Pre-Processing Structured Data for Standard Machine Learning Algorithms by Supervised Graph Propositionalization - a Case Study with Medicinal Chemistry Datasets

Thashmee Karunaratne
Dept. of Computer and Systems Sciences,
Stockholm University Forum 100,
SE-164 40 Kista, Sweden
si-thk@dsv.su.se

Henrik Boström
Dept. of Computer and Systems Sciences,
Stockholm University Forum 100,
SE-164 40 Kista, Sweden
henrik.bostrom@dsv.su.se

Ulf Norinder
AstraZeneca R&D, Södertälje,
Sweden
ulf.norinder@astrazeneca.com

***Abstract* - Graph propositionalization methods can be used to transform structured and relational data into fixed-length feature vectors, enabling standard machine learning algorithms to be used for generating predictive models. It is however not clear how well different propositionalization methods work in conjunction with different standard machine learning algorithms. Three different graph propositionalization methods are investigated in conjunction with three standard learning algorithms: random forests, support vector machines and nearest neighbor classifiers. An experiment on 21 datasets from the domain of medicinal chemistry shows that the choice of propositionalization method may have a significant impact on the resulting accuracy. The empirical investigation further shows that for datasets from this domain, the use of the maximal frequent item set approach for propositionalization results in the most accurate classifiers, significantly outperforming the two other graph propositionalization methods considered in this study, SUBDUE and MOSS, for all three learning methods.**

Key words – structured data, graph propositionalization, random forests, support-vector machines, k-nearest neighbor, medicinal chemistry

I. INTRODUCTION

Standard machine learning algorithms require data to be represented in an attribute-value format, where the examples are represented using feature vectors of fixed length [1]. However, in many cases, relevant information cannot easily be represented in a single table, since relations between the attributes are frequently occurring [2]. Therefore, in order to apply standard learning algorithms in such domains, either the relational aspects have to be omitted or some suitable transformation of the relations into an attribute-value format is required. To avoid loss of information, the latter approach should be adopted if possible, hence requiring an appropriate preprocessing method for including the structural information into the attribute-value format.

Propositionalization concerns enabling propositional (attribute-value) learners to handle relational learning problems [3]. A typical propositionalization method results in a feature vector representation, where the features encode relations [4]. A relational learner based on propositionalization first computes a set of propositional features and then uses a (standard) machine learning algorithm for model generation [5]. The decoupling of feature generation and model development hence enables standard learning algorithms to be used also for relational learning [3]. Graph propositionalization methods are a subclass of these methods that consider examples represented as graphs and in some way convert these into fixed-length feature vectors. Several approaches for generating classification models through propositionalization have been proposed in the past [3,4,5,6,7]. The propositionalization methods are usually embedded in discovery, prediction or classification methods, and are therefore seldom evaluated separately as pre-processing methods for standard learning algorithms. In particular, it has so far not been carefully investigated in what way the predictive performance is affected by the choice of propositionalization method to use together with some selected standard machine learning algorithm. This study aims to shed more light on this question.

Commonly employed datasets for evaluating machine learning methods, such as those in the UCI repository [9], mostly concern non-relational problems and are hence not useful in large-scale evaluations of different graph propositionalization methods. In this study, we instead turn to the field of medicinal chemistry, in particular to the prediction of biological effects of chemical compounds. Several datasets from this domain are publicly available and they concern modeling tasks of various biochemical activities, such as absorption, distribution, metabolism, excretion and toxicity (ADMET) [10]. What is particularly interesting with these datasets from our perspective is that chemical compounds are naturally represented as graphs, and hence they could serve as a test bed for propositionalization methods. In this study, we will empirically evaluate propositionalization methods when used in conjunction with different standard machine learning algorithms on datasets from the domain of medicinal chemistry.

The rest of the paper is organized as follows. In the next section, the three graph propositionalization approaches that will be evaluated in this study are presented. In section three, the experimental setup for the evaluation of these methods in the domain of medicinal chemistry is presented. In section four, the experimental results are shown and analyzed, and finally, the main conclusions and directions for future work are given in section five.

II. GRAPH PROPOSITIONALIZATION METHODS

Graphs are natural data structures for structured or relational data, where nodes represent objects and edges represent relations. For example, a dataset containing web data may be represented as a set of graphs, where the nodes represent web pages and the edges represent links between the web pages. The graph propositionalization methods we consider in this study assume that nodes and edges are labeled, e.g., a node representing a web page could be labeled ‘student’, while an edge representing a link could be labeled ‘title’. We follow the graph propositionalization approach in [6], by which each example is represented by a set of triples on the form (L_i,L_j,E_k), such that there is an edge, which has a label E_k in the graph of the example between nodes i and j that are labeled L_i and L_j respectively. Following [6], such a set is referred to as a *fingerprint*. We further assume that each fingerprint has been assigned a class label, e.g., ‘positive’ or ‘negative’.

The graph propositionalization methods considered in this study generate feature vectors where each feature corresponds to a sub-graph. We have selected two state-of-the-art methods for sub-graph mining; SUBDUE [11] and MOSS [12]. SUBDUE is a generic method for sub-graph discovery, while MOSS is a domain-specific sub-graph discovery method for chemo-informatics. In this study, we also consider another domain-independent graph propositionalization method that uses the algorithm for maximal frequent itemset mining [13,14] on the fingerprint representation described above. All these three methods, which are described below, may take labeled graphs as input.

SUBDUE – SUBDUE [11] is a graph-based knowledge discovery system that finds structural and relational patterns in data representing entities and relationships. SUBDUE represents relational data using labeled, directed graphs, where the entities are represented by labeled vertices or sub-graphs, and relationships are represented by labeled edges between the entities [11]. It uses the minimum description length (MDL) principle to measure the interestingness of the sub-graphs discovered. SUBDUE employs a step-by-step procedure, which starts from single nodes and performs a computationally constrained beam search in order to expand it by another node or edge, which results in a sub-graph. Therefore it typically generates a small number of sub-graphs that best compress the dataset. We have chosen SUBDUE as a reference method, since it is publicly available and the sub-graphs discovered by SUBDUE could be used as (propositional) features for standard machine learning algorithms. We have used the implementation of SUBDUE described in [15]. In order to discover sub-graphs efficiently, we have used the parameters *eval=2* and *prune*, as described in [15].

MOSS – MOSS [12] is an algorithm that can be used for discovering frequent molecular fragments in chemo-informatics databases. In doing so, MOSS also views molecules as graphs, where the atoms are the nodes and the bonds between atoms are the edges of the graph. The molecular fragment discovery algorithm of MOSS discovers molecular fragments that are referred to as sub-graphs, which best separate molecules that belong to different classes. The MOSS algorithm searches for arbitrarily connected sub-graphs, avoiding frequent embeddings of previously discovered sub-graphs by using a specific search strategy. The algorithm maintains parallel embeddings of a fragment into all molecules throughout the growth process and exploits a local order of the atoms and bonds of a fragment to effectively prune the search tree, which allows for a restricted depth-first search algorithm, similar to the Eclat association rule mining algorithm [12]. MOSS selects substructures that have a certain minimum support in a given set of molecules, i.e., are part of at least a certain percentage of the molecules. However, in order to restrict the search space, the algorithm considers only connected substructures, i.e., sub-graphs containing nodes that are connected to at least one other node. Similar to SUBDUE, MOSS has been included as a reference method, since it is publicly available and appears to be suitable in particular for the medicinal chemistry domain, where the frequent fragments discovered by MOSS can be used as features. The publicly available implementation of MOSS is used in our experiments together with the default parameter settings [16].

Supervised maximum frequent itemsets (SMFI) – The popular Apriori algorithm for finding association rules was first introduced in [17]. This algorithm is based on analyzing common item sets in a set of transactions, which is technically referred to as frequent itemset mining. The search space of this problem, which is all frequent item sets, is typically huge. Several approaches to efficiently explore the search space have been proposed that only consider a sufficiently large subset of all frequent itemsets. Among these, methods for finding the set of all closed or maximal itemsets are the most popular [18]. Given below is the definition of frequent itemset mining and maximal frequent itemset mining [13,14]:

Definition: *maximal frequent itemset*: Suppose we are given a set of items *I*, which is a set of literals. An itemset $I \subseteq \mathcal{I}$ is some subset of items. For itemset $\mathcal{I}$, a transaction including *I* is called an occurrence of *I*. The *denotation* of *I*, denoted by *T(I)* is the set of the occurrences of *I*. $|T(I)|$ is called the frequency of *I*; and denoted by *frq(I)*: For a given constant θ, called a *minimum support*, itemset *I* is frequent if $frq(I) \geq \theta$. If a frequent itemset *I* is included in no other frequent itemset, *I* is called *maximal*.

When adapting this method for feature selection, a triplet (L_i,L_j,E_k) described above is considered as an item in the itemset and a fingerprint is considered as a single transaction. Therefore, the maximal frequent item set $I_l \subseteq \mathcal{I}_l$, is defined in such a way that $\mathcal{I}_l$ is the collection of all triples in the input graphs with respect to the class *l*. The maximal frequent itemset with respect to class *l* is generated using maximal frequent item set mining algorithm as defined in [14]. The collection of all the discovered maximum frequent itemsets $\sum_{1}^{n} I_l$ with respect to each class are taken together as the feature set discovered by the method. This approach is an extension to the unsupervised way of finding maximal frequent itemsets in [19], which considered all the input graphs in the maximal frequent itemset mining algorithm irrespective of the class.

In the experiments, we have used the maximal frequent itemset mining algorithm as implemented in MAFIA [18], together with 6 levels of support, namely 0.5, 0.4, 0.2, 0.1, 0.025 and 0.01. SMFI does not have an inbuilt pruning method, similar to the MDL principle employed in SUBDUE or minimum support employed in MOSS. Instead, discrete levels of support are applied, which may not be fully optimal. The training set is partitioned into subsets according to class membership and the maximal frequent itemset mining algorithm is applied to each of these subsets of examples separately. The discovered itemsets with respect to all the classes are combined together as the feature set. 10-fold cross-validation is employed on the training set to investigate which of the above parameter setting gives the highest accuracy, and this level of support is used for generating a model from the entire training set.

III. EXPERIMENTAL SETUP

A. Methods

The feature sets generated by each of the three propositionalization methods, SUBDUE, MOSS and SMFI, which were described in the previous section, are used in conjunction with three popular machine learning algorithms, as implemented in the WEKA data mining toolkit [20], namely random forests (RF) [21], support vector machines (the SMO algorithm) [22] and the k-nearest neighbor algorithm [23]. The number of trees generated by the random forest algorithm is set to 50. The parameters for the SMO algorithm are the RBF kernel with complexity 2. The IBk algorithm with the number of nearest neighbors k = 3 is used as the nearest neighbor classifier. We consider accuracy as the performance criterion, which is estimated using 10-fold cross-validation.

B. Data sets

21 publicly available compound sets from the domain of medicinal chemistry [24] are used to test the null hypothesis, i.e., there is no difference in performance between the propositionalization methods. The compounds in each set have associated binary class labels, representing whether a molecule is active or not with respect to some measure of biological activity. The compounds in these datasets are represented by graphs, i.e., each atom in a molecule is a node of the corresponding graph and the arcs represent bonds among atoms.

It has been suggested in the past, e.g., in [3,25], that using the domain specific feature sets in conjunction with the machine learning approaches could help increasing the classifier accuracy. Therefore in addition to the features generated by the propositionalization methods, we also have used publicly available chemical descriptors as input features in conjunction with the generated features. The SELMA [26] descriptors consist of 94 different global properties of molecules and the Scitegic Extended Connectivity Fingerprints (ECFI) [27] descriptors consist of presence and non-presence of 1024 different molecular fragments. These descriptors are represented in attribute-value format and they have been used in conjunction with the feature sets generated by the three graph propositionalization methods.

IV. EXPERIMENTAL RESULTS

The results from employing 10-fold cross-validation on the 21 data sets using the standard random forest algorithm, support vector machines and the nearest neighbor classifier in conjunction with the three different propositionalization methods are shown in Table I, II and III below. Each of the three tables corresponds to one of the learning algorithms, and the names of the data sets are given in the first column of each table. The subsequent columns contain results for the three propositionalization methods, when these features are used on their own (generated features only), when they are used in conjunction with SELMA descriptors (features+selma), and together with the ECFI descriptors (features+ecfi), respectively.

A. Hypothesis tests

In testing the null hypothesis, the predictive performances of the three learning algorithms are analyzed separately. For each algorithm, the results of different feature sets, i.e., generated features only, features+selma and features+ecfi, are also analyzed separately. Therefore the same null hypothesis is tested in nine cases. For each case, the significance of the differences of the accuracies yielded by the three propositionalization methods is tested using the Friedman test [28]. The accuracies of the three methods are ranked and the average ranks are used for pair-wise tests according to [28]. A propositionalization method that fails to produce a feature set is assigned the lowest rank. Table IV gives the ranking of each method for all the nine cases. A single case corresponds to one feature set given in column 1 and one learning algorithm given in column 3, 4 or 5. Therefore the ranks given in Table IV are the relative performance of SMFI, SUBDUE and MOSS for the respective feature set and classifier algorithm.

Table IV: AVERAGE RANKINGS OF THE THREE METHODS

Method		**RF**	**SMO**	**IBk**
Features only	smfi	1.24	1.10	1.30
	subdue	2.14	2.24	2.10
	moss	2.62	2.67	2.62
Features+ Selma	smfi	1.50	1.60	1.64
	subdue	2.21	2.26	1.76
	moss	2.29	2.14	2.60
Features+ecfi	smfi	1.83	1.74	1.64
	subdue	1.83	2.14	1.76
	moss	2.33	2.12	2.60

The null hypothesis is not rejected when the generated features are used with the ECFI descriptors in conjunction with random forests and support vector machines. The pairs of methods for which one can conclude that there is a significant difference of ranks is summarized in Table V below. The winning method is given in boldface in Table V.

TABLE V: RESULTS OF PAIR-WISE STATISTICAL TESTS FOR THE THREE PROPOSITIONALIZATION METHODS

Classifier	**features only**	**features+selma**	**features+ecfi**
RF	**smfi** vs subdue **smfi** vs moss	**smfi** vs subdue **smfi** vs moss	Null hypothesis is not rejected.
SMO	**smfi** vs subdue **smfi** vs moss	**smfi** vs subdue **smfi** vs moss	Null hypothesis is not rejected.
IBk	**smfi** vs subdue **smfi** vs moss **subdue** vs moss	**smfi** vs moss **subdue** vs moss	**smfi** vs moss **subdue** vs moss

It is interesting to note that the classifier accuracy for SMFI is significantly better than for SUBDUE and MOSS, when the features generated by the methods are considered alone in the classification (features only). The differences between the methods are not as emphasized when they are considered in conjunction with existing chemical descriptors. In particular, the differences are not significant at all when the ECFI descriptors are used with random forests and SMO.

The method SMFI was able to discover at least one important feature with respect to all the 21 datasets, whereas SUBDUE and MOSS fails to do so for some datasets. It is worth noting that MOSS could not discover any feature for 6 out of 21 datasets, which is also reflected by its low performance. One reason for the inability to generate features in some cases could be that MOSS considers only sub-graphs that are connected. In contrast, the discovered patterns of SUBDUE could contain even a single node, while the feature discovery approach of SMFI naturally allows discovering sets of sub-graphs that are not connected. For example, a feature discovered by SMFI such as {S-c, c-O, c-c, c-N, c-c}, which is a collection of molecular fragments, could either be inter-connected or disconnected. The other two methods will not include such a set of sub-graphs in one feature. The success of SMFI indicates that features of this type may be quite important for the type of datasets considered in this study. It is also worthwhile to note that SMFI is a naïve kernel which simply computes the average kernel between all pairs of edges. [8] suggests that such a naïve kernel could perform comparable to sophisticated kernel methods.

V. CONCLUDING REMARKS

Graph propositionalization methods can be used as pre-processing methods for standard machine learning algorithms, transforming structured data into fixed-length feature vectors. In this study, three propositionalization methods, MOSS, SUBDUE and SMFI, have been studied in conjunction with three standard machine learning algorithms, namely random forests, support vector machines and nearest neighbor classifiers. An empirical evaluation on 21 data sets from the domain of medicinal chemistry showed that the choice of propositionalization method may have a significant impact on the resulting accuracy. The experiment further showed that the SMFI method outperformed the other two graph propositionalization methods, SUBDUE and MOSS, for all the three learning algorithms that were considered in this study.

As a further improvement to the graph propositionalization using SMFI, one could employ a suitable technique for finding the optimal threshold for the mfi algorithm in [18]. This contrasts to the approach employed in this study, which selects parameter values from a finite set based on cross-validation on the training set. It could also be interesting to investigate the classifier performance with different feature sets merged in one feature vector. For example, different molecular descriptors, features discovered by subgraphs, kernels, ILP or boosting methods could be used in one such feature vector. Another direction for future work, which would be interesting from a medicinal chemistry perspective, is to compare the predictive performance of the classifiers obtained from using only the existing molecular descriptors to those generated from the extended feature sets, and investigate to what extent QSAR modeling with the standard molecular descriptors could be improved by using them in conjunction with graph propositionalization methods. One other possible direction for further studies is to carry out a similar experiment on data from other domains, such as web document classification. One could also combine different feature sets derived by different feature discovery methods and investigate in which way the combinations of such feature set affect the performance.

REFERENCES

[1]. Eibe Frank, Mark A.Hall, Geoffrey Holmes, Richard Kirkby, Bernhard Pfahringer, and I. H. Witten, "Weka: A machine learning workbench for data mining", *Data Mining and Knowledge Discovery Handbook,* 978-0-387-24435-8© Springer, Berlin, 2005

[2]. Nicolas Lachiche, "Good and Bad Practices in Propositionalisation". *AI*IA 2005*: 3673. Springer 2005, ISBN 3-540-29041-9 50-61

[3]. Kramer, S. and De Raedt, L., Feature construction with version spaces for biochemical applications, *Proc. 18th Int. Conf. Machine Learning*, 258-265, 2001

[4]. S.Kramer, N. Lavrač and P Flach, "Propositionalization approaches to relational data mining", In S. Dĕzeroski, Ed. *Relational Data Mining*, Springer-Verlag New York, 2000, 262-286

[5]. Mark-A. Krogel, Simon Rawles, Filip Železný, Peter A. Flach, Nada Lavrač and Stefan Wrobel, "Comparative Evaluation of Approaches to Propositionalization", *Inductive Logic Programming*, Lecture Notes in Computer Science Volume 2835/2003, 2003

[6]. T. Karunaratne and H. Boström, "Learning to Classify Structured Data by Graph Propositionalization", Proceedings of the Second *IASTED International Conference on Computational Intelligence*, 2006, 393-398

[7]. S. Kramer, "Relational learning Vs. propositionalization", *AI communications*, ISSN 0921-7126, 215-281, IOS press

[8]. S. V. N. Vishwanathan, Nicol N. Schraudolph, Imre Risi Kondor, and Karsten M. Borgwardt. Graph Kernels. *Journal of Machine Learning Research*, 11:1201–1242, April 2010

[9]. C. Blake and C. Merz. UCI repository of machine learning databases, 1998

[10]. Terry R. Stouch, James R. Kenyon, Stephen R. Johnson, Xue-Qing Chen, Arthur Doweyko & Yi Li, in silico ADME/Tox: why models fail, *Journal of computer aided Molecular Design*, 17: 83-92, 2003, Kluwer

[11]. L. B. Holder and D. J. Cook, Graph-based Data Mining, J. Wang (ed.), Encyclopedia of Data Warehousing and Mining, Idea Group Publishing, 2005

[12]. Mining Molecular Fragments: Finding Relevant Substructures of Molecules Christian Borgelt and Michael R. Berthold *IEEE International Conference on Data Mining* (ICDM 2002, Maebashi, Japan), 51-58 IEEE Press, Piscataway, NJ, USA 2002

[13]. B. Goethals and M. J. Zaki. "FIMI'03: Workshop on frequent itemset mining implementations. In B. Goethals and M. J. Zaki, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI2003), volume 90 of CEUR Workshop Proceedings, Melbourne, Florida, USA, 19 November 2003

[14]. T. Uno, M. Kiyomi, and H. Arimura. LCM ver. 2: efficient mining algorithms for frequent/closed/maximal itemsets. In B. Goethals, M. J. Zaki, and R. Bayardo, editors, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI 2004), volume 126 of CEUR Workshop Proceedings, Brighton, UK, 1 November 2004

[15]. SUBDUE-Graph based Knowledge Discovery, http://ailab.wsu.edu/subdue/ Last visited: 15/04/2010

[16]. MoSS- Molecular substructure miner, http://www.borgelt.net/moss.html Last visited: 15/04/2010

[17]. R. Agrawal and R. Srikant. "Fast algorithms for mining association rules". In 20th VLDB Conference, Sept. 1994, 487-499

[18]. Doug Burdick, Manuel Calimlim and Johannes Gehrke., "MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Database" *In Proceedings of the 17th International Conference on Data Engineering*, Heidelberg, Germany, April 2001

[19]. T. Karunaratne and H. Boström, "Graph Propositionalization for random forests", Proceedings of the 8th *International Conference on Machine Learning and Applications (ICMLA)*, 2009

[20]. Ian H. Witten and Eibe Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, USA, 2005

[21]. Leo Breiman, Random Forests. *Machine Learning*. 45(1):5-32, 2001

[22]. J. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization, In B. Schoelkopf and C. Burges and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, 1998

[23]. D. Aha, D. Kibler, Instance-based learning algorithms, *Machine Learning*. 6:37-66, 1991

[24]. Data Repository of Bren School of Information and Computer Science, University of California, Irvine, ftp://ftp.ics.uci.edu/pub/baldig/learning/QSAR_Sutherland/, 2005 Last visited: 17/08/2009

[25]. Matthias Rupp, Timon Schroeter, Ramona Steri, Heiko Zettl, Ewgenij Proschak, Katja Hansen, Oliver Rau, Oliver Schwarz, Lutz Müller-Kuhrt, Manfred Schubert-Zsilavecz, Klaus-Robert

Müller, Gisbert Schneider, From Machine Learning to Natural Product Derivatives that Selectively Activate Transcription Factor PPARγ, *ChemMedChem*, 5(2), pp 191 – 194, 2009, WILEY-VCH Verlag, Weinheim

[26]. Olsson, T., Sherbukhin, V., SELMA, Synthesis and Structure Administration (SaSA), AstraZeneca R&D Mölndal, Sweden

[27]. David Rogers and Mathew Hahn, Extended-Connectivity Fingerprints, *J. Chem. Inf. Model.* 2010, *50,* 742–754

[28]. Salvador Garcia and Francisco Herrera, An Extension on. “Statistical comparisons of classifiers over multiple data sets” for all Pairwise Comparisons, *Journal of Machine Learning Research*, 9:2677-2694, 2008

TABLE I. CLASSIFICATION ACCURACIES FOR RANDOM FORESTS

Name	generated features only			features+selma			features+ecfi		
	sm fi	sub due	mo ss	sm fi	sub due	mo ss	sm fi	sub due	mo ss
AI	.70	.52	.51	.75	.75	.76	.77	.80	.80
AMPH1	.50	.45	.47	.52	.53	.58	.56	.55	.58
ATA	.47	.52	*	.66	.62	*	.77	.76	*
caco	.62	.47	.33	.82	.78	.68	.69	.65	.63
COMT	.73	.60	*	.74	.72	*	.79	.82	*
EDC	.60	.54	.50	.82	.78	.76	.70	.72	.69
HIVPR	.61	.48	.51	.82	.79	.81	.72	.74	.73
HIVRT	.70	.55	.50	.87	.86	.87	.82	.82	.80
HPTP	.81	.71	.51	.88	.86	.87	.87	.87	.85
dbp	.90	*	*	.90	*	*	.92	*	*
nct	.68	.69	.68	84	.85	.86	.84	.86	.86
ace	.79	.76	.55	.89	.88	.87	.87	.87	.88
ache	.68	.57	.81	.75	.74	.73	.65	.63	.68
bzr	.70	.52	.62	.77	.76	.75	.73	.74	.74
chang	.81	.78	*	.78	.70	*	.82	.76	*
gpb	.75	.72	.50	.73	.75	.75	.79	.79	.77
silverman	.63	*	*	.71	*	*	.73	*	*
therm	.74	.69	*	.63	.64	*	.73	.72	*
thr	.62	*	.50	.73	*	.73	.71	*	.69
cox2	.56	.58	.50	.78	.76	.77	.73	.69	.72
dhfr	.60	.57	*	.86	.86	*	.88	.89	*

*No features were discovered by the method

TABLE II. CLASSIFICATION ACCURACIES FOR SUPPORT VECTOR MACHINES

Name	generated features only			features+selma			features+ecfi		
	smf i	sub due	mo ss	sm fi	sub due	mo ss	sm fi	su bd ue	mo ss
AI	.64	.46	.51	.81	.81	.81	.70	.70	.71
AMPH1	.52	.49	.47	.53	.55	.56	.53	.53	.54
ATA	.51	.49	*	.77	.75	*	.70	.69	*
caco	.53	.50	.40	.65	.63	.59	.66	.70	.63
COMT	.73	.60	*	.80	.80	*	.75	.71	*
EDC	.50	.51	.49	.74	.73	.74	.77	.75	.76
HIVPR	.55	.47	.51	.73	.73	.73	.77	.76	.77
HIVRT	.59	.53	.50	.79	.79	.79	.78	.78	.76
HPTP	.77	.72	.50	.87	.87	.87	.83	.81	.83
dbp	.88	*	*	.91	*	*	.89	*	*
nct	.69	.67	.59	.85	.85	.86	.77	.73	.75
ace	.79	.76	.47	.90	.90	.90	.83	.83	.85
ache	.61	.54	.72	.66	.64	.67	.68	.67	.69
bzr	.66	.51	.51	.73	.74	.75	.73	.74	.75
chang	.68	.60	*	.78	.78	*	.82	.72	*
gpb	.69	.66	.46	.80	.80	.80	.75	.63	.62
silverman	.52	*	*	.76	*	*	.60	*	*
therm	.65	.58	*	.77	.77	*	.70	.72	*
thr	.47	*	.46	.70	*	.70	.54	*	.55
cox2	.56	.54	.50	.74	.72	.73	.76	.73	.74
dhfr	.59	.54	*	.88	.88	*	.83	.83	*

TABLE III. CLASSIFICATION ACCURACIES FOR THE NEAREST NEIGHBOR CLASSIFIER

Name	generated features only			features+selma			features+ecfi		
	smf i	sub due	mo ss	sm fi	sub due	mo ss	sm fi	sub due	mo ss
AI	.70	.48	.51	.76	.75	.72	.84	.84	.85
AMPH1	.50	.45	.50	.55	.55	.54	.57	.58	.55
ATA	.46	.53	*	.63	.64	*	.55	.55	*
caco	.59	.50	.45	.60	.68	.56	.60	.60	.59
COMT	.73	.60	*	.73	.72	*	.78	.78	*
EDC	.62	.54	.50	.77	.77	.76	.71	.70	.69
HIVPR	.63	.48	.50	.69	.69	.72	.64	.64	.64
HIVRT	.70	.55	.49	.78	.77	.77	.72	.72	.72
HPTP	.78	.71	.50	.85	.85	.84	.83	.83	.81
dbp	.89	*	*	.90	*	*	.81	*	*
nct	.70	.71	.61	.68	.67	.67	.80	.78	.79
ace	.79	.76	.62	.84	.86	.85	.87	.86	.87
ache	.65	.57	.73	.64	.65	.69	.67	.66	.69
bzr	.69	.54	.61	.69	.70	.68	.62	.59	.59
chang	.85	.84	*	.75	.77	*	.78	.78	*
gpb	.79	.69	.59	.71	.71	.63	.71	.71	.75
silverman	.63	.53	*	.70	.66	*	.59	.66	*
therm	.77	.68	*	.61	.57	*	.66	.69	*
thr	.61	*	.50	.65	*	.66	.56	*	.57
cox2	.54	.58	.50	.68	.68	.68	.67	.66	.66
dhfr	.59	.59	*	.81	.82	*	.83	.83	*