

# 1                   Supplementary Materials:

## 2                   PLS2 IN METABOLOMICS

3       Matteo Stocchero, Emanuela Locci, Ernesto d'Aloja, Matteo Nioi, Eugenio Baraldi and Giuseppe  
4       Giordano

5       **Basic R-functions**

6       **PLS2 and ptPLS2**

7       Input: matrix X of the predictors, matrix Y of the responses, number A of latent variables of the  
8       model.

9       Output: score matrix T, score matrix of the post-transformed model Tpt, weight matrix W, matrix  
10      Wstar ( $W^*$ ), matrix of the regression coefficient B.

```

11      PLS2<-function(X,Y,A) {
12         W<-matrix(rep(0,A*ncol(X)),nrow=ncol(X),ncol=A)
13         P<-matrix(rep(0,A*ncol(X)),nrow=ncol(X),ncol=A)
14         T<-matrix(rep(0,A*nrow(X)),nrow=nrow(X),ncol=A)
15         E<-X
16         for (i in 1:A) {
17             W[i]<-svd(t(Y)%*%E)$v[,1]
18             T[i]<-E%*%W[i]
19             P[i]<-t(E)%*%T[i]/sum(T[i]*T[i])
20             E<-E-T[,i]%*%t(P[,i])
21         }
22         Ws<-W%*%solve(t(P)%*%W)
23         B<-Ws%*%solve(t(T)%*%T)%*%t(T)%*%Y
24         d1<-svd(t(Y)%*%X%*%W)
25         Np<-length(which(d1$d>10^-8))
26         G<-step.2(X,Y,W)
27         if (A<=Np) Tpt<-T
28         if (A>Np) Tpt<-step.3(X,Y,W,G)$Tpt
29         pls2<-list(T=T,Tpt=Tpt,W=W,Wstar=Ws,B=B)
30         return(pls2)
31     }
32     step.2<-function(X,Y,W) {
33         G<-matrix(rep(0,ncol(W)*ncol(W)),ncol=ncol(W))
34         I<-diag(rep(1,ncol(W)))
35         d1<-svd(t(Y)%*%X%*%W)
36         N<-length(which(d1$d>10^-8))
37         V<-Re(d1$v[,1:N])

```

```

38     d2<-eigen((I-V%*%t(V))%*%(t(W)%*%t(X)%*%X%*%W))
39     M<-length(which(Re(d2$values)>10^-8))
40     Go<-Re(d2$vectors[,1:M])
41     Gp<-Re(eigen(t(I-Go%*%t(Go))%*%(I-Go%*%t(Go))) $vectors[,1:(ncol(W)-M)])
42     G<-cbind(Go,Gp)
43     return(G)
44 }
45 step.3<-function(X,Y,W,G) {
46   Tpt<-matrix(rep(0,nrow(X)*ncol(W)),ncol=ncol(W))
47   E<-X
48   F<-Y
49   Wpt<-W%*%G
50   for (i in 1:ncol(W)) {
51     Tpt[,i]<-E%*%Wpt[,i]
52     Q<-Tpt[,i]%*%t(Tpt[,i])/sum(Tpt[,i]^2)
53     E<-E-Q%*%E
54     F<-F-Q%*%F
55   }
56   Ppt<-t(X)%*%Tpt%*%solve(t(Tpt)%*%Tpt)
57   Wspt<-Wpt%*%solve(t(Ppt)%*%Wpt)
58   ptmodel<-list(Tpt=Tpt,Wpt=Wpt,Wspt=Wspt)
59   return(ptmodel)
60 }
61 oCPLS2
62 Input: matrix X of the predictors, matrix Y of the responses, matrix Z of the constraints, number A of
63 latent variables of the model.
64 Output: score matrix T, score matrix of the post-transformed model Tpt, weight matrix W, matrix
65 Wstar ( $W^*$ ), matrix of the regression coefficient B.
66 oCPLS2<-function(X,Y,Z,A) {
67   W<-matrix(rep(0,A*ncol(X)),nrow=ncol(X),ncol=A)
68   P<-matrix(rep(0,A*ncol(X)),nrow=ncol(X),ncol=A)
69   T<-matrix(rep(0,A*nrow(X)),nrow=nrow(X),ncol=A)
70   B<-t(Z)%*%X
71   h<-svd(B)
72   R<-length(which(h$d>10^-8))
73   V<-h$v[1:R]
74   Q<-diag(rep(1,ncol(X)))-V%*%t(V)
75   E<-X
76   F<-Y
77   for (i in 1:A) {

```

```

78     W[,i]<- svd(t(F)%%E%%Q)$v[,1]
79     T[,i]<-E%%W[,i]
80     P[,i]<-t(E)%%T[,i]/sum(T[,i]*T[,i])
81     E<-E-T[,i]%%t(P[,i])
82     F<-F-T[,i]%%t(T[,i])%%F/sum(T[,i]*T[,i])
83   }
84   Ws<-W%%solve(t(P)%%W)
85   B<-Ws%%solve(t(T)%%T)%%t(T)%%Y
86   d1<-svd(t(Y)%%X%%W)
87   Np<-length(which(d1$d>10^-8))
88   G<-step.2(X,Y,W)
89   if (A<=Np) Tpt<-T
90   if (A>Np) Tpt<-step.3(X,Y,W,G)$Tpt
91   ocpls2<-list(T=T,Tpt=Tpt,W=W,Wstar=Ws,B=B)
92   return(ocpls2)
93 }

```

#### 94 KPLS2

95 Input: matrix X of the predictors, matrix Y of the responses, number A of latent variables of the  
 96 model, (a, b, c) parameters of the polynomial kernel  $k(x,y)=(a(x^t y)+b)^p$ , matrix Xtest of the test set to  
 97 be predicted

98 Output: normalized score matrix Tn, normalized score matrix of the test set Tnpred, calculated  
 99 response Ycalc, predicted response Ypred

```

100 KPLS2<-function(X,Y,A,a,b,p,Xtest) {
101   Tn<-matrix(rep(0,nrow(X)*A),ncol=A)
102   U<-matrix(rep(0,nrow(Y)*A),ncol=A)
103   K<-K.matrix(X,a,b,p)
104   F<-Y
105   for (i in 1:A) {
106     Tn[,i]<-Re(eigen(K%%F%%t(F))$vectors[,1])
107     U[,i]<-F%%t(F)%%Tn[,i]
108     Q<-diag(1,nrow(X))-Tn[,i]%%t(Tn[,i])
109     K<-Q%%K%%Q
110     F<-Q%%F
111   }
112   Ymod<-Y-F
113   K<-K.matrix(X,a,b,p)
114   Kp<-K.matrixpred(Xtest,X,a,b,p)
115   Ypred<-Kp%%U%%solve(t(Tn)%%K%%U)%%t(Tn)%%Y
116   Tnpred<-Kp%%U%%solve(t(Tn)%%K%%U)
117   r<-list(Tn=Tn ,Tnpred=Tnpred, Ycalc=Ymod, Ypred=Ypred)
118   return(r)

```

```
119 }
120 K.matrix<-function(X,a,b,p) {
121   Km<-matrix(rep(-999,nrow(X)*nrow(X)),nrow=nrow(X))
122   for (i in 1:nrow(X)) {
123     for (j in 1:nrow(X)) {
124       Km[i,j]<-(a*t(X[i,])%*%X[j,]+b)^p
125     }
126   }
127   p<-matrix(rep(1,nrow(X)),ncol=1)
128   Qc<-diag(1,nrow(X))-(1/nrow(X))*p%*%t(p)
129   Km<-Qc%*%Km%*%Qc
130   return(Km)
131 }
132 K.matrixpred<-function(X1,X2,a,b,p) {
133   Kp<-matrix(rep(-999,nrow(X1)*nrow(X2)),nrow=nrow(X1))
134   K<-matrix(rep(-999,nrow(X2)*nrow(X2)),nrow=nrow(X2))
135   for (i in 1:nrow(X1)) {
136     for (j in 1:nrow(X2)) {
137       Kp[i,j]<-(a*t(X1[i,])%*%X2[j,]+b)^p
138     }
139   }
140   for (i in 1:nrow(X2)) {
141     for (j in 1:nrow(X2)) {
142       K[i,j]<-(a*t(X2[i,])%*%X2[j,]+b)^p
143     }
144   }
145   p<-matrix(rep(1,nrow(X2)),ncol=1)
146   pp<-matrix(rep(1,nrow(X1)),ncol=1)
147   Qc<-diag(1,nrow(X2))-(1/nrow(X2))*p%*%t(p)
148   P<-(1/nrow(X2))*pp%*%t(p)
149   Kp<-(Kp-P%*%K)%*%Qc
150   return(Kp)
151 }
```