MedeA QT: An Interactive QSPR Toolbox

Contents

1 Introduction

MedeA QT, the *MedeA* QSPR Toolbox, employs an interactive graphical user interface to allow you to explore and analyze the relationships between descriptors and system properties. *MedeA QT* employs statistical methods, such as partial least squares (PLS), to compute correlations.

2 The Partial Least Squares (PLS) and Ordinary Least Squares (OLS) Methods

Partial Least Squares (PLS) [1], [2], [3], [4], [5] was developed by Herman and Svante Wold and is now widely used in chemometrics and related areas. A PLS model is based on the determination of multidimensional directions in descriptor space that provide the maximum variance in both descriptor and activity space. Additional information on PLS modeling is provided below. PLS regression is particularly suited to situations that are challenging for ordinary least squares (OLS) methods. (Incidentally, Ordinary Least Squares (OLS) and Multiple Linear Regression (MLR) are identical methods).

For example, where there is collinearity in supplied descriptors, and where there are relatively few activities relative to descriptors, ordinary least squares will likely require the handling of ill-conditioned matrices, and the OLS algorithm may not yield stable models. The PLS method is generally numerically stable and can handle duplicated descriptor columns, for example, that would lead to difficulties in matrix operations for OLS least-squares model-building algorithms.

The difference between OLS and PLS can be summarized as follows. OLS adjusts a set of descriptor coefficients, given a fixed descriptor space, to minimize the squared deviations between estimated and observed activities. In contrast, PLS seeks to maximally span descriptor and activity space, and generate correlations between descriptor space and activity space. PLS tackles these three objectives in the inherent optimization problem that the algorithm solves to create a predictive model.

The numerical basis of the PLS method can be summarized as follows. In the OLS case, the essential equation to be solved is as follows:

[1] A. Boulesteix, K. Strimmer, "Partial least squares: a versatile tool for the analysis of high-dimensional genomic data", *Briefings in bioinformatics*, **8** (2006)

[2] R. Rosipal, N. Krämer, "Overview and recent advances in partial least squares", In *Subspace, latent structure and feature selection*, (pp. 34-51), Springer, Berlin, Heidelberg (2006)

[3] R. Tobias, "An introduction to partial least squares regression", In *Proceedings of the twentieth annual SAS users group international conference* (pp. 1250-1257), SAS Institute Inc. Cary, NC (1995)

[4] S. Wold, A. Ruhe, H. Wold, W.J. Dunn, "The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses", *SIAM Journal on Scientific and Statistical Computing*, **5**, 735 (1984)

[5] S. Wold, P. Geladi, K. Esbensen, J. Öhman, "Multiway principal components and PLS analysis", *Journal of chemometrics*, **1**, 41 (1987)

$$y = Xb + e \tag{1}$$

where *X* is a matrix of descriptors for each measured activity, *b* is a vector of regression coefficients, *y* is the vector of activities, and *e* is a vector of errors. This equation is typically and straightforwardly solved forming normal equations and using matrix inversion, yielding the vector of regression coefficients, *b*.

PLS uses a decomposition of the *X* matrix and the *y* vector in the following manner:

$$X = TP^T + E \tag{2}$$

$$y = Tq + f \tag{3}$$

Here *T* is known as a score matrix and *P* as a loading matrix, terminology which originates in principal component regression (PCR) which is related to PLS. *E* and *f* are matrix and vector of residuals, respectively. The PLS algorithm determines the vectors that comprise *T* and *U* based on the maximization of their variance, and this maximization provides the simultaneous multi-objective optimization that provides PLS with its properties.

Given (2), then

$$XW = TP^TW \tag{4}$$

where *W* is a weight matrix. Hence *T* is

$$T = XW(P^TW)^{-1} \tag{5}$$

and from (3),

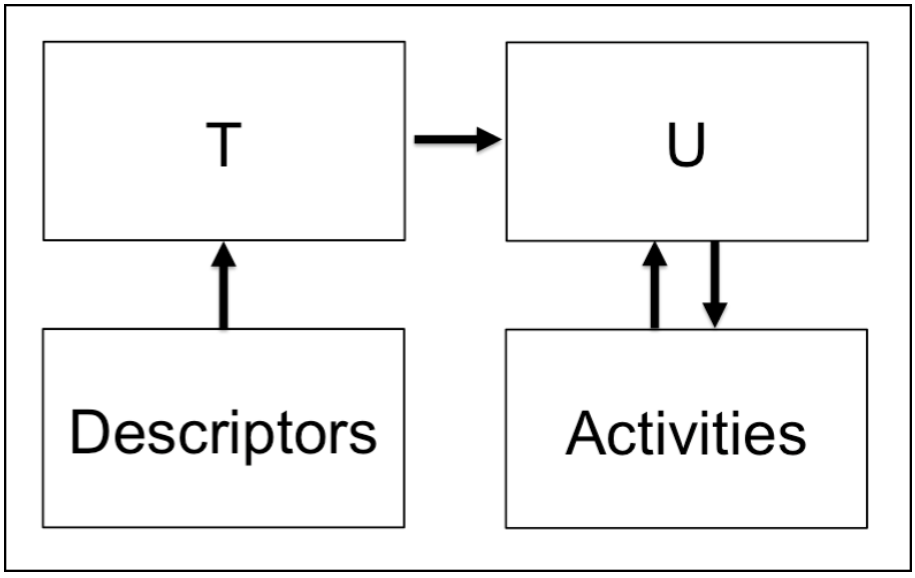
$$y = Tq = XW(P^TW)^{-1}q \tag{6}$$

as *y* = *Xb*, the PLS correlation coefficients, *b*, are:

$$b = W(P^TW)^{-1}q \tag{7}$$

Residual terms, which do not affect the transformations, are omitted from the equations above.

Herman and Svante Wold and other researchers have shown that focusing on the three objectives in the PLS algorithm (spanning descriptor and activity spaces as well as maximization of correlation) results in a modeling approach with desirable characteristics, including numerical stability and robustness with respect to outlying descriptors. Recognizing that effectively PLS transforms the supplied descriptor space has resulted in an alternative interpretation of the PLS acronym: 'Projection to Latent Structures', which is less widely used than 'Partial Least Squares' but perhaps more accurately describes the method. 'Partial Least Squares' is employed here as this term is more widely employed in the literature.



The diagram above illustrates the essence of the PLS method. The 'descriptor' space supplied to describe the systems of interest is converted into a set of latent directions, labeled **T** in the diagram. Activities are also transformed into a latent or derived form, labeled **U** in the diagram. (Note that the labels in the diagram are not reflective of the equation variables used above). The key point is that the transformations leading to **T** and **U** reduce the dimensionality of the original descriptor space. In PLS, the order of the reduced dimensionality employed is known as the 'Latent Factors Rank', this order is automatically selected by *MedeA QT* in normal use, but can also be set by the user. When the 'Latent Factor Rank' is equal to the number of descriptors, PLS is equivalent to ordinary least squares (OLS). The key benefit of the dimensionality contraction inherent in PLS is the reduction in collinearity difficulties in creating correlative models.

Several algorithms have been described to generate the score and loading matrices. These are typically either iterative or recursive and from an initial starting build the matrices step by step.

The algorithm employed by *MedeA QT* is known as Nonlinear Iterative Partial Least Squares and proceeds through the simple regression of descriptor vectors from the *X* matrix to create the vectors of the *P* and *Q* matrices. The algorithm terminates when a defined number of latent variables have been employed or no further correlation between *y* and *t* vectors is obtained. Extensive additional information on the PLS method is available in the literature.

It is worth noting that the technical literature related to PLS is diverse for several reasons. The method has been developed comparatively recently, it is applied in many fields each with differing terminologies, and a wide range of algorithmic variants of PLS have been, and continue to be, developed. Several selected publications are listed below that provide an overview of the method.

Note: *MedeA QT* transforms supplied data prior to creating a model by mean centered normalization using the mean and standard deviation of a given property type. This standardization improves the numerical behavior and stability of the operations employed in the OLS and PLS procedures used by *MedeA QT*. The model saved is represented in its standard form. Hence, for each observation and activity employed in the model, the mean of the set of observations and their standard deviations will also be reported, in addition to the coefficients of the mean centered and normalized model.

To employ the model in normalized form, you first normalize the properties for which you wish to determine an activity, by subtracting the mean and dividing by the standard deviation for that property, then you multiply the normalized terms by the model coefficients reported by *MedeA QT*.

These transformations are handled automatically in both the interactive and flowchart implementations of *MedeA QT*.

Alternatively, if you wish to view or use the model in non-mean centered, non-normalized form, you can simply transform the model using the following equations:

$$x_i = x_{inormalized} \cdot y_{istdev} / x_{istdev} \tag{8}$$

$$c = \overline{y} - \sum_i x_i \overline{x_i} \tag{9}$$

Where x_i is the i^{th} non-normalized model coefficient, $x_{inormalized}$ is the i^{th} normalized model coefficient, y_{istdev} is the standard deviation of the supplied activities, and x_{istdev} the standard deviation of the i^{th} property, $\overline{y}$ is the mean of the activities supplied to *MedeA QT*, $\overline{x_i}$ is the mean of the i^{th} property and c is the intercept of the non-normalized model.

The xml file produced by *MedeA QT* allows you to employ the generated model outside the *MedeA* environment. For example, consider a simple three parameter model created using *MedeA QT* to describe the refractive index of organic liquids which is represented in the *MedeA QT* model xml file as follows:

```
<Activity>
  <Mean>1.471736</Mean>
  <StdDev>0.0891799</StdDev>
  <Name>Refractive Index</Name>
</Activity>
<LatentFactors>
  <Automatic>yes</Automatic>
  <Rank>2</Rank>
</LatentFactors>
<Factors>
  <Factor>
    <Mean>595.445</Mean>
    <StdDev>74.9170</StdDev>
    <Coefficient>0.489374</Coefficient>
    <Descriptor>Tc</Descriptor>
  </Factor>
  <Factor>
    <Mean>-44.9522</Mean>
    <StdDev>194.4090</StdDev>
    <Coefficient>0.2566745</Coefficient>
    <Descriptor>Hf</Descriptor>
  </Factor>
  <Factor>
    <Mean>33.9467</Mean>
    <StdDev>171.0877</StdDev>
    <Coefficient>0.324600</Coefficient>
    <Descriptor>Gf</Descriptor>
  </Factor>
</Factors>
```

activity

T_c

H_f

G_f

A section from a *MedeA QT* model xml file, showing the model parameters and the mean and standard deviations of descriptors and activity employed in the creation of the model.

Hence, if you wish to compute the refractive index of a molecule with the following properties *T_c* = 682.33, *H_f* = -70.99, and *G_f* = 136.38 (outside the *MedeA* environment), you can either normalize and mean center each of these properties and use the model's factor coefficients to compute its activity, or create a QSPR equation for the non-normalized properties. Illustrations of both approaches follow.

Computing the properties using the factor coefficients reported by *MedeA QT*, the mean centered normalized properties are:

$$T_c' = (682.33 - 595.445)/74.9170 = 1.15975012 \tag{10}$$

$$H_f' = (-70.99 + 44.9522)/194.409 = -0.1339331 \tag{11}$$

$$G_f' = (136.38 - 33.9467)/171.088 = 0.598718084 \tag{12}$$

And the activity, *A*, predicted by the model is:

$$A = 0.0891799 \cdot (0.4894 \cdot 1.15975012 - 0.2567 \cdot 0.1339331 + 0.3246 \cdot 0.598718084) + 1.471736 \tag{13}$$

or

$$A = 1.5366184 \tag{14}$$

Alternatively, the model may be converted into its non-mean centered, non-standardized form, using the

equations above, as follows:

$$c_1 = 0.0891799 \cdot 0.4894/74.9170 = 0.000582573 \tag{15}$$

$$c_2 = 0.0891799 \cdot 0.2567/194.409 = 0.000117754 \tag{16}$$

$$c_3 = 0.0891799 \cdot 0.3246/171.088 = 0.000169199 \tag{17}$$

$$c_0 = 1.471736 - (0.000582573 \cdot 595.445 - 0.000117754 \cdot 44.9522 + 0.000169199 \cdot 33.9467) \tag{18}$$

or

$$c_0 = 1.124395226 \tag{19}$$

And the activity, *A*, predicted by the model is:

$$A = 1.124395226 + 0.000582573 \cdot 682.33 - 0.000117754 \cdot 70.99 + 0.000169199 \cdot 136.38 \tag{20}$$

or

$$A = 1.5366183 \tag{21}$$

Hence the two representations are identical and can be inter-converted using the information in the *MedeA QT* model xml file. This illustration also shows how the non-mean centered, non-standardized form may employ mathematical operations with numbers of different magnitudes, which can affect numerical results.