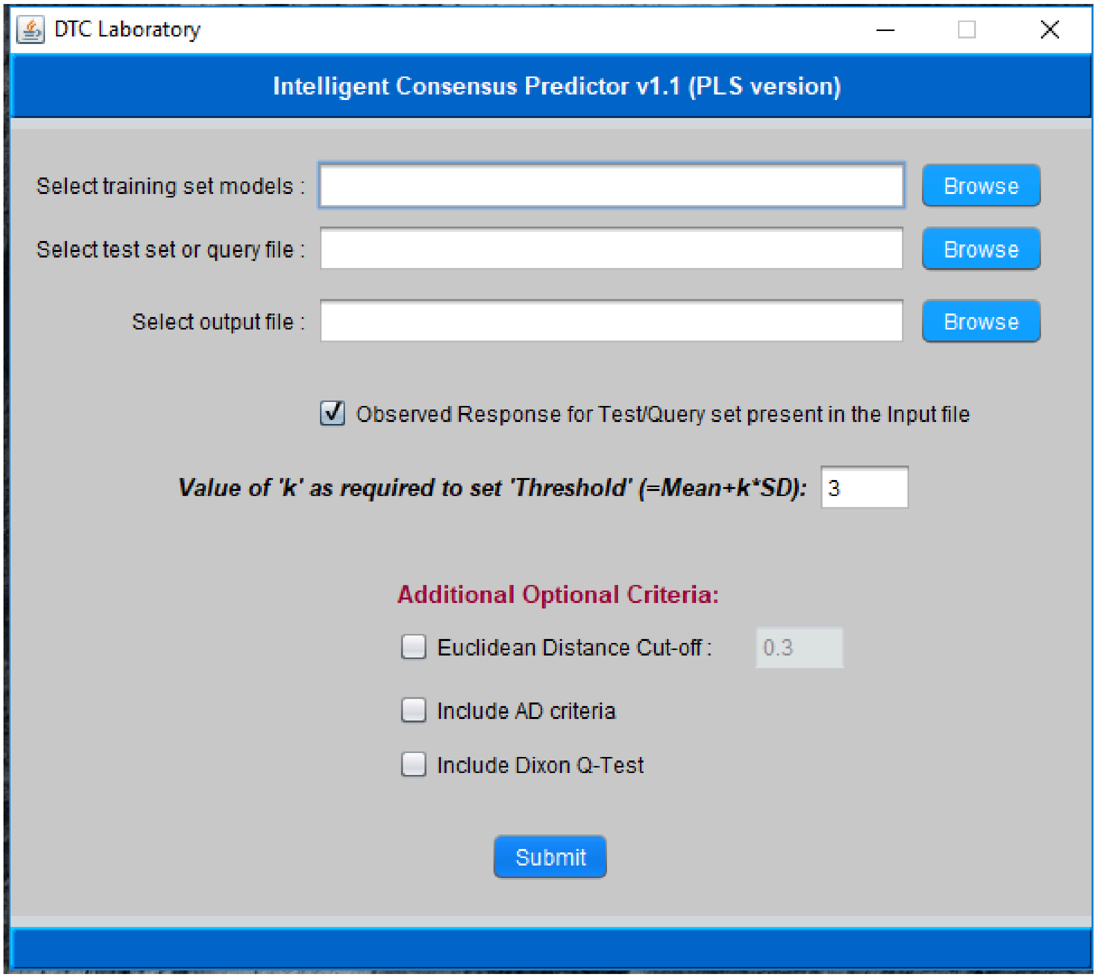
Intelligent Consensus Predictor (PLS version) *v1.1*

Intelligent Consensus Predictor v1.1 (for PLS models) tool judges the performance of "intelligent" consensus predictions obtained from multiple **QSAR** (*PLS*) models developed against a particular response and compares them with the prediction quality obtained from the individual models. This tool performs three different and unique ways of consensus predictions (to be detailed later on) along with the individual model predictions. Further, the quality of predictions is judged based on several external validation metrics such as Q^2_{F1}, Q^2_{F2}, Q^2_{F3}, CCC, r_m^2 and *MAE etc*. Moreover, this tool also provides few optional criteria (i.e., Euclidean distance cut-off, applicability domain[1] and Dixon-Q test[2]) that might help in improving the quality of prediction for a query molecule. The optimum settings can be fixed using the available QSAR models and corresponding external set compounds with known response values, while the same setting can be later employed for predictions of newly designed query molecules.



DTC Laboratory

Intelligent Consensus Predictor v1.1 (PLS version)

Select training set models :

Select test set or query file :

Select output file :

☑ Observed Response for Test/Query set present in the Input file

*Value of 'k' as required to set 'Threshold' (=Mean+k*SD):* 3

Additional Optional Criteria:

□ Euclidean Distance Cut-off : 0.3

□ Include AD criteria

□ Include Dixon Q-Test

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*) before using the program (*sample files are provided in the '***Data***' Folder, see the next section to understand the format of input files*).

How to prepare the input files, comprising Training models and corresponding Test sets

It is easy to prepare the required input files (*2 files*), i.e., Training and Test sets. There are 2 commonly used file formats (or extensions) that are recognized by the tool, namely, ***.xls***, and ***.xlsx***. One can easily store the relevant data in these file formats using **Microsoft Excel** software. ***Multiple model information should be stored in separate spreadsheets and same sequence/order of training models and corresponding test sets must be followed in the respective training and test input files.*** 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 four components, i.e., compound number (or serial number; first column), descriptor values in different columns (subsequent columns), quantitative response values (*activity/property*) (last column) and number of components (mention it with spreadsheet name in the format mentioned below). Now these four components should be arranged in the following way:

First Row: **Header ID,** *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 ID**(*only numerical values*)

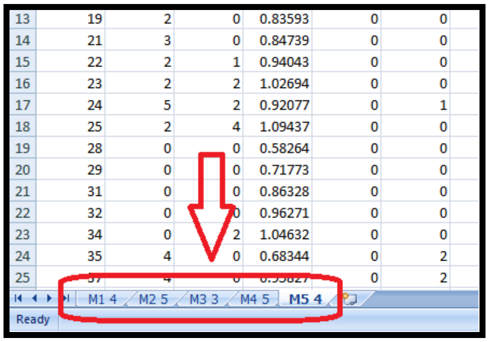
Subsequent columns: **Descriptors** (Independent variables)(*only numerical values*)

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

Number of components/latent variables: Please note that the user has to provide the number of components/latent variables information in the training set input file only (no need in test set input file) for each model. This information should be provided as spreadsheet name in the following manner:

UniqueModelName[SingleSpace]NumberOfComponentValue

For example: *Model2 5* (also see the following snap of input training file)



13	19	2	0	0.83593	0	0
14	21	3	0	0.84739	0	0
15	22	2	1	0.94043	0	0
16	23	2	2	1.02694	0	0
17	24	5	2	0.92077	0	1
18	25	2	4	1.09437	0	0
19	28	0	0	0.58264	0	0
20	29	0	0	0.71773	0	0
21	31	0	0	0.86328	0	0
22	32	0	0	0.96271	0	0
23	34	0	2	1.04632	0	0
24	35	4	0	0.68344	0	2
25	37	4	0	0.95827	0	2

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

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/

References:

1. Kunal Roy, SupratikKar, and Pravin Ambure. "On a simple approach for determining applicability domain of QSAR models." Chemometrics and Intelligent Laboratory Systems 145 (2015): 22-29.
2. https://en.wikipedia.org/wiki/Dixon%27s_Q_test

Disclaimer

This program has been developed in Java programming language and is *platform independent*. The software is validated on known data sets. Please report for discrepancy of result for any other dataset. Contact us at the following address:

Dr. Kunal Roy,

Drug Theoretics and Cheminformatics Lab.,

Dept. of Pharmaceutical Technology, Jadavpur University,

Kolkata, West Bengal,

INDIA-700032

E-mail ID: kunalroy_in@yahoo.com