Prediction of Metabolic Transformations using Cross Venn-ABERS Predictors

Staffan Arvidsson STAFFAN.ARVIDSSON@FARMBIO.UU.SE
Ola Spjuth OLA.SPJUTH@FARMBIO.UU.SE
Department of Pharmaceutical Biosciences, PO Box 591, SE-751 24 Uppsala, Sweden

Lars Carlsson LARS.A.CARLSSON@ASTRAZENECA.COM
Quantitative Biology, Discovery Sciences, Innovative Medicines & Early Development, AstraZeneca

Paolo Toccaceli PAOLO.TOCCACELI@RHUL.AC.UK
Department of Computer Science, Royal Holloway, University of London

Editors: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, and Harris Papadopoulos

Abstract

Prediction of drug metabolism is an important topic in the drug discovery process, and we here present a study using probabilistic predictions applying Cross Venn-ABERS Predictors (CVAPs) on data for site-of-metabolism. We used a dataset of 73599 biotransformations, applied SMIRKS to define biotransformations of interest and constructed five datasets where chemical structures were represented using signatures descriptors. The results show that CVAP produces well-calibrated predictions for all datasets with good predictive capability, making CVAP an interesting method for further exploration in drug discovery applications.

Keywords: Venn-ABERS, Cross Venn-ABERS Predictor, Site-of-Metabolism, Drug discovery, Machine Learning, Support Vector Machine, Signatures descriptor, QSAR

1. Introduction

Computational methods are widespread throughout the various parts of the drug discovery process. One important component is predictive modeling of experiments on chemical compounds, where the chemical structure of the compound is represented numerically (algorithms for such representations are called 'descriptors' in the field of cheminformatics). The descriptor values constitute the input data, and the dependent variable is the measured value from a biological experiment (e.g. activity, toxicity or inhibition). This methodology is referred to as QSAR (Quantitative Structure-Activity Relationships)(Hansch, 1969), and the objective is normally to predict the outcome for an unseen chemical structure, to provide e.g. an early warning, indication of potential problems, or an estimate of the effect the compound might have on a protein target (receptor). The learning methods used include SVM (Burbidge et al., 2001) and Random Forest (Svetnik et al., 2003). QSAR models are commonly used in the early stages of drug discovery, for example to provide decision aid about carcinogenicity (Helma, 2006), toxicity (Spycher et al., 2008), solubility (Johnson et al., 2007) and for ADME (Absorption, Distribution, Metabolism, Excretion) profiling (Munteanu et al., 2010; Gedeck and Lewis, 2008).

Drug metabolism is an important topic in drug discovery, as the rate of metabolism determines the duration and intensity of a drug's pharmacologic action. Another reason is that the compounds formed in the metabolic processes might be biologically active themselves, so called reactive metabolites. It is therefore crucial to understand how drugs are processed by the body in order to excrete them.

One approach is to predict which sites in a molecule are likely to be where the body starts to modify the compound, so called site-of-metabolism (SOM) (Rydberg et al., 2010; Carlsson et al., 2010). These methods in many cases build on a knowledge base of established biotransformations and use data mining approaches to predict SOM for a query compound. To define reaction centers in reactions one can use Maximum Common Substructure (MCS) searches like in the MetaPrint2D method (Carlsson et al., 2010), or using SMIRKS (DAYLIGHT, 2008). In both cases, differences between the query molecule and the metabolite are used to define type of reaction and reaction centers.

Most contemporary approaches in QSAR do not report valid confidence measures or class probabilities. In some cases, confidence in predictions is very valuable, where in other cases knowing the probabilities of e.g. the predicted classes is desirable. Conformal Prediction is a statistical learning theory proposed by Vovk et. al (Vovk et al., 2005), where predictions incorporate a valid indication of their own accuracy and reliability. Conformal Prediction has recently been introduced in the QSAR field (Norinder et al., 2014) offering a compelling alternative to the concept of applicability domain. However, studies on the use of probabilistic prediction with e.g. Venn-ABERS Predictors have so far not been extensively described.

The aim of this paper is to evaluate the applicability of Cross Venn-ABERS Predictors (CVAPs) within the field of predictive metabolism. To the best of our knowledge there does not exist any previous studies using CVAPs on SOM predictions, making it an interesting case study. If CVAP can produce well-calibrated results, it would be useful in the drug discovery process, making it possible to make more informed decisions and weigh decision making based in found risks and costs. One of the potential use cases would be SOM of potential drug candidates, were a predictor can be trained for each biotransformation type of interest and the probability for each transformation can be found. Thus giving the possibility to find candidates that might be degraded too rapidly or result in e.g. toxic metabolites, and potentially avoid further expensive in vitro or in vivo test of poor candidates.

2. Methods

2.1. Data

Data was extracted from the 2005 version of MDL Metabolite Database (Elsevier MDL (2005)), containing 73599 chemical reactions describing biotransformations. Each record contains an experimentally determined biotransformation, mapping a substrate to a product (or compound to metabolite). Experiments are performed in different experimental settings, i.e. in rabbit, mouse, in vitro or human. A single substrate can be tested in several settings and have multiple resulting products, possibly due to different experimental settings.

Preprocessing The preprocessing needed prior to training and evaluation of the CVAPs is outlined in Figure 1, with reaction types defined using SMIRKS.

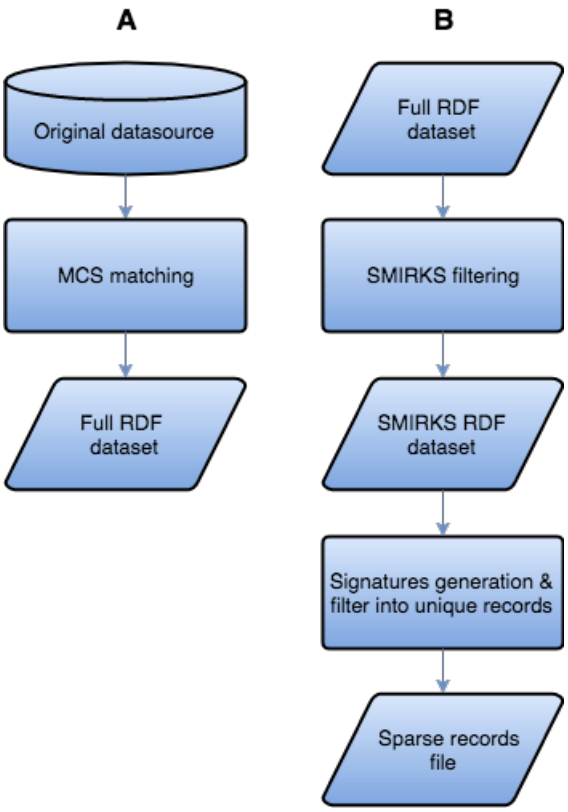

Figure 1: The workflow for preprocessing of data. Workflow **A** (left) was performed once for the complete dataset, calculating the Maximum Common Substructure (MCS) for each biotransformation. MCS performs an exhaustive search for finding the atom-atom mapping between two chemical compounds. This calculation is computationally very costly but it is only required to be performed once. Workflow **B** (right) was performed once for every SMIRKS of interest and results in a numerical dataset that can be used for training and validating the CVAPs. SMIRKS filtering was performed using the pseudo code in Algorithm 1. The SMIRKS dataset was then converted to a numerical dataset by using signatures descriptors (Faulon et al., 2003). There is also a filtration needed as some substrates are present in multiple biotransformations, the filtration is performed by keeping response value 1 in case there is any record having a response value of 1, otherwise the response value 0 is kept.

SMIRKS is a language used for describing chemical reaction transformations in a generic way (DAYLIGHT, 2008). The basic syntactics of SMIRKS is written on the form `substrate >> product`, where `substrate` and `product` are generic representations of the reaction centers of a substrate and product (in our case a compound and its metabolite). As an illustrative example, see Figure 2, where two biotransformations are matched against the SMIRKS `[$([c:1])]>>[c:1][OH]`, which express a hydroxylation of an aromatic carbon atom.

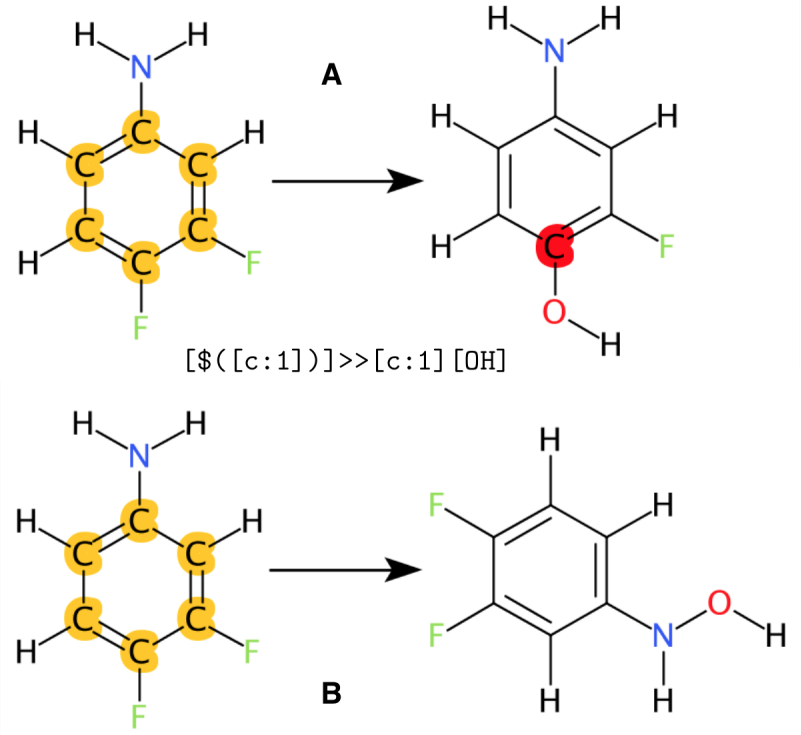

Figure 2: SMIRKS matching for two biotransformations **A** and **B**. In both transformations the substrate part of the SMIRKS, `$([c:1])`, matches every carbon atom that is aromatic, i.e. part of a ring structure, and they are highlighted in yellow. Only transformation **A** also has a match in the product part of the SMIRKS, `[c:1][OH]`, representing an aromatic carbon atom covalently bound to a hydroxy group, atom highlighted in red. Biotransformation **A** thus has a complete SMIRKS match, giving a response value of 1, whereas transformation **B** only match in the substrate part, giving a response value of 0.

Processing was split up in two steps, step A was performed once for the complete dataset and could then be reused for all later computations. Step B was then performed once for each SMIRKS of interest. Step A involves the computationally demanding task of finding the Maximum Common Substructure (MCS) for every biotransformation in the original dataset, trying to map individual atoms in the substrate to the same preserved

or altered atoms in the product. MCS was performed using the Isomorphism class in the Chemistry Development Kit (CDK) version 1.5.13 (Steinbeck et al., 2003, 2006), using algorithm CDKMCS.

Step B was performed once for every SMIRKS of interest (five chosen for this paper), where the SMIRKS filtering task relies on the atom-atom mapping generated in the MCS algorithm. The SMIRKS filtering is described in Algorithm 1 and results in a new dataset only comprising the substrate molecules and the found response, which is either 0 (SMIRKS not matching) or 1 (SMIRKS matching). The second task of step B was converting molecule data into numerical data in LibSVM format, which was done by using signatures descriptors (Faulon et al., 2003). This task also filtered the produced records so that every substrate was only represented once in the final dataset. Filtration was done in a fashion so that if there was *any* record having response 1 the final response was set to 1, otherwise the final response was set to 0. The motivation behind this is that several biotransformations can be possible for any given substrate, having evidence of another possible biotransformation should not influence the prediction of the current SMIRKS.

Algorithm 1: SMIRKS filtering - converting a set of biotransformations into only substrates which matches in the substrate part of the SMIRKS and their respective response values

```
Function Filter(reactions, reactionType)
    filteredResult ← empty list
    foreach reaction in reactions do
        singleResult ← FilterReaction(reaction, reactionType)
        if singleResult != null then
            filteredResult.append(singleResult)
        end
    end
    return filteredResult
Function FilterReaction(reaction, reactionType)
    substratePart ← reactionType.getSubstratePart()
    substrate ← reaction.getSubstrate()
    if substrate matches substratePart then
        response ← 0
        if reaction matches reactionType then
            response ← 1
        end
        return (substrate, response)
    end
    else
        return null
    end
```

Five SMIRKS of interest were chosen and picked for analysis in this paper, see Table 1. Four of these datasets were skewed towards a higher representation of class 0, except for one which instead was skewed towards class 1 with a 4:1 ratio.

Table 1: Overview of the data used within this paper. The datasets were predominantly skewed towards class 0 except for the Aromatization dataset where only 25% of the samples were part of class 0.

Dataset name	**SMIRKS**	**Biotransformations**	**Class 0**	**Class 1**
Alkyl hydroxylation	`[$([C:1])]>>[C:1][OH]`	17793	12064 (68%)	5729 (32%)
Aromatic hydroxylation	`[$([c:1])]>>[c:1][OH]`	14691	12476 (85%)	2215 (15%)
Carboxylation	`[$([CH3:1])]>>[C:1](=O)O`	12580	8047 (64%)	4533 (36%)
Oxidation of tertiary amine	`[$([N;X3:1])]>>[N+:1][O-]`	11040	10715 (97%)	325 (3%)
Aromatization	`[$([*;R;!a:1])]>>[a:1]`	9518	2357 (25%)	7161 (75%)

2.2. Algorithms

The algorithms employed in this paper are based on the theory and ideas previously published by Vovk et al. (2015). This section will contain the basics needed for grasping the concepts and making this paper self-contained, mostly following the notations introduced in the cited paper. First we introduce the concept of *observations* $z = (x, y)$, each consisting of an *object* x and a binary *label* $y \in \{0, 1\}$. The labels used throughout this paper are 0="no SMIRKS match" (no reaction present) and 1="SMIRKS match" (reaction present).

Venn Predictor The Venn-ABERS predictor used in this paper is a subclass of Venn predictors. Venn predictors have the desired property of always producing well-calibrated probability predictions (Vovk et al., 2015). By calibration we refer to the following property:

$$\mathbb{P}\{y = 1 | p_{pred}(x) = p\} = p \tag{1}$$

Informally, this means in the long term the relative frequency of objects with the desired property among those with predicted probability p of having that property is indeed p. If we limit ourselves to the binary classification case of the Venn-ABERS predictors used in this paper, two probabilities are output for each test object; one of these two predicted probabilities is the one that is calibrated, but which one it is depends on the test object. While this might seem not helpful, in practice the two probabilities are close enough not to affect the result. If they do differ, then this is diagnostic of inherent uncertainty in the prediction.

It is often far more practical to deal with a point probability rather than multiprobabilities or probability intervals. One principled way to merge the two probabilities p_0 and p_1 is by calculating the combination that minimizes a chosen loss function. In the case of log loss, this occurs for $p = \frac{p_1}{1-p_0+p_1}$. Formally, this point probability no longer enjoys the calibration property of the multi probabilistic prediction; however, experimental evidence (Vovk et al., 2015) suggests that the point predictions still exhibit high accuracy.

Inductive Venn-ABERS Predictor The Inductive Venn-ABERS Predictor (IVAP) is *inductive* in the sense that a model or prediction rule can be built using a batch of training observations and the model or prediction rule can be reused for predicting all test

objects. IVAPs are based on an underlying *scoring algorithm*, which could be any standard machine learning algorithm. In this paper we have used the Support Vector Machine (SVM) implementation SVC in Scikit-learn (Pedregosa et al., 2011) using the Radial Basis Function kernel. The steps needed to train an IVAP are outlined in Algorithm 2, where D_c denotes the *calibration set* and D_p denotes the *proper training set*. The result of a trained IVAP is both the trained underlying algorithm, which is trained on the proper training set, and two arrays of numbers, scores s and true values y for all observations in the *calibration set*.

Algorithm 2: Training an `IVAP`

Input: D, the training dataset with l observations
Result: A trained `IVAP`

1 Split D into two mutually exclusive subsets, D_p and D_c, each with l_p and l_c observations respectively and $l = l_p + l_c$.
2 Train the underlying scoring algorithm on all observations in D_p.
3 Predict all objects from D_c, giving scores $s_1, \ldots, s_{l_c}$.
4 Save the scoring algorithm and the tuples (y_i, s_i) for each calibration observation i, where $i = 1, \ldots, l_c$.

The steps performed when predicting new objects using an IVAP are outlined in Algorithm 3[1]. In Line 3 and 4 isotonic regression is fitted to two series, each assuming either of the two hypothetical labels of the new prediction object. This results in two prediction values produced, p_0 and p_1, respectively. As previously declared, either p_0 *or* p_1 is the true prediction value, but it is not possible to know which one. However, p_0 and p_1 satisfy $p_0 < p_1$ and they can be considered as the lower and upper boundaries of a *probability interval*. The authors of the original paper (Vovk et al., 2015) claim that p_0 and p_1 in practice are close, a claim that was confirmed herein where the mean interval width was ranging between 0.014 and 0.022 and the median was between 0.006 and 0.013 in the five datasets.

Cross Venn-ABERS Predictor The Cross Venn-ABERS Predictor (CVAP) is built up by combining the results from k IVAPs, where k is a definable parameter of the CVAP algorithm. We used CVAP as described by Vovk et al. (2015), converting the k probability intervals $(p_0^1, p_1^1), \ldots, (p_0^k, p_1^k)$ generated by the IVAPs into a single probability prediction p. The training procedure is as follows:

- Randomly split the training set into k folds.
- For each of the k IVAPs: use $k-1$ folds as *proper training set* and the remaining fold as *calibration set*. Shift the fold used for *calibration set* for each IVAP in such way that each observation will be part of the *calibration set* once and in the *proper training set* the $k-1$ other times.

1. Algorithm 3 is computationally intensive because it requires computing two isotonic regressions on the calibration set plus test completion (test object plus hypothetical labels) for every test object. Indeed all those isotonic regressions operate on similar data sets; this fact can be exploited to drastically reduce the computational cost of computing Inductive Venn-ABERS on the same calibration set for many test objects (Vovk et al., 2015).

Algorithm 3: Predict a new object using an `IVAP`

Input: (`IVAP`, x), A trained `IVAP` using Algorithm 2 and a new object x
Result: (p_0, p_1), lower and higher probability for x to be of class 1

1 Load scoring algorithm and the tuples $(y_i,\ s_i)$ for each object i in $i = 1, \ldots, l_c$ of the calibration set
2 Predict the score s_{new} for object x using the scoring algorithm.
3 Fit isotonic regression to the series $(s_1, y_1), \ldots, (s_{l_c}, y_{l_c}), (s_{new}, 0)$, generating a function $f_0(s)$.
4 Fit isotonic regression to the series $(s_1, y_1), \ldots, (s_{l_c}, y_{l_c}), (s_{new}, 1)$, generating a function $f_1(s)$.
5 $(p_0, p_1) := (f_0(s_{new}), f_1(s_{new}))$.

Producing single probabilistic predictions follows these steps:

- For a new object x, predict the p_0 and p_1 values using each IVAP.
- Let $\texttt{GM}(p_1)$ stand for the geometric mean for the sequence of k p_1 values given from the IVAPs and $\texttt{GM}(1 - p_0)$ stand for the geometric mean for the sequence of k $(1 - p_0)$ values. The probabilistic prediction is then $p = \texttt{GM}(p_1)/(\texttt{GM}(1 - p_0) + \texttt{GM}(p_1))$.

We also underline that the result from a CVAP does not have to be a precise probability prediction, but instead combining k IVAPs could be performed in order to calculate more accurate probability intervals, e.g. by calculating the mean or median value of the p_0 and p_1 predictions. The CVAP would then not lose the desired validity guarantee of Venn Predictors, but would on the other hand still produce imprecise probabilities. The interval width of the probability predictions from a Venn-ABERS predictor give a measure of the uncertainty in the prediction, it is thus a good idea to take it into account even if precise probability prediction are calculated and used.

3. Results

In the evaluation of the CVAPs an *outer* k-fold cross validation was performed on top of the *inner* k folds of the CVAPs. Each record was thus part of the test set once and part of the training set the remaining $k - 1$ times, the results presented hereafter are the aggregation of all k test sets. $k = 10$ was used in both the inner loop, i.e. training 10 IVAPs per CVAP, and outer loop, i.e. training 10 CVAPs per dataset. Consequently the total number of trained IVAPs was 100 for each dataset. The folds were picked randomly at both levels, not considering the class label of the records.

Parameter height used for computing signatures descriptors in the preprocessing step was set to 1 to 3, resulting in 112710 features. SVM parameters C was set to 50 and γ was set to 0.002, all according to previously found optimal default values (Alvarsson et al., 2014).

The performance of the CVAPs was evaluated by producing calibration plots, Figure 3, which plots the *observed probabilities*, frequency of true labels being of class 1, against the *expected probabilities*, the predicted probability from the CVAPs. For a perfectly calibrated

predictor one would expect the Pearson's correlation coefficient to be 1, and the slope to be 1. Figure 3 shows the Pearson's ρ to be between 0.967 to 0.997 and that most of the plots produce one-to-one correlation between expected and observed probability. Some of the datasets show more jagged curves, which seems to be linked to the number of test examples of a given expected probability (the red line). The plot for dataset Oxidation of tertiary amine, with only 3% of the examples being of class 1 (Table 1), stops around 0.8, meaning that it did not predict any test example to be of class 1 with a higher probability than 0.8.

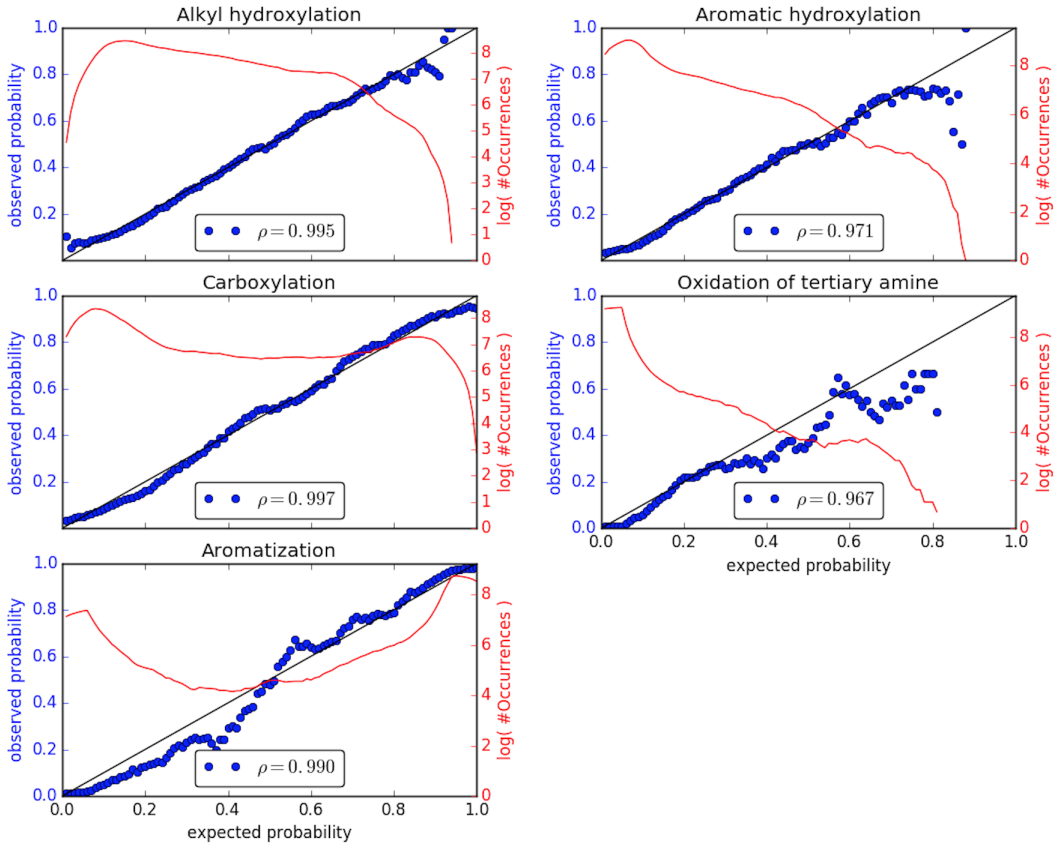

Figure 3: Calibration plots for all datasets, showing how well-calibrated the predictions are compared to expected outcome. The plots have been generated by picking 100 points linearly distributed between 0 and 1, forming the *expected probability* on the x-axis. Observations predicted with a probability p of ± 0.05 from each expected probability point are used when calculating the *observed probability* (blue dots) and occurrence (red line). The observed probability is the fraction of observations that belong to class 1. ρ indicates the Pearson's correlation coefficient.

Another important factor in the evaluation of the models is the width of the predicted probability intervals (of the IVAPs), right column in Figure 4. Width of the probability interval give a measure of the uncertainty of the predictions, were a small interval width means that the prediction is well fitted and there is a smaller uncertainty in the prediction.

The mean interval width was between 0.014 and 0.022 and the median was between 0.006 and 0.013 in the five datasets, the histogram also shows that almost all intervals are less than 0.1.

Figure 4, left column, shows the distribution of the predicted probabilities. The desired result is that predictions are either close to 0 or close to 1, giving informative results that indicate either of the two classes. Only the last plot has two clear peaks close to 0 and 1 and producing 'optimal results'. The forth plot has a clear peak close to 0, mostly indicative of the skewed class distribution, actually only predicting a probability over 0.5 for 77 molecules, which should be compared to the 325 molecules known to belong to class 1. Clearly this dataset did not perform well and a different sampling strategy should be tested to try to improve the results. The Carboxylation dataset also have a histogram that approaches having two peaks, but has a lot of molecules predicted somewhere in the middle. The two first plots also looks affected by the skewed class distribution, even though not as extreme as in the forth plot, producing undesired distribution of the predicted probabilities.

Further evaluation was performed by calculating the *log loss* and *area under the receiver operating characteristic curve* (AUC), Table 2. Log loss is calculated using:

$$log\ loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log p_i + (1-y_i)\log(1-p_i)] \tag{2}$$

To help in the interpretation of the log loss value, we can observe that a perfect probabilistic predictor would of course have log loss equal to 0, where a predictor that always predicted probability 0.5 would result in log loss of 0.693. Log loss penalizes harshly extreme predictions that turn out to be wrong, as $\log(p) \to -\infty$ as $p \to 0^+$. Difficult datasets that are hard to model are penalized, which can be seen when combining the histogram plots in Figure 4 and the log loss values in Table 2. For the three largest dataset the histogram shows many samples that are predicted in the "gray zone" and not producing predictions close to either 0 or 1. The log loss value for these datasets was also significantly higher than the two other datasets. The worst dataset, considering log loss, was the biggest dataset, which was next to best considering the Pearson's ρ and visual interpretation of the calibration plots in Figure 3. These measurements are thus complementary in the evaluation of the predictions.

AUC numbers ranging from 0.753 to 0.964 indicate that the predictions are good, although we emphasize that well-calibrated predictions is a more important feature and AUC measures are mostly included here due to being the de facto standard evaluation measure for classification predictors.

4. Conclusion

In this paper we have produced results that show that the CVAP framework works well for SOM predictions. The results are promising, showing well-calibrated results for most of the datasets, only producing poor results for the extremely skewed dataset were only 3% of data belonged to class 1. We hypothesize that the poorer result mostly depend upon study design and the dataset to be more difficult because of the imbalance in class distribution. Incorporation of stratified fold-splits, over sampling of the minority class or

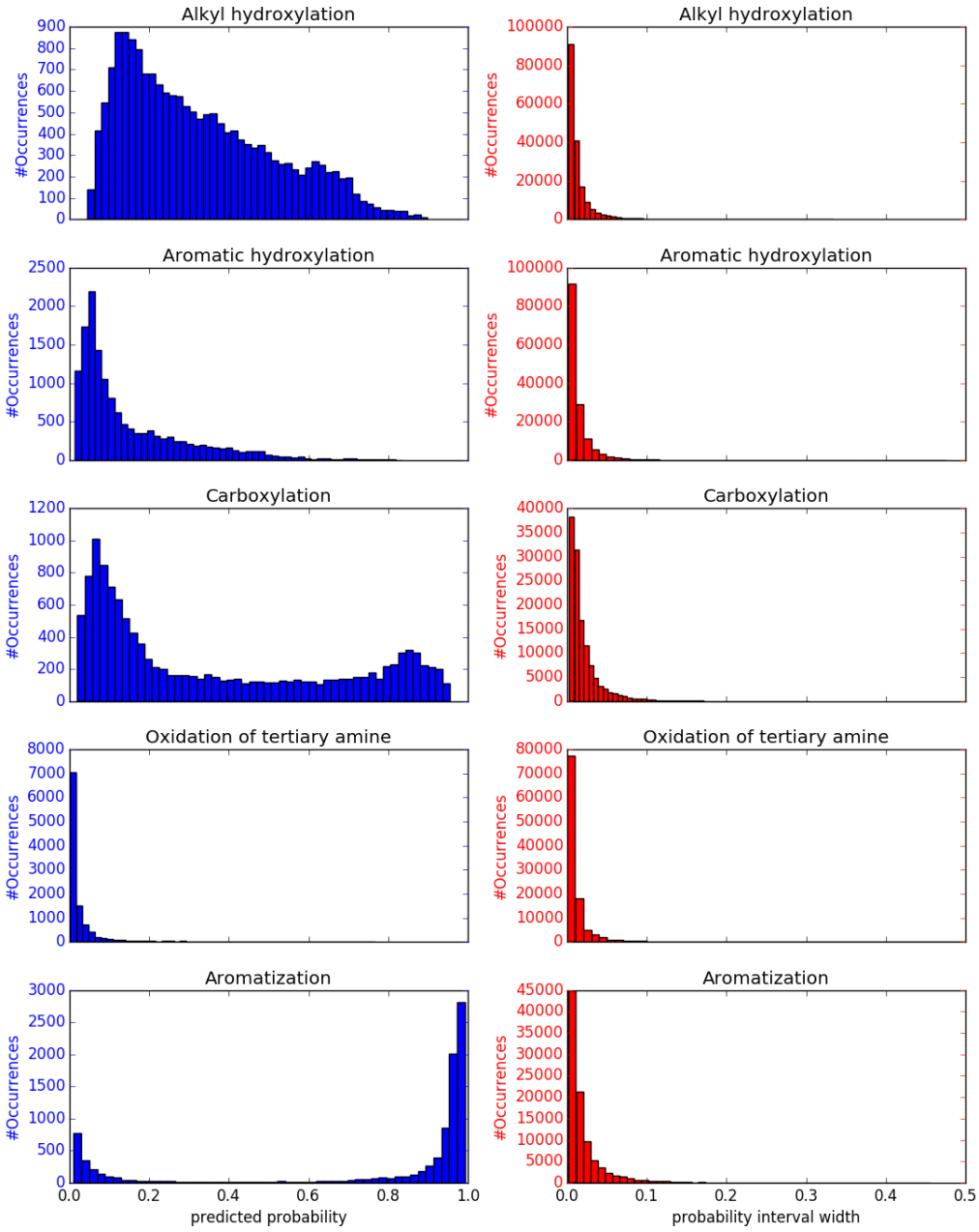

Figure 4: Histogram plots showing the distribution of the predicted values (left). The desired result is clear predictions, close to either 0 or 1, thus indicating high probability for one of either of the classes. The last dataset produced very clear predictions were almost all probabilities was predicted close to 0 or 1. Also the third dataset produced a higher frequency of probabilities close to 0 or 1, even though there is a background of predictions all over the spectra. Histogram plots over probability interval width (right) showing that most intervals are small, indicating that there is high certainty in most of the predictions.

Table 2: The log loss computed for each of the datasets using the `log_loss` function in Scikit-learn version 0.18.1 (Pedregosa et al., 2011) and area under the receiver operating characteristic curve (AUC).

Dataset name	**Log Loss**	**AUC**
Alkyl hydroxylation	0.538	0.753
Aromatic hydroxylation	0.348	0.793
Carboxylation	0.410	0.881
Oxidation of tertiary amine	0.093	0.904
Aromatization	0.173	0.964

boosting might help improving the predictive performance in this case. The interval widths and AUC measures also indicate that the models perform well on the given dataset, whereas the distribution of the predictions in left column of Figure 4 tells us that not all models are as informative as desired, a result likely more depending on the datasets than the CVAP framework itself. Producing probability based predictions is a desired property within drug discovery applications and machine learning in general, making CVAP a framework worth exploring further.

References

Jonathan Alvarsson, Martin Eklund, Claes Andersson, Lars Carlsson, Ola Spjuth, and Jarl ES Wikberg. Benchmarking study of parameter variation when using signature fingerprints together with support vector machines. *Journal of chemical information and modeling*, 54(11):3211–3217, 2014.

Robert Burbidge, Matthew Trotter, B Buxton, and Sl Holden. Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Computers & chemistry*, 26(1):5–14, 2001.

Lars Carlsson, Ola Spjuth, Samuel Adams, Robert C Glen, and Scott Boyer. Use of historic metabolic biotransformation data as a means of anticipating metabolic sites using metaprint2d and bioclipse. *BMC bioinformatics*, 11(1):362, 2010.

Chemical Information Systems Inc. DAYLIGHT. Smirks - a reaction transform language, 2008. URL http://www.daylight.com/dayhtml/doc/theory/theory.smirks.html.

Jean-Loup Faulon, Donald P Visco, and Ramdas S Pophale. The signature molecular descriptor. 1. using extended valence sequences in qsar and qspr studies. *Journal of chemical information and computer sciences*, 43(3):707–720, 2003.

Peter Gedeck and Richard A Lewis. Exploiting qsar models in lead optimization. *Curr. Opin. Drug Discov. Devel.*, 11(4):569–575, Jul 2008.

Corwin Hansch. Quantitative approach to biochemical structure-activity relationships. *Acc. Chem. Res.*, 2(8):232–239, 1969.

Christoph Helma. Lazy structure-activity relationships (lazar) for the prediction of rodent carcinogenicity and salmonella mutagenicity. *Mol. Divers.*, 10(2):147–58, May 2006.

S.R. Johnson, X.Q. Chen, D. Murphy, and O. Gudmundsson. A computational model for the prediction of aqueous solubility that includes crystal packing, intrinsic solubility, and ionization effects. *Mol. Pharmaceutics*, 4(4):513–523, 2007.

Elsevier MDL. Mdl metabolite database, 2005. URL http://accelrys.com/products/collaborative-science/databases/bioactivity-databases/biovia-metabolite.html.

Cristian R Munteanu, Enrique Fernández-Blanco, José A Seoane, Pilar Izquierdo-Novo, José Angel Rodríguez-Fernández, José María Prieto-González, Juan R Rabuñal, and Alejandro Pazos. Drug discovery and design for complex diseases through qsar computational methods. *Curr. Pharm. Des.*, 16(24):2640–55, 2010.

Ulf Norinder, Lars Carlsson, Scott Boyer, and Martin Eklund. Introducing conformal prediction in predictive modeling. a transparent and flexible alternative to applicability domain determination. *Journal of chemical information and modeling*, 54(6):1596–1603, 2014.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Patrik Rydberg, David E Gloriam, Jed Zaretzki, Curt Breneman, and Lars Olsen. Smartcyp: A 2d method for prediction of cytochrome p450-mediated drug metabolism. *ACS medicinal chemistry letters*, 1(3):96–100, 2010.

Simon Spycher, Pavel Smejtek, Tatiana I. Netzeva, , and Beate I. Escher. Toward a class-independent quantitative structure-activity relationship model for uncouplers of oxidative phosphorylation. *Chem. Res. Toxicol.*, 21(4):911–927, 2008.

Christoph Steinbeck, Yongquan Han, Stefan Kuhn, Oliver Horlacher, Edgar Luttmann, and Egon Willighagen. The chemistry development kit (cdk): an open-source java library for chemo- and bioinformatics. *J Chem Inf Comput Sci*, 43(2):493–500, 2003. doi: 10.1021/ci025584y.

Christoph Steinbeck, Christian Hoppe, Stefan Kuhn, Matteo Floris, Rajarshi Guha, and Egon L Willighagen. Recent developments of the chemistry development kit (cdk) - an open-source java library for chemo- and bioinformatics. *Curr Pharm Des*, 12(17):2111–20, 2006.

Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6):1947–1958, 2003.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic learning in a random world*. Springer, New York, 2005.

Vladimir Vovk, Ivan Patej, and Valentina Fedorova. Large-scale probabilistic prediction with and without validity guarantees. *Proceedings of NIPS 2015*, 2015. URL http://alrw.net/articles/13.pdf.