Similarity to Molecules in the Training Set Is a Good Discriminator for Prediction Accuracy in QSAR

Robert P. Sheridan,*,† Bradley P. Feuston,‡ Vladimir N. Maiorov,† and Simon K. Kearsley†

Molecular Systems Department, RY50S-100 Merck Research Laboratories, Rahway, New Jersey 07065, and Molecular Systems Department, WP53F-301 Merck Research Laboratories, West Point, Pennsylvania 19486

Received July 12, 2004

How well can a QSAR model predict the activity of a molecule not in the training set used to create the model? A set of retrospective cross-validation experiments using 20 diverse in-house activity sets were done to find a good discriminator of prediction accuracy as measured by root-mean-square difference between observed and predicted activity. Among the measures we tested, two seem useful: the similarity of the molecule to be predicted to the nearest molecule in the training set and/or the number of neighbors in the training set, where neighbors are those more similar than a user-chosen cutoff. The molecules with the highest similarity and/or the most neighbors are the best-predicted. This trend holds true for narrow training sets and, to a lesser degree, for many diverse training sets and does not depend on which QSAR method or descriptor is used. One may define the similarity using a different descriptor than that used for the QSAR model. The similarity dependence for diverse training sets is somewhat unexpected. It appears to be greater for those data sets where the association of similar activities vs similar structures (as encoded in the Patterson plot) is stronger. We propose a way to estimate the reliability of the prediction of an arbitrary chemical structure on a given QSAR model, given the training set from which the model was derived.

INTRODUCTION

Quantitative Structure Activity Relationship (QSAR) methods relate molecular features to some kind of "response"—a biological activity or physical property. One purpose of QSAR models is to understand what molecular features give rise to activity. Another is to predict the response of molecules not yet seen. Typically a model is created from a "training set" of molecules and their responses, where the molecules are represented by some user-specified set of molecular descriptors (reviewed in refs 1 and 2). Once validated, the model is used to predict the response of new molecules not in the training set. Among the most important issues in QSAR having to do with the accuracy of prediction are overfitting (whether the model fits the idiosyncracies of a particular training set at the expense of the predictivity of a similar set of molecules) and extrapolation (whether a model can be applied to a new molecule not like those in the training set). The first has been considered for a long time, and a recent review[3] covers the major issues.

The second issue, however, is seldom addressed. As with any statistical models, QSAR models are limited in applicability by the data from which they are constructed. For example, if a QSAR model is trained on a narrow series of molecules, say benzodiazepines, we intuitively expect the model to apply to a new benzodiazepine but not to any arbitrary drug-like molecule. In contrast, if the training set is very diverse and very large, we would expect the model to apply to nearly any new molecule because it has been trained on a wider representation of chemical space. When the training set is large but contains only a few series, the situation is less clear. There is very little in the QSAR literature about how the reliability of a prediction falls as the new molecule departs from the training set. The standard suggestion is that one run cross-validation studies on one's data to get an overall rms error for prediction. This is a reasonable check for overfitting. However, as a measure of extrapolation it is deficient because it depends on the dubious assumption that the distribution of potential molecules to be predicted is very similar to that of the molecules used to train the QSAR model. Moreover, a single number cannot apply equally well to molecules very similar to those in the training set and those very different.

Extrapolation is an especially important issue now that QSAR models are commonly made available on corporate intranet Web sites. A user is allowed to sketch any arbitrary drug-like molecule and ask for a prediction. Since many QSAR models have been generated from a limited number of chemical classes (probably not at all like the sketched molecule), the reliability of such a prediction is likely to be poor, but there usually is no way for the user to know this, being unfamiliar with the data set from which the model was derived. We need a way of quantitating the reliability and presenting it to the user.

Generally it is felt that if a new molecule somehow resembles, or is in the "domain" or "space" of the training set, it is likely to be well-predicted, otherwise there is significant "extrapolation" and the prediction is unreliable. The difficulty is in defining a useful definition of domain from which to measure extrapolation. A simplifying assumption is that the domain can be defined independently of the activities in the training set or which molecular features are important for activity. As an example of a very simple definition, one may define the domain as the multidimensional rectangle containing a given fraction (say 95%

* Corresponding author e-mail: sheridan@merck.com.
† RY50S-100 Merck Research Laboratories.
‡ WP53F-301 Merck Research Laboratories.

10.1021/ci049782w CCC: $27.50
Published on Web 10/07/2004

or 100%) of the training set, each dimension corresponding to a chemical descriptor. See Figure 1A for a cartoon diagram. If a molecule (e.g. points 1 and 2) falls into the rectangle, it is in the domain of the training set. A more sophisticated method is to take descriptor correlations into account as in Figure 1B. One determines whether the new molecule is in the multidimensional ellipsoid containing the training set (e.g. points 1 and 2). In linear regression statistics,[4] for instance, there is a specific measure of how far outside the ellipsoid a molecule is: $h_{00} = \mathbf{x}^T (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}$, where **X** is the matrix of molecules and descriptors in the training set and **x** is the descriptor vector of the molecule to be predicted. Of course, not all the space inside the ellipsoid is necessarily covered by the training set. A different definition of domain might take into account local neighborhoods around the training set molecules as in Figure 1C. Point 2 is an example of a new molecule that falls within the ellipsoid but is not actually near any training set molecules. It is not clear whether it will be well-predicted or not.

Chemical descriptors can be divided into two basic categories: whole-molecule and substructure. The user is referred to refs 1, 2, 5, and 6. Whole-molecular descriptors define a molecule by a series of numbers, each number capturing a feature of the entire molecule, e.g. molecular weight, logP, number of H-bond donors, Wiener index, WHIM descriptors, etc. In a substructure description a molecule is represented by a set of 2D or 3D substructures and (sometimes) the frequencies at which the substructures occur in the molecule. Daylight fingerprints, UNITY fingerprints, ISIS keys, etc. are examples. For QSARs using only whole-molecular descriptors, one can use the paradigms in Figures 1A−C without too much complication. The new molecule is always in the space of the training set because one can always define a molecular weight, logP, etc. for the new molecule.

However, many QSAR models are based on substructure descriptors, which are useful because biological activities are often better predicted by the presence or absence of a particular group of atoms in a molecule than by its global properties, and the descriptors can be mapped back onto the molecules to indicate which parts confer activity. Using substructure descriptors has at least three complications as far as defining a domain of a training set. First, the dimensionality is usually much higher, i.e., several thousand unique substructures in a typical training set vs a few dozen whole-molecule descriptors. Second, a new molecule may have substructures that do not occur in the training set, i.e., it can be outside the space of the training set. Third, it is possible for two substructure descriptors to be perfectly correlated; in that case some mathematical operations such as taking the inverse of a matrix, as in h_{00}, are impossible. In this work our domain definitions will be compatible with substructure descriptors.

In this paper we do the following:

1. We propose retrospective cross-validation of data sets as a way to test the ability of various "extrapolation measures" to discriminate well-predicted vs poorly predicted molecules (Methods section 1).

2. We propose a number of extrapolation measures (Methods section 6) to indicate how far a molecule to be predicted departs from a training set. We show on an example data set, using a particular QSAR method (random_forest) and descriptor (AP) as the standard, that two measures are

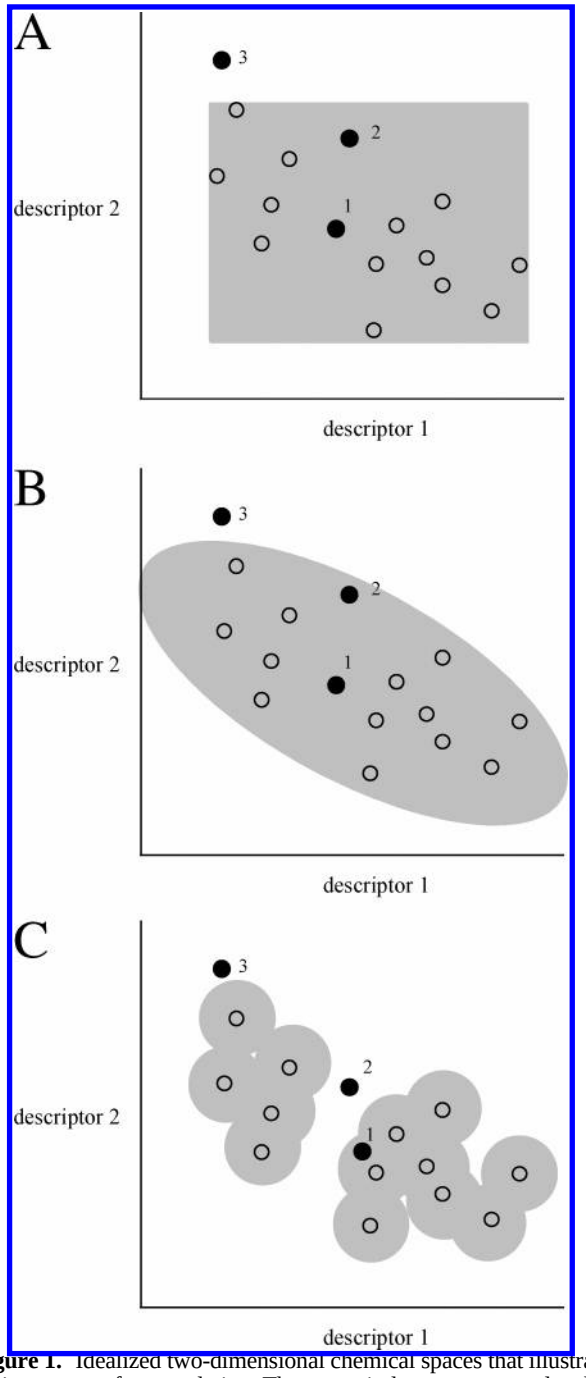


Figure 1. Idealized two-dimensional chemical spaces that illustrate various types of extrapolation. The open circles represent molecules in the training set. The filled circles labeled 1, 2, and 3 represent new molecules to be predicted. The shaded areas represent the "domain" of the training set. A. Domain defined by the range of the descriptors in the training set. In this case molecules 1 and 2 are inside the domain and presumably can be predicted by a QSAR model derived from the training set. Molecule 3 is outside the domain, and its prediction is presumably less reliable. B. Domain defined by an ellipsoid containing the training set. This paradigm takes correlations between the descriptors into account. Molecule 1 is close to the center of the ellipse and presumably can be predicted well. Molecule 2 is near the border, and its prediction is uncertain. Molecule 3 is outside the ellipsoid, and presumably its prediction is less reliable. C. Domain defined by the neighborhood of the training set molecules. Molecule 1 is near a few molecules in the training set and is presumed to be well-predicted. Molecules 2 and 3 are not near any molecules in the training set, although molecule 2 would be in the ellipse by the definition of B.

good discriminators: similarity (Dice definition, AP descriptor) to the nearest compound in the training set and number of neighbors in the training set. Molecules with higher similarity and/or more neighbors are better predicted (Results section 3). Plots of the root-mean-square difference between observed and predicted activity vs these extrapolation measures are a good way of showing this trend. The trend holds true for training sets that consist of a single or a few chemical series and (to a lesser degree) for some training sets that are diverse. This is demonstrated for a number of in-house data sets (Results sections 4 and 5).

3. We show that the above trend is not sensitive to the size of the cross-validation sets (Results section 6), the QSAR method (Results section 7), or the descriptors used for the QSAR (Results section 8). We also demonstrate that the descriptor used to calculate the extrapolation measures does not have to be the same as the descriptor used for the QSAR (Results section 9). The addition of artificial noise to a data set tends to reduce the dependence of predictivity on the extrapolation measures (Results section 10).

4. That there should be dependence of predictivity vs the extrapolation measures for diverse training sets is unexpected. Some in-house data sets show strong dependence, some weaker. We show that the strength of the dependence is related to the "neighborhood behavior" of the individual data set (Results section 11).

5. Finally, we propose (Discussion) a method by which the "error bar in prediction" for an arbitrary drug-like molecule against a given QSAR model can be displayed to the user.

METHODS

1. Overview. We do retrospective cross-validation experiments to simulate what happens when a QSAR model generated on a training set of molecules is applied to a new set of molecules. We start with a data set of structures and their corresponding activities (see Methods section 10).

1. A user-specified number of molecules is randomly selected from the original data set or a subset therefrom (section 5). These form a "training set". Molecules not in the training set form the "test set".

2. A QSAR model is generated from the training set. This involves selecting a descriptor set (section 2) and a QSAR method (section 4).

3. The model is used to predict the activity of all molecules in the original data set and note is made of which were in the training set.

4. Each molecule in the original data set is assigned an extrapolation measure of how close it is to the training set. This requires defining such a measure (section 6).

Steps 1−4 can be repeated (here 10 times) to ensure better statistics.

5. The absolute difference between predicted and observed activity for each molecule (ABSDIFF) is plotted against each candidate extrapolation measure. Similarly, root-mean-square ABSDIFF may be plotted against the extrapolation measure (section 8).

2. Descriptors. Descriptors are used for QSAR and to calculate similarity. We use the descriptors listed below. The first three are substructure descriptors, the last two are whole-molecule descriptors.

1. The regular atom pair AP.[7]

2. The topological torsion TT.[8]

3. Daylight fingerprints, DF (www.daylight.com).

4. The E-state descriptors [9] as implemented in CERIUS2 (www.accelrys.com). E-state descriptors depend on user-defined formal charge states. We assume a charge state appropriate for pH 7.4.

5. WHIM descriptors:[10] These depend on 3D structures, which are generated by CORINA [11] version 2.4.

Overall the AP descriptor gives the best predictions for any given QSAR method, so that will be our standard descriptor.

3. Similarity. Some of the extrapolation measures and one of the QSAR methods use the notion of chemical similarity. There are many possible definitions of similarity based on substructure descriptors (cosine, Tanimoto, Dice, etc.). Unless otherwise stated, we will use the Dice definition with the AP descriptor,[7] and that will be our standard.

4. QSAR Methods. We use the following methods:

1. random_forest:[12,13] Random_forest is an ensemble recursive partitioning method, where a random subset of descriptors is used at each branching of the tree. We use an ensemble of 100 trees in all cases.

2. trendvector:[14] Our implementation is a samples-based, partial-least squares on the presence or absence of substructure descriptors.

3. mrlnet: This is an ensemble artificial neural network method. We use 100 neural nets. The input parameters are the principal components of the original descriptors. There are 5 nodes in the hidden layer.

4. k-NN: For regression problems, the activity of the molecule to be predicted is taken as the mean activity of the k nearest neighbors in the training set as defined by Dice similarity. Here we use $k = 5$.

5. Support Vector Machine (SVM):[15] We use the SVM-light implementation[16] (www.cs.cornell.edu/People/svm_light) with the linear kernel.

It should be noted that trendvector and mrlnet weight the molecules so that no one chemical class can dominate. All methods are used with the "out of the box" defaults with no attempt to tune adjustable parameters to get the best fit on a particular data set. Most of the QSAR methods (with the possible exception of k-NN) are not particularly sensitive to large numbers of irrelevant descriptors, so we do no descriptor selection. Averaged over many data sets, random_forest has proven to give the best predictions and is the least sensitive to adjustable parameters, so that will be our standard QSAR method.

5. Training Sets. In practice QSAR data sets may be narrow (i.e. consist of one or a few chemical series) or diverse (many chemical series), and we wish to simulate all possibilities, so our training sets will be of three types: narrow, random, and disparate. Narrow training sets are generated by clustering the original data set using the method of Butina[17] using the AP descriptor and a similarity cutoff of 0.7. A user-defined number of those molecules (usually about half the total number) in the n largest clusters are randomly picked and assigned to the training set. "narrow-1" training sets are from the single largest cluster and can be thought of as a single chemical series. "narrow-5" training sets are from the 5 largest clusters. This simulates training data consisting of a few narrow series.

"Random" training sets are generated by randomly selecting a user-defined number (usually 500) of the molecules in the original data set. These training sets have a diversity similar to that of the original set. "Disparate" training sets are generated by randomly selecting a user-defined number (usually about half) of the cluster centers. These training sets are more diverse than a random selection.

Note that the training sets are small compared to the original sets so as to have many molecules for which the prediction will involve true extrapolation.

6. Extrapolation Measures. These are some measures of how close a test set molecule is to the training set:

1. Fraction of unique descriptors in the range of the training set. There are several flavors of this: (a) The fraction of unique descriptors in a test set molecule that are also in the training set, counting the presence or absence only. This is based on the original suggestion of Carhart et al.[7] for when a new molecule can be predicted by trendvector. The suggested cutoff is 0.95 for AP. (b) The fraction of descriptors in the test set molecule in the range of descriptors in the training set, taking into account the numerical frequencies of the descriptors. If one assumes the distribution of values for any given descriptor over all the molecules is normal, one can say that a descriptor is in the range if it falls between -3 and $+3$ standard deviations from the mean for that descriptor. (c) Alternatively, one can use the percentiles, so for instance, a descriptor is in the range if it is higher than the 10% percentile of the descriptor distribution and lower than the 90% percentile.

Note that a new descriptor (not in any of the training set molecules) is by definition outside the range of the training set, since the frequency of that descriptor is zero for all molecules in the training set.

2. Similarity to molecules in the training set. (a) The mean similarity of a test set molecule to the most similar k members of the training set, using all the descriptors. We tried $k = 1, 3, 5$ (called SIMILARITYNEAREST1, -3, and -5, respectively). A variation of this is to calculate this similarity using only the subset of descriptors in the training set. (b) The similarity of the test set molecule to the descriptor centroid of the training set.[18]

3. Count of neighbors in the training set. (a) The simplest method is to pick a similarity cutoff above which the test set molecule and a training set molecule can be considered "neighbors" (here 0.7 for AP, where compounds appear to be obvious analogues by eye) and count the molecules in the training set that exceed that cutoff. (b) A more complicated approach is to define a falloff function f(Similarity) that defines how much of a neighbor a training set molecule is as a function of similarity to the test set molecule. The number of neighbors is the sum of f(Similarity) over all members of the training set. For instance, we use a linear function where a molecule is 1.0 of a neighbor at similarity 1.0 and 0.0 of a neighbor at a similarity 0.6; anything between is linearly interpolated.

7. Statistics. The conventional way of summarizing "lack of fit" in QSAR models is the root-mean-square (rms) difference between observed and predicted activities. Another is the square of the correlation between observed and predicted (R^2). What rms error is equivalent to a prediction being "no better than guessing"? One method is to scramble the correspondence of predicted and observed values and take the rms of the differences. This will be referred to as the "scrambled rms".

8. Rms Plots. We will produce a number of scatterplots of the absolute difference between predicted and observed activity (ABSDIFF) vs various extrapolation measures, where each point in the plot represents the prediction of one molecule. (It is assumed that the deviations are equally likely to be positive or negative.) We wish to summarize the statistics in these plots by generating a new plot that shows the root-mean-square ABSDIFF (rms-ABSDIFF) on the *y*-axis vs the extrapolation measure on the *x*-axis. One calculates the rms-ABSDIFF for the molecules within a sliding window over *x*, the nature of the window being appropriate for the specific measure. Note that rms-ABSDIFF is mathematically equivalent to the rms error usually used to summarize QSAR fits, except that we are calculating rms-ABSDIFF for subsets of predictions in a window instead of all predictions.

Plots of this type are necessary to confirm trends as a function of the extrapolation measure because the eye can sometimes see false trends due to uneven distribution in *x*. For instance, the eye tends to perceive an increase in *y* based on the maximum values of *y* in a scatterplot. However, regions of *x* with the most points will naturally have the highest *y*'s purely by chance, even if there is no real trend in *y* vs *x*. The calculated rms-ABSDIFF is not misled by this.

9. Neighborhood Behavior. Patterson et al.[19] proposed a plot by which one could monitor, given a set of chemical descriptors, how likely structurally similar molecules were to have similar activities. Each point in the plot represents a pair of molecules in the training set. As originally formulated, the *x*-axis is the distance between the molecules in the descriptor space and the *y*-axis is the absolute difference in activities. If there is a good "neighborhood behavior" for that activity and descriptor, almost all points will appear in the lower triangle. That is, molecules that are close in descriptor space will almost always have small differences in activities. On the other hand, molecules far apart in descriptor space may have very small or very large differences in activities. Despite recent criticisms of the idea that similar molecules should have similar activities,[5,6] the Patterson plot shows that, given an appropriate descriptor, neighborhood behavior is present in most data sets of carefully measured activities. For substructure descriptors it is natural to redefine the "distance" in the Patterson plot as 1-Similarity. Our experience with large diverse data sets and substructure descriptors is that such plots always have a lower triangle appearance. However, as before, appearances can be deceiving. There are relatively few pairs of similar molecules, so the maximum value of absolute difference is likely to be small on the left side of the plot even if there is no real trend in absolute difference vs 1-Similarity. We therefore use the following quantitative measure of neighborhood behavior: Calculate the mean absolute difference in activity for all the pairs in the data set. Then calculate the mean absolute difference for those pairs where the molecules have an AP Similarity ≥ 0.7 (or 1-Similarity ≤ 0.3 in the Patterson plot). We call the ratio of these numbers the "Patterson ratio". If the ratio is 1, there is no neighborhood behavior. Larger ratios mean stronger behavior.

10. Data Sets. For our purpose we use in-house data sets with the following characteristics: The activity data should

Table 1. Data Sets Used in This Study

data set	description	total number	clusters by Butina algorithm	range of activity (orders of magnitude)	fraction observations ignored when calculating rms-ABSDIFF
LOGD	shake flask water/octanol partition at pH 7.4	11283	1974	10.6	0.00
LOGSOL	molar solubility at pH 7.4	4624	957	7.8	0.00
PKA	pK_a of ionizable group	2686	954	12.4	0.00
GPCR1	binding to G-protein coupled receptor	3626	697	6.0	0.32
GPCR2	binding to G-protein coupled receptor	3920	582	7.3	0.08
GPCR3	binding to G-protein coupled receptor	2274	198	8.8	0.48
GPCR4	binding to G-protein coupled receptor	5239	655	6.1	0.00
GPCR5	binding to G-protein coupled receptor	2933	158	5.5	0.00
GPCR6	binding to G-protein coupled receptor	7636	516	8.3	0.06
GPCR7	binding to G-protein coupled receptor	3177	406	5.5	0.06
CHAN1	binding to channel protein allosteric site	7455	1102	5.3	0.00
CHAN2	binding to channel protein	23125	4295	11.9	0.17
ENZ1	inhibition of enzyme	6337	1507	6.6	0.02
ENZ2	inhibition of enzyme	5941	1111	5.4	0.13
ENZ3	inhibition of enzyme	4219	415	11.7	0.00
ENZ4	inhibition of enzyme	3025	563	6.3	0.00
ENZ5	inhibition of enzyme	6924	1133	8.8	0.05
NHR1	binding to nuclear hormone receptor	2856	630	5.3	0.22
NHR2	binding to nuclear hormone receptor	5864	942	7.0	0.30
CA	binding to cell-surface adhesion protein	4339	516	7.8	0.00

be quantitative (e.g. IC_{50}, EC_{50}, or K_i), there should be a good range of activity (at least 5 orders of magnitude), the data set should contain many chemical classes (there should be at least 100 clusters by the Butina algorithm), and the activities should be fittable by our in-house QSAR methodology and descriptors. Data sets used in this study are summarized in Table 1. There are three physical properties, plus binding to various receptors, inhibition of enzymes, etc. We use -log(IC_{50}), etc. as the activity. If there is more than one determination for the same molecule, the mean -log(IC_{50}) is used.

Data sets from pharmaceutical companies have a number of complications. One is that large data sets are likely to contain a small fraction of errors, both in the activity value and in the assignment of chemical structure to the sample being measured. Another is that certain classes of molecules (typically the more active ones) are likely to be overrepresented. While the trendvector and mrlnet methods compensate for overrepresentation in the training set, the other methods do not, but generally there is no major problem when there is no compensation. A third complication is that data sets often contain many indefinite values indicative of an upper limit of the measurement for the activity, e.g. "IC_{50} > 100000 nM", equivalent to "-log(IC_{50}) < −5". It is desirable to keep these values when generating the QSAR model (they indicate that some molecules are very inactive). However, they add noise to calculations of ABSDIFF because we do not know the true observed value. Indefinite activities accounting for ≥5% of the activities in a data set are noted. Molecules with these observed activities are ignored when calculating rms for the overall QSAR statistics and for the rms-ABSDIFF plots.

RESULTS

1. Overall Statistics. Statistics for the predictions of the data sets with the random_forest method and AP descriptor, which we consider the standard for further comparison, are in Table 2. Generally the test set predictions for the narrow-1 and narrow-5 training sets are very poor, indicated by the low R^2 and the rms error close to the scrambled rms. This is expected: since the vast majority of molecules to be predicted do not resemble anything in the training set, the majority of predictions will be poor. Two exceptions seem to be the ENZ3 and NHR2 narrow sets. (More on that below.) On the other hand, the rms error from the random and disparate sets is usually much lower than the scrambled rms. Again this is expected: if the training set contains enough diverse molecules, the QSAR model derived from it should be able to predict most new molecules.

Note that the rms errors in Table 2 and subsequent tables indicate the error averaged over all molecules in test set, whereas below we will show that some test set molecules will tend to be better predicted than average depending on their extrapolation measures, for example the number of neighbors in the training set.

2. Plots of Observed vs Predicted. We will discuss the LOGD data set as an example in detail. Among the data sets, it shows what we regard as the most ideal behavior. One possible reason is that LOGD, as a physical property, does not depend strongly on stereochemistry, so there are no misleadingly high ABSDIFFs due to stereoisomers, which have identical AP descriptors, but can have very different observed activities on receptors or enzymes. Also, we do not expect the "activity" to be confined to specific chemical classes. Finally, it is a large data set for which good statistics are available. Figure 2 shows observed vs predicted for two LOGD training sets, using the AP descriptor and the random_forest method. Clearly, the training set is always well-predicted as expected for random_forest. For the narrow-1 training set (Figure 2A), the majority of molecules in the test set are poorly predicted, falling far off the diagonal. This is consistent with the high rms error and low R^2 in Table 2. In contrast, for the random training set (Figure 2B) most molecules are not too badly predicted, as indicated by most points being near the diagonal. The plots for narrow-5 and disparate training sets (not shown) are very similar to those for narrow-1 and random, respectively. The plots for the other data sets are qualitatively very similar to the corresponding plots for LOGD. The exceptions are again ENZ3 and NHR2. For them, plots for the narrow sets show much less scatter.

Table 2. QSAR Statistics for Data Sets Using the random_forest Method and the AP Descriptor

training set	Ntraining	rms error test set	rms error test set by scramble	R^2 for test set	training set	Ntraining	rms error test set	rms error test set by scramble	R^2 for test set
LOGD narrow-1	100	2.41	2.48	0.08	CHAN1 narrow-1	100	0.90	0.94	0.05
LOGD narrow-5	250	1.61	2.06	0.30	CHAN1 narrow-5	400	0.71	0.87	0.21
LOGD random	500	1.08	2.36	0.68	CHAN1 random	500	0.58	0.92	0.47
LOGD disparate	500	1.08	2.35	0.68	CHAN1 disparate	500	0.60	0.92	0.43
LOGSOL narrow-1	100	1.86	1.98	0.09	CHAN2 narrow-1	100	0.91	0.91	0.01
LOGSOL narrow-5	250	2.15	2.28	0.09	CHAN2 narrow-5	400	0.86	0.90	0.03
LOGSOL random	500	1.34	2.19	0.52	CHAN2 random	500	0.73	0.92	0.34
LOGSOL disparate	500	1.37	2.14	0.50	CHAN2 disparate	500	0.75	0.91	0.25
PKA narrow-1	30	1.84	1.86	0.00	ENZ1 narrow-1	80	1.58	1.62	0.01
PKA narrow-5	80	1.97	2.18	0.13	ENZ1 narrow-5	300	1.18	1.48	0.44
PKA random	500	1.45	2.05	0.49	ENZ1 random	500	0.92	1.69	0.69
PKA disparate	500	1.48	2.02	0.47	ENZ1 disparate	500	1.13	1.69	0.62
GPCR1 narrow-1	60	1.17	1.38	0.11	ENZ2 narrow-1	80	1.05	1.12	0.06
GPCR1 narrow-5	300	1.09	1.37	0.24	ENZ2 narrow-5	300	0.91	1.17	0.44
GPCR1 random	500	0.93	1.52	0.48	ENZ2 random	500	0.66	1.37	0.71
GPCR1 disparate	500	1.09	1.48	0.44	ENZ2 disparate	500	0.79	1.39	0.65
GPCR2 narrow-1	100	1.67	1.81	0.18	ENZ3 narrow-1	300	1.50	2.12	0.37
GPCR2 narrow-5	400	1.49	1.98	0.59	ENZ3 narrow-5	500	1.31	2.22	0.51
GPCR2 random	500	0.75	2.33	0.85	ENZ3 random	500	1.02	2.33	0.71
GPCR2 disparate	500	0.97	2.21	0.80	ENZ3 disparate	200	1.33	2.21	0.64
GPCR3 narrow-1	100	1.23	1.26	0.00	ENZ4 narrow-1	70	1.48	1.58	0.12
GPCR3 narrow-5	300	0.84	1.17	0.30	ENZ4 narrow-5	200	1.23	1.57	0.30
GPCR3 random	500	0.76	1.22	0.42	ENZ4 random	500	0.90	1.67	0.63
GPCR3 disparate	100	0.84	1.15	0.30	ENZ4 disparate	200	1.01	1.67	0.53
GPCR4 narrow-1	150	1.26	1.36	0.10	ENZ5 narrow-1	100	1.90	2.07	0.15
GPCR4 narrow-5	400	1.08	1.33	0.35	ENZ5 narrow-5	300	1.78	2.09	0.30
GPCR4 random	500	0.78	1.43	0.58	ENZ5 random	500	1.30	2.29	0.60
GPCR4 disparate	300	0.93	1.38	0.49	ENZ5 disparate	500	1.46	2.28	0.56
GPCR5 narrow-1	150	0.78	0.88	0.14	NHR1 narrow-1	100	1.38	1.60	0.31
GPCR5 narrow-5	500	0.83	0.98	0.19	NHR1 narrow-5	300	0.90	1.48	0.50
GPCR5 random	500	0.52	1.02	0.61	NHR1 random	500	0.77	1.67	0.65
GPCR5 disparate	80	0.68	0.98	0.43	NHR1 disparate	300	0.94	1.64	0.57
GPCR6 narrow-1	250	0.72	0.95	0.29	NHR2 narrow-1	200	1.48	1.62	0.25
GPCR6 narrow-5	500	0.66	0.95	0.36	NHR2 narrow-5	350	1.18	1.39	0.35
GPCR6 random	500	0.62	0.99	0.44	NHR2 random	500	0.79	1.42	0.58
GPCR6 disparate	300	0.69	0.96	0.38	NHR2 disparate	500	0.91	1.37	0.52
GPCR7 narrow-1	150	1.09	1.19	0.11	CA narrow-1	100	1.97	1.99	0.03
GPCR7 narrow-5	300	0.85	1.28	0.48	CA narrow-5	400	1.18	1.65	0.51
GPCR7 random	500	0.65	1.37	0.70	CA random	500	0.83	1.78	0.70
GPCR7 disparate	200	0.82	1.37	0.59	CA disparate	200	1.00	1.75	0.61

Inspection shows that this is due to the fact that the QSAR model for the narrow and random/disparate sets are fortuitously similar; the activity in both cases depends on the same AP descriptors involving aromatic atoms at a specific topological distance.

3. What Is a Good Extrapolation Measure? Figure 3A−D shows plots of ABSDIFF vs various extrapolation measures for the LOGD narrow-1 training set. A good discriminator is one where there is a large separation along the *x*-axis between poorly predicted (large ABSDIFF) and well-predicted molecules (small ABSDIFF). The fraction of descriptors in the training set (Figure 3A, Methods section 6, measure 1a) is clearly among the worst discriminators. If that value is even slightly lower than 1.0, the ABSDIFF is large. (It appears that the suggestion of Carhart et al.[7] to use a cutoff as high as 0.95 is justified; that is where the distribution of ABSDIFF changes.) Plots of the 10−90th percentile (Figure 3B, measure 1c) show some small segregation of poorly predicted molecules (high ABSDIFF) toward the left side of the plot. The segregation is not noticeably improved by changing the percentiles to 5−95th or 15−85th percentile (not shown). Similarity using 2, 3, or 4 standard deviations (measure 1b, not shown) gives very similar plots. Mean similarity to the first nearest neighbor (SIMILARITYNEAREST1) in the training set (Figure 3C, measure 2a) shows much more discrimination. The mean similarity to the 3- and 5-nearest neighbors (not shown) give similar plots, except that molecules in the training set can then have similarities < 1. Using only those descriptors in the training set to calculate the similarity makes no noticeable difference. Similarity to the centroid (measure 2b, not shown) is worse than the similarity to the 1-, 3-, or 5-nearest neighbors. Number of neighbors in the training set with 0.7 as the cutoff (Figure 3D, measure 3a) seems the best, with the highest ABSDIFFs in the test set clearly on the far left. Using the more complicated count of neighbors (measure 3b) produces a very similar plot (not shown) to Figure 3D. This is not surprising because the two types of counts are highly correlated. Note that the *x*-axis for these plots (NEIGHBORCOUNT) is really the number of neighbors in the training set +1. This is done so that the log(NEIGHBORCOUNT) can be taken if one wishes to compress the range of the plot, although that is not done here.

For the random training sets in Figure 3E−H, the ABSDIFF is generally lower than for the narrow training set, consistent with Figure 2. No extrapolation measure but NEIGHBORCOUNT (Figure 3H) shows a strong visual trend of ABSDIFF for the random set. There is perhaps a weak

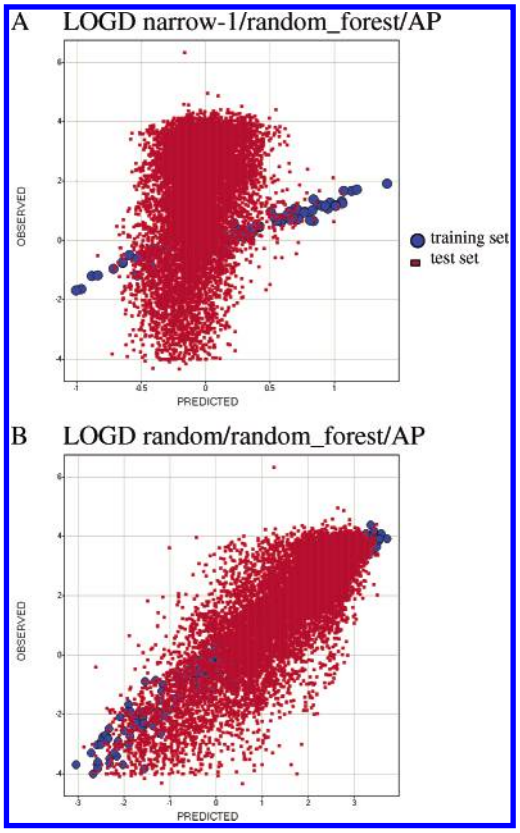


Figure 2. Observed activity vs predicted activity for random_forest QSAR model on the LOGD data set using the AP descriptor. Large blue circles are molecules in the training set and small red squares are molecules in the test set. These plots display data for one training set/test set split. A. narrow-1 training set. B. random training set.

trend for SIMILARITYNEAREST1 (Figure 3G). The behavior of the disparate training set (not shown) is very similar to that of the random training set except that the maximum number of neighbors is smaller, as expected.

We have two extrapolation measures that can discriminate better-predicted molecules from the others: NEIGHBORCOUNT and SIMILARITYNEAREST1. Note that in both cases it is the *spread* of ABSDIFFs that is of interest. For instance, if we look at Figure 3C, we see that at SIMILARITYNEAREST1 > 0.6, all of the ABSDIFFS have a low value. In contrast, at SIMILARITYNEAREST1 < 0.6, most of the ABSDIFFS are low, but some can reach very high values. Looked at another way, the spread of errors for the prediction increases as SIMILARITYNEAREST1 decreases. Thus we cannot say whether an individual molecule will have a low or high ABSDIFF, only that it is more probable to have a low ABSDIFF if SIMILARITYNEAREST1 is high.

4. Plots of rms-ABSDIFF vs NEIGHBORCOUNT. From here on we switch to plots of rms-ABSDIFF vs NEIGHBORCOUNT. rms-ABSDIFF is one quantitative measure of "error bar" for the prediction. Since the number of molecules falls quickly with NEIGHBORCOUNT, there are very few molecules with high NEIGHBORCOUNTs. Thus, to gather more uniform statistics, it is necessary to calculate the rms over a moving window of NEIGHBORCOUNT, and the width of the window needs to increase as NEIGHBORCOUNT increases. For x = NEIGHBORCOUNT, we calculate the rms-ABSDIFF for molecules in the range x-0.25x to x+0.25x. For example, for NEIGHBORCOUNT = 4, the rms-ABSDIFF includes molecules with NEIGHBORCOUNT in the range 3−5. (NEIGHBORCOUNT = 4 is the lowest value where there is averaging.) If there are <50 molecules in the window, the point is not plotted. Training sets are not shown in Figure 4 and subsequent figures because there are usually too few molecules in a window for them to be plotted.

Figure 4 shows these plots (on the left) for all the training sets for selected data sets. For the narrow-1 sets (red), the rms-ABSDIFF is high at NEIGHBORCOUNT = 1 but falls sharply by NEIGHBORCOUNT = 2 or 3. The rms-ABSDIFF at NEIGHBORCOUNT = 1 is usually near the scrambled rms. Again, this is consistent with poor predictions for molecules that have no neighbors in training set but much better predictions if there is at least one neighbor. The narrow-5 sets (blue) behave similarly. The exceptions again are ENZ3 and NHR2 for the reasons noted above.

For random (green) and disparate (black) sets at NEIGHBORCOUNT = 1, the rms-ABSDIFF is usually lower than the scrambled rms, again indicating fairly good predictions even for those molecules having no neighbors in the training set. For some data sets, like PKA and LOGD, rms-ABSDIFF seems to fall off with increasing NEIGHBORCOUNT indicating that as the number of neighbors grows, the predictions get even better. However, others like GPCR2 and NHR2 show little or no falloff.

Later, when we discuss why data sets differ, we will need to quantitate the falloff behavior of these plots. One simple way to do this is to find the ratio of rms-ABSDIFF at NEIGHBORCOUNT = 1 to the rms-ABSDIFF at some value of NEIGHBORCOUNT high enough that most of the plots have started to level off but low enough that most plots have good enough statistics. NEIGHBORCOUNT = 25 is a reasonable, if arbitrary, choice. We will call this the "falloff ratio 1/25". The scale of this number depends, of course, on the smoothing scheme we use to generate the curves, which is consistently applied among the data sets.

There are some cases where the curves do not fall off smoothly but contain "wiggles". These are explainable in retrospect. The largest wiggles in all our plots are in the narrow-5 and random curves for the PKA set (Figure 4B). They are primarily due to the fact that sometimes molecules in a series have a very different activity than would be predicted from a QSAR model trained on similar molecules. Thus the rms-ABSDIFF is unexpectedly high when the neighbors include members of that series. In the PKA case, and we might have expected in retrospect that PKA would be problematical in this regard, the phenomenon is due to one extreme outlier. Where most of the molecules in the particular series have a phenyl or nitrogen-containing heterocycle as one of the substituents (pK_a < 2), one has a 4-amino-pyridyl, which is basic (pK_a > 9). There are a few disparate training sets where the rms-ABSDIFF apparently increases with NEIGHBORCOUNT, opposite to expectation, for example in Figure 4H. This is due to a similar phenomenon. In these cases, the number of molecules with a high NEIGHBORCOUNT is small, and an appreciable fraction of them are very similar in structure but have very different observed activities, so they all cannot be fit well by the same model, hence the large rms-ABSDIFF. In some cases the activities are different because the molecules are stereoisomers, which cannot be distinguished by the AP descriptors. Wiggles tend to be more severe in smaller data

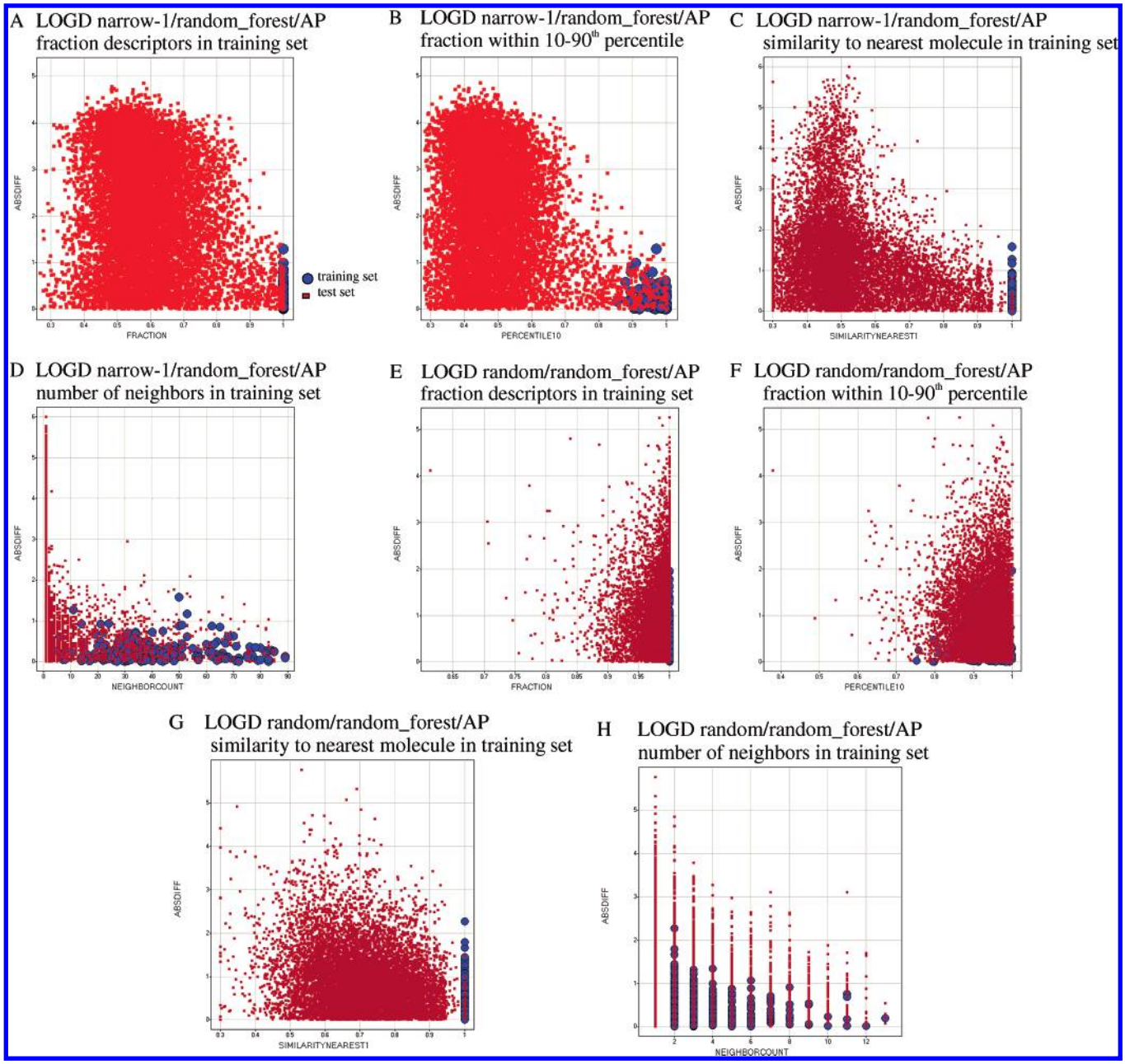


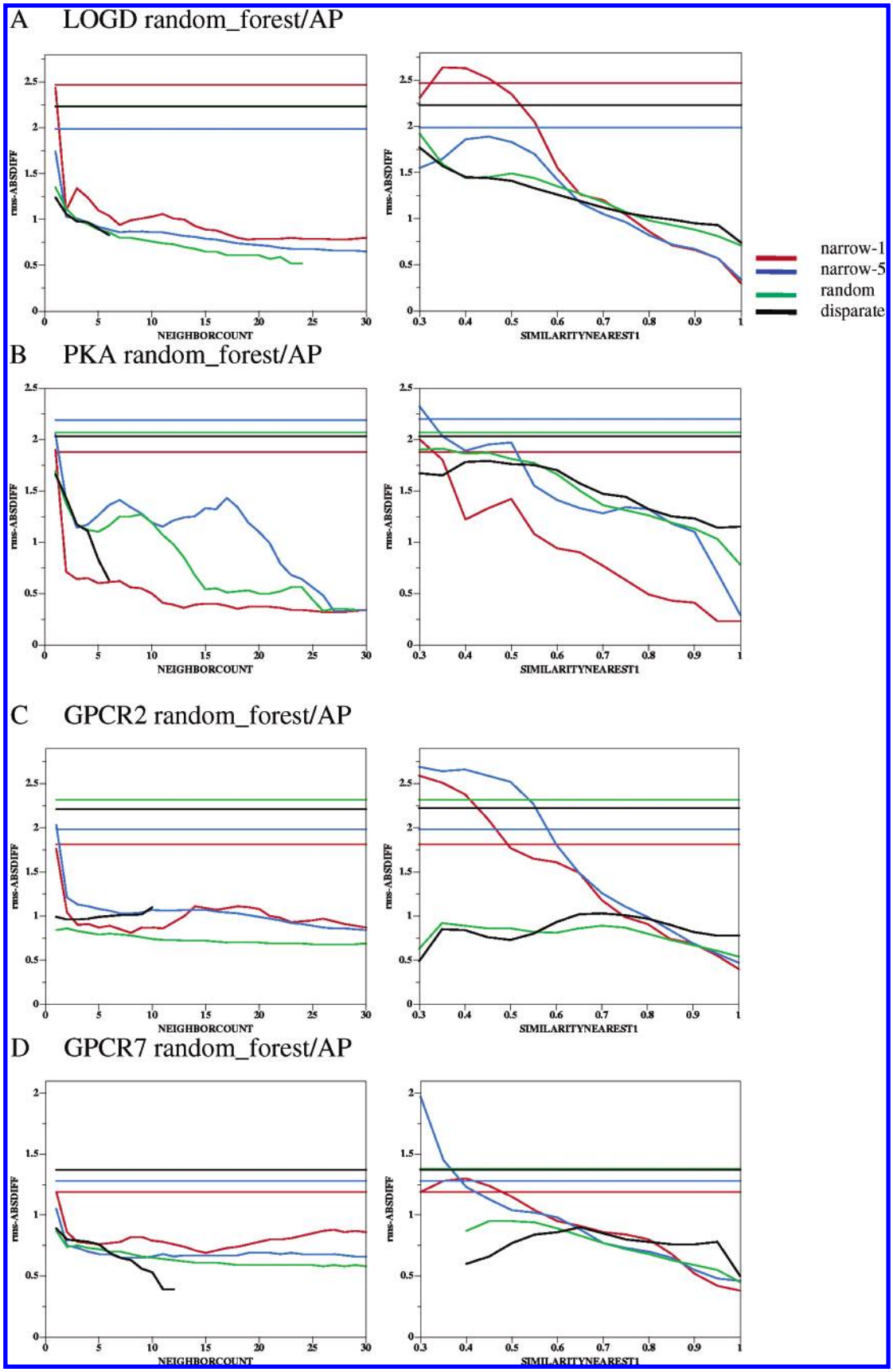
Figure 3. The absolute difference between observed and predicted activity (ABSDIFF) as a function of various measures of extrapolation for LOGD training sets. Large blue circles are molecules in the training set and small red squares are molecules in the test set. These plots present data for one training set/test set split. A. narrow-1 training set, fraction of the descriptors of the molecule to be predicted that are also present in the training set. B. narrow-1 training set, fraction of the descriptors of the molecule that fall within the 10−90th percentile of the values in the training set. Each descriptor has its own distribution. C. narrow-1 training set, similarity of the molecule to be predicted to the most similar molecule in the training set. The left boundary of the *x*-axis means ≤ 0.3. D. narrow-1 training set, the number of neighbors in the training set (Similarity ≥ 0.7 AP) for the molecule to be predicted. The *x*-axis (NEIGHBORCOUNT) is actually neighbors+1, so the log can be taken. E. random training set, fraction of the descriptors of the molecule to be predicted that are also present in the training set. F. random training set, fraction of the descriptors of the molecule that fall within the 10−90th percentile of the values in the training set. G. random training set, similarity of the molecule to be predicted to the most similar molecule in the training set. H. random training set, the number of neighbors in the training set.

sets where there are fewer molecules to dilute the effect of outliers.

5. Plots of rms-ABSDIFF vs SIMILARITYNEAREST1. For this type of plot, the windows are ±0.05 similarity unit. For instance, for SIMILARITYNEAREST1 = 0.6, the rms-ABSDIFF is calculated for those molecules with SIMILARITYNEAREST1 between 0.55 and 0.65.

Figure 4 shows these plots (on the right) for all the training sets for selected data sets. The narrow-1 (red) and narrow-5 (blue) training sets generally show a decrease in rms-ABSDIFF as SIMILARITYNEAREST1 increases. Some of the random (green) and disparate (black) training sets show a decrease, and some do not. Generally the data sets that show a large decrease in rms-ABSDIFF vs SIMILARITYNEAREST1 are those that show a large decrease in rms-ABSDIFF vs NEIGHBORCOUNT.

6. Does the Size of the Training Set Matter? We need to be sure that the behaviors noted above are intrinsic to the data set and not dependent on various indiosyncracies of the QSAR method, descriptors, etc. The first thing to check is


A LOGD random_forest/AP
B PKA random_forest/AP
C GPCR2 random_forest/AP
D GPCR7 random_forest/AP
rms-ABSDIFF
NEIGHBORCOUNT
SIMILARITYNEAREST1
narrow-1
narrow-5
random
disparate

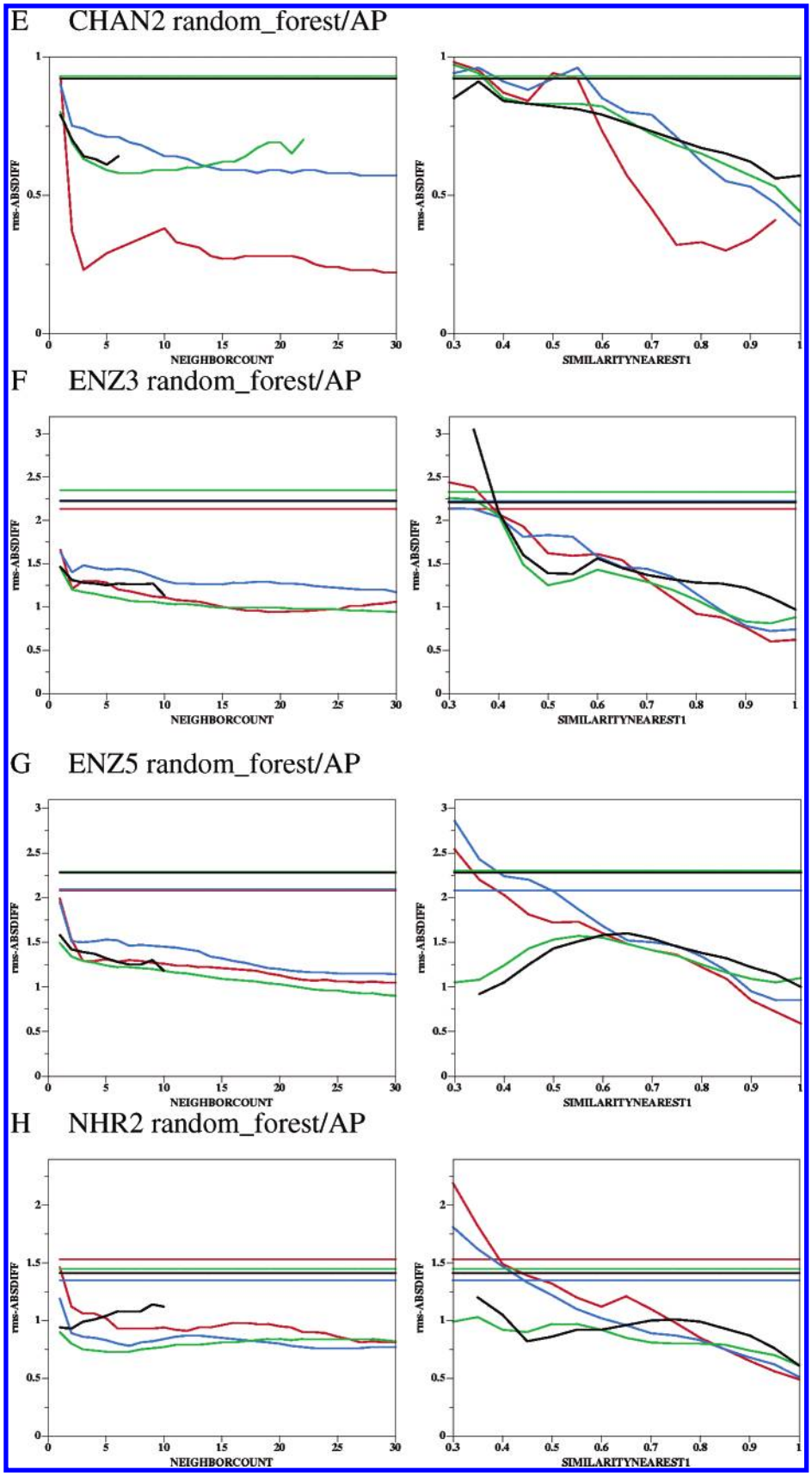


Figure 4. rms-ABSDIFF vs NEIGHBORCOUNT and rms-ABSDIFF vs SIMILARITYNEAREST1 for the random_forest method and AP descriptor for selected data sets: narrow-1 training set (red), narrow-5 training set (blue), random training set (green), disparate training set (black). These plots summarize data pooled from 10 training set/test set splits. The horizontal line of the corresponding color represents the value of "scrambled rms" which is equivalent to "no better than guessing". It is possible for a particular rms-ABSDIFF to be greater than the scrambled rms, especially for points representing a small number of molecules.

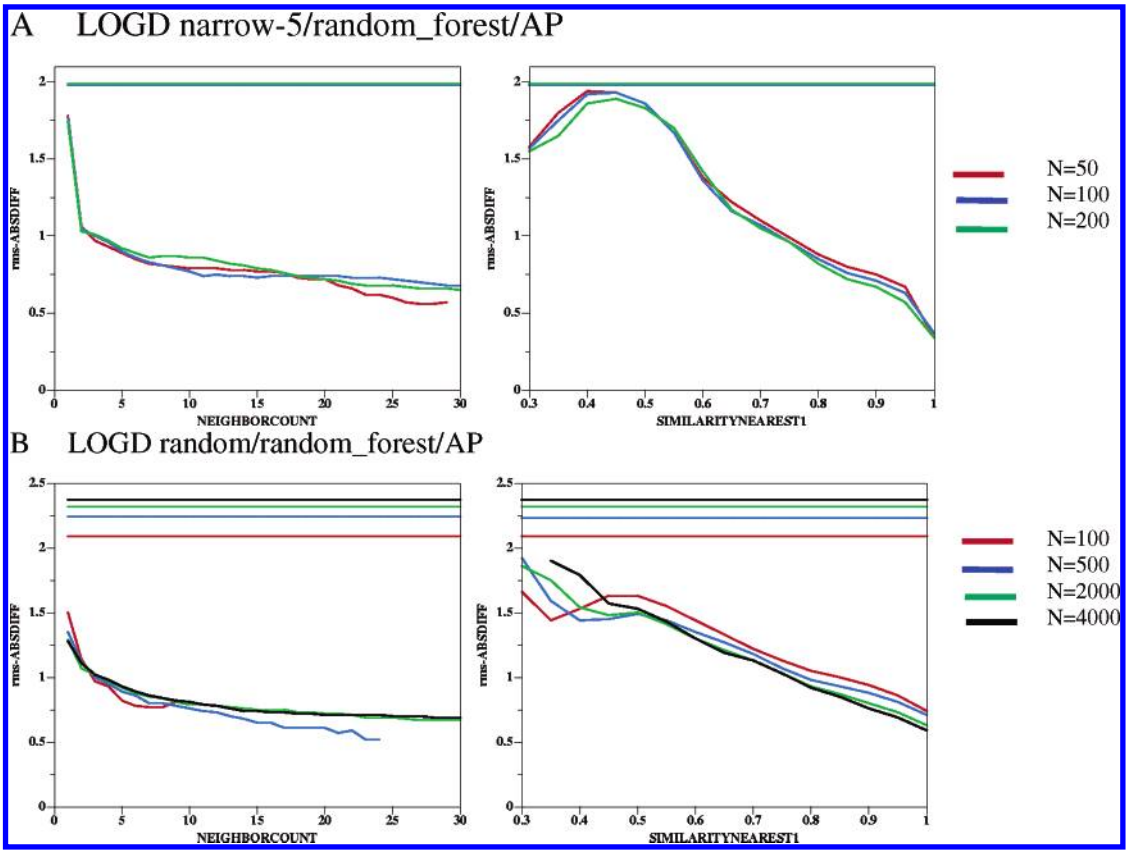


Figure 5. rms-ABSDIFF vs NEIGHBORCOUNT and rms-ABSDIFF vs SIMILARITYNEAREST1 for LOGD training sets of different sizes (random_forest method and AP descriptor). These plots summarize data pooled from 10 training set/test set splits. A. narrow-5 training sets: N = 50 (red), 100 (blue), 200 (green). B. random training sets: 100 (red), 500 (blue), 2000 (green), 4000 (black).

Table 3. QSAR Statistics for the LOGD Data Set, Using the random_forest Method and AP Descriptor, Where the Size of Training Set Is Varied

training set	Ntraining	rms error test set	rms error test set by scramble	R^2 for test set
LOGD narrow-5	50	1.71	1.99	0.23
	100	1.68	1.97	0.27
	250	1.61	2.06	0.30
LOGD random	100	1.42	2.09	0.50
	500	1.08	2.36	0.68
	2000	0.98	2.32	0.76
	4000	0.90	2.37	0.79
LOGD disparate	100	1.44	2.11	0.45
	500	1.08	2.35	0.68
	2000	0.95	2.31	0.76

the effect of the size of the training set. We will look at the LOGD data set as an example. Table 3 shows the statistics. The rms error in the test set goes down and the R2 goes up when the size of the training set goes up. This is expected: if the training set is larger, more test set molecules have a large number of neighbors in the training set, and the average error in prediction goes down. Figure 5 shows the effect of changing the size of the training set on the rms-ABSDIFF vs NEIGHBORCOUNT and SIMILARITYNEAREST1 plots. As the training sets get larger, the maximum number of neighbors a test molecule may have gets larger in proportion, as expected. However, the placement of the curves seems insensitive to the number in the training set, which varies by a factor of up to 40.

7. Does the QSAR Method Matter? As might be expected, the overall R^2 for prediction can vary between

Table 4. QSAR Statistics for the LOGD Training Set, Using Various QSAR Methods and the AP Descriptor

training set	QSAR method	rms error test set	rms error test set by scramble	R^2 for test set
LOGD narrow-1	random_forest	2.41	2.48	0.08
	trendvector	2.46	2.53	0.04
	mrlnet	2.48	2.59	0.02
	k-NN	2.06	2.32	0.17
	SVM	2.06	2.32	0.23
LOGD narrow-5	random_forest	1.61	2.06	0.30
	trendvector	1.72	1.97	0.23
	mrlnet	1.69	1.91	0.24
	k-NN	1.65	2.27	0.28
	SVM	1.48	2.08	0.46
LOGD random	random_forest	1.08	2.36	0.68
	trendvector	1.15	2.45	0.63
	mrlnet	1.15	2.44	0.66
	k-NN	1.17	2.44	0.62
	SVM	1.09	2.34	0.68
LOGD disparate	random_forest	1.08	2.35	0.68
	trendvector	1.18	2.54	0.62
	mrlnet	1.10	2.54	0.66
	k-NN	1.17	2.44	0.62
	SVM	1.06	2.36	0.69

methods. Over the 20 random training sets the mean R^2 is as follows: random_forest 0.58, SVM 0.56, mrlnet 0.51, k-NN 0.50, trendvector 0.44. On the other hand, for any specific training set, it is very hard to predict which method will do the best. Table 4 contains the summary statistics for the LOGD data set and AP descriptor for the five QSAR methods. Figure 6 shows the plots of rms-ABSDIFF vs NEIGHBORCOUNT and SIMILARITY1. There may be a small shift along the rms-ABSDIFF axis, which reflects the

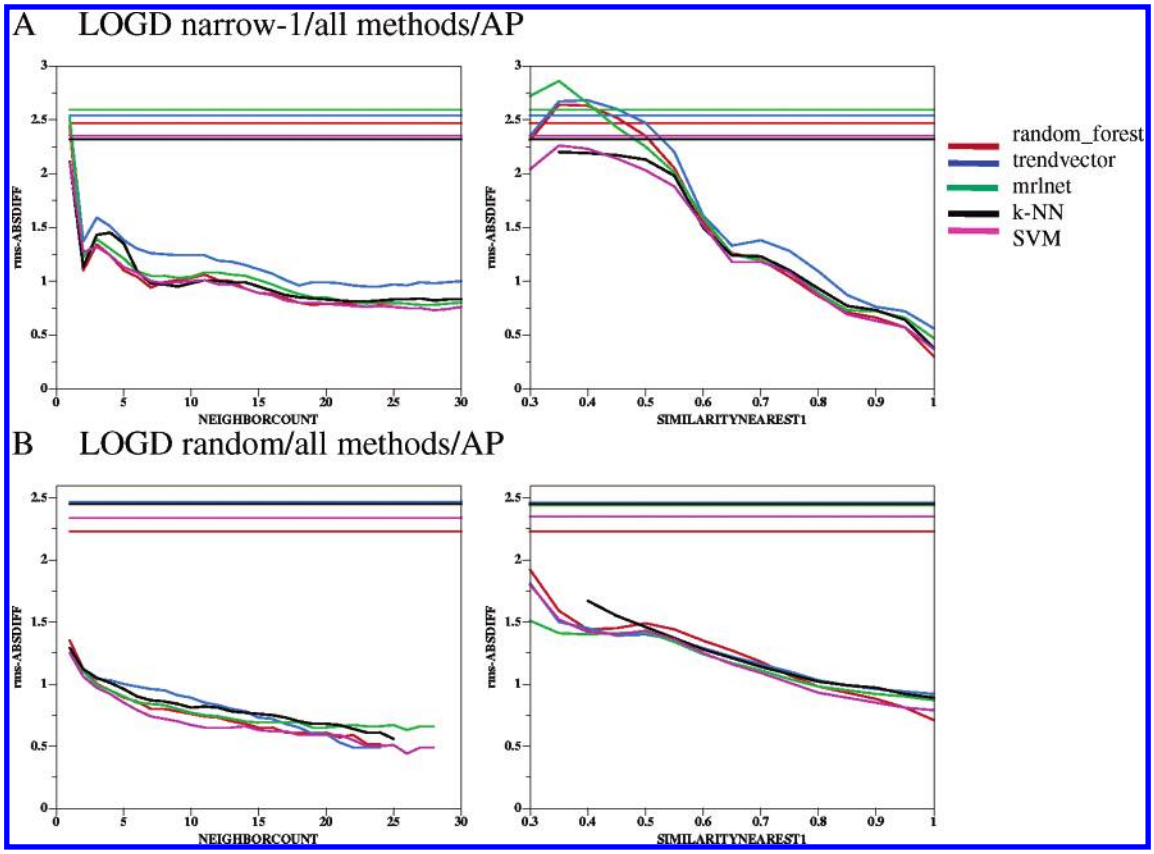


Figure 6. rms-ABSDIFF vs NEIGHBORCOUNT and rms-ABSDIFF vs SIMILARITYNEAREST1 for LOGD data sets using five QSAR methods and the AP descriptor: random_forest (red), trendvector (blue), mrlnet (green), k-NN (black), SVM (magenta). These plots summarize data pooled from 10 training set/test set splits. A. narrow-1 training set. B. random training set.

different abilities of the QSAR methods to predict, but shapes of the curves are roughly the same. This is significant given that each method has a very different philosophy of constructing a model.

8. Does the Descriptor Used To Calculate Similarity/Neighbors Have To Be the Same as the Descriptor Used for the QSAR? If similarity using all descriptors (regardless of whether they are in the training set or whether they are important for activity) provides good extrapolation measures, it might be possible to define the similarity independently of the descriptors used to generate the QSAR. One way to demonstrate the independence is to calculate NEIGHBORCOUNT and SIMILARITYNEAREST1 using AP but generate the QSAR model using a completely different descriptor. Table 5 contains the summary statistics for the LOGD data set and random_forest method for 5 different descriptors. There is some difference in the overall statistics between descriptors as might be expected. Plots for the rms-ABSDIFF vs NEIGHBORCOUNT and SIMILARITYNEAREST1 are in Figure 7. The curves for the other descriptors are qualitatively very similar to that for AP. An exception is that the WHIM descriptor seems especially poor for the random set, as indicated by the higher rms-ABSDIFF. We can conclude that the dependence of rms-ABSDIFF vs NEIGHBORCOUNT and SIMILARITYNEAREST1 does not rely on these extrapolation measures being calculated with the same descriptor as the QSAR model.

9. What Happens When We Change the Definition of Similarity/Neighborhood in the Extrapolation Measures? The converse test to the one above is to use the AP descriptor to generate the QSAR model and to use another descriptor to calculate NEIGHBORCOUNT and SIMILARITYNEAREST1. A popular definition of "neighborhood" in the literature is a cutoff of 0.85 using Daylight fingerprints and the Tanimoto definition of similarity.[2,5,6] Tanimoto and Dice similarities are monotonic, so it is easy to translate a Tanimoto cutoff into a Dice cutoff: 0.85 in Tanimoto corresponds to 0.93 in Dice. Figure 8 shows the rms-ABSDIFF vs NEIGHBORCOUNT and SIMILARITYNEAREST1 curves for the two different definitions of similarity/neighbors. Again, although

Table 5. QSAR Statistics for the LOGD Data Set, Using the random_forest Method and Various Descriptors

training set	QSAR descriptor	rms error test set	rms error test set by scramble	R^2 for test set
LOGD narrow-1	AP	2.41	2.48	0.08
	TT	2.32	2.35	0.02
	DF	2.39	2.40	0.00
	E-state	2.25	2.39	0.20
	WHIM	2.35	2.35	0.00
LOGD narrow-5	AP	1.61	2.06	0.30
	TT	1.80	2.00	0.15
	DF	1.86	1.98	0.06
	E-state	1.62	2.00	0.35
	WHIM	1.85	1.99	0.06
LOGD random	AP	1.08	2.36	0.68
	TT	1.22	2.24	0.61
	DF	1.31	2.17	0.56
	E-state	1.11	2.28	0.68
	WHIM	1.72	2.05	0.18
LOGD disparate	AP	1.08	2.35	0.68
	TT	1.20	2.24	0.62
	DF	1.31	2.15	0.56
	E-state	1.11	2.29	0.68
	WHIM	1.75	2.04	0.14

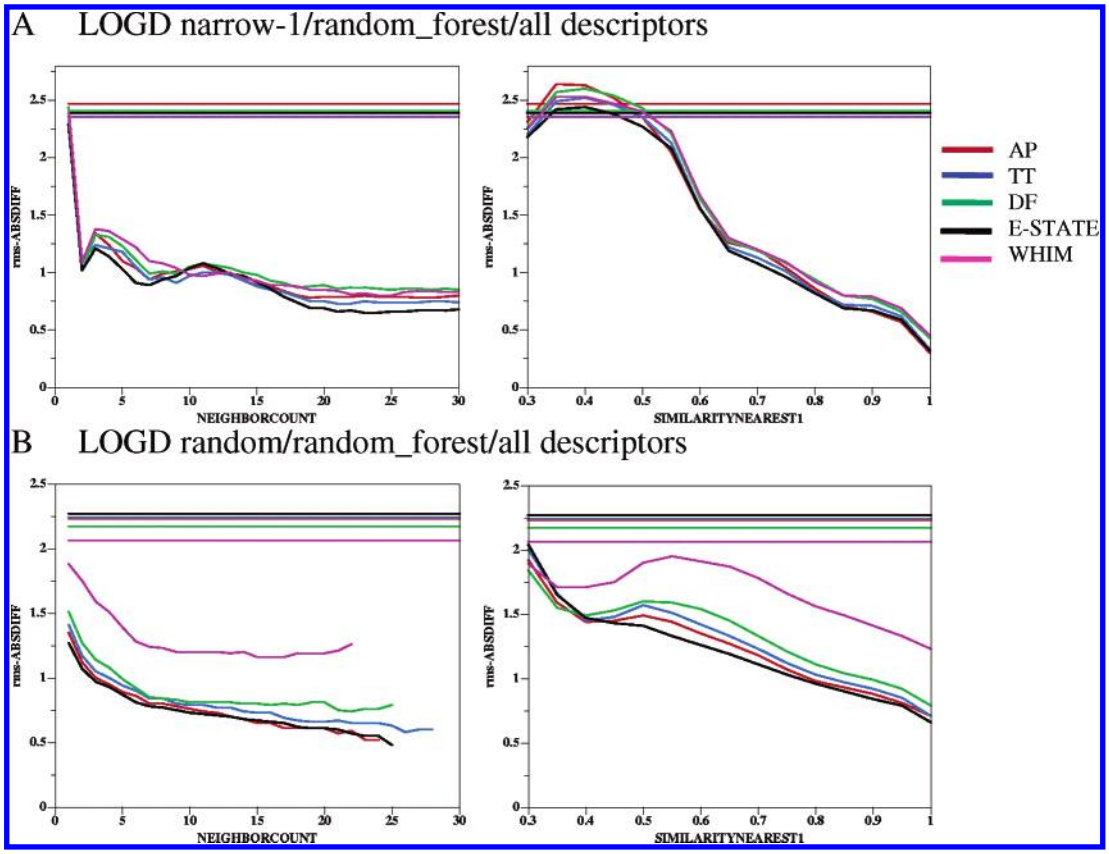


Figure 7. rms-ABSDIFF vs NEIGHBORCOUNT and rms-ABSDIFF vs SIMILARITYNEAREST1 for LOGD training sets using the random_forest method with various descriptors. In all cases the AP descriptor was used to calculate NEIGHBORCOUNT and SIMILARITYNEAREST1. These plots summarize data pooled from 10 training set/test set splits. AP (red), TT (blue), DF (green), E-STATE (black), WHIM (magenta). A. narrow-1 training set. B. random training set.

Table 6. Falloff Ratios for the Random Training Sets Using Various QSAR Methods and the AP Descriptor

data set	falloff ratio 1/25 random_forest	falloff ratio 1/25 trendvector	falloff ratio 1/25 mrlnet	falloff ratio 1/25 k-NN	falloff ratio 1/25 SVM	falloff ratio 1/25 mean	Patterson ratio
LOGD	2.53	2.63	1.86	2.11	2.45	2.32	2.28
LOGSOL	1.40	1.08	1.14	1.34	1.40	1.27	1.70
PKA	3.80	1.36	3.55	4.71	3.80	3.44	2.60
GPCR1	1.32	0.87	1.2	1.41	1.25	1.21	1.35
GPCR2	1.23	0.81	1.16	1.58	1.36	1.23	2.15
GPCR3	1.40	1.28	1.34	1.58	1.48	1.42	1.58
GPCR4	1.01	0.73	0.97	1.25	1.06	1.00	1.38
GPCR5	1.98	1.44	1.86	2.05	1.89	1.84	1.62
GPCR7	1.68	0.99	1.34	1.65	1.45	1.42	1.69
GPCR6	1.05	0.94	1.08	1.31	1.10	1.10	1.24
CHAN1	1.10	0.99	0.91	1.01	1.05	1.01	1.48
CHAN2	1.86	1.45	1.16	1.38	1.37	1.44	1.58
ENZ1	1.19	0.97	1.26	1.45	1.27	1.23	1.56
ENZ2	1.45	1.26	1.52	1.79	1.56	1.52	1.87
ENZ3	1.49	1.11	1.37	1.66	1.50	1.43	2.14
ENZ4	2.28	1.38	2.04	1.77	2.26	1.95	2.00
ENZ5	1.55	1.21	1.42	1.72	1.56	1.49	1.67
NHR1	1.55	0.99	1.48	1.74	1.76	1.50	1.77
NHR2	1.07	0.78	0.96	1.02	1.16	1.00	1.43
CA	1.75	1.09	1.68	1.99	1.68	1.64	2.13
mean	1.63 ± 0.65	1.17 ± 0.41	1.46 ± 0.58	1.73 ± 0.76	1.62 ± 0.63		

the numbers of molecules at any given range of NEIGHBORCOUNT or SIMILARITYNEAREST1 can change drastically, the overall shape of the curves does not. Again we see that the descriptors for calculating the extrapolation measures and for generating the QSAR do not have to be the same.

10. What Happens When Noise Is Added to the Activities? Any QSAR method is vulnerable to noise in the data. To monitor the effect of noise, we introduced noise artificially by adding random numbers to the observed activities uniformly in the range of ± 0.5, ± 1.0, ± 2.0, and ± 4.0. The curves for LOGD (random_forest method, AP descriptor, AP for extrapolation measures) are shown in Figure 9. At higher noise levels, the rms-ABSDIFF rises, not surprisingly, with more of a rise at high NEIGHBORCOUNT or SIMILARITYNEAREST1. This tends to flatten the curves. Also wiggles become more prominent.

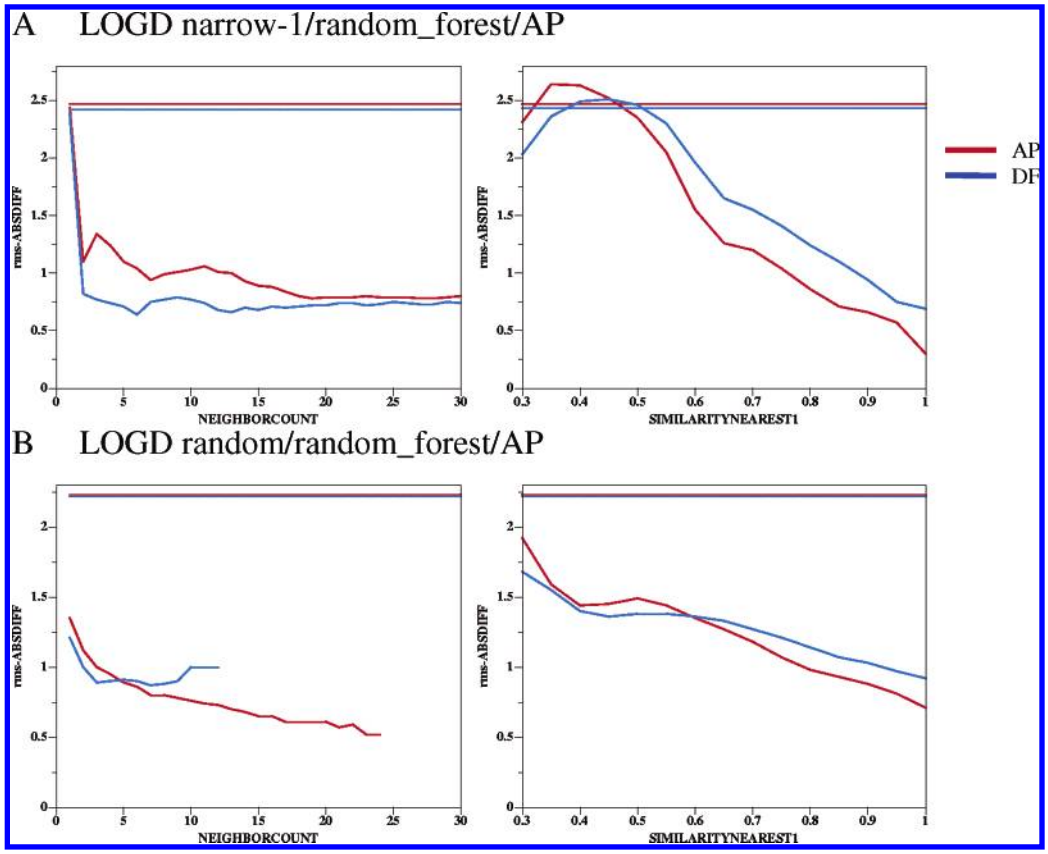


Figure 8. rms-ABSDIFF vs NEIGHBORCOUNT and rms-ABSDIFF vs SIMILARITYNEAREST1 for LOGD training sets using the random_forest method with the AP descriptor. For NEIGHBORCOUNT, neighbors were defined two ways: AP descriptors cutoff 0.7 Dice (red) and DF fingerprints cutoff 0.93 Dice, which is equivalent to 0.85 Tanimoto (blue). SIMILARITYNEAREST1 defined by the Dice similarity using AP descriptors (red) or DF fingerprints (blue). These plots summarize data pooled from 10 training set/test set splits.

Table 7. Falloff Ratio and Patterson Ratio for the LOGD Random Training Set, Using the random_forest Method and AP Descriptor, as Artificial Noise Is Added to the Activity

range of random number added to activity	falloff ratio 1/25 random_forest	Patterson ratio
0	2.53	2.28
±0.5	2.40	2.05
±1.0	2.07	1.79
±2.0	1.27	1.50
±4.0	0.98	1.28

11. Why Does the Falloff for rms-ABSDIFF vs NEIGHBORCOUNT Vary from One Data Set to Another? That there is some dependence of the prediction accuracy as a function of the number of neighbors in the training set is not necessarily expected for random and disparate training sets and deserves further study. Table 6 lists the falloff ratios for all random training sets and all QSAR methods. As a rule, the data sets with higher falloff ratios will have higher falloff ratios whatever the QSAR method. The QSAR method does seem to shift the falloff ratios of the entire set, as indicated by the row labeled "mean" in Table 6, with k-NN giving the highest falloffs overall, and trendvector the lowest. We might expect k-NN to give the highest falloffs because that method uses explicit similarity to training set molecules to make predictions, but it is not much different from random_forest, which does not. We can create a consensus falloff ratio (Table 6) for the data sets by taking the mean falloff ratios over all QSAR methods.

Having established above that the falloff for random training sets is not strongly dependent on the QSAR method or descriptor or the size of the training set, we need to look at intrinsic properties of the data sets. After examining a number of properties (size of the data set, diversity of the data set, distribution of activities, the extent to which the more active classes of molecules are overrepresented, the number of indefinite values, etc.) we found the best correlation of the falloff ratio is with the Patterson ratio (shown in Table 6). Although all data sets have a Patterson ratio > 1, indicating some neighborhood behavior, they do vary. Figure 10 shows the Patterson plots for NHR2 and PKA, one of the lowest and highest Patterson ratios, respectively. Figure 11 shows the plot of consensus falloff ratio vs Patterson ratio. The trend is clearly that the higher the Patterson ratio, the stronger the falloff behavior.

Adding artificial noise to the observations also decreases the falloff ratio (as seen in Figure 10B) and the Patterson ratio. Values for LOGD (random_forest method, AP descriptor) are in Table 7.

DISCUSSION

As far as we know, ours is one of the very few attempts to empirically quantitate the degradation of prediction accuracy of a QSAR model as the molecule to be predicted departs from the training set. We have shown that plots of rms-ABSDIFF vs NEIGHBORCOUNT and SIMILARITYNEAREST1 derived from cross-validation can be used to estimate the "error bars" of prediction. A single number, e.g. the cross-validated root-mean-square error over the entire set, cannot convey this information. The qualitative dependence of rms-ABSDIFF vs these extrapolation measures is

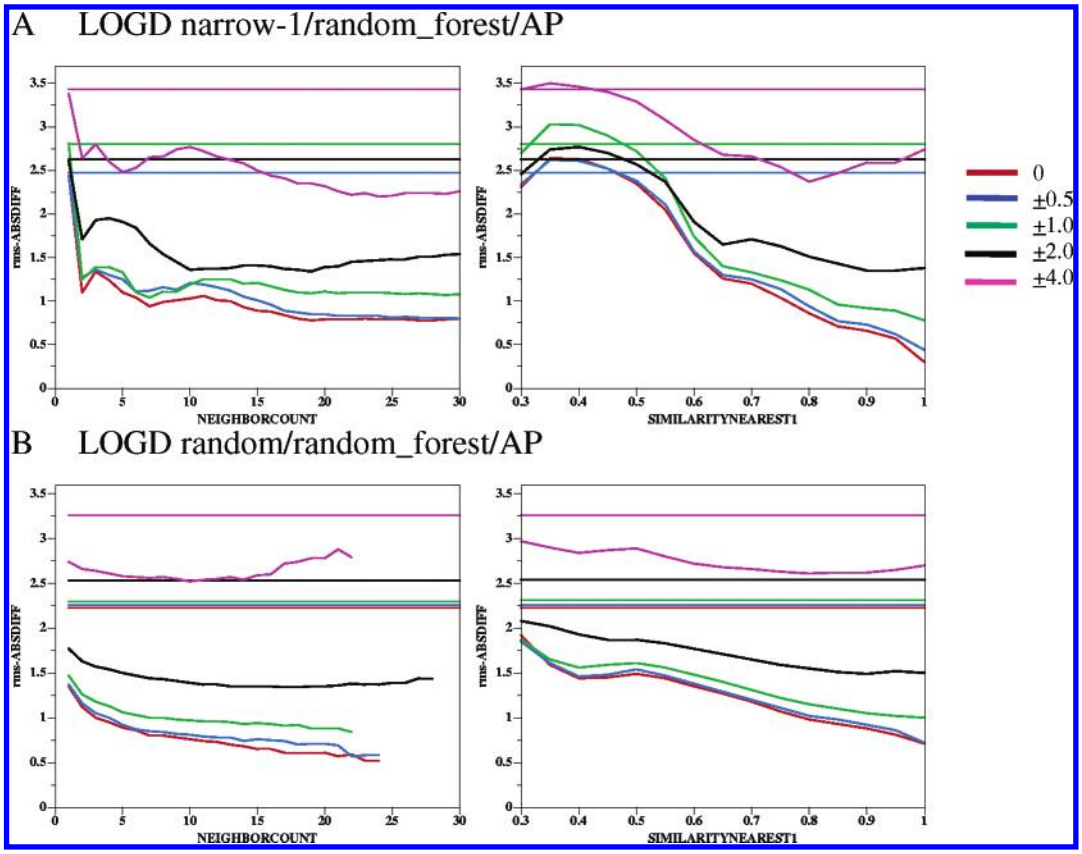


Figure 9. rms-ABSDIFF vs NEIGHBORCOUNT and rms-ABSDIFF vs SIMILARITYNEAREST1 for LOGD training sets using the random forest method with the AP descriptor. Noise is artificially added to the activity data by adding a random number in the range 0 (red), ±0.5 (blue), ±1.0 (green), ±2.0 (black), ±4.0(magenta). A narrow-1 training set. B. random training set.

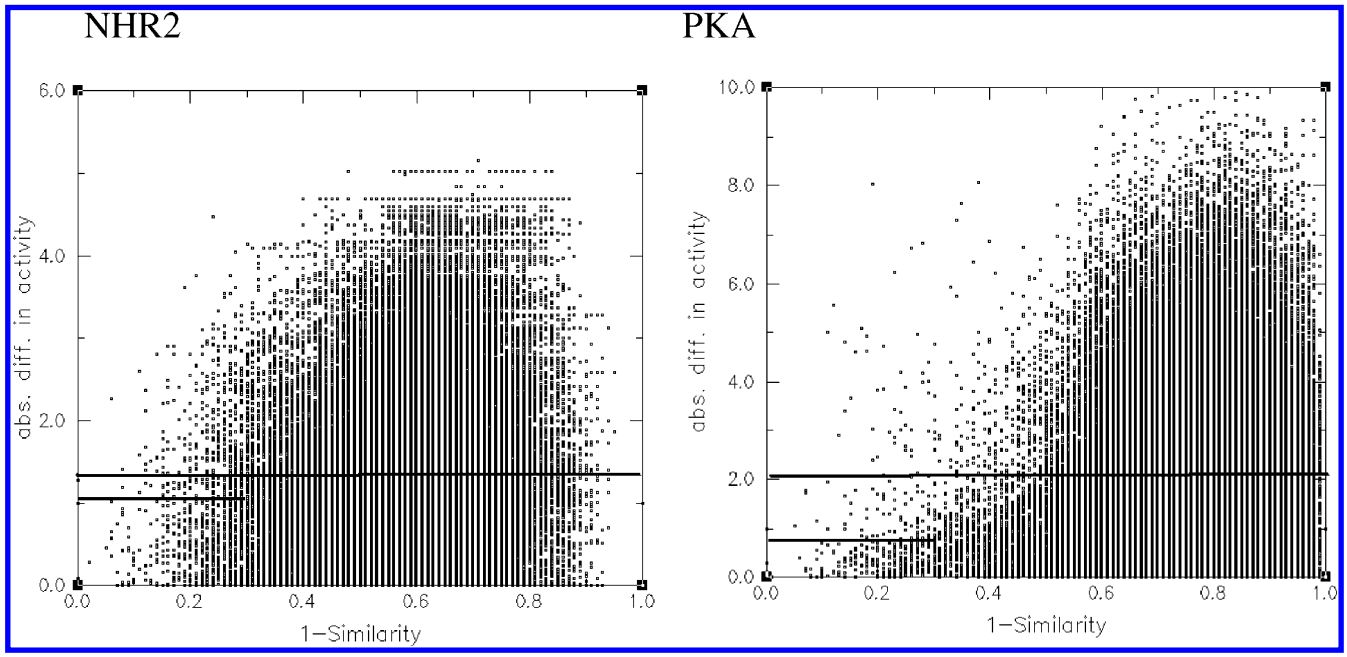


Figure 10. Patterson plots for NHR2 and PKA. The long horizontal bar represents the mean absolute difference in activity for all pairs in the data set. The short horizontal bar represents the mean absolute differences in observed activity for all pairs of molecules whose similarities ≥ 0.7 for the AP descriptor (1-Similarity ≤ 0.3). The ratio between these values is the Patterson ratio.

not dependent on the QSAR method or descriptor used to generate the model. NEIGHBORCOUNT and SIMILARITYNEAREST1 can be defined independently of the descriptors used in the QSAR model. This is useful because some QSAR methods (e.g. random_forest) can use any mixture of substructure descriptors and incommensurate whole-molecule descriptors, and it is not obvious how to define a similarity in that space. Another reason is that the similarities are reusable from one model to another.

Since NEIGHBORCOUNT and SIMILARITYNEAREST1 have proved to be good discriminators, then Figure 1C is a useful way of thinking about QSAR, in the sense that the "domain" of a training set is defined by similarity neighborhoods around individual molecules. This definition

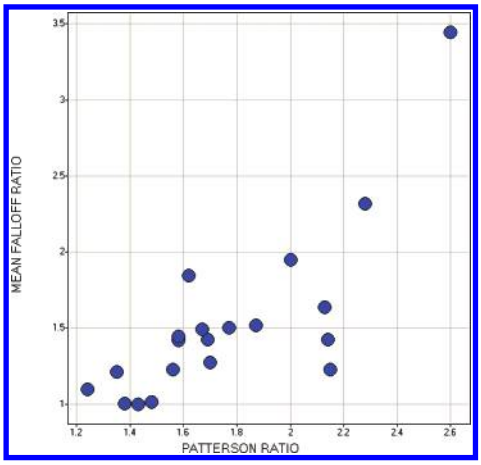


Figure 11. Mean falloff ratio (1/25) for 20 random training sets vs the Patterson ratio.

is intuitively appealing, easy to visualize, and easy to calculate whether the molecule to be predicted has descriptors not included in the training set. This is in contrast to, for instance, defining a domain as an ellipsoid in some multidimensional space of latent variables, which is very difficult for anyone to visualize. Here we used the Dice definition of similarity with the AP descriptor as the standard, but other definitions will probably prove useful. One important implication of having a definition of domain based on individual training set compounds is that to get the full use of a QSAR model, one must also keep the connection tables (or at least the descriptors) of the training set.

In regard to the plots, measures of "error bar for prediction" other than rms-ABSDIFF are certainly conceivable. The root-mean-square is a standard in statistics and is consistent with how errors are conventionally expressed in QSAR, but two reasonable alternatives are median (50th percentile) ABSDIFF and 95th percentile ABSDIFF. The median is less sensitive to outliers than either root-mean-square or 95th percentile and produces curves with much smaller wiggles. On the other hand, the 95th percentile may be closer to what chemists expect when they say "what error am I fairly sure to be below?". In this paper we have dealt with numerical activities only. Many QSAR models are based on two or more classifications (e.g., "thrombin inhibitor" vs "trypsin inhibitor" vs "elastase inhibitor"), and rms-ABSDIFF does not apply. An analogous error measure for classification problems might be "percent incorrectly classified".

Another issue is whether we have the best curve smoothing scheme. Here we implemented a simple moving window scheme where all molecules within the window count equally. More sophisticated schemes (splines, LOESS, etc.) are possible. In particular, a pertinent smoothing issue is that many curves have large wiggles due to outliers. One can imagine a scheme where such outliers were downweighted.

It was somewhat surprising to us that plots of rms-ABSDIFF vs NEIGHBORCOUNT and SIMILARITYNEAREST1 seem to show discrimination whether the training sets are narrow or diverse. For narrow training sets, we expected the quality of the predictions to vary sharply with similarity to members the training set, and we almost always observe that. We did not necessarily expect there to be any discrimination in diverse (i.e. random or disparate) training sets. We thought that, given a large and diverse training set, and a QSAR method that fits to individual descriptors rather than whole molecules (i.e. all methods but k-NN), the model would have seen enough chemical space to be able to predict truly novel molecules as accurately as familiar ones. That is apparently not the case. For many random and disparate training sets we see a smaller (compared to the narrow sets), but definite, falloff of rms-ABSDIFF vs NEIGHBORCOUNT or SIMILARITYNEAREST1. We first suspected an artifact such as the following: a molecule appears well-predicted at high NEIGHBORCOUNT because it belongs to a chemical class that is over-represented in the training set. However, the fact that the trend is strong for some disparate training sets, where such overrepresentation has largely been eliminated, argues against such an artifact. We have noted that the falloff ratio for random sets is correlated with the neighborhood behavior as encapsulated in the Patterson plot. Is this a causal link, i.e., if similar molecules have similar activity, does that automatically imply that activities of new molecules would be better predicted if similar ones are in the training set, or are both phenomena just independently reflecting the amount of noise in the data? It is hard to say at present.

Because various departures from ideality (the addition of artificial noise, the presence of strong outliers, smaller data sets, poor Patterson behavior, poor fitting by the QSAR method, etc.) tend to decrease the falloff ratio and/or introduce irregularities into the curves, a reasonable working hypothesis that a smooth curve with at least a moderate falloff represents the baseline condition for diverse sets. If this hypothesis is true, it would require some explanation. Why should there be a steady decrease of prediction accuracy as the molecule to be predicted becomes less similar to molecules in a diverse training set? One speculation is that QSAR methods, even when they do not explicitly include the notion of similarity between molecules, are implicitly taking such information into account. For instance partial-least squares can be reformulated as starting from a matrix of molecule distances or covariances,[20] which would be correlated with similarity. Another idea is that biological activities are not completely determined by the presence or absence of chemical substructures independent of their surroundings but depend somewhat on the context of the whole molecule. Therefore QSAR models can never extrapolate with complete accuracy to a class of molecules not seen by the model. A third possibility is that in real data sets, where molecules come in series, groups conferring activity are not statistically separable from the series in which they are found, so no QSAR method can ever develop a model free of "series bias". The first possibility says something about our QSAR methods, the second about biological activities, and the third about the way we construct our data sets. However, the real reason does not matter, as long as similarity-based extrapolation measures provide a reliable empirical way of discriminating good from poor predictions for a given model.

Finally, we make a proposal on how rms-ABSDIFF vs NEIGHBORCOUNT or SIMILARITYNEAREST1 plots can be used in practice. Given a preexisting QSAR model built by method Q with descriptors D from a data set T, we would like to construct plots to represent the situation where a large diverse sample of arbitrary drug-like molecules is predicted against the model derived from all molecules in T. (It is

assumed that the QSAR model has already been validated to the satisfaction of the user and that the user has chosen a good Q and D.) Our proposal is that, using the recipe in the Methods section, we can approximate the ideal plots by cross-validation using models derived from subsets of T:

1. Cross-validate predictions by repeated splits on a data set T using method Q and descriptor D. The fraction F of the data set will be in the training set. Since we wish to reflect the diversity inherent in T, the splits will be done by random selection from T. Calculate similarities using a suitable descriptor D′ (and a reasonable cutoff in D′ for defining neighborhoods). D′ does not have to be the same as D. If the set is large, the results will not be sensitive to F. Calculate the scrambled rms for the whole set. Generate the curves rms-ABSDIFF (or 95th percentile, etc., if preferred) vs NEIGHBORCOUNT and rms-ABSDIFF vs SIMILARITYNEAREST1 for the test set compounds. Step 1 is done once for each model.

2. For each molecule M to be predicted, one calculates the similarity (using D′) to all molecules in T. One then has the NEIGHBORCOUNT and SIMILARITYNEAREST1 values for M and can read its likely rms-ABSDIFF off one or both curves. The most similar molecules in the training set to M can be displayed to the user. One should always compare the estimated rms-ABSDIFF to the scrambled rms, which represents "no better than guessing". It should be reemphasized that rms-ABSDIFF, or any alternative measure of error bar, in these plots represent an error averaged over multiple molecules (and multiple predictions of the same molecule). Thus, the rms-ABSDIFF for M is the "rms error in prediction for a typical molecule with the same NEIGHBORCOUNT (or SIMILARITYNEAREST1) as M". It is not possible to find the true error for M until its activity is actually measured.

For this paper we have the luxury of choosing data sets that are large and diverse, so our cross-validation subsets are likely representative of the diversity in T for a range of F, and we can gather good statistics for the test set over a reasonable range of NEIGHBORCOUNT. Real data sets with less than ideal properties will raise various issues that may make our proposal more difficult to implement. First, large amounts of noise or outliers in the data will lessen the probability that smooth discriminating curves can be generated. Second, if T is small results may vary with F. A reasonable compromise needs to be found between leaving out few molecules (more representative models and fewer examples of prediction but larger number of neighbors in the training set) or leaving out many (more examples and less representative models but fewer neighbors in the training set). It may be advisable to cross-validate at both small and large values of F to build up the statistics at both low and high values of NEIGHBORCOUNT.

Third, if T is not very diverse, it may be difficult to generate a reliable rms-ABSDIFF where NEIGHBORCOUNT = 1 since there are few or no examples of molecules not similar to the others. One possible approximation is to use the scrambled rms, i.e., the "no better than guessing" value because in most of our narrow training set examples the rms-ABSDIFF at NEIGHBORCOUNT = 1 approaches this number.

ACKNOWLEDGMENT

Subhas Chakravorty was instrumental in adding Daylight fingerprints and CERIUS2 descriptors (including E-state) to the QSAR suite. Matt Walker implemented the WHIM descriptors. Andy Liaw wrote the random_forest C-code based on Fortran code originally written by Leo Breiman and Adele Cutler. Gene Fluder wrote turbosim, a rapid calculator of global similarity for large sets of molecules. A number of Molecular Systems Applications members pointed out useful data sets. A large number of Merck biologists, over many years, generated the data we used in this paper. J. Chris Culberson wrote a facility to allow easy access to the data. Andy Liaw, Chris Tong, and Vladimir Svetnik of the Biometrics Department provided useful discussions.

REFERENCES AND NOTES

(1) Livingston, D. J. The characterization of chemical structures using molecular properties. A survey. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 195−209.

(2) Bajorath, J. Selected concepts and investigations in compound classification, molecular descriptor analysis, and virtual screening. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 233−245.

(3) Hawkins, D. M. The problem of overfitting. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1−12.

(4) Montgomery, D. C.; Peck, E. A.; Vining, G. G. *Introduction to Linear Regression Analysis*; John Wiley & Sons: 2001.

(5) Martin, Y. C.; Kofron, J. L.; Traphagen, L. M. Do structurally similar molecules have similar biological activity? *J. Med. Chem.* **2002**, *45*, 4350−4358.

(6) Nikolova, N.; Jaworska, J. Approaches to measure chemical similarity−a review. *QSAR Comb. Sci.* **2003**, *22*, 1006−1026.

(7) Carhart, R. E.; Smith, D. H.; Ventkataraghavan, R. Atom pairs as molecular features in structure−activity studies: definition and application. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 64−73.

(8) Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological torsions: a new molecular descriptor for SAR applications. comparison with other descriptors. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 82−85.

(9) Kier, L. B.; Hall, L. H. *Molecular Connectivity in Structure−Activity Analysis*; John Wiley & Sons: New York, 1986.

(10) Todeschini, R.; Lasagni, M.; Marengo, E. New molecular descriptors for 2D and 3D structures. Theory. *J. Chemometrics* **1994**, *8*, 263−272.

(11) Gasteiger, J.; Rudolf, C.; Sadowski, J. Automatic generation of 3D-atomic coordinates for organic molecules. *Tetrahedron Comput. Methodol.* **1990**, *3*, 537−547.

(12) Brieman, L. Random forests. *Machine Learning* **2001**, *45*, 5−32.

(13) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random forest: a classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947−1958.

(14) Sheridan, R. P.; Nachbar, R. B.; Bush, B. L. Extending the trend vector: the trend matrix and sample-based partial least squares. *J. Comput.-Aided Mol. Des.* **1994**, *8*, 323−340.

(15) Vapnik, V. N. *The Nature of Statistical Learning Theory*; Springer-Verlag: 1995.

(16) Joachims, T. *Making large-Scale SVM Learning Practical. Advances in Kernel Methods − Support Vector Learning*; Schölkopf, B., Burges, C., Smola, A. Eds.; MIT-Press: 1999.

(17) Butina, D. Unsupervised database clustering based on Daylight's fingerprint and Tanimoto dissimilarity: a fast and automated way to cluster small and large data sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 747−750.

(18) Sheridan, R. P. The centroid approximation for mixtures: calculating similarity and deriving structure−activity relationships. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1456−1469.

(19) Patterson, D. E.; Cramer, R. D.; Ferguson, A. M.; Clark, R. D.; Weinberger, L. E. Neighborhood behavior: a useful concept for validation of 'molecular diversity' descriptors. *J. Med. Chem.* **1996**, *39*, 3049−3059.

(20) Bush, B. L.; Nachbar, R. B., Jr. Sample-distance partial least squares: PLS optimized for many variables, with application to CoMFA. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 587−619.

CI049782W