JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

Comparison of Random Forest and Pipeline Pilot Naïve Bayes in Prospective QSAR Predictions

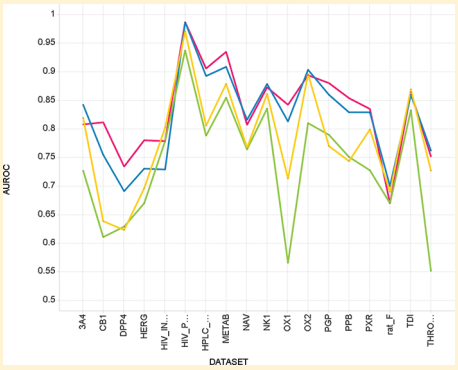
Bin Chen,[†] Robert P. Sheridan,[‡,*] Viktor Hornak,[‡] and Johannes H. Voigt[‡]

[†]School of Informatics and Computing, Indiana University at Bloomington, Bloomington, Indiana 47405, United States

[‡]Chemistry Modeling and Informatics Department, Merck Research Laboratories, Rahway, New Jersey 07065, United States

Supporting Information

ABSTRACT: Random forest is currently considered one of the best QSAR methods available in terms of accuracy of prediction. However, it is computationally intensive. Naïve Bayes is a simple, robust classification method. The Laplacian-modified Naïve Bayes implementation is the preferred QSAR method in the widely used commercial chemoinformatics platform Pipeline Pilot. We made a comparison of the ability of Pipeline Pilot Naïve Bayes (PLPNB) and random forest to make accurate predictions on 18 large, diverse in-house QSAR data sets. These include on-target and ADME-related activities. These data sets were set up as classification problems with either binary or multicategory activities. We used a time-split method of dividing training and test sets, as we feel this is a realistic way of simulating prospective prediction. PLPNB is computationally efficient. However, random forest predictions are at least as good and in many cases significantly better than those of PLPNB on our data sets. PLPNB performs better with ECFP4 and ECFP6 descriptors, which are native to Pipeline Pilot, and more poorly with other descriptors we tried.



INTRODUCTION

In QSAR, a statistical model is built that relates a biological activity to chemical structure as represented by descriptors. That model is then used to predict the activities of new molecules. There are many QSAR methods in the literature: multiple linear regression,[1−3] PLS,[4] Naïve Bayes,[5−7] neural networks,[8−10] random forest,[11,12] SVM,[13,14] Gaussian Processes,[15] etc. They vary in computational cost, accuracy of prediction, type of descriptors they can handle, sensitivity to noise, whether the models they produce are interpretable, etc. At the time of writing two methods, random forest and SVM, seem to be something of a "gold standard" in terms of prediction accuracy.[16,17]

Random forest (RF) is an ensemble recursive partitioning method where each recursive partitioning "tree" is generated from a bootstrapped sample of compounds, and random subset of descriptors is used at each branching of each tree. The trees are not pruned. Two useful features of RF are that one does not have to do descriptor selection to obtain good results and that predictions appear robust to changes in the adjustable parameters. (SVM in contrast requires the problem-specific optimization of at least one adjustable parameter.) Various implementations of RF can handle regression problems or classification problems. The downside of RF is that it is fairly expensive in memory and CPU time.

Naïve Bayes (NB) is a classifier method that relates the presence of certain features to the probability of a compound being in a particular class. "Naïve" refers to the fact that it is assumed that the features are independent. NB can handle multiple classes. It works surprisingly well given its simplicity, one suggestion being that the dependence between features at least partly cancels out.[7] In the past there have been comparisons of NB and RF, for example Svetnik et al.[16] and Caruna and Niculescu-Mizil.[18] In those studies NB gives quite poor predictions relative to RF. On the other hand, NB has some good points. It is very efficient computationally, and there is some suggestion that NB can work better than some more complex methods in the presence of noisy data.[19] There is a large literature on NB, for example,[20−23] and this is partly because NB is implemented in one of the most widely used commercial software platforms Pipeline Pilot (www.accelrys.com). We will refer to the implementation of Laplacian-modified NB in Pipeline Pilot as PLPNB.

At pharmaceutical companies *in silico* methods are commonly used to predict on-target or ADME (Absorption Distribution Metabolism Excretion) activity in an effort to save on experimental testing.[24−26] We want QSAR methods that make accurate predictions, but there are other considerations. We must sometimes deal with very large (>100,000 molecules) data sets, so computational cost is an important issue. Pharmacological data (especially ADME-related assays) can be very noisy and can contain large amount of "qualified data" (e.g., IC50 > 30 μM). While one might intuitively expect more sophisticated QSAR methods to make better predictions, experience has shown that this is often not true. It certainly is conceivable that in the world of real pharmacological data an efficient and noise-robust classification method like NB might prove more useful than a more

Received: December 20, 2011
Published: February 23, 2012

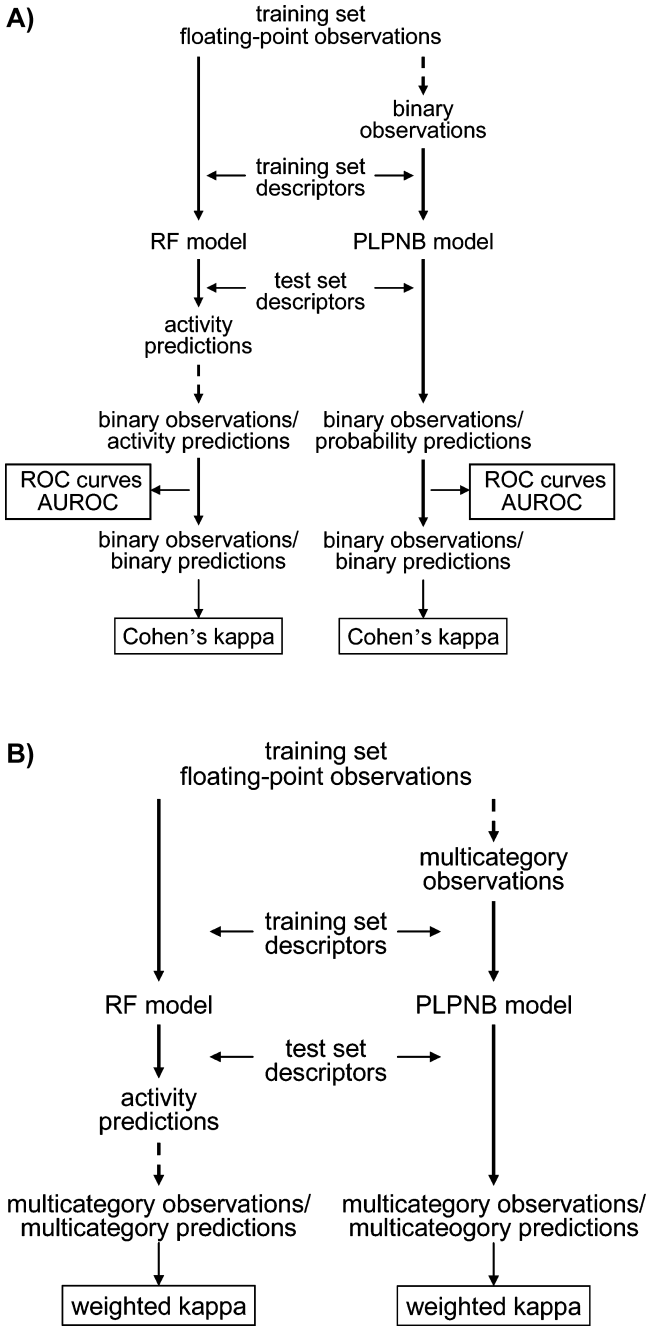


Figure 1. A. Workflow for handling binary data. The dashed arrows indicate conversion of floating-point values to categories. B. Workflow for handling multicategory data.

complicated method like RF. Thus a head-to-head comparison between RF and PLPNB on real-world QSAR data sets was undertaken.

In this paper we compare RF and PLPNB in their ability to make prospective predictions on 18 large, diverse in-house data sets given a common set of descriptors. Under such realistic conditions, RF makes as good or better predictions than PLPNB, although at a much greater computational cost.

METHODS

Descriptors. Our preference with QSAR is to deal with substructure descriptors (e.g., atom pairs, MACCS keys, circular fingerprints, etc.), as opposed to descriptors that apply to the whole molecule (e.g., number of donors, LOGP, molecular weight, etc.). There are many possible substructure descriptors we could use, but here we will deal with descriptors we have found most useful for RF and with the descriptors most commonly used for PLPNB.

In our hands the most useful set of descriptors for RF is the union of AP (the original "atom pair" descriptor from Carhart et al.[27]) and DP descriptors ("donor–acceptor pair" called "BP" in Kearsley et al.[28]). Both descriptors are of the form

Atom type i − (distance in bonds) − Atom type j

For AP, atom type includes the element, number of non-hydrogen neighbors, and number of pi electrons; it is very specific. For DP, atom type is one of seven (cation, anion, neutral donor, neutral acceptor, polar, hydrophobe, and other); it is more general.

ECFP4 and ECFP6 are Pipleline Pilot-native "circular fingerprint" descriptors[29] at radii of up to 4 and 6 bonds, respectively, around a given atom. ECFP6 contains ECFP4 as a subset. These descriptors are widely used outside Pipeline Pilot and are regarded as among the best descriptors for similarity, for example.[30−33] The ECFP4 and ECFP6 descriptors we use here were generated from Pipeline Pilot and stored in files for use by both QSAR methods.

RF normally handles input data as a matrix of molecules and descriptors. Each element E(i,k) of the matrix is the count of descriptor k in molecule i. NB by default handles just the presence or absence of descriptor k. Optionally, NB can use descriptor counts if it combines the name with the count. For example in a molecule where descriptor k occurred 3 times, the descriptor in that molecule would get a special name like "k_3"; whereas a molecule where the descriptor occurred twice would have the descriptor "k_2". NB would still count only the presence or absence of "k_3" and "k_2" and treat them as independent descriptors.

Implementation of Methods. The version of RF we are using is a modification of the original FORTRAN code from Breiman.[11] It has been parallelized to run one tree per processor. Such parallelization is necessary to run some of our larger data sets in a reasonable time. For all RF models, we generate 100 trees with m/3 descriptors used at each branch-point, where m is the total number of descriptors.

PLPNB is as implemented in Pipeline Pilot Version 7.5.

In order to have PLPNB use non-PLP descriptors, and to be sure both PLPNP and RF are seeing exactly the same descriptors, it was necessary to write a wrapper that runs in our Linux modeling environment to pass activity and descriptor data to Pipeline Pilot running on Windows. The wrapper allowed us to easily make the comparison between methods running on different platforms.

Categorical Predictions in PLPNB. NB handles only categorical data as input. We examine two types in this paper:

1 Binary for instance "0" = inactive and "1" = "active"
2 Multicategory. Here we deal with three categories. For example "1" = low, "2" = medium", "3" = "high".

The current implementation of PLPNB, given binary data, returns normalized probability predictions (i.e., floating-point values). Given multicategory data, it returns multicategory predictions.

We tested least three ways of making a multicategory model in PLPNB:

1 The first, which is the "native" method for PLPNB, takes "1", "2", and "3" as unrelated categories and generates

dx.doi.org/10.1021/ci200615h | J. Chem. Inf. Model. 2012, 52, 792−803

Table 1. Data Sets for Prospective Prediction

data set	description	N	binary cutoff	multicategory cutoffs	mean pairwise similarity in training set[b]	mean SIMILARITYNEAREST for test set[c]
3A4	CYP 3A4 inhibition −log(IC50) M	50000	5	5, 7	0.31 ± 0.10	0.75 ± 0.12
CB1[a]	CB1 binding −log(IC50) M	11640	7	6, 8	0.38 ± 0.13	0.71 ± 0.15
DPP4[a]	DPP4 inhibition −log(IC50) M	8327	6	6, 8	0.33 ± 0.12	0.71 ± 0.10
HERG	HERG inhibition −log(IC50) M	50000	5	5, 7	0.31 ± 0.09	0.70 ± 0.13
HIV_INTEGRASE[a]	HIV integrase cell based assay −log(IC50) M	2421	6	5, 7	0.46 ± 0.16	0.90 ± 0.06
HIV_PROTEASE[a]	HIV protease inhibition −log(IC50) M	4311	6	6, 8	0.48 ± 0.13	0.76 ± 0.11
HPLC_LOGD	logD measured by HPLC method	50000	2	1, 5	0.31 ± 0.09	0.72 ± 0.12
METAB	percent remaining after 30 min microsomal incubation	2092	40	30, 70	0.34 ± 0.10	0.80 ± 0.13
NAV	NAV1.5 inhibition −log(IC50) M	46245	5	5, 7	0.31 ± 0.10	0.78 ± 0.13
NK1[a]	NK1 (substance P) receptor binding −log(IC50) M	13482	8	7, 9	0.40 ± 0.12	0.83 ± 0.08
OX1[a]	Orexin 1 inhibition −log(KI) M	7135	6	6, 8	0.42 ± 0.13	0.81 ± 0.11
OX2[a]	Orexin 2 inhibition −log(KI) M	14875	7	6, 8	0.40 ± 0.12	0.81 ± 0.14
PGP	log(BA/AB) human p-glycoprotein	8603	0.4	0.4, 1	0.33 ± 0.10	0.67 ± 0.13
PPB	human plasma protein binding log(bound/unbound)	11622	1.5	1, 2	0.34 ± 0.10	0.73 ± 0.14
PXR	pregnane X receptor maximum activation (percent) relative to rifampicin	50000	40	40, 80	0.31 ± 0.09	0.74 ± 0.12
RAT_F	log(rat bioavailability) at 2 mg/kg	7821	1	1, 1.7	0.30 ± 0.11	0.64 ± 0.14
TDI	time dependent 3A4 inhibitions log(IC50 without NADPH/IC50 with NADPH)	5559	0.4	0.2,1	0.35 ± 0.10	0.70 ± 0.12
THROMBIN[a]	human thrombin inhibition −log(IC50) M	6924	6	6, 8	0.36 ± 0.11	0.70 ± 0.12

[a]On-target data set. [b]Based on AP descriptors and Dice similarity index. Compared to 0.26 ± 0.10 for pairs of random compounds and 0.7 for clear analogs. [c]SIMILARITYNEAREST = similarity to the most similar compound in the training set.

Table 2. Descriptor Count in the NAV Data Set (~46,000 Molecules)

descriptor	mean unique per molecule	total unique in data set
APDP	300	7952
TT	45	3673
7PATHS	343	154738
DRUGBITS	14	288
ECFP4	62	40277
ECFP6	86	146524

probabilistic models of "1" vs "non-1", "2" vs "non-2", and "3" vs "non-3". The predicted category for a molecule is determined by which category has the highest normalized probability.

2 The second method, developed by us and which we call the "zscore" method, modifies the native method to better take into account differences in probability distributions among the classes. Again this requires making three binary models: "1" vs "non-1", "2" vs "non-2", and "3" vs non-3". Let m(i), and sd(i) be the mean leave-one-out predicted probability for all the compounds in the training set for category i. Let s(i) be the "best split" for the leave-one-out probability. "Best" here means the cutoff that minimizes the percent misclassified for the i vs non-i model. For a prediction, let p(i) be the predicted probability for being in category i based on the model for i vs non-i.

We define zscore(i) = (p(i) − s(i))/sd(i)

The molecule being predicted is assigned to whichever class i has the highest zscore.

3 The third method, which we call the "two-model method", makes the assumption that "3" > "2" > "1". It was developed at Merck and depends on generating two binary models: "1" vs "2 + 3" and "1 + 2" vs "3". Two separate predictions are made for each molecule. If the molecule is predicted as "1" in the first model and "1 + 2" in the second model, it is declared "1". If the molecule is predicted as "2 + 3" in the first model and "3" in the second model, it is declared as "3". Anything else is declared "2".

Metrics. There are a number of metrics that one can use to compare predicted and observed activities given that at least the observed data is binary or multicategory:

1 ROC curves. If the predictions are floating point (probability or quantitative) and the observations are binary, one can sort the predictions by decreasing activity and generate a "receiver operator characteristic" (ROC) curve: fraction true positives on the *y*-axis and fraction false positives on the *x*-axis. The ROC curve measures how much the front of the sorted list is enriched in actives. A ROC curve that is a left step function is perfect, i.e. all the true positives have been found before any false positives. A ROC curve that is along the diagonal is no better than random.

2 Area under the ROC curve (AUROC). A perfect sorting of predictions would have an AUROC of 1.0. A sorting that was no better than random would have an AUROC of 0.5.

3 Cohen's kappa. We will use this when both the predictions and observations are binary. Kappa measures how far the agreement of predicted and observed is above that expected by chance based on the frequency of the categories. Kappa = 1 is perfect agreement; kappa = 0 is no better than chance. If predictions are floating-point and there are two categories, one needs to convert the predictions to binary before calculating kappa, and this means finding a cut point in the predictions; below the cut point a prediction is assigned to be "0" and above the cut point it is assigned "1". For PLPNB models, the cut point that gives the maximum kappa based on the self-fit predictions of the training set is stored with the model. This cut point is read from the model and used to classify

dx.doi.org/10.1021/ci200615h | J. Chem. Inf. Model. 2012, 52, 792−803

Table 3. Metrics for Binary Time-Split Predictions (All Molecules)

data set	PLPNB/ APDP	PLPNB/ ECFP4	PLPNB/ ECFP6	RF/ APDP	RF/ ECFP4
AUROC					
3A4	0.727	0.820	0.823	0.808	0.842
CB1	0.611	0.639	0.647	0.812	0.755
DPP4	0.629	0.623	0.633	0.734	0.691
HERG	0.670	0.696	0.690	0.780	0.731
HIV_INTEGRASE	0.779	0.802	0.805	0.779	0.729
HIV_PROTEASE	0.937	0.971	0.974	0.987	0.987
HPLC_LOGD	0.788	0.806	0.795	0.906	0.893
METAB	0.855	0.879	0.879	0.935	0.909
NAV	0.764	0.767	0.771	0.807	0.816
NK1	0.836	0.861	0.870	0.873	0.879
OX1	0.566	0.713	0.748	0.842	0.813
OX2	0.810	0.896	0.908	0.894	0.904
PGP	0.790	0.770	0.772	0.880	0.860
PPB	0.751	0.744	0.743	0.854	0.829
PXR	0.728	0.799	0.807	0.835	0.829
RAT_F	0.670	0.690	0.690	0.670	0.700
TDI	0.833	0.869	0.872	0.864	0.860
THROMBIN	0.552	0.727	0.740	0.752	0.762
mean	0.739	0.782	0.787	0.834	0.820
Cohen's kappa					
3A4	0.26	0.40	0.42	0.42	0.49
CB1	0.24	0.27	0.28	0.43	0.38
DPP4	0.15	0.11	0.12	0.33	0.10
HERG	0.24	0.26	0.25	0.4	0.33
HIV_INTEGRASE	0.31	0.35	0.33	0.26	0.19
HIV_PROTEASE	0.35	0.58	0.59	0.74	0.65
HPLC_LOGD	0.37	0.46	0.43	0.62	0.58
METAB	0.57	0.62	0.64	0.65	0.63
NAV	0.35	0.35	0.33	0.40	0.38
NK1	0.39	0.45	0.46	0.46	0.50
OX1	0.11	0.31	0.39	0.43	0.45
OX2	0.42	0.69	0.69	0.63	0.67
PGP	0.45	0.44	0.44	0.56	0.52
PPB	0.40	0.33	0.31	0.50	0.44
PXR	0.32	0.45	0.45	0.47	0.49
RAT_F	0.13	0.20	0.23	0.19	0.20
TDI	0.46	0.50	0.53	0.57	0.57
THROMBIN	0.01	0.17	0.25	0.22	0.27
mean	0.31	0.39	0.40	0.46	0.44

the predictions. For RF predictions, the division is made at the cutoff for the observed activities. For example, if the cutoff used to construct the binary observed data is 6, a prediction ≥6 would be set to "1".

4 Weighted kappa. If there are three or more categories and the order of the categories is meaningful e.g. "3" > "2" > "1", one may use weighted kappa. Weighted kappa penalizes the case where the predicted categories are off by 1 less than the case where they are off by 2, for example, if a "3" is predicted as a "2" vs a "3" predicted as a "1", respectively. If there only two categories, weighted kappa is mathematically equivalent to Cohen's kappa. If the predictions are quantitative, they must be converted to multicategory before calculating weighted kappa. For example if the cutoffs for converting the observed data are 6 and 8, a prediction <6 would be set to "1", a prediction between 6 and 8 would be assigned a "2" and prediction ≥8 would be set to "3".

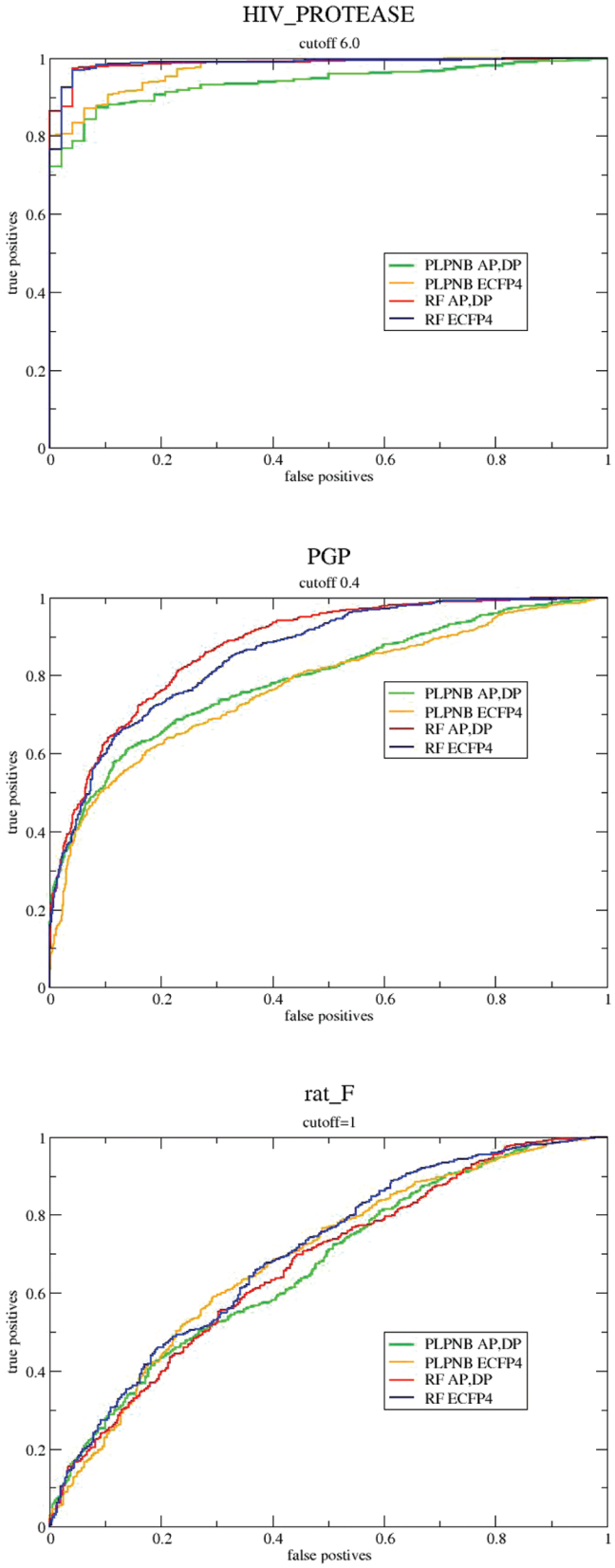


Figure 2. Example ROC curves for time-split binary data sets.

We should point out that metrics 1 and 2 consider only whether compounds are predicted in the correct order. Metrics 3 and 4 require more quantitative agreement of predicted and observed activities.

dx.doi.org/10.1021/ci200615h | J. Chem. Inf. Model. 2012, 52, 792–803

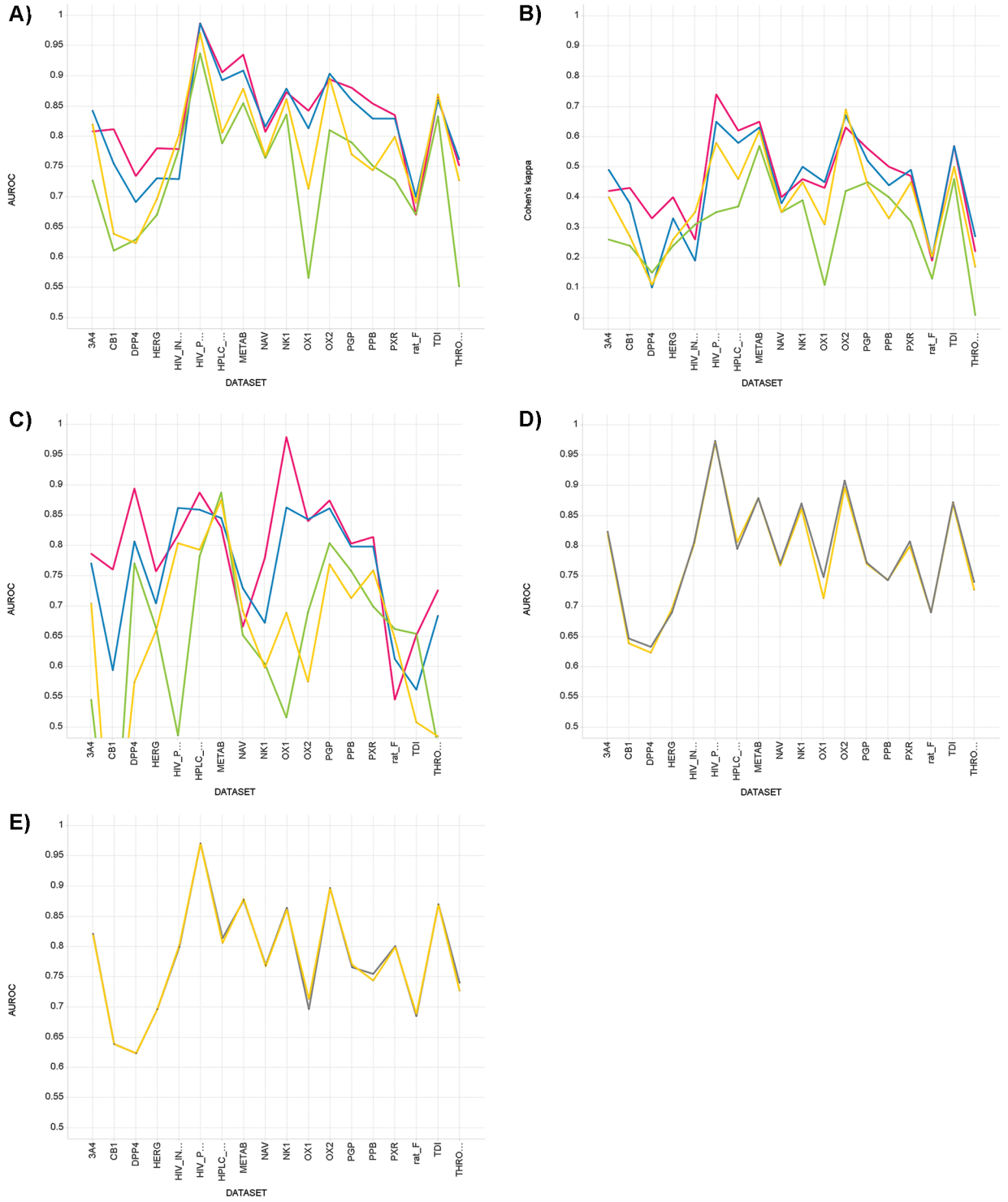


Figure 3. A. AUROC for binary time-split showing all method/descriptor combinations. Data sets are in alphabetical order along the *x*-axis. Red = RF/APDP, blue = RF/ECFP4, green = PLPNB/APDP, orange = PLPNB/ECFP4. B. Cohen's kappa for binary time-split showing all method/descriptor combinations. Same color key as Figure 3A. C. AUROC for binary time-split showing all method/descriptor combinations for compounds dissimilar to the training set, where "dissimilar" means <0.6 AP/Dice to the most similar compound. HIV_INTEGRASE is missing because there are only a few dissimilar compounds. Same color key as Figure 3A. D. AUROC for binary time-split showing PLPNB/ECFP4 (orange) vs PLPNB/ECFP6 (gray). E. AUROC for binary time-split showing PLPNB/ECFP4 "presence only" (orange) vs PLPNB/ECFP4 "descriptor count" (gray).

Training and Test Sets. There are a number of approaches for judging how well QSAR methods make predictions. Normally one holds out a subset of the data as the test set, makes a QSAR model on what is left (the training set), then one predicts the activities of the test set and compares the predictions to the observed activities. The simplest way of making the split between training set and test set is by random selection. One other suggestion is to construct training and test

dx.doi.org/10.1021/ci200615h | *J. Chem. Inf. Model.* 2012, 52, 792–803

Table 4. Metrics for Binary Time-Split Predictions on Compounds Dissimilar to Anything in the Training Set

data set	PLPNB/ APDP	PLPNB/ ECFP4	RF/ APDP	RF/ ECFP4
AUROC				
3A4	0.545	0.704	0.786	0.77
CB1	0.232	0.234	0.760	0.594
DPP4	0.770	0.574	0.894	0.807
HERG	0.664	0.659	0.757	0.704
HIV_PROTEASE	0.486	0.804	0.817	0.862
HPLC_LOGD	0.782	0.793	0.887	0.859
METAB	0.887	0.875	0.830	0.845
NAV	0.652	0.692	0.666	0.730
NK1	0.605	0.598	0.779	0.672
OX1	0.516	0.689	0.979	0.863
OX2	0.690	0.575	0.840	0.843
PGP	0.804	0.769	0.874	0.861
PPB	0.758	0.713	0.803	0.798
PXR	0.699	0.759	0.814	0.798
RAT_F	0.662	0.647	0.545	0.613
TDI	0.654	0.508	0.652	0.562
THROMBIN	0.467	0.485	0.726	0.684
mean	0.640	0.652	0.789	0.757
Cohen's kappa				
3A4	0.06	0.23	0.39	0.32
CB1	−0.12	0.00	0.00	−0.02
DPP4	0.30	−0.01	0.66	0.01
HERG	0.20	0.22	0.34	0.27
HIV_PROTEASE	−0.08	0.39	0.36	0.16
HPLC_LOGD	0.39	0.39	0.51	0.51
METAB	0.45	0.38	0.40	0.23
NAV	0.18	0.15	0.20	0.16
NK1	−0.08	−0.26	0.04	0.04
OX1	−0.03	−0.02	−0.03	−0.03
OX2	0.00	0.00	0.00	0.00
PGP	0.47	0.33	0.49	0.42
PPB	0.39	0.36	0.36	0.28
PXR	0.24	0.36	0.42	0.42
RAT_F	0.13	0.11	0.06	0.07
TDI	0.14	−0.01	0.25	0.15
THROMBIN	−0.05	−0.1	0.03	0.04
mean	0.15	0.15	0.26	0.18

sets from compounds from the same clusters.[34,35] However, both of these are likely to give optimistic results because each molecule in the test set is likely to have a similar compound in the training set, and it is well understood that molecules with neighbors in the training set are more likely to be predicted well.[36] On the other hand, methods such as "leave class out"[37] are unduly pessimistic because by definition the test set will have no similar compounds in the training set. In actual practice in a pharmaceutical environment, QSAR models are applied "prospectively", that is, predictions are made for compounds not yet tested in the appropriate assay at the time the model was made, and these compounds may or may not have analogs in the model. The best way of simulating this is to generate training and test sets by "time-split", i.e. one builds a model on assay data available at a certain date and tests the model on data that are generated later. Indeed we have found that, for regressions, the R^2 from time-split validation better estimates the R^2 for true prospective prediction than the R^2 from any "split at random" scheme. For this paper we take the first 75% of the molecules assayed as the training set and take the last 25% as the test set. How many test set compounds have analogs in the training set will vary between data sets, but with a large enough collection of data sets we can get a reasonable idea of the merits of QSAR method/descriptor combinations.

Workflow. PLPNB is confined to categorical observed data, and therefore comparisons between the methods must be made referring to categorical observations. On the other hand, our parallelized Breiman implementation of RF, which is the only implementation available to us that can handle some of the larger data sets, takes floating-point activities and generates quantitative predictions, i.e. it natively does regressions but not classifications. We felt it would be a realistic test to run each method in its "native" state. That is, PLPNB would build a model with binary observed activities and return floating-point probability predictions or work with three-category observed activities and return three-category predictions. RF would work in regression mode, starting with floating-point observed activities and returning floating-point predictions. The observations and/or predictions would be converted to categories using the appropriate cutoffs before metrics were calculated. Figure 1A,B shows the workflow for doing the comparison for the binary and multicategory cases, respectively.

Data Sets. For this study we arbitrarily chose 18 Merck data sets shown in Table 1. These include a mix of on-target data sets and ADME data sets. Some data sets are so large (>100,000) that we randomly selected a smaller subset of compounds (50,000) to expedite the study. It is useful to use proprietary data sets for two reasons:

1 We wanted data sets which are realistically large and have a realistic level of noise but are not as noisy as high-throughput data sets.
2 Time-splitting requires dates of testing, and these are almost impossible to find in public domain data sets.

The sixth column in Table 1 shows the mean pairwise AP/Dice similarity for the training sets. The distribution of those similarities are shown in Supporting Information Figure S1. The mean similarity is generally not far from that of randomly selected compounds (0.26) and far from the similarity one would see for obvious analogs (~0.7). This indicates that most of these data sets, both ADME and on-target, are very diverse. HIV_PROTEASE, and to a lesser extent, HIV_INTEGRASE, and CB1 are less diverse than the others. One would come to similar conclusions with other measures of similarity (e.g., ECFP4/Tanimoto). One can also monitor the coverage of chemical space in the test sets by the training set. For each test set compound one can find the similarity to the nearest compound in the training set (SIMILARITYNEAREST). The mean SIMILARITYNEAREST for each test set is listed in the seventh column of Table 1, and the distributions of SIMILARITYNEAREST are shown in Supporting Information Figure S2. For nearly all data sets, there is a distribution of compounds both near and far from the training set. The one exception is HIV_INTEGRASE where almost all of the test set compounds are close to the training set.

The cutoff used to assign binary activities is given in the third column of Table 1 and the cutoffs for multicategory activities are in the fourth column. The cutoffs are chosen more or less arbitrarily, but with an idea of what is "biologically meaningful". It is possible to assign cutoffs such that the training sets would be so imbalanced (e.g., many "inactives" and very few "actives") that no QSAR method would be able to make reasonable

Table 5. Weighted Kappa for Muticategorical Time-Split Predictions

data set	PLPNB/native/APDP	PLPNB/native/ECFP4	PLPNB/zscore/APDP	PLPNB/zscore/ECFP4	PLPNB/two-model/APDP	PLPNB/two-model/ECFP4	RF/APDP	RF/ECFP4
3A4	0.26	0.43	--[a]	--[a]	0.29	0.44	0.39	0.45
CB1	0.24	0.24	0.23	0.23	0.23	0.23	0.4	0.29
DPP4	0.12	0.19	0.11	0.14	0.13	0.13	0.16	0.06
HERG	0.20	0.23	0.21	0.25	0.23	0.24	0.38	0.32
HIV_INTEGRASE	0.25	0.26	0.42	0.38	0.45	0.44	0.44	0.39
HIV_PROTEASE	0.46	0.51	0.42	0.50	0.43	0.49	0.51	0.48
HPLC_LOGD	0.23	0.28	0.26	0.25	0.28	0.31	0.33	0.37
METAB	0.56	0.58	0.55	0.57	0.55	0.58	0.59	0.61
NAV	0.14	0.20	0.12	0.21	0.28	0.25	0.4	0.38
NK1	0.30	0.35	0.35	0.38	0.31	0.43	0.46	0.44
OX1	0.11	0.12	0.14	0.16	0.12	0.16	0.41	0.43
OX2	0.40	0.50	0.36	0.34	0.43	0.53	0.47	0.51
PGP	0.46	0.43	0.35	0.44	0.43	0.44	0.50	0.49
PPB	0.32	0.26	0.33	0.28	0.32	0.29	0.42	0.41
PXR	0.25	0.37	0.25	0.34	0.27	0.37	0.41	0.42
RAT_F	0.14	0.14	0.12	0.18	0.10	0.17	0.13	0.16
TDI	0.46	0.48	0.42	0.46	0.46	0.48	0.48	0.52
THROMBIN	0.15	0.22	0.14	0.19	0.16	0.26	0.20	0.24
mean	0.28	0.32	0.27	0.29	0.30	0.35	0.39	0.39

[a]Predictions returned a single category, so weighted kappa could not be calculated.

predictions. In a separate study we determined that the cutoffs in Table 1 are not problematical in that regard.

The cutoffs in Table 1 would apply to the observed activities for parts of the RF and PLPNB workflow and to predicted activities for parts of the RF workflow. For example for 3A4, in the binary case a value of $-\log(IC50) < 5$ would be categorized as "0" (inactive) and $-\log(IC50) \geq 5$ would be "1" (active). In the multicategory case $-\log(IC50) < 5$ would be "1" (low), $5 \leq -\log(IC5) < 7$ would be "2" (medium), and $-\log(IC50) \geq 7$ would be "3" (high).

A number of these data sets contain significant amounts of "qualified data", for example, one might know IC50 > 30 μM because 30 μM was the highest concentration tested during a titration. For PLPNB all the qualified data indicating a high IC50 would be treated as "0" (inactive) for the binary case or "1" (low) for the multicategory case. For the purposes of RF regression, those activities were treated as fixed numbers, in this example 30 μM or $-\log(IC50) = 4.5$. It may seem problematical to include qualified data in a regression, but if those data are included in the PLPNB model, they must also be included in the RF model to make a proper comparison. Also, we found leaving very inactive compounds out of RF models is deleterious to prediction accuracy; without very inactive compounds in the training set, inactive compounds in test sets are predicted to be more active than they are.

Robust Comparison of Methods. The set of 18 data sets here is our "target set." One common way of determining whether one method is "better" than another is to take for each method the mean value of some metric, say AUROC, over all the data sets. (Here a "method" would be a QSAR method/descriptor combination.) Method A is presumably better than method B if the mean AUROC of A is higher than that of B. However, one needs to monitor how sensitive the order of the mean AUROCs are to the exact composition of the target set. It is possible that one or two data sets in the target set may be anomalous and skewing the conclusion. One suggestion from our laboratory[38] for visualizing this information is "target set perturbation". Here, in a modification to the original method, we generate 1000 perturbed target sets by bootstrapping the targets with replacement. In a target set perturbation plot one arranges the methods along the *x*-axis and the mean (over 1000 bootstrap samples) of the mean AUROC is plotted along the *y*-axis with the standard deviation of the mean AUROC as an error bar. If the error bars of two methods significantly overlap, one has to conclude that they are not truly distinguishable, at least not in the target set on hand. In this paper we add a simple way of quantitating the robustness. If the mean AUROC of method A is greater than that of method B in n out of 1000 samples, then the probability that A is not really better than B in this study is 1−n/1000.

RESULTS

We attempted to build QSAR models on the training sets according to the workflows in Figure 1 for every method/descriptor combination and generated the appropriate metrics on the test sets.

Timing and Applicability of Descriptors. One aspect we had not appreciated before executing this study was the different computational requirements of model-building based on which descriptor was used. Generally we found that generating a PLPNB model takes at most several minutes elapsed time for any of the data sets with any of the descriptors. In retrospect this is not at all surprising, since model-building with NB requires only a single pass through the data. In contrast, generating an RF model could take a few hours or overnight with the ECFP4 descriptors, even with our parallelized version of RF running on 100 Linux cluster nodes. It was not practical in terms of clock time or memory to generate RF models on some of the larger training sets with ECFP6. The explanation for this has to do with the number of descriptors. Table 2 shows the number of unique descriptors per molecule and the total descriptors for an example data set. Although the number of ECFP4s and ECFP6s per molecule is relatively low compared to APDP, the number of unique descriptors over the whole data set (which is what determines the "size" of a QSAR problem) is high for ECFP4 and even higher for ECFP6. Therefore we had to limit the study to ECFP4 as the common Pipeline Pilot-native descriptor for RF and PLPNB. Later we

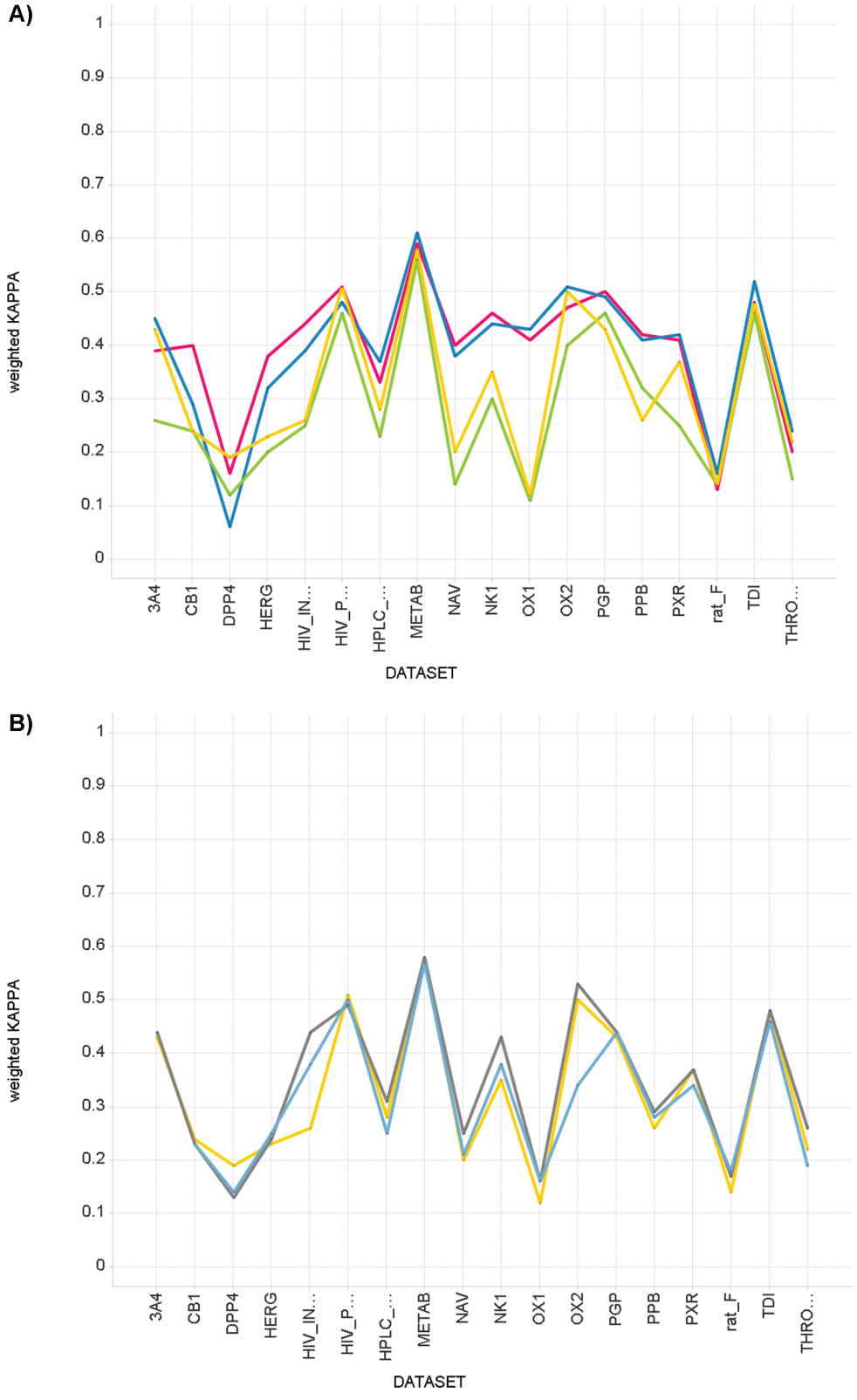


Figure 4. A. Weighted kappa for multicategory time-split showing all method/descriptor combinations. Data sets are in alphabetical order along the *x*-axis. The PLPNB method shown is the native method. Red = RF/APDP, blue = RF/ECFP4, green = PLPNB/APDP, orange = PLPNB/ECFP4. B. Weighted kappa of different methods of multicategory prediction in PLPNB/ECFP4. Orange = native method, blue = Zscore method, gray = two-model method.

will see that the bond radius (4 vs 6) of ECFP descriptors used with PLPNB does not matter.

On the other hand, prediction of a test set against either PLPNB or RF models takes about the same time, less than a minute for all data sets/method/descriptor combinations.

Binary Time-Split. For all data sets, we calculated the ROC curves, AUROC and Cohen's kappa for all descriptor combinations except RF/ECFP6. These data are in Table 3. Examples of ROC curves are shown in Figure 2. We show a very good, an average, and a poor set of predictions. Figure 3 A,B shows

dx.doi.org/10.1021/ci200615h | J. Chem. Inf. Model. 2012, 52, 792–803

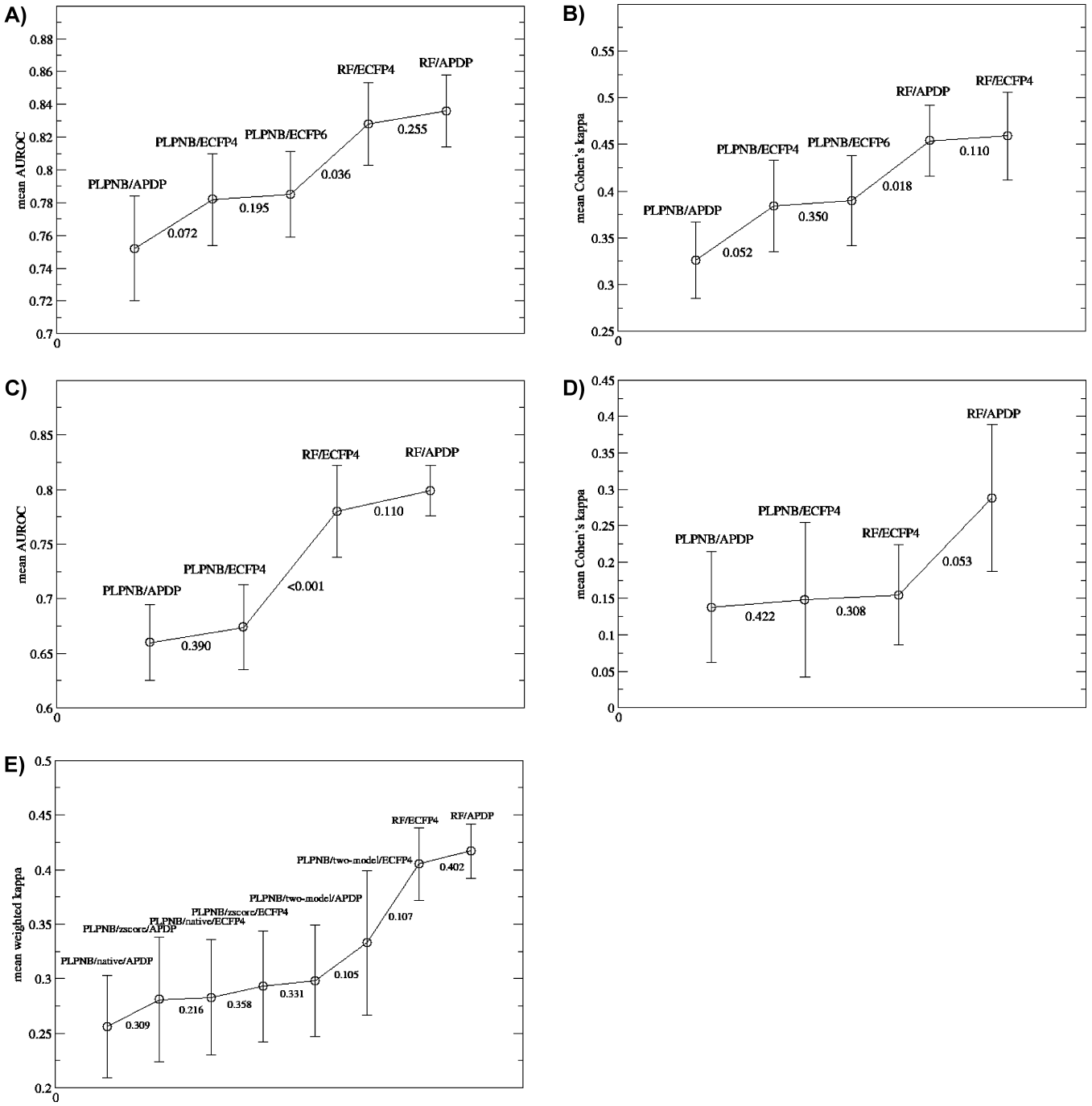


Figure 5. A. Target set perturbation plot for AUROC on binary time-split predictions. The mean (over 1000 bootstrap resamplings) of the mean AUROC is shown as a circle. The standard deviation is shown as error bars. B. Target set perturbation plot for AUROC on binary time-split predictions. C. Target set perturbation plot for AUROC on binary time-split predictions for dissimilar compounds. D. Target set perturbation plot for Cohen's kappa on binary time-split predictions for dissimilar compounds. E. Target set perturbation plot for weighted kappa for multicategory prediction.

AUROC and Cohen's kappa for all data sets. Both metrics tell the same story. While AUROC and kappa vary quite a bit from data set to data set, RF (red and blue lines for APDP and ECFP4, respectively) tends to be as good or better than PLPNB/ECFP4 (orange line), with RF/APDP being especially good. The combination PLPNB/APDP (green line) appears to be poor. Based on the idea that NB is less sensitive to noise, one might expect the data sets on which PLPNB does as well as RF to be the "noisy" ones, i.e. the ones less well predicted overall, but that does not seem to be the case. There also does not seem to be a simple relationship between high AUROC and kappa and the mean pairwise similarity in the training set or the mean SIMILARITYNEAREST of the test set to the nearest compound in the training set.

It is important to establish whether the superiority of RF holds for molecules far from the training set, analogous to a "leave class out" validation. These data are in Table 4. In Figure 3C the AUROC includes only those compounds that are less similar than 0.6 to anything in the training set, where similarity is measured by the AP descriptor and Dice similarity index.

dx.doi.org/10.1021/ci200615h | J. Chem. Inf. Model. 2012, 52, 792–803

(This is about equivalent to ~0.35 or ~0.40 using ECFP4 or ECFP6, respectively, with the Tanimoto similarity index.) At this level of similarity, compounds do not appear to be obvious analogs to most chemists. Again RF (red and blue) is as good or better than PLPNB. Cohen's kappa supports the same conclusion. Thus, RF is superior to PLPNB also for compounds that are structurally less similar to the training set molecules.

Figure 3D shows that PLPNB does very similarly with ECFP4 and ECFP6. Figure 3E shows that using descriptor counts with PLPNB is almost identical to using the default "presence only".

Multicategory Time-Split. For all data sets, we calculated weighted kappa for all descriptor combinations. These data are in Table 5. Since the binary data sets showed no real difference between ECPF4 and ECFP6 and between "presence only" and "descriptor count", we did not attempt varying these parameters in the multicategory problems.

Figure 4A shows weighted kappa for all the data sets. Again, RF (red and blue lines) is approximately as good or better than any of the PLPNB methods. Figure 4B shows the relative merits of the three multicategory methods for PLPNB. It is hard to pick out a clear winner from Figure 4B; however, the robustness analysis (discussed below) shows that the two-model method is on average better than the other two.

Target Set Perturbation Plots. Probabilities for pairs of methods are in the Supporting Information as Tables S1–S3. Target set perturbation plots are shown in Figure 5. In each the methods are arranged along the *x*-axis in increasing goodness of prediction. On the line connecting methods A and B is the probability that B is not better than A. Generally speaking, the conclusions one draws about the relative goodness of QSAR method/descriptor combinations from looking at the average over all data sets in Figure 5 is congruent with the conclusions drawn by looking at the set of individual data sets in Figures 3 and 4. One observation is that it is somewhat harder using Cohen's kappa to distinguish one method from another than it is using AUROC. In retrospect, this might be expected because putting the predictions into categories, as is necessary for calculating kappa, loses information.

In the case of binary time splits (Figure 5A,B), RF is clearly distinct from PLPNB methods. PLPNB/ECFP4 is not distinguishable from PLPNB/ECFP6, and RF/APDP is not distinguishable from RF/ECFP4. The inferiority of PLPNB/APDP relative to PLPNB/ECFP4 seems clear, but it may not be as significant as the other distinctions. In the case of binary split on dissimilar compounds (Figure 5 C,D), there is some disagreement between the metrics. AUROC shows a clear superiority of RF over PLPNB, while Cohen's kappa shows only that RF/AP is probably distinct from all other methods. In neither case is PLPNB/APDP distinct from PLPNB/ECFP4. For multicategory, Figure 5E and Table S4 does not include 3A4 because some data are missing for that target (see Table 5). RF/APDP is not distinguishable from RF/ECFP4. PLPNB/native and PLPNB/zscore are not distinguishable in either descriptor. However, PLPNB/two-model/ECFP4 is clearly better than the other PLPNB methods and is intermediate between PLPNB/native/ECFP4 and RF/ECFP4. RF using either descriptor is clearly better than PLPNB/native or PLPNB/zscore.

DISCUSSION

We can draw the following conclusions from this study:

1 PLPNB is a very fast and efficient method.
2 PLPNB does poorly with the APDP descriptor in combination with the Pipeline Pilot-native descriptors ECFP4 and ECFP6. Despite the fact that there are a few-fold more ECFP6 descriptors than ECFP4 descriptors, ECFP6 does not seem to add additional structure/activity information. RF does well with either APDP and ECFP4.
3 Of the three multicategory schemes we have tried for PLPNB, the two-model method seems superior.
4 For PLPNB it does not seem to matter whether one uses descriptor counts or "presence only".
5 Predictions from RF are at least as good and in most cases better than those from PLPNB. This is true for on-target data as well as ADME data.

We obtained similar results for points 2 and 5 in an earlier pilot study with smaller binary data sets where cross-validation was used instead of time-split validation and where the classification version of RF was used instead of the regression version (data not shown), so we are confident in the robustness of the conclusions, aside from the formal robustness analysis shown here.

For point 2 above, there is literature precedent that PLPNB would be equally predictive with ECFP4 and ECFP6. Prathipati et al.[39] ran PLPNB on a single model (antituberculosis compounds) using ECFP4, ECFP5, ECFP6,..., ECFP12 and found the predictivity on test sets was about equal for all of those descriptors. In contrast, it was somewhat unexpected that PLPNB should do poorly with APDP descriptors compared to ECFP4 and ECFP6. We have also found simlar poor performance with the TT descriptor[40] and the DRUGBITS[16] descriptor (data not shown). The latter is consistent with the work of Svetnik et al.[16] One possible explanation for why Pipeline Pilot-native descriptors are needed for PLPNB to do well has to do with the number of unique descriptors per problem (see Table 2). Since NB by definition includes no coupling between descriptors, one might have to make up for that by including more descriptors. There might not be sufficient descriptors in the problem with APDP, TT, or DRUGBITS. One can test that speculation by examining yet other descriptors in those cases where the difference between PLPNB/APDP and PLPNB/ECFP4 is the greatest (OX1 and THROMBIN). 7PATHS is an in-house generalization of TT descriptors, for paths of between 1 and 7 atoms. There are as many 7PATHS descriptors as ECFP6 descriptors in most data sets. PLPNB does as well with 7PATHS descriptors on OX1 and THROMBIN as with ECFP4 and ECFP6, consistent with the idea that PLPNB requires a large number of descriptors to work well (though not necessarily descriptors native to Pipeline Pilot). This may explain in-house anecdotal observations that NB behaves better in Pipeline Pilot than in other implementations: In Pipeline Pilot, a large number of descriptors per data set is available by default.

For point 5 above, earlier work[16,18] did suggest that RF was superior to NB most of the time. However, those tests were done on smaller data sets, the validation was done by cross-validation, and Pipeline Pilot-native descriptors were not used. It was necessary for us to settle the RF vs PLPNB question by running tests where each method could be used with the best descriptors for the other method. Also here we do a very realistic validation on large and diverse data sets. By the "No Free Lunch Theorum" we would not expect any method/descriptor to work best on every data set. However, the combination RF/APDP seems consistently very good, especially for compounds dissimilar to the training set, and it is less expensive than RF/ECFP4. Overall our recommendation is that, for maximum accuracy of prediction *in silico* models should be

generated with RF/APDP. RF is compute-intensive in the model-building phase for large data sets, taking perhaps several hours for the largest data sets, even using a parallelized version of RF and the less demanding APDP descriptors. Where model updates are done every few weeks or months, this is not a limitation. RF also has other advantages. One is that it is more flexible than PLPNB in that one may make either quantitative predictions or classifications.

One can imagine a situation in which speed in building a model is important (e.g., for very large numbers of molecules and very large numbers of descriptors); then PLPNB would be more attractive.

■ ASSOCIATED CONTENT

Supporting Information

Distribution of pairwise similarities in the training sets. Distributions of similarity to the nearest compound in the training set for test sets. Pairwise probabilities of methods being the same for bootstrap resampling of the target set. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: sheridan@merck.com.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank Joseph Shpungin for parallelizing random forest so that it can handle very large data sets. Dr. Hongwu Wang developed the "two-model" method for multicategory PLPNB. The QSAR infrastructure used in this work depends on the MIX modeling infrastructure, and the authors are grateful to other members of the MIX team. A large number of Merck biologists, over many years, generated the data for examples used in this paper.

■ REFERENCES

(1) Randic, M. Retro-regression—another important multivariate regression improvement. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 602−606.
(2) Livingstone, D. J.; Salt, D. W. Judging the significance of multiple linear regression models. *J. Med. Chem.* **2005**, *48*, 661−663.
(3) Kaneko, H.; Arakawa, M.; Funatsu, K. Development of a new regression analysis method using independent component analysis. *J. Chem. Inf. Model.* **2008**, *48*, 534−541.
(4) Wold, S.; Sjöström, M.; Eriksson, L. PLS-regression: a basic tool of chemometrics. *Chemom. Intell. Lab. Syst.* **2001**, *58*, 109−130.
(5) Vogt, M.; Bajorath, J. Bayesian similarity searching in high-dimensional descriptor spaces combined with Kullback-Leibler descriptor divergence analysis. *J. Chem. Inf. Model* **2008**, *48*, 247−255.
(6) Klon, A. E.; Lowrie, J. F.; Diller, D. J. Improved naïve Bayesian modeling of numeric data for absorption, distribution, metabolism, and excretion (ADME) property prediction. *J. Chem. Inf. Model.* **2006**, *46*, 1945−1956.
(7) Zhang, H. The optimality of Naive Bayes. FLAIRS 2004 conferernce. http://www.cs.unb.ca/profs/hzhang/publications/FLAIRS04ZhangH.pdf (accessed March 2, 2012).
(8) Agrafiotis, D. K.; Cedeno, W.; Lobanov, V. S. On the use of neural network ensembles in QSAR and QSPR. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 903−911.
(9) Mosier, P. D.; Jurs, P. C. QSAR/QSPR studies using probabilistic neural networks and generalized regression neural networks. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1460−1470.
(10) Burden, F. R.; Winkler, D. A. Robust QSAR models using Bayesian regularized neural networks. *J. Med. Chem.* **1999**, *42*, 3183−3187.
(11) Breiman, L. Random forests. *Machine Learning* **2001**, *45*, 5−32.
(12) 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.
(13) Vapnik, V. *Statistical Learning Theory*; Wiley-WHC: New York, 1998.
(14) Warmuth, M. K.; Liao, J.; Ratsch, G.; Mathieson, M.; Putta, S.; Lemmen, C. Active learning with support vector machines in the drug discovery process. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 667−673.
(15) Obrezanova, O.; Segall, M. D. Gaussian processes for classification: QSAR modeling of ADMET and target activity. *J. Chem. Inf. Model.* **2010**, *50*, 1053−1061.
(16) Svetnick, V.; Wang, T.; Tong, C.; Liaw, A.; Sheridan, R. P.; Song, Q. Boosting: an ensemble learning tool for compound classification and QSAR modeling. *J. Chem. Inf. Model.* **2005**, *45*, 786−799.
(17) Bruce, C. L.; Melville, J. L.; Picket, S. D.; Hirst, J. D. Contemporary QSAR classifiers compared. *J. Chem. Inf. Model.* **2007**, *47*, 219−227.
(18) Caruana, R.; Niculescu-Mizil, A. An empirical comparison of supervised learning algorithm. In *Proceedings of the 23rd International Conference on Machine Learning, Pittsburg, PA*. ACM: New York, 2006.
(19) Glick, M.; Klon, A. E.; Acklin, P.; Davies, J. W. Enriching extremely noisy high-throughput screening data using a naive Bayes classifier. *J. Biomol. Screening* **2004**, *9*, 32−36.
(20) Nidhi; Glick, M.; Davies, J. W.; Jenkins, J. L. Prediction of biological targets for compounds using multiple-category Bayesian models trained on chemogenomics databases. *J. Chem. Inf. Model.* **2006**, *46*, 1124−1133.
(21) Paolini, G. V.; Shapland, R. H. B.; van Hoorn, W. P.; Mason, J. S.; Hopkins, A. L. Global mapping of pharmacological space. *Nat. Biotechnol.* **2006**, *24*, 805−815.
(22) Rogers, D.; Brown, R. D.; Hahn, M. Using extended-connectivity fingerprints with Laplacian-modified Bayesian analysis in high throughput screening follow-up. *J. Biomol. Screening* **2005**, *7*, 682−686.
(23) Xia, X.; Maliski, E. G.; Gallant, P.; Rogers, D. Classification of kinase inhibitors using a Bayesian model. *J. Med. Chem.* **2004**, *47*, 4463−4470.
(24) Nantasenamat, C.; Isaranjura-Na-Ayudha, C.; Prachayasittikul, V. Advances in computational methods to predict the biological activity of compounds. *Expert. Opin.Drug. Discov.* **2010**, *5*, 633−654.
(25) Sprous, D. G.; Palmer, R. K.; Swanson, J. T.; Lawless, M. QSAR in the pharmaceutical research setting: QSAR models for broad, large problems. *Curr. Top. Med. Chem.* **2010**, *10*, 619−637.
(26) Michielan, L.; Moro, S. Pharmaceutical perspectives of nonlinear QSAR strategies. *J. Chem. Inf. Model.* **2010**, *50*, 961−978.
(27) 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.
(28) Kearsley, S. K.; Sallamack, S.; Fluder, E. M.; Andose, J. D.; Mosley, R. T.; Sheridan, R. P. Chemical similarity using physiochemical property descriptors. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 118−27.
(29) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(30) Heikamp, K.; Bajorath, J. Large-scale similarity search profiling of ChEMBL compound data sets. *J. Chem. Inf. Model.* **2011**, *51*, 1831−1839.
(31) Hert, J.; Willet, P.; Wilton, D. J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. Comparison of topological descriptors for similarity-based virtual screening using multiple bioactive reference structures. *Org. Biomol. Chem.* **2004**, *2*, 3256−3266.
(32) Bender, A.; Jenkins, J. L.; Scheiber, J.; Sukaru, S. C. K.; Glick, M.; Davies, J. W. How similar are similarity searching methods? A

principal component analysis of molecular descriptor space. *J. Chem. Inf. Model.* **2009**, *49*, 108−119.
(33) Arif, S. M.; Holliday, J. D.; Willett, P. Analysis and use of fragment-occurrence data in similarity-based virtual screening. *J. Comput.-Aided Mol. Des.* **2009**, *23*, 655−668.
(34) Golbraikh, A.; Tropsha, A. Predictive QSAR modeling based on diversity sampling of experimental datasets for the training and test set selection. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 357−369.
(35) Leonard, J. T.; Roy, K. On selection of training and test sets for the development of predictive QSAR models. *QSAR Comb. Sci.* **2006**, *25*, 235−251.
(36) Sheridan, R. P.; Feuston, B. P.; Maiorov, V. N.; Kearsley, S. K. Similarity to molecules in the training set is a good discriminator for prediction accuracy in QSAR. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1912−1928.
(37) Lombardo, F.; Obach, R. S.; Shalaeva, M. Y.; Gao, F. Prediction of human volume of distribution values for neutral and basic drugs. 2. Extended data seta and leave-class-out statistics. *J. Med. Chem.* **2004**, *47*, 1242−1250.
(38) Sheridan, R. P. Alternative global goodness metrics and sensitivity analysis: heuristics to check the robustness of conclusions from studies comparing virtual screening methods. *J. Chem. Inf. Model.* **2008**, *48*, 426−433.
(39) Prathipati, P.; Ma, N. L.; Keller, T. H. Global Bayesian models for the prioritization of antitubercular agents. *J. Chem. Inf. Model.* **2008**, *48*, 2362−2370.
(40) 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.