# Matlab code

function result=MCEE(obj)

% Author: Yitao Li, Tianlu Chen

% Date: 2017-9-5

% Maintainer: Tianlu Chen (chentianlu@sjtu.edu.cn)

% matlab R2014a

%INPUTS

%obj: a structure containing raw data (metabolites), confounding and instruction variables

%DATA TO USE

%obj.M: metabolite concentrations, each column in obj.M is a variable (I x J)

%obj.Y: classification variables, each column in obj.Y is a variable (I x G)

%obj.F: confounding variables, each column in obj.Y is a variable (I x K)

%OUTPUTS

%result.T: the decomposed, orthogonal or independent components;

%%%result.W: T = XW; the vector used to obtain the orthogonal or independent components

%%%result.P: TP' is the estimation of X based on T; the matrix TP' has the same size as X

%result.Z: Z = X - T\*P'; Z is the "corrected" X with less or no relationship with F

%result.time: CPU time for correction

tic

ss=0.2;%%%%%%%%%%%%%%%%%%%%%%%%% Adjustable parament

%%%%% GLM on M and F

c=size(obj.M,2);

cc=size(obj.Y,2);

ccc=size(obj.F,2);

Pcrude\_n=zeros(c,ccc);% p values from normal model

Pcrude\_p=zeros(c,ccc);

Pcrude\_ig=zeros(c,ccc);

Pcrude\_g=zeros(c,ccc);

for i=1:c

x=obj.M(:,i);

for j=1:ccc

y=obj.F(:,j);

b\_n = glmfit(x,y,'normal');

yfit\_n = glmval(b\_n,x,'identity');

[~, p\_n] = chi2gof(yfit\_n);

Pcrude\_n(i,j)=p\_n;

b\_p = glmfit(x,y,'poisson');

yfit\_p = glmval(b\_p,x,'log');

[~, p\_p] = chi2gof(yfit\_p);

Pcrude\_p(i,j)=p\_p;

b\_ig = glmfit(x,y,'inverse gaussian');

yfit\_ig = glmval(b\_ig,x,-2);

[~, p\_ig] = chi2gof(yfit\_ig);

Pcrude\_ig(i,j)=p\_ig;

b\_g = glmfit(x,y,'gamma');

yfit\_g = glmval(b\_g,x,'reciprocal');

[~, p\_g] = chi2gof(yfit\_g);

Pcrude\_g(i,j)=p\_g;

end

end

d\_n=Pcrude\_n<0.01;

d\_p=Pcrude\_p<0.01;

d\_ig=Pcrude\_ig<0.01;

d\_g=Pcrude\_g<0.01;

d=d\_n+d\_p+d\_ig+d\_g;% variables with p<0.01 from any one of the 4 models were labeled 1.

inx0=find(sum(d,2)>=ss); % indexes of variables related to F (union of all factors). Some of them were also related to Y.

inx00=inx0;

siz=size(obj.Y);

sizz=siz(2);

allinx=cell(size(obj.M,2),sizz\*2)

for isize=1:sizz

for i12=1:2

group11=find(obj.Y(:,isize)==i12);

Y11=obj.Y(group11,:);

F11=obj.F(group11,:);

M11=obj.M(group11,:);

c11=size(M11,2);

cc11=size(Y11,2);

ccc11=size(F11,2);

Pcrude\_n11=zeros(c11,ccc11);% p values from normal model

Pcrude\_p11=zeros(c11,ccc11);

Pcrude\_ig11=zeros(c11,ccc11);

Pcrude\_g11=zeros(c11,ccc11);

for i11=1:c11

x11=M11(:,i11);

for j11=1:ccc11

y11=F11(:,j11);

b\_n11 = glmfit(x11,y11,'normal');

yfit\_n11 = glmval(b\_n11,x11,'identity');

[~, p\_n11] = chi2gof(yfit\_n11);

Pcrude\_n11(i11,j11)=p\_n11;

b\_p11 = glmfit(x11,y11,'poisson');

yfit\_p11 = glmval(b\_p11,x11,'log');

[~, p\_p11] = chi2gof(yfit\_p11);

Pcrude\_p11(i11,j11)=p\_p11;

b\_ig11 = glmfit(x11,y11,'inverse gaussian');

yfit\_ig11 = glmval(b\_ig11,x11,-2);

[~, p\_ig11] = chi2gof(yfit\_ig11);

Pcrude\_ig11(i11,j11)=p\_ig11;

b\_g11 = glmfit(x11,y11,'gamma');

yfit\_g11 = glmval(b\_g11,x11,'reciprocal');

[~, p\_g11] = chi2gof(yfit\_g11);

Pcrude\_g11(i11,j11)=p\_g11;

end

end

d\_n11=Pcrude\_n11<0.01;

d\_p11=Pcrude\_p11<0.01;

d\_ig11=Pcrude\_ig11<0.01;

d\_g11=Pcrude\_g11<0.01;

d11=d\_n11+d\_p11+d\_ig11+d\_g11;% variables with p<0.01 from any one of the 4 models were labeled 1.

inx11=find(sum(d11,2)>=ss);

inx00= union(inx00,inx11)

end

end

inx=union(inx0,inx00);

%%%% GLM on M and Y

for i22=1:c

x22=obj.M(:,i22);

for j22=1:cc

y22=obj.Y(:,j22);

b\_n22 = glmfit(x22,y22,'normal');

yfit\_n22 = glmval(b\_n22,x22,'identity');

[~, p\_n22] = chi2gof(yfit\_n22);

Pcrude\_n22(i22,j22)=p\_n22;

b\_p22 = glmfit(x22,y22,'poisson');

yfit\_p22 = glmval(b\_p22,x22,'log');

[~, p\_p22] = chi2gof(yfit\_p22);

Pcrude\_p22(i22,j22)=p\_p22;

b\_ig22 = glmfit(x22,y22,'inverse gaussian');

yfit\_ig22 = glmval(b\_ig22,x22,-2);

[~, p\_ig22] = chi2gof(yfit\_ig22);

Pcrude\_ig22(i22,j22)=p\_ig22;

b\_g22 = glmfit(x22,y22,'gamma');

yfit\_g22 = glmval(b\_g22,x22,'reciprocal');

[~, p\_g22] = chi2gof(yfit\_g22);

Pcrude\_g22(i22,j22)=p\_g22;

end

end

d\_n22=Pcrude\_n22<0.01;

d\_p22=Pcrude\_p22<0.01;

d\_ig22=Pcrude\_ig22<0.01;

d\_g22=Pcrude\_g22<0.01;

d22=d\_n22+d\_p22+d\_ig22+d\_g22;% variables with p<0.01 from any one of the 4 models were labeled 1.

for j22=1:cc

inx22=find(d22(:,cc)>=1);

pp(:,1)=inx22;

pp(:,2)=Pcrude\_n22(inx22,cc);

pp(:,3)=Pcrude\_p22(inx22,cc);

pp(:,4)=Pcrude\_ig22(inx22,cc);

pp(:,5)=Pcrude\_g22(inx22,cc);

pp0 = pp(:,2:5);

pp(:,6)=mean(pp0,2);

[~,index]=sort(pp(:,6));

pp1=pp(index,:);

number=round(ss\*size(pp1,1));

pp2=pp1(1:number,1);

pp3=intersect(inx,pp2);

inx(ismember(inx,pp3))=[];

end

%% remove factor-related component from best fitted model, and model selection was done by R2

% model selection

B=[];

YFIT=[];

I=[];

RSQ=[];

for i=1:length(inx)

xx=[obj.Y,obj.F];

yy=obj.M(:,inx(i));

b1=glmfit(xx,yy,'normal');% normal distribution

yfit1 = glmval(b1,xx,'identity');

% [h, p] = chi2gof(yfit);

yresid1 = yy - yfit1;

SSresid1 = sum(yresid1.^2);

SStotal1 = (length(yy)-1) \* var(yy);

rsq1 = 1 - SSresid1/SStotal1;

b2=glmfit(xx,yy,'poisson');% poisson distribution

yfit2 = glmval(b2,xx,'log');

% [h, p] = chi2gof(yfit);

yresid2 = yy - yfit2;

SSresid2 = sum(yresid2.^2);

SStotal2 = (length(yy)-1) \* var(yy);

rsq2 = 1 - SSresid2/SStotal2;

b3=glmfit(xx,yy,'inverse gaussian');% inverse gaussian distribution

yfit3 = glmval(b3,xx,-2);

% [h, p] = chi2gof(yfit);

yresid3 = yy - yfit3;

SSresid3 = sum(yresid3.^2);

SStotal3 = (length(yy)-1) \* var(yy);

rsq3 = 1 - SSresid3/SStotal3;

b4=glmfit(xx,yy,'gamma');% gamma distribution

yfit4 = glmval(b4,xx,'reciprocal');

% [h, p] = chi2gof(yfit);

yresid4 = yy - yfit4;

SSresid4 = sum(yresid4.^2);

SStotal4 = (length(yy)-1) \* var(yy);

rsq4 = 1 - SSresid4/SStotal4;

R2=[rsq1,rsq2,rsq3,rsq4];

[rsq,index]=max(R2);

RSQ=[RSQ,rsq];

I=[I,index];

% final model saved in B and YFIT

switch index

case 1

B=[B,b1];

YFIT=[YFIT,yfit1];

submode='normal';

case 2

B=[B,b2];

YFIT=[YFIT,yfit2];

submode='poisson';

case 3

B=[B,b3];

YFIT=[YFIT,yfit3];

submode='inverse gaussian';

case 4

B=[B,b4];

YFIT=[YFIT,yfit4];

submode='gamma';

otherwise

warning('Model selection failed!');

end

end

obj.T=obj.M(:,inx);% variables need revision

obj.Z=obj.M;% corrected data

for i=1:length(inx)

for j=1:ccc

obj.Z(:,inx(i))=obj.Z(:,inx(i))-B((j+1),i)\*obj.F(:,j);

end

end

obj.timeCPU.GLM = toc;

result.T=obj.T;

result.Z=obj.Z;

%% add the abs(min) of each column

min\_z=min(result.Z);

for i=1:size(obj.Z,2)

a=find(result.Z(:,i)<0);

if(a)

result.Z(:,i)= result.Z(:,i)+abs(min\_z(i));

end

end