User guide

This word file contains two MATLAB mscript files for optimization of spike and mass ratios respectively:

1. `Mg_OptimizeSp.m` (pg 2 to 5)
2. `Mg_OptimizeW.m` (pg 6 to 9)

And one function file which is used by the above mscript files:

`Mg_fsolve_equations.m` (pg 10)

Please save them as three different files in MATLAB, but store them in the same folder.

```
% Mg_OptimizeSp.m
% This file is used for the optimization of spike composition.
% The sample is treated as a reference standard since the value of fsample
does not affect the optimization
% Rx1 and Rx2 refer to the isotope ratio 25Mg/24Mg and 26Mg/24Mg of the
reference standard respectively.
% aMg24x, aMg25x and aMg26x refer to the abundances of Mg-24, Mg-25 and Mg-26
respectively in the reference standard
% aMg24y, aMg25y and aMg26y refer to the abundances of Mg-24, Mg-25 and Mg-26
respectively in the spike.
% Ry1 and Ry2 refer to isotope ratio 25Mg/24Mg and 26Mg/24Mg of the spike
respectively.
% W1 and W2 refer to the mass ratio of spike to sample in Blend 1 and Blend 2
respectively.
% M1 and M2 refer to the isotope mass ratio of 25Mg/24Mg and 26Mg/24Mg
respectively.
% RB1Mix1 and RB2Mix1 refer to the isotope ratios 25Mg/24Mg and 26Mg/24Mg of
Blend1 respectively.
% RB1Mix2 and RB2Mix2 refer to the isotope ratios 25Mg/24Mg and 26Mg/24Mg of
Blend2 respectively.
% M refers to the number of Monte Carlo Simulations used to solve for
fsample, Fmethod(blend1) and Fmethod(blend2)
% Error refers to the calculated combined standard uncertainty of fsample
% A_XX refers to the mole ratio calculated in file: % Mg_fsolve_equations.m,
XX refers to the corresponding blend and isotope ratio

clear all

tic

global Rx1 Rx2 aMg24x aMg25x aMg26x
global Ry1 Ry2 aMg24y aMg25y aMg26y
global W1 W2
global M1 M2
global RB1Mix1 RB2Mix1 RB1Mix2 RB2Mix2
global M Error
global A_mix1R1 A_mix1R2 A_mix2R1 A_mix2R2

fid = fopen('Optimum_spike.txt', 'w'); %save results into file.txt

%Model ratios
%Isotope ratio of Reference Std
%Rx1 = 25Mg/24Mg; Rx2 = 26Mg/24Mg

Rx1 = 0.12663;
Rx2 = 0.13932;

aMg24x = 1./(1+Rx1+Rx2);
aMg25x = Rx1.*aMg24x;
aMg26x = Rx2.*aMg24x;

%Isotopic mass
mMg24 = 23.98504187;
mMg25 = 24.985837;
mMg26 = 25.982593;
```

```
%Isotopic mass ratios
M1 = mMg25./mMg24;
M2 = mMg26./mMg24;

W1 = 0.5;
W2 = 5;

aMg25y_start = 0.01;
aMg25y_final = 0.99;

aMg26y_start = 0.01;
aMg26y_final = 0.99;

step = 0.01;

M = 1000;

RANDN_1 = randn(M,1); %store randn numbers
RANDN_2 = randn(M,1); %store randn numbers

aMg25y = aMg25y_start;

while aMg25y <= aMg25y_final

aMg26y = aMg26y_start;

while aMg26y <= aMg26y_final %while loop

aMg24y = 1-aMg26y-aMg25y;

if aMg24y> 0.0001

Ry1 = aMg25y./aMg24y;
Ry2 = aMg26y./aMg24y;

%Blend ratio of sample and spike by setting fsample and Fmethod = 0
%mRB1Mix1 refers to the mean of RB1Mix1, to be used as input variables for
%Monte Carlo Simulations (refer to Eqn. S-19). Since C = 1, it is left out
%in the equations below.

%Blend 1
mRB1Mix1 = (aMg25x + aMg25y.*W1)./(aMg24x + aMg24y.*W1);
mRB2Mix1 = (aMg26x + aMg26y.*W1)./(aMg24x + aMg24y.*W1);

%Blend 2
mRB1Mix2 = (aMg25x + aMg25y.*W2)./(aMg24x + aMg24y.*W2);
mRB2Mix2 = (aMg26x + aMg26y.*W2)./(aMg24x + aMg24y.*W2);

%Std dev of Blend ratios are calculated here:

RSD1 = 0.00091; %internal RSD after normalization (rep of TIMS technique)
RSD2 = 0.0018;
```

```
sdRB1Mix1= RSD1/100.*mRB1Mix1;
sdRB2Mix1= RSD2/100.*mRB2Mix1;

sdRB1Mix2= RSD1/100.*mRB1Mix2;
sdRB2Mix2= RSD2/100.*mRB2Mix2;

%MonteCarlo Simulation

%fsolve settings
options = optimset('Display','off','TolFun',1e-20); %Take note of Tolerance.
guess = zeros(3,1);


fsample = zeros(M,1);

    for i = 1:M,

    RB1Mix1 = sdRB1Mix1.*(RANDN_1(i)) + mRB1Mix1;

    RB2Mix1 = sdRB2Mix1.*(RANDN_1(i)) + mRB2Mix1;

    RB1Mix2 = sdRB1Mix2.*(RANDN_2(i)) + mRB1Mix2;

    RB2Mix2 = sdRB2Mix2.*(RANDN_2(i)) + mRB2Mix2;

    [x,fval]= fsolve(@Mg_fsolve_equations, guess, options);

    fsample(i) = x(2);

    end

    Error = std(fsample);

    aMg26y = aMg26y + step;

    fprintf(fid,'%g\n', Error);
    fprintf('%g\n', Error);

    else

    break %break loop if aMg24y less than 0.

end

end

    fprintf('the loop is done for , %g\n', aMg25y)
    aMg25y = aMg25y + step;

end
```

```
fclose(fid);

fid = fopen('Optimum_spike.txt');
A = fscanf(fid, '%g');

%Arrange vector A into matrix B
N = (aMg26y_final-aMg26y_start)/step;
B = NaN(N,N);
proto = flipud(tril(ones(N)));
idx = find(proto);
B(idx) = A;
fclose(fid);

X = aMg25y_start:step:(aMg25y_final-step);
Y = aMg26y_start:step:(aMg26y_final-step);

%truncate values greater than 0.001
subB = B(:,1:N) ;
subB(subB>0.0001) = 0.0001 ;
B(:,1:N) = subB;

% Create figure
figure1 = figure;

%create pseudocolor plot

pcolor(X,Y,B)

% Create xlabel
xlabel({'Abundance of ^25Mg(spike)'});

% Create ylabel
ylabel({'Abundance of ^26Mg(spike)'});

colorbar;

toc
```

```
% Mg_OptimizeW.m
% This file is used for the optimization of mass ratios denoted as W.
% The sample is treated as a reference standard since the value of fsample
does not affect the optimization
% Rx1 and Rx2 refer to the isotope ratio 25Mg/24Mg and 26Mg/24Mg of the
reference standard respectively.
% aMg24x, aMg25x and aMg26x refer to the abundances of Mg-24, Mg-25 and Mg-26
respectively in the reference standard
% aMg24y, aMg25y and aMg26y refer to the abundances of Mg-24, Mg-25 and Mg-26
respectively in the spike.
% Ry1 and Ry2 refer to isotope ratio 25Mg/24Mg and 26Mg/24Mg of the spike
respectively.
% W1 and W2 refer to the mass ratio of spike to sample in Blend 1 and Blend 2
respectively.
% M1 and M2 refer to the isotope mass ratio of 25Mg/24Mg and 26Mg/24Mg
respectively.
% RB1Mix1 and RB2Mix1 refer to the isotope ratios 25Mg/24Mg and 26Mg/24Mg of
Blend1 respectively.
% RB1Mix2 and RB2Mix2 refer to the isotope ratios 25Mg/24Mg and 26Mg/24Mg of
Blend2 respectively.
% M refers to the number of Monte Carlo Simulations used to solve for
fsample, Fmethod(blend1) and Fmethod(blend2)
% Error refers to the calculated combined standard uncertainty of fsample
% A_XX refers to the mole ratio calculated in file: % Mg_fsolve_equations.m,
XX refers to the corresponding blend and isotope ratio

clear all

tic

global Rx1 Rx2 aMg24x aMg25x aMg26x
global Ry1 Ry2 aMg24y aMg25y aMg26y
global W1 W2
global M1 M2
global RB1Mix1 RB2Mix1 RB1Mix2 RB2Mix2
global M Error
global A_mix1R1 A_mix1R2 A_mix2R1 A_mix2R2

fid = fopen('Optimum_W.txt', 'w'); %save results into file.txt

Rx1 = 0.12663;
Rx2 = 0.13932;

aMg24x = 1./(1+Rx1+Rx2);
aMg25x = Rx1.*aMg24x;
aMg26x = Rx2.*aMg24x;

%Isotopic mass
mMg24 = 23.98504187;
mMg25 = 24.985837;
mMg26 = 25.982593;

%Isotopic mass ratios
M1 = mMg25./mMg24;
M2 = mMg26./mMg24;
```

```
%Spike composition
aMg25y = 0.1;
aMg26y = 0.8;
aMg24y = 1- aMg25y - aMg26y;

Ry1 = aMg25y./aMg24y;
Ry2 = aMg26y./aMg24y;

W1_start = 0.1;
W1_end = 10.1;

W2_start = 0.2;
W2_end = 10.2;

step1 = 0.2;
step2 = 0.2;


W1 = W1_start;

M = 1000;

RANDN_1 = randn(M,1); %store randn numbers
RANDN_2 = randn(M,1); %store randn numbers

while W1 <= W1_end;

   W2 = W2_start;

while W2 <= W2_end;

%Blend ratio of sample and spike by setting fsample and Fmethod = 0
%mRB1Mix1 refers to the mean of RB1Mix1, to be used as input variables for
%Monte Carlo Simulations (refer to Eqn. S-19). Since C = 1, it is left out
%in the equations below.

%Blend 1
mRB1Mix1 = (aMg25x + aMg25y.*W1)./(aMg24x + aMg24y.*W1);
mRB2Mix1 = (aMg26x + aMg26y.*W1)./(aMg24x + aMg24y.*W1);

%Blend 2
mRB1Mix2 = (aMg25x + aMg25y.*W2)./(aMg24x + aMg24y.*W2);
mRB2Mix2 = (aMg26x + aMg26y.*W2)./(aMg24x + aMg24y.*W2);

%Std dev of Blend ratios are calculated here:

RSD1 = 0.00091; %internal RSD after normalization (rep of TIMS technique)
RSD2 = 0.0018;

sdRB1Mix1= RSD1/100.*mRB1Mix1;
sdRB2Mix1= RSD2/100.*mRB2Mix1;
```

```
sdRB1Mix2= RSD1/100.*mRB1Mix2;
sdRB2Mix2= RSD2/100.*mRB2Mix2;

%fsolve settings
options = optimset('Display','off','TolFun',1e-20); %Take note of Tolerance.
guess = zeros(3,1);

%Measured ratios of 25Mg/24Mg and 26Mg/24Mg are correlated.

fsample = zeros(M,1);

    for i = 1:M,

    RB1Mix1 = sdRB1Mix1.*RANDN_1(i) + mRB1Mix1;

    RB2Mix1 = sdRB2Mix1.*RANDN_1(i) + mRB2Mix1;

    RB1Mix2 = sdRB1Mix2.*RANDN_2(i) + mRB1Mix2;

    RB2Mix2 = sdRB2Mix2.*RANDN_2(i) + mRB2Mix2;

    [x,fval]= fsolve(@Mg_fsolve_equations, guess, options);

    fsample(i) = x(2);

    end

    Error = std(fsample);

    W2 = W2+step2;

    fprintf(fid,'%g\n', Error);
    fprintf('%g\n', Error);

end

W1 = W1+step1;

end

fclose(fid);

fid = fopen('Optimum_W.txt');
A = fscanf(fid, '%g');

%Arrange vector A into matrix B
N = sqrt(length(A));
B = vec2mat(A,N);
fclose(fid);

%truncate values
subB = B(:,1:N);
subB(subB>0.00001) = 0.00001;
```

```
B(:,1:N) = subB;

%create figure
figure1 = figure;

J = 0.1: 0.2: 10.1;
K = 0.2: 0.2: 10.2;

pcolor(J,K,B)

% Create xlabel, inner loop
xlabel({'Mass ratio of spike to sample, W_1'});

% Create ylabel, outer loop
ylabel({'Mass ratio of spike to sample, W_2'});

colorbar;

toc
```

```
function F = Mg_fsolve_equations(x);  %#ok<NOSEM>

% This file contains the function used to solve the non-linear equations
involving the unknowns:
% x(1) = Fmethod(blend1)
% x(2) = fsample
% x(3) = Fmethod(blend2)

global A_mix1R1 A_mix1R2 A_mix2R1 A_mix2R2
global RB1Mix1 RB2Mix1 RB1Mix2 RB2Mix2
global Ry1 Ry2 Rx1 Rx2  Difference_mix1 Difference_mix2
global W1 W2
global M1 M2
global Sum_W Sum_Wsq Sum_A Sum_AW intercept

M1 = 1.04172580291602;
M2 = 1.08328320379121;

A_mix1R1 = (RB1Mix1.*M1.^-x(1)-Rx1.*M1.^x(2))./(Ry1 - RB1Mix1.*M1.^-x(1));
A_mix1R2 = (RB2Mix1.*M2.^-x(1)-Rx2.*M2.^x(2))./(Ry2 - RB2Mix1.*M2.^-x(1));

A_mix2R1 = (RB1Mix2.*M1.^-x(3)-Rx1.*M1.^x(2))./(Ry1 - RB1Mix2.*M1.^-x(3));
A_mix2R2 = (RB2Mix2.*M2.^-x(3)-Rx2.*M2.^x(2))./(Ry2 - RB2Mix2.*M2.^-x(3));

Sum_A = A_mix1R1 + A_mix1R2 + A_mix2R1 + A_mix2R2;
Sum_AW = A_mix1R1.*W1 + A_mix1R2.*W1 + A_mix2R1.*W2 + A_mix2R2.*W2;
Sum_Wsq = W1.^2 + W1.^2 + W2.^2 + W2.^2;
Sum_W = W1 + W1 + W2 + W2;

intercept = Sum_A.*Sum_Wsq - Sum_W.*Sum_AW;

Difference_mix1= A_mix1R1 - A_mix1R2;
Difference_mix2= A_mix2R1 - A_mix2R2;

F=[Difference_mix1; Difference_mix2; intercept];
```