Double Cross-Validation Software Tool for MLR QSAR Model Development

The double cross-validation process comprises two nested cross-validation loops which are referred as internal and external cross-validation loops. In the outer (external) loop of double cross-validation, all data objects are divided into two subsets referred to as **training** and **test** sets. The training set is used in the inner (internal) loop of double cross-validation for model building and model selection, while the test set is exclusively used for model assessment. So in the internal loop, the training set is repeatedly split into **calibration** and **validation** data sets. The calibration objects are used to develop different models whereas the validation objects are used to estimate the models' error. Finally, the model with the lowest prediction errors (validation set) in the inner loop is selected (*Note that the selection of a model should not be based on the performance of the model on the test set because a test set validation is used to simulate the performance of the model on an external set which is unseen and thus any use of the test set in selection of the model or model descriptors does not represent a true external validation*). Then, the test objects in the outer loop are employed to assess the predictive performance of the selected model. This method of multiple splits of the training set into calibration and validation sets obviates the bias introduced in variable selection in case of usage of a single training set of fixed composition **[1]**.

Double Cross-validation Tool ***version 2.0*** performs multiple linear regression (MLR) model development using the double cross-validation process as mentioned above. Additionally and optionally, this tool can also simultaneously develop PLS models using NIPALS algorithm **[2]**. *Note that in this tool the optimal number of components for the PLS models are selected based on the fact that further addition of an additional component does not increase leave-one-out(LOO)* Q^2 *value for the training set by at least 5%.* Now, the user has to provide the training and test sets (descriptors and the response variable) information in the input file. In this tool, the calculations are divided into **2 steps**:

Step 1: ***Development of 'k' models based on k-fold cross-validation of the training set and all the information of 'k' models (MLR equations, validation parameters etc.) are stored in the output folder.***

This tool performs ***k*-fold cross-validation** in which the data objects are split, based on the sorted response values, into *k* subsets of approximately equal size. Each of the *k* subsets is omitted once (*used as the validation set*) and the remaining data (*used as the calibration set*) are used to develop the model. Thus, *k* models are built and each model is validated with the omitted data subset. The value of '*k*' is user defined, while the default value is kept **10**.

Two variable selection methods are available in this tool, i.e., Stepwise-MLR (S-MLR) and Genetic Algorithm (GA). On successful execution of **step 1**, the output files will be generated in the user defined output folder. User can now check the "***_AllModels_Summary.csv***" file to select the superior model based on the internal and external validation metrics.

Step 2: ***Selection of the optimum model using 3 different ways as mentioned below:***

1. Consensus model predictions: Here user has to provide 2 or more (let say '***n***') model numbers (between 1 to k) in the respective text field provided in the tool. The tool will use these selected models to predict '***n***' response values for each test object. Further, the average of all '***n***' predictions is employed to evaluate the quality of predictions for the test set.

2. *Model with the least MAE (Validation set):* Here, the tool will select the model with the least mean absolute error (MAE) for the validation set. This selected model is then used to check the prediction quality of the test set.

3. Model with the best descriptor combination (For MLR models): Here, user has to provide the model numbers (between 1 to k) in the respective text field provided in the tool. The tool will first find the unique descriptors (i.e. non-redundant) and then develop all possible subsets (*number of descriptors in each subset will be user defined*) out of these unique number of descriptors in the selected models. Finally, the tool will select

top 10 models (best subsets) based on prediction quality of the training set (i.e. lowest cross-validated error) and store the respective information in the output folder.

4. *Model with the Pooled descriptor*: For PLS models, one can develop PLS model with pooled descriptors from the top models. Here, user has to provide the model numbers (between 1 to k; top models) in the respective text field provided in the tool. The tool will first find the unique descriptors (i.e. non-redundant) present in the selected models and then develop a PLS model from pooled unique descriptors.

Double Cross-Validation Program Folder

The program folder will consist of three folders "**Data**", "**Lib**" and "**Output**". For user convenience, user may keep input files in the "**Data**" folder and may save output file in "**Output**" folder. "**Lib**" folder consists of library files required for running the program. Check the format of training set and test set input files (*.xlsx/.xls/.csv*) before using the program (*sample file provided in* **'Data'** *Folder*; *see the next section to understand the format of input files*). "**Lib**" folder also consist of a descriptor database file ("***DescriptorDatabase.xlsx***") with basic information about descriptors calculated using cerius2, dragon and PaDEL software. This information is used to display brief description about each selected descriptors in the output files.

How to prepare the input files, i.e., Training and Test sets

It is easy to prepare the required input files (*2 files*), i.e., Training and Test sets. There are 3 commonly used file formats (or extensions) that are recognized by the tool, namely, ***.csv***, ***.xls***, ***.xlsx***. One can easily store the relevant data in these file formats using **Microsoft Excel/Open Office** software. Now the data in these files are stored in a definite way to maintain the uniformity and thus assist the tool to extract the data in a proper manner. Thus, the input data comprises of three components i.e. compound number (or serial number), descriptors and response variable (*activity/property*) information. Now these three components should be arranged in the following way:

First Row. **Header** *i.e.* name for each column, for instances, descriptor names, and response variable name. *It can be numerical, alphabet or alphanumerical in nature.*

First column: Serial number/Compound number (*only numerical values*)

Subsequent columns: Descriptors (Independent variables) (*only numerical values*)

Last column: Response variable (Dependent variable) (*only numerical values*)

Note: For further clarification, please check the sample input files provided in the "**Data**" Folder.

Reference:

1. *Baumann, D. and Baumann, K., 2014*. Reliable estimation of prediction errors for QSAR models under model uncertainty using double cross-validation. *J. Cheminformatics*, *6*(1), p.47.
2. *S. Wold, M. Sjostrom, L. Eriksson*, PLS-regression: a basic tool of chemometrics, *Chemometrics and Intelligent Laboratory Systems*, 58 (2001) 109-130.

Java External Library Used

Apache POI – the Java API for Microsoft Documents

- Available at http://poi.apache.org/

XMLBeans

- Available at http://xmlbeans.apache.org/

Contact us at the following addresses:

Dr. Kunal Roy,

Drug Theoretics and Cheminformatics Lab.,

Dept. of Pharmaceutical Technology,

Jadavpur University,

Kolkata, West Bengal,

INDIA-700032

Email Id: kunalroy_in@yahoo.com

Software Developer details:

Pravin Ambure,

Research Scholar,

Drug Theoretics and Cheminformatics Lab.,

Dept. of Pharmaceutical Technology,

Jadavpur University,

Kolkata, West Bengal,

INDIA-700032

E-mail Id: ambure.pharmait@gmail.com