

```

load DataLinseed
x = Data( : , 1 : 8 );% Model inputs
Fx = Data( : , end );% Model output (Seed yield)
%% Normalization
[xn, xs] = mapminmax(x, -1, 1); [Fxn, Fxs] = mapminmax(Fx, -1, 1);
%% Create the network
hiddenSizes = 11; net = feedforwardnet( hiddenSizes );
%% Transfer function
net.layers{1}.transferFcn='tansig';
net.layers{2}.transferFcn='purelin';
net = configure( net, xn, Fxn );% Configure the network
%% performance function: Mean squared error(mse)
net.performFcn = 'mse';
%% Set up Division of Data for Training, Validation, Testing
net.divideFcn = 'dividerand';
net.divideParam.trainRatio = 0.80; % training set
net.divideParam.valRatio = 0.10; % validation set
net.divideParam.testRatio = 0.10; % test set
%% Train phase
net.trainFcn = 'trainlm';
[ net, resultout ] = train( net, xn, Fxn );
%% Network output
pvn = sim( net, xn );%The values predicted by the MLP

```

Figure S1. MATLAB code for modeling linseed product performance with MLP

```

load DataLinseed
x = Data( : , 1 : 8 );% Model inputs
Fx = Data( : , end );% Model output (Seed yield)
ModelSpec='quadratic';%'linear','interactions','purequadratic','quadratic'
mlr = fitlm( x(trainInd,:), Fx(trainInd,:),ModelSpec,'Intercept',true);
pv = predict(mlr,x); %The values predicted by the MLR

```

Figure S2. MATLAB code for modeling linseed product performance using MLR

```

Options=gaoptimset('PopulationType','doubleVector','PopultionSize',100,
'SelectionFcn','selectiontournament','CrossoverFcn','crossoverscattered','MutationFcn','Gene
rations',1000,'Vectorize','on');
[x, fval]=ga( @Fmlp,nVar, [ ], [ ], [ ], [ ],LB ,UB,[ ],[ ],Options );
%%
function out = Fmlp( x )
    x = x'; global net, load MLP
    xn = mapminmax('apply', x, xs ); pvn = sim(net, xn);
    out = -1 * mapminmax('reverse', pvn, Fxs );
end

```

Figure S3. MATLAB code for modeling linseed product performance using MLR