Data Repository Item

This document contains the input files for running the simulations shown in Figures 3–5 of Bethke et al., "Origin of microbiological zoning in groundwater flows". The simulations were run using program X1t of The Geochemist's Workbench® software package, version 7.0.2.

File **SRB.x1t** sets up the first simulation in the paper:

```
#  X1t input for simulating microbial activity and growth in an
#  aquifer containing sulfate-reducing bacteria and methanogens.

read AquiferBaseModel.x1t

#  Set a source of ferrous iron in the sediment, to react with sulfide
produced.

scope = initial
  swap FeO(ox) for Fe++
  .1 volume% FeO(ox)

scope = inlet
  swap FeO(ox) for Fe++

#  Run the model by entering "go".

suffix SRB
```

File **FeRB.x1t** sets up the second simulation:

```
# X1t input for simulating microbial activity and growth in an
# aquifer containing iron-reducing and sulfate-reducing bacteria and
# methanogens.

read AquiferBaseModel.x1t
suppress FeO(ox)

# Goethite in sediment supplies ferric iron to iron reducers, where
allowed
# by rate law (see file FeRB.cpp).

scope = initial
  swap Goethite for Fe+++
  .1 volume% Goethite

scope = inlet
  swap Goethite for Fe+++

# Iron reducers.  "surface" is a place holder for moles of biologically
# accessible ferric iron sites, per kg water.

kinetic microbe-FeRB \
  rxn = "CH3COO- + 8 Goethite + 15 H+ -> 12 H2O + 8 Fe++ + 2 HCO3-", \
  rate_law = "FeRB.dll:FeRB", \
```

```
  biomass = 10^-6, growth_yield = 9000, decay_con = 10^-9, \
   ATP_energy = -45, ATP_number = 4/3, order1 = 1/8, \
  mpower(CH3COO-) = 1, mpowerD(CH3COO-) = 1, \
  rate_con = 1.5e-5, KD = 1e-6, KA = 1e+6, surface = .03e-9

# Run the model by entering "go".

suffix FeRB
```

Both **SRB.x1t** and **FeRB.x1t** read file **AquiferBaseModel.x1t**:

```
#  Base model for reaction transport simulation of development
#  of microbial zoning in an aquifer.

#  Load chemical species considered.

data = thermo-aquifer.dat
suppress Pyrite, Troilite, Pyrrhotite, Hematite, Magnetite
decouple CH3COO-, CH4(aq), HS-, Fe+++

#  Aquifer properties.

time end     =  .1 m.y.
length       = 200 km
Nx           = 100
discharge    =  10 m/yr
porosity     =  30%
dispersivity =   1 m

#  Composition of recharging groundwater.

scope = inlet
  Ca++    =   1.    mmolal
  HCO3-   =   2.    mmolal
  SO4--   =    .05  mmolal
  CH3COO- =    .001 umolal
  HS-     =    .001 umolal
  CH4(aq) =    .001 umolal
  Fe++    =    1.   umolal
  pH = 6
  balance on HCO3-

#  Initial condition.

scope initial = inlet

#  Acetate supply from fermentation in fined-grained sediments.

react 4 umol/m3yr CH3COO-
react 2 umol/m3yr Ca++

#  Sulfate reducers.

kinetic microbe-SRB \
```

```
  rxn = "CH3COO- + SO4-- -> 2*HCO3- + HS-", \
  biomass = 10^-6, growth_yield = 4300, decay_con = 10^-9, \
  ATP_energy = -45, ATP_number = 1, order1 = 1/5, \
  mpower(CH3COO-) = 1, mpowerD(CH3COO-) = 1, \
  mpower(SO4--)   = 1, mpowerA(SO4--)   = 1, \
  rate_con = 10^-9, KD = 70e-6, KA = 200e-6

#  Methanogens.

kinetic microbe-Meth \
  rxn = "CH3COO- + H2O -> CH4(aq) + HCO3-", \
  biomass = 10^-6, growth_yield = 2000, decay_con = 10^-9, \
  ATP_energy = -45, ATP_number = 1/2, order1 = 1/2, \
  mpower(CH3COO-) = 1, mpowerD(CH3COO-) = 1, \
  rate_con = 3e-9, KD = 5e-3

#  Time weighting and plot increment.

theta = 1
dxplot .1 log
```

Dataset **thermo-aquifer.dat** is the GWB thermodynamic dataset **thermo.dat** with two minerals added:

```
…
  626 minerals
…

FeO(ox)                          type=
     formula=
     mole vol.=   12.000 cc      mole wt.=   71.8464 g
     3 species in reaction
      -2.000 H+              1.000 Fe++            1.000 H2O
       500.0000    6.0000  500.0000  500.0000
       500.0000  500.0000  500.0000  500.0000

FeS(s)                           type= sulfide
     formula= FeS
     mole vol.=   18.200 cc      mole wt.=   87.9070 g
     3 species in reaction
      -1.000 H+              1.000 Fe++            1.000 HS-
       500.0000   -8.0000  500.0000  500.0000
       500.0000  500.0000  500.0000  500.0000
```

To create library **FeRB.dll**, compile and link File **FeRB.cpp**:

```
// To compile and link under MS Windows:
//
//   cl /c FeRB.cpp
//   link /dll FeRB.obj x1t.lib /out:FeRB.dll
//

#define EXPORT  extern "C" __declspec(dllexport)
```

```
#include <windows.h>

#include <stdio.h>
#include <math.h>
#include "gwb_context.h"

// Rate law for FeRB metabolism and growth.

EXPORT double FeRB(int i, Reactant &r, GWBcontext &c)
{
   #include "ratelaw_param.h"
   double aff, ba_iron, prod, prodKd1, prodKd2;

   if (-1 == i) {

      // Thermodynamic factor.

      aff = QoverK * exp(-ATP_number * ATP_energy / (8.3143 * TK));
      aff = pow(aff, order1);
      aff = min(aff, 1.0);
      TPF = 1.0 - aff;

      // Promoting and inhibiting species.

      prod = 1.0;
      for (int j=0; j<ncatal; j++)
         prod *= pow(acatal[j], pcatal[j]);

      // F-D term, for acetate.

      prodKd1 = KD;
      prodKd2 = 1.0;
      for (int j=0; j<nspecD; j++)
         if (signD[j] > 0)
            prodKd1 *= pow(actD[j], powerD[j]);
         else if (signD[j] < 0)
            prodKd2 *= pow(actD[j], powerD[j]);

      FD = nspecD > 0 ? prodKd2 / pow(prodKd1 + prodKd2, PKD) : ANULL;

      // Update biomass.

      if (Deltat > 0.0) {
         biomass = biomass0 + (growth_yield*dndt0/Wmass -
                   decay_con*biomass0) * Deltat;
         biomass = min(biomass, 2.0*biomass0);
         biomass = max(biomass, 0.5*biomass0);
         biomass = max(biomass, 0.0);
      }

      // F-A term, for biologically accessible ferric iron. "sparea" is
a place holder
      //  for moles of available ferric sites per kg H2O, on LHS of
domain.  There is no
      //  ferric iron available on RHS.
```

```
      if (Ix > Nx/3) {
         ba_iron = 0.0;
         FA = 0.0;
      }
      else {
         ba_iron = sparea;
         FA = (biomass/ba_iron) / (KA + biomass/ba_iron);
      }

      // Return reaction rate.

      /*static double t0;
      if (Inode == 4 && c.Time != t0) {
         printf("biomass = %g, attached = %g, rate = %g\n", biomass,
attached, rprime*attached);
         //tty_pause();
         t0 = c.Time;
      }*/

      return Wmass * rate_con * ba_iron * (prod / (pow(prodKd1 +
prodKd2, PKD))) * FA * TPF;
   }
   else
      return ANULL;
}
```