

Open Source Bayesian Models. 2. Mining a “Big Dataset” To Create and Validate Models with ChEMBL

Alex M. Clark^{*,†} and Sean Ekins^{*,‡,§,||}

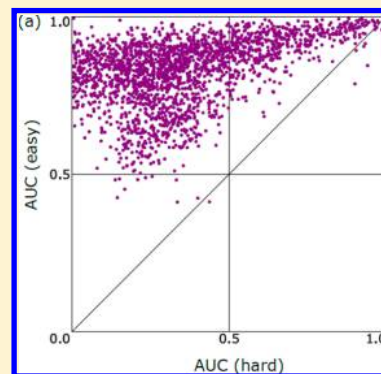
[†]Molecular Materials Informatics, Inc., 1900 St. Jacques No. 302, Montreal H3J 2S1, Quebec, Canada

[‡]Collaborations Pharmaceuticals, Inc., 5616 Hilltop Needmore Road, Fuquay-Varina, North Carolina 27526, United States

[§]Collaborations in Chemistry, 5616 Hilltop Needmore Road, Fuquay-Varina, North Carolina 27526, United States

^{||}Collaborative Drug Discovery, 1633 Bayshore Highway, Suite 342, Burlingame, California 94010, United States

ABSTRACT: In an associated paper, we have described a reference implementation of Laplacian-corrected naïve Bayesian model building using extended connectivity (ECFP)- and molecular function class fingerprints of maximum diameter 6 (FCFP)-type fingerprints. As a follow-up, we have now undertaken a large-scale validation study in order to ensure that the technique generalizes to a broad variety of drug discovery datasets. To achieve this, we have used the ChEMBL (version 20) database and split it into more than 2000 separate datasets, each of which consists of compounds and measurements with the same target and activity measurement. In order to test these datasets with the two-state Bayesian classification, we developed an automated algorithm for detecting a suitable threshold for active/inactive designation, which we applied to all collections. With these datasets, we were able to establish that our Bayesian model implementation is effective for the large majority of cases, and we were able to quantify the impact of fingerprint folding on the receiver operator curve cross-validation metrics. We were also able to study the impact that the choice of training/testing set partitioning has on the resulting recall rates. The datasets have been made publicly available to be downloaded, along with the corresponding model data files, which can be used in conjunction with the CDK and several mobile apps. We have also explored some novel visualization methods which leverage the structural origins of the ECFP/FCFP fingerprints to attribute regions of a molecule responsible for positive and negative contributions to activity. The ability to score molecules across thousands of relevant datasets across organisms also may help to access desirable and undesirable off-target effects as well as suggest potential targets for compounds derived from phenotypic screens.



INTRODUCTION

Until recently, the general paradigm has been that a scientist (perhaps a computational chemist or medicinal chemist) would generate or find a dataset of interest relating to his or her project and build a computational model that could be used to score additional molecules prior to testing. Compared to a decade ago, we are now experiencing a time in which public biological data are in abundance in various databases. PubChem,^{1–3} and the mandate of publishing NIH-funded experimental data into this database, has been a major contributor to this shift, followed by ChEMBL,^{4–6} as well as commercial vendors with public datasets like CDD.⁷ This creates an opportunity for data mining to expand how we approach drug discovery from a single target or disease centric approach to one which covers potentially thousands of targets and organisms for which we have accessible data.

For example, our work in the area of tuberculosis using datasets derived from phenotypic high-throughput screening (HTS);^{8–11} the hit rate of these screens tends to be in the low single digits. We and others have leveraged these data to produce validated computational models^{12–17} with prospective testing hit rates >20% with low or no cytotoxicity.^{18,19} We have also combined datasets to use all 350,000 molecules with *in vitro*

data from a single laboratory for computational models to produce “bigger data” models.²⁰ Building and validating these models for a single organism was time-consuming and focused on comparing Bayesian to other machine learning approaches. None of these models was made freely accessible until recently.²¹ If we are to scale this type of approach to many other targets and organisms, the model building has to be simplified or pre-computed. Similarly, if one were to consider the hundreds of studies that have been published describing computational models for absorption, distribution, metabolism excretion, and toxicity (ADME/Tox), physicochemical properties, or for that matter target endpoints, these are also unlikely to be accessible to anyone but those who generated them. This is clearly undesirable if we are to rapidly build on the work of others. The only way to possibly change this (if it is desired) is to make the creation of open access models relatively easy²² or to create a repository of validated models that can be accessed by different software platforms.

The focus of our study is Laplacian-corrected naïve Bayesian models (which we call Bayesian models for simplicity), which

Received: March 13, 2015

Published: May 21, 2015

we^{18–20,23–35} and many others^{36–53} have used and validated versus other machine learning methods. Bayesian models also have been a valuable part of the computer-aided drug discovery toolkit and commercially accessible in Pipeline Pilot.^{43,45,54} In the accompanying article,²¹ we have described implementation of Bayesian models with molecular function class fingerprints of maximum diameter 6 (FCFP6) descriptors that could be run in open source software using CDK components. We now evaluate bit folding (into shortened bit strings, which is a trade-off between performance and efficacy) and model cutoff selection for cross-validation. We have also tested the scalability of the Bayesian models with extended connectivity (ECFP6) and FCFP6 descriptors using the ChEMBL collection (version 20). This involved assembling thousands of models by grouping together targets and measurement types and selecting an activity cutoff threshold for each case. This extensive collection has allowed us to prepare a comprehensive analysis of how well the models can be expected to work in realistic drug discovery scenarios and to draw attention to edge cases.

In the process of this work, we have made this very large repository of computational models open access and freely available to the community. The datasets we have extracted from ChEMBL are compiled with the primary objective of providing a large array of real-world drug discovery datasets for the purpose of testing and validating algorithms, but the methods we have developed could easily be adapted to provide supporting data for prospective projects. While there are resources such as qsar.db.org,⁵⁵ ochem.eu,⁵⁶ and Chembench,⁵⁷ these do not currently house the number of models that are possible when extracting a large subset from public databases like ChEMBL. This perhaps may lead to changing the way we may think of doing drug discovery to study more than one target or organism simultaneously, implying more of a chemogenomics approach.^{42,58–64}

We have previously described how the Bayesian models using ECFP6 fingerprints can be implemented in a mobile app called TB Mobile to enable *Mycobacterium tuberculosis* target prediction as a way to de-orphan compounds derived from phenotypic screening.⁶⁵ We now expand on how Bayesian models implemented in other cheminformatics software or apps can be used to attribute regions of a molecule responsible for positive and negative contributions to activity. Such models could be used in a mobile app or desktop cheminformatics workflows.^{66,67}

■ EXPERIMENTAL SECTION

Data and Materials Availability. Graphical representations of the ChEMBL-extracted subsets and their partitioning thresholds, the raw datasets, and the derived Bayesian models are available at <http://molsync.com/bayesian2>, and may be used without restriction.

Naïve Bayesian Definition and Pseudocode. The CDK codebase for creating Bayesian models with ECFP6 and FCFP6 descriptors is in the latest version on Github (<http://github.com/cdk/cdk>: look for the tools section, for class `org.openscience.cdk.fingerprint.model.Bayesian`). The use of this has recently been described.²¹

Bit Folding. The method we are describing has been implemented with two molecular fingerprint schemes in mind: ECFP6 and FCFP6, specifically the open source reference implementations that we have described previously. Each of these methods takes a molecular structure as input and returns a list of 32-bit integers. This means that there is a diabolical

case in which a molecule could return a dense list of 4 billion fingerprints, but in practice these are sparse, and seldom exceed 100. Building a Bayesian model involves assembling the union of all these fingerprint bits during the process of working out the contributions for each of them, but even for models with hundreds of thousands of molecules, there are typically rarely more than several thousand unique fingerprint hash codes, though of course this varies considerably depending on the nature of the source data.

Because individual fingerprint bits are sparse for individual molecules and for whole datasets, the fingerprint hash codes are typically stored as arrays of sorted integers, which is most efficient for storage purposes and also facilitates speedy calculations of other types (e.g., Tanimoto coefficients). Despite the efficiency of storing sparse fingerprints, there are nonetheless legitimate reasons for exploring the option of further collapsing the size, which can be done by *folding* the fingerprints. The ECFP6 and FCFP6 methods internally fold the fingerprint hash codes into 32 bits, but this range can be further reduced, e.g., to 16 bits by taking the modulo of 65,536 (bitwise *and* 0xFFFF). This reduces the discriminatory power of the fingerprints, which has the potential to reduce the predictivity of the model. For example, if two different structural features are represented by fingerprints 0xABCD1234 and 0x56781234, respectively, the folded 16-bit hash codes are now both represented by 0x1234. The more aggressively the fingerprints are folded, the fewer options the Bayesian model has to distinguish between important structural differences. Empirically, however, it can be observed that the full 32-bit range of the original fingerprints is excessive. Reducing to a smaller range has some advantages: the resulting Bayesian models can be stored in a smaller serialized file, and having an operational range that is small enough to fit into a flat array can improve performance (i.e., array size is 2^{bits} , which means that allocating more than 4 billion placeholders for the original 32 bits is unrealistic, but using half as many bits or less is quite practical).

Cross-Validation. One of the most popular metrics for evaluating the performance of these kinds of Bayesian models is to plot the receiver operator characteristic (ROC) curve. Creating this graph involves selecting a series of thresholds, and for each threshold, molecules for which the raw prediction value is greater are considered to be predicted active. For each threshold the number of true positives, true negatives, false positives, and false negatives can be determined. The ROC curve is created by plotting the ratio of true positives on the Y-axis and false positives on the X-axis. Both axes are scaled from 0 to 1. A favorable result is one for which the Y value increases rapidly for small values of X and quickly reaches a plateau, ideally for values of $Y = 1$. The entire curve can be reduced to a single number by integrated the area under the curve (AUC). An AUC value of 1 indicates a perfect model, while a value of less than 0.5 indicates that it is antipredictive.

The method for generating the raw predictions with which the ROC curve is built requires a decision about how to partition the training data. Typically the data are segregated into N different partitions, and for each partition, the molecules not in that partition are used to rebuild the model, and molecules that *are* in the partition are used as a test set, and their predicted values are stored. This means that N different models are created, and each molecule is used as a testing datum once; the values are all combined together once the process is complete.

The *leave-one-out* partitioning is the boundary extreme: for N molecules in the input set, N different Bayesian models are created, each with $N - 1$ entries. This approach is seldom used, partly because it has undesirable performance characteristics (it scales as $O(N^2)$ with regard to the number of input molecules) and partly because the resulting metrics tend to err on the side of generosity.

More commonly the dataset is split into three or five partitions, which typically performs well, since the time taken is only several times longer than the original model-building operation, which is already efficient and runs in linear time. Because each of these segmentations involves splitting the dataset into two parts for which both parts are of size greater than one, the choice of partitions can influence the metrics. For the implementation we describe, the ordering issue is dealt with crudely: the input molecules are kept in their original order, except that actives are listed first, followed by inactives. They are then split evenly into equal sized partitions.

This guarantees that each of the partitions has an equal number of actives and inactives (± 1), but it must be kept in mind that the validation metrics are technically *order specific*; e.g., if a five-fold split was requested and the dataset was constructed so that every fifth molecule was extremely similar to the molecule five positions previous, it could cause the validation metric to underperform, since the training/test sets are modeling on maximally different structural classes. For cases where stability of metrics is a significant issue, this problem could be solved by sorting the input molecules according to a canonical ordering system.

Calculation of the ROC curve points and the corresponding integral can be done efficiently once the cross-validation predictions have been computed for each compound. A simple and efficient algorithm is overall $O(N)$, except for a single sorting operation, which is presumed $O(N \log N)$. The first step is to compile a sorted list of threshold values that can be used to partition the dataset:

```

let E = cross-validation prediction for each molecule
let O = index order for E
let T = empty array
append T, min(E) - 0.01 × range(E)
for i = 1, number of molecules - 1:
  let t1 = E[O[i]], t2 = E[O[i+1]]
  if t1 = t2: skip loop
  if threshold ≠ last T:
    append T, 0.5 × (t1 + t2)
append T, max(E) + 0.01 × range(E)

```

The resulting list of threshold values has a maximum length of one greater than the number of molecules. The first and last values are guaranteed to include all and none of the compounds, respectively. The following step is to calculate x and y values for each of these thresholds, which is done by keeping a running total of true and false positives:

```

let PT = 0, PF = 0
let Rx, Ry, Rt = empty arrays
for threshold in T; i = 1:
  while i ≤ number of molecules:
    if threshold < E[O[i]]: break loop
    if is_active(O[i]):
      increment PT
    else
      increment PF

```

```

increment i
let x = PF ÷ number of inactives
let y = PT ÷ number of actives
if x = last Rx and y = last Ry: skip loop
append Rx, x; Ry, y; Rt, Ti

```

At the end of this step, the arrays R_x and R_y contain an ordered list of values from 0 to 1 that can be used to graphically plot the ROC curve. The R_t array contains the corresponding threshold value for each of these points, which is not necessary for plotting or integral calculation but is used subsequently in the calibration step. Note that in both of these steps, each of the arrays has a maximum capacity that is known before it is populated, which can be used to further optimize performance.

Calculating the integral is a trivial matter of adding up the areas of the rectangles:

```

let AUC = 0
for i = 1, length(RxRy) - 1:
  let w = Rx[i+1] - Rx[i]
  let h = Ry[i+1] - Ry[i]
  let AUC = AUC + w × h

```

Calibration. One of the main caveats with the Laplacian-modified Bayesian method that we have chosen is that the raw prediction values have no absolute meaning. In some modeling scenarios this is not an issue, e.g., if the predicted values are used to correlate against some other property, or to rank proposed compounds. In many cases, however, it is desirable to achieve an outcome that is directly compared to the input activities, which are binary classifications (active/inactive). If a suitable *threshold* can be decided upon, the software can report on which predictions are above or below the threshold. If a suitable *range* can be found, the predictions can be scaled to a range that can be used in lieu of an actual probability, i.e., a value of 0 (or less) is definitively inactive, 1 (or more) is definitively active, and values closer to 0.5 are somewhat indeterminate.

Selection of a threshold is complicated by the fact that its ideal value is application dependent: if the cost of a false positive is much greater than the cost of a false negative, then a higher threshold is better, and vice versa. These conditions are not known to a fully automated algorithm, and so it is useful to compute a proposed prediction-to-probability transform, that the user may optionally choose to apply to the raw output.

Simply put, the threshold is chosen as the location on the ROC curve where the value of $Y - X$ is greatest. This is the point at which the true positives/negatives are optimally balanced against their false analogs. It assumes that false positives and false negatives (as ratios) are both equally undesirable. This is simple to calculate, since the ROC curve is created out of a series of tentative thresholds, each of which already has the X - and Y -values computed for it:

```

let Rx, Ry, Rt = ROC data (left-to-right)
let best = index for which Ry[i] - Rx[i] is highest
let mid = Rt[best]
let i1 = 1, i2 = length (Rx, Ry)
while i1 < best - 1:
  if Rx[i1] > 0: break loop
  increment i1
while i2 > best + 1:
  if Ry[i2] < 1: break loop
  decrement i2
let delta = min(Rt[i1] - mid, mid - Rt[i2])
let low = mid - delta, high = mid + delta

```


Note that the ROC data from the previous step were calculated in right-to-left order, but for the above pseudocode, the order has been reversed. The first step is to find the “best” point on the ROC curve, where $Y - X$ is greater than for any other threshold, which is demonstrated in Figure 1. This is

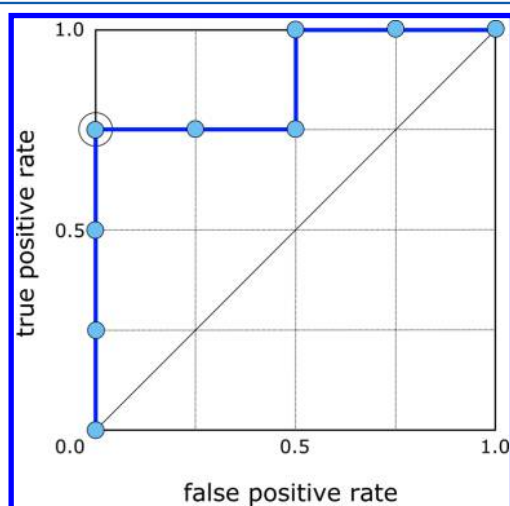


Figure 1. ROC curve for the trivial example with just eight compounds in the training set.

taken to be the midpoint. The upper and lower bounds are taken to be the highest and lowest threshold, decremented/incremented as necessary so that the position no longer touches

the edge of the graph (i.e., $X = 0$ and $Y = 1$, respectively). The shortest distance between these two thresholds and the middle threshold is used to compute a *low* and *high* threshold, and for calibration purposes, these are used to mark $P = 0$ and $P = 1$ when predictions are scaled to a probability-like range.

This range can then be used to transform predictions so that *most* molecules with fingerprint schemes that fall well within the range of the model give calibrated results between 0 and 1. While the calibration should not be used as a probability in the strict statistical sense, it provides some degree of comparability between different models and makes them easier for scientists to interpret:

$$\text{calibrated} = \frac{\text{raw prediction} - \text{low}}{\text{high} - \text{low}}$$

Unlike an actual probability value, calibrated results can occur outside of the 0–1 range. This occurs increasingly often when models built on small datasets are applied to molecules that are not well within the training domain.

Worked Example. Figure 2 shows the source materials for a very simple model-building example. The structures for the first four rows are used as inactives for the training set, while the next four rows are actives for the training set. As can be seen, the actives vs inactives differ only in that the actives have a nitrogen atom substituted for one of the secondary or tertiary carbon atoms. The last two molecules are used as test set examples.

In each case, ECFP6 fingerprints have been generated for the chemical structure and folded into a sub-range of 64









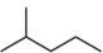
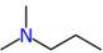
	structure	activity	5 15 19 24 30 37 48 61														prediction	calibrated	
			3	14	18	21	29	33	44	51									
1		inactive																-0.511	-1.79
2		inactive																-0.847	-5.52
3		inactive																-1.764	-15.67
4		inactive																-0.847	-5.52
5		active																0.575	10.25
6		active																1.163	16.76
7		active																0.786	12.58
8		active																0.891	13.75
9		unknown																-1.070	-7.99
10		unknown																0.380	8.09

Figure 2. Source materials for a trivial example. The training set consists of four inactive and four active compounds. For each case, the unique fingerprint bits, folded into a range of 0–63, are shown, along with raw and calibrated predictions from the model.

fingerprints, i.e., a reduction from the original 32 bits down to 6 bits. For example, the original fingerprints for row 1 have the integer hash codes -1027418143, -801752141, -9543903, and 790592664. Upon reduction to 6 bits, the unique hash codes are 24, 33, and 51. Note that the number of fingerprint hash codes goes from four to three, due to the collisions caused by the reduction of precision. Normally this degree of folding would be unadvisable, but for brevity purposes it is effective for a small and easy-to-model example.

In Figure 2, each fingerprint that was observed in at least one of the eight molecules found in the training set is represented with a column, each of which is a set with 16 members: 3, 5, 14, 15, 18, 19, 21, 24, 29, 30, 33, 37, 44, 48, 51, and 61.

Creating the model is illustrated in Table 1: for each unique fingerprint bit, the number of actives containing the bit (*A*) and

Table 1. Calculation of Bayesian Model Contributions for Each Unique Fingerprint

fingerprint bit	A	T	$[A + 1]/[T \cdot R + 1]$	log
3	1	1	1.333	0.288
5	0	1	0.667	-0.405
14	2	5	0.857	-0.154
15	3	3	1.600	0.470
18	2	2	1.500	0.405
19	1	1	1.333	0.288
21	1	1	1.333	0.288
24	1	3	0.800	-0.223
29	3	4	1.333	0.288
30	1	1	1.333	0.288
33	2	6	0.750	-0.288
37	1	1	1.333	0.288
44	0	1	0.667	-0.405
48	0	2	0.500	-0.693
51	4	8	1.000	0.000
61	1	2	1.000	0.000

total number of molecules containing the bit (*T*) are tabulated. The ratio is calculated, where the denominator includes the constant *R*, which is the overall proportion of actives, which in this case is $4/8 = 0.5$. The natural log of the ratio is shown in the last column, which is the *contribution* that corresponds to the presence of the bit.

Obtaining the raw prediction from the Bayesian model is done by accumulating the contributions for each fingerprint. For example, the structure shown in row 9 has folded fingerprints 3, 14, 24, 27, 31, 33, 38, 48, 51, and 61. The raw prediction from the model is $0.288 - 0.154 - 0.223 - 0.288 - 0.693 + 0 + 0 = -1.070$. Note that three of the fingerprints

(27, 31, and 38) are not present in the model, so they do not contribute to the prediction score.

Note that the test set examples (rows 9 and 10) share many of these same fingerprints, but they have additional fingerprints that do not appear in the training set (for row 9, 27, 31, and 58; for row 10, 4 and 11).

Producing internal validation metrics involves repeating the Bayesian model creation some number of times, with a certain portion of the training set temporarily designated as the test set. For the “leave-one-out” approach, eight models are created, each with seven entries in the training set and one in the test set. In this example, the leave-one-out predictions for each of the rows, respectively, are -0.320, -0.342, -0.622, -0.407, -0.357, 1.088, -0.288, -0.006.

In order to prepare the data for the ROC, the midpoints between the sorted list of internal validation predictions can be used as postulated thresholds, with additional values above and below the limits. For each of these putative thresholds, it is possible to tabulate the true/false positives and negatives, as shown in Table 2. The values of *Y* and *X* are shown as the proportion of *true* and *false* positives, respectively. These values are shown graphically in Figure 2.

Integrating the area under the curve gives a value of 0.875. If it were possible to identify a threshold for which all active compounds are above and all inactive compounds are below, the integral would be 1.0, but in practice this is usually only observed for very small datasets.

Determination of the calibration begins by selecting the point on the ROC curve at which the value of $Y - X$ is greatest, which in this case is the point for which threshold = -0.304, $X = 0$, and $Y = 0.75$. Because it provides the best performance in terms of distinguishing between correct and incorrect predictions, this threshold is mapped to a calibrated value of 0.5, so that when calibrated predictions are treated as probability-like values, this is the threshold at which a prediction will be rounded up or down into the *active* or *inactive* categories. In order to find the range, two threshold points are selected: the upper bound being -0.147, and the lower bound being -0.350. The upper bound is selected as the next-threshold-up from the midpoint, since the midpoint itself corresponds to a threshold with zero false positives (i.e., the value of *X* is 0). The lower bound is the lowest threshold for which a false positive occurs (i.e., the position on the curve no longer corresponds to $Y = 1$). The shortest distance between the midpoint threshold and either of the boundaries is 0.046, and so the transform is chosen such that raw Bayesian predictions between -0.350 and -0.258 correspond to calibrated “probability-like” values of 0 and 1.

As can be seen in the last column of Figure 1, the *calibrated* values do not much resemble probabilities in terms of their

Table 2. Computed Data Used To Build the ROC Curve, Using a Trivial Example Dataset of Eight Compounds

threshold	true positives	false positives	true negatives	false negatives	Y	X
-0.622 - ϵ	4	4	0	0	1.0	1.0
-0.514	4	3	1	0	1.0	0.75
-0.382	4	2	2	0	1.0	0.5
-0.350	3	2	2	1	0.75	0.5
-0.326	3	1	3	1	0.75	0.25
-0.304	3	0	4	0	0.75	0.0
-0.147	2	0	4	2	0.5	0.0
1.041	1	0	4	3	0.25	0.0
1.088 + ϵ	0	0	4	4	0.0	0.0

scale, which is an artifact of using a very small dataset as a pedagogical example. However, all of the calibrated predictions for inactive molecules score less than 0.5, and inactives greater than 0.5, so in terms of binary classification the model provides correct results for all 8 items in the training set. Additionally, it can be observed that the last two structures, which are labeled as having *unknown* activity, provide calibrated predictions that indicate inactive for the hydrocarbon (row 9) and active for the amine (row 10), which is the same structure–activity relationship that is unambiguously encoded within the training set.

Performance and Scalability. The building of the model is $O(N)$ with regard to the number of molecules in the training set. For purposes of building the raw model, if the overall ratio of actives to inactives is known, only a single pass through the database is required. This makes it favorable for integration into stream-like workflows, since rows can be considered separately as transient objects that do not need to be stored after they have been accumulated into the respective totals. The rate-limiting step for model building is typically the I/O involved with fetching a molecule object (e.g., reading it from an MDL SDfile) and generating the corresponding fingerprints.

For fingerprint calculation, the ECFP6 and FCFP6 schemes to which we have limited our analysis are both fast to compute. They require only the molecular connection table graph typically associated with a 2D structure (e.g., an MDL Molfile with minimal interpretation, a SMILES string that has been upconverted into a Molfile-like data structure, or any other related format). Besides obtaining the parity of chiral centers, the geometry is irrelevant, so there is no need to perform conformation embedding, partial charge calculation, or any other potentially lengthy preparation step. The fingerprint generation is $O(N)$ with regard to the number of atoms. The method is specific to small molecules with heavy atom counts rarely exceeding a hundred, so for overall model-building purposes this can be considered as a constant.

Because the actual model building consists of incrementing counters for fingerprint indices, which are independent and not order dependent, it would be straightforward to split the process into multiple threads.

By making use of dictionary objects, the sparseness of the distribution of the fingerprint hashcodes is balanced against the cost of looking up the hash codes within the rate limiting step of the model building. Since the sizes of these dictionaries is bounded above by the number of unique fingerprints, the $O(\log N)$ lookup time for most dictionary implementations (which typically use hash table lookup methods) is reasonable. However, for highly performance sensitive implementations that use *folded* fingerprint hash codes, it may be possible to squeeze additional performance by using a flat array instead of a dictionary, though the benefits are unlikely to be significant unless the performance cost of I/O and fingerprint generation has been removed (e.g., by pre-loading and pre-computing).

While creation of the raw Bayesian model can be done in a single pass without keeping the training set in memory for the duration, it should be noted that for many use cases it is appropriate to analyze the results by performing an internal cross-validation study. Since the validation is functionally equivalent to rebuilding the model several times, it means that it is necessary to either store the fingerprints in memory until the validation is complete or be able to re-iterate over the data collection multiple times.

RESULTS

Training Data. In order to evaluate the model-building method, it was decided to compile a large quantity of real-world drug discovery data, using the ChEMBL collection.^{4–6} Because this curated database has a large number of well-defined targets in a structured form, it is possible to extract thousands of individual collections, with comparable assay measurements against the same target. By subjecting the model creation process to numerous automated tests, it is possible to gain an idea of how robust the method is, as well as ensuring that it operates adequately on a number of realistic edge cases.

Extracting the data that make up ChEMBL involves a number of steps, which begin with setting up a MySQL database and inloading ChEMBL release 20 as an SQL data dump, as described in the documentation. Once the data are loaded, the objective is to compose criteria that can be used to formulate an SQL query that combines each of the targets and assays in such a way that multiple submission blocks can be combined, as long as they are measuring the same property for the same target.

The list of assay:target entities can be obtained with the following query:

```
SELECT DISTINCT
    t.tid, a.assay_id, target_type, t.pref_name,
    t.organism, y.assay_desc
FROM target_dictionary t, assays a, assay_type y
WHERE t.tid=a.tid AND a.assay_type=y.assay_type
    AND (target_type='SINGLE PROTEIN' OR
    target_type='ORGANISM' OR target_type='CELL-LINE' OR
    target_type='PROTEIN COMPLEX' OR target_type='TISSUE'
    OR target_type='PROTEIN FAMILY')
ORDER BY tid, assay_id
```

From this list is obtained a set of entries featuring target and assay keys, with target type, name and organism, and assay type. For subsequent processing, each group of entries for which all properties were identical except for the assay keys was considered to represent a potential dataset, compiled by grouping together structure–activity records corresponding to any of the activity keys. For each block of activities, the individual activity records were compiled with the query template:

```
SELECT * FROM activities WHERE
    (assay_id = a1 OR assay_id = a2 OR ...)
AND standard_value>0
AND (standard_units='M' OR standard_units='mM' OR
    standard_units='uM' OR standard_units='nM')
AND (standard_type='GI50' OR standard_type='IC50' OR
    standard_type='Ki' OR standard_type='EC50' OR
    standard_type='AC50' OR standard_type='Kd')
AND (standard_relation='=' OR standard_relation='<' OR
    standard_relation='>' OR standard_relation='<=' OR
    standard_relation='>=')
```

The first limit clause restricts the query to any of the assay identifiers for the block, which varies from one to thousands. Common ADME properties are excluded from consideration in this exercise by skipping targets that have more than 100,000 assays, since the objective is to procure a large number of small to medium size activity measurements with some diversity across the drug development spectrum. The results are further winnowed down by restricting the units and measurement types. Combined with the target type restriction in the previous

query, these constraints no doubt exclude some viable datasets, but for validation purposes, there is plenty still remaining.

Once the results are obtained for the preliminary subset, any instances where the number of activity measurements is less than 100 is skipped. The resulting activity values are analyzed and converted to $-\log$ measurements (i.e., pK_i , pIC_{50} , etc.). The *molregno* column in the result set is used to identify individual compounds. In cases where the same molecule occurs with more than one measurement, the measurements are reconciled if possible: if multiple equality measurements occur, they are averaged; if two inequality measurements are degenerate, then the most limiting is taken (e.g., for <5 is preferred over <6); or if two measurements are incompatible (e.g., <3 and >4 , or <6 and $=7$), then the compound is excluded from the set. Once the analysis is complete, if the reduction of degeneracy/inadmissible results has not reduced the dataset size to below 100 rows, it is retained. Each of the molecule records is cross-referenced to an MDL Molfile representation with the query:

```
SELECT molfile FROM compound_structures WHERE molregno=...
```

The resulting collection is converted into a molecular datasheet, with columns for molecular structure, ChEMBL ID (*molregno*), relation (one of $=$, $<$, or $>$), and activity ($-\log_{10}$ of concentration in mol L^{-1}). The target name, organism, type, assay, and measurement information is stored in the datasheet header.

In this way, 2152 datasets were obtained, each with at least 100 activity measurements, for which the largest had just under 10,000 molecules.

Cutoff Determination. Since the Bayesian models work on classification of *active* versus *inactive*, and the extracted datasets contain continuous physical properties, it is necessary to select a cutoff value to apply to the assay measurement, above which a compound is considered active. Unfortunately, selection of an appropriate cutoff depends on the target, the available data, and the objectives of the model. For many projects, a threshold of 6 (corresponding to $1 \mu\text{M}$ or less for inhibition) is a sensible default, but in cases where very few compounds would qualify as actives, a lower threshold may be preferable, or for projects where there are a number of very strong inhibitors and the objective is to identify compounds that meet stringent criteria, a higher threshold is appropriate.

As well as balancing the active/inactive ratio, there are other criteria that are emergent from the data: all other things being equal, it is preferable to pick a threshold in a region with few nearby data points. Since all measurements are subject to experimental error, reducing the number that are close to the threshold reduces the likelihood of misclassification.

Another consideration is that the choice of threshold may do well to take into account the way that the *structures* are partitioned: if certain structural features that are in actual fact beneficial to activity are mostly on the *active* side of the threshold, then the ensuing model will have increased ability to identify structure–activity relationships than it would if a larger proportion of the discriminatory fingerprints were on either side of the threshold. As it happens, the most effective way to determine how well a threshold is able to discriminate between SAR features is to actually build the model for a sampling of putative thresholds. This can be done efficiently for larger datasets by taking a representative subset, building a number of test models, and using the ROC integral as an indication of partitioning efficacy.

Figure 3 shows six selected examples from the whole collection, representing cytochrome P450 2C9, cytochrome P450 2D6,⁶⁸ bromodomain-containing protein 4,⁶⁹ *M. tuberculosis* enoyl-acyl-carrier protein (InhA),⁷⁰ hERG,^{27,71} and tyrosine protein kinase ABL.^{72,73} These are all examples of targets or off-targets. Each example shows an overall population distribution with activity on the X-axis (as $-\log_{10}$ of concentration in mol L^{-1}). Each activity value has been accumulated as a Gaussian distribution for specific values, or a rectangle for ranges (e.g., <3 , >6 , etc.), each with an area of 1. Most regions of the graph are smooth, which facilitates numerical calculation of the second derivative, which is used to identify regional minima which, all other things being equal, are often good places to impose a cutoff threshold.

For datasets that are not already small, a sampled subset of the collection (e.g., 100 molecules) is extracted in a way that favors structural diversity as well as a distribution of activity values that is similar to the whole dataset. For this subset, each of the unique interstitial midpoints between activity values is considered as a possible cutoff threshold and evaluated by composing a desirability score made up of the following:

1. the ROC integral from a Bayesian model using the subset of molecules and the threshold for partitioning into active/inactive (higher is better)
2. the second derivative of the population, interpolated from the current threshold (lower is better)
3. the ratio of actives to inactives, if the whole collection were partitioned according to the threshold: $(\text{actives}+1)/(\text{inactives}+1)$ or its reciprocal, whichever is greater

These three terms favor thresholds that separate effectively according to structure–activity patterns (1), separate in terms of experimentally determined activity (2), and balance actives versus inactives (3). While the notion of including the ROC integral for a modeled subset in the determination of an ideal threshold may seem like a circular argument, it should be kept in mind that the ultimate objective is to find the best way to classify molecules on the basis of their structure–activity relationship, and building sample models is a crude but effective way to measure this. The relative weightings of these contributions to the overall desirability score may need to be adjusted according to the project, but it can be seen in Figure 3 that when each is scaled such that they are equally significant, it works well for a number of cases. Graphics and data are available for *all* of the datasets at <http://molsync.com/bayesian2>, and can be conveniently viewed with any contemporary Web browser.

This method is primarily useful for preparing example datasets that are too numerous to individually research in order to come up with a scientifically justifiable rationale for locating an activity threshold for each one. Nonetheless, it may find some utility in practical drug discovery projects, when a good choice of threshold is unclear, or when it is useful to know how the choice of threshold affects the predictive power of the model. We have found this method to be highly effective for generating a large number of realistic datasets for evaluating our Bayesian model-building implementation, and we are currently considering other applications.

Statistics. Each of the datasets extracted from ChEMBL was used to build Bayesian models using the activity threshold as described above, with several different parameters: for each of ECFP6 and FCFP6 fingerprint types, folding sizes from 64

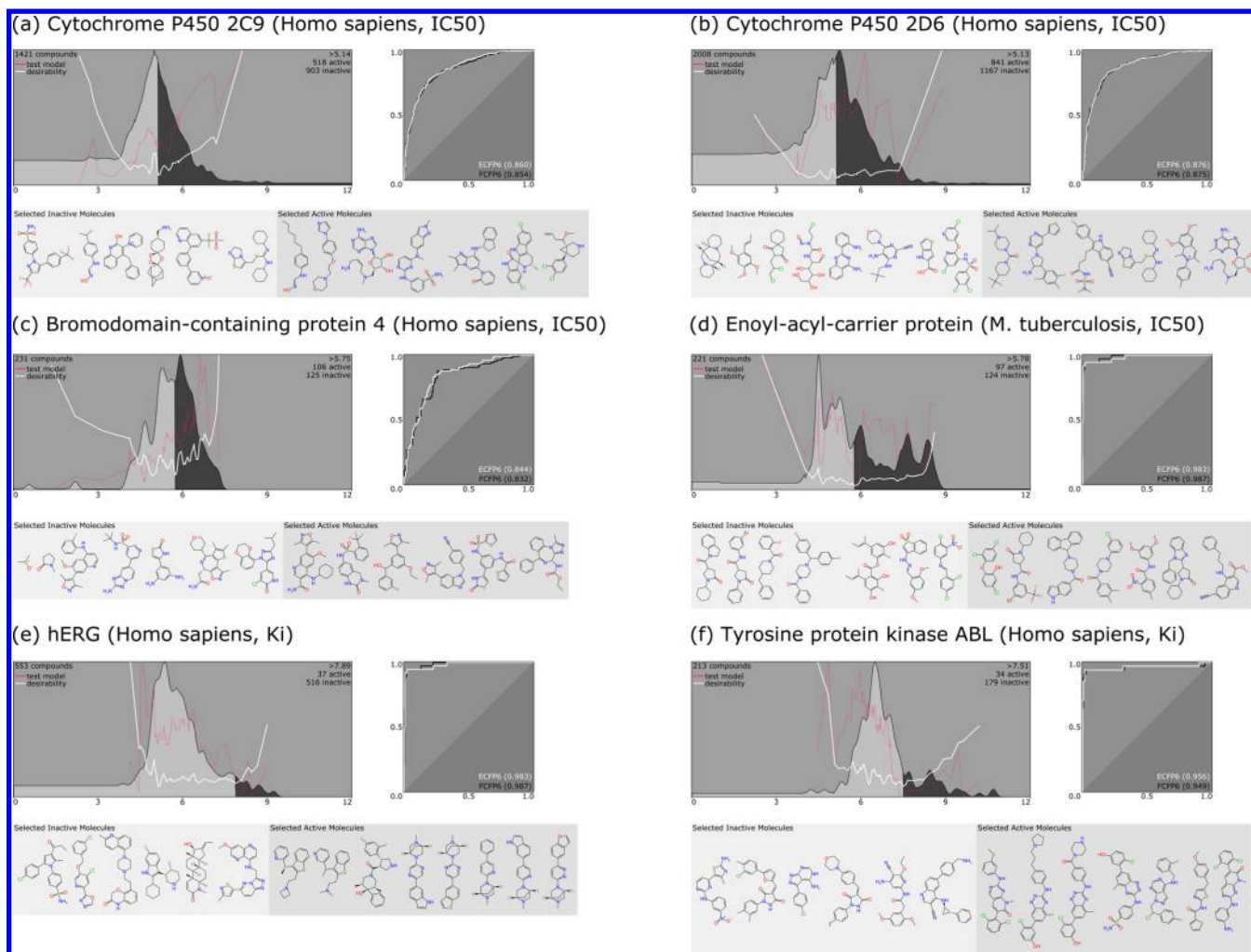


Figure 3. Selected examples of extracted datasets from ChEMBL and the analysis leading to the detection of a suitable activity threshold. Each example shows a plot of population versus activity, for which the solid curve shows the integral, which is colored to show inactives (below threshold, light gray) and actives (above threshold, dark gray). The ROC integral for subset models at various thresholds is plotted, as is the overall desirability composite score. To the right is shown the ROC curve for ECFP6 and FCFP6 models, built using the whole dataset at the determined threshold. A representative diverse selection of “active” and “inactive” molecules is shown underneath.

through 65,536 were used, as well as no folding (which corresponds to 2^{32} possible unique fingerprints).

Figure 4 shows the relationship between increasing the range of the bits with each of the two fingerprinting schemes. The distribution, mean, and standard deviation of the ROC integrals (with five-fold cross-validation) are plotted on the Y-axis. When bits are folded into 64 categories, the average performance is 0.741 for ECFP6 and 0.740 for FCFP6. As should be expected, the best performance is achieved with no extra fingerprint folding (ROC integrals of 0.830 and 0.825, respectively), but as can be observed, the responses attain somewhat of a plateau at around 4096. For practical purposes, if storage size of models is an issue, reducing the folding to 1024 results in relatively little degradation, but any further reduction is generally inadvisable.

In general, the cross-validation metrics for the sample datasets perform well, as ideally the ROC value should be greater than 0.70, as determined by cross-validation metrics. In order to simulate the boundaries for how well the fingerprints work in a more practical scenario, each dataset was subsequently re-examined by partitioning into two halves, each with the same number of actives and inactives. Two different partitioning

strategies were applied to each dataset in order to split it into *training* and *testing sets*:

- *balanced*: The actives and inactives were selected so that the training and testing sets were most similar to one another, achieving a split that reflected a training set with a very similar structure–activity trend to the testing set.
- *diabolical*: The actives and inactives were split so that the training and testing sets formed two distinct clusters with as little structural commonality as possible, achieving a split that simulates a testing set that originates from a completely different domain of structural features.

The algorithm for splitting between training/testing sets for balanced/diabolical extremes is crude but effective. ECFP6 fingerprints are used, with the Tanimoto coefficient as the similarity metric:

- seed the training set with the **inactive** structure with greatest average similarity to all other inactive structures
- iterate until all **inactive** compounds have been designated as *training* or *testing*:
 - add one inactive compound to the **testing** set: if preparing the *diabolical* case, select the inactive

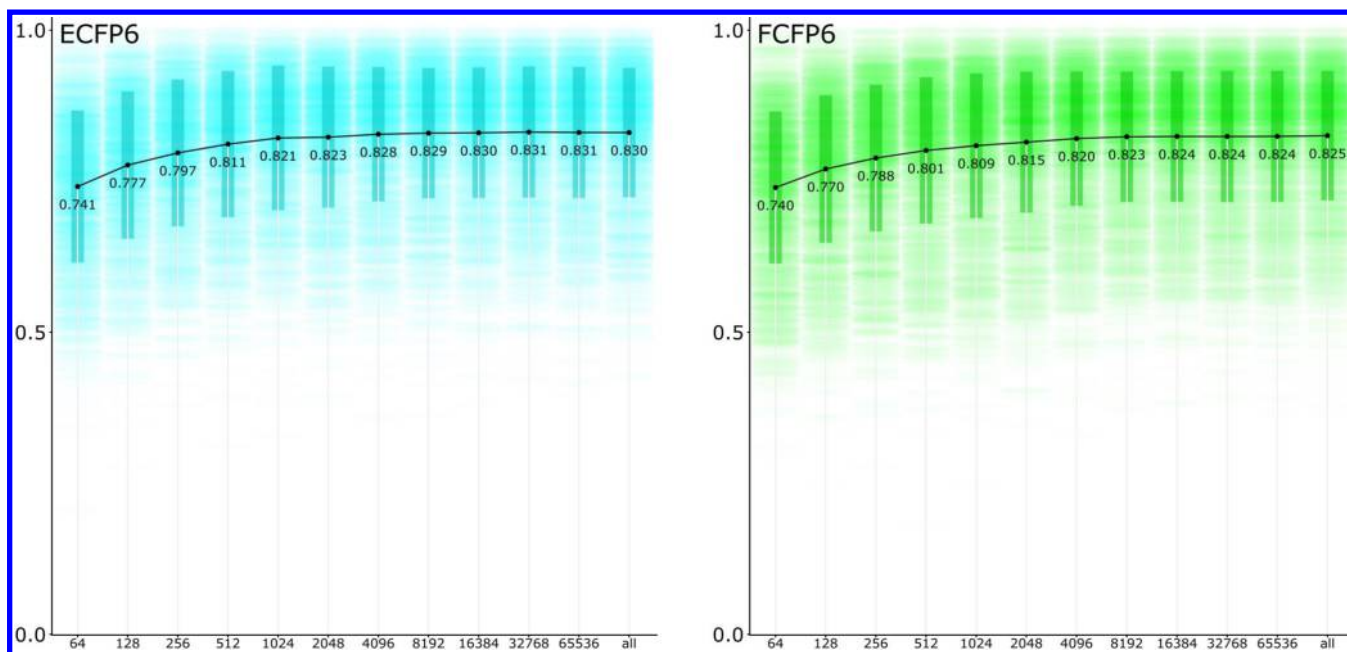


Figure 4. Effect of folding on ECFP6 and FCFP6 fingerprints. ROC integrals are plotted on the Y-axis, for a series of folding sizes. The average ROC integral is shown, with the standard deviation as a solid bar.

- compound with the **lowest** similarity to inactive compounds already in the training set; if preparing the *balanced* case, select the inactive compound with the **highest** similarity
 - add to the **training** set the inactive compound with the **highest** similarity to inactive compounds already in the training set
- iterate until all **active** compounds have been designated as *training* or *testing*:
 - add one active compound to the **training** set: if preparing the *diabolical* case, select the compound most similar to inactives in the testing set; if preparing the *balanced* case, select the compound most similar to the inactives in the training set
 - add to the **testing** set the active compound with the **highest** similarity to the inactives in the training set

In practice, it may be necessary to cap the number of similarity comparisons in order to analyze all datasets within a reasonable length of time, since the formal scalability is $O(N^3)$. Limiting the number of comparisons means that for larger datasets the partitioning will not be optimally balanced/optimally diabolical, but the trend is nonetheless strongly pronounced.

The net effect of these two partitioning schemes is that the balanced case creates training and testing sets for which the structural similarity patterns should be very similar in both cases, and so a model created for one should be highly applicable to the other. It would be expected that an ROC curve derived from external validation should lead to similar efficacy as for cross-validation metrics. The diabolical case, on the other hand, is contrived to search for two divergent *clusters* of dissimilarity among the inactive population and carry this trend through to the active population. While in some structurally homogeneous datasets the partitioning method has little effect, and in larger datasets there are sometimes enough

interlocking structure–activity trends they partially mitigate it, the difference is very clear.

Figure 5 shows that in aggregate the difference in partitioning effect is profound. The models were constructed using ECFP6 fingerprints (with no folding), and the testing sets were used exclusively to derive the ROC curve (as opposed to the progressive method usually used for cross-validation). Overall, the balanced “easy” cases gave rise to an average ROC integral of 0.831 ± 0.107 , which is essentially identical to the results obtained by performing five-fold cross-validation with the original sets (0.830 ± 0.105). The diabolical “hard” cases yielded an average of 0.387 ± 0.229 , which confirms the result that would be expected: building a training set based on molecules with as little as possible structural commonality with the test set results in predictions that are even worse than random, and for small datasets are usually antipredictive. The difference between the two is directly illustrated in Figure 5a, which indicates that all but a handful of the balanced cases give ROC integrals well over 0.5, whereas for the unbalanced cases the results vary significantly, with most of them being very poor. Figure 5b,c shows the relationship between the ROC integrals of these two classes plotted against the size of the original dataset. It can be seen that as the dataset size grows the variation decreases, which is also to be expected, since the homogeneity and existence of genuine structure–activity trends varies tremendously within the smaller collections.

Deployment. At the time of writing, the Bayesian model creation and use is a very recent addition to the CDK project, and there are as yet no user-facing software packages capable of making use of the new functionality, although work is underway.²¹ It is, however, possible to import Bayesian models constructed using the ECFP6 fingerprinting scheme into the Mobile Molecular DataSheet (MMDS)⁶⁶ app. Figure 6 illustrates an example where several of the Bayesian models derived from ChEMBL subsets are imported into the app (Figure 6a–d). Once models have been imported, they can be used to calculate predictions for collections of molecules

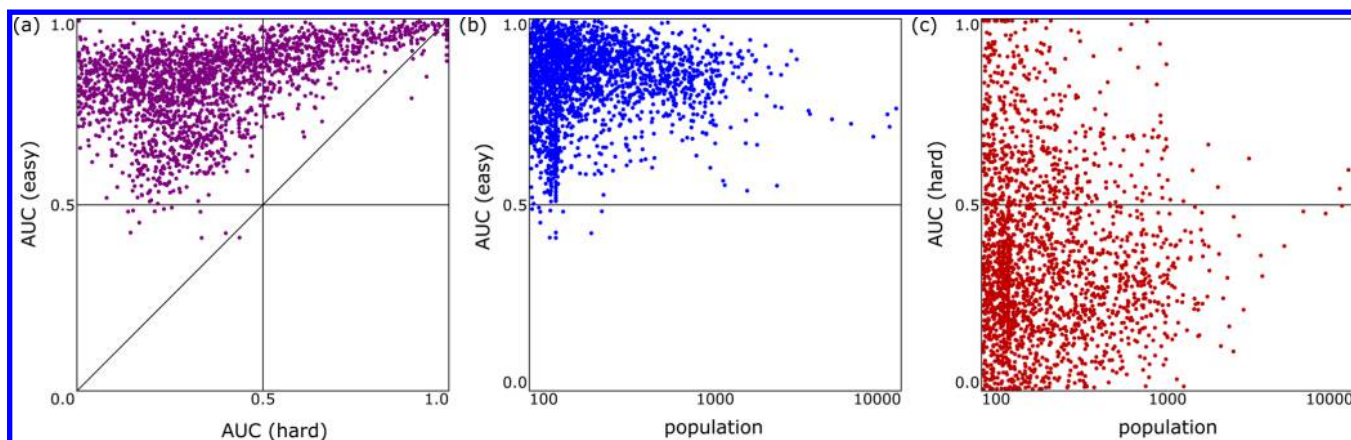


Figure 5. Effects of training set selection. (a) ROC integrals for the generously balanced “easy” set (Y-axis) vs the diabolically partitioned “hard” set (X-axis). (b,c) ROC integrals for the easy and hard sets vs population size of the original training set.

(Figure 6e–g). This can be carried out locally within the mobile app, since the model is stored, and the app has the ability to calculate the same ECFP6 fingerprints as provided by the CDK library. Given that the fingerprint calculation is already available, applying predictions from a pre-existing model is straightforward.

Figure 6h shows the visualization of a single proposed structure (representing a compound with whole cell activity against *M. tuberculosis*¹⁹) which has been evaluated according to several imported Bayesian models. In addition to displaying the calibrated “probability-like” numeric prediction, the app also presents a color-coded structure, where green and red are used to indicate parts of the molecule responsible for high or low probability of activity, respectively. The color-coding is obtained by re-deriving each of the ECFP6 fingerprints and intercepting the atom indices that were responsible for the generation of the particular fingerprint index. Since the fingerprint index corresponds to a contribution from the Bayesian model, it can be normalized and summed across the atoms associated with the fingerprint. The end result is a relative value for each of the atoms, which can be used to draw attention to favorable and unfavorable structural fragments. We are actively pursuing more ambitious and rigorous visualization methods for exploiting the fact that the ingredients of the Bayesian models can be backtracked to individual atoms and fragments from the constituent molecules.

DISCUSSION

A considerable number of previous studies have used ECFP or FCFP fingerprints with the Bayesian algorithm.^{18–20,23–49,51,52,74–76} For example, ECFP4 fingerprints with the Laplacian-modified naïve Bayesian classifier were used to build models for each target and target family from reporter gene screens, in order to create frequent hitter models.⁷⁷ Paolini et al. in 2006 used a Bayesian classifier approach to create 698 target specific models (over 200,000 molecules and 561,000 measurements)⁷⁸ using <10 μ M as the activity cutoff. These models were then used to predict the Cerep Bioprint data. Models could predict intra-gene family interactions with high confidence.⁷⁸ Costache et al. used Bayesian models with FCFP6 fingerprints to model drug likeness, and the model was used to score an amine library.⁷⁴ Similarity ensemble analysis (SEA) was described by Keiser et al.,⁷⁹ using 246 targets and 65,241 molecules and comparing the Tanimoto similarity for each pair of molecules. This approach

was used to identify new targets for several known drugs that were not expected. SEA has since been implemented online and uses ECFP6 fingerprints. The same group used this approach with an earlier version of ChEMBL using over 2000 targets and 167,000 molecules. This enabled identification of targets for compounds from a zebrafish phenotypic screen.⁸⁰ Bender et al.⁴¹ used a set of 70 targets and 100,269 data points, with a cutoff of 10 μ M for IC₅₀ or K_i. A multicategory Bayesian model with ECFP4 fingerprints was used to predict pre-clinical pharmacology. These models had correct classification statistics of 94%. The world drug index was also used to build models for adverse drug reactions these, in turn had a classification accuracy of 91.7%. Riniker et al. used 93 in-house and 95 PubChem datasets to build machine learning models with HTS fingerprints and chemical fingerprints.⁸¹ ECFP4 fingerprints were used with random forest, logistic regression, and naïve Bayesian. Naïve Bayesian performed the worst out of the three algorithms.⁸¹ It was, however, shown that combining HTS fingerprints and ECFP4 fingerprints may be useful for scaffold hopping. Balfour and Bajorath⁷⁶ recently described methods to visualize and graphically interpret Bayesian classification models using log odds ratios for model features which are then viewed on a circular visualization. Features with at least 90% odds ratios were back projected onto test compounds using red and green for negative and positive. This group tested their approach on three datasets.⁷⁶ These examples illustrate the diversity of uses for these fingerprints with the Bayesian algorithm.

It is likely that profiling “big datasets” is also going to be increasingly the norm.^{82,83} For example a recent study mined public bioassay (PubChem) datasets for compounds that have rat *in vivo* acute toxicity data, an approach that could be used in other big data initiatives like ToxCast and Tox21.⁸⁴ The use of tools for mining and visualizing large chemistry related datasets will be important.⁸⁵ For example the latest version of the ToxCast database as of December 2014 has over 1800 molecules⁸⁶ tested against over 800 assays or endpoints. This dataset could be readily used as a further example of using the approach described herein, except in this case with a very complete data matrix. Such a dataset could be very useful for profiling similar compounds in a manner analogous to the many compound and microarray profiling databases, e.g., the Connectivity Map,⁸⁷ which uses genome-wide transcriptional expression data from cultured human cells that were treated with various bioactive small molecules alongside the use of simple pattern-matching algorithms to allow connections to be

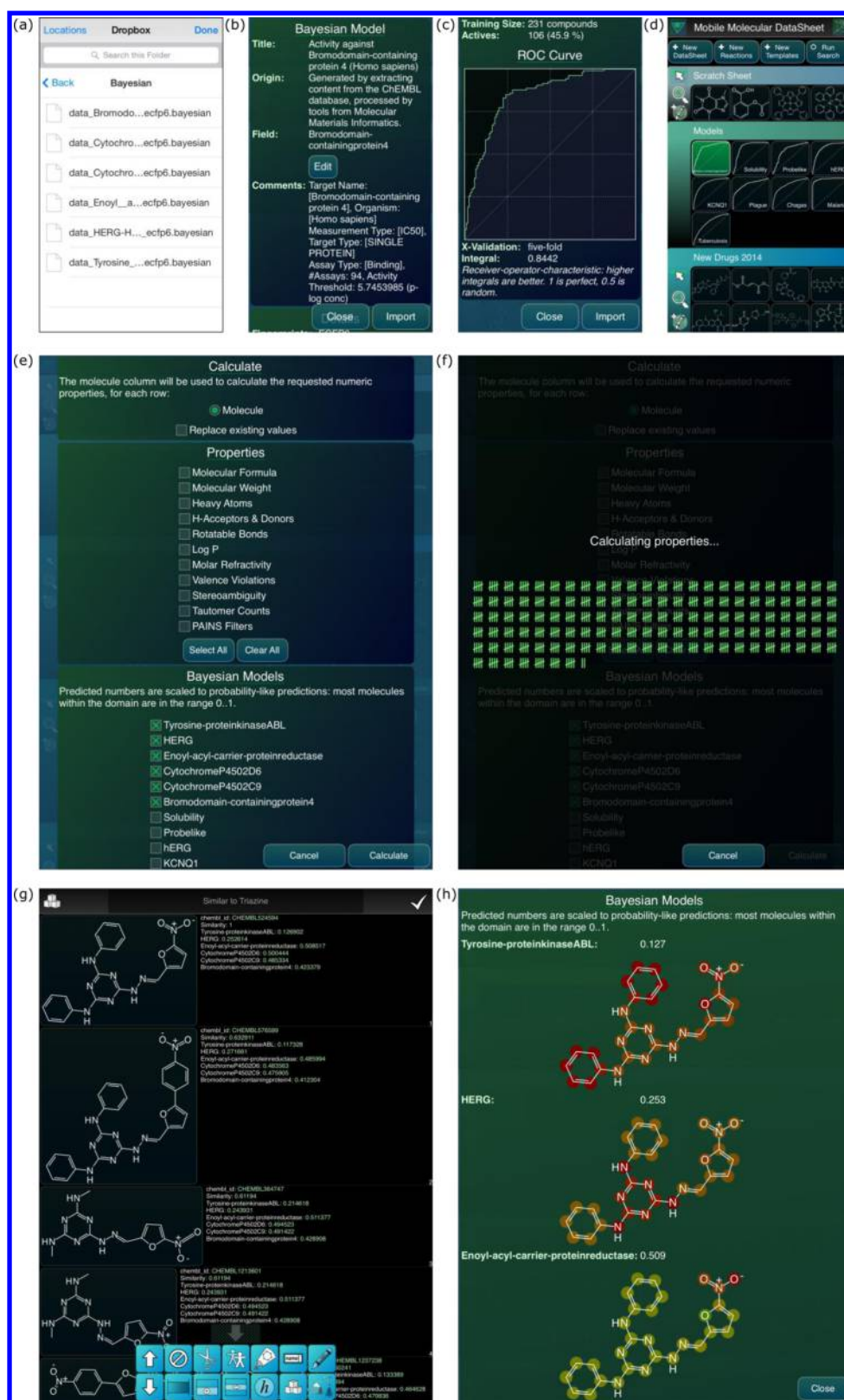


Figure 6. (a–d) Importing selected Bayesian models from ChEMBL-derived data into the Mobile Molecular DataSheet (MMDS) app. (e–g) Using models to make predictions for a collection of structures. (h) Viewing predictions for one structure in MMDS: atoms with high positive or negative contributions to the model are colored green or red, respectively, with intermediate cases in yellow.

made among molecules, genes, and diseases via gene-expression changes. However, one caveat of the ToxCast dataset is the heavy emphasis on environmental related chemicals (90%) compared with small molecule drugs.⁸⁶

In the accompanying paper, we have described using our Bayesian algorithm and fingerprint implementation to build tens of models in the CDD database or in a mobile app.²¹ We have now applied this model-building approach to building and

validating over 2000 models as well as determining the optimum bit folding. In the process of this study we have evaluated how we can make models informative and yet compact. We have automated the process of cutoff selection, and we have made thousands of validated Bayesian models openly available. Future work that leverages the current study could include extension to larger datasets in ChEMBL, as well as applying to all datasets in PubChem. As many of these datasets are from screens of hundreds of thousands of compounds in order to identify new chemical probes, the scale of these models is somewhat larger. An array of such models could complement the over 2000 we have created in this study. One could also imagine such datasets could be used for parallel profiling of compounds in a manner complementary to the SEA analysis.^{79,80} This could enable a potential approach to predict targets for compounds derived from phenotypic screens. In this study we have not addressed ADME datasets in ChEMBL, as the selection of cut offs where there are different units (like for intrinsic clearance or Caco-2 permeability) may need some more careful consideration.

Recent papers on open source software for data management, drug discovery, and target prediction include DiSCUS for virtual screening data management⁸⁸ and iDrug, a Web-accessible and interactive drug discovery and design platform⁸⁹ that uses pharmacophore mapping in crystal structures and reverse mapping to predict targets for molecules. The approach of a model repository which could be used for target identification and virtual screening could certainly mesh with these efforts. It represents an opportunity to run these models in software such as the CDK or even mobile apps like MMDS which can upload the model. This enables anyone to benefit from the open sharing of such models. Of course in a pharmaceutical or biotech company one could imagine models derived from proprietary data would only need to be shared in house or with close collaborators. In this case, secure collaborative software would be used to transfer and run the model.⁷ Once such models are made available they can be used for a variety of applications from drug repurposing⁹⁰ to off-target profiling for toxicity assessment^{41,58,91} or suggesting future studies to run.

CONCLUSION

In part 1 we introduced an open source implementation of a successful Bayesian modeling schema,²¹ and in part 2 we have followed up this work with a thorough description of the cross-validation analysis algorithms, calibration, and methods for studying the efficacy on real-world datasets. The public availability of high quality curated collections of bioactivity data such as ChEMBL are an invaluable resource to users of computer-aided drug discovery methods. Nonetheless, converting such data sources into usable validation sets involves additional effort, starting with examining the database schema in order to decide how to group comparable assays, which we have described. For purposes of two-state Bayesian classification, a threshold is needed in order to convert continuous assay measurements, and we have invented a method for automated classification that is more effective than picking an arbitrary cutoff for all datasets, and more economical than manually investigating thousands of cases and applying expert judgment. In the future we intend to continue to develop this threshold-determination technique so that it is applicable to prospective discovery campaigns, but in the interim we have found it to be very useful for bulk evaluation of modeling

techniques. As well as describing the underlying algorithms, we have also made the data available, and we encourage other developers to make use of them.

In this article we have mainly focused on the validation of a specific modeling technique, but we have carried out these studies with the intention of exploring new ways to leverage model quantity as well as quality in this “big data” age. By being able to easily apply hundreds of activity models to any given molecule, it may be possible to accelerate serendipitous discovery of multipurpose (or repurposed) drugs, as well as provide an array of alerts with regard to undesirable ADME/tox properties. By making use of public data and open source modeling methods, the opportunity exists to make this functionality available to a broad range of researchers, rather than being limited to well-funded scientists within large pharmaceutical companies. For this reason, we expect the greatest impact to be made within the realm of neglected and rare disease research, which is relatively underserved and culturally more attuned to generously sharing data and models.^{17,92}

AUTHOR INFORMATION

Corresponding Authors

*E-mail (A.M.C.): aclark@molmatinf.com.

*E-mail (S.E.): ekinssean@yahoo.com. Phone: (215) 687-1320.

Author Contributions

A.M.C. developed all software and models; S.E. selected models for discussion; both authors wrote the manuscript.

Notes

The authors declare the following competing financial interest(s): S.E. is a consultant for Collaborative Drug Discovery Inc. A.M.C. is the founder of Molecular Materials Informatics, Inc.

ACKNOWLEDGMENTS

S.E. and A.M.C. kindly acknowledge many valuable discussions with Antony J. Williams on sharing models and mobile apps for chemistry and Dr. Joel S. Freundlich on drug discovery. This project was not funded but benefited from the open source software supported by Award No. 9R44TR000942-02, “Biocomputation across distributed private datasets to enhance drug discovery”, from the NIH National Center for Advancing Translational Sciences.

ABBREVIATIONS

ADME/Tox, absorption, distribution, metabolism, excretion, and toxicity; AUC, area under the curve; CDD, Collaborative Drug Discovery; CDK, Chemistry Development Kit; ECFP6, extended connectivity; FCFP6, molecular function class fingerprints of maximum diameter 6; hERG, human ether-a-go-go related gene; HTS, high-throughput screening; MMDS, Mobile Molecular DataSheet; ONS, Open Notebook Science; QSAR, quantitative structure–activity relationship; ROC, receiver operator curve

REFERENCES

- (1) Li, Q.; Cheng, T.; Wang, Y.; Bryant, S. H. PubChem as a public resource for drug discovery. *Drug Discov. Today* **2010**, *15*, 1052–1057.
- (2) Wang, Y.; Xiao, J.; Suzek, T. O.; Zhang, J.; Wang, J.; Bryant, S. H. PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic Acids Res.* **2009**, *37*, W623–W633.
- (3) Wang, Y.; Bolton, E.; Dracheva, S.; Karapetyan, K.; Shoemaker, B. A.; Suzek, T. O.; Wang, J.; Xiao, J.; Zhang, J.; Bryant, S. H. An

overview of the PubChem BioAssay resource. *Nucleic Acids Res.* **2010**, *38*, D255–D266.

(4) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, *40*, D1100–D1107.

(5) Bento, A. P.; Gaulton, A.; Hersey, A.; Bellis, L. J.; Chambers, J.; Davies, M.; Kruger, F. A.; Light, Y.; Mak, L.; McGlinchey, S.; Nowotka, M.; Papadatos, G.; Santos, R.; Overington, J. P. The ChEMBL bioactivity database: an update. *Nucleic Acids Res.* **2014**, *42*, D1083–D1090.

(6) Papadatos, G.; Overington, J. P. The ChEMBL database: a taster for medicinal chemists. *Future Med. Chem.* **2014**, *6*, 361–364.

(7) Ekins, S.; Bunin, B. A. The Collaborative Drug Discovery (CDD) database. *Methods Mol. Biol.* **2013**, *993*, 139–154.

(8) Balle, L.; Field, R. A.; Duncan, K.; Young, R. J. New small-molecule synthetic antimycobacterials. *Antimicrob. Agents Chemother.* **2005**, *49*, 2153–2163.

(9) Reynolds, R. C.; Ananthan, S.; Faaleolea, E.; Hobrath, J. V.; Kwong, C. D.; Maddox, C.; Rasmussen, L.; Sosa, M. I.; Thammasuvimol, E.; White, E. L.; Zhang, W.; Secrist, J. A., III. High throughput screening of a library based on kinase inhibitor scaffolds against *Mycobacterium tuberculosis* H37Rv. *Tuberculosis (Edinb.)* **2012**, *92*, 72–83.

(10) Maddry, J. A.; Ananthan, S.; Goldman, R. C.; Hobrath, J. V.; Kwong, C. D.; Maddox, C.; Rasmussen, L.; Reynolds, R. C.; Secrist, J. A., III; Sosa, M. I.; White, E. L.; Zhang, W. Antituberculosis activity of the molecular libraries screening center network library. *Tuberculosis (Edinb.)* **2009**, *89*, 354–363.

(11) Ananthan, S.; Faaleolea, E. R.; Goldman, R. C.; Hobrath, J. V.; Kwong, C. D.; Laughon, B. E.; Maddry, J. A.; Mehta, A.; Rasmussen, L.; Reynolds, R. C.; Secrist, J. A., III; Shindo, N.; Showe, D. N.; Sosa, M. I.; Suling, W. J.; White, E. L. High-throughput screening for inhibitors of *Mycobacterium tuberculosis* H37Rv. *Tuberculosis (Edinb.)* **2009**, *89*, 334–353.

(12) Prakash, O.; Ghosh, I. Developing an antituberculosis compounds database and data mining in the search of a motif responsible for the activity of a diverse class of antituberculosis agents. *J. Chem. Inf. Model.* **2006**, *46*, 17–23.

(13) Garcia-Garcia, A.; Galvez, J.; de Julian-Ortiz, J. V.; Garcia-Domenech, R.; Munoz, C.; Guna, R.; Borrás, R. Search of chemical scaffolds for novel antituberculosis agents. *J. Biomol. Screen.* **2005**, *10*, 206–214.

(14) Planche, A. S.; Scotti, M. T.; Lopez, A. G.; de Paulo Emerenciano, V.; Perez, E. M.; Uriarte, E. Design of novel antituberculosis compounds using graph-theoretical and substructural approaches. *Mol. Divers.* **2009**, *13*, 445–458.

(15) Sundaramurthi, J. C.; Brindha, S.; Reddy, T. B.; Hanna, L. E. Informatics resources for tuberculosis—towards drug discovery. *Tuberculosis (Edinb.)* **2012**, *92*, 133–138.

(16) Ekins, S.; Freundlich, J. S.; Choi, I.; Sarker, M.; Talcott, C. Computational Databases, Pathway and Cheminformatics Tools for Tuberculosis Drug Discovery. *Trends Microbiol.* **2011**, *19*, 65–74.

(17) Ekins, S.; Freundlich, J. S. Computational models for tuberculosis drug discovery. *Methods Mol. Biol.* **2013**, *993*, 245–262.

(18) Ekins, S.; Reynolds, R. C.; Franzblau, S. G.; Wan, B.; Freundlich, J. S.; Bunin, B. A. Enhancing Hit Identification in *Mycobacterium tuberculosis* Drug Discovery Using Validated Dual-Event Bayesian Models. *PLoS One* **2013**, *8*, No. e63240.

(19) Ekins, S.; Reynolds, R.; Kim, H.; Koo, M.-S.; Ekonomidis, M.; Talaue, M.; Paget, S. D.; Woolhiser, L. K.; Lenaerts, A. J.; Bunin, B. A.; Connell, N.; Freundlich, J. S. Bayesian Models Leveraging Bioactivity and Cytotoxicity Information for Drug Discovery. *Chem. Biol.* **2013**, *20*, 370–378.

(20) Ekins, S.; Freundlich, J. S.; Reynolds, R. C. Are bigger data sets better for machine learning? Fusing single-point and dual-event dose response data for *Mycobacterium tuberculosis*. *J. Chem. Inf. Model.* **2014**, *54*, 2157–2165.

(21) Clark, A. M.; Dole, K.; Coulon-Spector, A.; McNutt, A.; Grass, G.; Freundlich, J. S.; Reynolds, R. C.; Ekins, S. Open Source Bayesian Models. 1. Application to ADME/Tox and Drug Discovery Datasets. *J. Chem. Inf. Model.* **2015**, DOI: 10.1021/acs.jcim.5b00143, (preceding paper in this issue).

(22) Ekins, S.; Williams, A. J. Precompetitive Preclinical ADME/Tox Data: Set It Free on The Web to Facilitate Computational Model Building to Assist Drug Development. *Lab Chip* **2010**, *10*, 13–22.

(23) Litterman, N. K.; Lipinski, C. A.; Bunin, B. A.; Ekins, S. Computational Prediction and Validation of an Expert's Evaluation of Chemical Probes. *J. Chem. Inf. Model.* **2014**, *54*, 2996–3004.

(24) Ekins, S.; Pottorf, R.; Reynolds, R. C.; Williams, A. J.; Clark, A. M.; Freundlich, J. S. Looking back to the future: predicting in vivo efficacy of small molecules versus *Mycobacterium tuberculosis*. *J. Chem. Inf. Model.* **2014**, *54*, 1070–1082.

(25) Ekins, S.; Freundlich, J. S.; Hobrath, J. V.; Lucile White, E.; Reynolds, R. C. Combining computational methods for hit to lead optimization in *Mycobacterium tuberculosis* drug discovery. *Pharm. Res.* **2014**, *31*, 414–435.

(26) Ekins, S.; Casey, A. C.; Roberts, D.; Parish, T.; Bunin, B. A. Bayesian models for screening and TB Mobile for target inference with *Mycobacterium tuberculosis*. *Tuberculosis (Edinb.)* **2014**, *94*, 162–169.

(27) Ekins, S. Progress in computational toxicology. *J. Pharmacol. Toxicol. Methods* **2014**, *69*, 115–140.

(28) Dong, Z.; Ekins, S.; Polli, J. E. Structure-activity relationship for FDA approved drugs as inhibitors of the human sodium taurocholate cotransporting polypeptide (NTCP). *Mol. Pharmaceutics* **2013**, *10*, 1008–1019.

(29) Astorga, B.; Ekins, S.; Morales, M.; Wright, S. H. Molecular Determinants of Ligand Selectivity for the Human Multidrug And Toxin Extrusion Proteins, MATE1 and MATE-2K. *J. Pharmacol. Exp. Ther.* **2012**, *341*, 743–55.

(30) Pan, Y.; Li, L.; Kim, G.; Ekins, S.; Wang, H.; Swaan, P. W. Identification and Validation of Novel hPXR Activators Amongst Prescribed Drugs via Ligand-Based Virtual Screening. *Drug Metab. Dispos.* **2011**, *39*, 337–344.

(31) Zientek, M.; Stoner, C.; Ayscue, R.; Klug-McLeod, J.; Jiang, Y.; West, M.; Collins, C.; Ekins, S. Integrated in silico-in vitro strategy for addressing cytochrome P450 3A4 time-dependent inhibition. *Chem. Res. Toxicol.* **2010**, *23*, 664–76.

(32) Ekins, S.; Williams, A. J.; Xu, J. J. A Predictive Ligand-Based Bayesian Model for Human Drug Induced Liver Injury. *Drug Metab. Dispos.* **2010**, *38*, 2302–2308.

(33) Diao, L.; Ekins, S.; Polli, J. E. Quantitative Structure Activity Relationship for Inhibition of Human Organic Cation/Carnitine Transporter. *Mol. Pharmaceutics* **2010**, *7*, 2120–2130.

(34) Zheng, X.; Ekins, S.; Raufman, J. P.; Polli, J. E. Computational models for drug inhibition of the human apical sodium-dependent bile acid transporter. *Mol. Pharmaceutics* **2009**, *6*, 1591–1603.

(35) Ekins, S.; Kortagere, S.; Iyer, M.; Reschly, E. J.; Lill, M. A.; Redinbo, M. R.; Krasowski, M. D. Challenges predicting ligand-receptor interactions of promiscuous proteins: the nuclear receptor PXR. *PLoS Comput. Biol.* **2009**, *5*, No. e1000594.

(36) Wang, S.; Li, Y.; Wang, J.; Chen, L.; Zhang, L.; Yu, H.; Hou, T. ADMET evaluation in drug discovery. 12. Development of binary classification models for prediction of hERG potassium channel blockage. *Mol. Pharmaceutics* **2012**, *9*, 996–1010.

(37) Singh, N.; Chaudhury, S.; Liu, R.; Abdulhameed, M. D.; Tawa, G.; Wallqvist, A. QSAR Classification Model for Antibacterial Compounds and Its Use in Virtual Screening. *J. Chem. Inf. Model.* **2012**, *52*, 2559–2569.

(38) Martins, I. F.; Teixeira, A. L.; Pinheiro, L.; Falcao, A. O. A Bayesian approach to in silico blood-brain barrier penetration modeling. *J. Chem. Inf. Model.* **2012**, *52*, 1686–1697.

(39) Langdon, S. R.; Mulgrew, J.; Paolini, G. V.; van Hoorn, W. P. Predicting cytotoxicity from heterogeneous data sources with Bayesian learning. *J. Cheminform.* **2010**, *2*, 11.

- (40) 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.
- (41) Bender, A.; Scheiber, J.; Glick, M.; Davies, J. W.; Azzaoui, K.; Hamon, J.; Urban, L.; Whitebread, S.; Jenkins, J. L. Analysis of Pharmacology Data and the Prediction of Adverse Drug Reactions and Off-Target Effects from Chemical Structure. *ChemMedChem* **2007**, *2*, 861–873.
- (42) 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.
- (43) Klon, A. E.; Lowrie, J. F.; Diller, D. J. Improved naive Bayesian modeling of numerical data for absorption, distribution, metabolism and excretion (ADME) property prediction. *J. Chem. Inf. Model.* **2006**, *46*, 1945–1956.
- (44) Hassan, M.; Brown, R. D.; Varma-O'Brien, S.; Rogers, D. Cheminformatics analysis and learning in a data pipelining environment. *Mol. Divers.* **2006**, *10*, 283–299.
- (45) Rogers, D.; Brown, R. D.; Hahn, M. Using extended-connectivity fingerprints with Laplacian-modified Bayesian analysis in high-throughput screening follow-up. *J. Biomol. Screen.* **2005**, *10*, 682–686.
- (46) Arimoto, R.; Prasad, M. A.; Gifford, E. M. Development of CYP3A4 inhibition models: comparisons of machine-learning techniques and molecular descriptors. *J. Biomol. Screen.* **2005**, *10*, 197–205.
- (47) Xia, X.; Maliski, E. G.; Gallant, P.; Rogers, D. Classification of kinase inhibitors using a Bayesian model. *J. Med. Chem.* **2004**, *47*, 4463–4470.
- (48) Wang, L.; Le, X.; Li, L.; Ju, Y.; Lin, Z.; Gu, Q.; Xu, J. Discovering new agents active against methicillin-resistant *Staphylococcus aureus* with ligand-based approaches. *J. Chem. Inf. Model.* **2014**, *54*, 3186–3197.
- (49) Fang, J.; Yang, R.; Gao, L.; Zhou, D.; Yang, S.; Liu, A. L.; Du, G. H. Predictions of BuChE inhibitors using support vector machine and naive Bayesian classification techniques in drug discovery. *J. Chem. Inf. Model.* **2013**, *53*, 3009–3020.
- (50) Tian, S.; Wang, J.; Li, Y.; Xu, X.; Hou, T. Drug-likeness analysis of traditional Chinese medicines: prediction of drug-likeness using machine learning approaches. *Mol. Pharmaceutics* **2012**, *9*, 2875–2886.
- (51) Lee, J. H.; Lee, S.; Choi, S. In silico classification of adenosine receptor antagonists using Laplacian-modified naive Bayesian, support vector machine, and recursive partitioning. *J. Mol. Graph. Model.* **2010**, *28*, 883–890.
- (52) Klon, A. E. Bayesian modeling in virtual high throughput screening. *Comb. Chem. High Throughput Screen.* **2009**, *12*, 469–483.
- (53) Schroeter, T. S.; Schwaighofer, A.; Mika, S.; Ter Laak, A.; Suelzle, D.; Ganzer, U.; Heinrich, N.; Muller, K. R. Estimating the domain of applicability for machine learning QSAR models: a study on aqueous solubility of drug discovery molecules. *J. Comput. Aided Mol. Des.* **2007**, *21*, 651–664.
- (54) Chen, B.; Sheridan, R. P.; Hornak, V.; Voigt, J. H. Comparison of random forest and Pipeline Pilot Naive Bayes in prospective QSAR predictions. *J. Chem. Inf. Model.* **2012**, *52*, 792–803.
- (55) Aruoja, V.; Moosus, M.; Kahru, A.; Sihtmaa, M.; Maran, U. Measurement of baseline toxicity and QSAR analysis of 50 non-polar and 58 polar narcotic chemicals for the alga *Pseudokirchneriella subcapitata*. *Chemosphere* **2014**, *96*, 23–32.
- (56) Sushko, I.; Novotarskyi, S.; Korner, R.; Pandey, A. K.; Rupp, M.; Teetz, W.; Brandmaier, S.; Abdelaziz, A.; Prokopenko, V. V.; Tanchuk, V. Y.; Todeschini, R.; Varnek, A.; Marcou, G.; Ertl, P.; Potemkin, V.; Grishina, M.; Gasteiger, J.; Schwab, C.; Baskin, I. I.; Palyulin, V. A.; Radchenko, E. V.; Welsh, W. J.; Kholodovych, V.; Chekmarev, D.; Cherkasov, A.; Aires-de-Sousa, J.; Zhang, Q. Y.; Bender, A.; Nigsch, F.; Patiny, L.; Williams, A.; Tkachenko, V.; Tetko, I. V. Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information. *J. Comput. Aided Mol. Des.* **2011**, *25*, 533–554.
- (57) Walker, T.; Grulke, C. M.; Pozefsky, D.; Tropsha, A. Chembench: a cheminformatics workbench. *Bioinformatics* **2010**, *26*, 3000–3001.
- (58) Scheiber, J.; Chen, B.; Milik, M.; Sukuru, S. C.; Bender, A.; Mikhailov, D.; Whitebread, S.; Hamon, J.; Azzaoui, K.; Urban, L.; Glick, M.; Davies, J. W.; Jenkins, J. L. Gaining insight into off-target mediated effects of drug candidates with a comprehensive systems chemical biology analysis. *J. Chem. Inf. Model.* **2009**, *49*, 308–317.
- (59) Cases, M.; Mestres, J. A chemogenomic approach to drug discovery: focus on cardiovascular diseases. *Drug Discov. Today* **2009**, *14*, 479–485.
- (60) Muskavitch, M. A.; Barteneva, N.; Gubbels, M. J. Chemogenomics and parasitology: small molecules and cell-based assays to study infectious processes. *Comb. Chem. High Throughput Screen.* **2008**, *11*, 624–646.
- (61) Ganter, B.; Tugendreich, S.; Pearson, C. I.; Ayanoglu, E.; Baumhueter, S.; Bostian, K. A.; Brady, L.; Browne, L. J.; Calvin, J. T.; Day, G. J.; Breckenridge, N.; Dunlea, S.; Eynon, B. P.; Furness, L. M.; Ferng, J.; Fielden, M. R.; Fujimoto, S. Y.; Gong, L.; Hu, C.; Idury, R.; Judo, M. S.; Kolaja, K. L.; Lee, M. D.; McSorley, C.; Minor, J. M.; Nair, R. V.; Natsoulis, G.; Nguyen, P.; Nicholson, S. M.; Pham, H.; Roter, A. H.; Sun, D.; Tan, S.; Thode, S.; Tolley, A. M.; Vladimirova, A.; Yang, J.; Zhou, Z.; Jarnagin, K. Development of a large-scale chemogenomics database to improve drug candidate selection and to understand mechanisms of chemical toxicity and action. *J. Biotechnol.* **2005**, *119*, 219–244.
- (62) Vieth, M.; Higgs, R. E.; Robertson, D. H.; Shapiro, M.; Gragg, E. A.; Hemmerle, H. Kinomics-structural biology and chemogenomics of kinase inhibitors and targets. *Biochim. Biophys. Acta* **2004**, *1697*, 243–257.
- (63) Mestres, J. Computational chemogenomics approaches to systematic knowledge-based drug discovery. *Curr. Opin. Drug Discov. Devel.* **2004**, *7*, 304–313.
- (64) Bredel, M.; Jacoby, E. Chemogenomics: an emerging strategy for rapid target and drug discovery. *Nat. Rev. Genet.* **2004**, *5*, 262–275.
- (65) Clark, A. M.; Sarker, M.; Ekins, S. New target predictions and visualization tools incorporating open source molecular fingerprints for TB Mobile 2.0. *J. Cheminform.* **2014**, *6*, 38.
- (66) Clark, A. M.; Williams, A. J.; Ekins, S. Cheminformatics workflows using mobile apps. *Chem-Bio Informatics J.* **2013**, *13*, 1–18.
- (67) Clark, A. M.; Ekins, S.; Williams, A. J. Redefining Cheminformatics with Intuitive Collaborative Mobile Apps. *Mol. Inform.* **2012**, *31*, 569–584.
- (68) Gleeson, M. P.; Davis, A. M.; Chohan, K. K.; Paine, S. W.; Boyer, S.; Gavanaghan, C. L.; Arnby, C. H.; Kankkonen, C.; Albertson, N. Generation of in-silico cytochrome P450 1A2, 2C9, 2C19, 2D6, and 3A4 inhibition QSAR models. *J. Comput. Aided Mol. Des.* **2007**, *21*, 559–573.
- (69) Jennings, L. E.; Measures, A. R.; Wilson, B. G.; Conway, S. J. Phenotypic screening and fragment-based approaches to the discovery of small-molecule bromodomain ligands. *Future Med. Chem.* **2014**, *6*, 179–204.
- (70) Perryman, A. L.; Yu, W.; Wang, X.; Ekins, S.; Forli, S.; Li, S. G.; Freundlich, J. S.; Tonge, P. J.; Olson, A. J. A Virtual Screen Discovers Novel, Fragment-Sized Inhibitors of *Mycobacterium tuberculosis* InhA. *J. Chem. Inf. Model.* **2015**, *55*, 645–659.
- (71) Chekmarev, D. S.; Kholodovych, V.; Balakin, K. V.; Ivanenkov, Y.; Ekins, S.; Welsh, W. J. Shape signatures: new descriptors for predicting cardiotoxicity in silico. *Chem. Res. Toxicol.* **2008**, *21*, 1304–1314.
- (72) Cui, J.; Fu, R.; Zhou, L. H.; Chen, S. P.; Li, G. W.; Qian, S. X.; Liu, S. BCR-ABL tyrosine kinase inhibitor pharmacophore model derived from a series of phenylaminopyrimidine-based (PAP) derivatives. *Bioorg. Med. Chem. Lett.* **2013**, *23*, 2442–2450.
- (73) Zamecnikova, A. Targeting the BCR-ABL tyrosine kinase in chronic myeloid leukemia as a model of rational drug design in cancer. *Exp. Rev. Hematol.* **2010**, *3*, 45–56.
- (74) Costache, A. D.; Trawick, D.; Bohl, D.; Sem, D. S. AmineDB: large scale docking of amines with CYP2D6 and scoring for druglike

properties—towards defining the scope of the chemical defense against foreign amines in humans. *Xenobiotica* **2007**, *37*, 221–245.

(75) Hansen, K.; Mika, S.; Schroeter, T.; Sutter, A.; ter Laak, A.; Steger-Hartmann, T.; Heinrich, N.; Muller, K. R. Benchmark dataset for in silico prediction of Ames mutagenicity. *J. Chem. Inf. Model.* **2009**, *49*, 2077–2081.

(76) Balfer, J.; Bajorath, J. Introduction of a methodology for visualization and graphical interpretation of Bayesian classification models. *J. Chem. Inf. Model.* **2014**, *54*, 2451–2468.

(77) Crisman, T. J.; Parker, C. N.; Jenkins, J. L.; Scheiber, J.; Thoma, M.; Kang, Z. B.; Kim, R.; Bender, A.; Nettles, J. H.; Davies, J. W.; Glick, M. Understanding false positives in reporter gene assays: in silico chemogenomics approaches to prioritize cell-based HTS data. *J. Chem. Inf. Model.* **2007**, *47*, 1319–1327.

(78) Paolini, G. V.; Shapland, R. H.; van Hoorn, W. P.; Mason, J. S.; Hopkins, A. L. Global mapping of pharmacological space. *Nat. Biotechnol.* **2006**, *24*, 805–815.

(79) Keiser, M. J.; Roth, B. L.; Armbruster, B. N.; Ernsberger, P.; Irwin, J. J.; Shoichet, B. K. Relating protein pharmacology by ligand chemistry. *Nat. Biotechnol.* **2007**, *25*, 197–206.

(80) Laggner, C.; Kokel, D.; Setola, V.; Tolia, A.; Lin, H.; Irwin, J. J.; Keiser, M. J.; Cheung, C. Y.; Minor, D. L., Jr.; Roth, B. L.; Peterson, R. T.; Shoichet, B. K. Chemical informatics and target identification in a zebrafish phenotypic screen. *Nat. Chem. Biol.* **2012**, *8*, 144–146.

(81) Riniker, S.; Wang, Y.; Jenkins, J. L.; Landrum, G. A. Using information from historical high-throughput screens to predict active compounds. *J. Chem. Inf. Model.* **2014**, *54*, 1880–1891.

(82) Bird, C. L.; Frey, J. G. Chemical information matters: an e-Research perspective on information and data sharing in the chemical sciences. *Chem. Soc. Rev.* **2013**, *42*, 6754–6776.

(83) Ekins, S.; Clark, A. M.; Swamidass, S. J.; Litterman, N.; Williams, A. J. Bigger data, collaborative tools and the future of predictive drug discovery. *J. Comput. Aided Mol. Des.* **2014**, *28*, 997–1008.

(84) Zhang, J.; Hsieh, J. H.; Zhu, H. Profiling animal toxicants by automatically mining public bioassay data: a big data approach for computational toxicology. *PLoS One* **2014**, *9*, No. e99863.

(85) Gaspar, H. A.; Baskin, I. I.; Marcou, G.; Horvath, D.; Varnek, A. Chemical data visualization and analysis with incremental generative topographic mapping: big data challenge. *J. Chem. Inf. Model.* **2015**, *55*, 84–94.

(86) Shah, F.; Greene, N. Analysis of Pfizer compounds in EPA's ToxCast chemicals-assay space. *Chem. Res. Toxicol.* **2014**, *27*, 86–98.

(87) Lamb, J. The Connectivity Map: a new tool for biomedical research. *Nat. Rev. Cancer* **2007**, *7*, 54–60.

(88) Wojcikowski, M.; Zielenkiewicz, P.; Siedlecki, P. DiSCuS: an open platform for (not only) virtual screening results management. *J. Chem. Inf. Model.* **2014**, *54*, 347–354.

(89) Wang, X.; Chen, H.; Yang, F.; Gong, J.; Li, S.; Pei, J.; Liu, X.; Jiang, H.; Lai, L.; Li, H. iDrug: a web-accessible and interactive drug discovery and design platform. *J. Cheminform.* **2014**, *6*, 28.

(90) Ekins, S.; Williams, A. J.; Krasowski, M. D.; Freundlich, J. S. In silico repositioning of approved drugs for rare and neglected diseases. *Drug Discov. Today* **2011**, *16*, 298–310.

(91) Lounkine, E.; Keiser, M. J.; Whitebread, S.; Mikhailov, D.; Hamon, J.; Jenkins, J. L.; Lavan, P.; Weber, E.; Doak, A. K.; Cote, S.; Shoichet, B. K.; Urban, L. Large-scale prediction and testing of drug activity on side-effect targets. *Nature* **2012**, *486*, 361–367.

(92) Ponder, E. L.; Freundlich, J. S.; Sarker, M.; Ekins, S. Computational Models For Neglected Diseases: Gaps and Opportunities. *Pharm. Res.* **2013**, *31*, 271–277.