A Measure of Domain of Applicability for QSAR Modelling Based on Intelligent *K*-Means Clustering

Robert W. Stanforth[ab], Evgueni Kolossov[a]* and Boris Mirkin[b]
[a] ID Business Solutions Ltd., 2 Occam Court, Surrey Research Park, Guildford GU2 7QB, UK
[b] School of Computer Science, Birkbeck College, London WC1E 7HX, UK
* To receive all correspondence
E-mail: EKolossov@id-bs.com

Keywords: QSAR, domain of applicability, *K*-means, fuzzy clustering

Received: July 05, 2006; Accepted: February 13, 2007

Abstract

The prediction of a biological activity using a Quantitative Structure-Activity Relationship (QSAR) model is valid only if the compound in question is inside the model's domain of applicability. The existing methods for determining the domain of applicability in descriptor space suffer from problems including poor handling of non-convex training sets and computational inefficiency.

In this paper we propose a cluster-based approach to modelling the domain of applicability, which may overcome some of the shortcomings of the existing approaches described. We investigate applying an intelligent version of the *K*-means clustering algorithm to this problem, modelling the training set as a collection of clusters in descriptor space. A test compound is assigned 'fuzzy membership' of each individual cluster, from which an overall distance may be calculated. Finally, we experimentally assess how this cluster-based approach compares with the existing methods.

1 Introduction

As with any regression, a QSAR model cannot be expected to extrapolate well: it cannot be expected to give a reliable prediction for a compound dissimilar to those in the model's training set. The term 'domain of applicability' of a model denotes the region of chemical space that is adequately represented by similar compounds in the training set, such that predictions within the domain will not suffer from this extrapolation problem.

In order to be of practical use, it must be possible to determine whether or not a given compound is inside or outside the domain of applicability [1]. In the latter case, it is also desirable to be able to calculate how far outside the domain the compound is: the reliability of prediction will be only slightly impaired just outside the domain, while the prediction error steadily worsens as the distance from the domain increases [2].

The domain of applicability problem has become of interest rather recently; therefore, several different approaches have been published. One should distinguish between them based on the domain representation used. The domain can be represented by its:

(a) Bounding box;

(b) Convex hull;

(c) Ellipsoid;

(d) Nearest neighbours.

Let us describe these approaches briefly and illustrate them with Figure 1.

The bounding box method [2, 3] is the simplest and fastest of all approaches considered so far. The domain of applicability is taken to be the smallest axis-aligned rectangular box in descriptor space that contains the whole training set. In other words, a compound is deemed to be inside the domain if and only if, for each descriptor, the descriptor value for that compound is in the range of values taken by that descriptor over the whole training set. Alternatively, in order to overcome sensitivity to outliers, each descriptor range may be based

on a percentile range of that descriptor over the training set (instead of the entire range of the descriptor over the training set).

The bounding box method ensures that there is no extrapolation with respect to any individual descriptor. However, modelling the training set as a rectangular box is too crude to avoid extrapolation in multivariate descriptor space. In the absence of careful experimental design to ensure statistical independence of the descriptors, using the bounding box method to estimate domain of applicability will typically result in substantial regions of false positives in which compounds are erroneously deemed to be inside the domain even if they differ from it in essential features.

If principal components analysis (PCA) is applied as a pre-processing step, then the results of the bounding box method can be improved [3]. The resulting principal components will be uncorrelated with one another over the training set and, moreover, if the training data is drawn from a multivariate normal distribution in descriptor space then the principal components will satisfy the stronger condition of statistical independence assumed by the bounding-box method. However, since the principal components themselves (as directions in descriptor space) may suffer from a lack of any meaningful interpretation, their statistical independence becomes questionable as a sufficient assumption for applying the bounding-box method. PCA implicitly aggregates all descriptors into a single descriptor space, and therefore the domain of applicability should logically be expressed isotropically as a region of descriptor space rather than as the conjunction of artificial ranges. (The most dramatic manifestation of this problem comes with a training set whose distribution in descriptor space is perfectly spherical. The choice of principal components is then arbitrary, and so any choice of mean-centred cube could equally well arise as the domain of applicability, resulting in substantial ambiguity.)

The convex hull method [3, 4] improves on the bounding box approach by taking the domain of applicability to be the smallest convex region of descriptor space containing the whole training set. This ensures that the domain is restricted to consist of precisely those points at which the model can be applied without extrapolation.

There are still problems with the convex hull method, however. If the training set covers a non-convex region of descriptor space then false positives may still occur in regions of concavity: in such interior regions unrepresented by the training set, interpolation can suffer the same problems as extrapolation, so they are not necessarily part of the domain. Furthermore, the computational complexity of the convex hull method is prohibitive both in time and in storage.

Another method is based on the concept of 'leverage' or 'Mahalanobis distance' of a test compound: $h(\mathbf{x}) = N^{-1} + \mathbf{x}^T(X^TX)^{-1}\mathbf{x}$ where the vector $\mathbf{x}$ represents the test compound in centred descriptor space and X is the training data matrix whose N rows represent the training compounds in the centred descriptor space [2, 3, 5, 6]. (For this method 'centred' means that the grand mean of the training data is taken as the origin of descriptor space.) The leverage of a compound is a measure related to the statistical error of prediction at that compound, and can be viewed as a measure of extrapolation.

Although the leverage calculations have a sound mathematical footing, they rely on the assumption that there is an underlying linear model applicable both inside and outside the domain, with unreliability of prediction arising purely from statistical error in estimating the model parameters rather than from limitations in the applicability of the underlying model. Not unrelated is the observation that leverage would provide only a crude estimate of the shape of the domain of applicability: contours of $h(\mathbf{x})$ are always ellipsoids in descriptor space.

An altogether different approach is taken by the 'nearest neighbour' methods, most notably '*k* nearest neighbours' (*k*-NN) [2, 6]. In this approach a distance measure is constructed by taking the average distances from the test compound to the *k* nearest compounds in the training set. More sophisticated variants include probability density estimation [3], in which a 'membership-of-domain' value is estimated via Parzen's Window as the average over all training structures of a narrow Gaussian distribution centred at each training point in descriptor space [7].

These methods both give appealing results [3, 8], but in much the same way as the convex hull method their dependence on every single training compound gives rise to substantial time and storage requirements. Indeed, the whole training set must be stored with the model, and must be processed in its entirety every time a distance-to-domain calculation is performed. This is in contrast to both the bounding box and the leverage-based method. In the former, the set of ranges over all dimensions provides an easy-to-store and easy-to-use representation of the estimated domain. In the latter, once the covariance matrix $(X^TX)^{-1}$ has been calculated up front it can be reused for all subsequent distance-to-domain calculations without knowledge of the individual training compounds.

2 Methods

It can be argued that the sensitive dependence of the nearest neighbour and convex hull methods on the individual test compounds is not just inefficient but is overkill. Like the model itself, its domain of applicability should depend on the broad trends of training set rather than on its individual fluctuations. We therefore aim to work with a model of the domain, not as crude as the rectangular box or ellipsoidal models yielded by the bounding box and leverage-based methods, but nevertheless less unwieldy than the entire training set. Such a model is provided by clustering the training set.

2.1 Modelling the Domain of Applicability Using Intelligent *K*-Means Clustering

Our strategy for modelling the domain of applicability is to use a version of the well-known *K*-means algorithm [9] to produce a cluster-based representation of the training set.

K-means clustering is founded on the approximation of each point in a dataset by the centroid of that point's cluster. This approximation can be quantified by decomposing data scatter into contributions that are 'explained' and 'unexplained' by the cluster model as follows [9]:

$$\sum_{i \in T} \mathbf{x}_i \cdot \mathbf{x}_i = \sum_{k=1}^{K} N_k \mathbf{c}_k \cdot \mathbf{c}_k + \sum_{k=1}^{K} \sum_{i \in C_k} (\mathbf{x}_i - \mathbf{c}_k) \cdot (\mathbf{x}_i - \mathbf{c}_k) \tag{2.1}$$

where $\{ \mathbf{x}_i : i \in T \}$ is the training set and the K clusters C_k have sizes N_k and centroids $\mathbf{c}_k$ respectively. This suggests optimisation of the so-called square-error *K*-means criterion:

$$\sum_{k=1}^{K} \sum_{i \in C_k} (\mathbf{x}_i - \mathbf{c}_k) \cdot (\mathbf{x}_i - \mathbf{c}_k) \tag{2.2}$$

Equation (2.1) holds for any inner product in descriptor space, and we choose the standard scalar product having normalised each descriptor to have 1st and 99th percentile values equal to −1 and +1 respectively.

K-means clustering requires an initialisation step to specify the number of clusters K and the initial positions of their centroids. The intelligent *K*-means algorithm described in [9] achieves this initialisation by using the so-called Anomalous Pattern Clustering (APC), a version of 2-means in which:

(1) the initial two centroids are specified as follows: the first one is the data gravity centre, a point whose components are grand means of the corresponding components in the entire data set, and the second is an entity which is the farthest away from the gravity centre (most deviant entity);

(2) the first, grand mean, centroid never changes so that the only changeable centroid is the second one.

Therefore APC starts with a cluster of points that are closer to the second, most deviant, centroid than to the first one, the grand mean. Then the second centroid is updated for the cluster's gravity centre, after which the second cluster itself is updated with the changed centroid. This procedure is reiterated until convergence. Then the second cluster thus found is removed, and the next 'most deviant' cluster is extracted from the remaining set – without ever changing the first centroid. The process of cluster extraction goes on until no non-clustered entities remain or any other stopping criterion is reached.

The clusters resulting from APC, with smallest clusters removed, are taken as the input to *K*-means algorithm. This algorithm proceeds by optimising centroids $\mathbf{c}_k$ and cluster memberships C_k alternately until the process is stabilized; that is, a (local) minimum of the *K*-means criterion (2.2) is reached. This combined procedure constitutes the intelligent *K*-means algorithm [9].

Anomalous Pattern Clustering works fast and, while not guaranteeing that the local solution reached with *K*-means is the global minimum of criterion (2.1), provides a well-shaped and reproducible starting configuration, avoiding the random elements of other popular *K*-means initialization methods [10].

2.2 Fuzzy Membership

Having modelled the domain of applicability as a collection of clusters whose centroids are known, we could adopt a 'distance to nearest cluster' approach to measuring distance-to-domain. However, such a measure, even if enhanced by averaging the distance to the *k* nearest clusters in the manner of the *k* nearest-neighbours (*k*-NN) technique, inevitably suffers from abruptness caused by the imminent changes of the nearest neighbour sets in nearby points.

Instead, we utilise a smoother distance-to-domain function by allowing all clusters to contribute. Rather than assigning a test compound **x** exclusively to its nearest cluster (or sharing it equally amongst its *k* nearest clusters), we grant it 'joint membership' of many clusters to varying degrees. To this end, we employ the well-known concept of 'fuzzy membership' [11]. A point **x** is assigned a proportion $z_k(\mathbf{x})$ of membership of each cluster C_k, between 0 (no membership of the cluster) and 1 (exclusive membership of the cluster). The summary membership (over all clusters) for any point in this arrangement must be 1. Obtaining these apportionments of membership can be formulated as a minimisation problem as follows:

$$\begin{aligned} \text{minimise} \quad & F(\mathbf{x}) = \sum_{k=1}^{K} z_k(\mathbf{x})^{\alpha} d_k(\mathbf{x}) \\ \text{subject to} \quad & \sum_{k=1}^{K} z_k(\mathbf{x}) = 1 \end{aligned} \tag{2.3}$$

where $d_k(\mathbf{x})$ is some measure of distance from **x** to cluster C_k. We shall take $d_k(\mathbf{x})$ to be the distance to the cluster's centroid $\mathbf{c}_k$, utilising the same Euclidean squared distance in normalised descriptor space was as used in the *K*-means algorithm.

The parameter $\alpha \geq 1$ in the above controls the degree of fuzziness: at $\alpha=1$ the membership coincides with exclusive 'nearest cluster' membership, while as α increases to infinity the membership weights $z_k(\mathbf{x})$ converge to the common value K^{-1}. For $\alpha>1$ the solution can be written explicitly [11]:

$$z_k(\mathbf{x}) = \frac{d_k(\mathbf{x})^{\frac{-1}{\alpha-1}}}{\sum_{l=1}^{K} d_l(\mathbf{x})^{\frac{-1}{\alpha-1}}} \tag{2.4}$$

(If $d_k(\mathbf{x}) = 0$ for some cluster C_k then **x** is assigned exclusive membership of that cluster.)

There is a fuzzy variant of the *K*-means algorithm in which the summary value of $F(\mathbf{x})$ in (2.3) over the whole training set is used as the criterion to be minimized in the process of clustering, thereby determining both the fuzzy membership values and the centroids [11]. No specific interpretation is attached to this summary value in fuzzy clustering. However, in our

case (of 'crisp clustering' followed by 'fuzzy membership' assignment), the minimised value of $F(\mathbf{x})$ in problem (2.3) expresses the maximally possible extent of belongingness of point $\mathbf{x}$ to the pre-clustered domain. The associated optimal membership values express the extent of participation of the different cluster centroids in the measure.

2.3 Weighted Average

The distance-to-domain measure can now be explicitly expressed as a weighted average of the distance-to-cluster measures $d_k(\mathbf{x})$ based on the fuzzy memberships $z_k(\mathbf{x})$.

The distance-to-domain measure is taken to be the attained minimal value of the fuzzy membership objective function $F(\mathbf{x})$ itself in (2.3), giving rise to the following distance measure for $\alpha>1$:

$$D_\alpha(\mathbf{x}) = \sum_{k=1}^{K} z_k(\mathbf{x})^\alpha \, d_k(\mathbf{x}) = \frac{\sum_{k=1}^{K} d_k(\mathbf{x})^{\frac{-1}{\alpha-1}}}{\left(\sum_{k=1}^{K} d_k(\mathbf{x})^{\frac{-1}{\alpha-1}}\right)^\alpha} = \left(\sum_{k=1}^{K} d_k(\mathbf{x})^{\frac{-1}{\alpha-1}}\right)^{-(\alpha-1)} \tag{2.5}$$

(The distance is taken to be zero if $d_k(\mathbf{x}) = 0$ for any cluster C_k.) This expression can be interpreted as a generalisation of the harmonic mean; indeed, when $\alpha=2$, it becomes the harmonic mean *per se* (up to the constant multiple K^{-1}).

It may be of interest to note that a more straightforward distance-to-domain function can be constructed, in which the raw membership values $z_k(\mathbf{x})$ in (2.4) are taken (instead of $z_k(\mathbf{x})^\alpha$) as the weightings of the distances $d_k(\mathbf{x})$, thereby constituting a genuine weighted average:

$$D_{\alpha;W}(\mathbf{x}) = \sum_{k=1}^{K} z_k(\mathbf{x}) d_k(\mathbf{x}) = \frac{\sum_{k=1}^{K} d_k(\mathbf{x})^{\frac{\alpha-2}{\alpha-1}}}{\sum_{k=1}^{K} d_k(\mathbf{x})^{\frac{-1}{\alpha-1}}} \tag{2.6}$$

The value $\alpha=2$ is often taken purely for computational expediency. However, (2.5) and (2.6) demonstrate a remarkable significance to that particular value. When $\alpha=2$, it makes no difference whether one uses $z_k(\mathbf{x})$ or $z_k(\mathbf{x})^\alpha$ for the weightings: in either case the distance-to-domain measure is proportional to the harmonic mean of the distance-to-cluster measures:

$$K^{-1}D_{2;W}(\mathbf{x}) = D_2(\mathbf{x}) = \left(\sum_{k=1}^{K} d_k(\mathbf{x})^{-1} \right)^{-1} \tag{2.7}$$

2.4 Assembling the Distance

Although the *K*-means criterion (2.2), being distance-oriented, favours clusters that are roughly spherical, the intelligent *K*-means algorithm does tend to result in clusters being more voluminous further from the origin. We can take this into account by using yet another distance-to-cluster measure, tempered by cluster radius as follows:

$$d_k(\mathbf{x}) = \frac{(\mathbf{x} - \mathbf{c}_k) \cdot (\mathbf{x} - \mathbf{c}_k)}{r_k^{\,2}} \tag{2.8}$$

where r_k^2 is the 95th percentile value of $(\mathbf{x}_i - \mathbf{c}_k) \cdot (\mathbf{x}_i - \mathbf{c}_k)$ over $i \in C_k$.. Clusters with only one point (or in which all points coincide in descriptor space) are disregarded.

Finally, we rescale to allow a distance-to-domain of 1 to be interpreted as a nominal threshold for being inside the domain:

$$Dc(\mathbf{x}) = \frac{D_2(\mathbf{x})}{A} \tag{2.9}$$

where A is the 95th percentile value of $D_2(\mathbf{x}_i)$ over the whole training set ($i \in T$). This means that 5% of compounds will be nominally outside the domain – typically only just so – and ensures that the distance-to-domain measure is not unduly influenced by outlying points.

3 Results and Discussion

Two experiments are presented for assessment of the practical utility of the cluster-based distance-to-domain measure. Firstly, an 'internal validation' experiment was used to test whether retraining the distance-to-domain on a subset yields a comparable measure. Secondly, an 'external validation' experiment was performed to investigate correlation between distance to domain and prediction error in a multiple regression model.

3.1 Datasets

Two datasets were used for the experimentation. Internal validation was based on a training set compiled within IDBS for a log*P* model, consisting of 13066 compounds. The IDBS PredictionBase software [12] was used to identify descriptors with high (individual) correlation with experimental log*P*, and acceptable Shannon entropy values [13], over that training set. On this basis, ten topological and information-content descriptors [14] were selected.

For the external validation, a model for toxicity of phenols described by Cronin *et al* [15] was used. This was based on a training set of 185 compounds and 12 descriptors, with a further 50 compounds used to form the external validation set. Predicted activity values were derived from the model as trained using multivariate least-squares fitting (LSF) in all 12 descriptors, yielding a coefficient of multiple correlation of r^2=0.83. The IDBS PredictionBase software [12] was used to verify the suitability of this model in terms of stability and predictivity. Although the model is stable (with a leave-one-out coefficient of multiple correlation of q^2=0.83), there is some variation in the quality of predictions over the external validation set (with a root mean square prediction error of 1.73 model standard deviations); see Figure 2. We shall investigate whether the poorer predictions are associated with a higher distance-to-domain.

3.2 Internal Validation

We use 10-fold cross-validation to assess the stability of the cluster-based distance-to-domain measure, retraining the measure on a subset of the original training set and recalculating distances for the remaining points. We then check that the measures are concordant.

Recall that we are interested in whether a compound is outside the domain and, if so, by how far. However, we are less concerned with the quantitative distance-to-domain value of a

compound that is *inside* the domain. We reflect this in our validation by applying a 'clamping' function $g_t(d) = \max(d, t)$ to the distance-to-domain values d to force them to be at least as great as some minimum threshold value t. A range of values of t from 0.7 to 1.3 was used to investigate the influence of the compounds near the nominal boundary of $D(\mathbf{x}) = 1$.

Formally, the cross validation procedure can be described as follows:

1. Train the distance-to-domain measure D on the entire dataset T.
2. Record $D(\mathbf{x}_i)$ for all $i \in T$.
3. Randomly partition T into 10 equal-sized groups $T_1 \ldots T_{10}$.
4. For each of the 10 groups T_j:
    a. Retrain the distance-to-domain measure $D^{(j)}$ on the depleted dataset $T \backslash T_j$ formed be leaving out group T_j.
    b. Compute Δ_j, the root mean square value of the relative differences $[\, g_t(D^{(j)}(\mathbf{x}_i)) - g_t(D(\mathbf{x}_i)) \,] \,/\, g_t(D(\mathbf{x}_i))$ over T.

This procedure was applied to the latest distance measure $D(\mathbf{x})=Dc(\mathbf{x})$ in (2.9), and to the following four additional distance-to-domain measures (in normalised descriptor space) for comparison:

Bounding Box: variant in which the distance-to-domain is taken to be the maximum squared normalised descriptor value: $D(\mathbf{x}) = D([x_1, \ldots, x_d]) = \max \{x_1^2, \ldots, x_d^2\}$;

Leverage: using normalised value $N\, h(\mathbf{x}) \,/\, 3(d+1)$ where d is the dimension of the descriptor space [7];

Nearest Neighbours *k*-NN: mean squared distance to nearest 10 training points, normalised analogously to equation (2.9) such that 95$^{\text{th}}$ percentile value of this measure over the training set is 1;

Cluster-Based: exactly as derived in §2.4 except without taking cluster radius into account: i.e. taking $r_k^2=1$ in equation (2.8).

A training set of around 13000 compounds and 10 descriptors was used. The same 10-fold partitioning was used in step 3 for each of the five measures, yielding the results displayed in Table 1.

Table 1 shows that the cluster-based distance-to-domain measures have stabilities that compare will with the existing measures. Indeed, outside the nominal boundary of $D(\mathbf{x})=1$, our cluster-based distance-to-domain measure $Dc(\mathbf{x})$ is the most stable of those analysed. The minor loss of instability inside that boundary is due to the fact that, close to a centroid, the cluster-based distance-to-domain measures start to approximate the distance to the nearest centroid, and therefore become sensitive to the details of the clustering. The *k*-NN method is unsurprisingly the least stable: its dependence on each individual point in the training set understandably gives rise to significantly altered measures on a reduced training set.

Although the Δ_j values in step 4b were based on relative differences (chosen because they render Δ_j invariant under rescaling the distance-to-domain measure), similar qualitative results were obtained for four of the five methods when the experiment was rerun using absolute differences. The exception was the leverage method, which became noticeably less stable than the other methods. This can be attributed to near-singularity of the curvature matrix (X^TX) in certain directions in descriptor space causing sensitivity to small changes; the corresponding absolute deviations in leverage along these directions will then grow quadratically with **x**. Still, with the relative differences, the leverage method came as the second best.

3.3 External Validation

The original motivation for distance-to-domain analysis is to identify where reliability of a prediction may be compromised due to lack of similar compounds in the training set. In order to verify this relationship between domain of applicability and reliability of prediction, the distance-to-domain measure was applied to an extra validation set compounds with known biological activity values. The root mean square error in prediction over the whole validation set was compared with the corresponding value obtained by considering only those compounds inside the domain.

The linear least squares fitting (LSF) model of phenol toxicity [15] described in §3.1 was used. The means of squared prediction errors were first calculated over the whole validation set. Then, for each distance-to-domain measure and for a number of threshold values, the mean squared errors were recalculated over those compounds with distance-to-domain less than the threshold. The results are shown in Table 2 and Figure 3.

Once again the results come out in favour of the new cluster-based methods derived in §2.4. Both these new methods are successful in defining domains with improved model predictivity, even slightly outperforming the *k*-NN method: it would appear that the extra degree to which *k*-NN produces a fine-grained model of the dataset is not capturing any extra information on the domain from which the dataset is drawn as compared with the cluster-based model.

Considering the 'D<1.0' row, we can form an F-statistic from the ratio of the mean-square error 1.657^2 (of the 46 compounds inside the domain) to the corresponding mean-square error 2.444^2 of the 4 compounds outside the domain. This F-statistic of 0.459 (with 46 and 4 degrees of freedom) is 91% significant.

The leverage measure does not perform so well, with only a marginal enrichment in predictivity on disregarding the high-leverage validation compounds. Its corresponding F-

statistic for the ‘D<1.0’ case is only 78% significant. These results do not invalidate the statistical theory behind leverage, but rather highlight the different assumptions made. In deriving the cluster-based method we assume that an approximately linear model holds in the neighbourhood of the training data. The statistical analysis of leverage, on the other hand, assumes that *some* linear model is applicable globally, and that prediction errors far from the dataset arise solely from errors in estimating the model parameters.

4 Conclusions

A new cluster-based measure of distance-to-domain for descriptor-based QSAR models was derived, designed to overcome existing methods’ difficulties of crudeness and computational complexity. The measure combines the intelligent *K*-means clustering algorithm with a new interpretation of a conventional optimisation criterion in fuzzy clustering, leading, somewhat surprisingly, to a modified harmonic mean measure.

Experimentation demonstrated that this distance-to-domain measure is both more stable than existing methods, and more indicative of prediction error.

The computational complexity of this cluster-based method also proved to be quite acceptable. Modelling the domain using clusters affords a much faster calculation than in *k*-NN, as well as more parsimonious in terms of additional stored model parameters, without compromising the acuity of the measure.

References

(1) CEFIC, *(Q)SARs for Human Health and the Environment*, in workshop report *Report on Regulatory Acceptance of (Q)SARs*, Setubal, Portugal, **2002**.

(2) R. P. Sheridan, B. P. Feuston, V. N. Maiorov, S. K. Kearsley, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1912-1928.

(3) J. Jaworska, N. Nikolova-Jeliazkova, T. Aldenberg, *ATLA* **2005**, *33*, 445-459.

(4) J. A. Fernández Pierna, F. Wahl, O. E. de Noord, D. L. Massart, *Chemom. Intell. Lab. Syst.* **2002**, *63*, 27-39.

(5) L. Eriksson, J. Jaworska, A. P. Worth, M. T. D. Cronin, R. M. McDowell, *Environ. Health Perspect.* **2003**, *111*, 1361-1375.

(6) A. Tropsha, P. Gramatica, V. K. Gombar, *QSAR Comb. Sci.* **2003**, *22*, 69-77.

(7) E. Parzen, *Ann. Math. Stat.* **1962**, *33*, 1065-1076.

(8) A. Golbraikh, A. Tropsha, *J. Mol. Graph. Mod.* **2002**, *20*, 269-276.

(9) B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*, Chapman & Hall/CRC, London, **2005**.

(10) A. Smellie, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1929-1935.

(11) J. C. Bezdek, S. K. Pal (Eds.), *Fuzzy Models for Pattern Recognition*, IEEE Press, New York, **1992**.

(12) http://www.idbs.com/PredictionBase/

(13) J. W. Godden, F. L. Stahura, J. Bajorath, *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 796-800.

(14) J. Devillers, A. T. Balaban (Eds.), *Topological Indices and Related Descriptors in QSAR and QSPR*, Gordon and Breach Science Publishers, Amsterdam, **1999**.

(15) A. O. Aptula, N. G. Jeliazkova, T. W. Schultz, M. T. D. Cronin, *QSAR Comb. Sci.* **2005**, *24*, 385-396.

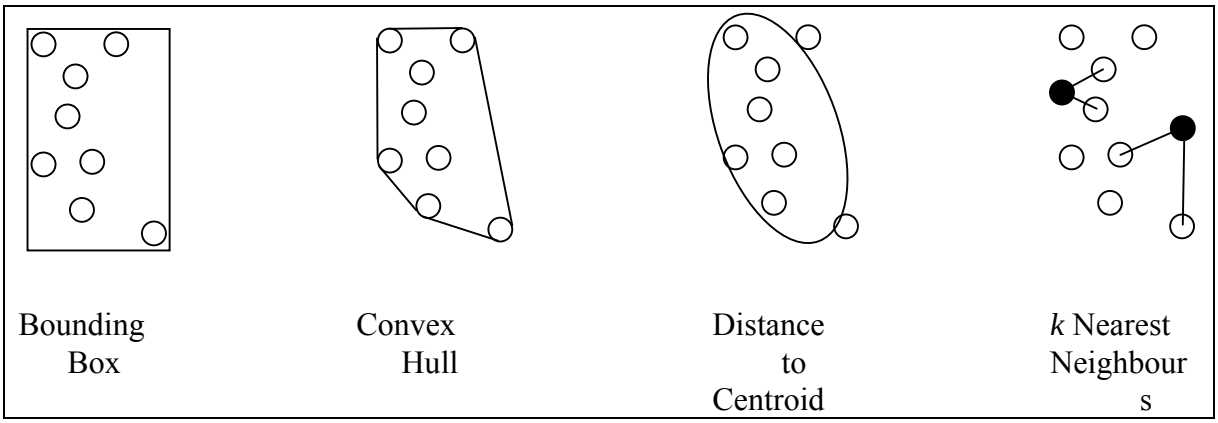

Figure 1. Existing measures of domain of applicability.
For the "bounding box" and "convex hull" methods, the domain of applicability is the smallest axis-aligned rectangular box or convex region (respectively) containing all data points. For the "distance to centroid" method, the ellipsoidal region is centred on the data set's grand mean, and has its principal axes based on the eigenvectors of the data set's variance/covariance matrix. In the "*k* nearest neighbours" method (illustrated here with k=2), the distance of a point to the domain is taken as the averaged distance to the *k* nearest data points.

Figure 2. Analysis of Predictivity Using IDBS PredictionBase Software.
The left-hand graph plots the 185 training structures for the phenol toxicity model of Cronin *et al* [15], while the right-hand graph depicts the 50 external validation structures. In each case the experimentally determined 50% growth inhibition concentration (IGC50) value is plotted against the calculated (predicted) value. The 95% confidence regions are shaded, constituting 1.96 model standard deviations above and below of the leading diagonal.

Figure 3. External validation of distance-to-domain measures.
For each of the five distance-to-domain measures used (bounding box, leverage, *k*-NN, cluster-based and D_C), root mean square prediction error is plotted against the number of structures deemed to be inside the domain, for a variety of threshold distance values.

Table 1. Internal validation of distance-to-domain measures.

threshold distance[c]	Bounding Box[a]	Leverage[a]	*k*-NN[a]	Cluster-Based[a,b]	$Dc(\mathbf{x})$[a,b]
0.70	0.0502	0.0388	0.0498	0.0491	0.0472
0.75	0.0458	0.0366	0.0471	0.0454	0.0413
0.80	0.0422	0.0346	0.0446	0.0423	0.0366
0.85	0.0392	0.0329	0.0424	0.0395	0.0328
0.90	0.0368	0.0313	0.0404	0.0371	0.0296
0.95	0.0347	0.0299	0.0386	0.0351	0.0269
1.00	**0.0330**	**0.0287**	**0.0369**	**0.0333**	**0.0245**
1.05	0.0316	0.0275	0.0355	0.0316	0.0223
1.10	0.0303	0.0265	0.0346	0.0301	0.0203
1.15	0.0291	0.0255	0.0329	0.0288	0.0185
1.20	0.0281	0.0246	0.0318	0.0275	0.0170
1.25	0.0271	0.0237	0.0309	0.0264	0.0156
1.30	0.0262	0.0229	0.0300	0.0254	0.0145

[a] These figures relate to the IDBS LogP dataset consisting of 13066 compounds, described by 10 descriptors. 10-fold cross-validation was performed on this dataset: each distance measure was trained on the whole training set and compared with the same measure retrained on the remaining compounds.

[b] Clustering this dataset resulted in 17 clusters being generated.

[c] Each row tabulates root mean square relative deviations of distance-to-domain, subject to a minimum threshold distance, averaged over the 10 cross-validation iterations.

Table 2. External validation of distance-to-domain measures.

	Bounding Box[c]	Leverage[c]	*k*-NN[c]	Cluster-Based[b,c]	*Dc*(**x**)[b,c]
Entire validation set[a]	1.733 (50)	1.733 (50)	1.733 (50)	1.733 (50)	1.733 (50)
inside domain: $D<0.8$	1.018 (2)	1.701 (44)	1.667 (45)	1.646 (44)	1.667 (45)
inside domain: $D<0.9$	1.338 (8)	1.707 (47)	1.667 (45)	1.667 (45)	1.667 (45)
inside domain: $D<1.0$	1.754 (27)	1.707 (47)	1.667 (45)	1.657 (46)	1.657 (46)
inside domain: $D<1.1$	1.686 (40)	1.707 (47)	1.705 (47)	1.649 (47)	1.657 (46)
inside domain: $D<1.2$	1.669 (41)	1.707 (47)	1.705 (47)	1.649 (47)	1.657 (46)

[a] The figures in this table are based on a toxicity model in 12 descriptors trained using 185 compounds and validated using 50 external compounds. The model has coefficient of multiple correlation $r^2=0.831$.

[b] Clustering this training set resulted in six clusters being generated.

[c] The 'prediction error' statistics tabulated here are root mean squared prediction errors expressed as a multiple of the model's standard deviation. They were calculated over the whole validation set (first row), and, for each distance-to-domain measure, over only those compounds inside the domain according to various thresholds of that measure (second and subsequent rows). In each case the number of validation compounds involved is displayed in parentheses.