Faculty of Medicine
and Dentistry

Palacký University
Olomouc

Mining of biologically relevant patterns from QSAR models

Doctoral dissertation

Mariia Matveieva

Supervisor

Pavlo Polishchuk, M.Sc., Ph.D.

Olomouc 2022

Lékařská
fakulta

Univerzita Palackého
v Olomouci

Vytěžovaní biologicky relevantních vzorců z modelů QSAR

Disertační práce

Mariia Matveieva

Školitel

Pavlo Polishchuk, M.Sc., Ph.D.

Olomouc 2022

I hereby declare that this Ph.D. dissertation has been composed solely by myself, and embodies the original research work of my own investigation. I played the major role in the preparation and execution of the experiments, and the data analysis and interpretation are entirely by own work. Any collaborative contributions from colleagues have been explicitly acknowledged, and references to all supporting literatures and resources have been duly cited. I further declare that this dissertation has not been submitted, in whole or in part, for any other degree or professional qualification.

In Olomouc 2022 Maria Matveieva, Mgr.

Acknowledgements

First of all, I would like to thank my supervisor, assistant professor Pavel Polishchuk for his diligent guidance, great ideas and scientific discussions during my studies. I would like to thank associate professor Marián Hajdúch and doctor Petr Džubák for the opportunity to realize my studies at IMTM and take advantage of its facilities and funding. I would like to thank all the co-authors of projects related to this work, especially, professor Mark Cronin and doctor Oleg Tinkov. My special thanks belong to my family and friends who have always supported me.

CONTENTS

Abstract

This dissertation elaborates on problems of structural interpretation of QSAR models. Development of the validation framework of interpretation methods was performed in the first place. The framework consists of specifically designed data sets. They are purposed for systematic evaluation of the ability of interpretation approaches to retrieve patterns important for activity studied. We applied the framework to study the behavior of most used machine learning (ML) algorithms, molecular descriptors and an interpretation approach: Universal approach for interpretation of QSAR models (UIA) (1, 2).

We implemented a new Extension of UIA to improve *global* (data set level) interpretation by this approach. Results produced by UIA in the form of fragment contributions show certain variability. This variability can be caused by different chemical contexts of those fragments and is observed for the majority of biological end-points. The Extension identifies groups of compounds (clusters) comprising the same structural pattern, where the pattern has substantially different influence on the studied property, and retrieves chemical contexts within these clusters. Retrospective analysis of toxicity to *Tetrahymena pyriformis* showed that the clustering technique explains distribution of contributions of particular molecular groups/fragments and enhances explanatory power of the UIA.

To address practical aspects of model interpretation we applied UIA and the Extension developed to real case data sets. First, we studied aquatic toxicity. The results made it possible to rank contributions of molecular patterns (fragments) to toxicity against three different aquatic organisms. The study confirmed known toxicophore features and proposed new fragments stably influencing all three studied endpoints, thus proving the approach useful. The Extension was also applied to modeling of anticancer activity (toxicity of small molecules against cancer cell lines). Important patterns have been retrieved which information can be used in compound optimization.

All the methodology developed was implemented as open-source software. The benchmarking framework is available at https://github.com/ci-lab-cz/ibenchmark. The Extension to SPCI software was implemented in the open-source R package (https://github.com/DrrDom/rspci). In addition, multiple undersampling technique was added to SPCI software. This improved modelling results for unbalanced classification data sets. Interpretation based on such models also proved feasible. An open-source interpretation tool for graph neural networks using UIA was proposed and implemented within DeepChem project.

Keywords: QSAR explainability, QSAR interpretation, Gaussian Mixture Modelling, QSAR interpretability benchmark, synthetic data set.

Title: Mining of biologically relevant patterns from QSAR models

Author: Mariia Matveieva

ORCID: 0000-0001-5373-9923

Supervisor: Assistant Professor Pavlo Polishchuk, M.Sc., Ph.D.

Ph.D. programme: Pediatrics (full-time form)

Institution: Institute of Molecular and Translational Medicine, Faculty of Medicine and Dentistry, Palacký University and University Hospital in Olomouc

Year: 2022

Pages: 115

Anotace

Disertační práce se zabývá problematikou strukturální interpretace modelů QSAR. Nejprve byl proveden vývoj validačního frameworku pro interpretační metody. Framework se skládá ze speciálně navržených datových souboru. Jsou určeny pro hodnocení schopnosti interpretačních přístupů vytěžovat data důležitá pro studovanou aktivitu malých molekul. Aplikovali jsme framework ke studiu nejpoužívanějších algoritmů strojového učení (ML), molekulárních deskriptorů a interpretačního přístupu: Univerzálního přístupu pro interpretaci modelů QSAR (UIA) (1, 2).

Implementovali jsme nové rozšíření UIA, abychom zlepšili globální interpretaci pomoci tohoto přístupu. Výsledky produkované UIA ve formě příspěvků molekulárních fragmentu vykazují určitou variabilitu. Tato variabilita může být způsobena různými chemickými kontexty těchto fragmentů a je pozorována u většiny biologických aktivit. Rozšíření identifikuje skupiny sloučenin (shluky) obsahující stejný strukturní vzorec, kde vzorec má podstatně odlišný vliv na studovanou vlastnost, a vyhledá chemické souvislosti v rámci těchto shluků. Retrospektivní analýza toxicity pro *Tetrahymena pyriformis* ukázala, že technika shlukování vysvětluje distribuci příspěvků jednotlivých molekulárních skupin / fragmentů a zvyšuje vysvětlovací schopnost UIA.

K řešení praktických aspektů interpretace modelu jsme aplikovali UIA a vyvinuté rozšíření na reálné datové soubory. Nejprve, jsme studovali vodní toxicitu. Výsledky umožnily seřadit příspěvky molekulárních vzorů (fragmentů) k toxicitě vůči třem různým vodním organismům. Studie potvrdila známé toxikofory a navrhla nové fragmenty stabilně ovlivňující všechny tři studované aktivity, čímž se prokázala užitečnost tohoto přístupu. Rozšíření bylo také aplikováno na modelování protirakovinné aktivity (toxicita malých molekul proti rakovinným buněčným liniím). Byly získány nové potenciálně důležité vzorce, které lze použít při optimalizaci sloučenin. Veškerá vyvinutá metodika byla implementována jako open-source software. Validační framework je dostupný na https://github.com/ci-lab-cz/ibenchmark. Rozšíření pro SPCI bylo implementováno v open-source R balíčku (https://github.com/DrrDom/rspci). V rámci projektu DeepChem byl navržen a implementován open-source interpretační nástroj pro grafové neuronové sítě využívající UIA.

Klíčová slova: Vysvětlitelnost QSAR, interpretace QSAR, Gaussian Mixture Modelling, benchmark interpretovatelnosti QSAR, syntetický datový soubor.

Název: Vytěžovaní biologicky relevantních prvku z QSAR modelů

Autor: Mariia Matveieva

ORCID: 0000-0001-5373-9923

Školitel: odborný asistent Pavlo Polishchuk, M.Sc., Ph.D.

Doktorský studijní program: Pediatrie (prezenční forma)

Instituce: Ústav molekulární a translační medicíny, Lékařská fakulta, Univerzita Palackého a Fakultní nemocnice v Olomouci

Rok: 2022

Počet stránek: 115

1. Introduction to quantitative structure–activity relationship

Quantitative structure–activity relationship (QSAR) modelling was born more than 100 years ago. In early XX century Meyer and Overton suggested that the narcotic action of a group of organic compounds was related to their olive oil/water partition coefficients (1). This relationship demonstrated the central axiom of structure-activity modeling – the activity of substances is governed by their properties, which in turn are determined by their chemical structure. Therefore, there are relationships between structure and properties and activity.

QSAR models are regression or classification models used in the chemical and biological sciences and engineering.

QSAR regression models relate a set of "predictor" variables (X) to the potency of the response variable (Y), while classification QSAR models relate the predictor variables to a categorical value of the response variable. A model can be represented as the mapping function f *:*

$$y = f(x), \tag{1-1}$$

Where y is the value for Y predicted by the model, i.e. approximation of the response-variable; f typically is a complex machine learning model; it doesn't need to have an analytical form; X are the predictors consisting of physicochemical properties or theoretical molecular descriptors of chemicals (2).

As an example, biological activity can be expressed quantitatively as the concentration of a substance required to give a certain biological response. The mathematical expression, if carefully validated can then be used to predict the modeled response of other chemical structures (3).

At present, QSAR modeling is one of the basic tools of modern drug design and environmental sciences. Models have developed into robust and reliable systems, at the same time, they became highly complex and non-interpretable: so-called “black boxes”. Present dissertation is dedicated to “opening the black box”, that is model interpretability problem. The importance of interpretability aspect is justified in Section 2.

1.1. Molecular descriptors

Molecular graphs (Figure 1.1) are naturally used for the representation of chemical structures, if necessary, supplemented with information on the three-dimensional coordinates of atoms, as well as atomic and bond attributes (properties).

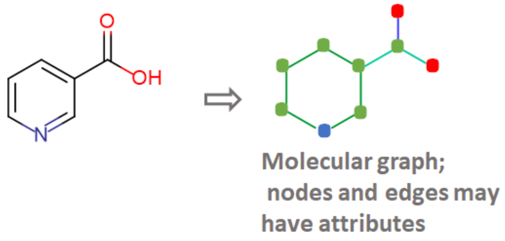


Figure 1.1 Example of chemical compound graph representation

To build successful models it is important to capture relevant aspects of structures. For instance, for membrane permeability of compounds the key role may belong to lipophilicity and size.

Until recent time, graphs themselves were not used for modelling, mainly due to the requirement to pass a *set of variables – feature vector* as input. We will describe here traditional *molecular descriptors;* graph-based methods are mentioned in 6.2.2.

Molecular descriptors represent structures as a *set of variables* used to build models, also called *feature vector.* There are different types of such features capturing various aspects of molecules.

Fragment descriptors (4),(5) exist in two main variants - binary and count-based. Binary fragment descriptors (Figure 1.2) indicate whether a given fragment (substructure) is contained in a structural formula (that is, whether a given subgraph is contained in a molecular graph describing a given chemical compound), while count-based fragment descriptors indicate how many times a given fragment (substructure) is contained in a structural formula (that is, how many times a given subgraph is contained in a molecular graph describing a given chemical compound).

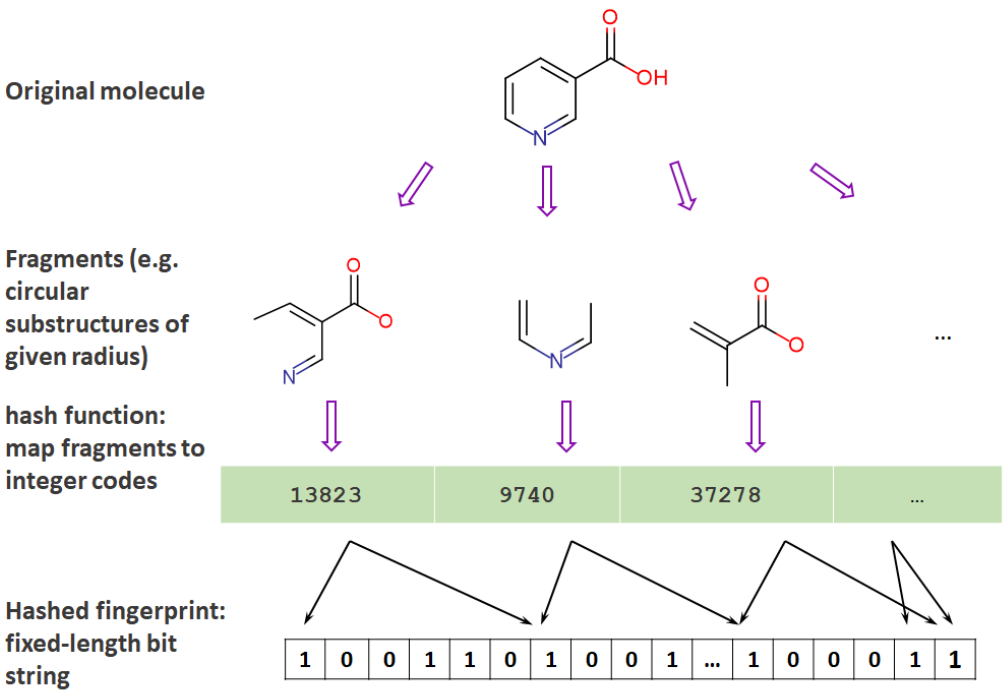

Original molecule
O
OH
N
Fragments (e.g. circular substructures of given radius)
O
O
N
N
O
O
...
hash function: map fragments to integer codes
13823
9740
37278
...
Hashed fingerprint: fixed-length bit string
1 0 0 1 1 0 1 0 0 1 ... 1 0 0 0 1 1

Figure 1.2 Visualization of the basic algorithm to generate hashed binary Morgan fingerprint. Layer with radius of 2 bonds is shown

Topological indexes are invariants of a molecular graph, expressed as numerical values that characterize the structure of a molecule as a whole. Usually, topological indices do not reflect the multiplicity of chemical bonds and types of atoms (C, N, O, etc.), hydrogen atoms are not taken into account. The most famous topological indices include the Hosoi index, the Wiener index, the Randic index, the Balaban index, and others (5).

Physicochemical descriptors are numerical characteristics obtained as a result of modeling the physicochemical properties of chemical compounds, or quantities that have a clear physicochemical interpretation. The most commonly used descriptors are: lipophilicity (LogP), molar refraction (MR), molecular weight (MW), hydrogen bond descriptors, molecular volume and surface area.

Quantum-chemical descriptors (6) are numerical values obtained as a result of quantum-chemical calculations. The most frequently used descriptors are: energies of boundary molecular orbitals (HOMO and LUMO), partial charges on atoms and partial bond orders, energies of cationic, anionic and radical localization, dipole and higher multipole moments of the electrostatic potential distribution.

Pharmacophore descriptors show whether the simplest pharmacophores, consisting of pairs or triplets of pharmacophore centers with a specified distance between them, can be contained within the molecule analyzed (7).

Molecular descriptors are most fully described in the monograph (8), which can be considered an encyclopedia of molecular descriptors.

1.2. QSAR modelling methods

QSAR studies largely rely on machine learning (ML) algorithms. Machine learning can be most generally and informally defined as *pattern recognition*. It relies on *large datasets (big data)* to retrieve common patterns, trends, regularities etc. In bioactivity modelling it can, for example, use small molecule screening database to learn structural features of molecules related to their activity. This information can be used to optimize potential drug candidates, understand mechanisms of action, or screen out inactive or toxic compounds. Among the most popular and universal ML methods used in QSAR are Random Forest (RF), Support Vector Machines (SVM), Gradient Boosting (GBM) and Neural Networks (NN).

Decision Trees and Random Forest

Decision trees (DT) is an example of intuitive, natural method of machine intelligence, serving as a base for more complex methods. DTs are the building blocks of the RF model. Let’s consider an example of a dataset consisting of the molecules at the top of the Figure 1.3. There are two classes – *actives* (red) and *inactives* (green) and the task is to separate the classes using their features: *molecular weight (MW)* and *lipophilicity (logP)*.

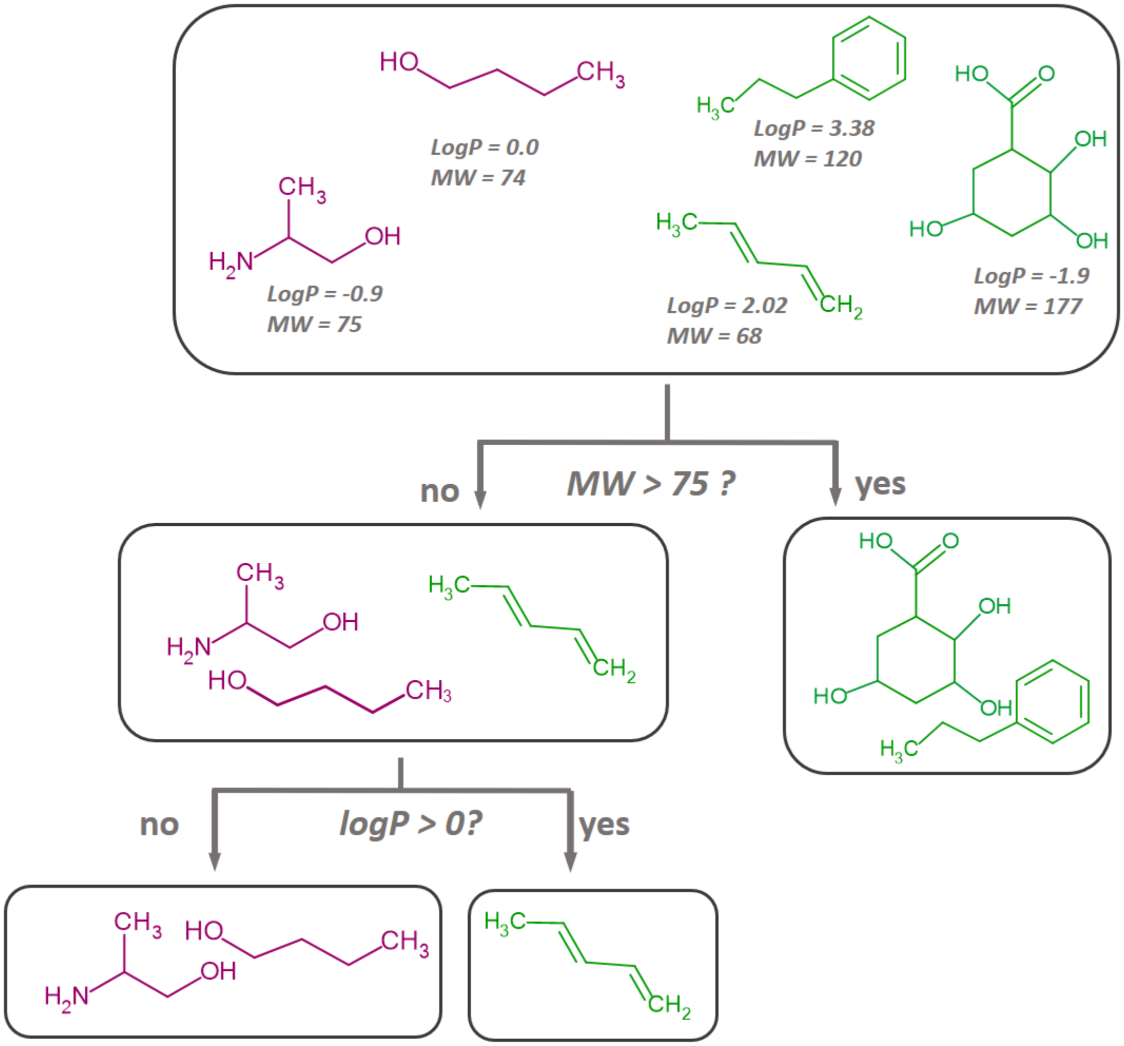


Figure 1.3 Decision tree example for chemistry-related classification task. Explained in text

We can use the question, “Is molecular weight greater than 75?” to split the first *node*. The *Yes* branch (the greens) is all inactives, but the *No* branch can still be split further. Now we can use the second feature and ask, “Is *logP* greater than 0?” to make a second split. The tree is built, since no further split is needed. In real applications like bioactivity modelling the data is much higher-dimensional (dimensionality corresponds to the number of features) and the decision trees are complex and hard-to-visualize. At each step the feature and the value of that feature is chosen to perform the next split. Most widely accepted criteria for this choice are *Gini impurity* (9) and *information entropy (10)*. The idea behind both is to achieve efficient separation of classes.

While being simple and intuitive, DTs suffer from high variance of resulting models: DTs are unstable. In our example, if we used different initial data, we could get a totally different branching pattern, that is, a different model. In bioactivity modelling, two different models can predict the same compound as belonging to two different classes: active and inactive. The result depends heavily on training dataset used to create the model, making the system unreliable.

Random forest is an ensemble method based on DTs proposed by Leo Breiman in 1991 (11). The method produces much more robust models than DTs owing to two powerful ideas: *bootstrap aggregating („bagging“)* and random feature subspaces. Large number of trees is built at the first step (Figure 1.4), each of which uses *independent identically drawn* data sample – i.e. bootstrap sample. Predictor variables are randomly subsampled at each split. Typically, *square root* of overall number of features defines the sample size. Trees are grown until perfect (or nearly perfect) classification accuracy or regression approximation is achieved, which requires them to be deep. After all trees are built the final model output is produced as an average of predictions of individual trees (in regression case) or using majority vote (in classification case). The results are stable. Speaking simplistically, this is achieved by mutual cancellation of errors of all individual trees.

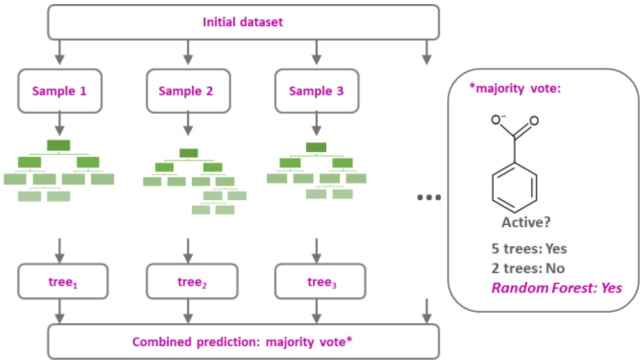


Figure 1.4 Random Forest example for chemistry-related classification task. Explained in text

Gradient boosting

Gradient boosting machine (GBM) is a powerful tree-based learning method, but unlike RF it is based on sequential (as opposed to parallel) tree building and the trees are shallow (as opposed to deep) (12, 13). The technique called *boosting* employs gradual improvement of model robustness by starting with a simple model. Initial model is stable (has low variance), but produces *biased* result: the predictions are shifted towards some value (e.g. mean). At the next step another model is built to correct errors of the first model, and the two are added up. The process continues until the result achieves desired accuracy. An example of a regression model is given in Figure 1.5.   Each next model realizes a small step in the direction of (negative) *gradient* (hence *gradient* boosting) *of loss function in model space*. Classification setting is analogous; the main difference lies in a loss function choice.

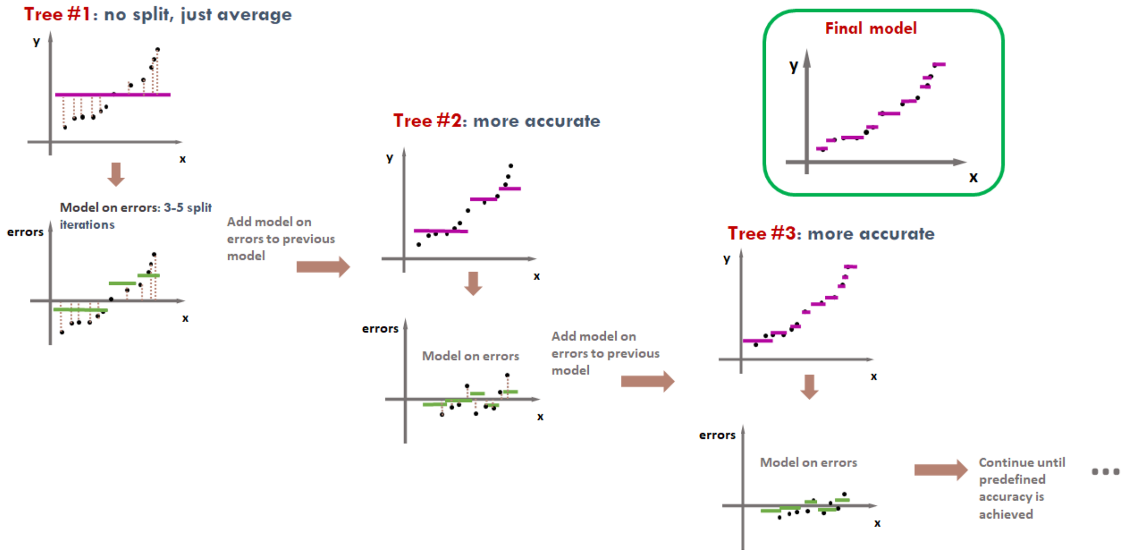

Tree #1: no split, just average
y
x
errors
Model on errors: 3-5 split iterations
x
Add model on errors to previous model
Tree #2: more accurate
y
x
errors
Model on errors
x
Add model on errors to previous model
Final model
y
x
Tree #3: more accurate
y
x
errors
Model on errors
x
Continue until predefined accuracy is achieved
•••

Figure 1.5 Gradient boosting model example for regression task

Support vector machines

SVM was developed by Vapnik and colleagues   (14-16) in 1990s, though the idea was published as early as 1964 by Vapnik in his doctoral dissertation. As already discussed, some methods of ML suffer from unstable results. According to *Vapnik-Chervonenkis theory*, stable result (lower variance) can be achieved if data points are separated with *maximum margin*. Let's continue with molecules from Figure 1.3, representing them this time in 2D coordinates of their features, Figure 1.6a. The two classes can be separated by a grey solid line, providing a model with perfect accuracy: greens on the left- and reds on the right-hand side. However, there are possibly many lines, providing such solution (e.g. dashed line). The line with *maximum margin* is the one, that provides the longest distance (brown arrow) between molecules of different classes; marginal molecules are called *support* vectors (i.e. *butanol* and *pentadiene*). Such a task can be uniquely solved in the course of *constrained minimization*, which minimizes the norm of the vector of coefficients *W* in equation of the line (1-2, while support vectors are not allowed to cross the "borders". Such a method can be extended to non-linear problems, i.e. when the data cannot be separated by a line (or hyperplane in higher-dimensional feature spaces). *Kernel trick* Figure 1.6b allows to separate the data via transforming it to higher-dimensional space, using a kernel function. The trick is that the transformation doesn't need to be done explicitly, thanks to special property of kernel functions. The computational cost is greatly reduced relatively to explicit mapping. After kernel is applied, the inverse transformation is done, again, implicitly.

$$X^T W + b = 0, \qquad (1\text{-}2)$$

where X is a vector of features, representing the data, and W and b are coefficients and bias.

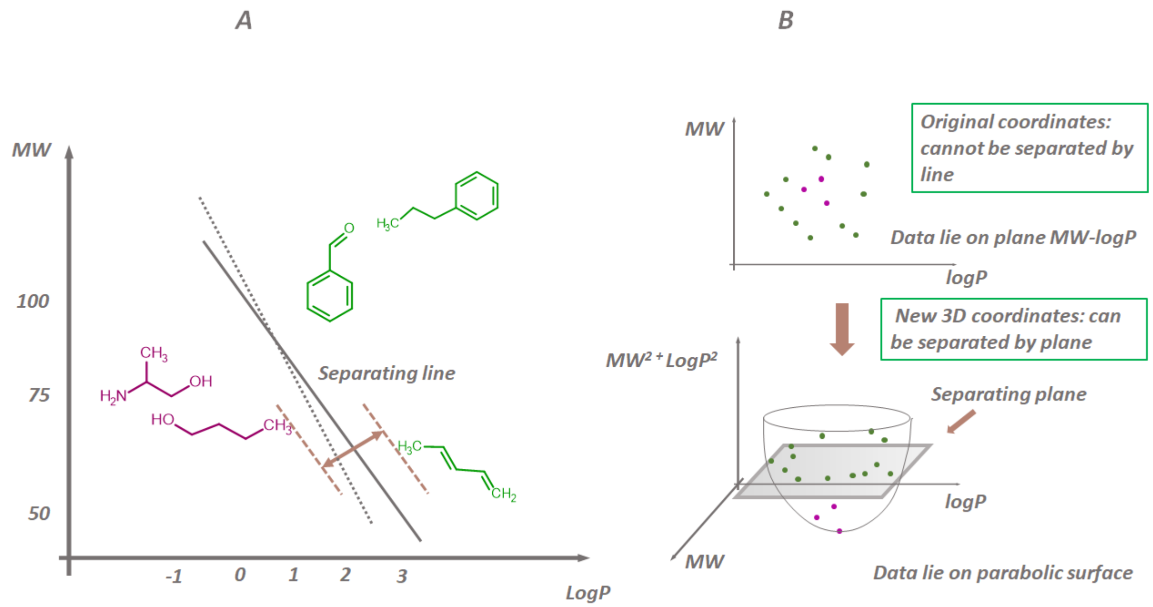

A
B
MW
100
75
50
-1
0
1
2
3
LogP
Separating line
CH3
H2N
OH
HO
CH3
O
H3C
H3C
CH2
Original coordinates: cannot be separated by line
MW
Data lie on plane MW-logP
logP
New 3D coordinates: can be separated by plane
MW² + LogP²
Separating plane
logP
MW
Data lie on parabolic surface

Figure 1.6 A: SVM example for chemistry-related classification task. B: kernel trick illustration. Explained in text.

Neural networks and deep learning

Artificial neural networks (NN) represent the most popular ML method nowadays, owing to their algorithmic (architectural) flexibility and universality. NNs find applications in nearly every field of science and technology: from language processing (machine translation) and self-driving cars to drug discovery and development.

NNs borrow the idea from human brain functioning, hence the name. The idea of mimicking the brain to perform computations originated in 1943 (17), however modern *deep* NN became possible only thanks to works of LeCun, Bengio, Hinton et al. (18), (19, 20). Using a regression example, let’s illustrate how NN would solve the task, Figure 1.7.The basic unit of NN is a neuron: a unit that takes input values and passes them to the next neuron, after a (non-linear) transformation. Neurons are combined into a net, connecting them by weights. The data (e.g. molecular features X and activities Y) are fed to the net and the final output produced is then compared to original activities. The weights are adjusted by gradient descent method until the result is close enough to original data. This is done via backpropagation procedure (21). NN are *universal function* *approximators*, but they are prone to high variance. Different techniques are applied to reduce the variance, e.g. dropout(20), batch normalization(22), layer normalization etc. Deep learning refers to NNs with large number of layers, including recurrent NNs, transformers, convolutional NNs, deep belief NNs, graph convolutional NNs, etc.

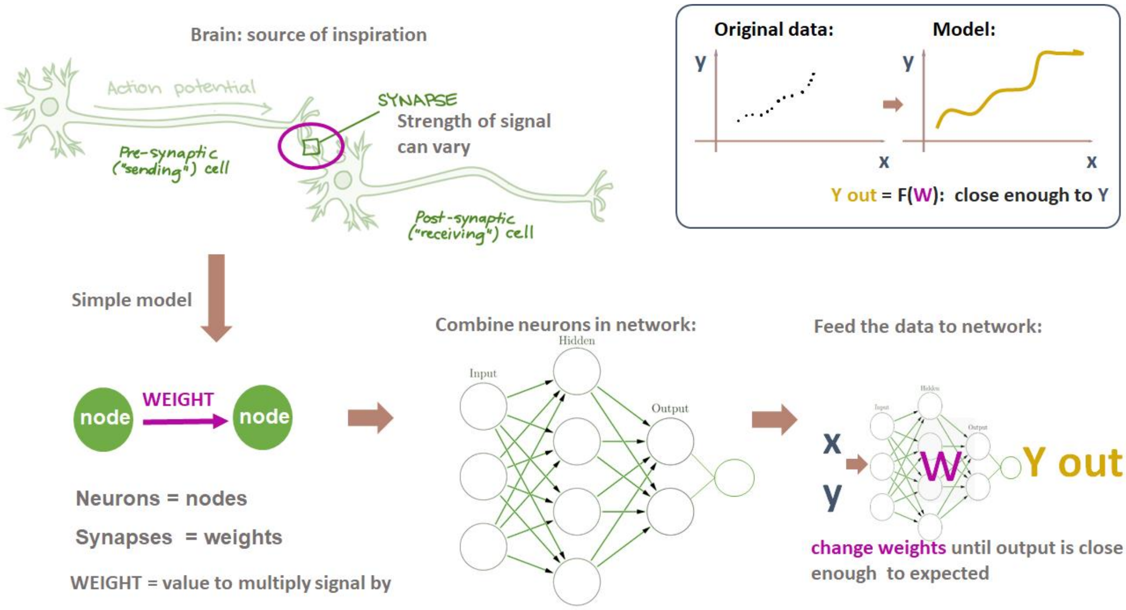

Brain: source of inspiration
Action potential
SYNAPSE
Strength of signal can vary
Pre-synaptic ("sending") cell
Post-synaptic ("receiving") cell
Original data:
Model:
y
x
y
x
Y out = F(W): close enough to Y
Simple model
node
WEIGHT
node
Neurons = nodes
Synapses = weights
WEIGHT = value to multiply signal by
Combine neurons in network:
Hidden
Input
Output
Feed the data to network:
x
y
W
Y out
change weights until output is close enough to expected

Figure 1.7 Artificial neural networks: main ideas

Graph neural networks in cheminformatics

Molecules can be naturally represented as graphs (Figure 1.8). This allows to apply graph-based NNs to directly model structure-activity relationships without calculating descriptors. Nodes of a graph, which are connected, exchange messages via *message function*. Messages are aggregated using *aggregation function*. These message passing-aggregation steps can be repeated arbitrary number of times (e.g. 3 on Figure 1.8), and then values at each node are globally aggregated using a readout function, for instance, sum. These message-passings can be viewed as graph convolution operation.

Pioneer examples of graph-based NN used in chemistry are (23, 24); (25) is an example of self-attention graph NN (explained in detail in 0).

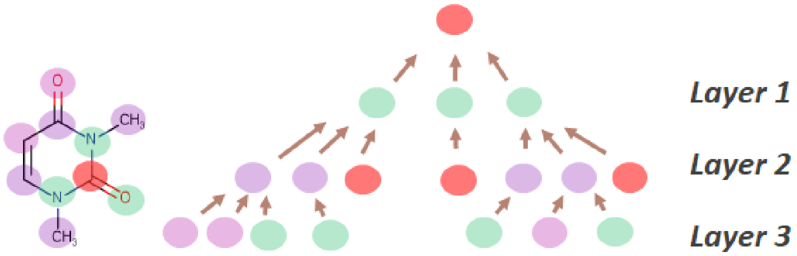


Figure 1.8 The idea of graph convolution for molecules. Red carbon atom is chosen as an example of a center on which convolution starts. All atoms are used iteratively.

2. Introduction to interpretability of QSAR models

Mechanistic interpretation of QSAR models allows to understand complex nature of biological or physicochemical processes under study. Interpretation can produce results in the form of molecular patterns important for compounds' activity (Figure 2.1). The methods providing such results allow for the calculation of the contributions of individual atoms (26), arbitrary fragments (27), or predicted activity changes corresponding to given molecular transformations (28). This information can be applied to drug and product development to optimize the structures of studied compounds by increasing efficacy and reducing harmful effects. Interpretation can also serve as a means to confirm the validity of the model, i.e. that the model has captured relevant and meaningful relationships between activity and structure (29). It is also important for regulatory application, for example, the fifth of the OECD Principles for the Validation of QSARs requires, where possible, mechanistic interpretation on QSAR models (30). Whilst this principle is optional it is considered helpful to get round the long held belief that QSAR models were “black boxes” and interpretation was not always possible.

2.1. Classification of methods

Interpretation methods can be divided into two major groups: global and local (also called instance-based). Global methods provide insights into general trends captured by the model, while local methods explain decisions on individual instances. Interpretation methods can be categorized into model-specific and model-agnostic ones. In QSAR modelling context interpretation can be either structural or feature-based. The former explains models decisions by picking important substructures (e.g. atoms), as opposed to important features (descriptors) in the latter. Also, we can distinguish groups of approaches by their methodology: perturbation-based, gradient-based, surrogate and others (discussed below). Table 2.1 refers existing methods to categories/groups. Note, that some methods fall simultaneously into several groups. For instance, if we apply gradient-based methods to graph-based architectures, we can obtain atom-level attribution, that is, structural; at the same time, applying it to descriptor based models will result in feature-based attribution. The classification we suggest is not uniquely correct, other possible versions are proposed in (31), (32), (29).

Table 2.1 Methods of QSAR model interpretation

METHODS		Model-agnostic	Model-specific
Local (instance-based)	Feature-based	• LIME (33) • Shapley sampling values (34) • Partial derivatives with *numeric differentiation(35)*	• *Gradient-based:* sensitivity analysis (36), ,Gradient*Input, CAM (37), Grad-CAM (38), integrated gradients (39), smooth-grad (40) • Layer-wise relevance propagation (41) • Attention-based NN (42)
	structural	• *Perturbation-based:* SPCI (43), Similarity maps (26), computational matched molecular pairs/series (44) • LIME (33)*	• Gradient-based: Gradient*Input*, CAM*, Grad-CAM*, integrated gradients *, smooth-grad (40)* • Layer-wise relevance propagation* • Attention-based NN* • Subgraph methods:

		**When features are substructures*	GNNExplainer (45) **When applied to graph models* (39, 46, 47)
Global	Feature-based	• *Perturbation-based:* feature importance by permutation	• Rule extraction methods • Gradient-based: Regression coefficients (in linear models)
	structural	• *Perturbation-based:* SPCI (aggregating individual results for fragment-based mode only; for atom-based mode aggregating makes no sense)	*Not reported*

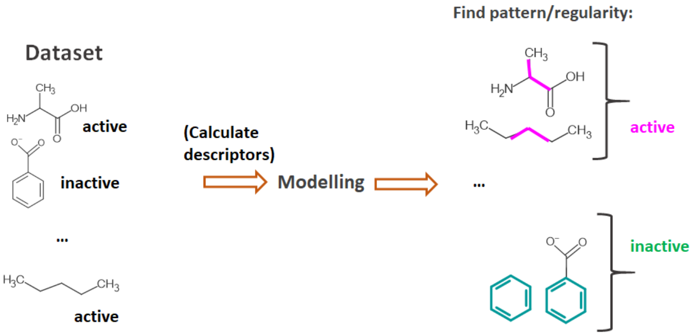


Figure 2.1 Structural interpretation of QSAR models: main idea

2.2. Model-specific interpretation methods

These approaches are applicable only to limited number of models. For instance, gradient-based methods find derivatives of model’s response with respect to input features, thus allowing to evaluate the influence of those features on the output.

Gradient-based methods

These methods judge about the importance of input features by the magnitude of their derivatives (gradients) with respect to the output. In the case when automatic differentiation is used, it requires knowing the model’s structure and a differentiable model (typically NN). However, utilizing numeric differentiation can turn methods into model-agnostic (see 2.3).

The simplest example of such an approach is interpreting regression coefficients in linear models as partial derivatives with respect to input features. Popular gradient-based methods include sensitivity analysis, Integrated Gradients (39) et al. They typically provide feature-based interpretation, i.e. the results indicate important descriptors, not substructures. However, when applied to graph-based NN they can find gradient with respect to input nodes – that is atoms – producing direct structural interpretation (examples are given below)(48).

Sensitivity analysis. In QSAR domain Aoyama and Ichikawa in their early work (49) proposed calculating partial derivatives for NN models that can be done in the analytical form. At present, for deep NN, gradients are calculated using automatic differentiation, e.g. (36). This methodology is referred to as sensitivity analysis. However, there is an ambiguity, since sensitivity analysis can also be defined as *the study of “how the uncertainty in the output of model can be divided and allocated to different sources of uncertainty in its inputs”* (50). Therefore, *perturbation-based* methods discussed below also fall under this category, as does *partial derivatives* method (35) employing model-agnostic numeric differentiation (as opposed to automatic differentiation).

.

Some methods were developed initially for convolutional NN (CNN).We will consider two examples: *CAM* and *Grad-CAM* (38). Unlike sensitivity analysis/partial derivatives, they focus not on the input layer of NN, but last convolutional layer. It is well-known that such last-layer features tend to be more semantically meaningful as opposed to input space (e.g., faces instead of edges in the case of image-related tasks). CAM multiplies weights of *final fully connected layer* by outputs of last convolutional layer. It requires global average pooling to be placed between the two layers. Grad-CAM improves on CAM by using gradient of *final fully connected layer* w.r.t. last convolutional layer. The difference is that it allows to use any number of differentiable layers after convolution instead of a single global average pooling. In principle, Grad-CAM can be applied to any intermediate convolutional layer, or used in *layer-average version*. Final equations for Grad-CAM are *(2-1),(2-2)*. For CAM $w_k^{(c)}$ is given by the weight of last fully connected layer. CAM and Grad-CAM can be adapted to graph-based NN (48)*(2-3)(2-4)*. Example architectures of CNN and graph CNN are shown on Figure 2.2, where top variant can be used with CAM and bottom variant with Grad-CAM.

$$L^{(c)}_{Grad-CAM}(i, j) = ReLU\left(\sum_k w_k^{(c)} \cdot F_k(i, j)\right) \quad (2\text{-}1)$$

$$w_k^{(c)} = 1/Z \cdot \sum_i \sum_j \cdot \partial Y^{(c)} / \partial\, (F_k(i,j)), \quad (2\text{-}2)$$

where $F_k(i, j)$ is the activation of k-th feature map in the target layer of the model for input location (i,j), and $Y^{(c)}$ is the model output score for class c (before softmax).

$$L^{(c)}_{Grad-CAM}(l, n) = ReLU\left(\sum_k w_k^{(c)} \cdot F^l_{n,k}(X, A)\right) \quad (2\text{-}3)$$

$$w_k^{(c)} = 1/N \cdot \sum_n \cdot \partial Y^{(c)} / \partial\, (F^l_{n,k}), \quad (2\text{-}4)$$

where $F^l_{n,k}(X,A)$ is the activation of feature k in the target layer l of the model at a node n, and $Y^{(c)}$ is the model output score for class c (before softmax). X, A – feature matrix and adjacency matrix of a given input molecule.

CAM and Grad-CAM have been applied to chemical graph-based tasks on a number of classification problems and a single regression problem: Crippen logP (39, 46, 47). CAM performed better across various graph-based model architectures. It remained, however, unexplained by the authors of (47) why Grad-CAM performed poorly (both in "last-layer version" and "layer-averaged version"). That clarification is needed, as CAM and Grad-CAM are equivalent in graph CNN context when applied to last layer provided that same architecture is used (48).

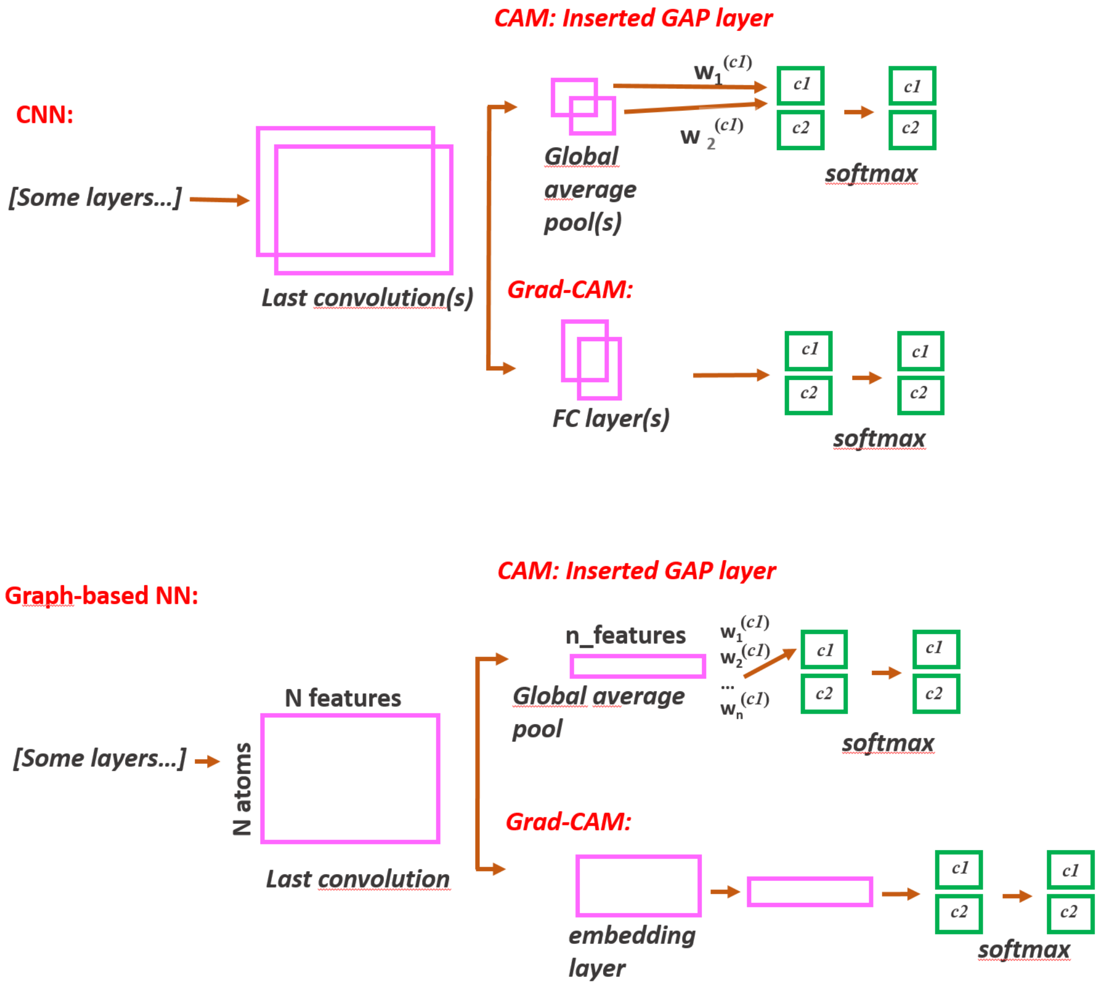


Figure 2.2. Example architectures compatible with CAM and GRAD-CAM in the context of CNN (top) and graph-based NN (bottom). Explained in text.

Integrated Gradients (IG) is another method based on derivatives. It can be also viewed as a model-agnostic method, since it can be applied to any model differentiable w.r.t. its input at any point of interest. The method requires the choice of a *baseline point*; the difference between model output at a current input and that *baseline point* is distributed among input variables. The higher portion of this difference is conferred to a variable, the more important it is. The method utilizes path-integration: straight path connecting two points is parametrized by value α between 0 and 1. Along this path each input point (vector of coordinates in input space, *x*) is defined by *g(α) = α·x.* To obtain the attribution for i-th variable, we then integrate the gradient of model output along the path with respect to $g_i(\alpha)$ as follows:

$$IntegratedGrads_i\ (x) = \int_0^1 \frac{\partial Fg(\alpha))}{\partial g_i\ (\alpha)} \cdot \frac{\partial g_i\ (\alpha)}{\partial \alpha} d\alpha \tag{2-5}$$

Since g(α) = α·x and given the baseline is *all-zeros:*

$$IntegratedGrads_i\ (x) = x_i \ \cdot \int_0^1 \frac{\partial F(\alpha \cdot x))}{\partial(\alpha \cdot xi)} \cdot d\alpha \tag{2-6}$$

Attributions can be interpreted as “projections” of total “*delta F”* onto respective planes; however, this holds only if each variable’s derivative is independent from all other variables (Figure 2.3), e.g. linear, paraboloid etc. Important properties of the method is *completeness (“summation to delta”)*, which means:

$$\sum_i IntegratedGrads_i\ (x) = \Delta F \tag{2-7}$$

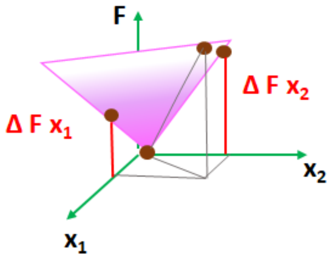


Figure 2.3. Geometric interpretation of IG attribution for a simple case: here linear.

Riemann approximation can be applied to compute the integral in practice (39):

$$IntegratedGrads_i^{approx.}(x) :: = (x_i - x_i') \cdot \textstyle\sum_{k=1}^{m} (\partial F(x' + (x - x') \cdot k/m)/\partial x_i) \cdot 1/m\,, \qquad (2\text{-}8)$$

where x' is a baseline-point, F –function modelled.

IG have been applied to chemistry-related tasks (39, 46, 47). In all works, authors used a molecule with all features equal to zero as the baseline point. In (46) and (47) to ensure the baseline would be predicted with 50% probability active (in classification) they added 20% of such molecules to training data labeling half them as "active" and half as "inactive". The authors of (39, 46, 47) performed comparison of attribution methods across several classification and regression tasks and various graph-based model architectures. They used artificial (synthetic) datasets of molecular graphs with pre-defined vertex labels, therefore, comparing these labels with experimental results they could evaluate *attribution accuracy*, which appeared the highest for CAM and IG.

It is important to choose "the right" baseline-point, since local behavior of *F* fully determines the "true" contributions of variables, and the baseline should represent this local behavior. As shown in Figure 2.4, wrong choice of the baseline can lead to inadequate integration result, *e.g. zero, in the case when the function has extremum in given neighborhood*, while correct choice allows to attribute the difference to the variables more reasonably.

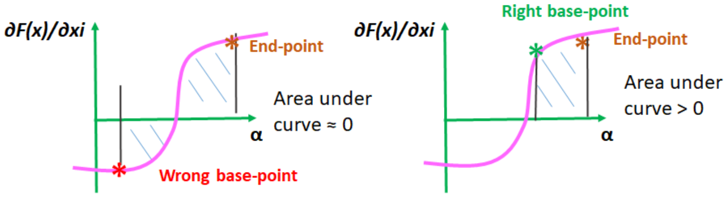


Figure 2.4. Geometric illustration of "wrong" and "right" baseline point for IG attribution.

Layer-wise relevance propagation (LRP) (41) is a method based on decomposition of model's output into relevances *(R)* of input features via redistribution of signal in NN from the final layer though all layers down to the first (Figure 2.5). Importantly, the "relevance conservation" rule must hold, i.e. for all layers:

$$f(x) = \ldots = \sum R^{(l+1)} = \sum R^{(l)} = \ldots = \sum R^{(1)} \qquad (2\text{-}9)$$

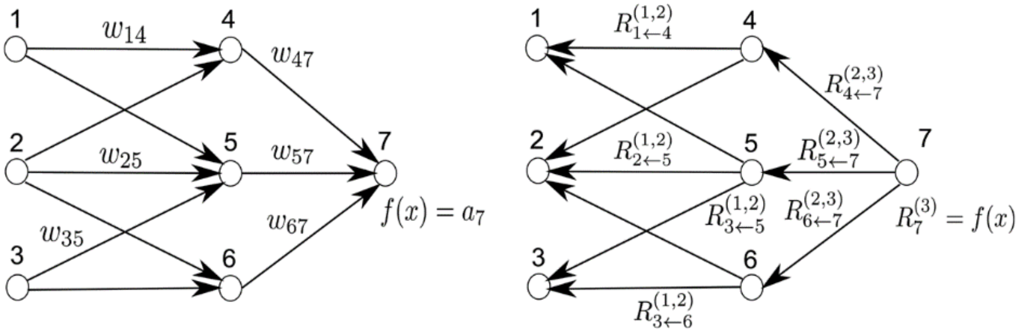

1
2
3
4
5
6
7
w_{14}
w_{47}
w_{25}
w_{57}
w_{35}
w_{67}
$f(x) = a_7$
1
2
3
4
5
6
7
$R^{(1,2)}_{1\leftarrow4}$
$R^{(2,3)}_{4\leftarrow7}$
$R^{(1,2)}_{2\leftarrow5}$
$R^{(2,3)}_{5\leftarrow7}$
$R^{(1,2)}_{3\leftarrow5}$
$R^{(2,3)}_{6\leftarrow7}$
$R^{(3)}_{7} = f(x)$
$R^{(1,2)}_{3\leftarrow6}$

Figure 2.5 Layer-wise relevance propagation. Left: forward flow of signal in NN, right: reverse flow of signal during LRP. Figure reproduced from (41)

Relevance at current layer *l* can be defined in compliance with conservation law as follows (41):

$$R_j^{(l)} = \sum_k R_k^{(l+1)} (a_j w_{jk} / \sum_h a_h w_{hk}), \qquad (2\text{-}10)$$

Where $R_k^{(l+1)}$ *is relevance of k-th neuron at previous layer (l+1) (in backwards direction),* a_j *is activation of current neuron at l,* w_{jk} *is weight between current neuron and k-th neuron at previous layer,* a_h *is h-th neuron activation at l.*

LRP has been applied to chemistry datasets (51). The authors used transformer architecture (52), trained on SMILES strings, and used output trained representations (embeddings) to build predictive models for a set of benchmark tasks. LRP allowed to identify input tokens important for activity of molecules. They supported this conclusion by showing 2 examples, which confirmed known functional groups relevant for aqueous solubility, and also known mutagenic groups (for Ames test model).

Subgraph methods

One example of methods operating directly on graphs is GNNExplainer (45). The main principle of the method is reducing redundant information in a graph which does not directly impact model’s decisions. For instance, if a subgraph is important, the prediction should be altered largely by removing or replacing it with another one. If removing or altering a feature does not affect the prediction outcome, the feature is considered not essential and thus should not be included in the explanation for a graph. GNNExplainer is formulated as an optimization problem, where a mutual information objective between the prediction of a graph neural network and the distribution of feasible subgraphs is maximized. Mathematically, the goal is to identify a subgraph $G_S \subseteq G$ with associated features $X_S = set(xj \mid vj \in G_S)$ that are relevant in explaining a target prediction $\hat{y} \in Y$ via a mutual information measure MI:

$$max_{GS}\, MI(Y,(G_S, X_S)) = H(Y) - H(Y|G= G_S,X= X_S) \qquad (2\text{-}11)$$

Besides, there is a secondary objective: the graph needs to be minimal. The paper addresses it by adding a loss (penalty term) for the number of edges. The authors tested their method on Ames mutagenicity classification dataset, and concluded that "GnnExplainer correctly identifies carbon ring as well as chemical groups NH2 and NO2, which are known to be mutagenic". However, they didn't show any example molecules. Currently, the method is applicable only to classification.

Models, interpretable by design

All the above methods are so-called *post-hoc*, i.e. they can be applied to model upon *it has been created*, however, there are also models interpretable by design. Simplest examples are linear regression, which is inherently explainable due to its simplicity (provided interpretable descriptors) and decision trees, provided they are not too deep. Far more complex, but interpretable method is *attention neural networks*. They implement a special architecture, allowing the model to learn only from those input features that are the most relevant. This is achieved via a special attention layer. The *weights* learned at this layer can be interpreted directly as importance scores of corresponding features. Attention can be applied to graphs too; examples of 1D-CNN (top) and graph CNN with *global attention pooling* (bottom) are shown on Figure 2.6. Unlike attention pooling, graph attention networks place *attention* between nodes (42), enabling *learnable* aggregation operation (Figure 2.7). In QSAR graph attention was applied to explain models on HIV, lipophilicity, Tox21 and FreeSolv datasets (53). Tang et al. utilized weights of a *self-attention* graph NN as explanations for lipophilicity and water solubility (25, 54). Zankov et al. applied attention to identify biologically-relevant conformations relying on attention-based multi-instance learning models (55).

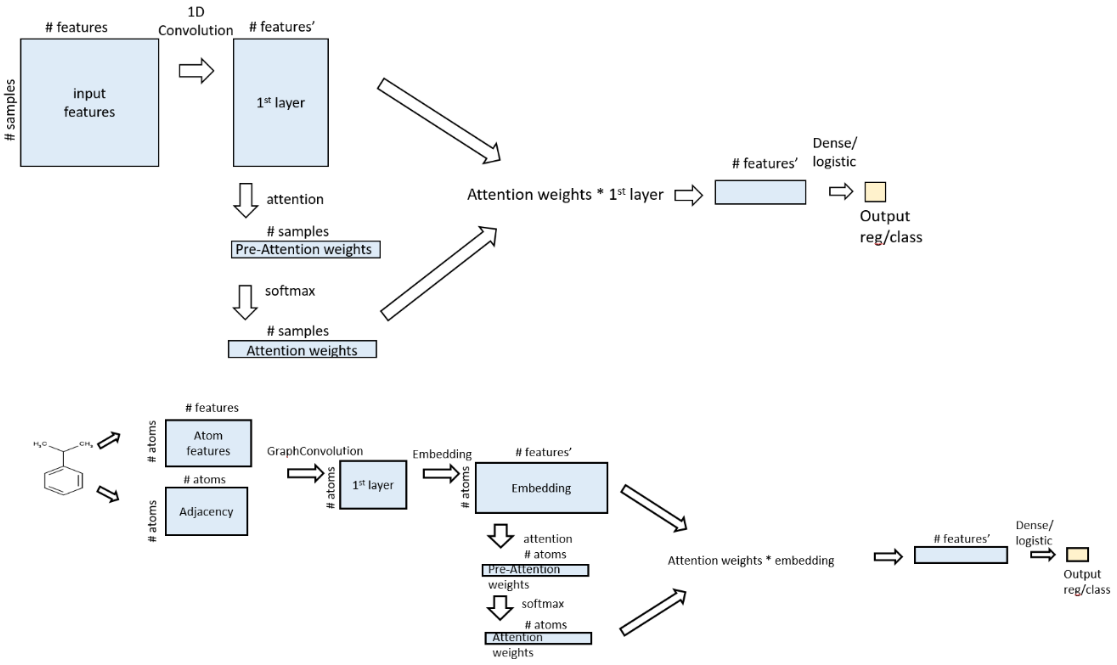


Figure 2.6.Illustration of attention mechanism in the context of 1D-CNN and graph CNN with attention pooling

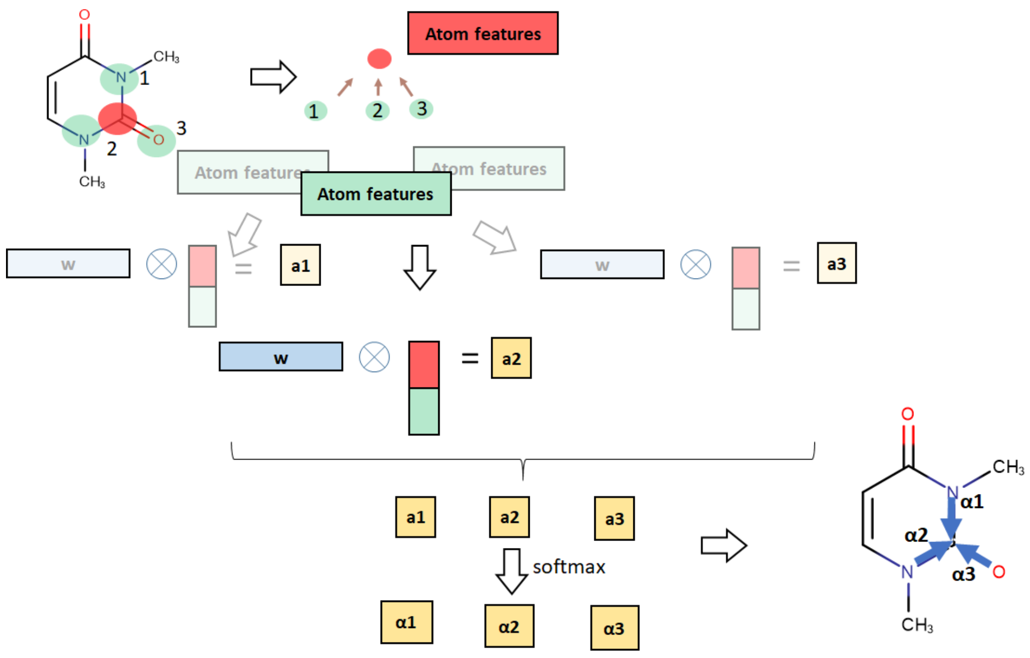


Figure 2.7 Illustration of graph attention networks. For each atom features are generated and the directed weight between an atom and its neighbor is determined as follows. Concatenated vector of features of the two is multiplied by **learnable w** and the result is passed to **softmax together with results of all other neighbors.** The output is **final attention weight** used then in node aggregation.

2.3. Model-agnostic interpretation methods

A number of methods can be applied to QSAR models universally. They become essential, when we employ multiple ML algorithms for which different interpretation methods are applicable, such as RF, GBM, SVM or NN (ML is discussed in 1.2). Moreover, *structural* methods compatible with any descriptors, even uninterpretable (e.g. topological indexes, discussed in 1.1) can be particularly convenient. The majority of such approaches are based on the following idea. The input is perturbed and the difference in results returned by model is measured. If the difference is large, then the feature is important. Examples are: universal approach for structural interpretation (UIA) (27), similarity maps (26), computational matched molecular pairs/series (44). All these methods can produce structural interpretation.

UIA implemented in SPCI software (56), (43), (27) removes any atom or fragment from the original molecule of interest, and then finds the contribution of that fragment as the difference between model's outputs as shown on Figure 2.8. In general the contributions of atoms (fragments) don't add up to molecules activity, i.e. method doesn't possess *completeness* property, that is expected because the majority of end-points are non-additive. The method is capable of capturing local behavior of the function, which is confirmed by experimental results (57). For instance, the same fragment in different local environment receives different contribution to overall activity, Figure 2.9. Carbonyl fragment in this example has high contribution to molecules activity when being conjugated with double bond or halogen, while having low contribution in other contexts (the activity studied in (57) was aquatic toxicity).

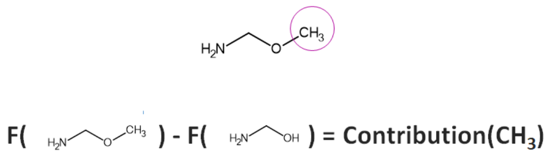


Figure 2.8. Schematic of UIA

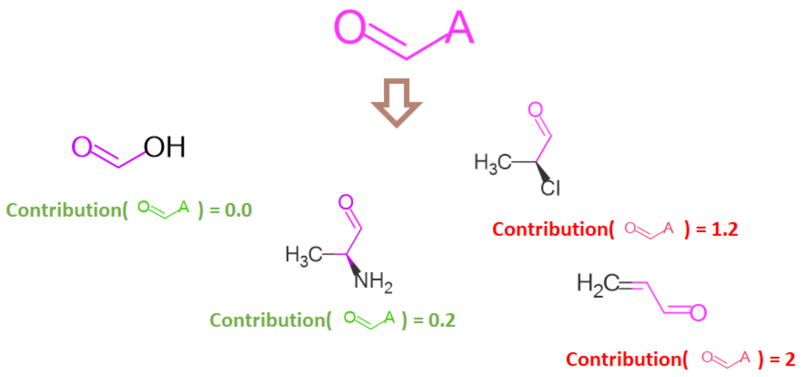


Figure 2.9 Illustration of the influence of molecular context on fragment's contribution within UIA framework. The fragment becomes activated when carbonyl appears in alpha position to chlorine or conjugated with a double bond.

Examples of feature-based model-agnostic methods include partial derivatives, feature importance by permutation, Shapley sampling values, LIME.

LIME by Ribeiro et al. (33) is a so-called surrogate method, because it finds an additive approximation of the function modelled in some local neighborhood of the point of interest. Coefficients in that equation are considered as contributions of features, which in turn should be interpretable. The approximation is done via sampling from neighborhood of given instance and subsequent fitting of regression line using LARS (58) method. Sampling here means perturbing original instance and using these new data points as input to the new linear classifier model, Figure 2.10. The loss function can be any, e.g. quadratic, with terms being weighed by their similarity with the original point (e.g. using exponential kernel for weighing). In chemistry context, we can use substructures/fingerprints as features, thus obtaining structural interpretation. LIME is applicable for both classification and regression.

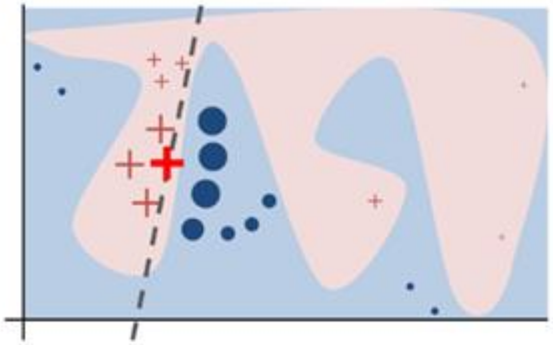


Figure 2.10. Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful. Figure reproduced from (33).

Downsides of the method are as follows:

- It requires interpretable descriptors, and these must be specifically chosen for each task. For some tasks it is not possible at all.
- Linear approximation can be too simple for particularly complex cases

3. Challenges in model interpretation and solutions proposed herein

Despite the fact that multiple interpretation approaches have been developed and new ones constantly appear, there are no suitable benchmarks to evaluate their applicability to interpretation of QSAR models. Often authors demonstrate applicability of their interpretation approaches on well-studied end-points like lipophilicity, solubility or toxicity where relevant patterns are well known (30). Interpretation is usually performed for pre-defined motifs or on a limited number of considered examples (19, 30, 31). For example, authors visually inspect a subset of molecules and compare calculated contributions with expert knowledge. Such non-systematic evaluation can be biased by a human expert and molecules chosen for analysis. Real data sets may have hidden biases which are difficult to control/reveal, some properties may depend on multiple factors or the response can be caused by different mechanisms of action. All these issues complicate proper validation of interpretation approaches based on real-world examples.

We propose creating synthetic data sets to overcome these issues. They can be designed in such a way that end-point values are pre-defined according to some *logic*, e.g. presence or absence of chemical patterns combined by Boolean operators determining compounds' activity (classification case). In regression case, the activity can be calculated as the sum of pre-defined atomic/fragment contributions. These data sets will be suitable to investigate the ability of models to capture the *logic* and the ability of interpretation approaches to retrieve it. Probing the data sets by creating a set of different models with different descriptors and then interpreting them serves as the "sanity check".

A distinct sub-task in this development was implementation of UIA for graph convolutional neural network (GC) models. GC models represent an important end-to-end modelling approach which doesn't need descriptors. Including these models into the study was necessary to complete the picture. Therefore, an adaptation of the approach was required.

Another major limitation of global structural interpretation approaches is that none of them take into account the molecular context of atoms or fragments considered. Context stands for neighboring and/or distant atoms which may affect properties of a given atom (fragment). They may significantly influence the fragment's behavior, e. g. transforming a "safe" non-toxic moiety into a reactive group. It has been demonstrated that consideration of the molecular environment may improve the outcome of the MMP analysis (59). Therefore, we expect that capturing molecular context will improve interpretation of QSAR models and explain variability of calculated contributions for identical fragments.

Herein we propose to combine UIA with Gaussian mixture modelling (GMM) which gives the ability to capture molecular environment for more correct interpretation.

Practical aspect of model interpretation was addressed by applying UIA and GMM-based extension developed here to real-world data sets: aquatic toxicity and anticancer activity (against cancer cell lines), with the aim of retrieving task-relevant patterns causal for specific biological response.

4. Aims of the dissertation

This dissertation addresses three main goals:

1. Development of the *validation framework* for QSAR model interpretation. In the framework we elaborate on designing *synthetic benchmark data sets* representing tasks of different complexity and develop metrics to evaluate interpretation accuracy. Using these benchmarks researchers will be able to *quantitatively* test interpretation methods using the data sets with *predefined ground truth* and reduced risk of bias.

   We will explore applicability of designed benchmarks using UIA. Additionally, we will study interpretability of widely used QSAR models. This will also allow to validate this interpretation approach more rigorously.
2. Development and real-case validation of an *extension* to UIA enabling *context-aware* interpretation. The Extension is based on Gaussian Mixture Modelling: clustering technique that helps to determine identical fragments occurring in different contexts. Those contexts correspond to neighboring (or more distant) atoms. The technique enables revealing not only important atoms/fragments but taking into account their environment.
3. Application of the developed and validated methodology to retrieve knowledge from QSAR models for selected biological activities. The results will show favorable/unfavorable fragments having influence on studied endpoints which will be further verified.

5. Materials and methods

The section describes the methodology used in our published works (57, 60). We do not describe here *new methodology developed* in this work as we present it in *Results and Discussion Section* (6).

5.1. The choice of endpoints for modelling and interpretation

QSAR becomes an important tool when the molecular target associated with activity is unknown or there is no single target at all. Such a situation can occur in lead optimization of ADME properties or prediction of toxicity end-points, in particular environmental toxicity or cytotoxicity where phenotypic screening is more widely used than biochemical assays. Below we describe biological endpoints chosen for modelling and interpretation in this dissertation.

Aquatic toxicity. Modeling of acute aquatic toxicity has a long history and is based on the premise that toxicity is governed by the ability of a chemical to reach the active site (e. g. pass through the cellular membrane) and its ability to interact there (61, 62). Many QSAR models for acute aquatic toxicity have been developed on a mechanistic basis (61, 62). However, despite extensive studies and numerous approaches proposed to allocate compounds to mechanisms of action, e. g. the Verhaar scheme (63), the schemes available are still limited in their applicability. Interpretation of QSAR models can augment knowledgebase of relevant toxicophores.

Anticancer activity. On early stages of anticancer research cell-based screening is used to identify promising active compounds. In our institute we collected and manage a large in-house database of outputs of cell-based anticancer assays. QSAR modeling of this data and subsequent interpretation of those models may reveal favorable patterns to design new active compounds and suggest possible mechanisms of action.

5.2. Data sets

Aquatic toxicity data sets

All data sets were obtained from the Toxicity Estimation Software Tool (T.E.S.T.), version 4.2, provided by the U.S. Environmental Protection Agency (64). All compounds were standardized according to the protocols proposed by Fourches and colleagues (65) . Briefly mixtures, inorganics, counterions, metals, and organometallic chemicals were removed. Moreover, specific chemotypes such as aromatic rings and nitro groups were normalized. In addition, we excluded duplicates as follows: (i) if duplicates had different biological activity, both compounds were excluded; and (ii) if the reported outcomes for the duplicates were the same, one chemical was retained in the dataset and the other excluded. To estimate the prediction ability of the models, we divided all data sets into training and test sets. For this purpose, the entire set of compounds was arranged in order of increasing toxicity, every fifth compound was placed in the test set, and the remainder in the training set. Training/test set distribution after removing duplicates is given in. The modeled endpoint for *Fathead minnow* and *Daphnia magna* was Log10(LC_{50} mol/L). The modeled endpoint for *Tetrahymena pyriformis* was – Log10(IGC_{50} mol/L). All datasets are given in the Supplementary material to our publication (60) (Tables 1S-3S).

Table 5.1 Aquatic toxicity data sets

Toxicity endpoints	Brief description	Number of compounds in the training/test set
***Fathead minnow*, LC_{50}, 96 h**	concentration of the test chemical in water in mg/L that causes 50 % of *Fathead minnow* to die after 96 h	642/161
***Daphnia magna*, LC_{50},48 h**	concentration of the test chemical in water in mg/L that causes 50 % of *Daphnia magna* to die after 48 h	268/67
***Tetrahymena pyriformis*, IGC_{50}, 48 h**	concentration of the test chemical in water in mg/L that causes 50 % growth inhibition of *Tetrahymena pyriformis* after 48 h	1424/356

T. pyriformis dataset for validation of Extension, developed in 6.4.

The data set was for the growth inhibition of the ciliated protozoan Tetrahymena pyriformis represented as lg(1/IGC50) (IGC50 in mol/l). Toxicity data were taken from the study of Ruusmann and Maran (66). Standardizer was used for structure standardisation and tautomerisation, in addition structures were checked for errors (67) and duplicates were removed. The data set curation workflow is available from the public repository – https://bitbucket.imtm.cz/projects/STD/repos/std/browse. The curated data set comprised 1984 compounds whose structures and activity values are provided in Supplementary materials to our publication (57). All modelling steps including descriptor calculation, model development and validation, molecule fragmentation and calculation of fragment contributions were performed with the open-source spci software (68).

Anticancer activity

All datasets were compiled from IMTM proprietary database. We used MTT assay data on cytotoxicity of compounds against six cell-lines: HCT116: colon carcinoma; HCT116-p53-/-: same, but with knocked out p53 gene (p53 null); K562: chronic myeloid leukemia; K562-TAX: same, but with acquired taxol-resistance; CCRF-CEM: acute lymphoblastic leukemia; and CEM-DNR: same, but with aquired daunorubicin-resistance. Curation protocol was the same as for *T. pyriformis dataset for validation of Extension*. Curated datasets comprised around 3800 compounds:

- HCT116 – 3841
- HCT116-p53-/- – 3849
- K562 – 3858
- K562-TAX – 3852
- CCRF-CEM - 3878
- CEM-DNR - 3845 molecules.

Most of them (3763) were the same for all data sets. Minor difference was due to incomplete activity data. For the ease of modelling and to reduce heterogeneity of data we labelled compounds with IC_{50} <= 10 μM as actives and the rest – as inactives. Thus, we switched from regression to classification task . All data sets had unbalanced classes: number of actives was substantially smaller relative to the number of inactives. This usually causes difficulties for modeling. Therefore, we applied the special multiple undesampling technique prior to modelling (see 5.4) which should solve this issue.

5.3.Descriptors

For modelling of benchmark data sets we employed the following descriptors: atom-pair fingerprints which enumerate pairs of particular atoms at a topological distance from 1 to 30 (AP), Morgan fingerprints representing atom-centered substructures of radius 2 (MG2), RDK fingerprints enumerating all possible substructures with atom count from 2 to 4 (RDK) and topological torsion fingerprints enumerating all possible linear four-atomic substructures (TT). AP, MG2 and RDK fingerprints were also used in their binary (bit vector) form of length 2048 (denoted bAP, bMG2, and bRDK). All fingerprints were calculated using RDKit (69).

For aquatic toxicity modelling we used counts of fragments having 2-4 heavy atoms as the descriptor set. Fragments with either all atoms connected or containing two disconnected parts were enumerated. Atoms were labelled according to their partial charge, lipophilicity, refractivity and ability to form H-bonds calculated using Chemaxon cxcalc utility (67). Descriptors were calculated by the *sirms* software (70).

For anticancer activity modelling Morgan fingerprints representing atom-centered substructures of radius 2 (MG2) were applied. Experiments with other descriptors didn’t result in any benefit in terms of model quality.

5.4. QSAR modelling

For all machine learning tasks models were built using Random Forest (RF), Partial Least Squares (PLS), Gradient Boosting Machine (GBM) and Support Vector machine with Gaussian kernel (SVM) from *Scikit-learn* Python package (71). Hyper parameters were optimized by the grid search in the course of five-fold cross-validation. A consensus model (obtained by averaging predictions of individual models which had appropriate Q^2 and RMSE) was used for interpretation in the case of all *aquatic toxicity* data sets. We used SPCI software which automates overall modeling workflow and interpretation (43). In the case of unbalanced data sets (cancer cytotoxicity end points) multiple undersampling technique (62) was applied. Consensus models were obtained based on models built for each subsample. In the modelling of benchmark datasets we additionally trained graph convolutional neural network models (GC) using DeepChem (72). This approach does not require fingerprints and learn internal representation of molecules in the course of modeling. We used the default architecture with 2 graph convolutional layers, each of size 64 and a *GraphPool* layer after each convolution (*GraphPool* performs max pooling on each atom's neighborhood). Output from the *GraphPool* layer was fed to "atom-level dense layer" (size 128) and global sum pooling *(GraphGather)* followed by linear or logistic regression layer depending on the task (regression, classification), Figure 5.1. We didn't apply batch normalization. GC models can be considered as models trained on "learnable" Morgan fingerprints of radius 2. For training of GC models validation subsets (15%) were separated from training sets to tune model hyper parameters. 1-nearest neighbor (1-NN) models as baseline models to examine data set modelability (73). Poor performance of 1-NN models would indicate that compounds are not easily distinguished within chosen descriptor space, indirectly indicating that data sets don't have an obvious bias.

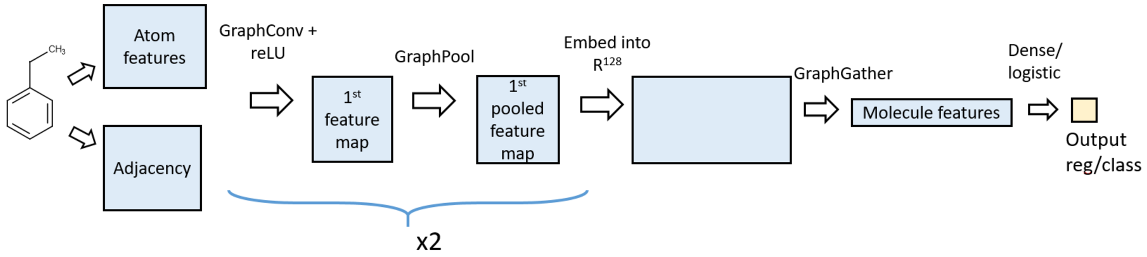


Figure 5.1 The architecture of GC model employed (72)

5.4.1. Models' performance metrics.

Predictive performance of models was assessed using cross-validation and, for benchmarking tasks, test sets. Q^2 and RMSE values were calculated for regression tasks and sensitivity, specificity and balanced accuracy were calculated for classifications tasks *(5-1)* - (5-5). Q^2 ranges from 0 to 1 (note, in principle, it can take negative values), with 1 indicating the best performance. RMSE ranges from 0 to infinity, with 0 being the best result. Sensitivity, specificity and balanced accuracy take values from 0 to 1, with 1 indicating perfect model.

$$Q^2 = 1 - \frac{\sum_i (y_{i,pred} - y_{i,obs})^2}{\sum_i (y_{i,pred} - \bar{y}_{obs})^2} \tag{5-1}$$

$$RMSE = \sqrt{\frac{\sum_i (y_{i,pred} - y_{i,obs})^2}{N}} \tag{5-2}$$

$$specificity = \frac{TN}{TN+FP} \tag{5-3}$$

$$sensitivity = \frac{TP}{TP+FN} \tag{5-4}$$

$$balanced\ accuracy = \frac{sensitivity + specificity}{2} \tag{5-5}$$

Where for regression:

y(i,pred) is predicted activity

y(i,obs) is observed activity

N is the number of molecules in the dataset

For classification:

TN – true negative count, number of correctly predicted inactive molecules

TP – true positive count, number of correctly predicted active molecules

FN – false negative count, number of incorrectly predicted active molecules

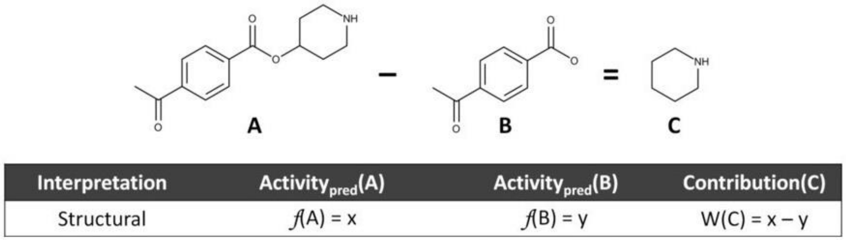
FP – false positive count, number of incorrectly predicted inactive molecules

5.5.Model Interpretation: Calculation of Fragment Contributions

In this dissertation the UIA for interpretation was applied; this utilizes the concept of matched molecular pairs. The approach can be summarized as follows. For a compound

A consisting of two fragments B and C the contribution of fragment C can be calculated as the difference between predicted activity values for the initial compound A and the counter-fragment B (obtained by removal of the fragment C from the molecule A) Figure 5.2. In this way the overall contribution of the fragment C in the units of a studied activity was calculated. In the case of classification, the difference is taken in terms of class probabilities. This is local (instance-based) interpretation which gives information about the contribution of a fragment in individual compounds. Aggregating and averaging of contributions of identical fragments allows for the ranking of different fragments and reveals general trends in structure-activity relationships (global interpretation). In this dissertation we improved on the global interpretation by means of using GMM-based clustering (see below, 6.4).

This interpretation approach is applicable not only to individual models but also to consensus models comprising multiple individual models. This property was extremely useful in the case on interpretation of models obtained by multiple undersampling technique to predict anticancer activity.



Interpretation	Activity_{pred}(A)	Activity_{pred}(B)	Contribution(C)
Structural	f(A) = x	f(B) = y	W(C) = x − y

Figure 5.2 Scheme for the structural interpretation of QSAR models.

Fragmentation of Molecules

For the purpose of model interpretation, we used two schemes of fragmentation: atom-based and fragment-based. In the former case every individual atom was removed for calculation of a contribution. This scheme was used for benchmark data sets only. In the latter case larger fragments were removed. They were enumerated by means of RDKit. Bonds to be cleaved during fragmentation were determined by a SMARTS pattern [!#1]!@!=!#[!#1]. All possible fragments having at most three attachment points were generated from the training set compounds. This scheme was applied to aquatic toxicity and anticancer activity tasks.

In *T. pyriformis* study contributions were calculated only for those fragments whose counter-fragments had at least two atoms since only such structures can be properly encoded by the descriptors used. In benchmark study contributions were calculated only for those fragments whose size was not more than 40% of molecule's size in terms of heavy atoms.

To virtually remove atoms we used the scheme similar to that used by Sheridan in his study (74)We replaced removed atoms with dummy atoms (with atomic number 0) and calculated descriptors. This resulted in appearance of new descriptors encoding dummy atoms, but these descriptors did not occur in the training sets, so the models ignored them during prediction. Thus, these atoms disappeared for models and prediction was made based on descriptors of the remaining part of the molecule. This scheme was implemented in spci software for RDKit-based fingerprints.

5.6.Molecular docking

For docking we used Autodock Vina 1.1.2(75). Both protein and ligands were prepared using Autodock Tools Software, which included water removal from protein, addition of hydrogens and manual check of aromatic atoms and torsions for ligand. All ligands were docked into a rigid binding site with grid size of 20 × 20 × 20 points with default step 1 Å and the top scored poses were saved.

6. Results and discussion

6.1. Development of the validation framework for QSAR model interpretation

This section describes the benchmark datasets and methodology developed by Matveieva and Polishchuk (76).

6.1.1. Design of synthetic datasets

Synthetic data sets with pre-defined patterns allow for systematic evaluation of QSAR interpretation approaches, because calculated contributions of atoms (fragments) can be compared with true values. These values are defined by the incorporated logic ("ground truth"). This logic can be, for example, presence or absence of chemical patterns combined by Boolean operators that determine compounds' activity (in the case of classification). In regression case, the activity can be calculated as the sum of pre-defined atomic / fragment contributions.

We created six data sets by selecting compounds from the ChEMBL23 database (Table 6.1), which was used as a source of chemically relevant structures. Structures of all compounds were standardized; duplicates were removed, as were compounds with a MW > 500. For all retained compounds we assigned contributions to atoms or groups of atoms according to the rules described below, and then calculated "activities" of compounds. To design regression data sets, we randomly selected compounds from the pool, with distributions close to normal (Figure 6.1). To design balanced classification data sets, we randomly selected equal number of compounds belonging to each class.

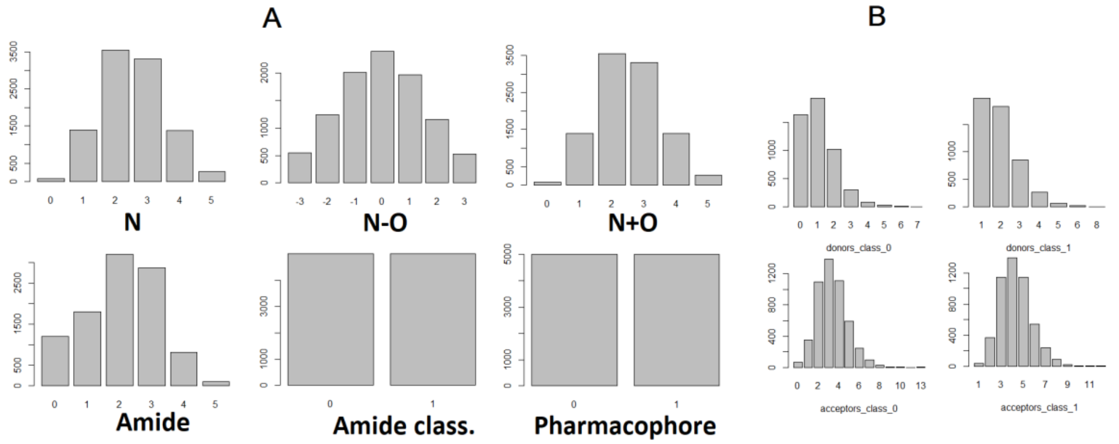


Figure 6.1 A) Distributions of modelled property in the datasets; B) Class-wise distributions of hydrogen bond donors and acceptors for the pharmacophore dataset

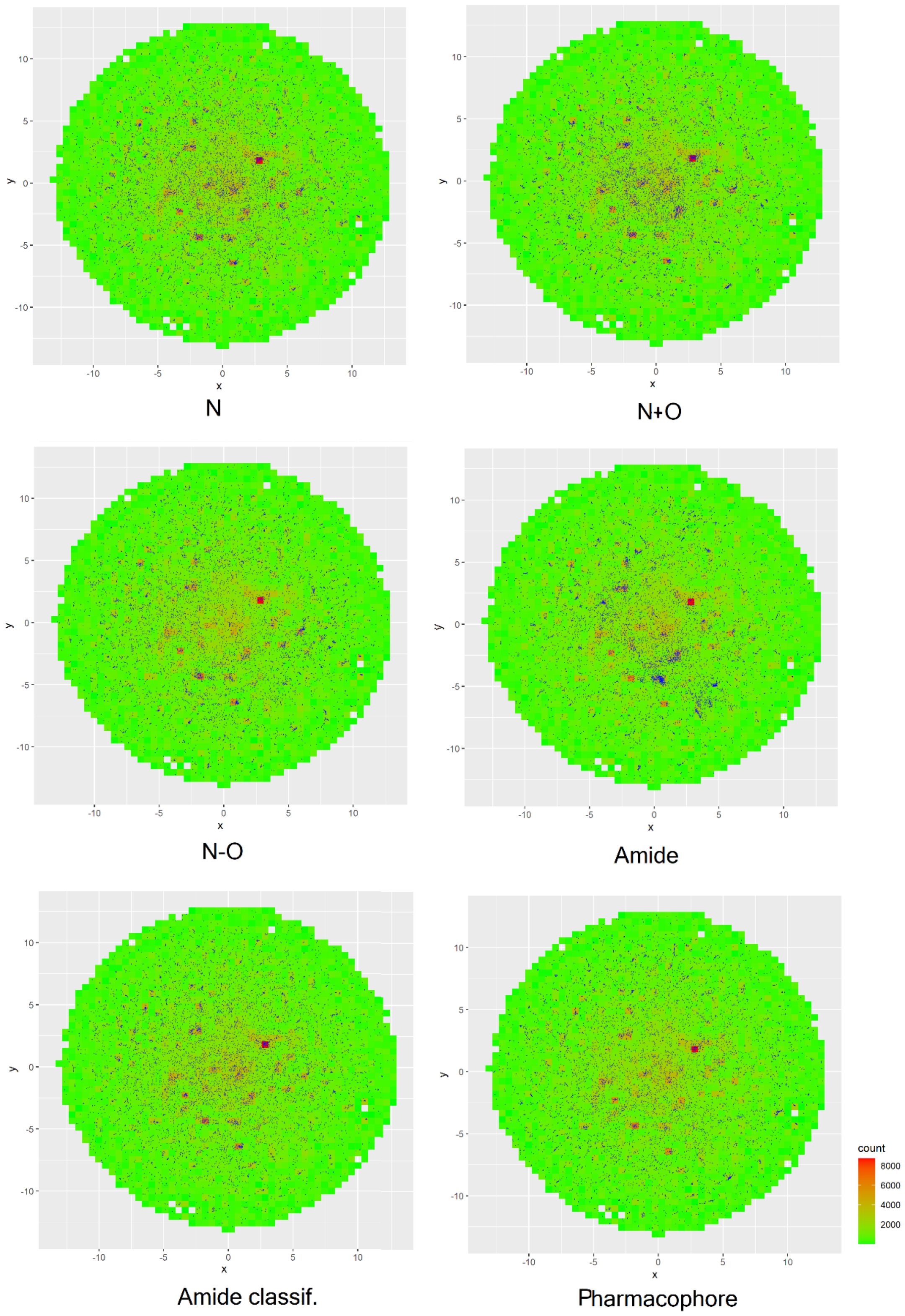
Three data sets represented simple additive properties. Patterns were defined as occurrence of certain atoms. The end-point of the first dataset (N data set) was the sum of nitrogen atoms. Thus, the expected contributions of nitrogen atoms were 1, and all other atoms - 0. The end-point of the second data set (N-O data set) was the sum of nitrogen atoms minus the sum of oxygen atoms. Thus, oxygen represented a negatively contributing pattern. Expected contribution of any nitrogen was 1, any oxygen -1, and all others 0. These first two datasets resemble simple additive properties like molecular weight, lipophilicity. The end-point of the third dataset (N+O dataset) was the sum of nitrogen and oxygen atoms divided by two, given that the number of nitrogens and oxygens in a molecule was strictly equal. Thus, two positively contributing patterns were co-occurring and both contributed equally to the endpoint. This represents a specific case to verify how a model treats correlated patterns and how this affects interpretation output. Modeling algorithm can treat nitrogens and oxygens equally or select only one of them as important feature. Both these cases will result in correct predictions. However, different interpretation output may result from rebuilding the model if it randomly prioritizes one of correlated features. The same may happen if correlated features are removed before model building and during analysis of interpretation outcomes the discarded features are not considered. Depending on which scenario will be realized, the interpretation output may be incomplete and/or misleading.

We calculated the correlation of selected atomic patterns with other elements to ensure that there is no explicit bias in data sets (see Appendix, Table 0.1). However, this does not guarantee that there are no correlations with more complex patterns.

Two other data sets represent additive end-points depending on local chemical context: this again resembles the case of additive properties like lipophilicity, but considering groups is more realistic than considering atoms alone. They were collected independently and consisted of different compounds. The end-point of the first one was the number of amide groups encoded with SMARTS NC=O. Thus, this was a regression task. The second one was a classification task, where compounds were considered active if they had at least one amide pattern and inactive otherwise. The expected contribution of any amide atom for both data sets was 1, because by removing such an atom the whole pattern disappears. This should result in decrease of predicted property value by 1 (or switch compound’s class), except for the following case. If a compound contains multiple amide groups, classification issue may occur, because there is no single group which determines the activity (we discuss results for this case in 6.2).

The last data set was designed based on a pharmacophore hypothesis and represents property depending on whole-molecule context. Compounds were labeled as active if at least one of their conformers had a pair of an H-bond donor and an H-bond acceptor 9-10 Å apart. If the pattern occurred in more than one conformer of a molecule, this had to be the same pair of atoms. Therefore, actives contained exactly one pharmacophore pair consistent across all conformers. If this pattern was absent in all conformers a compound was labeled inactive. Compounds with multiple pharmacophore pairs were excluded to avoid ambiguity in subsequent interpretation. We generated up to 25 conformers for each compound using RDKit. H-bond donors and acceptors were labeled using *pmapper* software (77). We ensured that distributions of H-bond donors and acceptors in active and inactive classes were similar (Figure 6.1). Atoms which were true pharmacophore centers had expected contribution 1; all other atoms - 0. This example is closest to a real case scenario. However, we used a two-point pharmacophore to ease modeling and interpretation. Using more complex pharmacophores with more features might require 3D descriptors, however we used 2D representation.

We verified that all data sets are representative of the source database and distributed similarly to it (Figure 6.2). For this purpose, we embedded molecules in binned t-sne plot (74) generated from ChEMBL23 database using implementation (75). Original input feature space was chosen to be 2048-dimensional MHFP6 fingerprints (76). Upon mapping the database to *t-sne* plot we binned it to 50×50 cells and visualized each of our datasets on it (blue dots). As can be seen, datasets cover major portion of ChEMBL without apparent bias. To create training and primary test sets all data sets were randomly split in 70/30 ratio. All datasets are provided in the repository https://github.com/ci-lab-cz/ibenchmark.



N

N+O

N-O

Amide

Amide classif.

Pharmacophore

Figure 6.2 All 6 datasets (dark-blue points designate molecules) embedded in binned t-sne plot (78) generated from ChEMBL23 database using implementation (79). Original

feature space for t-sne: 2048-dimensional MHFP6 fingerprints (80); perplexity: 50. Number of bins: 50×50.

Table 6.1. Synthetic data sets to benchmark interpretation of QSAR models (76).

Dataset	Property type	End-point	Train/test set size	Expected atom contribution
N	Regression; *atom-level additive property*	sum(N)	6995/2999	Nitrogen atoms:*1*; others: *0*
N-O	Regression; *atom-level additive property*	sum(N) - sum(O)	6893/2969	Nitrogen atoms: *1*; Oxygen atoms: -*1*; others: *0*
N+O	Regression; *atom-level additive property with hidden correlation*	(sum(N) + sum(O)) / 2, where sum(N) = sum(O)	7000/3000	Nitrogen and Oxygen atoms: *0.5*; others: *0*
Amide_reg	Regression; *group-level additive property*	sum(NC=O)	7000/3001	any atom of amide groups: *1*; others: *0*
Amide_class	Classification; *property determined at group level*	active: if sum(NC=O) > 0 inactive: if sum(NC=O) = 0	6998/3000	any atom of amide groups: *1*; others: *0*
Pharmacophore	Classification; *property determined*	active: at least one conformer with exactly one	7000/3000	atoms which are HBA or HBD of the pharmacophore:

	at whole-molecule level	pharmacophore match (same two atoms in all conformers); inactive: no pharmacophore matches for all conformers; pharmacophore match: HBD and HBA 9-10 Å apart		*1*; others: *0*.

6.1.2. Extended test sets

We created an additional *extended* test set for each task. Thus we could reveal possible weaknesses in data sets and challenge the generalization ability of trained models. Structures from primary test sets were subject to small perturbations. This was performed by applying the *mutate* operation from the CReM tool (81). We used the previously generated fragment database based on compounds from ChEMBL22 having maximum synthetic complexity score 2.5 (81, 82). This database contains fragments and their contexts of a given radius from corresponding attachment points. Fragments occurring in the same context are interchangeable and result in chemically valid structures. We chose context radius 3 and made all possible replacements of groups of up to three atoms with other groups of up to three atoms from the database. For each primary test set we generated about 300 000 new analogues and assigned end-point values to each compound using the same rules as for the corresponding data set (Table 6.1). The extended test sets provided more diverse examples of chemical space represented by primary test sets. The performance on these datasets is reported in Appendix, Figure 0.1.

6.1.3. Interpretation quality metrics

For benchmarking purposes, it is desired to have standard means/metrics to quantify interpretation quality. Within benchmarking we propose three metrics: *ROC-AUC*, *top-n-score*, *RMSE.*

We considered each molecule individually and analyzed the ability of interpretation methods to rank atoms in proper order, i.e. atoms with greater expected contributions should be ranked higher. (This is so-called local or instance-based interpretation.)

Metrics (where applicable) were computed separately for positively contributing atoms (hereafter positive atoms) and for negatively contributing atoms (hereafter negative atoms). Positive atoms increase activity in regression case or favor positive class prediction in classification case. Conversely, negative atoms decrease activity or favor negative class prediction.

ROC AUC

ROC AUC is an integral metric which demonstrates how well relevant atoms are ranked over others within a particular molecule. To get the final score we averaged AUC values for all considered molecules. In QSAR interpretation context this metric was first used by McCloskey et al (46). To calculate AUC for positive patterns (AUC^+) we set all negative atoms' labels to 0. Thus, AUC^+ characterized how highly positive patterns were ranked. To calculate AUC for negative patterns (AUC^-) we set negative atoms' labels to 1 and all others to 0. It worth noting that AUC cannot be calculated for molecules having no patterns pre-defined for a given dataset (expected contributions of all atoms are 0). Therefore, these molecules were not considered for the calculation of average AUC for individual data sets.

The weakness of ROC-AUC is that it is an integral measure and accounts for both relevant and irrelevant patterns. In practice it is more reasonable to find relevant features. To address this, we propose top-n score.

Top-n score

Top-n score is calculated as follows and should be more stringent:

$$\text{top-n score} = \frac{\sum_i m_i}{\sum_i n_i}, \qquad (6\text{-}1)$$

Where n_i is the total number of positive (negative) atoms in the i-th molecule, m_i is the number of positive (negative) atoms in n_i top ranked atoms according to their calculated contributions.

For instance, if a molecule has two true patterns with expected contributions +1 and interpretation retrieved only one of them among top two contributing patterns, the molecule will contribute n = 2 and m = 1 to the equation above. Top-n is an integral characteristic of a data set and varies from 0 to 1 (perfect interpretation). Analogously we calculated bottom-n score to estimate the ability to retrieve negative patterns.

RMSE

Additionally, we calculated root mean square error (RMSE) of predicted contributions for each molecule and averaged them across molecules in a data set to estimate deviation of calculated contributions from the expected values. This is less important metric, because proper ranking is more practically valuable than exact estimation of contributions which are generally unknown in real cases. But RMSE should be helpful for benchmarking purposes because it allows to investigate decision making of models and interpretation methods from another point of view and may reveal weaknesses or advantages not captured by other metrics.

The metrics described were implemented in the open-source repository to facilitate calculation of interpretation performance – https://github.com/ci-lab-cz/ibenchmark.

6.1.4. Chapter summary

In this chapter we elaborated on development of synthetic data sets for validation/benchmarking of interpretation methods. We designed them in a way that assigns labels to all atoms of molecules from a given set, so that interpretation can be quantitatively evaluated. This is achieved via comparison of results against those pre-assigned labels, and subsequent computation of metrics, which are proposed herein. (Any other metrics can be used as well.) Also, synthetic data set design allows to control the distribution of molecule-level labels (i.e. outcome variable; not to be confused with atom labels), which is expected to aid high quality modelling and interpretation, given robust and suitable methods.

6.2. Applying the framework to study QSAR model interpretability

This section summarizes the results of applying the framework developed in 6.1. to study the quality of interpretation produced by UIA (76). Present study is limited to a single interpretation approach; however, the framework applies to any fragment/atom-based method. Additionally, we analyzed the behavior of various fingerprints and ML methods in terms of their influence on interpretation results.

6.2.2. Implementation of UIA for graph convolutional neural networks.

In this work we used descriptor-based *Scikit-learn* models along with *Deepchem* GC models. The latter is the end-to-end model which doesn't require descriptors and directly uses molecular graph to learn *molecular representation*. Therefore, to enable estimating atom contributions the procedure described in 5.5 (*Fragmentation of molecules)* was modified as follows.

Deepchem GC models convert each molecule into two input matrices at the preliminary featurization step: 1) atom basic features and 2) connectivity information. To virtually remove an atom, we remove the corresponding row from the first matrix and adjust the connectivity table (the second matrix) respectively, Figure 6.3. These modified matrices are supplied as input for a GC model which makes prediction. The contribution is calculated by subtracting the predicted end-point value for a compound with a virtually removed atom from the value predicted for the whole structure. This feature was integrated into Deepchem, *version 2.3.* Examples can be found in the project's documentation: *Tutorial # 28. Atomic contributions for molecules* (83).

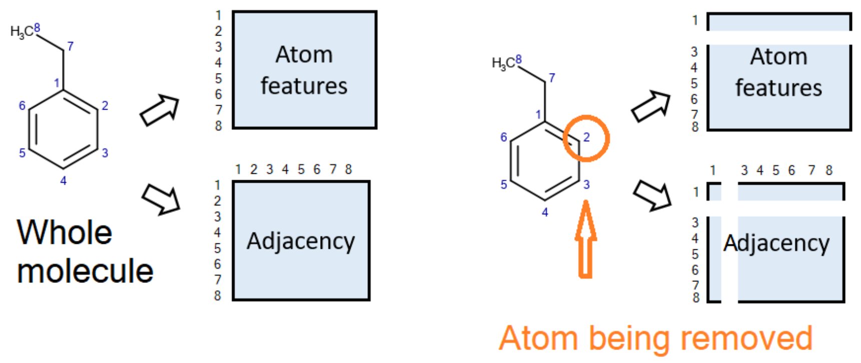


Figure 6.3 Scheme of removing of atoms when interpreting GC models

6.2.3. Results: interpretation performance

The quality of interpretation of models built on synthetic data sets was analyzed. We carried it out by evaluating three quality metrics described in 5.4.1. Throughout we tracked the dependence between model performance and interpretation quality, and paid attention to features (outliers, unusual behavior etc.) appearing in each data set. Models performance is discussed in more detail in Appendix, Figure 0.1.

N data set

High average AUC^+ values observed in the majority of cases indicate that atoms were ranked correctly (Figure 6.4a). For models having test set $R^2 > 0.81$ average AUC^+ was greater than 0.9 (Figure 6.4b). There is a clear correspondence between model predictivity and the ability to rank atoms. However, there were few outliers, namely SVM and PLS models trained on binary RDK fingerprints. AP, bAP and RDK fingerprints resulted in the highest average AUC^+ irrespective of machine learning method.

Top-n score characterizes the ability to rank true atoms on top. This metric was more sensitive to changes of R^2 than AUC^+ (Figure 6.4c-d). The most predictive models had high top-n scores. These were PLS, RF and GBM models trained on AP and bAP fingerprints and RF and GBM models trained on count-based Morgan fingerprints (top-n = 0.92-1.0). GC model had somewhat lower score (0.89) followed by models trained on count-based RDK fingerprints (top-n = 0.65-0.81). RMSE values varied in a wide range but followed the same trend as top-n (Figure 6.4e-f). Therefore, we did not analyze them in detail. It is worth noting, that models trained on binary atom pairs had similar interpretation performance to those trained on count-based atom pairs, whereas the former had poorer predictive ability. For other pairs count-based fingerprints always outperformed the corresponding binary ones.

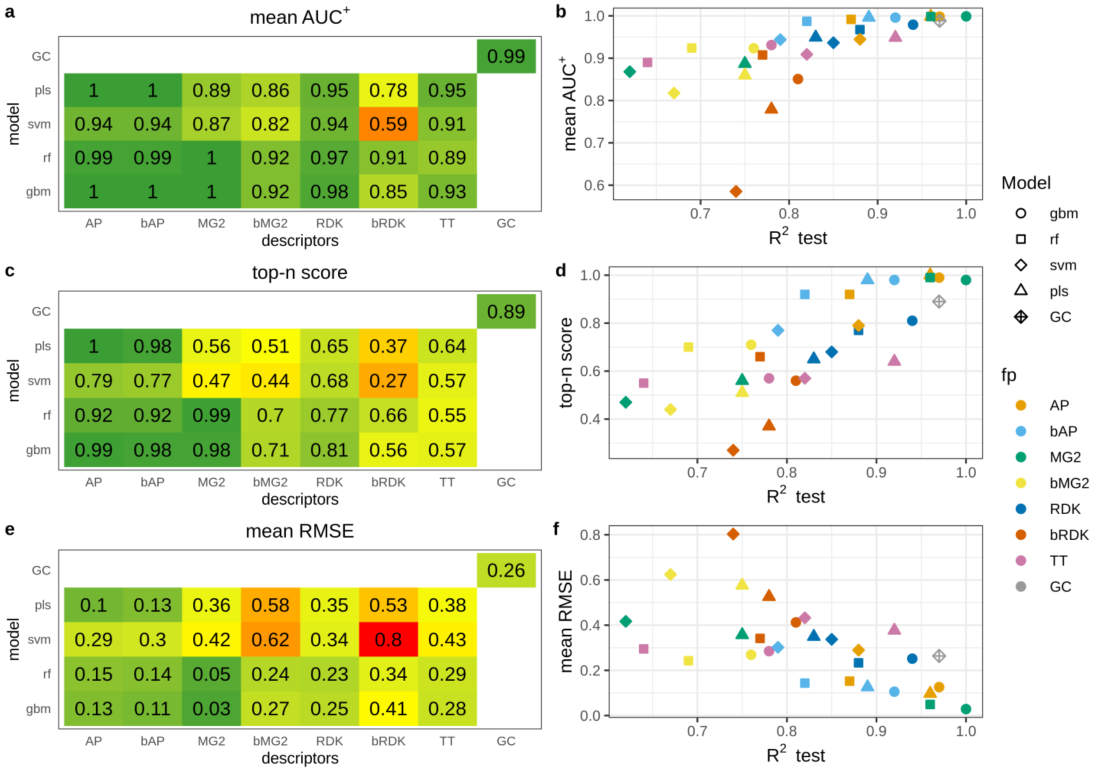


Figure 6.4. Interpretation performance of models trained on the N data set. Figure reproduced from (76)

N-O data set

This data set contained positive and negative atoms: nitrogen and oxygen, respectively. We found that overall ranking abilities for positive and negative patterns measured by AUC^+/AUC^- were similar (Figure 6.5a,c). The same agreement was observed for top-n and bottom-n scores (Figure 6.5e,g). Thus, models were able to detect positive and negative patterns with comparable accuracy. The only outlier was PLS models trained on binary Morgan (AUC$^+$ = 0.77, AUC$^-$=0.64, top-n = 0.35, bottom-n = 0.2) and RDK fingerprints (AUC$^+$ = 0.74, AUC$^-$=0.87, top-n = 0.29, bottom-n = 0.47). Because of high correspondence in positive and negative pattern detection we will discuss only positive patterns.

The relationship between model predictive ability and interpretation accuracy was less stringent then in the case of the N data set, but still remained. Models trained on AP and bAP fingerprints resulted in stably high interpretation performance (AUC$^+$ = 0.91-1.0, top-n = 0.61-1.0). RMSE was in agreement with other metrics.

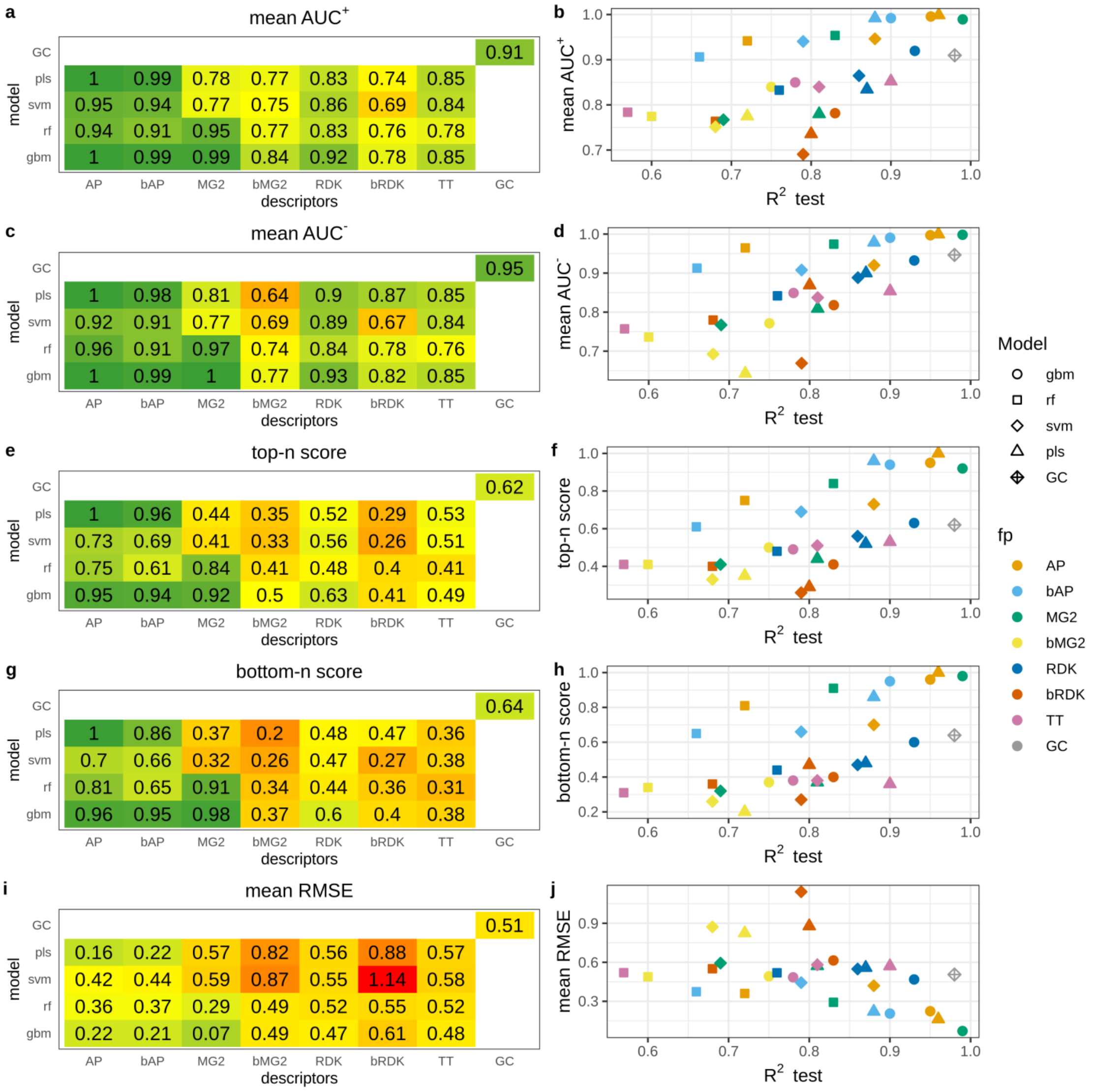


Figure 6.5. Interpretation performance of models trained on the N-O data set. Figure reproduced from (76)

Interpretation of the GC model showed decreased interpretation accuracy. For instance, top-n score was 0.62 for the GC model, whereas corresponding values for PLS and GBM models trained on AP descriptors were 1.0 and 0.95. However, all these models had comparable predictive performance (Figure 6.5c,d). We randomly chose subset of 100 molecules to inspect the behavior of the GC model. Atoms proximate to true positive/negative ones were often misrecognized; for instance, carbon atoms in carbonyl the lowest (pink). Atoms attached to nitrogen were ranked the highest (green) (Figure 6.6a-f). Some nitrogens in nitro groups were misinterpreted as negative (Figure 6.6a). Aromatic carbons were also falsely recognized as positive a number of times (Figure 6.6c,e).

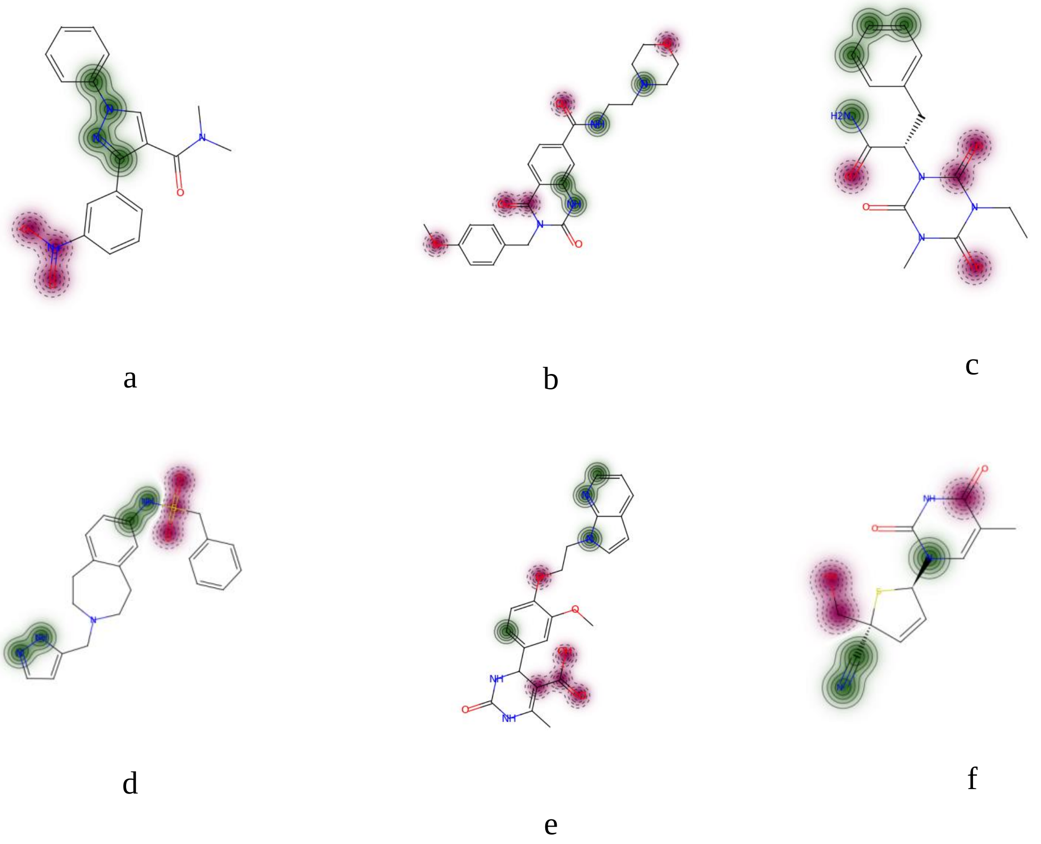


Figure 6.6. Top-scored (green) and bottom-scored (pink) atoms by the GC model for the N-O data set. The number of top and bottom highlighted atoms is equal to the total number of positive (nitrogen) and negative (oxygen) atoms in corresponding molecules. Figure reproduced from ((76)

Regression amide data set

The overall ranking ability was very similar to the N data set (Figure 6.7). SVM model trained on binary RDK fingerprints had very low interpretation accuracy ($AUC^+ = 0.62$) close to random ranking (0.5) and this was the main outlier from the trend. The relationship between top-n score and model predictive ability was quite stringent with two outliers: SVM models trained on binary and count-based RDK fingerprints (Figure 6.7c-d). While the former had relatively low score (0.38) the latter had the higher score (0.93) than other models with comparable predictive ability. It should be noted that models built on count-based RDK fingerprints were among the strongest in terms of interpretation accuracy. For example, SVM model trained on count-based RDK fingerprints had $R^2_{test} = 0.85$, $AUC^+ = 0.97$ and top-n = 0.93, while RF model trained on count-based Morgan fingerprints had much higher predictive ability ($R^2_{test} = 0.97$), but comparable interpretation accuracy ($AUC^+ = 0.99$, top-n = 0.94). The GC model had slightly lower interpretation performance than models of comparable predictive ability, similarly to the case of the N-O data set.

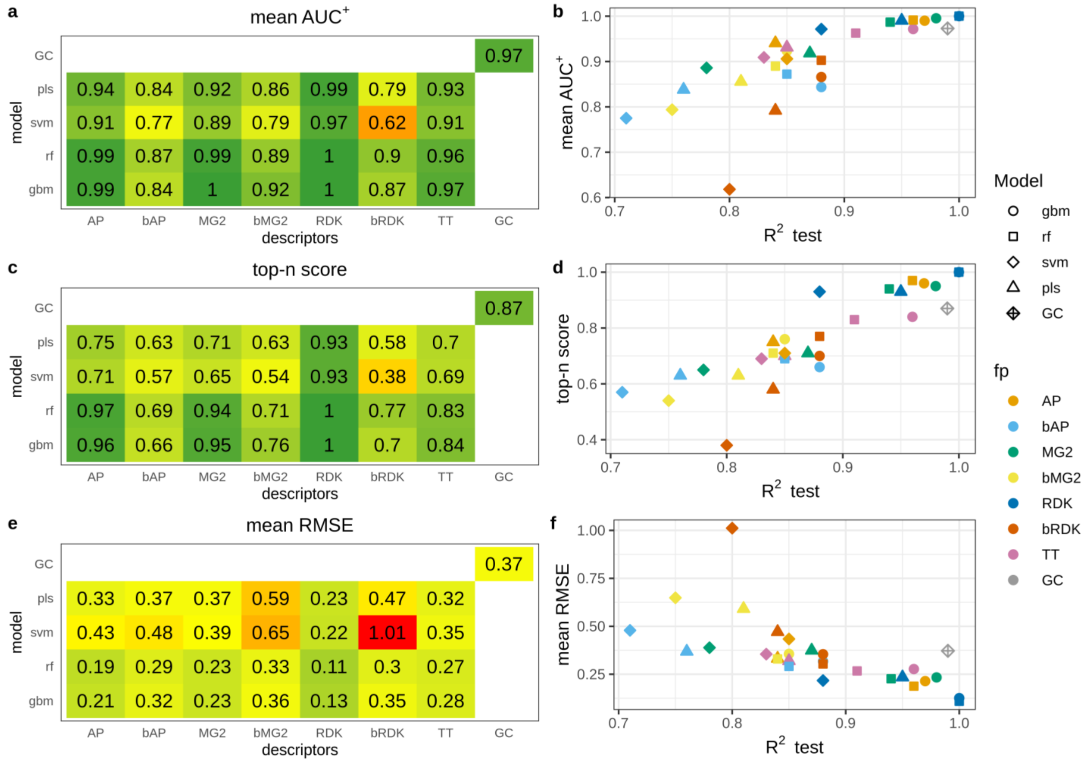


Figure 6.7. Interpretation performance of regression models trained on amide data set. Figure reproduced from (76)

Classification amide data set

This data set was simple for modeling and all models achieved high balanced accuracy (≥ 0.93). However, overall ranking ability for atoms of amide groups varied in a wide range (AUC^+ = 0.82-0.98) (Figure 6.8 a-b). There was no dependence between predictive ability and interpretation accuracy. This is a consequence of the interpretation approach which virtually removed an atom and calculated the contribution of the removed part as the difference between predicted active class probabilities. If the property depends on the presence or absence of a particular pattern, but there are multiple such patterns in a molecule, then removing one of them will keep the remaining structure active and the calculated difference will be small or even zero.

This effect was most pronounced in the case of top-n score which was below 0.5 for high quality models (Figure 6.8 c-d), meaning that only half of true atoms were ranked on top.

To confirm the issue of interpretation approach we investigated interpretation accuracy for subsets of molecules having different number of true patterns using three models (Table 6.2). In the case of GBM model trained on MG2 descriptors average AUC$^+$ slowly decreased with increasing the number of amide groups. Top-n score was more sensitive and substantially dropped for molecules having two amide groups (from 0.98 to 0.69). In the case of GBM model trained on AP descriptors all interpretation metrics were more sensitive to the number of true patterns. For example, for molecules having two amide patterns AUC$^+$ decreased from 0.96 to 0.77, and top-n score dropped from 0.89 to 0.58. Similar trend was observed for the GC model (Table 6.2).

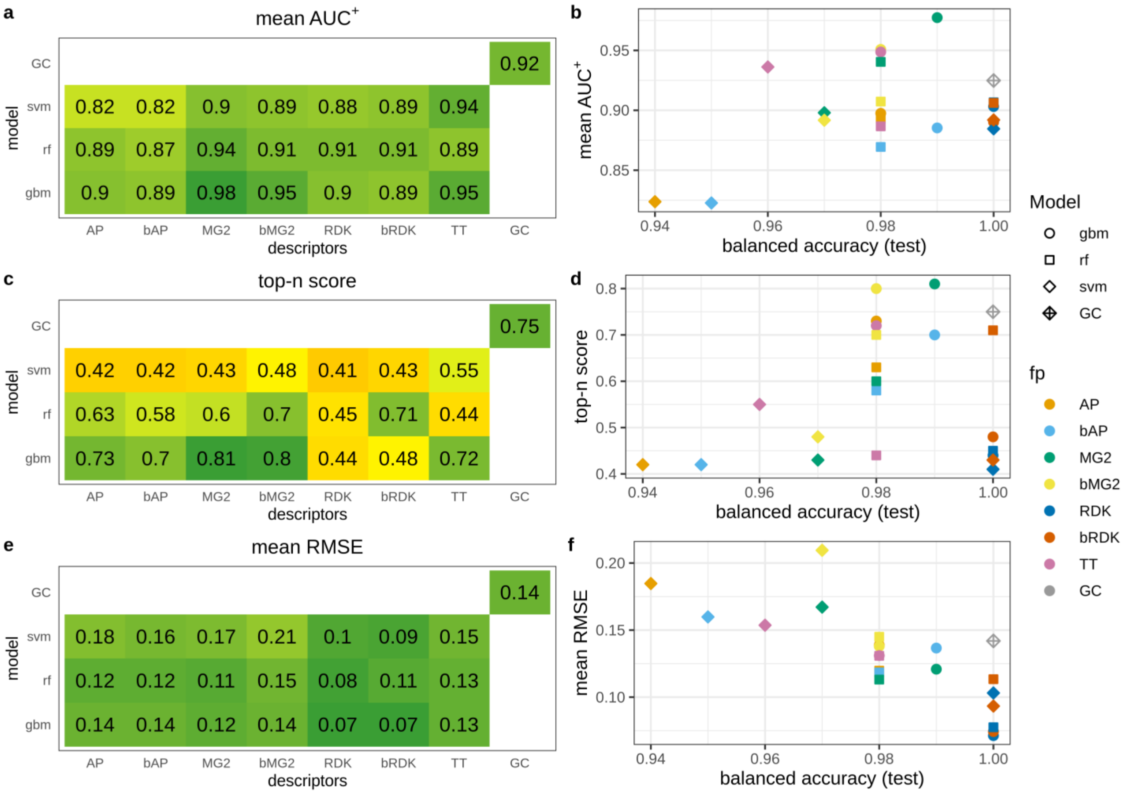


Figure 6.8 Interpretation performance of classification models trained on classification amide data set. Figure reproduced from (76)

Table 6.2 Interpretation performance of selected models calculated for subsets of molecules having different number of amide groups. Table reproduced from (76)

count of amide groups	GBM / MG2			GBM / AP			GC		
	mean AUC$^+$	top-n	mean RMSE	mean AUC$^+$	top-n	mean RMSE	mean AUC$^+$	top-n	mean RMSE
all	0.98	0.81	0.12	0.90	0.73	0.14	0.92	0.75	0.14
0	-	-	0.03	-	-	0.02	-	-	0.02
1	1	0.98	0.12	0.96	0.89	0.17	1	0.98	0.2
2	0.94	0.69	0.4	0.77	0.58	0.42	0.81	0.56	0.36
3	0.9	0.65	0.51	0.75	0.6	0.53	0.66	0.44	0.52
4	0.87	0.62	0.57	0.6	0.45	0.57	0.53	0.39	0.57
5	0.8	0.44	0.58	0.57	0.47	0.58	0.54	0.33	0.57
6	0.66	0.55	0.67	0.49	0.39	0.67	0.61	0.48	0.67

Pharmacophore data set

The pharmacophore dataset was the most difficult task and models achieved moderate balanced accuracy. Therefore, it was expected that interpretation accuracy would be relatively low (Figure 6.9). For this data set the correlation between model quality and interpretation accuracy was the most pronounced, and predictive ability of conventional models mostly depended on descriptors type. Both types of AP fingerprints produced the most accurate models. All models built on APs demonstrated reasonably high ranking ability (AUC$^+$ = 0.84-0.89). This observation can be explained by the nature of the end-point – two specific atoms at a distance of 10-11A. Atom pairs were the only descriptors which could capture such a long distance (covering to 30 bonds). RF and GBM models trained on count-based Morgan fingerprints and the GC model had moderate overall raking ability (AUC$^+$ = 0.7=0.79). Despite moderate predictive ability (balanced accuracy $_{test}$ ≥ 0.71), for large portion of models AUC$^+$ was close to 0.5. This raises a warning that high predictivity doesn’t guarantee correct interpretation.

AUC metric may not be suitable in this case due to the fact that each molecule of the active class has only two true pharmacophore centers being ranked against all remaining atoms. This could result in inadequately high AUC values if both true atoms were ranked not exactly on top. We expect top-n to be a more reasonable and stringent metric in this case. Models with the highest AUC$^+$ had low to moderate top-n scores: 0.30-0.57 (Figure 6.9c). This means that on average they identified only 30-57% of true pharmacophore centers within top 2 scored atoms. Most of other models had even lower scores. Seeking improvement of scores we calculated the average percentage of true centers in top 3 and top 5 atoms (Table 6.3).This is a common metric frequently used to measure accuracy of prediction of true active centers, for example sites of metabolism (84). The results demonstrated that probability to find true pharmacophore center substantially increased with considering more atoms. For the best model GBM/AP top 5 score reached 77%. For the GC model the increase was only to 50% (Table 6.3).

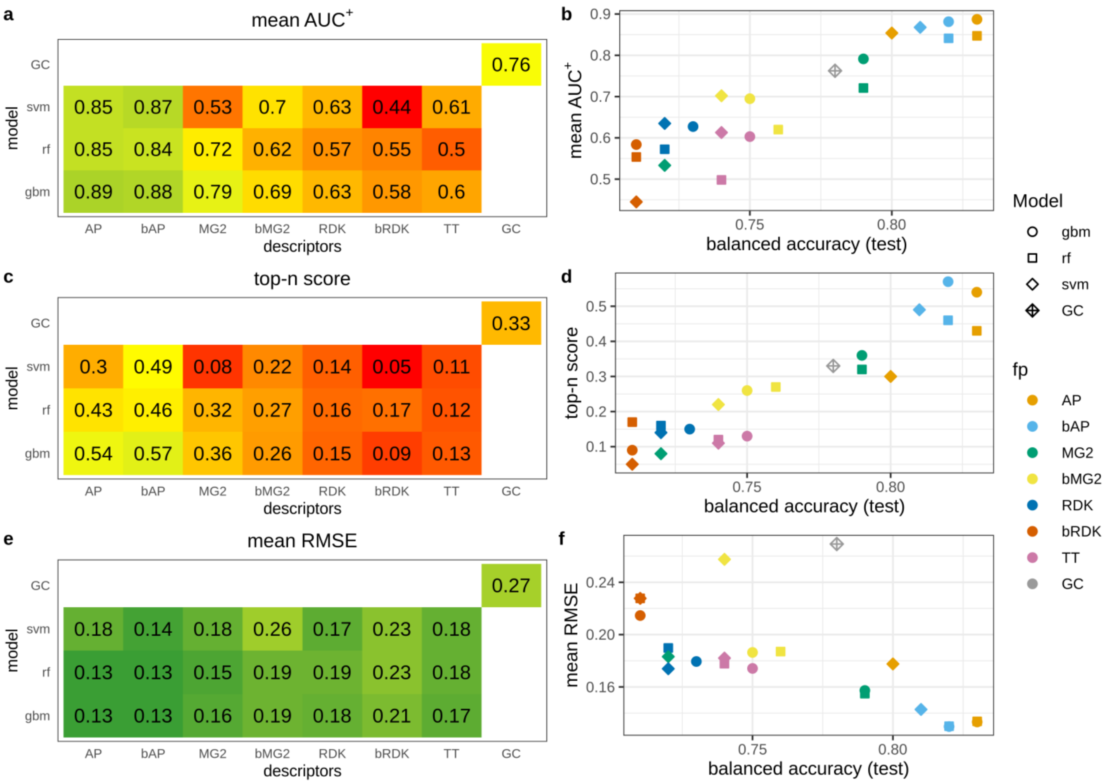


Figure 6.9. Interpretation performance of models trained on pharmacophore data set. Figure reproduced from (76)

Table 6.3. Average percentage of identified true pharmacophore centers in top scored atoms for the pharmacophore data set

Model	top-2	top-3	top-5
GBM / AP	54 %	63 %	77 %
RF / AP	43 %	54 %	67 %
SVM / AP	30 %	41 %	63 %
GC	33 %	39 %	50 %

Another way to improve results was applying fragment-based interpretation performance as opposed to atom-based. The motivation was to check whether selection of larger fragments would help to better locate true centers (though with less spatial resolution). We fragmented training set molecules as described in Model Interpretation: Calculation of Fragment Contributions. From resulting fragments, we kept only those of the size up to 7 heavy atoms and covering at most 40% of the total number of heavy atoms in a molecule. Effectively, since we did not break rings, this allowed us to estimate contributions of six-membered rings with one attached atom. To evaluate performance of fragment-based interpretation we chose top-2 metric calculated similarly to top-n metric for atoms. Top 2 scored fragments were chosen for each molecule and if both true centers were captured by these fragments the score was 1, if only one –0.5, if none –0. The scores were averaged among all molecules to get the final value. The metric top-2 for fragments was equivalent to top-n for atoms because each compound had exactly two true centers. Therefore, we compared them with each other (Figure 6.10). For models which demonstrated high performance in atom-based case, performance of fragment-based interpretation was not much better. However, models with relatively poor performance demonstrated substantial improvement. For example, GBM model trained on RDK descriptors could identify only 15% of true centers within top 2 scored atoms and 44% true centers within top 2 scored fragments.

RMSE has a little sense for classification models but we observed a clear relationship that RMSE values gradually increased with decrease of model predictive ability (Figure 6.9f).

The only outlier was the GC model. This could be explained by the fact that sigmoid activation function used on the output layer resulted in predicted probabilities tending to be either 0 or 1. Therefore, calculated contributions were also biased toward 0 or 1 that results in larger RMSE if contributions were not predicted correctly.

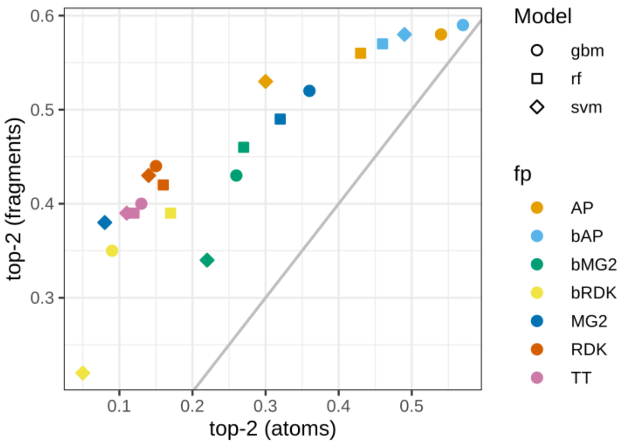


Figure 6.10. Top-2 score for atom- and fragment-based interpretation of models trained on the pharmacophore data set. Figure reproduced from (76)

N+O dataset

This data set is specifically constructed to investigate how interpretation approach and underlying model assign contributions to correlated patterns. In this case models can assign equal or similar contributions to both correlated patterns or prioritize one over the other. PLS and SVM models mainly resulted in comparable contributions for nitrogen and oxygen (Figure 6.11). RF and GBM models tended to prioritize one of them. Models trained on binary fingerprints resulted in more balanced contributions than models trained on count-based descriptors. The most striking example was GBM and RF models trained on MG2, where oxygen received much higher contributions than nitrogen.

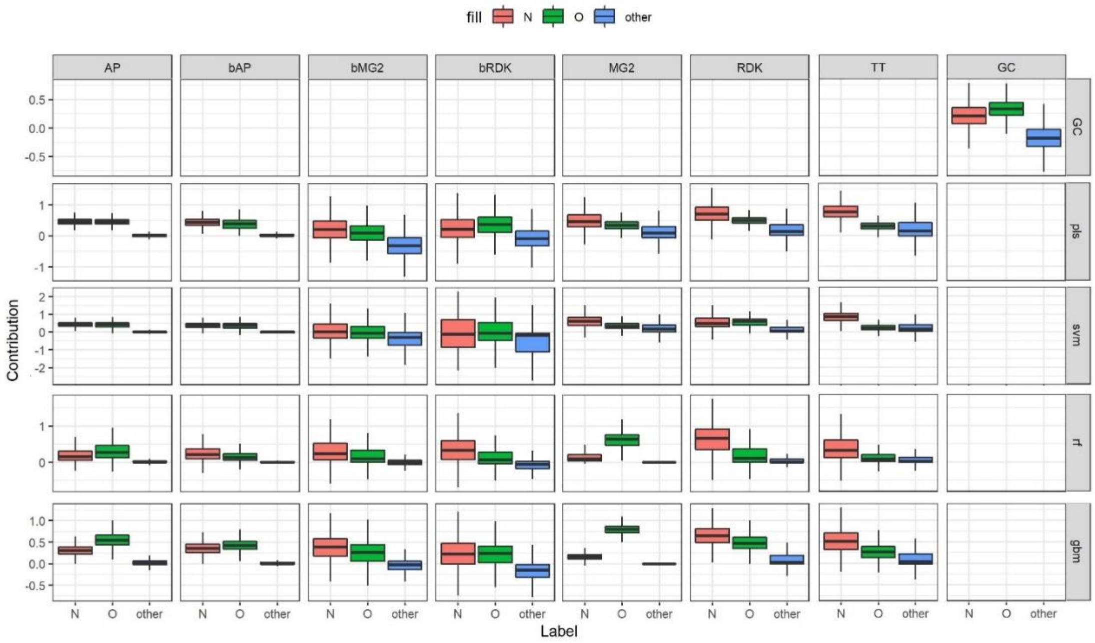


Figure 6.11. Contributions of nitrogen, oxygen and other atoms for models trained on the N+O data set. Figure reproduced from (76)

Consider our three metrics, defining both N and O atoms as true patterns first. The overall interpretation performance is high for models trained on AP and bAP, GBM and RF models trained on MG2, and GC model (Figure 6.12). However, not all of these models had high predictive performance. For example, SVM models trained on AP, bAP and MG2 had relative low R^2_{test} - 0.75-0.81. We also calculated AUC and top-n scores separately for cases where only O or N atoms were (mutually exclusively) considered true patterns. The closer the points in Figure 6.13 to the diagonal, the more balanced assigned contributions. It should be noted that overall interpretation performance does not correlate with interpretation performance calculated when only one of two patterns was considered true. For example, GBM/MG2 model showed perfect performance to retrieve O (AUC_O^+ = 1.0, top-n_O = 1.0) but low performance retrieving N (AUC_N^+ = 0.87, top-n_N = 0.0). At the same time overall interpretation performance when both patterns were considered true was very high (AUC^+ = 1.0, top-n = 0.98). This can be easily explained if one looks at contributions. Oxygen atoms received consistently higher contributions than N atoms, which have greater contributions than other atoms (Figure 6.11). Thus, O atoms were always on top and statistics was perfect. Considering both patterns as positive resulted in high performance because both were well separated from the rest.

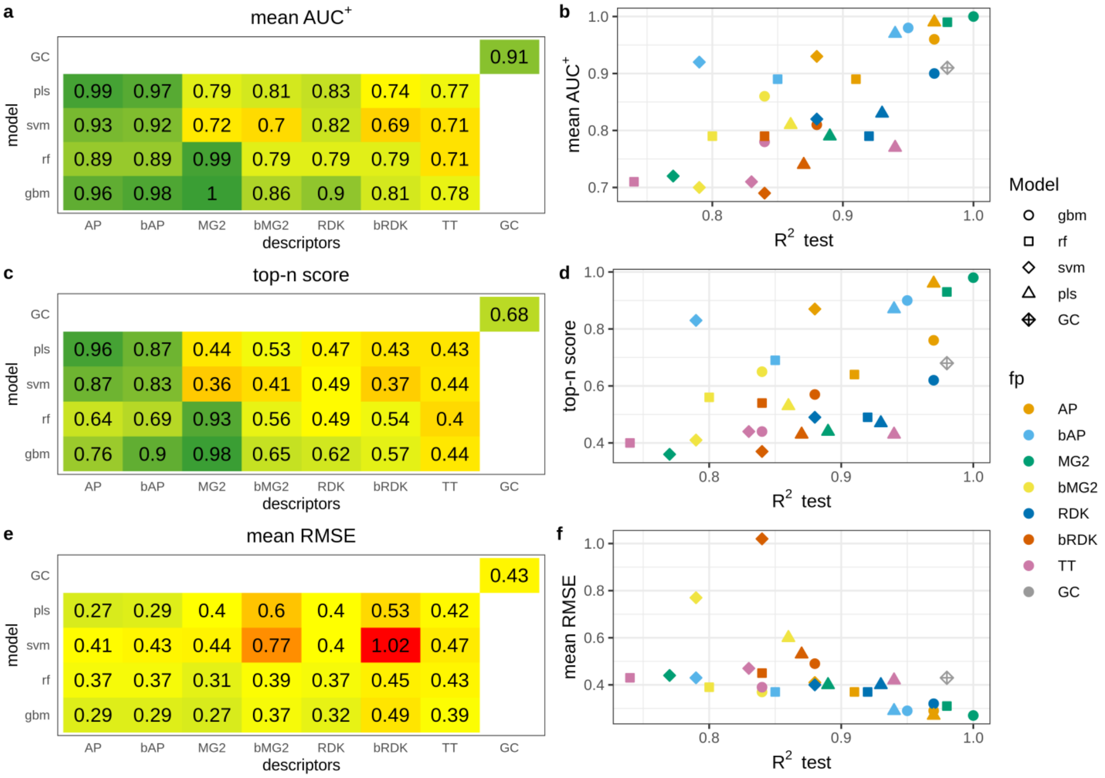


Figure 6.12. Interpretation performance of models trained on N+O data set. Both N and O atoms were considered as positive patterns. Figure reproduced from (76)

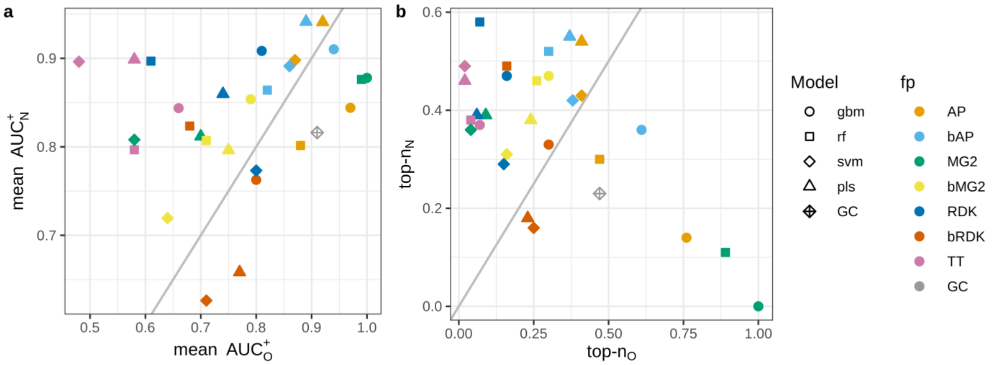


Figure 6.13. Interpretation performance for the N+O data set, where N or O atoms mutually exlusively were considered true patterns. Figure reproduced from (76)

6.2.4. Chapter summary

In this chapter we applied the benchmarking framework to test different modelling algorithms and descriptors, and a single interpretation method: UIA (it was a "pilot" study).

We pursued two aims: first, to test benchmarking datasets themselves, and second, to explore applicability of UIA. The latter has been previously applied to some particular real-case data, but no systematic evaluation has been performed so far (43). Additional purpose was to gain insights about which descriptor/model combinations are more suitable for interpretation.

The results showed the adequacy of the benchmark, since the dependencies between model quality and interpretation quality were established and consistent with previous understanding, first of all, Sheridan (74). Similarly, we claim that only highly predictive models may reach high interpretation accuracy. The UIA also proved valid, showing expected behavior.

We recommend to treat the benchmarking results as follows. High performance achieved on these data sets would support positive conclusion about the method's validity, and low performance would allow to screen out invalid methods. The latter can be claimed with higher confidence: if the method doesn't work on simple synthetic datasets it is not expected to work on more complex ones. We anticipate that this work will stimulate investigation of decision making of models, in particular neural networks, since synthetic data sets provide a more controlled environment.

6.3. Applying UIA to aquatic toxicity data sets

This section summarizes the results reported by Tinkov and colleagues (60). The article is dedicated to practical aim of model interpretation: finding relevant common toxicophore patterns. UIA as a structural method can help reveal substructures that govern biological activity/toxicity. In present application structural alerts of toxicity towards *Fathead minnow, Daphnia magna and Tetrahymena pyriformis* were retrieved.

All regression aquatic toxicity models showed satisfactory statistical performance and were comparable with existing QSAR models developed for the datasets studied (Table 6.4) (85).

Table 6.4 Predictive performance of the QSAR models developed using SIRMS

	Training set		Test set				
	5-fold CV		All compounds		Inside AD-only		
Model	R^2_{cv}	$RMSE_{cv}$	R^2_{test}	$RMSE_{test}$	Data coverage [%][a]	R^2_{AD}	$RMSE_{AD}$
			Fathead minnow				
GBM	0.65	0.86	0.59	0.95	34	0.49	0.93
RF	0.66	0.84	0.56	0.97		0.56	0.87
Consensus model	**0.68**	0.83	**0.60**	0.94		**0.54**	0.88
			Daphnia magna				
GBM	0.52	1.18	0.70	0.93	40	0.52	0.99
RF	0.50	1.21	0.70	0.93		0.53	0.98
Consensus model	**0.53**	1.17	**0.71**	0.91		**0.53**	0.97
			Tetrahymena pyriformis				
GBM	0.77	0.51	0.77	0.50	53	0.76	0.51
RF	0.75	0.52	0.76	0.52		0.73	0.55
Consensus model	**0.78**	0.50	**0.78**	0.49		**0.76**	0.52

[a] Calculated as the ratio of the number of compounds inside applicability domain (AD) and the total number of compounds of the given dataset.

The set of molecular fragments to study was formed from common functional groups, well-known toxicophores (43), and molecular fragments generated during automatic fragmentation of training set compounds.

Analysis of the results of automatic fragmentation (as opposed to common functional groups) involved only those molecular fragments that were found in at least 3 compounds of the training or test set, which allowed to increase confidence by reducing the influence of random factors (e.g. errors in experimental data or predicted toxicity values and fragment contributions).

The results are presented in Figure 6.14, Figure 6.15, Figure 6.16. It is possible to rank the contributions of fragments to various types of toxicity, which can be used to optimize target compounds. Comparing three datasets, it is possible to derive common toxicophores. Identification of such common structural alerts is important, since when a chemical enters water, it can affect various aquatic organisms.

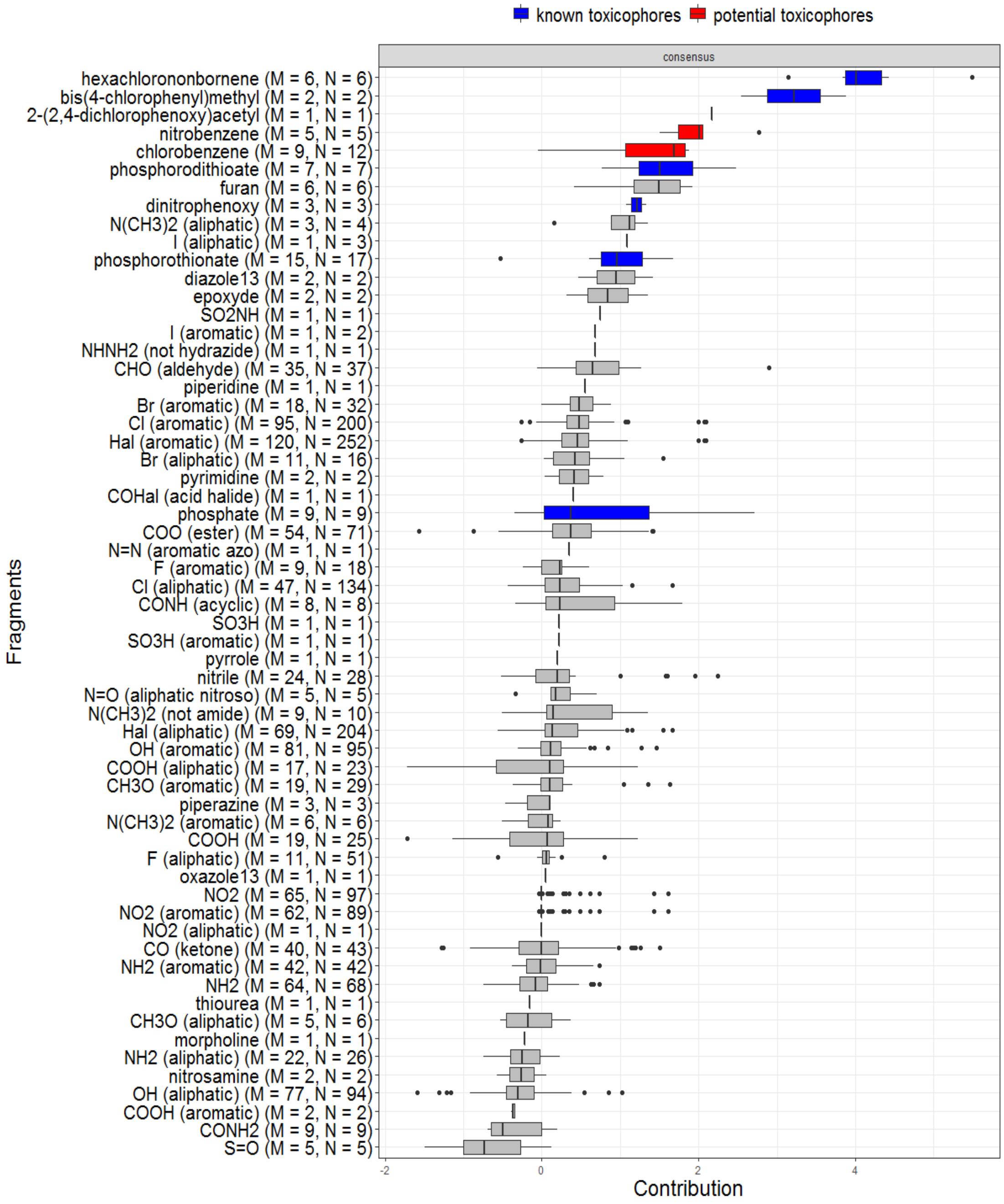


Figure 6.14 Contributions of different molecular fragments to toxicity towards Fathead minnow. Numbers in brackets: M is the number of compounds containing a fragment, and N is the number of fragments across the whole data set (some compounds have several identical fragments, and their contributions were estimated separately)

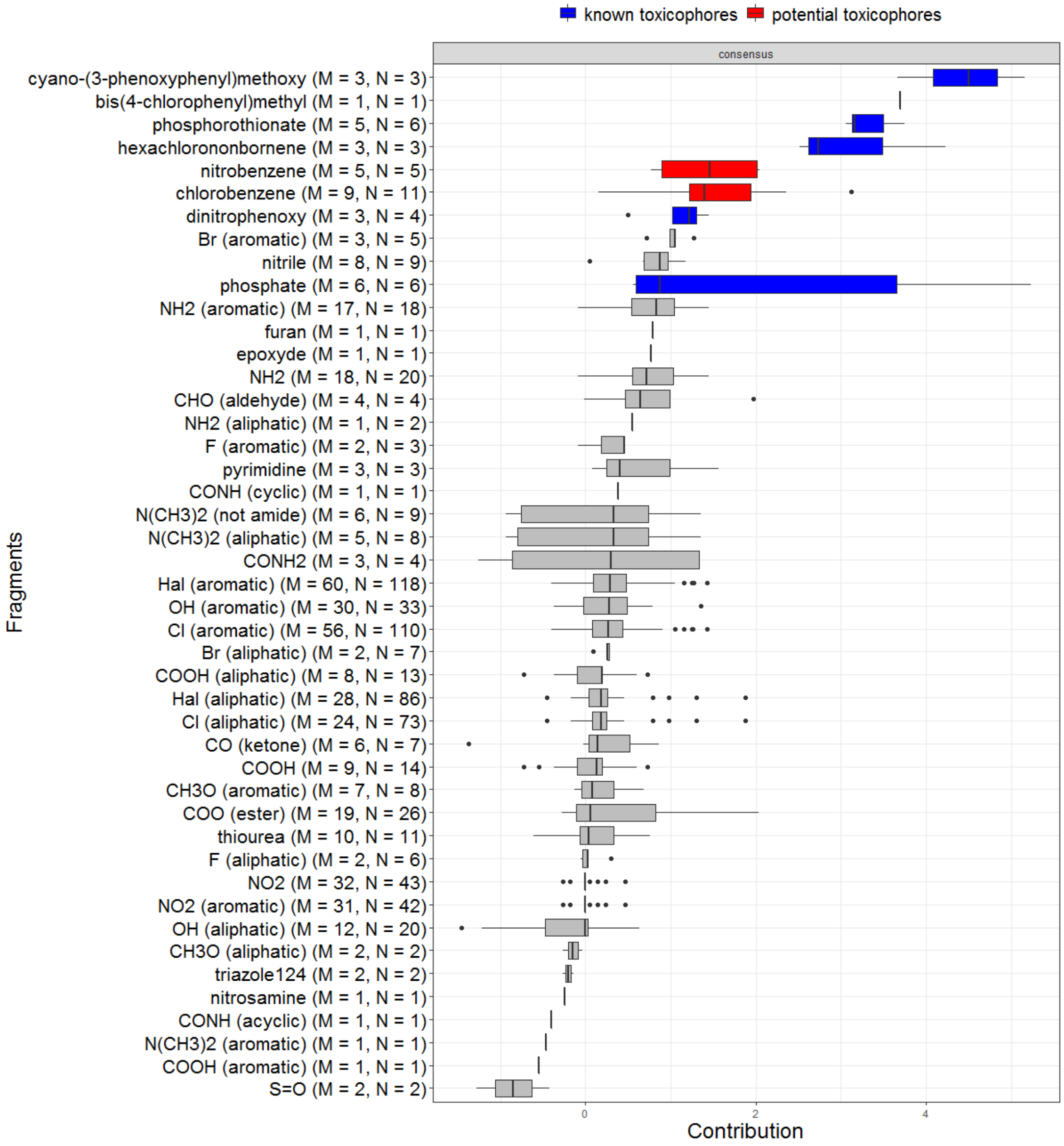


Figure 6.15 Contributions of different molecular fragments to toxicity toward Daphnia magna. Definitions of M and N were given in the Figure 6.14 caption.

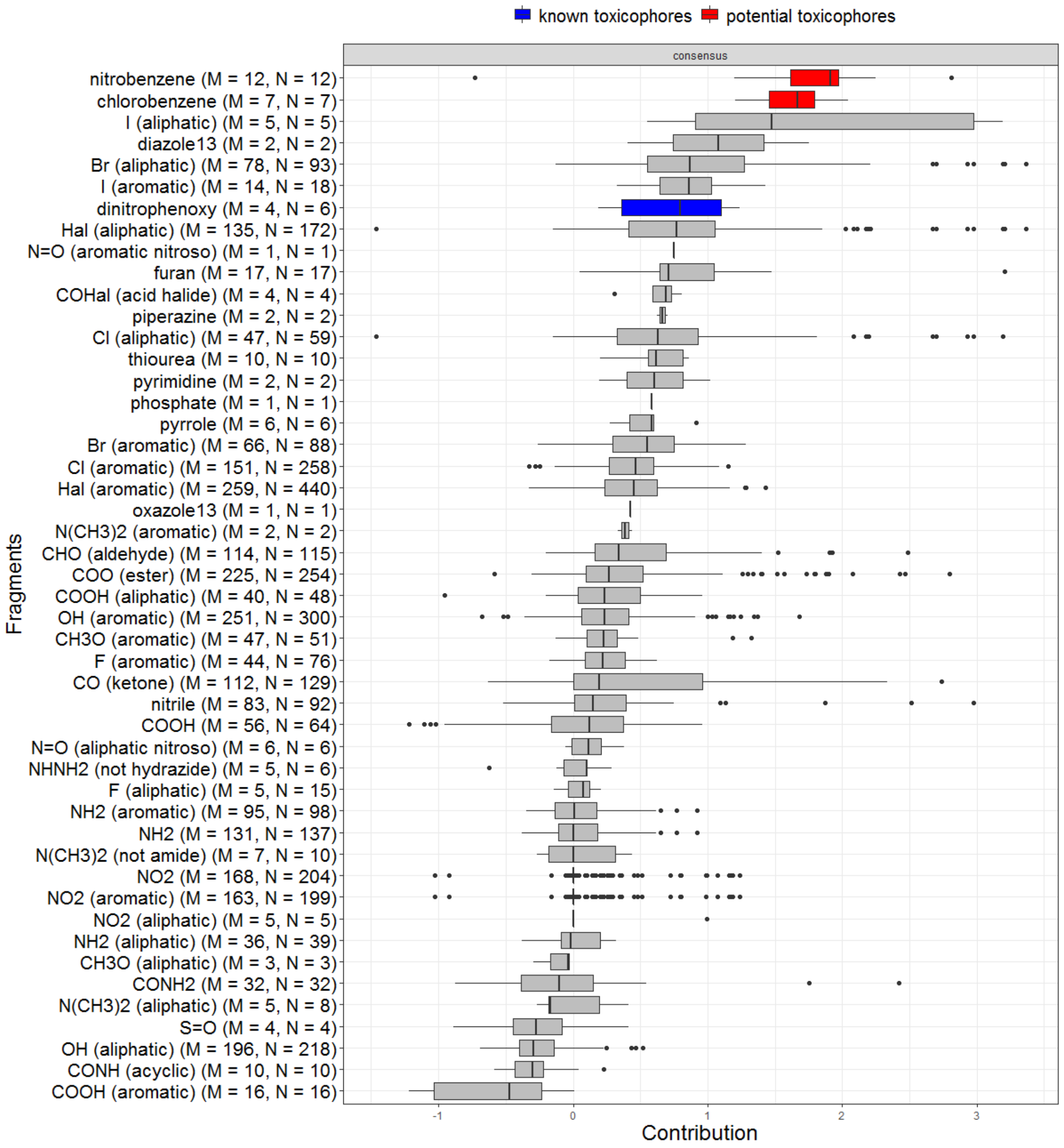
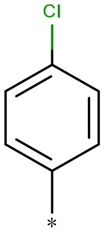
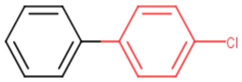
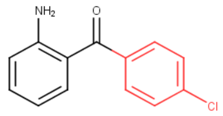
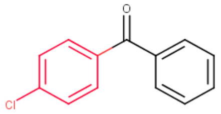


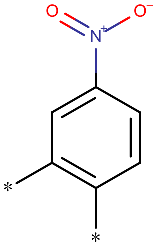
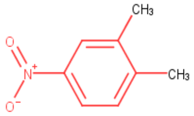
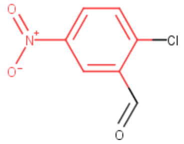
Figure 6.16 Contributions of different molecular fragments to toxicity toward *Tetrahymena pyriformis*. Definitions of M and N were given in the Figure 6.14 caption

The analysis confirmed traditional toxicophores – shown in blue – they have large contributions (86), (87), (88), (89). The main toxicophores for aquatic organisms are described in publications (63, 90),(91), (92) and included as "structural alerts" (Endpoint "Acute Aquatic Toxicity") in the expert system OCHEM (93).

In red, there are shown new fragments which were not recognized before. We identified fragments that have higher contribution to toxicity than known toxicophore phosphate and occur simultaneously in three data sets. They are 4-nitrobenzene and 4-chlorobenzene. Thereby, we defined more accurately the toxicophores “Mononitroaromatics” and “Aryl halide” used in the OCHEM. They were added to OCHEM ToxAlerts database (Alert ID: TA11520 and TA11521). Based on mechanism of toxic action described in (90, 92), it can be assumed that new alerts may participate in nucleophilic substitution reactions with biological nucleophiles, such as cysteine and/or lysine.

Table 6.5 Molecular fragments which simultaneously increase three acute aquatic toxicity endpoints

Fragment	SMARTS	Representative structures
4-chlorobenzene  	Clc1ccc([*:1])cc1	`  *Daphnia magna,*-lg(LC_{50})=5.7  *Fathead minnow,*-lg(LC_{50})=5.0 

4-nitrobenzene 	[O-][N+](=O)c1cc([*:1])c([*:1])cc1	 *Daphnia magna,*-lg(LC_{50})=4.0 *T. pyriformis,*-lg(IGC_{50})=3.6  *Fathead minnow,*-lg(LC_{50})=4.7

6.4. Development of GMM-based Extension of UIA and its retrospective validation on *T. pyriformis* data set.

This section summarizes results published by Matveieva and Polishchuk (57). The goal was to develop and validate an extension of UIA . This was aimed at overcoming the downside of *global averaging* of contributions as *pattern-mining* method. As can be seen from the previous chapter (6.3), contributions can have very broad ranges (Figure 6.14), and thus *mean value* is not very informative and even misleading. As discussed in (3) and shown in Figure 2.9, the influence of fragments on target property strongly depends on the context it appears in.

Analysis of distributions of fragment's contributions was proposed to identify groups of compounds (clusters) comprising the same fragment, where these fragments had substantially different contributions to the studied property. The workflow was implemented as follows.

6.4.2. Implementation of Extension

We employed GMM from the mclust R package (94) to develop an extension to UIA . This was implemented in the *rspi* R package (68). Examples can be found in package documentation (68). The new functionality added to *rspci* offers GMM model building and visualization steps.

The contributions of fragments were calculated for all molecules where they occurred. Distributions of contributions were analyzed for each fragment separately. The distribution can be modelled by single or multiple Gaussians (Figure 6.17). GMM utilizes the EM-algorithm for finding the optimal parameter values (mean and variance) by maximizing data log-likelihood function for a fixed number of Gaussian components. The number of components in our implementation by default is chosen using integrated completed data likelihood criterion. Variance of each component by default is variable.

Cases where the distribution of fragment contributions is represented by multiple Gaussians can be due to the different molecular context of that fragment in different molecules.

We applied SMARTSminer (95) to find patterns discriminating compounds corresponding to different Gaussians (clusters). SMARTSminer takes as its input two sets of molecules and searches for discriminative patterns (SMARTS) which appear more often in one set (“positive”) than in the other (“negative”). In the case of two clusters we submitted compounds corresponding to the cluster with lower contributions as “negative” and compounds corresponding to the cluster with higher contributions as “positive” and vice versa in order to find patterns discriminating both clusters from each other. In cases where more than two clusters were identified one cluster could be chosen as a “positive” set and the remaining ones could be combined into the “negative” set. Patterns detected were ranked according to the calculated σ-score (metric returned by SMARTSMiner). Additionally, the user can specify desired levels of positive and negative support. In *T.pyriformis* study (see below) minimum positive support was set to 0.7 (at least 70 % of molecules in the “positive” set must contain a pattern) and maximum negative support 0.3 (at most 30 % of molecules in the “negative” set may contain the same pattern). The top scored patterns output by SMARTSminer were analysed to find those that may influence or cause changes in toxicity.

An important note is that we used only training set for evaluation of interpretation performance. This should give more accurate results than test set because prediction error for a training set is smaller.

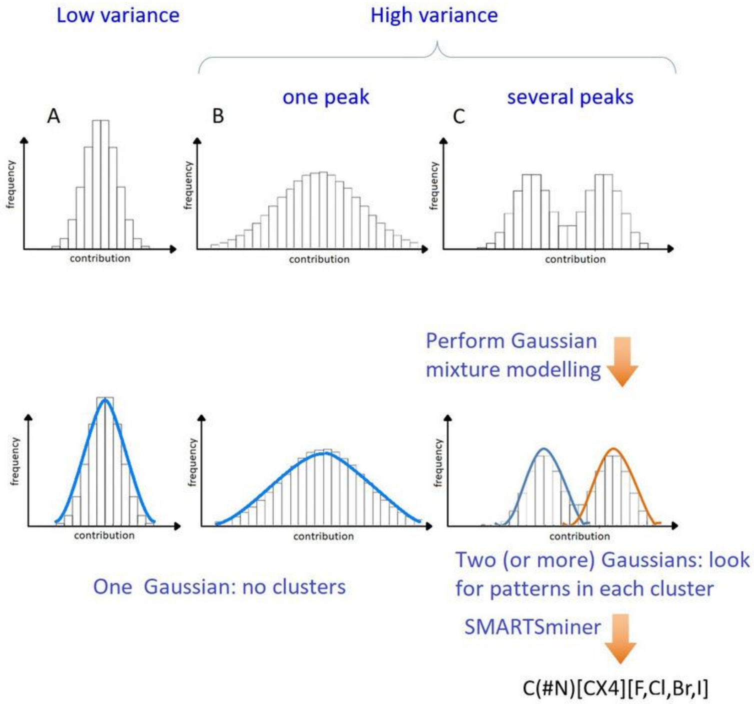

Low variance
High variance
one peak
several peaks
A
B
C
frequency
contribution
Perform Gaussian mixture modelling
One Gaussian: no clusters
Two (or more) Gaussians: look for patterns in each cluster
SMARTSminer
C(#N)[CX4][F,Cl,Br,I]

Figure 6.17 The idea of distribution analysis by GMM-based Clustering Extension. Distributions are fed to GMM algorithm, which automatically detects clusters, and if there are two or more – analysis can be performed manually or by SMARTSMiner. In the course of the analysis contexts that are important for activation of fragments are revealed and thereby structure-activity patterns can be mined.

6.4.3. Validation of the Extension on *T. pyriformis* data set

We validated the method by applying it to well-annotated *T. pyriformis* data set. If cluster analysis of distributions reveals known toxic alerts and doesn't produce obviously false alerts, we will consider our retrospective validation successful.

Predictive performance of SVM, RF and GBM models was reasonable (Table 6.6). Therefore, consensus prediction was obtained by averaging of predictions of SVM, RF and GBM models. We analyzed fragment contributions calculated from individual models and the consensus and found out that they were in close agreement. Therefore, we used the consensus model in further analysis because this helps avoid biases introduced by individual models.

Table 6.6. Predictive performance of QSAR models of toxicity on *Tetrahymena Pyriformis* estimated by 5-cold cross-validation.

Model	Q^2	RMSE
RF	0.76	0.51
SVM	0.73	0.55
GBM	0.77	0.50
PLS	0.35	0.85
Consensus (RF, SVM, GBM)	0.75	0.52

All compounds were fragmented and fragment contributions were calculated. For many fragments GMM detected only one cluster and these were excluded from further consideration. The procedure is shown in Figure 6.18.

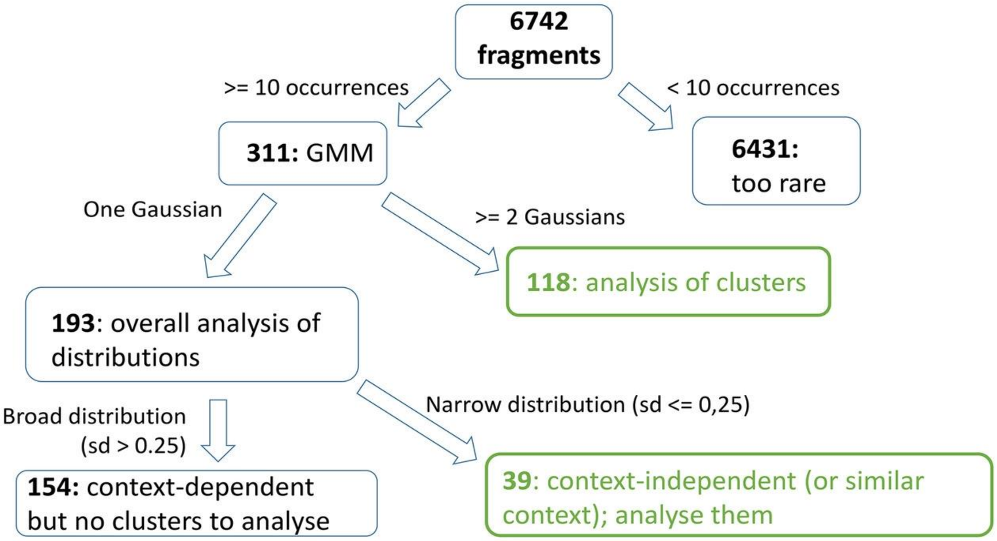

6742 fragments
>= 10 occurrences
< 10 occurrences
311: GMM
6431: too rare
One Gaussian
>= 2 Gaussians
118: analysis of clusters
193: overall analysis of distributions
Broad distribution (sd > 0.25)
Narrow distribution (sd <= 0,25)
154: context-dependent but no clusters to analyse
39: context-independent (or similar context); analyse them

Figure 6.18 Decision tree illustrating the workflow for the analysis of fragments. Green boxes contain fragments to be analyzed. The upper green box contains the fragments of main interest to this study since clusters were found in their distributions. The lower green box contains fragments having narrow distributions with no clusters (sd<=0.25, sd – standard deviation)

Reactive chemical species

We will discuss a series of fragments jointly, because they represent a common cause of aquatic toxicity: chemical reactivity. Distributions of contributions were frequently modeled by two Guassians. One had small variance and low average contributions whereas the other had a high average contribution and large variance. As was detected by SMARTSMiner (and visually), the second cluster was populated with compounds in which the fragment studied was “activated” by its context, becoming reactive. These fragments were: halogens, acetyl, carbamoyl, ester moiety, shown on

Contributions for halogen atoms had a wide distribution of values. For chlorine, bromine and iodine the distributions were similar. They had a large peak and a relatively long right-sided tail (Figure 6.19A). The major part of all observations for chlorine fell into one cluster (96% coverage of data) with mean contribution of 0.47. The long and small right-sided tail formed the second cluster (4%) with mean contribution of 1.16. One of the top-scored patterns was A[CD3H0](CCl)=[OX1-0]. It matches α-chloroketones, esters or amides present in the cluster. In the first cluster among the highest scored patterns we found SMARTS matching aromatic compounds. Regarding bromine 13% of observations fell into the second cluster (the right tail of the distribution) with mean contribution of 1.68. The highest scored SMARTS patterns match α-bromoketones and esters similarly to chlorine, but also bromoalkenes. There was a small number of compounds containing iodine atoms and only few compounds belonged to second cluster with high contribution values (mean contribution is 2.72); they were α-iodoketones and esters.

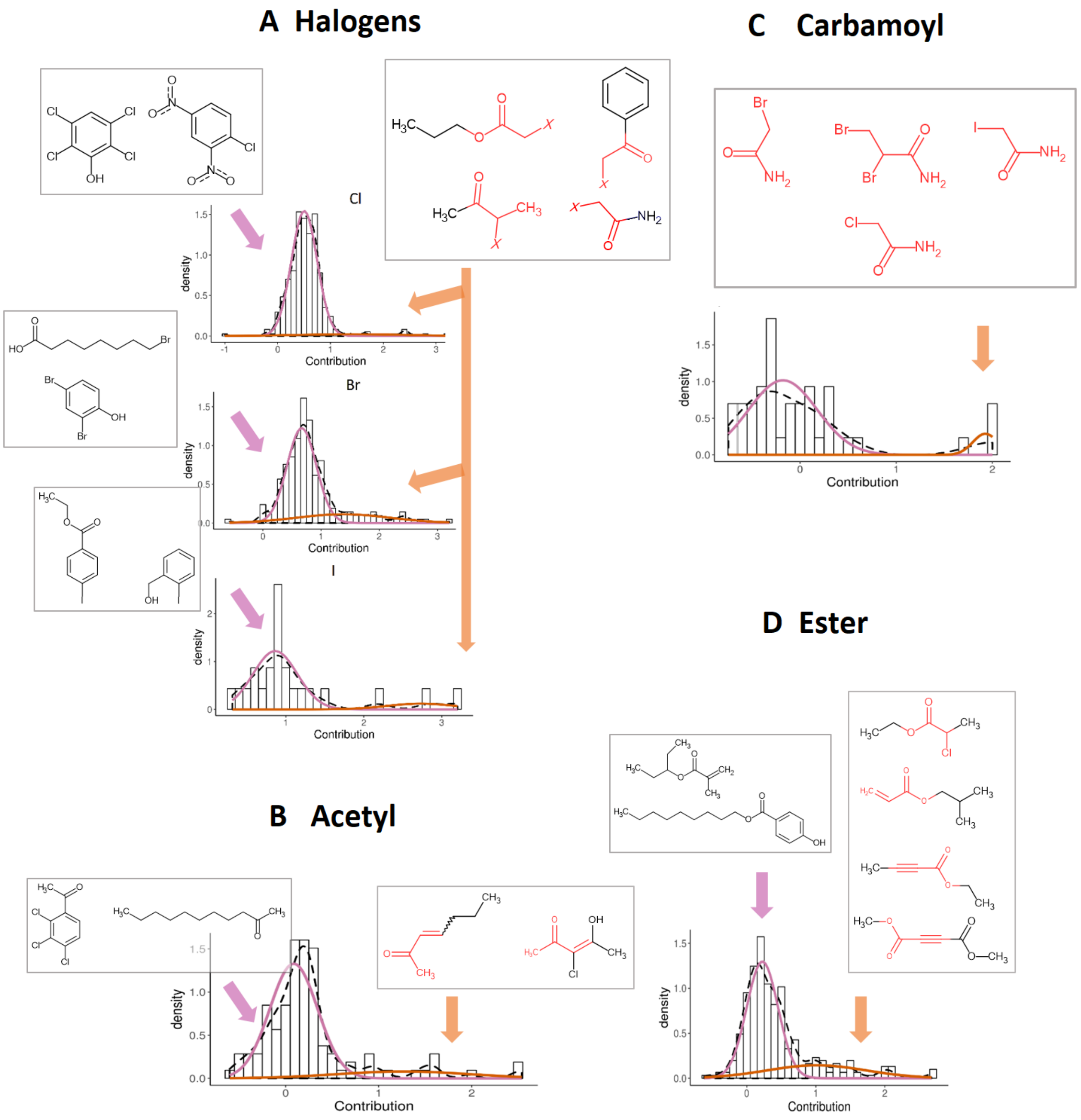


Figure 6.19. Distributions of contributions of reactive species: halogens, acetyl, carbamoyl, and ester group. Reactive patterns (combination of a fragment and its context) are shown in red. Arrows are color-coded according to Gaussians (clusters).

The right side tail of acetyl (methylcarbonyl) distribution covered by the second Gaussian contained fragments with higher contributions (Figure 6.19). Two patterns: C[CD3H0]([CD1H3])=[OX1-0] and C([CD1H3])[CX3]=[C,O] were found to be the most discriminative in the second cluster. However, the former matches acetyl itself connected to aliphatic carbon which appears to be not a toxicophore. The latter is also non-informative to our knowledge. Visual inspection of compounds from the second cluster revealed that the carbonyl group of the fragment studied is conjugated with a double bond in the corresponding compounds which can be potential Michael acceptors. The corresponding pattern [CD3H0]([CX3]=[CX3])=[OX1-0] was not found by the SMARTSminer.

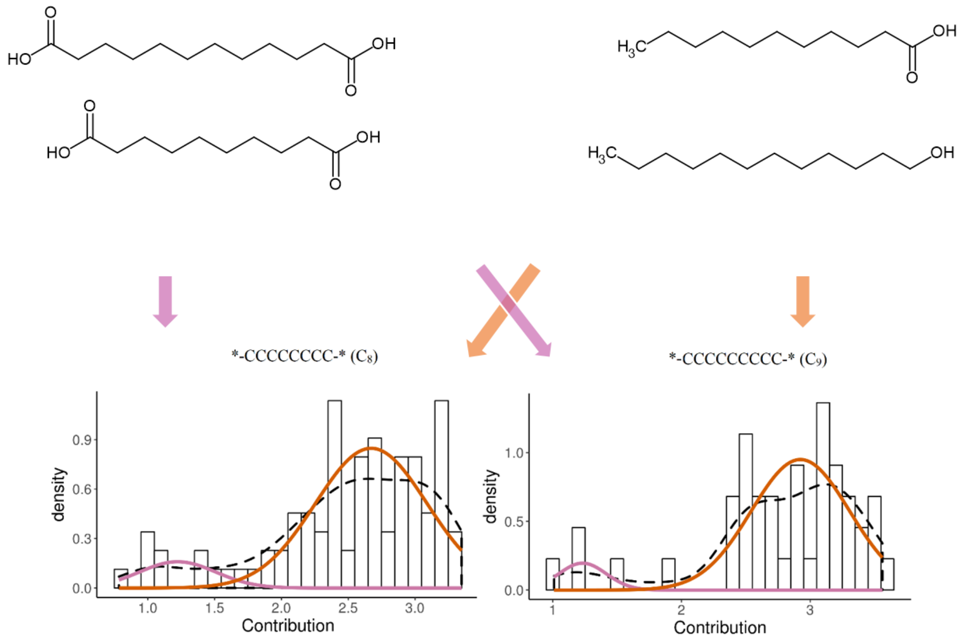
The cluster with higher contributions of ester group (Figure 6.19) corresponds to esters of α,β-unsaturated acids (mainly acrylic and 2-butynoic acid) and α-halogen carboxylic acids that was found by visual inspection. These compounds can participate in Michael addition or nucleophilic substitution reactions. SMARTSminer didn't retrieve them, because each of these two patterns has positive support (about 50%) which is lower than the chosen threshold (70%) and because the algorithm implemented in SMARTSminer doesn't use generalized bond patterns, e.g. "double or triple bond". These findings are "complementary" to the finding for halogens and acetyl, as the same reactive species were discovered.

No patterns were found by SMARTSminer for carbamoyl group. The contributions of carbamoyl groups were high only in four cases forming the second cluster, Figure 6.19. Those compounds feature halogens in α-position to carbamoyl moiety and can participate in nucleophilic substitution reactions as electrophiles. Thus, this is the "complementary" case to halogens discussed above, as the same reactive species were discovered.

Saturated linear C_8 and C_9 alkylene moieties

Two clusters with large difference between average contribution values were detected by GMM (6.20). According to SMARTS patterns found, compounds containing aliphatic carboxylic group appeared predominantly in the first cluster. Whereas C([C,O][C,O][CD1H3])[CD2H2][CD2H2][CD2H2][CD2H2][CD2H2][CD2H2][C,O] pattern encoding long linear alkyl chain was found discriminative for the second cluster. However, visual inspection showed that dicarboxylic acids were mainly present in the first cluster while the second one is more populated with monocarboxylic acids. This couldn't be captured by SMARTSminer because it cannot detect disjoint patterns and the length of alkyl chain between two carboxylic groups is variable.

Monocarboxylic acids have higher lipophilicity relatively to dicarboxylic ones and since lipophilicity is recognized as one of the major factors of environmental toxicity (96) this can explain substantially different contributions of alkylene chains in those compounds.



6.20. Distributions of contributions of C_8 and C_9 linear alkylene groups.

Hydroxyl and carboxyl groups

There was only one cluster found by GMM in both cases. Therefore, SMARTSminer could not be applied. However, the left shoulder on the distribution of carboxyl group was observed that suggested the hidden mixed distributions of hydroxyl groups in different chemical contexts. We checked whether these observations are related to context-dependence or these are artifacts. We used for fragmentation SMARTS patterns which explicitly match aliphatic and aromatic carboxyl and hydroxyl groups, Figure 6.21. Distributions of contributions of both these groups in the case of aliphatic and aromatic derivatives were significantly different according to the Kolmogorov-Smirnov two-sided test. Aliphatic hydroxyl groups (e.g. in aliphatic alcohols) have lower contributions to the toxicity in comparison to aromatic OH groups (in phenols). On smoothed densities of aromatic and aliphatic hydroxyl and carboxyl group contributions are shown in orange and pink. A carboxylic group showed lower toxicity in aromatic compounds than in aliphatic ones. This example demonstrates that GMM models cannot always separate contributions of fragments when distributions are substantially intersected. Therefore, visual inspection of contribution distributions would be required to detect such cases.

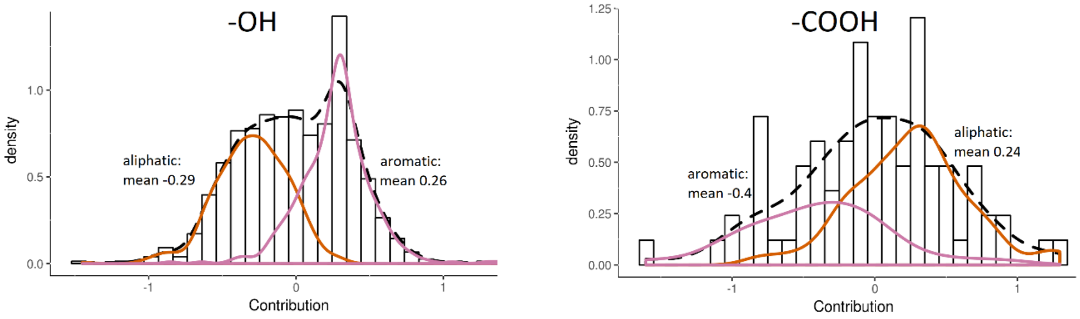


Figure 6.21 Distributions of contributions of carboxyl (right) and hydroxyl (left) groups with smoothed densities (black dashed line) and subpopulations of fragments in aliphatic and aromatic context matched explicitly (solid colored lines).

Physico-chemical interpretation of fragment contributions

Since we used descriptors encoding different physico-chemical properties we could estimate contribution of different physico-chemical terms to the studied toxicity, Figure 6.22. Polarizability of halogen atoms and carbamoyl groups having high contributions (the second clusters in all cases) had the largest impact on their toxicity. This can be due to reactivity of the detected patterns. The major contribution factor for high toxicity of alkylene chains was lipophilicity that is also supported by experimental findings (96). Thus, physico-chemical interpretation can provide more detailed knowledge about fragment contributions and help shed light on mechanisms of action as well.

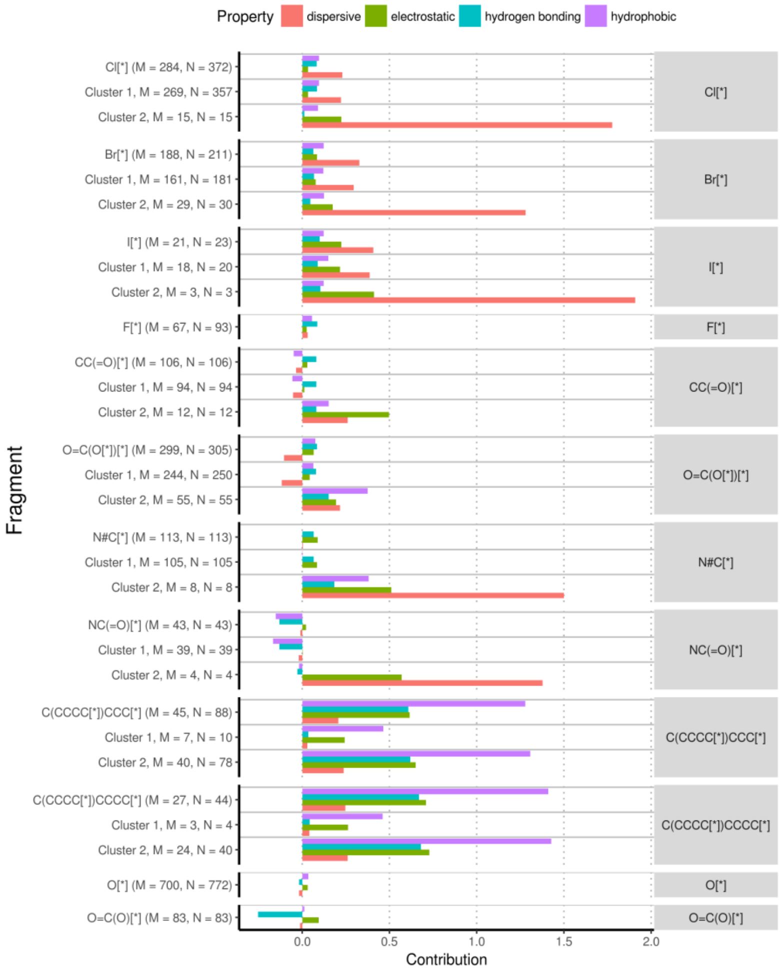


Figure 6.22. Median physico-chemical contributions of fragments to their toxicity on Tetrahymena pyriformis (M denotes the number of compounds and N – the number of fragments).

Applying SMARTSminer directly to the whole dataset

We applied SMARTSminer directly to the modeled dataset in order to find possible toxicophoric patterns and compare such a straightforward approach to ours. Patterns found match mostly aromatic and some heteroaromatic substructures which were not very informative (Figure 6.23).

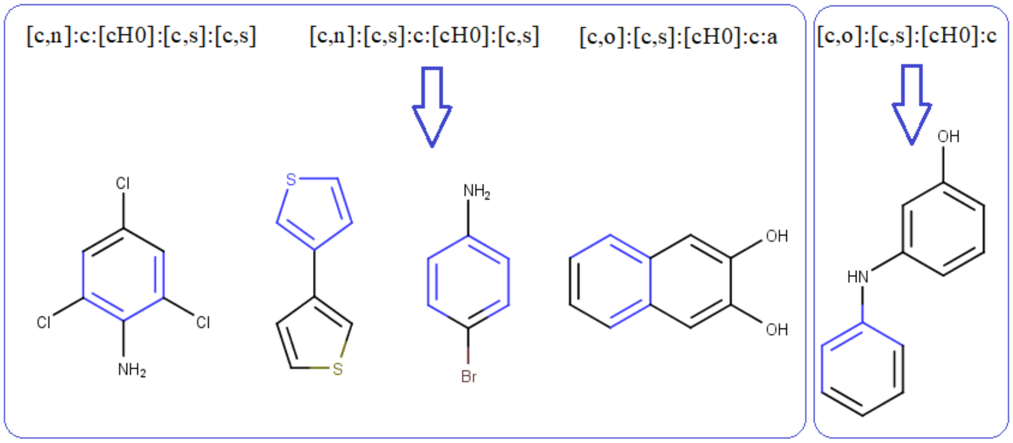


Figure 6.23. Top-ranked discriminative patterns found by SMARTSminer and examples of matched compounds from the „positive“ set (“negative” set consisted of 500 compounds with pEC_{50} <= 2.5; and “positive” set of 406 compounds with pEC_{50} >= 5).

Patterns identified by our approach could not be found because all of them had low positive support values (<0.1). Poor performance of SMARTSminer might be explained by high structural diversity of compounds in the data set and different or mixed mechanisms of their action. Thus, our method is advantageous over direct mining of patterns from the dataset, justifying application of GMM-based clustering.

6.4.4. Chapter summary

We developed an extension to UIA based on GMM and implemented it in R (*rspci* package). We demonstrated that this new functionality in combination with SMARTSminer allowed for the detection of the influence of different molecular contexts on the fragments having high contributions to the studied property. Patterns indicating different mechanisms of action were identified. In general, the results obtained were consistent and corresponded to expert knowledge about environment toxicophores. This confirmed the validity of the workflow developed. However, it has some limitations. If contributions of fragments in different contexts were numerically close, GMM could not separate them into clusters to perform further analysis, e.g. as observed for aliphatic and aromatic hydroxyl and carboxyl groups.

Overall, SMARTSminer helped automate the search for the molecular context of fragments. However, due to SMARTS inherent limitations, SMARTSminer could not retrieve results in some cases and manual inspection was required to retrieve reasonable patterns.

6.5. Identification of fragments relevant for anticancer activity of small molecules

Retrieving known structural patterns important for toxicity in present work confirmed interpretation to be practically useful. However, finding new features and applying this in discovery and optimization of potential drug candidates would be more lucrative.

Our interest in anticancer drug design motivated us to apply computational tools for mining structural patterns from structure-activity data. We employed our *in-house* database, which is as large as >4000 small molecules tested in MTT-assay, so manual SAR analysis would be tedious. Therefore, we built QSAR models and applied GMM clustering methodology to them. We used six different endpoints of cytoxicity against cancer cell lines. High accuracy of QSAR models is an important premise for success of mining of relevant patterns.

We focused on 3 pairs of cancer cell lines: HCT116 and HCT116-p53-/-; K562 and K562-TAX (taxol resistant); and CCRF-CEM and CEM-DNR (daunorubicin resistant). The choice is motivated by the possibility to find patterns, that can overcome difficult-to-treat drug resistance or genetic variants.

The application of GMM-based clustering allowed for automated pattern mining. Upon analysis, in a number of cases we rediscovered known scaffolds and mechanisms of action; however, the majority of cases shown below represents novel patterns. For these patterns we performed literature-based analysis of potential mechanisms of action and docking to different protein targets. Based on results, we can speculate about exact molecular targets related to patterns found. This research is an initial step and experimental studies are yet to be performed.

The study workflow was similar to 6.4.3. The scheme of analysis is analogous to Figure 6.18 . The difference mainly lies in applying more filters to reduce the final number of fragments. The aim was to focus on fragments with the most stable high influence on given property and strong statistical support (number of occurrences, cluster population).

We did not use SMARTSMiner, but visual inspection, thereby making our search more flexible to include disjoint whole-molecule features (difficult-to-find by SMARTSMiner).

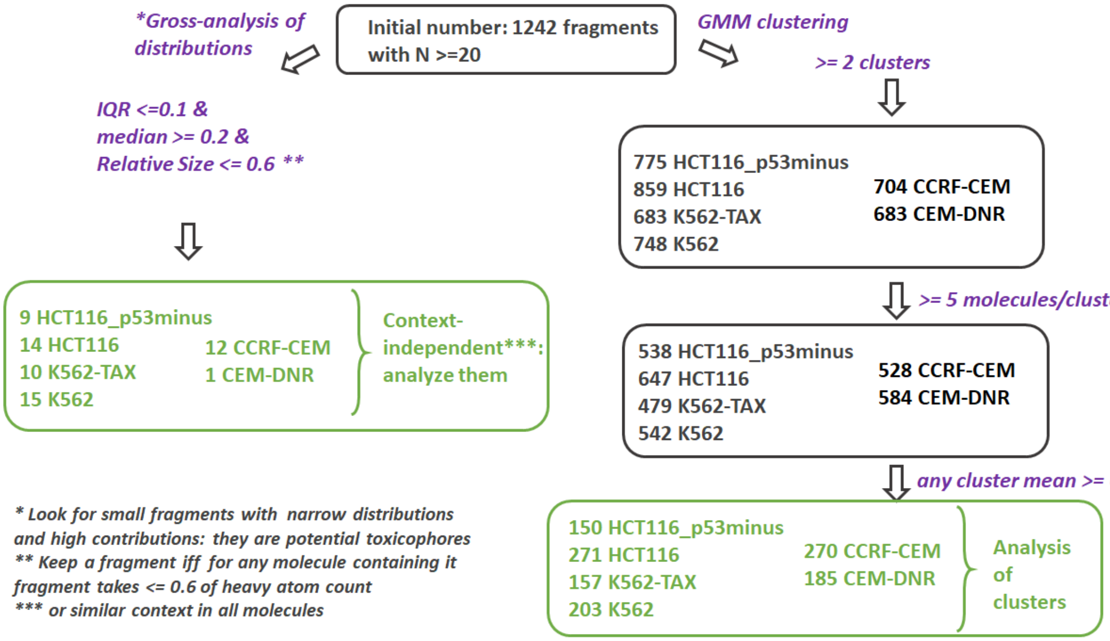


Figure 6.24 Decision tree illustrating the workflow for the analysis of fragments. Green boxes contain fragments to be analyzed. The lower-right green box contains the fragments of main interest: at least two clusters were found in their distributions; population of clusters was sufficient (>=5); mean contribution was high (>=0.2). The upper-left green box contains fragments having narrow distributions (IQR: interquartile range <=0.1) with high median and small size (the size matters because large fragments tend to have artificially high contribution).

6.5.2. Models' performance

We tested four types of fingerprints: AP, TT, RDK and MG2 and found their performance very similar with little advantage of MG2; so we proceeded only with models on these descriptors. Models had sufficiently high performance; predictions of all models were in strong agreement, so for final analysis we used consensus predictions for each data set.

Table 6.7 Test set predictive performance of QSAR models trained on 6 anticancer endpoints using MG2 fingerprints.

Dataset	Model	Balanced accuracy	Sensitivity	Specificity	Kappa
HCT116	gbm	0.83	0.84	0.82	0.5
HCT116	Rf	0.84	0.83	0.85	0.56
HCT116	svm	0.81	0.78	0.85	0.52
HCT116-p53-/-	svm	0.81	0.78	0.84	0.51
HCT116-p53-/-	gbm	0.82	0.83	0.82	0.5
HCT116-p53-/-	Rf	0.84	0.83	0.85	0.54
K562	gbm	0.81	0.83	0.79	0.46
K562	svm	0.8	0.79	0.8	0.46
K562	Rf	0.82	0.83	0.81	0.5
K562-TAX	svm	0.77	0.73	0.8	0.42
K562-TAX	gbm	0.79	0.8	0.77	0.42
K562-TAX	Rf	0.79	0.78	0.8	0.45
CCRF-CEM	svm	0.78	0.73	0.83	0.51
CCRF-CEM	gbm	0.8	0.79	0.81	0.52
CCRF-CEM	Rf	0.8	0.79	0.82	0.54
CEM-DNR	svm	0.75	0.71	0.79	0.36
CEM-DNR	gbm	0.77	0.77	0.76	0.36
CEM-DNR	Rf	0.78	0.77	0.78	0.3

6.5.3. Interpretation results

Purine fragment

For purine GMM-based clustering detected a specific environment, in which its contributions to cytotoxicity became high: Figure 6.25. This was found primarily for HCT116p53-/-. For this cell line purine fragment is highlighted in blue and the context – in orange. This context was *6-benzylamino group* attached in position (2) of purine. This combination of features represents known cyclin-dependent kinase (CDK) inhibitor scaffold. For the rest of cell lines compounds with this scaffold were not recognized as a distinct cluster, but they had high contributions (designated with orange circle for K562 in Figure 6.25).

Thus, GMM-based clustering rediscovered the pattern *"purine + 6-benzylamino group"*, representing known scaffold of CDK inhibitors.

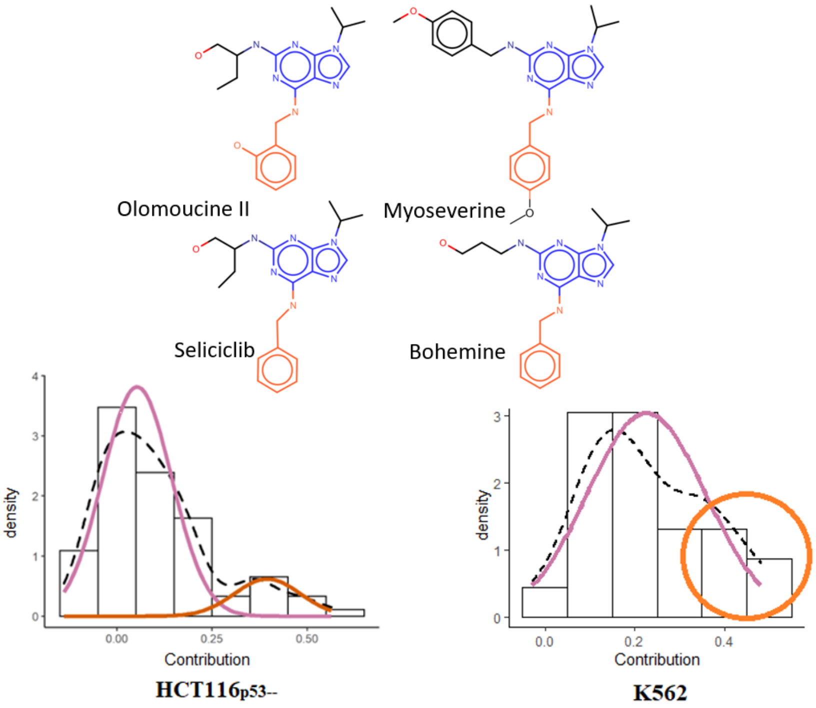


Figure 6.25. CDK inhibitor pattern revealed by GMM-based clustering for purine fragment. **Top:** examples of compounds with purine highlighted in blue, and 6-benzylamino group determining CDK inhibition - in orange. **Bottom:** visualization of clustering results for two example cell lines. For HCT116p53-/- orange cluster contains

CDK inhibitors. For K562 those compounds are located in the left most part, designated by orange circle.

Thiophene-2-thiol + pyrrolopyrimidine group

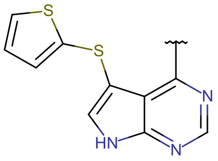


Figure 6.26. Thiophene-2-thiol + pyrrolopyrimidine fragment

Compounds bearing this scaffold (Figure 6.26) are structural analogues of nucleotides. Such agents realize their anticancer activity typically via purine metabolism disruption. Interestingly, for these compounds GMM-based clustering found, that introduction of a sugar moiety leads to activity loss. This is non-typical for antimetabolites. Thus, these molecules may act via a distinct mechanism, which was attempted to be studied by docking. We chose *ABL* and *c-KIT* kinases as possible targets, because compounds were active mostly against K562 cell line derived from chronic myeloid leukemia (CML), where these proteins are key For other cell lines sugar also led to activity loss, Figure 6.27. These compounds were published (97); the paper doesn't provide mechanism of action though. The majority compounds were non-toxic to BJ/MRC5 (97).

We performed docking of both nucleobases and nucleosides to human ABL kinase (protein data bank code 2HYY) and c-KIT kinase (protein data bank code 1T46). Additionally, we checked PIM and JAK2, because of some structural similarity to known ligands thereof. Results showed, that nucleobases had better (more negative) scores for 2HYY and 1T46 than nucleosides. The scores are shown in Table 6.8. In the case of ABL the difference is the most striking. If we compare scores with measured IC$_{50}$ we will find out that glycosides are inactive (pIC50<-4.3), while nucleobases are active in the range -4.73…-5.37 in log units: (97) Thus ABL (and less likely C-KIT) is supported as targets by docking. For PIM and JAK2 no evidence was contributed by docking.

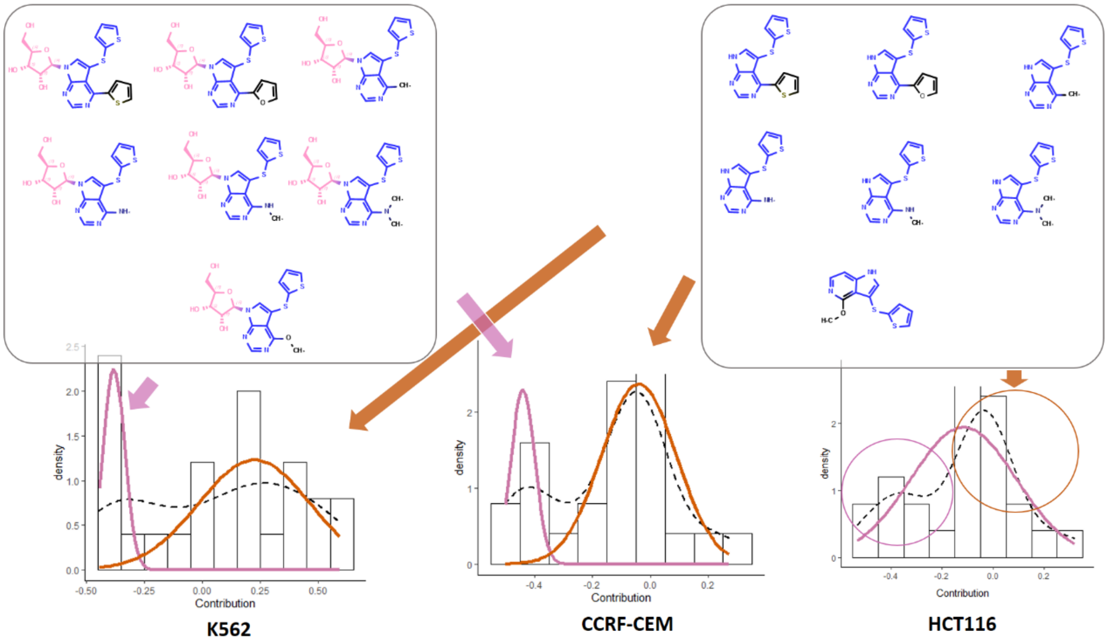
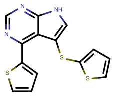
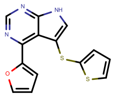
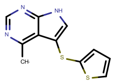
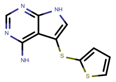
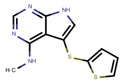
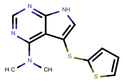
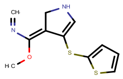


Figure 6.27 Pattern found by clustering for the fragment "thiophene-2-thiol + pyrrolopyrimidine". **Top-left:** structures are glycosides with fragment highlighted in blue and context highlighted in pink. **Top-right** – nucleobases with fragment in blue. Pink arrows point to location of glycosides Orange arrows – to location of nucleobases. **Bottom:** for K562 and CCRF-CEM there are two clusters. For HCT116 glycosides are located in the leftmost part and nucleobases – in the rightmost, designated by pink and orange circle.

Table 6.8 Mean docking scores for Thiophene-2-thiol-substituted pyrrolopyrimidine nucleobases (2nd cluster) and nucleosides (1st cluster). Tested targets are ABL and C-KIT kinase.

PDB code	**1T46**	**2HYY**
kinase	**C-KIT**	**ABL**
	Mean score	
1st cluster (nucleosides)	-6.24	-4.67
2nd cluster (nucleobases)	-6.83	-6.19
imatinib	*-13.8*	*-11.8*

Table 6.9 Docking scores and experimental activities of thiophene-2-thiol-substituted pyrrolopyrimidine nucleobases vs. nucleosides

Structure (nucleobase)	Vina docking score for ABL kinase		pIC50 for K562	
	Nucleobase	nucleoside	Nucleobase	nucleoside
	-5.7	-3.8	-5.29	<-4.3
	-5,9	-2.8	-5.37	<-4.3
	-6,9	-5.2	-4.85	<-4.3
	-6,5	-5.5	-4.73	-4.6
	-5.7	-6.0	-5.35	--
	-5.4	-4.0	-5.41	--
	-5.8	-5.4	-4.30	<-4.3

Quaternary amine-containing moiety

The fragment is shown in Figure 6.28. Clustering detected, that molecules featuring quaternary amine group at a distance of 4 single bonds away from (*para*-iodo)benzene ring were active against K562 and CCRF-CEM (Figure 6.29). These compounds are known from the literature as choline transport visualization agents (98). (Paper reports compounds with a shorter/longer linker as well.) However, neither literature search nor substructure search of REAXYS database detected any data about anticancer activity of those (or similar) molecules.

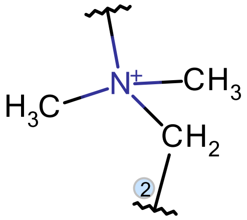


Figure 6.28. Quaternary amine-containing fragment

We investigated potential mechanism of action. One possible target is *Bcr-Abl* – key protein in pathology of CML. Another target important in CML is C-KIT kinase. We performed docking of all compounds shown in Figure 6.29 to human ABL kinase (protein data bank code 2HYY) and C-KIT kinase (protein data bank code 1T46).

The results show, that compounds with 4 bond-pattern (2nd cluster) have a little better scores than those with longer/shorter one, However, the two groups of compounds are significantly different in mean heavy atom count (HAC): 16 and 21, respectively. Scoring function used here favors a greater number of protein-ligand contacts and hence, larger molecule size (75). Therefore, we normalized scores by HAC and the difference vanished (Table 6.10. Mean docking scores for quaternary amine derivatives with and without “four-bond-distance” pattern (explained in text). Tested targets are ABL and C-KIT kinase., last 2 columns). The ligand from the pdb-complex – imatinib - has roughly the same scoring per heavy atom, Table 6.10. Thus, unlike previous example, this target is not supported by docking, though not ruled out.

Table 6.10. Mean docking scores for quaternary amine derivatives with and without “four-bond-distance” pattern (explained in text). Tested targets are ABL and C-KIT kinase.

PDB code	**1T46**	**2HYY**	**1T46**	**2HYY**
kinase	**C-KIT**	**ABL**	**C-KIT**	**ABL**
	score		**Score / HAC**	
1st cluster mean	-6.5	-6.6	-0.4	-0.41
2nd cluster mean	-7.35	-7.15	-0.35	-0.34
imatinib	*-13.8*	*-11.8*	-0.37	-0.31

We also docked our compounds to *m*- and *n*-cholinoreceptors (pdb codes: 5AFM, 6OL9), as they are choline derivatives. However, this also didn’t result any significant difference in scores between the two groups. Thus, all studied targets receive no support from docking and further studies are required.

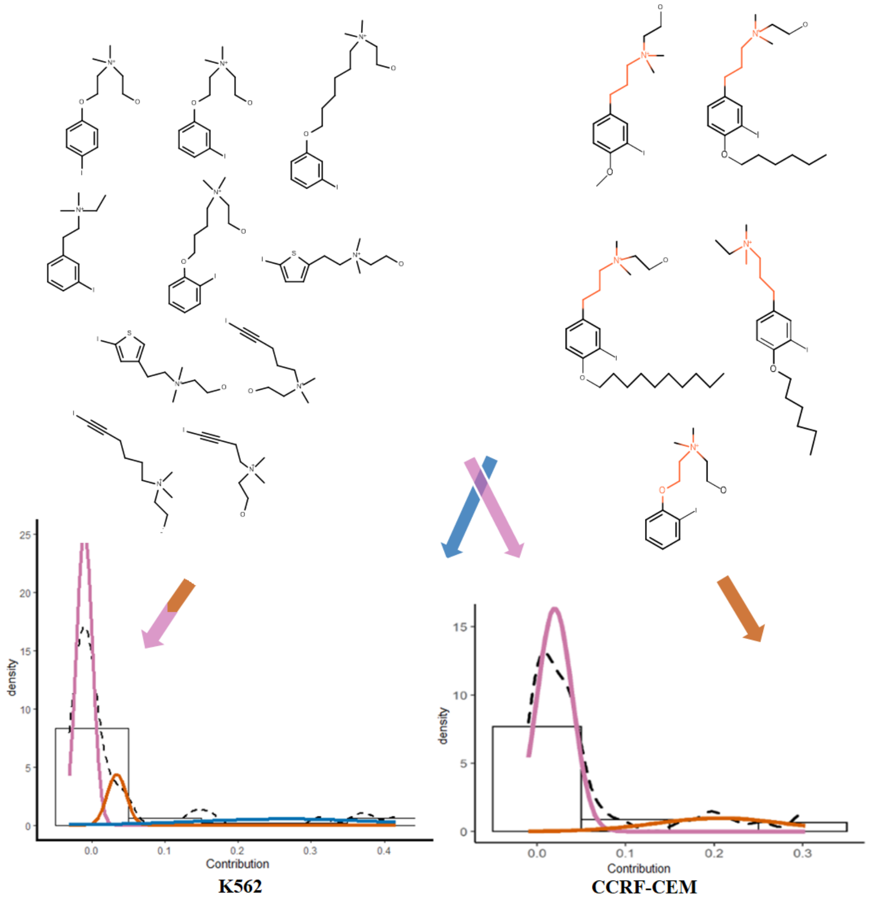


Figure 6.29 Pattern found by clustering for quaternary amine-containing moiety. The pattern highlighted in structures on the right in orange is **three single bonds-distant nitrogen and aromatic ring**. Blue and orange arrows point at clusters, where these structures belong (blue cluster for K562 and orange for CCRF-CEM). Pink and orange-pink arrows point at clusters where structures without the pattern belong.

Phenylhydrazine derivatives

The fragment shown on Figure 6.30 attracted our attention, since it had high contributions to all the endpoints, especially in two environments shown in orange Figure 6.30, Figure 6.31. Thus, highly-toxicophoric pattern can be generalized as*(o-hydroxy)phenylhydrazine + fused heteroaromatic system.*

Phenylhydrazine is a known cell poison (99); it causes heamolysis, single-strand DNA damage, and other severe harmful effects. This explains high contributions of this fragment. However, in molecular environment that we detected, it had extremely high contributions. Therefore, we attempted to explore possible mechanism of action of the fragment in that environment. We searched the REAXYS database and retrieved several similar scaffolds annotated with bioactivity, listed Table 6.11

We performed docking to dihydrofolate reductase (DHFR) and dihydroorotate dehydrogenase (DHODH). The results suggest that there's no difference between compounds, bearing the pattern and the rest (examples are given in Figure 6.31), as per scores (Table 6.12 Mean docking scores for phenyl hydrazine (explained in text).). All compounds scored high, so, although both these targets could be relevant for them, only experiment can confirm or disprove it.

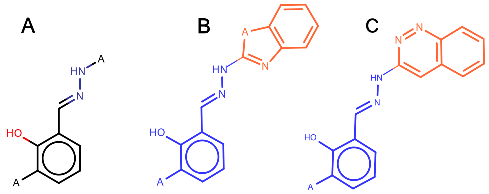


Figure 6.30. A: Phenylhydrazine fragment; B,C: two molecular contexts (in orange) "activating" the fragment.

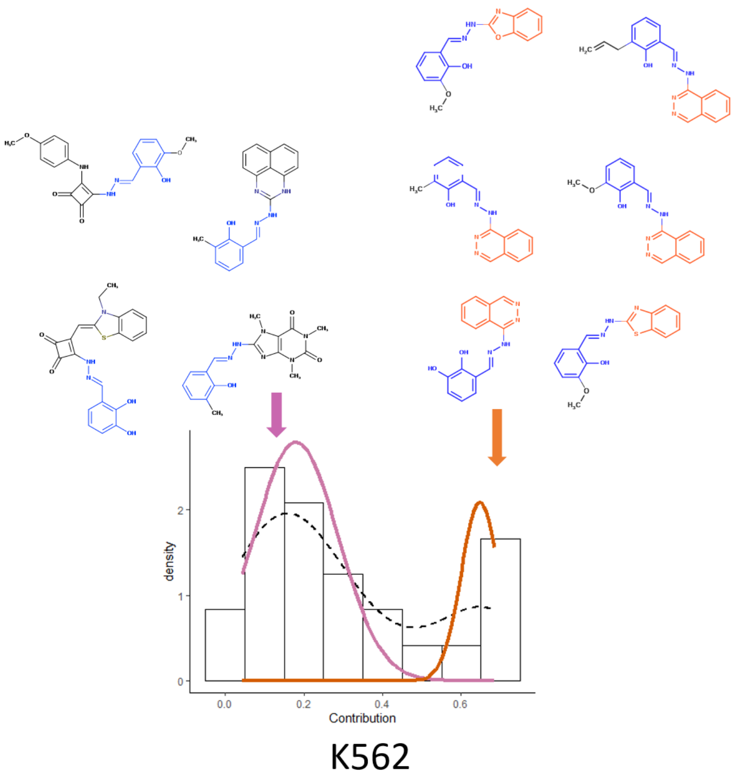
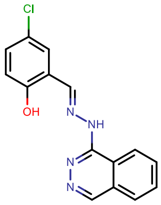
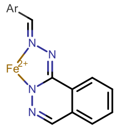


Figure 6.31 Pattern found by clustering for phenylhydrazine derivatives. Phenylhydrazine is shown in blue and the “activating” context– in orange.

Table 6.11 Possible targets for phenylhydrazine derivatives based on REAXYS database search.

Example	Target, mechanism
	Inhibition of DHODH
	SOD inhibition: oxidative stress

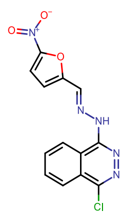
	DHFR inhibition: antimetabolite

Table 6.12 Mean docking scores for phenyl hydrazine (explained in text).

PDB code	**3f91**	**1d3g**	**3f91**	**1d3g**
kinase	**DHFR**	**DHODH**	**DHFR**	**DHODH**
	Score		**Score / HAC**	
1st cluster mean	-9.39	-10.6	-0.32	-0.35
2nd cluster mean	-9.1	-10.5	-0.42	-0.48
Native ligand from complex	-8.4	-13.2	-0.33	-0.48

6.5.4. Chapter summary

As a result of applying GMM-based clustering we have discovered a number of novel potential anticancer patterns, which we attempted to validate by docking and literature-based analysis. For quaternary amine-derivatives an anti-cancer pattern (against K562) was suggested by clustering: *aromatic ring three-single-bond distant from charged center* (highlighted in red on Figure 6.29). It has not been reported in literature before, though compounds were published as choline transport visualizing agents. For thiophene-2-thiol-substituted purine compounds (pattern shown on Figure 6.27) clustering suggested that activity against all cancer cell lines is greatly reduced by glycosylation. This observation has been published earlier, but no mechanism has been established. We performed docking to different protein targets and it strongly suggested Abl and C-KIT kinases: nucleobase compounds were scored higher than respective glycosides. For hydrazine-containing fragment we found a pattern with consistently high contributions and literature search and docking suggested several possible targets. Overall, the method developed in 6.4 showed to be helpful in search of anticancer patterns in large diverse datasets, though any results will require experimental validation.

6.6. Software

Benchmarking framework for interpretation methods of QSAR models

GitHub repository https://github.com/ci-lab-cz/ibenchmark hosts a collection of data sets, python script and examples *(Jupyter notebook)* to evaluate any method of interest. The input should be supplied as a text file containing interpretation results. The result is returned as a text file with performance metrics per-data set, and, if specified, per-molecule.

The GMM-based Extension to SPCI

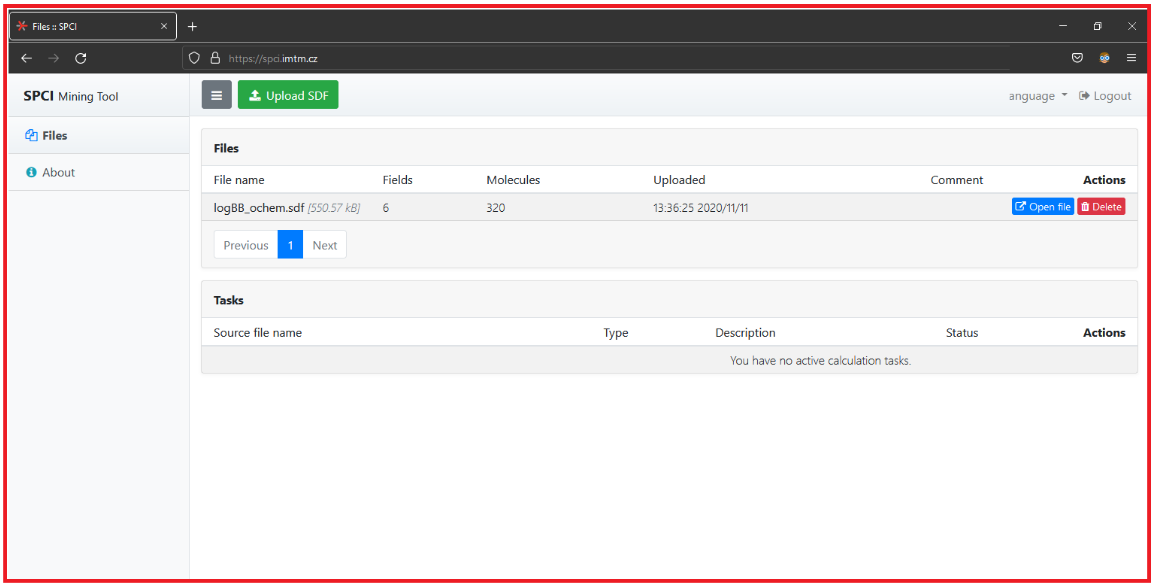
Software is implemented as an open-source R package *rspci* (https://github.com/DrrDom/rspci). The input data should contain contributions of fragments obtained by SPCI software (or any other fragment-based method, provided in the same format). Functionality of the package includes:

- Building GMM models for the data supplied (for all or selected fragments)
- Drawing the results: a histogram of the original data, kernel density estimate and Gaussians obtained. (See figures from chapters 6.4, 6.5 with fragments distributions)
- Saving plots to a specified file.

SPCI

UIA has been already implemented as a standalone application: *SPCI*. During the course of this work, we added more features to it:

- The list of available descriptors was extended and now includes binary and count-based fingerprints from RDKit (Atom-pair, Morgan2, RDK, Topological torsion)
- Building classification models for unbalanced data sets using the underdamping approach was implemented.
- We developed a web-based version of the tool but with limited functionality: https://spci.imtm.cz, Figure 6.32



Files

File name	Fields	Molecules	Uploaded	Comment	Actions
logBB_ochem.sdf *[550.57 kB]*	6	320	13:36:25 2020/11/11		Open file Delete

Tasks

Source file name	Type	Description	Status	Actions
You have no active calculation tasks.				

Figure 6.32 SPCI web application

UIA adaptation for GC models

Since SPCI software is designed to work only with descriptor-based models, we decided to add UIA functionality to some ML library for graph-based learning. We chose DeepChem because of its convenience in building graph-based models for chemistry. The feature is available for *version* >= *2.3*.

To perform interpretation on a given data set, the user needs to build a *GraphConvModel* (https://deepchem.readthedocs.io/en/latest/api_reference/models.html), and then apply ConvMolFeaturizer in “fragmentation mode” (https://deepchem.readthedocs.io/en/latest/api_reference/featurizers.html#convmolfeaturizer) to the data set. Then the difference in predictions between molecules and fragments is taken and the final results will be atomic contributions. These steps are already programmed; examples are given in the tutorial. So the user does not need coding (*Tutorial # 28. Atomic contributions for molecules* (83)).

Conclusions

1. The benchmarks developed for evaluation of interpretation approaches of QSAR models demonstrated their applicability on a wide range of models. It was clearly demonstrated that interpretation accuracy depends on predictive accuracy of models and, thus, interpretation of only predictive models makes sense. We established that some machine learning methods (e.g. gradient boosting) and descriptors (e.g. atom pairs) more often result in higher interpretation quality.

2. We demonstrated that end-to-end modeling methods, like graph convolution models, can be interpretable using the adapted universal interpretation approach. This confirms that the approach is applicable to any kind of models.

3. Extension implemented for the universal interpretation approach was able to identify different molecular contexts of fragments and explain observed variability of fragment contributions. Fragments occurring in certain environment were more chemically reactive explaining the toxicity of corresponding molecules.

4. Practical application of interpretation allowed to retrieve structure-activity relationships captured by models for a number of ecotoxicilogical end-points. We observed a good correspondence between fragments retrieved and known toxicophores, and also defined more precisely some of them.

5. Practical application of the Extension developed allowed to retrieve important structural motifs from the large diverse in-house database of anticancer compounds. Together with molecular docking this allowed to suggest possible mechanisms of cytotoxicity for particular cancer cell lines. For instance, we found out that for a series of *pyrrolopyrimidine* derivatives anticancer activity can be associated with C-KIT and ABL kinase.

Appendix

Statistical performance of models built for benchmarking data sets (6.2)

For all regression datasets baseline 1-NN models demonstrated poor results irrespective of descriptors and machine learning method used (Figure 0.1). In almost all cases R^2 was less than 0.3 for both test sets. In all cases performance of 1-NN models was significantly poorer than performance of other models. This indicates that these data sets do not have easily distinguishable patterns in chosen descriptor space and are not biased in that way.

Performance of models under study on regression data sets varied for both primary and extended test sets and depended on the combination of descriptors and machine learning method. However, there were at least several models with almost perfect predictions for each data set. GBM models trained on count-based Morgan fingerprints achieved consistently high performance on all regression tasks (R^2 = 0.95-1.0, Figure 0.1 a-h). Models trained on binary Morgan fingerprints followed by binary RDK fingerprints achieved the lowest performance across all data sets irrespective of the machine learning method used. Expectedly, binary fingerprints resulted in less predictive models than corresponding count-based fingerprints. SVM demonstrated lower accuracy on all regression data sets that could be explained by RBF kernel chosen whereas studied activities were additive and could be captured by simpler linear models. Performance of models on extended test sets was lower than on primary tests, but for highly predictive models this difference was absent or minimal. These models recognized correct patterns and challenging them by structural perturbations did not compromise their predictive performance. Therefore, we concluded that regression data sets do not have a hidden bias. Lower performance of weak models on extended test sets suggests rather model fault than a data set bias.

Moderate performance of 1-NN models achieved for classification amide data set was expected, because an amide group is essentially distinguishable by fingerprints used. Performance of GBM, RF, SVM and GC models was much higher. Balanced accuracy was greater than 0.9 for both primary and extended test sets (Figure 0.1 i-j) suggesting that in all cases models were able to capture relevant patterns. The second classification task, the pharmacophore data set, was much harder because the models trained on 2D descriptors should capture the 3D pattern. The best baseline 1-NN model trained on AP descriptors had balanced accuracy 0.66. Models under study demonstrated moderate performance, but higher than that of corresponding 1-NN models (Figure 0.1 k-l). GBM, RF and SVM models trained on count-based and binary AP descriptors had the highest balanced accuracy (> 0.8) on the primary test set. There was a slight difference between performance on primary and extended test sets. Models which had higher performance on the primary test set had higher performance on the extended test set similarly to regression models.

GC models were among the best ones across all data sets confirming this modeling approach to be competitive to conventional ones in terms of predictive ability.

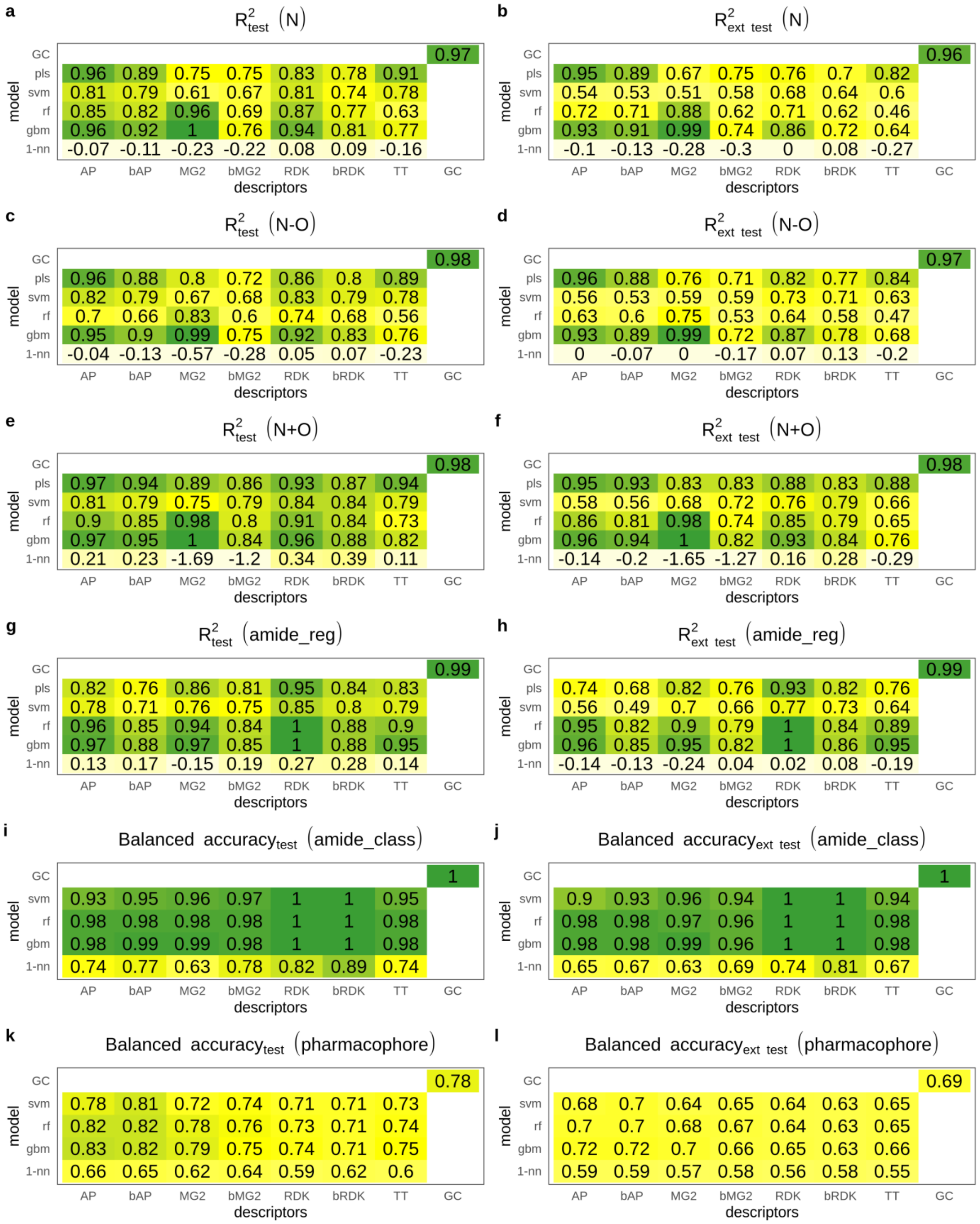


Figure 0.1 Performance of models on primary and extended test sets for regression and classification data sets.

Table 0.1 Correlations between count of patterns of interest for molecules of each regression data set and counts of the most common chemical elements.

Dataset	Pattern 1 (SMARTS)	Pattern 2 (SMARTS)					
		[C,c]	[O,o]	[S,s]	[N,n]	Cl	Br
N	[N,n]	0.09	-0.13	0.07	1	-0.02	-0.04
N-O	[N,n]	0.09	0.02	0.02	1	-0.04	-0.07
	[O,o]	0.06	1	0.07	0.02	-0.1	-0.04
N+O	[N,n]	0.23	1	0.1	1	-0.11	0.0
Amide	NC=O	0.09	0.3	0.11	0.25	0.04	0.02

Abbreviations

ADME – Absorption, Distribution, Metabolism and Excretion (of drugs, xenobiotics)

AP – Atom pairs (fingerprints)

BBB – blood-brain barrier

CDK – Cyclin-dependent kinase

CML – Chronic myeloid leukemia

GMM – Gaussian Mixture Modelling

DHFR – Dihydrofolate reductase

DHODH – Dihydroorotate dehydrogenase

HTS – High-throughput screening

MDR – Multidrug resistance proteins

MG2 – Morgan fingerprints of diameter 2 bonds

MTT - 3-(4,5-dimethylthiazol-2-yl)-2,5-diphenyltetrazolium bromide; dye used in cell viability assay (MTT-assay).

RDK – RDKit fingerprints

(Q)SAR – (quantitative) structure-activity relationship

SOD – Superoxid dismutase

TT – Topological torsions (fingerprints)

References

1. Dearden JC. (2016) The History and Development of Quantitative Structure-Activity Relationships (QSARs). *International Journal of Quantitative Structure-Property Relationships (IJQSPR)* **1:** 1-44.

2. Consonni V, Todeschini R, Puzyn T, Lesczynski J, Cronin M. (2010) MOLECULAR DESCRIPTORS. *Recent Advances in Qsar Studies: Methods and Applications* **8:** 29-102.

3. Tropsha A. (2010) Best Practices for QSAR Model Development, Validation, and Exploitation. *Molecular Informatics* **29:** 476-488.

4. Varnek A, Bajorath J. (2011) Fragment Descriptors in Structure-Property Modeling and Virtual Screening. *Chemoinformatics and Computational Chemical Biology* **672:** 213-243.

5. Baskin I, Varnek A. (2008) Building a chemical space based on fragment descriptors. *Combinatorial Chemistry & High Throughput Screening* **11:** 661-668.

6. Karelson M, Lobanov V, Katritzky A. (1996) Quantum-chemical descriptors in QSAR/QSPR studies. *Chemical Reviews* **96:** 1027-1043.

7. Bonachera F, Parent B, Barbosa F, Froloff N, Horvath D. (2006) Fuzzy tricentric pharmacophore fingerprints. 1. Topological fuzzy pharmacophore triplets and adapted molecular similarity scoring schemes. *Journal of Chemical Information and Modeling* **46:** 2457-2477.

8. Todeschini R, Consonni V. (2000) *Handbook of Molecular Descriptors.*

9. PRAAGMAN J. (1985) CLASSIFICATION AND REGRESSION TREES - BREIMAN,L, FRIEDMAN,JH, OLSHEN,RA, STONE,CJ. *European Journal of Operational Research* **19:** 144-144.

10. QUINLAN J. (1990) DECISION TREES AND DECISION-MAKING. *Ieee Transactions on Systems Man and Cybernetics* **20:** 339-346.

11. Breiman L. (2001) Random Forests. *Machine Learning* **45:** 5-32.

12. Friedman J. (2001) Greedy function approximation: A gradient boosting machine. *Annals of Statistics* **29:** 1189-1232.

13. Friedman J. (2002) Stochastic gradient boosting. *Computational Statistics & Data Analysis* **38:** 367-378.

14. GUYON I*, et al.* (1992) STRUCTURAL RISK MINIMIZATION FOR CHARACTER-RECOGNITION. *Advances in Neural Information Processing Systems 4* **4:** 471-479.

15. CORTES C, VAPNIK V. (1995) SUPPORT-VECTOR NETWORKS. *Machine Learning* **20:** 273-297.

16. Vapnik V*, et al.* (1997) Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems 9* **9:** 281-287.

17. MCCULLOCH W, PITTS W. (1990) A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY (REPRINTED FROM BULLETIN OF MATHEMATICAL BIOPHYSICS, VOL 5, PG 115-133, 1943). *Bulletin of Mathematical Biology* **52:** 99-115.

18. LeCun Y, Bengio Y, Hinton G. (2015) Deep learning. *Nature* **521:** 436-444.

19. RUMELHART D, HINTON G, WILLIAMS R. (1986) LEARNING REPRESENTATIONS BY BACK-PROPAGATING ERRORS. *Nature* **323:** 533-536.

20. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **15:** 1929-1958.

21. Baskin II, Ait AO, Halberstam NM, Palyulin VA, Zefirov NS. (2002) An approach to the interpretation of backpropagation neural network models in QSAR studies. *SAR and QSAR in Environmental Research* **13:** 35-41.

22. Laurent C, Pereyra G, Brakel P, Zhang Y, Bengio Y. (2016) BATCH NORMALIZED RECURRENT NEURAL NETWORKS. *2016 Ieee International Conference on Acoustics, Speech and Signal Processing Proceedings***:** 2657-2661.

23. Kearnes S, McCloskey K, Berndl M, Pande V, Riley P. (2016) Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design* **30:** 595-608.

24. Duvenaud D*, et al.* (2015) Convolutional Networks on Graphs for Learning Molecular Fingerprints. In.

25. Withnall M, Lindelof E, Engkvist O, Chen H. (2020) Building attention and edge message passing neural networks for bioactivity and physical-chemical property prediction. *Journal of Cheminformatics* **12**.

26. Riniker S, Landrum GA. (2013) Similarity maps - a visualization strategy for molecular fingerprints and machine-learning methods. *J Cheminform* **5:** 43.

27. Polishchuk PG, Kuz'min VE, Artemenko AG, Muratov EN. (2013) Universal Approach for Structural Interpretation of QSAR/QSPR Models. *Molecular Informatics* **32:** 843-853.

28. Sushko Y*, et al.* (2014) Prediction-driven matched molecular pairs to interpret QSARs and aid the molecular optimization process. *J Cheminform* **6:** 48.

29. Polishchuk P. (2017) Interpretation of Quantitative Structure-Activity Relationship Models: Past, Present, and Future. *J Chem Inf Model* **57:** 2618-2639.

30. (2006) Report on the Regulatory Uses and Applications in OECD Member Countries of (Quantitative) Structure-Activity Relationship [(Q)SAR] Models in the Assessment of New and Existing Chemicals. *OECD Papers* **6:** 79-157.

31. Jimenez-Luna J, Grisoni F, Schneider G. (2020) Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence* **2:** 573-584.

32. Murdoch W, Singh C, Kumbier K, Abbasi-Asl R, Yu B. (2019) Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences of the United States of America* **116:** 22071-22080.

33. Ribeiro MT, Singh S, Guestrin C. (2016) "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv:1602.04938*.

34. Strumbelj E, Kononenko I. (2014) Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* **41:** 647-665.

35. Marcou G*, et al.* (2012) Interpretability of SAR/QSAR Models of any Complexity by Atomic Contributions. *Molecular Informatics* **31:** 639-642.

36. Simonyan K, Vedaldi A, Zisserman A. (2014) Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *CoRR* **abs/1312.6034**.

37. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. (2016) Learning Deep Features for Discriminative Localization. *2016 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr)***:** 2921-2929.

38. Selvaraju R*, et al.* (2017) Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *2017 Ieee International Conference on Computer Vision (Iccv)***:** 618-626.

39. Sundararajan M, Taly A, Yan Q. (2017) Axiomatic Attribution for Deep Networks. In: *Proceedings of the 34th International Conference on Machine Learning.* Precup D, Teh YW (eds.), pp. 3319-3328.

40. Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M. (2017) Smoothgrad: Removing noise by adding noise. *Preprint at https://arxiv.org/abs/1706.03825*.

41. Bach S*, et al.* (2015) On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *Plos One* **10**.

42. Veličković P. (2017) Graph attention networks. In*.*

43. Polishchuk P*, et al.* (2016) Structural and Physico-Chemical Interpretation (SPCI) of QSAR Models and Its Comparison with Matched Molecular Pair Analysis. *Journal of Chemical Information and Modeling* **56:** 1455-1469.

44. Sushko Y*, et al.* (2014) Prediction-driven matched molecular pairs to interpret QSARs and aid the molecular optimization process. *Journal of Cheminformatics* **6:** 48.

45. Ying R*, et al.* (2019) GNNExplainer: Generating Explanations for Graph Neural Networks. *Advances in Neural Information Processing Systems 32 (Nips 2019)* **32**.

46. McCloskey K, Taly A, Monti F, Brenner M, Colwell L. (2019) Using attribution to decode binding mechanism in neural network models for chemistry. *Proceedings of the National Academy of Sciences of the United States of America* **116:** 11624-11629.

47. Wiltschko AB*, et al.* (2020) Evaluating Attribution for Graph Neural Networks. In: *Advances in Neural Information Processing Systems 33.*

48. Pope PE, Kolouri S, Rostami M. (2019) Explainability Methods for Graph Convolutional Neural Networks. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* pp. 10764-10773.

49. AOYAMA T, ICHIKAWA H. (1992) NEURAL NETWORKS AS NONLINEAR STRUCTURE ACTIVITY RELATIONSHIP ANALYZERS - USEFUL FUNCTIONS OF THE PARTIAL DERIVATIVE METHOD IN MULTILAYER NEURAL NETWORKS. *Journal of Chemical Information and Computer Sciences* **32:** 492-500.

50. Saltelli A. (2002) Sensitivity analysis for importance assessment. *Risk Analysis* **22:** 579-590.

51. Karpov P, Godin G, Tetko I. (2020) Transformer-CNN: Swiss knife for QSAR modeling and interpretation. *Journal of Cheminformatics* **12**.

52. Ashish Vaswani NS, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. (2017) Attention is All You Need. *arXiv preprint arXiv:1706.03762*.

53. Shang C*, et al.* (2018) Edge Attention-based Multi-Relational Graph Convolutional Networks. *Preprint at*

arXiv:1802.04944.

54. Tang B*, et al.* (2020) A self-attention based message passing neural network for predicting molecular lipophilicity and aqueous solubility. *Journal of Cheminformatics* **12**.

55. Zankov DV*, et al.* (2020) QSAR Modeling Based on Conformation Ensembles Using a Multi-Instance Learning Approach. *Preprint at 10.26434/chemrxiv.13456277*.

56. Polishchuk PG. SPCI: Structural and physico-chemical interpretation tool, https://github.com/DrrDom/spci. In*.*

57. Matveieva M, Cronin M, Polishchuk P. (2019) Interpretation of QSAR Models: Mining Structural Patterns Taking into Account Molecular Context. *Molecular Informatics* **38**.

58. Efron B, Hastie T, Johnstone I, Tibshirani R. (2004) Least angle regression. *Annals of Statistics* **32:** 407-451.

59. Papadatos G*, et al.* (2010) Lead Optimization Using Matched Molecular Pairs: Inclusion of Contextual Information for Enhanced Prediction of hERG Inhibition, Solubility, and Lipophilicity. *Journal of Chemical Information and Modeling* **50:** 1872-1886.

60. Tinkov O*, et al.* (2020) The Influence of Structural Patterns on Acute Aquatic Toxicity of Organic Compounds. *Molecular Informatics*.

61. Cronin M. (2017) (Q)SARs to predict environmental toxicities: current status and future needs. *Environmental Science-Processes & Impacts* **19:** 213-220.

62. Schultz T, Cronin M. (1999) Response-surface analyses for toxicity to Tetrahymena pyriformis: Reactive carbonyl-containing aliphatic chemicals. *Journal of Chemical Information and Computer Sciences* **39:** 304-309.

63. Verhaar H, Solbe J, Speksnijder J, van Leeuwen C, Hermens J. (2000) Classifying environmental pollutants: Part 3. External validation of the classification system. *Chemosphere* **40:** 875-883.

64. Martin T, Tkachenko V, Williams A. (2018) WebTEST (Web-services toxicity estimation software tool). *Abstracts of Papers of the American Chemical Society* **255**.

65. Alves VM*, et al.* (2016) Alarms about structural alerts. *Green Chemistry* **18:** 4348-4360.

66. Ruusmann V, Maran U. (2013) From data point timelines to a well curated data set, data mining of experimental data and chemical structure data from scientific articles, problems and possible solutions. *Journal of Computer-Aided Molecular Design* **27:** 583-603.

67. (2016) JChem 16.9.12. In. ChemAxon (http://www.chemaxon.com).

68. Polishchuk PG. Analysis of fragments contributions calculated by SPCI software. https://github.com/DrrDom/rspci.

69. RDKit: Open-source cheminformatics; http://www.rdkit.org. In.

70. Polishchuk PG. Simplex representation of molecular structure - a chemoinformatic tool for calculation of simplex descriptors, https://github.com/DrrDom/sirms. In.

71. Pedregosa F, *et al.* (2011) Scikit-learn: Machine Learning in Python. *JMLR* **12:** 2825–2830.

72. Ramsundar B. (2018) Molecular machine learning with DeepChem. *Abstracts of Papers of the American Chemical Society* **255**.

73. Golbraikh A, Muratov E, Fourches D, Tropsha A. (2014) Data Set Modelability by QSAR. *Journal of Chemical Information and Modeling* **54:** 1-4.

74. Sheridan R. (2019) Interpretation of QSAR Models by Coloring Atoms According to Changes in Predicted Activity: How Robust Is It? *Journal of Chemical Information and Modeling* **59:** 1324-1337.

75. Trott O, Olson A. (2010) Software News and Update AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization, and Multithreading. *Journal of Computational Chemistry* **31:** 455-461.

76. Matveieva M, Polishchuk P. (2021) Benchmarks for interpretation of QSAR models. *Journal of Cheminformatics* **13**.

77. Kutlushina A, Khakimova A, Madzhidov T, Polishchuk P. (2019) Kutlushina, A., et al. Ligand-Based Pharmacophore Modeling Using Novel 3D Pharmacophore Signatures (vol 23, pg 3094, 2018). *Molecules* **24**.

78. van der Maaten L, Hinton G. (2008) Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9:** 2579-2605.

79. Chan D, Rao R, Huang F, Canny J. (2018) t-SNE-CUDA: GPU-Accelerated t-SNE and its Applications to Modern Data. *2018 30th International Symposium on Computer Architecture and High Performance Computing (Sbac-Pad 2018)***:** 330-338.

80. Probst D, Reymond J. (2018) A probabilistic molecular fingerprint for big data settings. *Journal of Cheminformatics* **10**.

81. Polishchuk P. (2020) CReM: chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics* **12**.

82. Polishchuk P. (2020) Control of Synthetic Feasibility of Compounds Generated with CReM. *J. Chem. Inf. Model.* **60:** 6074–6080.

83. DeepChem. In., https://github.com/deepchem/deepchem.

84. Finkelmann A, Goldmann D, Schneider G, Goller A. (2018) MetScore: Site of Metabolism Prediction Beyond Cytochrome P450 Enzymes. *Chemmedchem* **13:** 2281-2289.

85. Wu X, Zhang Q, Hu J. (2016) QSAR study of the acute toxicity to fathead minnow based on a large dataset. *Sar and Qsar in Environmental Research* **27:** 147-164.

86. Islam R, Lynch J. (2012) Mechanism of action of the insecticides, lindane and fipronil, on glycine receptor chloride channels. *British Journal of Pharmacology* **165:** 2707-2720.

87. Tinkov O, Grigorev V, Razdolsky A, Grigoryeva L, Dearden J. (2020) Effect of the structural factors of organic compounds on the acute toxicity toward Daphnia magna. *Sar and Qsar in Environmental Research* **31:** 615-641.

88. Colovic M, Krstic D, Lazarevic-Pasti T, Bondzic A, Vasic V. (2013) Acetylcholinesterase Inhibitors: Pharmacology and Toxicology. *Current Neuropharmacology* **11:** 315-335.

89. Soderlund D*, et al.* (2002) Mechanisms of pyrethroid neurotoxicity: implications for cumulative risk assessment. *Toxicology* **171:** 3-59.

90. HERMENS J. (1990) ELECTROPHILES AND ACUTE TOXICITY TO FISH. *Environmental Health Perspectives* **87:** 219-225.

91. von der Ohe P*, et al.* (2005) Structural alerts - A new classification model to discriminate excess toxicity from narcotic effect levels of organic compounds in the acute daphnid assay. *Chemical Research in Toxicology* **18:** 536-555.

92. Enoch S, Ellison C, Schultz T, Cronin M. (2011) A review of the electrophilic reaction chemistry involved in covalent protein binding relevant to toxicity. *Critical Reviews in Toxicology* **41:** 783-802.

93. Sushko I*, et al.* (2011) Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information. *Journal of Computer-Aided Molecular Design* **25:** 533-554.

94. Adrian E. Raftery TBM, and Luca Scrucca. (2012) mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation. Technical Report No. 597. In. University of Washington.

95. Bietz S, Schomburg KT, Hilbig M, Rarey M. (2015) Discriminative Chemical Patterns: Automatic and Interactive Design. *J Chem Inf Model* **55:** 1535-1546.

96. Jaffe R. (1991) Fate of hydrophobic organic pollutants in the aquatic environment: a review. *Environ Pollut* **69:** 237-257.

97. Klecka M*, et al.* (2015) Synthesis and cytostatic activity of 7-arylsulfanyl-7-deazapurine bases and ribonucleosides. *Medchemcomm* **6:** 576-580.

98. Svec P*, et al.* (2020) Iodinated Choline Transport-Targeted Tracers. *Journal of Medicinal Chemistry* **63:** 15960-15978.

99. Available from: https://www.cdc.gov/niosh/npg/npgd0499.html.

List of publications covering the results presented

1. Matveieva M, Cronin M, Polishchuk P. (2019) Interpretation of QSAR Models: Mining Structural Patterns Taking into Account Molecular Context. *Molecular Informatics* **38**.
2. Matveieva M, Polishchuk P. (2021) Benchmarks for interpretation of QSAR models. *Journal of Cheminformatics* **13**.
3. Tinkov O, *et al.* (2020) The Influence of Structural Patterns on Acute Aquatic Toxicity of Organic Compounds. *Molecular Informatics*.