PUBLICLY AVAILABLE QUANTITATIVE STRUCTURE-ACTIVITY RELATIONSHIP ANALYSIS SOFTWARE DEVELOPED IN THE LABORATORY FOR MOLECULAR MODELING, UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

Summary.

We have developed a Quantitative Structure-Activity Relationship (QSAR) analysis web server which is available available online (http://ceccr.ibiblio.org). It allows registered users throughout the world submit and execute QSAR analyses of their own datasets. Currently, the server allows calculation of MolconnZ and Atom Pair descriptors. Other descriptor sets as well as the possibility to upload the user's descriptors will be added soon. Currently, the k-nearest neighbors QSAR method is the only method available on the server. Other QSAR methods will become available in the future. It will become possible to carry out in an automatic way QSAR studies using combinations of different sets of descriptors and different QSAR methodologies thus practically implementing our combinatorial QSAR approach. Results of calculations can be stored on the server or downloaded. The users can also use the existing models for prediction of the external compounds. They can also carry out public database mining available on the server.

Stand-alone versions of programs used on the server are available. Stand-alone versions include newer versions of some software. Several utilities not available on the server, such as Pairwise correlation analysis, Leverage Outlier detection prior to QSAR studies, Similarity analysis of classes of compounds and several other useful utilities are also available. In the future, they will be also added to the QSAR online server.

This document contains a brief description of methods implemented in the QSAR server and the User's Manual.

METHODS.

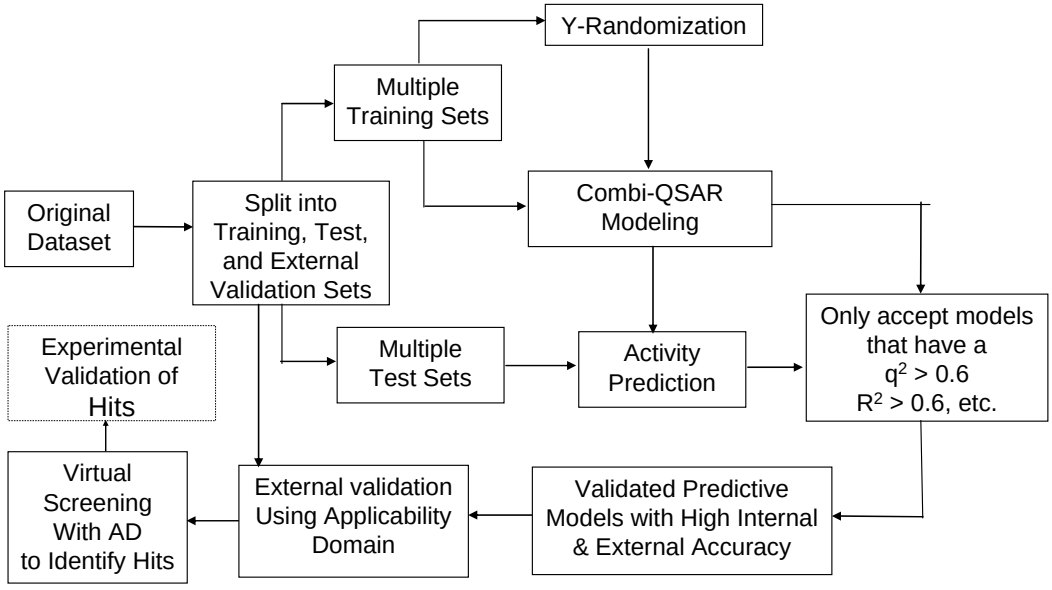

Figure 1. Flowchart of predictive QSAR modeling framework based on the validated combi- QSAR models.

Predictive QSAR modeling framework based on the validated combi- QSAR models is shown in Figure 1. Below is description of methods included in the current version of the server.

DESCRITORS

MolconnZ descriptors

MolconnZ descriptors included valence, path, cluster, path/cluster and chain molecular connectivity indices, kappa molecular shape indices, topological and electrotopological state indices, differential connectivity indices, graph's radius and diameter, Wiener and Platt indices, Shannon and Bonchev-Trinajstić information indices, counts of different vertices, counts of paths and edges between different types of vertices.

Atom Pair descriptors

AP descriptors are defined by types of atoms (or centers of double or triple chemical bonds) and shortest topological distances between them. The topological distance is the number of atoms along the shortest path connecting two atoms in a molecular graph. The general form of an atom pair descriptor is as follows:

Atom type i … (distance)… Atom type j.

In this study, the following 15 SYBYL atom types were used (1) negative charge center, NCC; (2) positive charge center, PCC; (3) hydrogen bond acceptor, HA; (4) hydrogen bond donor, HD; (5) aromatic ring center, ARC; (6) nitrogen atoms, N; (7) oxygen atoms, O; (8) sulfur atoms, S; (9) phosphorus atoms, P; (10) fluorine atoms, FL; (11) chlorine, bromine, iodine atoms, HAL; (12) carbon atoms, C; (13) all other elements, OE; (14) triple bond center, TBC; (15) double bond center, DBC. The total number of possible pairs is 120. 15 distance bins were defined in the interval from graph distance zero (i.e., zero atoms separating an atom pair) to 14 and greater. Thus, the total number of descriptors was 120x15=1800. Many of the AP descriptors have zero value for all molecules in a dataset (when certain atom types or atom pairs are absent in all molecular structures) and are excluded from QSAR studies.

PREPROCESSING OF DESCRIPTORS

Normalization of descriptors: Range-Scaling.

Descriptors of compounds of the Calibration set remained after exclusion of the external evaluation set were range-scaled. Let X_{ij} be the j-th descriptor value for compound i of the remaining part of the Calibration set (i=1,…,M; j=1,…,N), and $\min(X_j)$ and $\max(X_j)$ are the minimum and maximum values of this descriptor. The range-scaled values of X_{ij} are defined as follows:

$$X_{ij}^{rs} = \frac{X_{ij} - \min(X_j)}{\max(X_j) - \min(X_j)} \qquad (1)$$

Descriptors with zero variance, if any, were excluded.

Descriptor selection procedure: Pairwise Correlation Analysis.

This step is not included in the server version. Stand-alone software PairCorr3 for the Pairwise Correlation Analysis is provided. Pairwise Correlation Analysis of range-scaled descriptors is carried out according to the following algorithm. (i) Find descriptor with the highest variance. (ii) Discard all descriptors that have correlation coefficient higher than the predefined threshold with the descriptor found in the previous step. (iii) Among descriptors remaining, find descriptor with the next highest variance and go to step (ii). If there are no more descriptors, stop. The recommended value of the threshold correlation coefficient is 0.85.

DIVISION OF A DATASET INTO EXTERNAL EVALUATION AND MULTIPLE TRAINING AND TEST SETS

External evaluation set. The general workflow of the procedure implemented for the development of QSAR models is shown in Figure 1. Since our k-nearest-neighbor (kNN) QSAR procedure used in these calculations cannot predict activities beyond the highest and lowest activities of compounds included in the training set, two compounds, one with the highest and one with the lowest activity must always be included in the training sets. External evaluation set should be selected randomly from the entire dataset with these two compounds excluded. External evaluation set is used to simulate prediction of new compounds (i.e. chemical database and virtual library mining).

Training and test sets. The remaining part of a dataset (i.e. after selecting external evaluation set) is divided into multiple training and test sets using a sphere-exclusion algorithm (see the details below). Sphere-exclusion algorithm (Figure 2) ensures that the training set is distributed within the entire chemistry space, and that almost all test set compounds are within the model applicability domain. Compounds with highest and lowest activity values should be always included in the training sets. Training sets are used for building multiple k-nearest-neighbor (kNN) QSAR models. Test sets were used for validation of QSAR models.

Sphere-exclusion algorithm. Sphere-exclusion algorithm (Figure 2) starts with the calculation of the distance matrix **D** between representative points of compounds in the descriptor space. Let D_{min} and D_{max} be the minimum and maximum elements of **D**, respectively. N probe sphere radii are defined by the following formulas. $R_{min}=R_1=D_{min}$, $R_{max}=R_N=D_{max}/4$, $R_i=R_1+(i-1)*(R_N-R_1)/(N-1)$, where i=2,…,N-1. Alternatively, probe spheres are built with the radii $R_i=c_i(V/N)^{1/Y}$, where V is a volume of hyperparallelepiped in the descriptor space occupied by N representative points, and K is the number of descriptors, and $c_i$are defined by a user. Each probe sphere radius corresponds to one division into the training and test set. A sphere-exclusion algorithm consists of the following steps. (i) Select two (or more) compounds: one with the highest and one with the lowest activity values (and other compounds as defined by a user). Include them in the training set. Construct probe spheres around these compounds. Select compounds from these spheres and include them into test and training sets in the order defined by a user. Exclude all compounds from within these spheres from further consideration. (ii) If no more compounds left, stop. Otherwise, select the next point randomly (or the closest to one of the spheres built among closest to one of all spheres built, or the closest to one of the spheres built among farthest to one of all spheres built) and include it in the training set. Construct a probe sphere around this compound. Select compounds from this sphere and include them into test and training sets in the order defined by a user. Exclude all compounds from within this sphere from further consideration. Repeat step (ii). In these calculations, external evaluation set included 13 compounds.

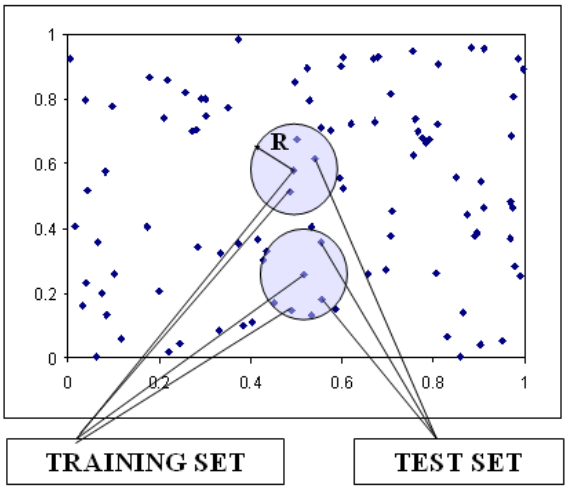

Figure 2. Division of a dataset into training and test sets using sphere-exclusion algorithm.

k-NEAREST NEIGHBORS QSAR PROCEDURE

k-nearest neighbors (kNN) QSAR. *k*NN QSAR is a stochastic variable selection procedure where the model optimization is driven by simulated annealing. The *k*NN procedure is aimed at the development of the model with the highest leave-one-out (LOO) cross-validated correlation coefficient R^2 (q^2) for the training set:

$$q^2 = 1 - \frac{\sum_{i=1}^{M}\left(y_i^{pred} - y_i^{obs}\right)^2}{\sum_{i=1}^{M}\left(y_i^{pred} - \bar{y}\right)^2} \qquad (2)$$

where y_i^{pred}, y_i^{obs} and $\bar{y}$ are predicted, observed and average activities of the i-th compound of the training set.

The procedure (Figure 3) starts with the random selection of a predefined number of descriptors from all descriptors. Activity of each compound y_i excluded in the LOO cross-validation procedure is predicted as the weighted average of activities of its nearest neighbors according to the following formula:

$$y_i = \frac{\sum_{j=1}^{K} y_j \exp(-d_{ij} / \sum_{l=1}^{K} d_{il})}{\sum_{j=1}^{K} \exp(-d_{ij} / \sum_{l=1}^{K} d_{il})}, \qquad (3)$$

where d_{ij} are distances between the *i*-th compound and its *k* nearest neighbors (j=1,…, K). The optimal number of nearest neighbors that yields the highest q^2 value is defined as part of the LOO cross-validation process as well. After each run of the LOO procedure, a predefined number of descriptors are randomly changed, and the new value of q^2 is defined. If q^2 (new) > q^2(old), the new set of descriptors is accepted. If q^2 (new) ≤ q^2(old), the new set of descriptors is accepted with probability p = exp(q^2(new) - q^2(old))/T, and rejected with probability (1-p), where T is a simulated annealing parameter, "temperature". During the process, T is gradually decreasing until the predefined value, and when this value is achieved the optimization process is terminated. In these calculations, the number of descriptors selected were 10, 12… 60. The number of descriptors selected changed at each step was 1 to 3. 10 models for each combination of division of a dataset into training and test sets and number of descriptors selected have been built. Maximum and minimum temperature values were 100 and 10^{-5}, and a coefficient for lowering temperature was 0.9.

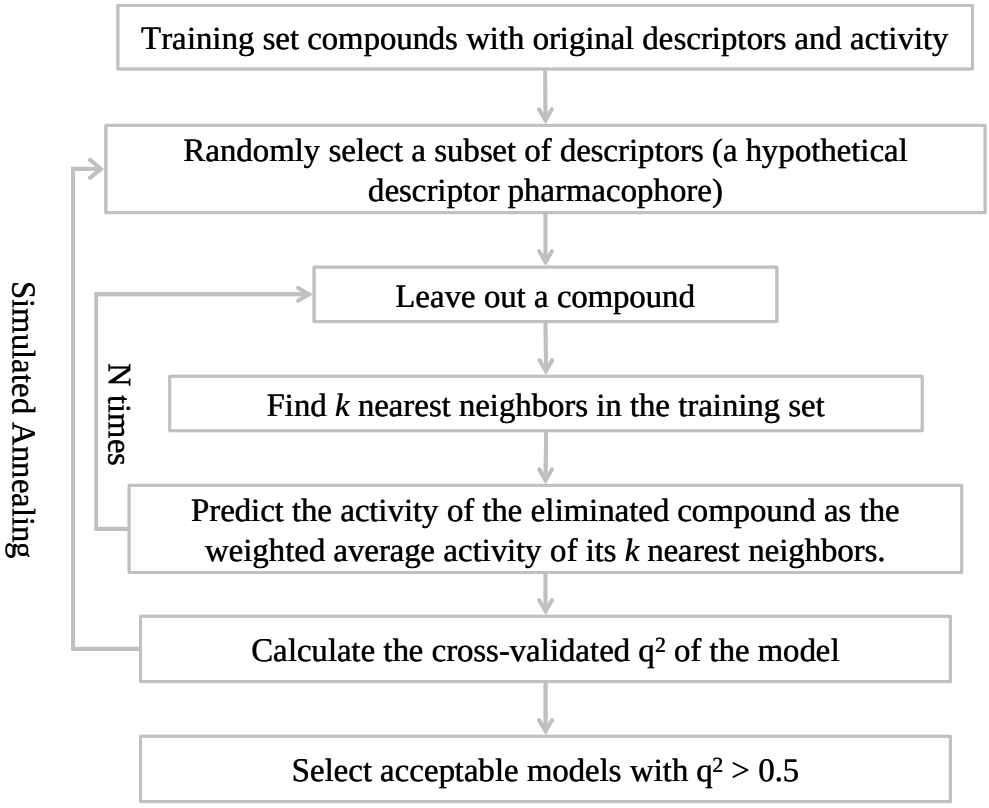

Figure 3. Flow chart of *k*NN-QSAR with Variable Selection.

Prediction of activities of test set compounds. Models satisfying criterion $q^2 > q^2$(threshold) (recommended value for q^2(threshold) is 0.5) are validated using test set compounds. Activities of the test set compounds are predicted only if these compounds are within the applicability domain of the respective training set models. We define this domain by a threshold distance between a test set compound and its k nearest neighbors in the training set in a multidimensional descriptor subspace of descriptors selected by the model. If the distance is beyond the threshold, the prediction is considered unreliable and is not made. This threshold distance is calculated as $D^2_{cutoff} = \langle D^2_{nn} \rangle + Z*VAR$, where $\langle D^2_{nn} \rangle$ is the squared mean distance between each of the training set compound and its k nearest neighbors, VAR is the variance of D_{nn}, and Z is a user-defined parameter (the recommended Z value is 0.5). For prediction, formula (3) is used. In this case, d_{ij} are distances between the *i*-th compound of the test set and its optimal number of K nearest neighbors (j=1,…,K) of the training set. We use the following statistics for prediction of test set compounds.[3]

$$R = \frac{\sum (y_i - \bar{y})(\tilde{y}_i - \bar{\tilde{y}})}{\sqrt{\sum (y_i - \bar{y})^2 \sum (\tilde{y}_i - \bar{\tilde{y}})^2}} \qquad (4)$$

$$R_0^2 = 1 - \frac{\sum (\tilde{y}_i - \tilde{y}_i^{r_0})^2}{\sum (\tilde{y}_i - \bar{\tilde{y}})^2} \quad (5a)$$

$$R'^2_0 = 1 - \frac{\sum (y_i - y_i^{r_0})^2}{\sum (y_i - \bar{y})^2} \quad (5b)$$

$$k = \frac{\sum y_i \tilde{y}_i}{\sum \tilde{y}_i^2} \quad (6a)$$

$$k' = \frac{\sum y_i \tilde{y}_i}{\sum y_i^2}, \quad (6b)$$

where y_i and $\tilde{y}_i$ are observed and predicted activities, R_0^2 and R'^2_0 are the coefficients of determination for regressions through the origin between predicted and observed, and observed and predicted activities, respectively, k and k′ are the corresponding slopes, and $y^{r_0} = k\tilde{y}$ and $\tilde{y}^{r_0} = k' y$ are the regressions through the origin for observed vs. predicted and predicted vs. observed activities. We recommend the following criteria as acceptable for the test sets: (i) correlation coefficient between predicted and observed activities $R^2>0.6$, (ii) coefficients of determination for regressions through the origin between predicted and observed activities R_0^2 and observed and predicted activities R'^2_0 and the corresponding slopes k and k' $\frac{R^2 - R_0^2}{R^2} < 0.1 \text{ and } 0.85 < k < 1.15 \text{ or } \frac{R^2 - R'^2_0}{R^2} < 0.1 \text{ and } 0.85 < k' < 1.15$, and (iii) $\left|R_0^2 - R'^2_0\right| < 0.3.$ If too many models satisfy these criteria, more strict criteria are used. Models satisfying these or more strict criteria criteria were used in the consensus prediction of the external evaluation set.

Y-randomization test. Y-randomization (randomization of response) is a widely used QSAR model validation approach.[2] It consists of rebuilding the models using randomized activities of the training set and subsequent assessment of the model statistics. It is expected that models obtained for the training sets with randomized activities should have significantly lower q^2 values than those obtained for the training sets with true activities, or at least these models do not have acceptable statistics for the test sets. If this condition is not satisfied, models built for the training sets with true activities are not reliable and should not be used for prediction of the external datasets. In Figure 1, calculations for Y-randomized test are shown by dashed lines.

Currently, Y-randomization test is not included in the server version. However, it is possible to use a stand-alone utility RandomizationSlow to randomize activities of the training sets and then resubmit the job using the online server.

Consensus prediction. All models satisfying acceptability criteria for the training and test sets are used for consensus prediction of compounds of the external evaluation set. Consensus prediction consists of averaging of predicted activity of each compound by multiple models. It has been shown that in most cases consensus prediction gives superior prediction statistics than most (and sometimes, all) of the individual models. Applicability domains were defined in consensus prediction for each individual model. Consensus prediction was considered acceptable, if it satisfied the above conditions (i)-(iii). Sometimes, when the range of activities of an external test set is small, a correlation coefficient between predicted and observed activities can be relatively low, despite the prediction is acceptable. It is because in this case, the error of prediction can become comparable to the whole range of activities. In this case, criteria other than correlation coefficient should be used, e.g. root mean square error of prediction (RMSE). For a model to be accepted, it should be comparable to RMSE for the test sets which had acceptable other prediction statistics, such as R^2, R_0^2, and k.Prior to the procedure, descriptors of the external evaluation set were normalized according to Formula (1). In this case, and $\min(X_j)$ and $\max(X_j)$ were defined by the part of the Calibration set without the external evaluation set.

Currently, RMSE analysis of the consensus prediction is not provided.

LEVERAGE OUTLIER DETECTION PRIOR TO QSAR STUDIES

Leverage outlier detection consists of the following steps:

- Calculate distance / similarity matrix in the entire descriptor space
- For each compound, find its nearest neighbor / most similar compound
- Find compounds which are out of the cutoff distance from their nearest neighbors or have similarity to the most similar compound lower than the predefined threshold
- Remove outliers, if necessary

Currently, stand-alone version of the outlier detection software is available. Online QSAR server does not provide this software.

SIMILARITY ANALYSIS OF CLASSES OF COMPOUNDS

For two classes of compounds, software finds which compounds of each class are dissimilar fgrom all compounds of another class. Similarity between compounds is defined by Z-Cutoff, Distance cutoff or Similarity cutoff value. Files of compounds of one class dissimilar from and similar to compounds of another class are created.

Similarity analysis software can be provided only as stand-alone software. Currently, this procedure is unavailable on the online QSAR server.

WORKING WITH THE WEB QSAR SERVER

1. Go to web site http://ceccr.ibiblio.org
2. If you are a new member, register. Your password will be sent to you by email.
3. If you are a registered user, sign in by entering your user name and password.

MODEL BUILDING

4. Click on Model Building.
5. Select Project Type. Currently, two project types can be selected: QSAR Continuous and QSAR Category.
6. Select a Dataset for analysis. A user can select a previously uploaded dataset (from a drop-down menu) or upload his or her dataset. User's dataset files should include a molecule file in SDF format and an activity file. In the future, mol2 and smiles molecule file formats will also be used.
7. Select a Tool (QSAR method) for analysis. Currently only one QSAR method is available: kNN (k nearest neighbors QSAR). In the future, PLS and SVM methods will be added. After choosing a Tool, a user can edit basic and advanced parameters. For kNN, Basic parameters include: Descriptor step size, Minimum Number of Descriptors selected, Maximum Number of Descriptors selected, and the Number of Runs for each set of parameters. Advanced parameters include: Maximum Number of Nearest Neighbors, Number of Pseudo neighbors, Different Simulated Annealing Parameters (Starting and ending temperatures, temperature coefficient, the maximum number of cycles performed for each temperature, and the number of permutations of descriptors selected), and the model applicability domain cutoff parameter. Model Acceptability criteria are different for Continuous and Category kNN. For continuous kNN, acceptability criteria include: minimum acceptable q^2 and R^2, intervals for slopes for regressions through the origin for predicted vs. observed and observed vs. predicted activities, and coefficients of determination for regressions through the origin. For category kNN, acceptability criteria include accuracy of classification for training and test sets.
8. Select Descriptors set. Currently, two sets of descriptors are available: MolconnZ and Atom Pair (ChemFeaturePair) descriptors. A user can choose if it is necessary to normalize descriptors. Currently, Range Scaling is the only normalization method used. In the future, Standard Normalization will be added.
9. A user can select, if it is necessary to randomly select an External Validation set from the entire dataset. If yes, the size of the External Validation set should be provided.
10. A user can select, if it is necessary to divide a dataset (or a subset of compounds after selecting an External Validation set) into training and test sets. Currently, only one splitting method is used: that based on the Sphere Exclusion algorithm. The user must enter the number of starting points

included in the training set, and the way the next probe sphere center is selected: randomly, closest representative point to the existing spheres, closest representative point to the existing spheres among the farthest from each of the existing spheres.

11. Finally, a user must enter the Job Name and submit the job.

MODEL ANALYSIS

12. After models are built, a user can click on the Model Analysis.
13. A user can select names of finished jobs and save, discard or download the models.
14. A user can perform external validation of the models and download the results.

USING MODELS FOR PREDICTION

15. A user can make predictions using Predictors (Models) built. Predictors are selected from the drop-down menu. To Apply the Prediction, a user must give a Job Name and select a database for prediction. A database can be selected from a list of databases available on the server, or uploaded by a user. Model Applicability Cutoff value must be provided. Currently, NCI database is available on the server.
16. Prediction results can be saved, discarded or downloaded.

WORKING WITH STAND-ALONE VERSION OF SPHERE EXCLUSION SOFTWARE

This software divides datasets into training and test sets.

Random2.exe
SE9v1.exe

To run this software, on command prompt enter

SE9

or

SE9 input.in

In the second case, you will have to prepare the input.in file. If input.in file does not exist, the program will transfer to the manual input, and the input.in file will be created by the software. See example of the input.in file below.

The program will ask you to enter descriptor ad activity file names.

Descriptor file has the following format:
Line 1: N1 N2 (Number of compounds, number of descriptors).
Line 2: IDs of descriptors (separated by spaces).
Lines 3 to N1+2: Compound's ID, Compound's Name, Compound's Descriptors (separated by spaces).
Line N1+3: Minimum values of descriptors (separated by spaces).
Line N1+4: Maximum values of descriptors (separated by spaces).
Descriptors in lines 3 to N1+2 must be normalized.

Activity file has the following format:

Lines 1 to N1: Compound's Name Compound's activity (with the space between them).

The program creates many output files:

- descriptors of training and test set compounds;
- activities of training and test set compounds;
- a list file containing names of training and test set files and the number of compounds included in training and test sets.
- a general (summary) output file containing names of training and test set files and the number of compounds included in training and test sets (more details than in the list file).

All these files have the same root you will be asked to enter.

EXAMPLE OF FILE NAMES

If, for example, you have entered "root", then

descriptors of training set compounds will be in root_a.* files,
descriptors of test set compounds will be in root_b.* files,
activities of training set compounds will be in root_a1.* files,
activities of test set compounds will be in root_b1.* files,
list file will have name root.list.
general output file will have name root.sum

* in file names above denotes numbers assigned by a program.

The program can use default parameters or parameters entered by the user.

You will be asked, if you want to use default parameters.

Default parameters are stored in **param9.txt** file. In this file, lines starting with # are comments. You can change default parameters by editing param9.txt file.

If you decided to use default parameters, but the param9.txt file was not found, the param9.txt file will be created automatically.
Parameters in this file will be used as default parameters.

PARAM9.TXT FILE CREATED BY THE SOFTWARE

```
# ---------------------------------------------------------------------
#
# DEFAULT PARAMETER FILE FOR SE9
#
# ---------------------------------------------------------------------
#
# THIS FILE WAS GENERATED  8/ 1/2005 21: 5:52 GMT BY SE9 SOFTWARE
#
# Copyright (C) 2002 A.Golbraikh & A.Tropsha
# School of Pharmacy CB #7360 Beard Hall
# University of North Carolina at Chapel Hill
# Chapel Hill, NC 27599-7360 USA
#
# ---------------------------------------------------------------------
#
# MINIMUM NUMBER OF COMPOUNDS IN THE TRAINING OR THE TEST SET
5
# MINIMUM PERCENT OF COMPOUNDS IN THE TRAINING OR THE TEST SET
6
# DIVISION IS BASED ON:
# 1 - DISSIMILARITY LEVELS
# 2 - DISTANCES BETWEEN POINTS
2
```

```
# MINIMUM DISSIMILARITY LEVEL (USE IF 1 - DISSIMILARITY LEVELS)
# 0.200000
# MAXIMUM DISSIMILARITY LEVEL (USE IF 1 - DISSIMILARITY LEVELS)
# 5.200000
# THE NUMBER OF STEPS (IF 1 - DISSIMILARITY LEVELS)
# THE NUMBER OF SPHERE RADII (IF 2 - DISTANCES BETWEEN POINTS)
50
# THE MAXIMUM NUMBER OF COMPOUNDS ASSIGNED TO THE TEST SET IN A ROW
# THE MAXIMUM NUMBER OF COMPOUNDS ASSIGNED TO THE TRAINING SET IN A ROW
1 1
# 1 - SELECTION OF THE NEXT TRAINING SET POINT IS BASED ON THE MINIMUM SPHERE CENTER DISTANCES
# 2 - SELECTION OF THE NEXT TRAINING SET POINT IS BASED ON THE MAXIMUM SPHERE CENTER DISTANCES
# 3 - RANDOM SELECTION OF THE NEXT TRAINING SET POINT
1
# THE NUMBER OF STARTING POINTS IN THE TRAINING SET
2
# 1 - MOST ACTIVE STARTING COMPOUNDS
# 2 - THE NUMBERS WILL BE ENTERED
# 3 - STARTING COMPOUNDS WILL BE SELECTED RANDOMLY
# 4 - MOST ACTIVE AND MOST INACTIVE STARTING COMPOUNDS (THE NUMBER OF STARTING POINTS SHOULD BE 2)
4
# USE IF 1 - MOST ACTIVE STARTING COMPOUND:
# 1 - MOST ACTIVE COMPOUND HAS THE HIGHEST ACTIVITY
# 2 - MOST ACTIVE COMPOUND HAS THE LOWEST ACTIVITY
# 1
# NUMBERS OF COMPOUNDS (IF 2 - THE NUMBERS WILL BE ENTERED)
# THE NUMBER OF ENTRIES BELOW MUST BE EQUAL TO THE NUMBER OF STARTING POINTS
#
 ---------------------------------------------------------------------
```

If you decided to enter parameters manually, you will be asked the corresponding questions.

The program implements two different approaches and three closely related algorithms for each approach. It will ask you to select one of the two options.

1. Division based on dissimilarity level.
2. Division based on distances between points.

Option 1. Probe sphere radii R=C(V/N)^1/K, where V is the volume in which the representative points are distributed in the descriptor space: V=Product(Dmax-Dmin), K is the number of descriptors, and C is the dissimilarity level.

Option 2. Probe sphere radii will be based on the value of R=(Rmax-Rmin), where Rmax and Rmin are the maximum and minimum distances between representative points.

RECOMMENDATION: If all normalized descriptors have minimum value 0 and maximum value 1, or the number of descriptors is lower than 10, any option can be used. Otherwise, use option 2.

If option 1 is selected, the program will ask you the minimum and maximum values of the dissimilarity level Cmin and Cmax, and the number of steps S between them. The total number of probe spheres will be equal to S+1 (from 0 to S); their radii will beRi=Cmin+i*(Cmax-Cmin)/S (i=0,...,S).

If option 2 is selected, the program will ask you to enter the number of probe sphere radii N. Radii will be calculated according to the following expression: R=Rmin+i*(Rmax-Rmin)/(4*N) (i=1,...,N).

Then you will be asked to input the following data:

- The minimum number of compounds in the training or the test set
- The minimum percent of compounds in the training or the test set

Then you will be asked to select the algorithm:

1. Division based on minimum sphere center distances.
2. Division based on maximum sphere center distances.
3. Division based on random sphere center selection.

In the first case, the next point of the training set is selected as the point closest to the center of one of the previous spheres.
In the second case, the next point of the training set is selected as the point closest to one of the sphere centers among farthest from the centers of all of the previous spheres.
In the third case, the next point of the training set is selected randomly.

Then you will be asked to enter the number of compounds which you want to include into the training set before the procedure starts.

If you entered zero, the first compound of the training set will be selected randomly.
If you entered a number higher than zero, you will be asked to select one of the following

three options:

1 - Select most active compounds as starting points
2 - Enter numbers of starting points
3 - Select initial compounds randomly

If the number of compounds which you want to include into the training set before the procedure starts is equal to 2, additional option will be added:

4 - One most active and one most inactive compound

If you select option 1, you will be asked to point out, if the most active compound has the highest or the lowest activity value.
If you select option 2, you will be asked to enter the numbers of compounds.

WORKING WITH THE STAND-ALONE VERSION OF CONTINUOUS kNN.

The package consists of the following files.

AllKnn2aLIN
AllKnn2LIN
rwknnLIN
knnpredictLIN
predact2rwknnLIN
regrprme5LIN
sorttabauto3LIN

The program is started as follows.

AllKnn2aLIN mode list_file output_table

or

AllKnn2LIN mode list_file output_table

The difference between these two programs: AllKnn2aLIN allows output of models with r<0.AllKnn2LIN does not allow output of models with r<0.

mode:

1 - to start building models
2 - to continue building models
3 - to predict test set by existing models with the target function higher than the specified threshold.

In modes 1 and 2, after building models, test set is also predicted

list_file:

Each line corresponds to one division into training and test sets and contains the following information:

Training set descriptors file name, training set activities file name, number of compounds in the training set, test set descriptors file name, test set activities file name, number of compounds in the test set.

The number of lines is unlimited.

output_table

Statistics for all accepted models will be found in the output_table file (sorted by model names) and sorted_output_table file (sorted by r^2 values for each split).

The name of the sorted file will be generated automatically as follows: if output_table name is output.tbl then sorted_output_table name will be outputsort.tbl. tbl extension will be added automatically to the output_table name, it is not necessary to provide it.

All files listed in the list_file must be in the working directory.

All descriptor and activity files must be in kNN format.

Output model list file *output_table.list (output_table is taken from the command line to run kNN*

Acceptable models are included in a list of model files together with the corresponding training set descriptor and activity files.

Each line of this file has the following Format:

Training set descriptor file name, training set activity file name, model name.

Descriptor file:

First line: number_of_compounds number_of_descriptors
Second line: descriptor_ids

Compound_lines: compound_id compound_name compound_descriptor1 compound_descriptor_2 ...
Last two lines (optional): minimum descriptor values and maximum descriptor values

Last two lines can be used for normalization of an external dataset or a database etc.

Last two lines will be missing,if descriptors are not normalized.

Activity file:

One line per compound: compound_name activity_value

Order of compounds in descriptor and activity files must be the same.

Default input parameters are in the following file: **knn.default.** This file must be in the working directory.

Here is an example of knn_category.default

Min_Number_Of_Descriptors: 20
Step: 5
Number_Of_Steps: 16
Number_Of_Cycles: 100
Number_Of_Neares_Neighbors: 5
Number_Of_Pseudo_Neighbors: 100
Number_Of_Mutations: 2
Runs_For_Each_Set_Of_Parameters: 10
T1: 100
T2: -5.0
Mu: 0.9
TcOverTb: -6.0
CutOff: 1.0
Minimum_q2: 0.5
Minimum_R2: 0.6
Slopes_between: 0.85 1.15
Rel_Diff_R2_R02: 0.1
Diff_R01_R02: 0.1
Stop: 50

In this example,

Models with 20,25,30,...,100 descriptors will be built (there are 16 steps, each step is 5 descriptors).

Number of Cycles is the maximum number of times the calculations are made with current temperature T.

Maximal number of nearest neighbors is 5. All values from 1 to 5 will be used.

Number of pseudo neighbors: 100% of compunds will be used as possible nearest neighbors.

Number of mutations: number of descriptors changed in the set of selected descriptors in each step of simulated annealing.

For each set of parameters 10 models will be built. So the total number of models for each split will be 17x10=170.

Simulated annealing parameters: T1=Tmax, T2=k in Tmin=10^k, Tnext=mu*Tprev, if Tb/Ta=10^(-6) and between Ta and Tb no improvement has been found than stop building model.

Cutoff: if Dnj>AVERAGE[D(nn)]+Cutoff*STDEV[D(nn)], do not predict. Dnj is a distance between a compound under prediction and its nearest neighbor, AVERAGE[D(nn)] and STDEV[D(nn)] are average and standard deviation for distances between nearest neighbors in the training set.

Minimum acceptable q2.

Minimum acceptable r2.

Acceptable interval for slopes for regressions through the origin (predicted vs. observed or observed vs. predicted activities).

Acceptable maximum for the relative difference between r2 and r02, where r02 is the coefficient of determination for regressions through the origin (predicted vs. observed or observed vs. predicted). (The previous two conditions must be satisfied for one of the regressions through the origin.)

Acceptable maximum difference between coefficients of determination for regressions through the origin (predicted vs. observed or observed vs. predicted).

Stop: 50. If there are 50 consecutive splits in the list_file which didn't give any good model, stop.

WORKING WITH STAND-ALONE VERSIONS OF CATEGORY kNN

The package consists of the following files.

AllKnn_categoryPREV
rwknn_categoryPREV
predact2_categoryPREV
knnpredict_categoryPREV

Spearman_categoryPREV
sorttabauto_categoryPREV

The program is started as follows.

AllKnn_category mode_list_file output_table target_function

mode:
1 - to start building models
2 - to continue building models
3 - to predict test set by existing models with the target function higher than the specified threshold

In modes 1 and 2, after building models, test set is also predicted

list_file:

Each line corresponds to one division into training and test sets and contains the following information:

Training set descriptors file name, training set activities file name, number of compounds in the training set, test set descriptors file name, test set activities file name, number of compounds in the test set.

The number of lines is unlimited.

output_table

Statistics for all accepted models will be found in the output_table (sorted by model names) file and sorted_output_table file (sorted by values of the corresponding target function for the test set). The name of the sorted file will be generated automatically as follows: if output_table name is output.tbl then sorted_output_table name will be outputsort.tbl. tbl extension will be added automatically to the output_table name, it is not necessary to provide it.

target_function
Currently, there are four target functions implemented.

1. TF=Ncorr/Ntot
2. TF=1/K*SUM[Ncorr(i)/Ni], i=1,...,K, where K is the number of categories
3. TF=1-SUM[|i-j|Nij]/MaxErr, Nij are the number of compounds in category i which are given category j; MaxErr is the maximal possible error. MaxErr=SUM1[(n-i)Ni]+SUM2[(i-1)Ni], in SUM1 i=1,...,[n/2]; in SUM2 i=[n/2]+1,...,K.

4. TF=1-SUM[|i-j|/MaxNormErr], MaxNormErr is the maximal normalized error. MaxNormErr={n(3n-2)-0.5[1+(-1)^(n+1)]}/4.

All files listed in the list_file must be in the working directory.
All descriptor and activity files must be in kNN format.

Output model list file *output_table.list (output_table is taken from the command line to run kNN*

Acceptable models are included in a list of model files together with the corresponding training set descriptor and activity files.

Each line of this file has the following Format:

Training set descriptor file name, training set activity file name, model name.

Descriptor file:
First line: number_of_compounds number_of_descriptors
Second line: descriptor_ids
Compound_lines: compound_id compound_name compound_descriptor1 compound_descriptor_2 ...
Last two lines (optional): minimum descriptor values maximum descriptor values
Last two lines can be used for normalization of an external dataset or a database etc.
Last two lines will be missing,if descriptors are not normalized.

Activity file:
One line per compound: compound_name activity_value (category)
Categories must be non-negative consecutive whole numbers
Order of compounds in descriptor and activity files must be the same.

output_table

The following information will be included for each accepted model:
number of compounds in the training set
the optimal number of nearest neighbors
values for four target functions for the training set
count of compounds and accuracy for each category for the training set
number of compounds in the test set
values for four target functions for the test set
count of compounds and accuracy for each category for the test set
file name for predicted and observed categories

Models in output_file are arranged by the order they were generated

sorted_output_table file will have the same format as the output_table file, but models in it will be sorted by the target function which was optimized.

Default input parameters are in the following file. **knn_category.default.** This file must be in the working directory.

Here is an example of knn_category.default

Min_Number_Of_Descriptors: 20
Step: 5
Number_Of_Steps: 16
Number_Of_Cycles: 100
Number_Of_Neares_Neighbors: 5
Number_Of_Pseudo_Neighbors: 100
Number_Of_Mutations: 2
Runs_For_Each_Set_Of_Parameters: 10
T1: 100
T2: -5.0
Mu: 0.9
TcOverTb: -6.0
Minimum_acc_train: 0.6
Minimum_acc_test: 0.6
CutOff: 1.0
Stop: 50

In this example,

Models with 20,25,30,...,100 descriptors will be built

(there are 16 steps, each step is 5 descriptors).
Number of Cycles is the maximum number of times the calculations are made with current temperature T.
Maximal number of nearest neighbors is 5. All values from 1 to 5 will be used.
Number of pseudo neighbors: 100% of compunds will be used as possible nearest neighbors.
Number of mutations: number of descriptors changed in the set of selected descriptors in each step of simulated annealing.

For each set of parameters 10models will be built. So the total number of models for each split will be 17x10=170.
Simulated annealing parameters: T1=Tmax, T2=k in Tmin=10^k, Tnext=mu*Tprev, if Tb/Ta=10^(-6) and between Ta and Tb no improvement has been found than stop building model.
Minimum_acc_train: minimum acceptable target function value for the training set.
Minimum_acc_test: minimum acceptable target function value for the test set.
Cutoff: if Dnj>AVERAGE[D(nn)]+Cutoff*STDEV[D(nn)], do not predict. Dnj is a distance between a compound under prediction and its nearest neighbor, AVERAGE[D(nn)] and STDEV[D(nn)] are average and standard deviation for distances between nearest neighbors in the training set.
Stop: 50. If there are 50 consecutive splits in the list_file which didn't give any good model, stop.

A version of kNN classification is available separately from ceccr. It takes into account imbalance between datasets.

Files:

AllKnn_categoryEXP5
rwknn_category
predact2_category
knnpredict_category
Spearman_category
sorttabauto_category

Here is example of the **knn_category.default** for this version

Min_Number_Of_Descriptors: 6
Step: 2
Number_Of_Steps: 22
Number_Of_Cycles: 100
Number_Of_Neares_Neighbors_min: 1
Number_Of_Neares_Neighbors_max: 9
Number_Of_Pseudo_Neighbors: 100
Number_Of_Mutations: 2
Runs_For_Each_Set_Of_Parameters: 10
T1: 100
T2: -5.0
Mu: 0.9
TcOverTb: -6.0
Number_of_categories: 2

Weight_1: 0.3
Weight_2: 0.7
Thresh_between_1_and_2: 0.3
Minimum_acc_train: 0.7
Minimum_acc_test: 0.7
Minimum_accuracy_1: 0.7
Minimum_accuracy_2: 0.7
Penalty: 0.5
CutOff: 0.50
Stop: 50

Compared to the ceccr version:

1. The maximum number of nearest neighbors is increased to 11, and the minimum number of nearest neighbors can be larger than 1.

This is example input for this part.
Number_Of_Neares_Neighbors_min: 1
Number_Of_Neares_Neighbors_max: 9

2. Weighting of categories is introduced.
We recommend using lower weights for larger classes or categories.

This is example input for this part.
Number_of_categories: 2
Weight_1: 0.3
Weight_2: 0.7

The number of weights must be equal to the number of categories.

3. Threshold between categories is introduced. If threshold =0.3,
a compound is given category 1, if the predicted category is lower than 1.3; otherwise it is given category 2.

This is example input for this part.
Thresh_between_1_and_2: 0.3

The number of thresholds must be equal to the number of categories minus one.

4. Acceptable models must provide specified accuracies of prediction for each category.

This is example input for this part.
Minimum_accuracy_1: 0.7
Minimum_accuracy_2: 0.7

5. Penalty for non-equal accuracies for different categories is introduced.

This is example input for this part.
Penalty: 0.5

DESCRIPTOR NORMALIZER: DescriptorNormalizer

This software performs range-scaling of descriptor file according to formula (1).

Input: non-normalized dataset descriptor file
Output: normalized dataset descriptor file

The last two lines of the normalized dataset descriptor file contain minimum and maximum values of each descriptor.

DATABASE NORMALIZER: Database_Normalizer

This software performs range-scaling of the database descriptor file according to formula (1). X_{min} and X_{max} are taken from the last two lines of the normalized dataset descriptor file, but descriptor values are taken from the database file.

Input: normalized descriptor file, non-normalized database descriptor file
Output: normalized database descriptor file

OUTLIER DETECTION AND REMOVAL

Some of this software can be used for other purposes as well.

DistanceMatrices2

Calculates distance/similarity matrices.

Distance/similarity measures included:

1. Euclidean distance.
2. Hamming Distance (Manhattan Distance, City Block Distance).
3. Minkowski Distance.
4. Tanimoto Coefficient (Jaccard Coefficient).
5. Dice Coefficient (Czekanowski Coefficient, Soerenson Coefficient).
6. Soergel Distance.
7. Cosine Coefficient (Ochiai Coefficient).
8. Tversky Similarity.
9. Chebyshev Distance.
10. Canberra Distance.

Input: Descriptor file.
Output: Distance/Similarity matrix.

NearestNeighbors2

Input: Distance/similarity matrix.

Output: For each compound, sorted distances similarities to other compounds. Distances are sorted in the ascending order. Similarities are sorted in descending order.

OutliersZC2

Input:

1. Distance/similarity matrix.
2. Output files from NearestNeighbors2.
3. Minimum Z-cutoff, step for Z-cutoff, and number of steps.

Output:

Lists of nearest neighbors for each compound at each Z-cutoff.
If there are no nearest neighbors for some Z-cutoff, a compound is outlier for this Z-cutoff.

OutliersDM2

1. Distance/similarity matrix.
2. Number of probe spheres.

Output:

Lists of nearest neighbors for each compound are compounds within probe spheres.
If there are no compounds within the probe sphere of certain radius, a compound is outlier for this radius.
is outlier for this Z-cutoff.

QuickSort2

QuickSort algorithm. Used with the corresponding NearestNeighbors software.

removeoutliers.x.awk and **removeoutliers.a.awk**

Remove otliers from descriptor and activity files.

Example of how to run these scripts.
Suppose you have descriptor and activity files data.x and data.a, and you created outliers file data.out.

awk -v ZCutoff=0.5 –v fileoutliers=data.out –f removeoutliers.x.awk data.x > data.no.outliers.x

will create data.no.outlies.x descriptor file with outliers removed. There will be a small temp file created as well, which must be placed as the first line of the new descriptor file.

awk -v ZCutoff=0.5 –v fileoutliers=data.out –f removeoutliers.x.awk data.x > data.no.outliers.a

will create data.no.outlies.a activity file.
PAIRWISE CORRELATION ANALYSIS: PairCorr3

Input:
Input descriptor file.
Threshold correlation coefficient

Output:
Descriptor file with descriptors the correlation coefficients between which are lower than the threshold correlated coefficient.

SIMILARITY ANALYSIS: SimilarityZC and SimilarityDM.

Input:

Descriptor files for classes 1 and 2.

Distance/similarity measures included:
1. Euclidean distance.
2. Hamming Distance (Manhattan Distance, City Block Distance).
3. Minkowski Distance.
4. Tanimoto Coefficient (Jaccard Coefficient).
5. Dice Coefficient (Czekanowski Coefficient, Soerenson Coefficient).
6. Soergel Distance.
7. Cosine Coefficient (Ochiai Coefficient).
8. Tversky Similarity.
9. Chebyshev Distance.
10. Canberra Distance.

Only for SimilarityZC:

Z-cutoff minimum
Z-cutoff step
Z-cutoff: number of steps

Only for SimilarityDM:

Distance-cutoff minimum
Distance-cutoff step
Distance-cutoff: number of steps

Output:
Descriptor files for compounds of class1 similar to those of class 2 (for different Z-Cutoff or Distance-Cutoff).
Descriptor files for compounds of class1 dissimilar from those of class 2 (for different Z-Cutoff or Distance-Cutoff).

RANDOMIZATION: RandomizationSlow

The program is used for scrambling activities of a dataset. It should be used for Y-randomization test. Using this program is straightforward.

Input: activity file
Output: scrambled activity file

SELECTION OF EXTERNAL EVALUATION SET: RandomDivSlow3

Input:
Descriptor file
Activity file
Number of compounds in the external evaluation set

Output:
Descriptor and activity files for the external evaluation set
Descriptor and activity files for the remaining subset
List of files created

The list is appended by one line every time the program is used.

Each line corresponds to one selection of the external evaluation set. It has the following format:

Remaining subset descriptors file name, remaining subset activities file name, number of compounds in the remaining subset, external evaluation set descriptors file name, external evaluation set activities file name, number of compounds in the external evaluation set.

CONSENSUS PREDICTION FOR CONTINUOUS RESPONSE VARIABLE

PredActivContinuousrwknnLIN

Input:
Model list file created by AllKnn2LIN or AllKnn2aLIN, or a similar file created by a user.
Descriptor file for prediction
Descriptor and activity files for training sets included in the model list
Z-Cutoff

Output:
List of compounds from yje descriptor file for prediction
List of files with predicted activities for each model
Files with predicted activities for each model

ConsPredContinuous

Input:
List of compounds from yje descriptor file for prediction

List of files with predicted activities for each model
Files with predicted activities for each model

Output:
Consensus prediction file.

Consensus prediction file includes predictions for each compound by each model, the number of models a compound was predicted (was within the applicability domain) and average and standard deviation of prediction for each compound.

CONSENSUS PREDICTION FOR CATEGORY RESPONSE VARIABLE

PredActivCategory

Input:
Model list file created by AllKnn2LIN or AllKnn2aLIN, or a similar file created by a user.
Descriptor file for prediction
Descriptor and activity files for training sets included in the model list
Z-Cutoff
Number of categories
Threshold for every two consecutive categories (see a new version for Category kNN above; for old version use 0.5)

Output:
List of compounds from yje descriptor file for prediction
List of files with predicted activities for each model
Files with predicted activities for each model

ConsPredCategory

Input:
List of compounds from yje descriptor file for prediction
List of files with predicted activities for each model
Files with predicted activities for each model
Minimum and maximum category number
Threshold for every two consecutive categories (recommended value: 0.5)

Output:
Consensus prediction file.

Consensus prediction file includes predictions for each compound by each model, number of models a compound was predicted (was within the applicability domain), and average and rounded value for category of each compound.