kd2 = 0.103 ± 0.0262
H1 = −4.85 ± 0.765
H2 = −6.54 ± 0.311
n = 0.971 ± 0.0405
f = 26.8 ± 2.86
Residuals
kcal/mol
Molar Ratio