Chapter 6

Virtual High-Throughput Screening with Two-Dimensional Kernels

Chloé-Agathe Azencott CAZENCOT@ICS.UCI.EDU
Department of Computer Science
Institute for Genomics and Bioinformatics
University of California, Irvine
Irvine, CA 92697-3435, USA

Pierre Baldi PFBALDI@ICS.UCI.EDU
Department of Computer Science
Institute for Genomics and Bioinformatics
University of California, Irvine
Irvine, CA 92697-3435, USA

Abstract

High-Throughput Screening (HTS) is an important technology that relies on massively testing large numbers of compounds for their activity on a given assay in order to identify potential drug leads in the early stages of a drug discovery pipeline. However, because identification of drug leads by HTS is very costly, it is of great interest to develop computational methods for virtual HTS (VHTS), in order to prioritize the compounds to be screened and identify a relatively small, but highly promising, subset from a screening library that can be tested more economically. Here we develop statistical machine learning methods, based on two-dimensional spectral kernels for small molecules and extended-connectivity molecular substructures (ECFPs), to address this task. We apply them to the HIVA dataset of the Agnostic Learning versus Prior Knowledge Challenge and obtain the best results with a balanced error rate of 0.2693 and an area under the ROC curve of 0.7643 on the testing set.

Keywords: virtual high-throughput screening, drug discovery, drug screening, kernels, HTS, SVM

6.1. Introduction: The Virtual High-Throughput Screening Problem

High-Throughput Screening (HTS) is an approach to drug discovery developed in the 1980's in the pharmaceutical industry that allows to massively test large numbers (up to millions) of compounds for their activity on a given assay in order to identify potential drug leads. Nowadays, it is possible to screen up to 100,000 molecules per day in a single HTS facility. This process, however, requires a considerable amount of resources and capital investment, for instance in terms of robotics, molecular libraries, and the amount of relevant protein that must be produced. A widely circulated figure is that HTS screening costs on the order of one dollar per compound, a price that cannot be afforded by most academic laboratories.

The *in silico* approach to HTS, also called virtual HTS (VHTS), attempts to computationally select from a list of molecular compounds only those most likely to possess the properties required to positively satisfy a given assay. When the 3D structure of a target protein is known, the most common approach to VHTS is docking, which consists in scoring the compatibility of each small molecule in the screening library with respect to the known, or putative, binding

pockets of the protein target. When the 3D structure of the targets is not known, or to further validate the results of a docking experiment, other computational methods must be used. In many cases, an initial list of positive and negative compounds may be known from previous, possibly small-scale, screening experiments. Therefore, in these cases, one is interested in using statistical machine learning or other methods to build a good molecular predictor and possibly clarify what are the desirable properties a molecule should have in order to positively satisfy the conditions of a given assay. The development of good VHTS methods is essential if one is to drastically reduce the number of compounds that must be experimentally assayed and reduce the time and cost of HTS.

Among the five datasets offered by the IJCNN-07 Agnostic Learning versus Prior Knowledge Challenge [1], we decided to focus on the HIVA set derived from the DTP AIDS Antiviral Screen program made available by the National Cancer Institute (NCI)[2]. This dataset contains assay results for 42,678 chemicals tested for their activity against the AIDS virus and provides a reasonable benchmark for the development of VHTS algorithms.

As in most chemoinformatics applications, such as the storage and search of large databases of small molecules (Chen et al., 2005; Swamidass and Baldi, 2007) or the prediction of their physical, chemical, and biological properties (Swamidass et al., 2005; Azencott et al., 2007), the issues of molecular data structures and representations play an essential role (Leach and Gillet, 2005). These representations and data structures are essential to define "molecular similarity", which in turn is crucial for developing efficient methods both to search the databases and predict molecular properties using kernel methods. Leveraging previous work in our group and in the literature, here we use SVMs in combination with 2D spectral representations of small molecules with Tanimoto and MinMax kernels to address the VHTS problem and tackle the HIVA challenge.

6.2. Molecular Data Representation

Small molecules are often described by graphs (King, 1983; Bonchev, 1991; McNaught and Wilinson, 1997), where vertices represent atoms and edges represent bonds. Other representations, such as one-dimensional SMILES strings (Weiniger et al., 1989) or three-dimensional descriptions based on the atomic coordinates, have been developed. Previous studies (Swamidass et al., 2005; Azencott et al., 2007) in our group as well as in other groups suggest, however, that these representations do not lead for now to better predictive performance. In this regard, it is worth noting for SMILES strings that the information they contain is identical to the information contained in the bond graphs. For 3D-based representations, the majority of the coordinates must be predicted, since only a relatively small fraction of molecular structures have been empirically solved. Furthermore, the 2D representation of molecules as graphs is also the representation of choice that underlies the structural similarity search algorithms of chemical databases such as ChemBank (Strausberg and Schreiber, 2003), ChemMine (Girke et al., 2005), or ChemDB (Chen et al., 2005, 2007).

6.2.1. Molecular Graphs

We describe a molecule as a labeled graph of bonds. Labels on the vertices represent the atom types and labels on the edges characterize the bonds. More precisely, vertices are labeled according to one of the following schemes:

1. http://www.agnostic.inf.ethz.ch/index.php
2. http://dtp.nci.nih.gov/docs/aids/aids_data.html

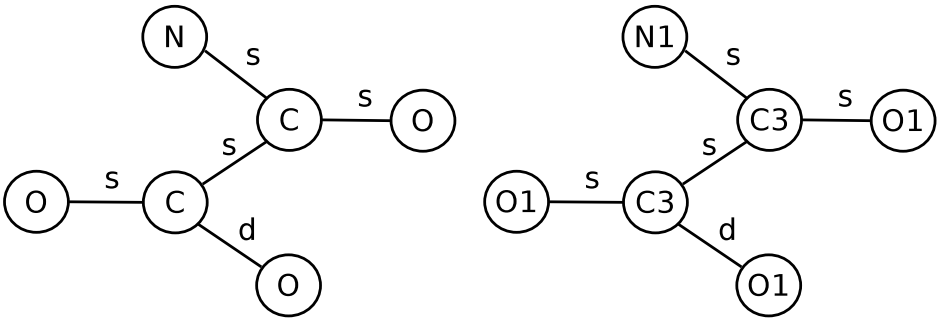

Figure 6.1: Example of molecular graphs. The vertices represent atoms, labeled with the Element scheme on the left and the Element-Connectivity scheme on the right. Bonds are represented by edges, labeled "s" for simple bonds and "d" for double bonds. Note that by convention Hydrogen atoms are ignored.

- E: Element. Each atom is simply labeled by its symbol (e.g. C for carbon, O for oxygen, N for nitrogen)
- EC: Element-Connectivity. Each atom is labeled by its symbol together with the number of atoms it is bonded to (e.g. C3 for a carbon with three atoms attached)

The bonds are simply labeled according to their type (e.g. single, double).

Figure 6.1 gives an example of the two-dimensional representation of a molecule as a graph.

From these graphs, a number of features can be extracted, such as the presence/absence or number of occurrences of particular functional groups. A more recent and general trend, however, has been to define features in terms of labeled subgraphs, such as labeled paths (Swamidass et al., 2005; Azencott et al., 2007) or labeled trees (Mahé et al., 2006), and to combinatorially extract and index all such features to represent molecules using large feature vectors, also known as fingerprints. While in other work we have compared the use of different features and have tried several of them on the HIVA challenge, here we focus on a class of shallow labeled trees, also known as extended-connectivity features in the literature (Hassan et al., 2006).

6.2.2. Extended-Connectivity Molecular Features

The concept of molecular connectivity (Razinger, 1982; Kier and Hall, 1986) leads to the idea of extended-connectivity substructures (Rogers and Brown, 2005; Hassan et al., 2006), which are labeled trees rooted at each vertex of the molecular graph. A depth parameter d controls the depth of the trees (Figure 6.2). For a given tree, this algorithm recursively labels each tree node (or atom) from the leaf nodes to the root, appending to each parent's label the labels of its children in the tree. Each resulting vertex label is then considered as a feature. For the labeling process to be unique, the vertices of the graph need to be ordered in a unique canonical way. This ordering is achieved using Morgan's algorithm (Morgan, 1965).

We extract extended-connectivity substructures of depth d up to 2, where the depth indicates the maximum distance, measured in number of bonds, to the root of each labeled tree. For example, a depth of two indicates that the label for a given atom will be composed of the labels for the neighboring atoms which are connected to it by at most two bonds. Other depths (3 to 6) have been tested but did not lead to any performance improvement.

Figure 6.2 shows an example of extended-connectivity labeling.

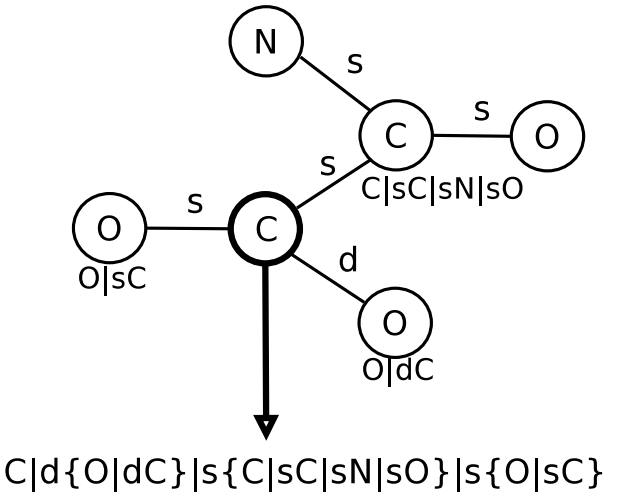

Figure 6.2: The extended-connectivity label of depth d up to 2 of the C atom circled in bold is given by the labels of depth up to 1 of its three neighboring atoms: (1) an O atom to which it is connected by a double bond, (2) a C atom to which it is connected by a single bond, and (3) an O atom to which it is connected by a single bond. If the EC scheme was to be used, the resulting label would be: `C3|d{O1|dC3}|s{C3|sC3|sN1|sO1}|s{O1|sC3}`.

6.2.3. Molecular Fingerprints

The molecular features are computed across the whole dataset. Each molecule can then be represented as a vector of fixed size N, where N is the total number of features found. For a given molecule, each component of the vector is set to 1 if the corresponding feature is present in the chemical, and 0 otherwise. We also use count vectors where each component of the vector is set to c, where c is the number of times the corresponding feature appears in the chemical. These feature vectors are actually extensions of traditional chemical fingerprints (Flower, 1998; Raymond and Willett, 2001).

The spectral or combinatorial approach to molecular fingerprints can easily be automated and has several advantages: (1) it alleviates the need for relying on expert knowledge, that may itself be incomplete, to select relevant molecular descriptors; (2) it produces a fixed-size representation for data of varying size.; and (3) it has been shown to be effective in the literature. Furthermore these long vectors, which have on the order of 100,000 components for the HIVA dataset, are also very sparse and can be efficiently compressed, in lossy or even lossless fashion (Baldi et al., 2007), to reduce their dimensionality and improve storage and computational complexity.

6.3. Support Vector Machines for virtual HTS

6.3.1. Kernels For Molecules

To define kernels for chemical compounds, we need to introduce a similarity measure between molecular fingerprints. Here we use the MinMax and Tanimoto similarity measures.

If $f_1 = (f_{1,1}, \ldots, f_{1,N})$ and $f_2 = (f_{2,1}, \ldots, f_{2,N})$ are two count fingerprints, the MinMax similarity measure (Swamidass et al., 2005; Ralaivola et al., 2005) is defined by

$$K(f_1, f_2) = \frac{\sum_i \min(f_{1,i}, f_{2,i})}{\sum_i \max(f_{1,i}, f_{2,i})} \tag{6.1}$$

In the case of binary fingerprints the MinMax measure reduces to the Tanimoto similarity measure defined by

$$K(f_1, f_2) = \frac{f_1 \cap f_2}{f_1 \cup f_2} \tag{6.2}$$

Both similarity measures have been shown (Swamidass et al., 2005) to be semi-definite positive and satisfy Mercer's kernel conditions. Thus the MinMax and Tanimoto kernels can be applied in combination with an SVM optimization framework to derive a molecular predictor in VHTS experiments.

6.3.2. Implementation

The HIVA dataset contains 42,678 examples. The associated pair-to-pair kernel matrix being rather large, an online implementation of SVM is desirable. Here we use the SVMTorch (Collobert and Bengio, 2001) implementation, which allows on-line learning and is thus well suited for our purpose.

Besides their size, one of the other issues with HTS datasets is that they are often highly unbalanced, usually containing far more negative than positive examples. This is the case of the HIVA dataset, which has about 28 times as many negative examples as positive examples. Without any further processing, this will negatively affect the predictor and bias it towards negative examples.

The most straightforward method to deal with class unbalance is to control the sensitivity (or C parameter) of the SVM (Veropoulos et al., 1999; Shin and Cho, 2003). By assigning a higher sensitivity to the under-represented class, one increases the coefficients of the corresponding support vectors, thus biasing the classifier towards the minority class. We first tested this method, which did not lead to significant improvements.

Another way of compensating for the small amount of positive examples is to re-sample the data, so as to train the SVM on a balanced set of examples. In this work we focus on over-sampling, which consists in replicating the under-represented class so as to get a more balanced number of examples. This method has been widely studied in the literature (Estabrooks et al., 2004; Orriols and Bernad-Mansilla, 2005).

If m is the number of training examples and m_+ the number of positive training examples, we randomly split the negative data in $\frac{m}{m_+}$ subsets of about m_+ examples and build $\frac{m}{m_+}$ classifiers, each trained on a set composed of one of the negative subsets together with the m_+ positive examples. Each individual classifier produces a value of +1 if its prediction is positive and -1 if its prediction is negative. Then these values are added, and the final decision is made by comparing the resulting sum to a threshold. As this method overcompensates and leads to a bias favoring the positive class, the decision threshold has to be adjusted to a value greater than 0. To address this problem, we apply this method using 10-fold cross-validation over the training set

and select the threshold that leads to the best performance on the training set. An SVM trained according to this algorithm will further be referred to as an oversampled SVM.

Eventually, we run a 10-fold cross-validation over the training set for each combination of labeling scheme, representation by bits or counts, and oversampling or not, and retain as best models the ones leading to optimal performance.

6.3.3. Performance Measures

The SVM classifiers associate a prediction value to each of the data points. We then order the values, thus ranking the data points, and set a threshold so as to separate predicted actives from predicted inactives. A number of performance measures can then be used in order to assess the performance and compare different methods.

The Agnostic Learning versus Prior Knowledge Challenge focused on the balanced error rate (BER) and area under the ROC curve (AUC) measures.

If $m_- = m - m_+$ is the number of negative examples, TP the number of true positives, TN the number of true negatives and FP the number of false positives, the BER is defined by

$$BER = 1 - \frac{1}{2}\left(\frac{TP}{m_+} + \frac{TN}{m_-}\right) \tag{6.3}$$

and the AUC is the area under the ROC curve defined by plotting the true positive rate $\frac{TP}{m_+}$ against the false positive rate $\frac{FP}{m_-}$ for each confidence value.

While these measures allow one to compare all the predictors to each other (especially in the Agnostic Learning track), they may not provide an optimal way of assessing VHTS methods. Indeed, these performance metrics do not address the "early recognition problem", in the sense that they do not quantify how efficient a given classifier is at retrieving active compounds *early*, i.e. at the top of the ranked list. High-enrichment for positives in the top of the list is highly desirable in VHTS, especially in conditions where only few compounds can be empirically tested.

An enrichment curve, representing the percentage of true positives captured as a function of the percentage of the ranked list screened, can be used to judge the ability of a predictor to recover active compounds early.

Whereas enrichment curves provide a graphical means for evaluating early recognition across many thresholds, capturing this property in a single numerical value is also desirable as a summary and to allow for easy comparison of several predictors. Truchon and Bayly (2007) develop this idea and propose a performance measure called Boltzmann-Enhanced Discrimination of Receiver Operating Characteristic (BEDROC) which partly addresses this issue.

The notion of BEDROC measure stems from the study of various virtual screening metrics, including the area under the enrichment curve (AUEC). If $TP(x)$ denotes the true positive rate, then the AUEC is defined by

$$AUEC = \int_0^1 TP(x)dx \tag{6.4}$$

The AUEC can be interpreted as the probability that an active compound will be ranked better than a compound selected at random by a uniform distribution. Therefore, in order to address the early recognition problem, Truchon and Bayly (2007) introduce the concept of weighted AUEC (wAUEC), defined by

$$wAUEC = \frac{\int_0^1 TP(x)w(x)dx}{\int_0^1 w(x)dx} \tag{6.5}$$

where $w(x)$ is a weighting probability distribution. The wAUEC is the probability that an active compound will be ranked better than a compound that would come from the probability distribution function w. By choosing for w an exponential distribution $w(x) = C(\alpha)e^{-\alpha x}$, which has higher values for low values of x, one gives a higher importance to the active compounds recognized at the top of the ranked list.

In the general case, the theoretical extreme values of the AUEC and the wAUEC measures depend on the number of actives and inactives of the problem being considered and differ from the usual 0 and 1 values associating for instance with the AUC measure. Note that the AUC is simply a scaled version of the AUEC, obtained through the following linear transformation:

$$AUC = \frac{AUEC - AUEC_{min}}{AUEC_{max} - AUEC_{min}} \tag{6.6}$$

Truchon and Bayly (2007) define the BEDROC by a similar scaling of the wAUEC:

$$BEDROC = \frac{wAUEC - wAUEC_{\min}}{wAUEC_{\max} - wAUEC_{\min}} \tag{6.7}$$

Therefore, the BEDROC measure can be seen as a generalization of the AUC metric that takes early recognition into account.

If $\alpha.\frac{m_+}{m} \ll 1$ and $\alpha \neq 0$, then the BEDROC measure is approximately equal to the wAUEC measure, and can be interpreted as the probability that an active compound will be ranked better than a compound selected at random from an exponential probability distribution function of parameter α.

Formally, if for every k in $[1,\ldots,m_+]$ we let r_k be the ranking of the k-th active compound, then the BEDROC metric can be estimated by

$$BEDROC \approx \frac{1}{\alpha\frac{m_+}{m}}\left(\frac{\sum_{k=1}^{m_+} e^{-\alpha.(r_k/N)}}{\frac{1-e^{-\alpha}}{e^{\alpha/m}-1}}\right) + \frac{1}{1+e^{-\alpha}} \tag{6.8}$$

In what follows, we use a typical value of $\alpha = 1$ for the early recognition parameter.

6.4. Results

The Agnostic Learning versus Prior Knowledge Challenge is run using a training set composed of 4,229 compounds randomly selected in the HIVA dataset, and a blind test set composed of the remaining 38,449 compounds. We optimize our models by 10-fold cross-validation on the training set and then evaluate their performance on the test set. The aim of the challenge is to reach the lowest possible BER on the testing set.

Table 6.1 reports the 10-fold cross-validation BER and AUC over the training set as well as the final performance of several of the tested methods. Combining molecular fingerprints with an Element labeling of atoms and a count-based fingerprint representation, together with an oversampled SVM framework, lead to the best entry among all competing groups for the HIVA dataset in the Prior Knowledge track, with a BER of 0.2693. The best 10-fold cross-validated BER on the training set, with a value of 0.1975, is achieved by the same method. We compare these results to those obtained by the winner of the Performance Prediction Challenge (Guyon et al., 2006), where the dataset was the same, but split in training and testing sets in a different fashion, and to the best results in the Agnostic Learning track[3], as well as to the second best results in the Prior Knowledge track. These second best results, with a BER of 0.2782,

3. available from http://clopinet.com/isabelle/Projects/agnostic/Results.html

have been obtained by S. Joshua Swamidass, also from our laboratory, by applying a neural-network-based approach to the same molecular fingerprints. This approach will be described elsewhere and has its own advantages, for instance in terms of speed. Both top entries in the Prior Knowledge track achieve better performance than the best entry in the Agnostic Learning track.

Table 6.1: 10-fold cross-validation BER and AUC over the HIVA training set, as well as final BER and AUC for several methods. (*) Winning entry. Best performance in bold and second best performance in italics. 'E' and 'EC' refer to the labeling schemes introduced in Section 6.2.1; 'binary' and 'counts' refer to the vector representations defined in Section 6.2.3; and 'oversampled' refers to an SVM trained on a balanced dataset obtained by replicating the underrepresented class as exposed in Section 6.3.2.

Method	Training set		Test set	
	BER	**AUC**	**BER**	**AUC**
E, binary (not oversampled)	0.2249	0.8293	0.2816	0.7550
E, binary (oversampled)	*0.1980*	0.8511	*0.2765*	0.7611
E, counts (not oversampled)	0.2238	0.8294	0.2799	0.7576
E, counts (oversampled) (*)	**0.1975**	0.8523	**0.2693**	0.7643
EC, binary (not oversampled)	0.2174	0.8338	0.2828	0.7673
EC, binary (oversampled)	0.2030	0.8413	0.2860	0.7595
EC, counts (not oversampled)	0.2189	0.8358	0.2826	0.7626
EC, counts (oversampled)	0.1993	0.8450	0.2820	0.7650
Second Best (Prior Knowledge)	0.2168	0.8198	0.2782	0.7072
Best (Agnostic Learning)	-	-	0.2827	0.7707
Performance Prediction Challenge	-	-	0.2757	0.7671

The 10-fold cross-validated enrichment curves over the training set for several methods are displayed on Figure 6.3. Close to the origin, the highest enrichment on these curves is clearly observed when using a non-oversampled SVM. This region is further magnified in Figure 6.4 which focuses on the first 10% of the ranked list. It suggests that a slightly better ability at early recognition is attained with the model derived from binary fingerprints using the element labeling scheme.

The actual enrichment curves obtained on the testing set are displayed on Figure 6.5. Here again, the best early recognition ability is clearly observed for non-oversampled SVM. Figure 6.6, which focuses on the first 10% of these enrichment curves, suggests that the model derived from count fingerprints obtained with the element labeling scheme has the best ability to recover actives at the top of the ranked list.

Table 6.2 presents the 10-fold cross-validation BEDROC over the HIVA training set as well as the final BEDROC of several methods. The best final BEDROC of 0.507 is also obtained with molecular fingerprints combined with an Element labeling of atoms and a count-based fingerprint representation, but together with an non-oversampled SVM framework. This method, which corresponds to the enrichment curve with the steepest slope before 5%, achieves a 10-fold cross-validated BEDROC of 0.609 on the training set, just behind the best value of 0.610 obtained when using a binary fingerprint representation instead of the count-based one.

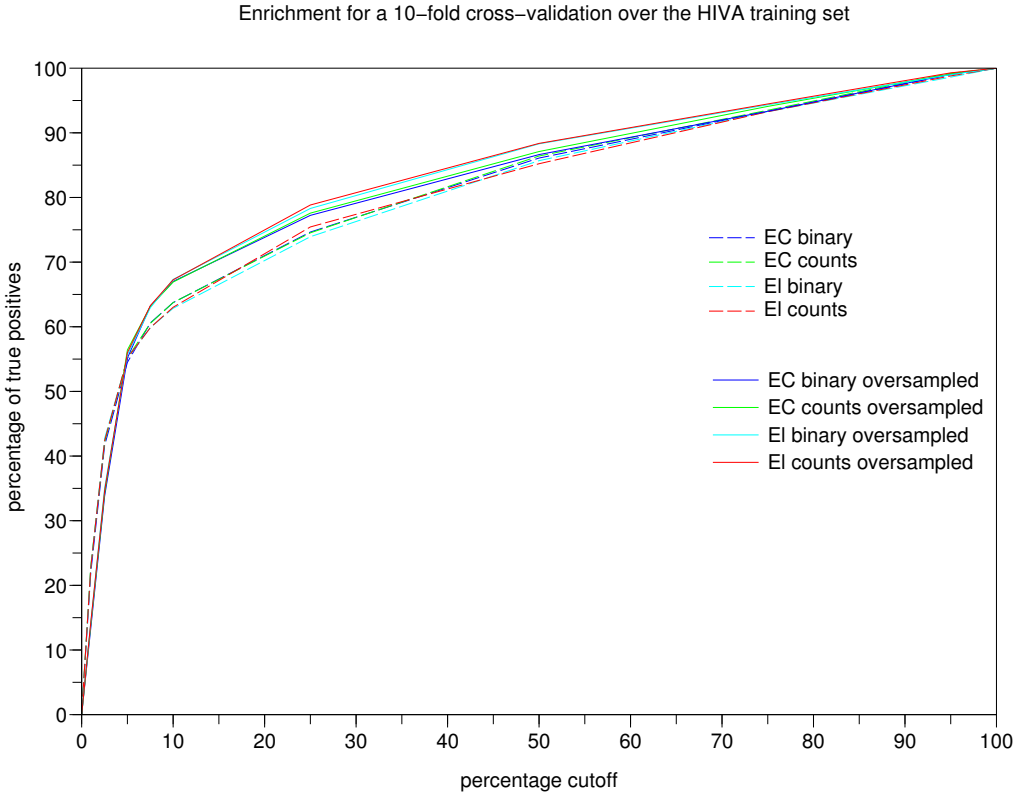

Figure 6.3: Ten-fold cross-validation enrichment curves over the HIVA training set for several methods. 'E' and 'EC' refer to the labeling schemes introduced in Section 6.2.1; 'binary' and 'counts' refer to the vector representations defined in Section 6.2.3; and 'oversampled' refers to an SVM trained on a balanced dataset obtained by replicating the underrepresented class as exposed in Section 6.3.2.

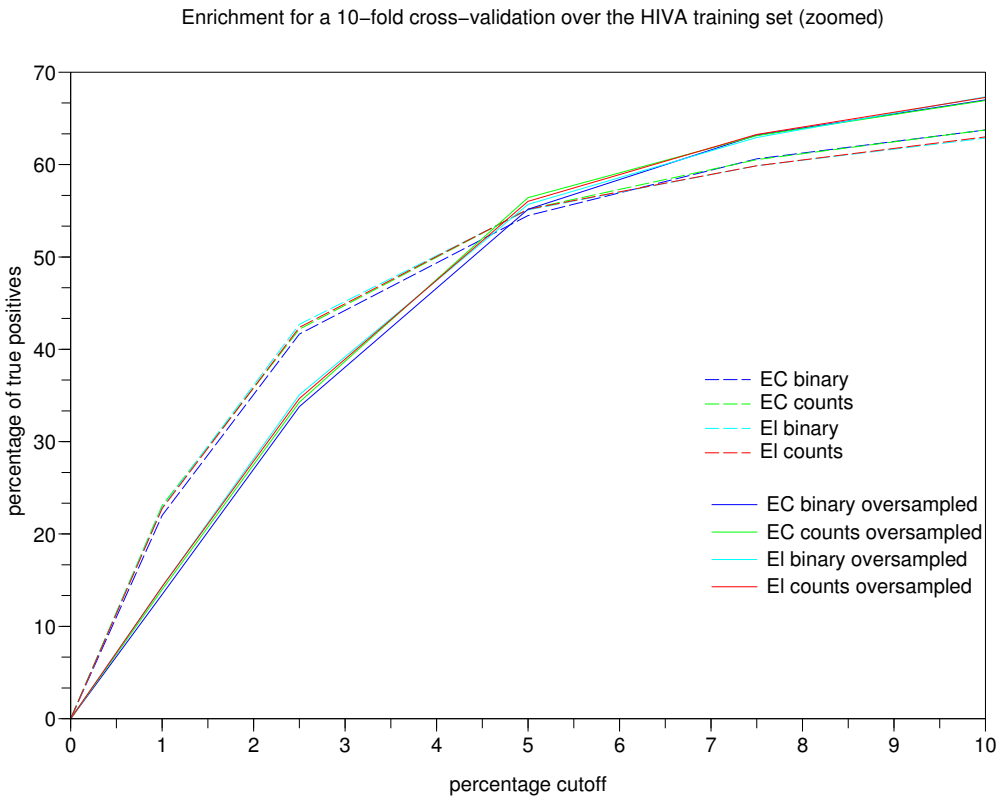

Figure 6.4: Ten-fold cross-validation enrichment curves, limited to the first 10% of the ranked list, over the HIVA training set for several methods. 'E' and 'EC' refer to the labeling schemes introduced in Section 6.2.1; 'binary' and 'counts' refer to the vector representations defined in Section 6.2.3; and 'oversampled' refers to an SVM trained on a balanced dataset obtained by replicating the underrepresented class as exposed in Section 6.3.2.

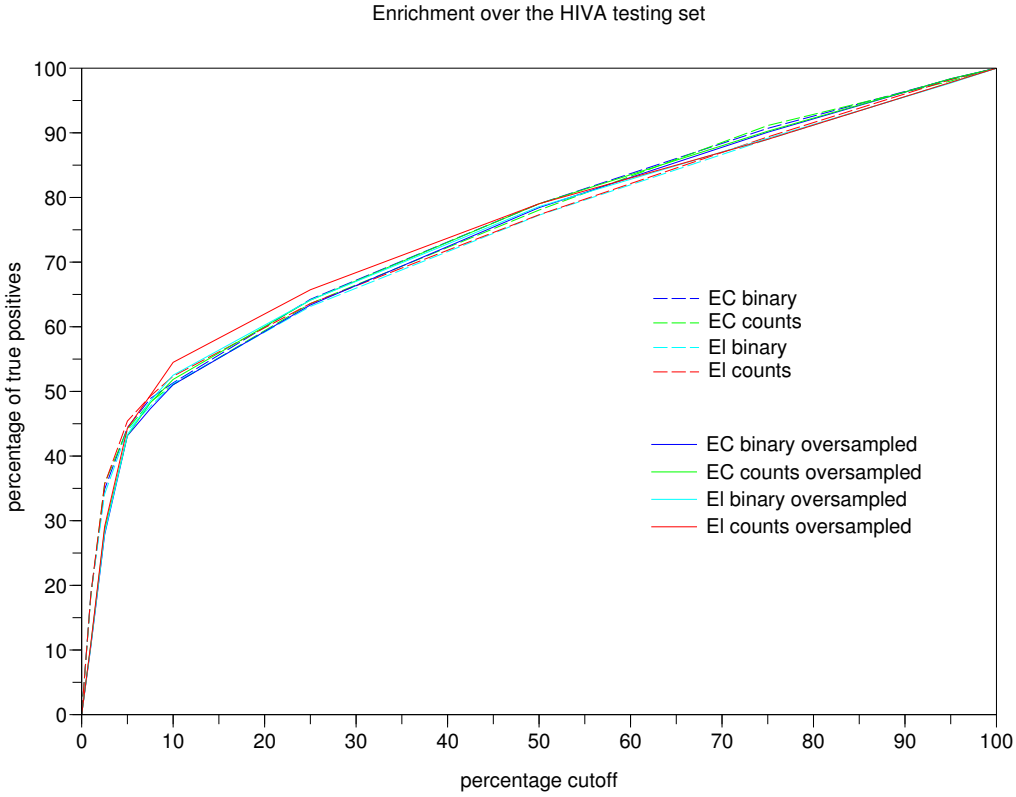

Figure 6.5: Actual enrichment curves over the HIVA testing set for several methods. 'E' and 'EC' refer to the labeling schemes introduced in Section 6.2.1; 'binary' and 'counts' refer to the vector representations defined in Section 6.2.3; and 'oversampled' refers to an SVM trained on a balanced dataset obtained by replicating the underrepresented class as exposed in Section 6.3.2.

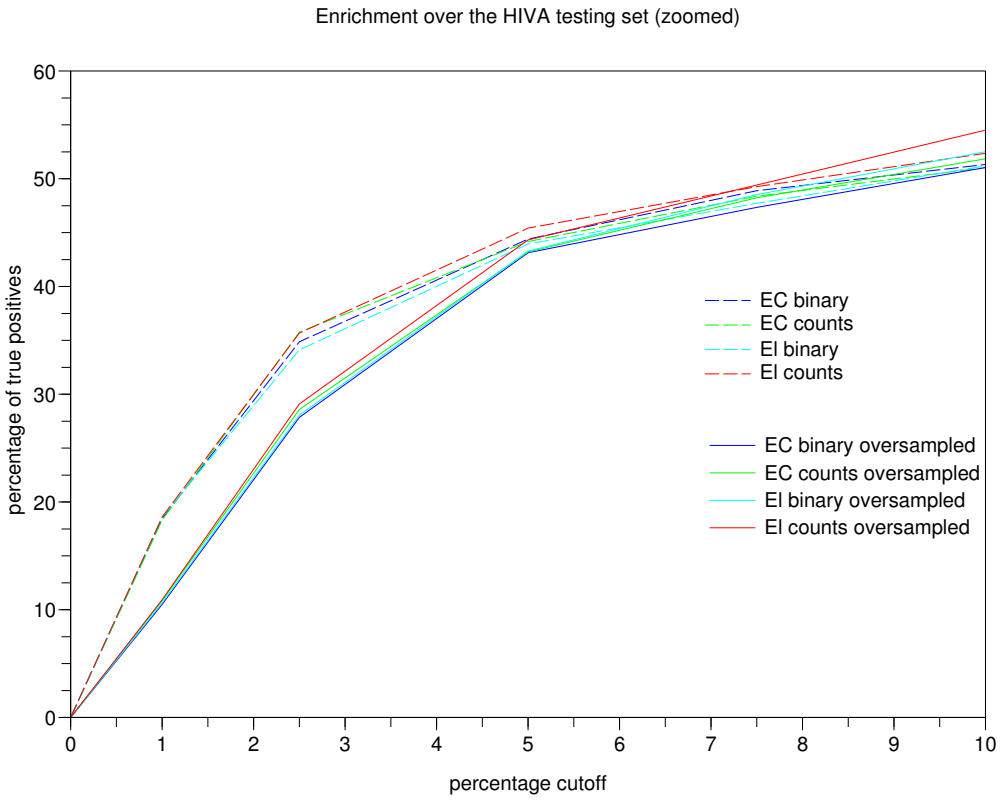

Figure 6.6: Actual enrichment curves, limited to the first 10% of the ranked list, over the HIVA testing set for several methods. 'E' and 'EC' refer to the labeling schemes introduced in Section 6.2.1; 'binary' and 'counts' refer to the vector representations defined in Section 6.2.3; and 'oversampled' refers to an SVM trained on a balanced dataset obtained by replicating the underrepresented class as exposed in Section 6.3.2.

Table 6.2: 10-fold cross-validation BEDROC over the training set as well as final BEDROC for several methods. (*) Winning entry. Best performance in bold and second best performance in italics. 'E' and 'EC' refer to the labeling schemes introduced in Section 6.2.1; 'binary' and 'counts' refer to the vector representations defined in Section 6.2.3; and 'oversampled' refers to an SVM trained on a balanced dataset obtained by replicating the underrepresented class as exposed in Section 6.3.2.

Method	Training set **BEDROC**	Test set **BEDROC**
E, binary (not oversampled)	**0.610**	*0.495*
E, binary (oversampled)	0.580	0.454
E, counts (not oversampled)	*0.609*	**0.507**
E, counts (oversampled) (*)	0.581	0.465
EC, binary (not oversampled)	0.606	0.499
EC, binary (oversampled)	0.573	0.446
EC, counts (not oversampled)	0.602	0.500
EC, counts (oversampled)	0.573	0.454
Second Best (Prior Knowledge)	0.607	0.483

6.5. Discussion

By defining feature vectors that capture molecular structural information, we have developed a kernel leading to the best results on the HIVA dataset in the Agnostic Learning versus Prior Knowledge Challenge.

The extended-connectivity molecular fingerprints present the advantage of being built automatically, without the need for human curation and expert knowledge. The results obtained with these representations are superior to those obtained using the set of binary molecular descriptors computed using the ChemTK package[4] which were offered in the Agnostic Learning track. Also, one of the challenge participants tried to collaborate with chemists to define meaningful features, but did not manage to get better results than using the Agnostic Learning features.

Overall, the results suggest that extended-connectivity fingerprints yield efficient molecular representations that can be successfully applied to a variety of chemoinformatics problems, from the prediction of molecular properties to the clustering of large libraries of compounds. These fingerprints are actually implemented in the current version of the ChemDB database (Chen et al., 2007) and routinely used to search compounds.

We also notice that the model selection method adopted here, although somewhat naïve being based only on the cross-validation performance over the training set, still allows us to efficiently choose the top classifiers and rank first in the competition. This is especially interesting because the test set is about nine times larger than the training set, raising concern of over-fitting. It may be of some interest to combine our features with the best methods of the Agnostic Learning track to see whether any further improvements can be derived.

Other extensions of this work include applying our best methods to other virtual HTS datasets. An important observation in this context is that the methods yielding best BER performance do not yield best BEDROC performance. This is because optimizing for early recognition is not equivalent to optimizing for overall classification. The enrichment curves, which are

4. http://www.sageinformatics.com

systematically steeper for low thresholds when using non-oversampled SVM, corroborate this observation. More precisely, it appears that oversampling improves the global performance of the classifier in terms of BER but not the early recognition in terms of BEDROC. This suggests that putting more emphasis on the positive training examples reduces the bias of the SVM, but also leads to assigning higher prediction values to some of the negative points. It is therefore critical to carefully choose which performance measure to optimize with regards to the specific problem being tackled and the resources available to conduct laboratory experiments to confirm the computational prediction.

Acknowledgments

Work supported by NIH Biomedical Informatics Training grant (LM-07443-01), NSF MRI grant (EIA-0321390), NSF grant 0513376, and a Microsoft Faculty Innovation Award to PB. We would like also to acknowledge the OpenBabel project and OpenEye Scientific Software for their free software academic licenses.

References

C.-A. Azencott, A. Ksikes, S. J. Swamidass, J. H. Chen, L. Ralaivola, and P. Baldi. One- to Four-Dimensional Kernels for Virtual Screening and the Prediction of Physical, Chemical, and Biological Properties. *J. Chem. Inf. Model*, 47(3):965–974, 2007.

P. Baldi, R. W. Benz, D. S. Hirshberg, and S. J. Swamidass. Lossless Compression of Chemical Fingerprints Using Integer Entropy Codes Improves Storage and Retrieval. *J. Chem. Inf. Model.*, 2007.

Danail Bonchev. *Chemical Graph Theory: Introduction and Fundamentals*. Taylor & Francis, 1991. ISBN 0856264547.

J. Chen, S. J. Swamidass, Y. Dou, J. Bruand, and P. Baldi. ChemDB: A Public Database Of Small Molecules And Related Chemoinformatics Resources. *Bioinformatics*, 21:4133–4139, 2005.

Jonathan H. Chen, Erik Linstead, S. Joshua Swamidass, Dennis Wang, and Pierre Baldi. ChemDB Update - Full-Text Search and Virtual Chemical Space. *Bioinformatics*, 2007. doi: 10.1093/bioinformatics/btm341. URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btm341v1.

R. Collobert and S. Bengio. SVMTorch: Support Vector Machines for Large-Scale Regression Problems. *J. Mach. Learn. Res.*, 1:143–160, Sep. 2001 2001. http://www.idiap.ch/learning/SVMTorch.html.

A. Estabrooks, T. Jo, and N. Japkowicz. A Multiple Resampling Method for Learning From Imbalanced Data Set. *Computational Intelligence*, 20(1), 2004.

D. R. Flower. On the Properties of Bit String-Based Measures of Chemical Similarity. *J. Chem. Inf. Comput. Sci.*, 38:378–386, 1998.

T. Girke, L.-C. Chen, and N. Raikhel. ChemMine. A Compound Mining Database For Chemical Genomics. *Plant Physiol.*, 138:573–577, 2005. URL http://bioinfo.ucr.edu/projects/PlantChemBase/search.php.

I. Guyon, A. Saffari, G. Dror, and J. M. Buhman. Performance Prediction Challenge. In *IEEE/INNS conference IJCNN 2006, Vancouver July 16-21*, 2006.

M. Hassan, R. D. Brown, S. Varma-O'Brien, and D. Rogers. Cheminformatics Analysis and Learning in a Data Pipelining Environment. *Molecular Diversity*, 10:283–299, 2006.

Lemont B Kier and Lower H Hall. *Molecular connectivity in structure-activity analysis*. Wiley, New York, 1986. ISBN 0-471-90983-1.

R.B. King. *Chemical Applications of Topology and Graph Theory*. Elsevier, October 1983. ISBN 0444422447.

A. R. Leach and V. J. Gillet. *An Introduction to Chemoinformatics*. Springer, 2005.

P. Mahé, L. Ralaivola, V. Stoven, and J.-P. Vert. The Pharmacophore Kernel for Virtual Screening with Support Vector Machines. *J. Chem. Inf. Model.*, 46:2003–2014, 2006.

Alan D. McNaught and Andrew Wilinson. Molecular Graph, 1997. URL http://www.iupac.org/publications/compendium/index.html.

H.L. Morgan. The Generation of Unique Machine Description for Chemical Structures - A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation*, 5:107–113, 1965.

A. Orriols and E. Bernad-Mansilla. The Class Imbalance Problem in Learning Classifier Systems: A Preliminary Study. In *Proceedings of the 2005 Workshops on Genetic and Evolutionary Computation (Washington, D.C., June 25 - 26, 2005)*, pages 74–78, New York, NY, 2005. ACM Press.

L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi. Graph Kernels for Chemical Informatics. *Neural Netw.*, 18(8):1093–1110, 2005.

J.W. Raymond and P. Willett. Effectiveness of Graph-Based and Fingerprint-Based Similarity Measures for Virtual Screening of 2D Chemical Structure Databases. *J. Comput.-Aided Mol. Des.*, 16:59–71, 2001.

Razinger. Extended Connectivity in Chemical Graphs. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretical Chimica Acta)*, 61:581–586, 1982. doi: 10.1007/BF02394734. URL http://dx.doi.org/10.1007/BF02394734.

David Rogers and Robert D. Brown. Using Extended-Connectivity Fingerprints with Laplacian-Modified Bayesian Analysis in High-Throughput Screening Follow-Up. *Journal of Biomolecular Screening*, 10:682–686, October 2005. doi: 10.1177/1087057105281365. URL http://jbx.sagepub.com/cgi/content/abstract/10/7/682.

H. Shin and S. Cho. How to Deal With Large Datasets, Class Imbalance and Binary Output in SVM Based Response Model. In *Proceedings of the Korean Data Mining Conference*, pages 93–107, 2003. Best Paper Award.

R.L. Strausberg and S.L. Schreiber. From Knowing To Controlling: A Path From Genomics To Drugs Using Small Molecule Probes. *Science*, 300:294–295, 2003. URL http://chembank.broad.harvard.edu/.

S. J. Swamidass, J. H. Chen, J. Bruand, P. Phung, L. Ralaivola, and P. Baldi. Kernels for Small Molecules and the Predicition of Mutagenicity, Toxicity, and Anti-Cancer Activity. *Bioinformatics*, 21(Supplement 1):i359–368, 2005. Proceedings of the 2005 ISMB Conference.

S.J. Swamidass and P. Baldi. Bounds and Algorithms for Exact Searches of Chemical Fingerprints in Linear and Sub-Linear Time. *Journal of Chemical Information and Modeling*, 47 (2):302–317, 2007.

J.-F. Truchon and C. I. Bayly. Evaluating Virtual Screening Methods: Good and Bad Metrics for the "Early Recognition" Problem. *J. Chem. Inf. Model.*, 47(2):488 –508, 2007.

K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the Sensitivity of Support Vector Machines. In *Proceedings of the International Joint Conference on AI*, pages 55–60, 1999.

D. Weiniger, A. Weiniger, and J.L. Weiniger. SMILES. 2. Algorithm for Generation of Uniques SMILES Notation. *J. Chem. Inf. Comput. Sci.*, 29:97–101, 1989.