**Supplementary Information**

**Surrogate approach to determine heavy metal loads in a moss species - *Barbula lambaranensis***

Clement Oluseye Ogunkunlea, Abdul M. Ziyathb, Saheed Sunkanmi Rufaia, Paul Ojo Fatobaa

aDepartment of Plant Biology, University of Ilorin, Ilorin, 240003 Nigeria.

bScience and Engineering Faculty, Queensland University of Technology, GPO Box 2434, Brisbane QLD 4001, Australia.

**E-mail:** seyeogunkunle@gmail.com; mohamed.ziyath@qut.edu.au; mailto:p.egodawatta@qut.edu.ausunlovecity@yahoo.com; pofatoba@unilorin.edu.ngmailto:g.ayoko@qut.edu.au

\*Corresponding Author: Abdul M. Ziyath; Tel: 61 4 0141 9767; Fax: 61 7 3138 1170

email: mohamed.ziyath@qut.edu.au



Investigated heavy metals

Heavy metal Load (mg/kg moss)

**Fig. S1: Box plots for heavy metals ions present in the moss samples**



**Fig. S2: Q-Q plots for (a) Zn; (b) Cd; (c) Cu; (d) Pb; (e) Ni; (f) Cr**

**Data Analysis Techniques**

1. **Principal Component Analysis (PCA)**

PCA is a common data exploration technique that transforms a set of correlated variables to an orthogonal set of principal components (PCs). The first PC accounts for the highest variance of the data matrix and the variance explained gradually reduces for the subsequent PCs. Therefore, first two PCs are often considered for data interpretation since they can be presented as two dimensional biplots that are easier to interpret. It is worthy to note that excluding the other PCs often result in the loss of useful data contained in the original data matrix.

Biplots are a combination of scores and loading plots. The object-object relationships are illustrated by the scores plot, while the loading plot explains the variable-variable relationship. Consequently, combining both scores and loading plots as biplot is useful in identifying and interpreting underlying relationships between objects and variables (Mostert et al., 2010). A detailed description of PCA can be found elsewhere (Adams, 1995).

1. **Design of Experiment**

Design of experiment or experimental design is a chemometric technique that facilitates the systematic study of the effects of variables on a response (Brereton 2003). A four level full factorial design that is used in this study varies the independent variables (factors) in four levels, namely 1, 2, 3 and 4. In a controlled laboratory environment, it is relatively easy to define each level with a fixed value. For example, if the study were to investigate the influence of pH and temperature on the product yield, the four levels for pH and temperature can be chosen as per Table S1. In full factorial design, each possible combination of factors are included the study as shown in Table 1 in the manuscript.

**Table S1: Four levels for independent variables**

|  |  |  |
| --- | --- | --- |
| **Level** | **pH** | **Temperature (0C)** |
| 1 | 6 | 20 |
| 2 | 7 | 40 |
| 3 | 8 | 60 |
| 4 | 9 | 80 |

In contrast to laboratory experiments, field experiments are affected by various external factors. Consequently, it is difficult to define the levels precisely. Consequently, in this study, the levels for the independent variables are chosen based on their quartiles such that the average of first quartile is Level 1, the average of second quartile is Level 2 and so on.

1. **Multiple Linear Regression**

MLR is a regression technique that is often used to develop mathematical relationships for a dependent variable based on a number of independent variables (Ni et al., 2001).

1. **Matlab® codes used for data analysis**

% TotalMatrixRaw was constructed by extracting the raw data from Microsoft® Excel® using xlsread function

TotalMatrixRaw = [ZnRaw, CdRaw, CuRaw, PbRaw, NiRaw, CrRaw];

%Outlier detection

figure;

boxplot(TotalMatrixRaw,'whisker',3,'symbol','ro',... 'Labels',{'Zn','Cd','Cu','Pb','Ni','Cr'})

h=findobj(gcf,'tag','Outliers');

xc = get(h,'XData');

yc = get(h,'YData');

OutliersTotalMatrix = [horzcat(xc{:}), horzcat(yc{:})];

%Outlier free data matrix ‘TotalMatrix’ was constructed by extracting the outlier free data from Microsoft® Excel® using xlsread function

TotalMatrix = [Zn, Cd, Cu, Pb, Ni, Cr];

%Testing normality using the Q-Q plots

figure;

subplot (2,3,1)

qqplot(Zn)

subplot (2,3,2)

qqplot(Cd)

subplot (2,3,3)

qqplot(Cu)

subplot (2,3,4)

qqplot(Pb)

subplot (2,3,5)

qqplot(Ni)

subplot (2,3,6)

qqplot(Cr)

% Identification of surrogate indicators using PCA

figure;

[pc,score,latent] = princomp(zscore(TotalMatrix));

VariExplained = cumsum(latent)./sum(latent)

biplot(pc(:,1:2),'Scores',score(:,1:2),...

'VarLabels',{'Zn' 'Cd' 'Cu' 'Pb' 'Ni' 'Cr'})

%Two factor four level full factorial experimental deisgn

dff = fullfact([4 4])

%MLR by calling a function CVMLR, the code for which is given below

IndependentVariables = [ones(size(Pb)) Pb Cu];

[bCd,RPECd,SECVCd,Q2Cd] = CVMLR (Cd,IndependentVariables);

[bZn,RPEZn,SECVZn,Q2Zn] = CVMLR (Zn,IndependentVariables);

[bNi,RPENi,SECVNi,Q2Ni] = CVMLR (Ni,IndependentVariables);

[bCr,RPECr,SECVCr,Q2Cr] = CVMLR (Cr,IndependentVariables);

b = [bCd bZn bNi bCr]

RPE = [RPECd RPEZn RPENi RPECr]

SECV = [SECVCd SECVZn SECVNi SECVCr]

Q2 = [Q2Cd Q2Zn Q2Ni Q2Cr]

--------------------------------------------------------------------

function [b,RPE,SECV,Q2] = CVMLR (y,x)

%CVMLR returns multiple linear regression coefficients (b) with relative

%prediction error (RPE), standard error of cross-validation (SECV) and

%cross-validated coefficient of determination (Q2). The input y is the data

%matrix containing dependent variables and x is the data matrix containing

%independent variables.

b = regress (y,x); %Matlab built-in function that performs multiple linear

%regression and returns the regression coefficients matrix, b. The first

%element of matrix b is the constant followed by regression coefficeints.

PredictMatrix = [];

for n = 1:length(y)

%Data matrix for model development, consists of all data points except one

 TestIndex = n;

 TrainIndex = setdiff(1:length(y),TestIndex);

 X=x(TrainIndex,:);

 Y=y(TrainIndex,1);

 %Prediction of Y using regression coefficient for the one left out sample

 PredictedY = [x(TestIndex,:)]\*b;

 %Data matrix consisting of measured and predicted values

 MeasuredVsPredict = [y(TestIndex,1), PredictedY];

 PredictMatrix = [PredictMatrix; MeasuredVsPredict];

end

[NSamples NVariables] = size(X);

MeasuredY = PredictMatrix(:,1);

PredictedY = PredictMatrix(:,2);

%Relative error of prediction

RPE = 100\*(sqrt (sumsqr (PredictedY-MeasuredY)/(sumsqr(MeasuredY))));

% Standard error of cross validation

SECV = sqrt(NSamples/(NSamples-1))\*sqrt(sumsqr(MeasuredY-PredictedY)/NSamples);

%Cross-validated R2

Q2 = 100\*(1-(sumsqr (PredictedY-MeasuredY)/sumsqr(MeasuredY-mean(MeasuredY))));

**References**

Adams, M.J., 1995. Chemometrics in Analytical Spectroscopy. Royal Society of Chemistry, Chemistry, Cambridge [England].

Brereton, R.G., 2003. Chemometrics: Data analysis for laboratory and chemical plant. John Wiley & Sons, Ltd, West Sussex.

Mostert, M.M.R., Ayoko, G.A., Kokot, S. 2010. Application of chemometrics to analysis of soil pollutants. Trends Analyt. Chem. 29, 430–445.

Ni, Y., Wang, L., Kokot, S., 2001. Simultaneous determination of nitrobenzene and nitro-substituted phenols by differential pulse voltammetry and chemometrics. Analyt Chim Acta 431, 101–113.