universität wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation /Title of the Doctoral Thesis

„Applied machine-learning in the field of next-generation pharmacophores“

verfasst von / submitted by
Stefan Michael Kohlbacher BSc MSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Doktor der Naturwissenschaften (Dr.rer.nat.)

Wien, 2022 / Vienna 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 796 610 449
Dissertationsgebiet lt. Studienblatt / field of study as it appears on the student record sheet:	Pharmazie
Betreut von / Supervisor:	Univ.-Prof. Mag. Dr. Thierry Langer

Acknowledgements

The past few years have given me the opportunity to fully experience science and its way of working. It was a rollercoaster with ups and downs, successes and setbacks, but in the end mostly ups and successes.

For that, I would like to thank Prof. Thierry Langer. He initially introduced me to and sparked my interest in chemoinformatics during my Master's Thesis. From there on I followed his guidance and the path before me. Besides highly valuable input, feedback, and constructive criticism, I highly appreciate Thierry's style of supervision. He encouraged me to be proactive in coming up with project ideas, solutions, and study designs and nudged me in the right direction if needed. Besides that Thierry repeatedly highlighted the importance of various soft skills needed to succeed in the scientific domain. Thierry, you have significantly shaped my career path, in a positive way, in the past few years. Thank you.

I would also like to thank Thomas Seidel for his continuous support, his guidance during the development of my projects, his input on solutions and ideas, and his availability as a discussion partner. Thomas, you taught me to be rigorous and with your input I could bring my scientific and programming skills to the next level. Thank you.

I would also like to thank all the other players who have shaped and impacted my development and thesis over the last few years. Unfortunately, I cannot list everyone, although I would like to highlight Sharon Bryant, Gerhard Ecker, Oliver Wieder, Marcus Wieder, and all colleagues from the Cheminformatics Research group.

Finally, a huge thank you goes to my family and friends who have supported and encouraged me through all these years. Thank you.

Table of contents

Preface

The work presented in this thesis was performed in the Chemoinformatics Research Group (University of Vienna, Department of Pharmaceutical Chemistry) under the supervision of Prof. Thierry Langer from July 2019 until August 2022.

Part I, Chapter 1 gives a brief overview of computer-assisted drug discovery, focusing on pharmacophore-based methods and machine learning in drug discovery. State-of-the-art methods and applications are introduced and discussed in Chapter 2, highlighting strengths, weak points, and possible pitfalls. Chapter 3 will conclude the first part by stating the motivation and aim of this thesis in light of previously discussed already existing methods.

Part II presents the main work of this thesis in the form of three peer-reviewed publications. The first study introduces a novel method for quantitative pharmacophore-based SAR analysis, QPhAR, and discusses the validation and applicability domain of the presented method. The second study focuses on the dissemination and best practices for QPhAR. The third and final study conducted introduces a method for automated pharmacophore modelling within the scope of QPhAR, as well as the application of QPhAR for 3D SAR analysis. Both of these methods are included in an end-to-end pharmacophore modelling workflow presented as a case study.

Part III concludes this thesis by highlighting the primary outcomes of each study and outlines potential future research.

The entire work was performed within the framework of the NeuroDeRisk (NDR) project (IMI2 grant 821528), funded by the European Union and EFPIA partners. This allowed fruitful collaborations with project members to develop methods applied in academia and the pharmaceutical industry.

I. Background

Chapter 1: Introduction

Drug discovery is a notoriously complex and lengthy process, often taking up to 15 years and costing billions of dollars to bring a novel drug to the market[1]. With decreasing success rates[2], various approaches, such as high-throughput screening[3] and phenotypic screening[4], were developed to facilitate the drug discovery process. Among plenty of *in-vitro* and *in-vivo* methods, a novel discipline has spawned with the rise of personal computers in the last century[5], computer-assisted drug design (CADD). Correspondingly, methods applied in CADD are named "*in-silico"* methods and the experiments are performed on the computer and not carried out manually in the lab. CADD implicitly promises to "solve" the drug discovery problem at hand if the interactions of a drug with an organism and its further effects could be fully simulated. While there is still a long way to go to fully simulate an organism, various algorithms have already been developed to assist experimental studies. In the following, an overview will be given regarding some CADD key concepts forming the theoretical basis of this thesis.

Pharmacophores and pharmacophore modelling

Pharmacophores were defined by C.G. Wermuth et al. as "*A pharmacophore is the ensemble of steric and electronic features that is necessary to ensure the optimal supramolecular interactions with a specific biological target structure and to trigger (or to block) its biological response"*[6]. Typically, pharmacophore models are composed of the following feature types: hydrophobic, aromatic, H-Bond donor/acceptor, and positive/negative ionisable, although most software packages allow the user to define custom pharmacophore feature types[7, 8]. Modelling protein-ligand interactions only in the form of interaction types introduces a level of abstraction to the drug discovery process. An aromatic feature, for example, could represent interactions from a phenyl ring, a thiophene ring, a pyridine ring, and many more. This allows the researcher to extract key insights from his data, focusing only on the physicochemical nature of the most critical interactions with almost complete disregard of underlying ligand structures.

Pharmacophores can be derived for ligands, proteins, or protein-ligand complexes[8–10]. All of these approaches have their pros and cons and should be seen as complementary methods rather than competitive methods. Protein-ligand complexes provide a holistic view of the protein-ligand interaction, taking into account both interaction partners to determine the actual interactions between the two. Such a pharmacophore is usually known as an interaction or structure-based pharmacophore[11, 12] and only contains pharmacophore features representing true interactions of the ligand with the protein in the given conformation. Pharmacophores derived solely from the three-dimensional arrangement and structure of residues in the target binding-site are widely known as apo-site pharmacophores. These pharmacophores usually contain a much larger amount of pharmacophore features which only represent possible energetically favourable interactions that the protein may form with a complementary ligand. The truly formed interactions then depend on the structure of an actual ligand and are usually much smaller in number than indicated by the apo-site pharmacophore. The most important type of pharmacophore for this thesis is the pharmacophore derived without taking any receptor structure information into account. It is in some way similar to the apo-site pharmacophore, however, here, the pharmacophore features are obtained from the opposing ligand side. These ligand-based pharmacophores contain much fewer pharmacophore features than apo-site pharmacophores and only represent those interactions that ligands may form

with the protein. In contrast to the apo-site pharmacophore, ligand-based pharmacophores usually provide a much more refined set of features with potential significance for high-affinity binding.

Suppose the researcher has a set of ligands available which should be analysed in order to gain some knowledge about the types of interactions which might be crucial for strong binding to the targeted protein. This process is known as “pharmacophore modelling”[8, 13], and pharmacophores represent the vehicle by which the thus obtained information is transported. By aligning the conformations of various known ligands, superimposing and clustering the pharmacophore features, and extracting common feature patterns, the researcher tries to develop a pharmacophore model that includes most, and ideally all, information from the ligands. This process is highly subjective, extremely tedious and laborious, depends strongly on the researcher's experience and expert knowledge, and is often difficult to reproduce by others without clear instructions. Nevertheless, it is a key component of the medicinal chemist’s toolbox nowadays.
The validation of the obtained pharmacophores is not less complex than the modelling process itself. Usually, a set of known active and inactive compounds is available to the researcher to validate the pharmacophore’s ability to distinguish between the two classes. However, the classification of molecules into actives and inactives poses a fundamental problem. Are molecules that are close to the activity cutoff, which would be treated similarly in the absence of the cutoff, really different from each other? Is it correct to treat one molecule as an active binder to the protein target and the other as a non-binder, despite their similarity[14]? Finally, finding a proper cutoff heavily relies on the researcher’s expertise and experience and is, just like the pharmacophore modelling process, highly subjective. An often approached workaround to this problem is to remove molecules in this grey area from the data set, artificially widening the gap. While this reduces the ambiguity of the molecule’s classification near the cutoff, it does not solve the underlying problem. Furthermore, as data points are removed, valuable information is lost in the validation process. A quantitative approach applying regression instead of classification for validating the pharmacophores could get rid of this problem entirely.
However, pharmacophores, in their original definition, are qualitative, meaning a pharmacophore feature and a possible interaction either exists or does not exist, leaving no room for intermediary outcomes. The energy of protein-ligand interactions, in contrast, is not qualitative, and some interactions might be stronger than others. This currently limits the application of pharmacophores to qualitative methods and to high-level analyses regarding interaction types. This raises another question. How can pharmacophores be compared to each other, in some cases even ranked?

Virtual screening and pharmacophore screening

Virtual screening (VS)[15] is the process of probing molecules *in-silico* against some predefined criteria. In contrast to *in-vitro* screening, which takes time and many resources, VS is relatively fast and hardly requires any resources except computational power[16, 17]. Mimicking the *in-vitro* process on a digital platform can speed up the screening process tremendously, evaluating millions of molecules within a few hours without the need to buy or synthesising them first[18]. Various methods and algorithms[19] have been developed in the past to perform VS, and can roughly be split into structure-based virtual screening methods (SBVS)[20], and ligand-based methods (LBVS)[21].
SBVS is often applied when the crystal structure of a protein has been identified and includes methods such as molecular-docking[22] and molecular dynamic simulations[23]. Both methods

evaluate the interaction capability of potential ligands with the molecular binding site of the protein and derive a corresponding quantitative metric. Much research has been put into this field in the past years since the availability of X-ray crystal structures enables an analysis of protein-ligand interactions with more confidence by the higher amount of provided information. However, due to the large complexity of these methods, they are often only used as a complementary filter for a previously obtained hit list from other screening methods[24, 25].

LBVS, on the other hand, is often applied when no crystal structure is available, the protein target is not yet known, or only a few ligands with binding information are known. Although, it can also be applied if the crystal structure has been identified. Despite missing complementary information about the binding site structure, ligand-based methods have shown promising results[21, 26, 27] and have become a popular alternative to structure-based methods. Ligand-based methods include, among others, pharmacophore-based virtual screening[8], molecular substructure search[28, 29], molecular similarity search[30], and classification models using machine-learning[31].

Due to the high level of abstraction inherent in pharmacophores, they became popular in the field of VS[15, 32], and are often applied in the early stages of the drug development process to filter large libraries of molecules. At this stage, it is essential to gather as many diverse drug candidates as possible to facilitate the following hit-to-lead optimisation phase[33, 34]. In the extreme case, a single molecule is sufficient to create a pharmacophore, which is used to obtain many other hits of different scaffolds. The ability of pharmacophores to retrieve structurally diverse molecules with the same binding motive is known as “scaffold hopping”[35] and is one of the strengths of pharmacophores in VS. Furthermore, pharmacophore-based VS screening is computationally relatively cheap, enabling a large number of molecules to be screened in a short period of time.

After the initial hit list has been obtained, the question arises, which of the hit molecules might be more relevant than others[36–38]? The qualitative nature of pharmacophores does not allow for a ranking of the obtained hit list[39]. Currently, there are two approaches to this question: using an external quantitative model or screening the same library of molecules multiple times with different pharmacophores[40]. The first approach satisfies the requirement to properly rank each hit to prioritise one over the other for further *in-vitro* and *in-vivo* tests. However, since the model is fundamentally different from the pharmacophore, its applicability domain, which is not always clearly defined, might be entirely out of scope. This introduces ambiguity to the modelling process, and it remains unknown whether other molecules from the library might have ranked differently. The second approach is also known as the “Common-hits approach”[40] and ranks molecules identified as a hit in multiple screening runs higher than molecules hit only once or just a few screening runs. This improves the confidence in high-ranked hits; however, the end result is still qualitative. A hit obtained might bind to the target of interest or not, and it is still unclear how strong the ligand will bind and what activity to expect. In contrast, a quantitative method would be able to rank hits according to their expected binding activity, yielding more detailed insights into the differences between the obtained hits.

QSAR

Quantitative structure-activity relationship (QSAR)[41–43] methods are inherently different to the qualitative approach of pharmacophores. In its simplest form, QSAR aims to identify correlations between molecular properties (structural and/or physicochemical) and their observed binding activity to a protein. The concept of QSAR was first introduced in the ’60s by Corwisch Hansch,

formerly known as the Hansch analysis[44]. The main motivation was - and still is - to identify structural motives that are relevant for strong binding of molecules to the investigated target. At first, QSAR employed simple mathematical tools such as linear regression[45] (Figure 1) to identify correlations between physicochemical properties, such as logP[46], and binding activity. Over time, as algorithms got more advanced, computational power increased, and molecular representations[47–49] got more sophisticated, the QSAR models grew more complex, identifying relationships from thousands of molecular descriptors. However, the rationale for applying QSAR is still the same; identification of key structural motives that drive molecular binding.

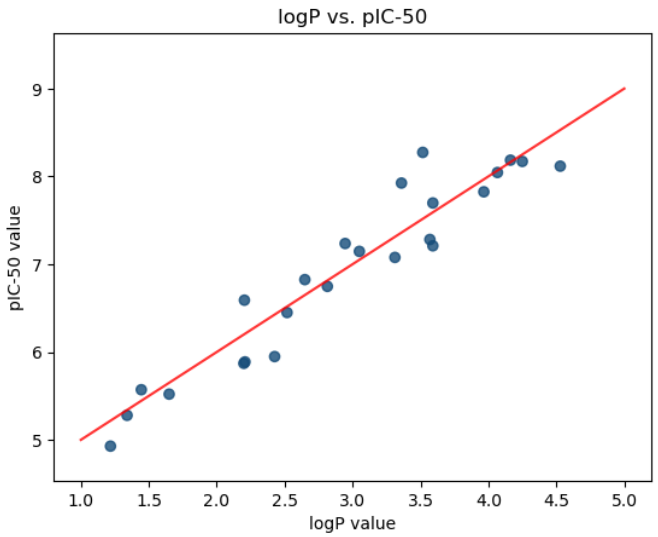

Figure 1: Early versions of QSAR by correlating physicochemical properties (logP) with biological activity (IC_{50})

Due to its beginnings and application in the drug discovery process, QSAR methods are mostly applied in the context of drug-like molecular structures. For a medicinal chemist, this yields immediate feedback on which functional groups or scaffolds to replace, which residues to add, or which properties to optimise. Applying mathematical models to molecules is not straightforward and introduces some challenges. Apart from the problem of how to represent molecules in mathematical equations, there are other challenges concerning the quality of resulting QSAR models when applied to molecular structures.

Specifically, in the case of 3D-QSAR[50], when the molecules' conformations are considered by the QSAR model, their coordinates might actually hinder rather than enable the generation of a useful model. Even though the precise location of atoms will provide additional information in a detailed analysis, QSAR models might struggle to extract meaningful information from them. On the one hand, the QSAR models are just approximations not intended for a detailed analysis. On the other hand, a lack of data will usually cause the models to "overfit"[51], meaning they perfectly describe the training set provided but fail to generalise to unseen new data. Common human sense, which lets us determine two neighbouring atoms of the same type with minor differences in their coordinates as the same, does not exist with precise mathematical models. Unless programmed explicitly by introducing thresholds or cutoffs, the neighbouring atoms will be considered differently. Such cutoffs can be artificially introduced by representing the molecules by their pharmacophores and pharmacophore features. Pharmacophore features represented by spheres add a level of fuzziness to the data representation. In detailed analyses, this effect is not desired, but

approximations such as QSAR models are likely to benefit from a certain degree of fuzziness. The two neighbouring atoms from the previous example will be treated equally in the case of pharmacophores due to the abstraction of their precise coordinates into fuzzy pharmacophore features with positional tolerance. This could facilitate the extraction of key binding motives, resulting in an increased ability to generalise to unseen data.

Secondly, various molecular motives might be recognised as similar by a human expert due to the equal types of interactions they might form with counterparties. An example would be a hydroxy and an amino group. Both are able to participate in hydrogen bond interactions as donors and acceptors (Figure 2).

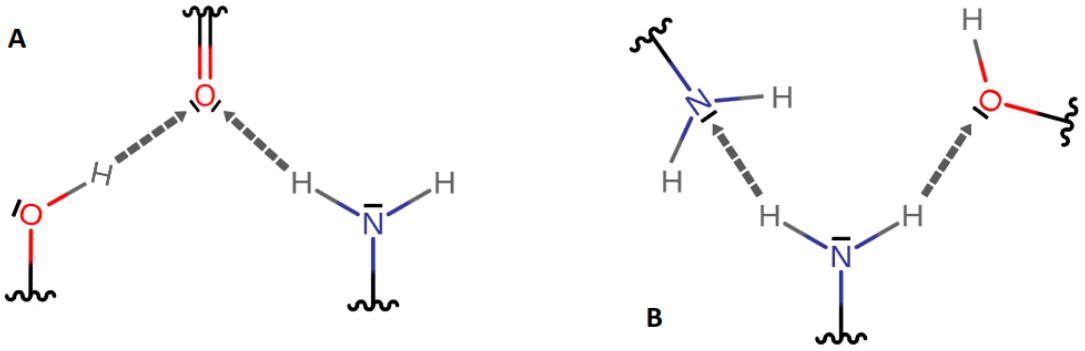

Figure 2: **A**: Amino and hydroxy group acting as HBD. **B**: Amino and hydroxyl group acting as HBA

Another example is the bioisosterism[52] of benzene rings and their analogues (Figure 3). Although their molecular structure looks utterly different, they have similar interaction profiles with protein targets[53]. Extracting such high-level properties from a molecular graph is not feasible for simple QSAR models, and even for complex QSAR models, it will cause problems. Pre-processing molecules by converting them to pharmacophores will extract binding interaction capability information from the molecules' functional groups but disregard their exact molecular structure. Thus, QSAR models could benefit from the pharmacophore representation instead of the molecule representation by extracting the relevant information from equal binding motives, facilitating the QSAR models' ability to generalise to unseen data.

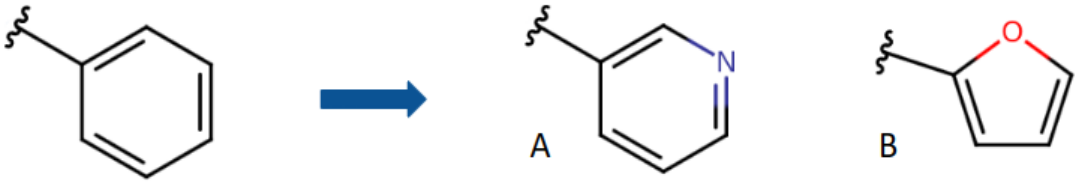

Figure 3: Bioisosterism of the phenyl ring and its analogues

The concept of pharmacophore features being able to represent various functional groups can easily be expanded to molecule fragments, even the entire molecule. The experienced reader will readily recognise this as the previously mentioned concept of scaffold hopping. So far, scaffold hopping (Figure 4) is only mentioned in the context of qualitative pharmacophores. However, when utilising pharmacophores to build QSAR models, it could be leveraged as an additional advantage for predicting out-of-domain molecules. Just like with single pharmacophore features and functional groups, molecules might have a similar binding motive but consist of an entirely different molecular

scaffold. QSAR models built from molecules should not be able to predict any values for such molecules (for more information, see Applicability domain). QSAR models built from pharmacophores, on the other hand, will recognise their similarities and readily offer a valid prediction for the new data samples.

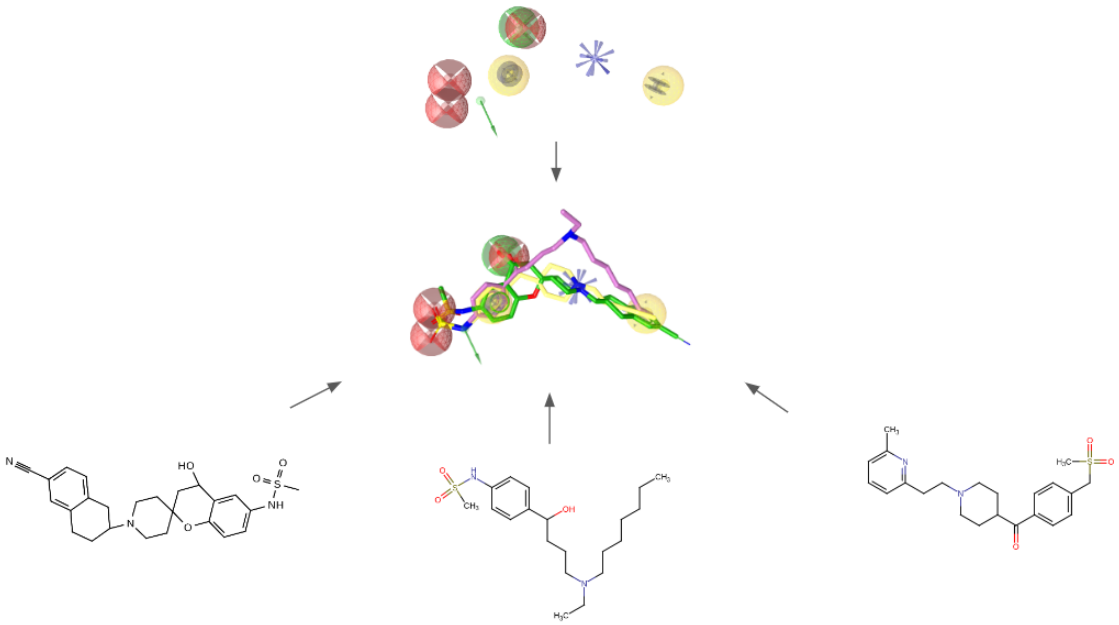

Figure 4: Alignment of various molecular scaffolds to the same pharmacophore highlighting its power of scaffold hopping.

In light of these various shortcomings of molecules, it is puzzling that pharmacophores are not considered more often as a suitable representation for QSAR models. Even though pharmacophores have been applied to quantitative modelling before (see Chapter 2: State of the art for more information), resources are scarce. Generalising the molecular coordinates and binding motives to pharmacophore features will allow the QSAR model to focus on identifying the relevant binding features without noise from other information. Especially when working closely with medicinal chemists, identification of those features is crucial, and explainability of the model should be a high priority for the computational chemist. QSAR based on molecular fingerprints[54] or physicochemical properties[55–57] runs the risk of identifying relationships with artefacts of the dataset. This means the built QSAR model is predictive for new data points, but for the wrong reason. Simply feeding the QSAR model information about the possible interactions in form of pharmacophore features puts the focus on identifying the key interactions instead of finding artefacts in the dataset.

Besides guiding the medicinal chemist in the quest for optimising lead compounds, QSAR models may also be applied a step earlier during hit identification. In contrast to pharmacophore screening, hits are obtained after predicting a library of molecules with the QSAR model, ranking the hits by their predicted values, and finally, selecting the top *n* molecules from the prioritised list. The hits are already ranked by their expected activity values, as compared to pharmacophore screening, where a list of presumably active compounds is obtained. Here, no further ranking is necessary.
Alternatively, a QSAR model can be used as the ranking method after a hit list has been obtained

from pharmacophore screening. The procedure is the same as when predicting a large molecule library. Thus, the hits are obtained in prioritised order after post-processing with a QSAR model.

Machine learning

As QSAR evolved over time and the datasets became more complex, the algorithms and methods processing the data had to become more powerful too[58, 59]. Therefore, it is no surprise that machine learning (ML)[60, 61] also found its way into the chemical domain and is applied to various QSAR models. ML is a specialised field in computer science that aims to use and develop algorithms which learn from a given set of data without explicit instructions. Many subdomains exist within ML, of which almost all are applied in computational chemistry. Among others, classification[62, 63], e.g., is used to categorise molecules into toxic and non-toxic molecules[64, 65]. Clustering[63, 66] is often applied to gain insights into large datasets, group the molecules, find hidden patterns, and add new molecules into the defined categories. Finally, regression[67, 68] is applied in QSAR models to identify quantitative relationships in a dataset. Regression models are a well-researched field, with a few of the most popular algorithms being linear regression[45] and its variations including regularisation[69], support-vector machines[70], and neural networks[71, 72].

Random forest regressor

A less popular algorithm in regression analysis[73], although widely popular in classification[74], is the random forest (RF)[75] algorithm. RFs are known for being extremely fast to train, being relatively robust to overfitting due to their random element, and offering a broad range of tools to inspect the model and provide insights on the model's explainability[76].
The RF is a collection of multiple individual decision trees[77, 78] (Figure 5). Given a dataset with multiple features, a decision tree will iteratively divide the dataset into various subsets based on the values from a single feature. The feature and the splitting criterion are chosen by a loss function, which aims to minimise the algorithm's loss[79]. More practically speaking, the loss function can be understood as an objective that aims to maximise the amount of information that can be explained by the model given its current parameters. At each dividing step, the splitting criterion that minimises the loss and maximises the explained information is chosen by the decision tree algorithm. The tree is a binary tree, whereas all data points with values below the criterion will be associated with the left child node, and all the data points with values above the criterion will be associated with the right child node. This process is repeated for each node until a predefined stopping criterion is reached or no node can be split anymore. A regression decision tree obtains its prediction by following the tree along its nodes for the given sample data point and applying the respective splitting criterion at each node. When a leaf node is reached, the average value of all data points within the left node is considered the predicted value of the sample data point. A RF regressor obtains its prediction for a sample data point by averaging over all predictions obtained from the individual decision trees.

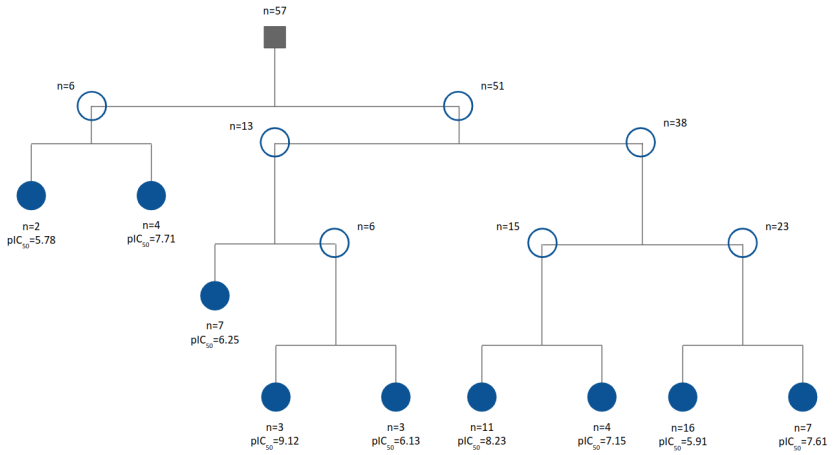

Figure 5: Example of a trained decision tree to predict bioactivity (pIC_{50}) for given input data.

Applicability domain

Major questions that should be asked in every machine learning study are: "To which data can this model be applied? Under what circumstances can the model's predictions be trusted? What are the model's boundaries? To which use cases can the model be extended while at the same time keeping its predictive power?" Questions like these and many more can generally be summarised and answered by defining an applicability domain (AD)[80, 81]. The AD of a ML model and QSAR model defines the type and range of data in which the model will yield results corresponding to the model's validation performance. The simplest question a suitable AD model will answer is: "Should the machine learning model be applied to this data point". If this question is answered with a "No", then the ML model's prediction should not be trusted, and a different model should be used. Despite its importance, many researchers still do not define or do not know how to define a model's AD[80].

The definition of an AD should always be tailored to the model, data, and problem at hand; however, a few accepted methods[81] exist and can be categorised into

- Range-based methods[82]
- Distance-based methods[83]
- Geometric methods[84]
- Probability density and distribution-based methods[85, 86]

Depending on the representation of the data some of these might be more applicable than others. The definition of an AD for molecules usually requires calculating mathematical descriptors for the molecules. Only then can the molecule's AD be defined, such as a range for physicochemical descriptors, similarity as a proxy for distance metrics, or the use of clustering methods for density-based methods.

Chapter 2: State of the art

Most of the research on pharmacophores concerns their qualitative nature and the development of better algorithms for virtual screening. Although hardly any research has been performed on pharmacophores in combination with QSAR, there are currently two software packages on the market which provide quantitative methods for pharmacophores in their portfolio. One of them is the popular Hypogen[87] algorithm available via BioVia's Discovery Studio[88], and the other one is PHASE[89], available as part of Schrödinger's Maestro Suite[90]. In this section, a brief overview of the strengths and shortcomings of the two methods, Hypogen and PHASE, will be given.

Hypogen

The Hypogen algorithm[87] aims to generate pharmacophores, named hypotheses in the Discovery Studio software, for predicting biological activity data. The algorithm is proprietary, and only an outline of it has been published. The dataset used by Hypogen must span a range of at least four orders of magnitude and contain structurally diverse molecules, although no restrictions are reported for structural diversity. The algorithm itself is split into three phases; a constructive, a subtractive, and an optimisation phase.

The constructive phase generates various hypotheses from a set of most active compounds, up to a maximum of eight compounds, which is dependent on the activity of the most active compound and the uncertainty in the measured activity of each compound. A compound is included in this set if it satisfies Equation 1.

$$MA * UncMA - (A / UncA) > 0.0$$

Equation 1: Equation to determine whether a compound is contained in the set of most active compounds. *MA* = activity of the most active compound; *UncMA* = uncertainty of the measured activity of the most active compound; *A* = activity of the compound in question; *UncA* = uncertainty of the measured activity of the compound in question.

From the set of most active compounds, the two most active ones are selected, and their pharmacophore features from all conformations are enumerated. These must match a minimum number of features from the remaining compounds; otherwise, they are removed. The result of the constructive phase is a collection of features matching the set of most active compounds.

Next, in the subtractive phase, this collection will be filtered to remove all features that also match the least active compounds from the initial dataset. The set of least active compounds is constructed, which follows Equation 2.

$$Log(A) - log(MA) > ActivityDistance$$

Equation 2: Equation to determine whether a compound is contained in the set of least active compounds. *MA* = activity of most active compound; *A* = activity of the compound in question; *ActivityDistance* = orders of magnitude that the compound's activity must be at least smaller than the activity of the most active compound. Per default, this value is set to 3.5.

All pharmacophore features contained in the set of least active compounds are enumerated and compared against the feature collection obtained from the most active set of compounds. If a feature from the most active set matches a feature from the least active set and this feature is shared by at least half the compounds in the least active set, then this feature is removed from the most active set.

Finally, the optimisation phase aims to improve the hypotheses which are remaining by optimising their score. The hypothesis' score is the sum of the hypothesis' weight, the hypothesis' error, and the

hypothesis' configuration. Scores are obtained for each hypothesis after applying small perturbations and measuring their complexity as well as the resulting error in activity estimates. The publication does not further specify how activity estimates are obtained from each hypothesis. The hypotheses with the best score are presented to the user.

Besides incomplete information on how the hypotheses generate their activity predictions, the hypogen algorithm has a significant disadvantage. It only uses information obtained from the few most active compounds. Any information included in lesser active compounds is not incorporated into the model, although this information is indirectly considered by removing hypotheses that match these compounds. The algorithm does not define an applicability domain and, in turn, offers no possibility for the user to gauge the confidence interval of the obtained predictions. Furthermore, due to the missing information from lesser active compounds in the resulting hypothesis, these compounds should be expected to be "out of domain", resulting in higher error and uncertainty in their predictions.
A positive characteristic of the algorithm is that it operates directly on the pharmacophore features and not the molecules to obtain the final hypotheses, although molecules are used for alignment.
The HypogenRefine algorithm aims to mitigate some shortcomings by removing the subtractive phase and instead placing exclusion volumes for features that match the least active compounds. How these exclusion volumes are incorporated into the activity prediction is unclear and could not be found in the literature.

PHASE

The PHASE algorithm[89] shows no similarity at all to the Hypogen algorithm. In advance, the main differences between the two algorithms are that PHASE is a grid-based approach, whereas Hypogen operates on the pharmacophore features' coordinates. The second difference is that Hypogen will generate various hypotheses, whereas PHASE will simply generate one 3D-QSAR model on top of an already existing pharmacophore.
Equipped with a training set of molecules with diverse activity values and a previously generated pharmacophore hypothesis, for example, from the set of training molecules, the PHASE algorithm will try to align the molecules to the pharmacophore. Structures that match less than three pharmacophore features are not considered by the rest of the algorithm. Once all molecules are aligned, a grid is placed around the pharmacophore and the aligned molecules, with a grid size of 1Å. The grid points will be populated with the pharmacophore feature values from the aligned training molecules. It is important to note that only features that can be mapped to the pharmacophore which served as the alignment template will be considered as a value for the grid points. Values from other pharmacophore features are disregarded by the algorithm. After all pharmacophore features are assigned to the grid points, the molecules are represented by bit-strings obtained from the grid points. Each grid point is associated with a bit location. Bits corresponding to grid points that have been populated by pharmacophore features are assigned a value of 1, and the other remaining bits are 0. This yields a unique string of bit values describing the molecules, whereas the bits can be interpreted as independent variables. Due to the large number of bits usually encountered, partial-least-squares[91] (PLS) is applied to retrieve a regression model from the dataset. The maximum number of PLS features used by PHASE equals one-fifth of the number of training molecules.

Among a few shortcomings in the PHASE algorithm, the most prominent one is that it only considers pharmacophore features that match the previously selected template pharmacophore. Similar to Hypogen, this limits the algorithm to a selected subset of features disregarding information from the remaining pharmacophore features. Furthermore, it requires molecules for alignment and model building, severely limiting the model's application to pharmacophores. Thus, it is impossible to purely rank pharmacophores with the PHASE algorithm if no molecule is associated, as would be the case in prioritising pharmacophores obtained from pharmacophore modelling.

Chapter 3: Aims and contributions of the thesis

The aim and the contribution of this thesis are guided by the overarching concept of developing next-generation pharmacophores with the help of machine learning, as defined in the project proposal of the IMI NeuroDeRisk[92] project. Considering the current state of research in the field of pharmacophores, the need for a quantitative pharmacophore model was identified. Despite a few commercial solutions offering such models in their software packages, they are facing critical shortcomings, which this thesis aims to improve.

At first, the work of this thesis aims to target the shortcomings of existing solutions and develop a novel quantitative algorithm employing pharmacophores and standard machine learning models to obtain its predictions. The generated model of such a method should itself be a pharmacophore to enable the prediction of external pharmacophores without the need for associated molecules. Furthermore, the developed algorithm should be able to work with small datasets, usually encountered in the early stages of the drug discovery process, where hardly any data is yet available. Finally, an applicability domain should be defined for the developed quantitative pharmacophore to provide additional insights into the quality of the predictions. The corresponding results will be presented in the first part of the Results and discussion section, QPHAR: quantitative pharmacophore activity relationship: method and validation.

Building on top of the first part and the developed QPhAR algorithm, the focus shall be applied to the dissemination and user-friendliness of the QPhAR algorithm. These results will be presented in A new set of KNIME nodes implementing the QPhAR algorithm.

Finally, a novel generation of algorithms for pharmacophore modelling should be devised in the last section, which aims to improve the current automated pharmacophore modelling algorithms. The algorithm should be able to derive a pharmacophore model in a deterministic manner without the need for a training set. The results are discussed in Applications of the novel quantitative pharmacophore activity relationship method QPhAR in virtual screening and lead-optimization.

II. Results and discussion

QPHAR: quantitative pharmacophore activity relationship: method and validation

Stefan M. Kohlbacher, Thierry Langer and Thomas Seidel

Kohlbacher, S.M., Langer, T. & Seidel, T. QPHAR: quantitative pharmacophore activity relationship: method and validation. J Cheminform 13, 57 (2021). https://doi.org/10.1186/s13321-021-00537-9

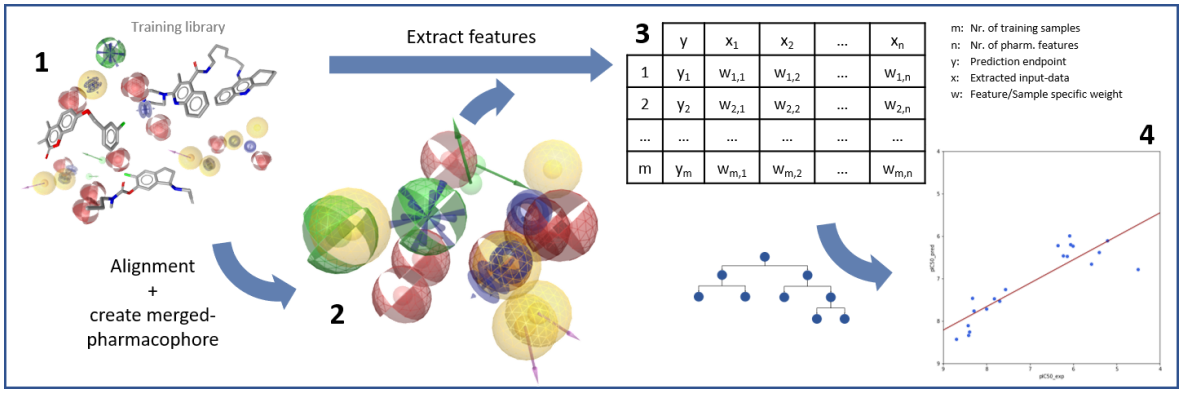

Figure 6: Graphical abstract published online[93] https://jcheminf.biomedcentral.com/articles/10.1186/s13321-021-00537-9

The concept of this study was devised in in-depth discussions between the three authors, S. Kohlbacher, T. Seidel, and T. Langer. The algorithm and code of the QPhAR algorithm were developed and written by S. Kohlbacher. Datasets selection, model training, and statistical validation of the model were carried out by S. Kohlbacher. The manuscript was written and prepared by S. Kohlbacher. T. Seidel and T. Langer have supervised the work and revised the manuscript.

Kohlbacher *et al. J Cheminform* (2021) 13:57
https://doi.org/10.1186/s13321-021-00537-9

Journal of Cheminformatics

METHODOLOGY Open Access

QPHAR: quantitative pharmacophore activity relationship: method and validation

Stefan M. Kohlbacher, Thierry Langer and Thomas Seidel*

Abstract

QSAR methods are widely applied in the drug discovery process, both in the hit-to-lead and lead optimization phase, as well as in the drug-approval process. Most QSAR algorithms are limited to using molecules as input and disregard pharmacophores or pharmacophoric features entirely. However, due to the high level of abstraction, pharmacophore representations provide some advantageous properties for building quantitative SAR models. The abstract depiction of molecular interactions avoids a bias towards overrepresented functional groups in small datasets. Furthermore, a well-crafted quantitative pharmacophore model can generalise to underrepresented or even missing molecular features in the training set by using pharmacophoric interaction patterns only. This paper presents a novel method to construct quantitative pharmacophore models and demonstrates its applicability and robustness on more than 250 diverse datasets. fivefold cross-validation on these datasets with default settings yielded an average RMSE of 0.62, with an average standard deviation of 0.18. Additional cross-validation studies on datasets with 15–20 training samples showed that robust quantitative pharmacophore models could be obtained. These low requirements for dataset sizes render quantitative pharmacophores a viable go-tomethod for medicinal chemists, especially in the lead-optimisation stage of drug discovery projects.

Keywords: Pharmacophore, QSAR, Regression, Machine learning, Quantitative-pharmacophore-model

Introduction

Quantitative structure–activity relationship (QSAR) studies were first introduced by Hansch et al. [1] in 1962 and have been growing in popularity ever since. Starting with simple correlations studies of chemical and biological properties, such as logP and K_i values, QSAR has evolved into a sophisticated method applying complex machine-learning (ML) models [2] on vast amounts of chemical data [3], often using more than a few thousand descriptors. QSAR models are not only useful for internal assistance in the drug discovery process, but highly validated and robust models have even been built by the FDA to assist the drug-approval process [4].

Over the years, QSAR has been influenced heavily by advanced machine learning [2] and other data processing systems, which effectively allows the researcher to extract more complex relationships from their data. With the use of more capable models, more complex input data can be processed. Countless descriptors or fingerprints [5] have been derived from 2D molecular structures but do not take into account spatial information and molecular conformation. Spatial information becomes even more critical when dealing with stereoisomers [6].

The popular QSAR modelling algorithm CoMFA [7], developed in the 80 s, uses 3D conformations of molecules as input, aligns them to each other, and then creates a predictive model from the molecules' calculated steric and electrostatic interaction fields. The concept has gained wide popularity but never extended to different input domains than molecules. The method PHASE [8] proposed by Schrödinger [9] has taken this approach

*Correspondence: thomas.seidel@univie.ac.at
Division of Pharmaceutical Chemistry, Department of Pharmaceutical Sciences, University of Vienna, Althanstraße 14, 1090 Vienna, Austria

a step further. In addition to, or instead of calculating electrostatic interaction fields of the molecules, it is possible to generate pharmacophore fields from the input molecules. The same ML-algorithm as used in CoMFA, PLS (partial least squares), is then applied to create a predictive quantitative model. At the time, this has been a novelty since pharmacophores have only been used for qualitative virtual screening studies. Using pharmacophore fields derived from functional groups for quantitative modelling extends the CoMFA concept, using abstract 3D information of molecules for QSAR. Nevertheless, these pharmacophore fields are derived from molecules and a pure quantitative algorithm applied on pharmacophores has never been presented before.

Using pharmacophores as input in QSAR studies has several advantages: due to the abstract nature of pharmacophores, they are less influenced by small spatial perturbations of molecular features characteristic for such interactions. For example, bioisosteres are often highly similar in their interaction profile. They might cover, however, entirely different functional groups and substructures. Building a QSAR model on such data inevitably introduces a bias towards the predominant bioisosteric form occurring in the dataset. Pharmacophores, on the other hand, transform different functional groups with the same interaction profile into an abstract chemical feature representation associated with a particular non-bonding interaction type, such as a π-stacking interaction or H-Bond donor/acceptor interaction. This generalisation makes quantitative models more robust and less dependent on the dataset being used. Primarily in biological assays, robust predictive models are essential to avoid modelling the experimental noise [10].

Virtual screening takes advantage of pharmacophores' abstract nature to achieve an effect known as "scaffold-hopping" [11]. Here, pharmacophores help to overcome a structural molecular bias by only considering the interaction patterns but not the molecular structures. A carefully constructed quantitative pharmacophore model will build on these advantages and the scaffold-hopping ability to harness its strengths. Besides abstracting molecular structures, pharmacophores also abstract the exact steric location and orientation of interactions by introducing tolerance ranges. Losing information on the precise position of possible interactions might not be desired with highly conserved protein targets. In general, however, generalisation is considered positive and avoids overfitted models.

Pharmacophore modelling is often used in combination with virtual screening to find novel hits. Deciding on the best pharmacophore model for virtual screening runs is often a tedious process relying on a large dataset of mostly artificially generated decoys and some truly active compounds. In addition to requiring large amounts of data, this evaluation process relies on the binary-classification of molecules into active and inactive ones. Molecules with similar activity values close to the cut-off are classified differently, although they demonstrate a quite similar experimental behaviour.

A quantitative pharmacophore model would be able to score other pharmacophore models and assign an estimated non-binary activity to these pharmacophores. The (biological) activity of pharmacophores can be interpreted as the expected activity of molecules matching such a pharmacophore. In the context of virtual screening, it is expected that the scored pharmacophore will retrieve molecules from a database with similar activity values. Therefore, the quantitative pharmacophore model can be easily applied as a ranking method to prioritise pharmacophore models generated by a researcher.

Despite the possible advantages hardly any research was done on quantitative pharmacophores and related methods. In contrast, QSAR applied on molecular structures has fostered plenty of research and a google-scholar search for the query "quantitative structure–activity relationships" yields close to 5 million results. Nevertheless, two commercially available tools have been released which are able to relate pharmacophores quantitatively to biological activity or other specified properties.

PHASE is a commercially available tool implemented in Maestro [9]. Besides pharmacophore perception, it allows for quantitative rationalisation of activity data based on 3D pharmacophore fields obtained from a set of ligands. Pharmacophores are created for each aligned ligand, whereas the alignment is not done automatically and needs to be considered by the user. The aligned pharmacophores are placed into a vectorised box, each voxel containing information about the value of the pharmacophore fields in that location. The box is used as input for a PLS-algorithm to regress the pharmacophore fields against a set of activity values. As output, the user gets a model displaying favourable as well as unfavourable regions contributing to the activity values. Additionally, the activity of new ligands can be predicted by feeding them to the model after alignment.

Even though PHASE provides one of two available QSAR methods for pharmacophores, it still relies on molecules as input for alignment and model building. Due to this shortcoming, apo-site derived pharmacophores or pharmacophores obtained from ligand-based modelling can only be predicted via workarounds. Therefore, pharmacophore QSAR within PHASE is similar to atom-based QSAR, except that an additional step for calculating the pharmacophore fields is carried out.

The second available method for pharmacophore QSAR is the Hypogen [12] algorithm implemented in the Catalyst program, which now is part of BioVia's [13] Discovery Studio [14]. The Hypogen algorithm works utterly different from the PHASE algorithm, directly operating on

pharmacophore features instead of using grids as a proxy. First, a subset of the most active compounds is chosen. All possible pharmacophore hypotheses from the two most active compounds are enumerated and must fit a minimum subset of the remaining compounds in the most active subset to be considered by the algorithm. From this generated set of pharmacophore hypotheses, the ones matching a group of inactive compounds are removed in a follow-up phase. In a third final phase, small perturbations are introduced to the remaining hypotheses, which are then scored based on the RMSE of predictions against the training set. The Hypogen refine algorithm extends this method by adding exclusion volumes and introducing another term in the loss function.

In contrast to PHASE, Hypogen is operating directly on pharmacophores without the need to provide the underlying molecules. However, a drawback of this method is still that it builds the quantitative models from a selected subset of highly active compounds. Even though refinement considers less-active compounds, we would expect predictions for pharmacophores obtained from less active molecules to be worse due to missing domain knowledge. After the model-building is done, no single quantitative model is selected, but a set of possible solutions is provided to the user, adding some ambiguity about the model's quality.

Having in mind the potential advantages of a quantitative method that is based on pure pharmacophoric representations, we developed and herein present a novel approach for the generation of quantitative pharmacophore models. Based on a small dataset of molecules and/or pharmacophores, the proposed algorithm will first find a consensus pharmacophore (merged-pharmacophore) from all training samples. The input pharmacophores, or pharmacophores generated from the input molecules, will then be aligned to the merged-pharmacophore. For each aligned pharmacophore, information regarding its position relative to the merged-pharmacophore is extracted. This information is then used as input to a simple machine learning algorithm which derives a quantitative relationship of the merged-pharmacophores' features with biological activities.

Methods

Datasets

Phase dataset

The dataset previously published by Debnath [15] et al. in 2002 was used as a benchmark datasets against the PHASE algorithm. Compounds were obtained in SMILES format, which served as input for the generation of 3D conformations. Conformations were generated using iConfGen [16] provided by LigandScout [17]. For all parameters default settings were used, and the maximum number of output conformations was set to 25. Training and test data were split according to the published results by Dixon et al. [8]. Molecule number 67 was removed from the dataset due to missing experimental activity data.

ChEMBL datasets

Datasets obtained from ChEMBL were used for cross-validation and assessment of the model's robustness. The 23 most popular QSAR targets (Table 3), according to a list published by Cortés-Ciriano [18], were chosen for validation studies. The ChEMBL database was queried with the UniProtId of these 23 targets using the chembl-webresource-client [19]. Biological activity data ('standard_value'), in the following referred to as activity, was obtained for each compound and filtered by the following parameters:

- standard_type: 'IC50' or 'Ki'.
- standard_units: 'nM'.
- standard_relation: '='.
- assay_type: 'B'.
- target_organism: 'Homo Sapiens'.

Activity readouts from biological assays depend heavily on the assay conditions and the environment. Due to the high experimental noise and the resulting inability to combine various assays, the datasets were separated by their 'assay_chembl_id'. This resulted in datasets of ~ 20 to ~ 100 molecules.

The assay datasets were cleaned in a post-processing step to ensure the data was qualified for QSAR modelling. Datasets with less than three log-units difference between the minimum and maximum activity value were dismissed. Furthermore, to avoid overfitting and bias when training the ML models, the assay datasets were filtered by heterogeneity. Ideally, compound activities are distributed equally over the activity value range. To model data heterogeneity, the KL-divergence was calculated against a uniform distribution. The higher the KL values were, the more clustered the dataset was found to be. All datasets with KL values above a cut-off of 0.75 were discarded. The KL-divergence was calculated according to Eq. (1), whereas P and Q denote discrete uniform distributions over activity values for a given dataset. P represents the estimated uniform distribution of the datasets, Q resembles the reference uniform distribution over activity values, a being the minimum and b the maximum. P is estimated by binning the activity values into N (sample size) bins. Each P(x) is defined by the frequency of activity values in each bin x.

$$Eq1 : KL(P, Q) = \sum_{x \epsilon X} P(x) * log\left(\frac{P(x)}{Q(x)}\right) \quad (1)$$

The ChEMBL datasets were split into training and validation data via a fivefold random split. This was done by

applying a stratified K-fold procedure for quantitative data. Since stratified K-fold data splitting is usually used on classification datasets, a workaround was constructed where activity data was binned in K, five, classes [20]. In addition, a 20–80 split (20% training data, 80% validation data) was generated from the datasets as well as the standard 80–20 split (80% training data, 20% validation data). The 20–80 split aims to mimic a typical SAR setting experienced by a medicinal chemist, where only a limited amount of data is usually available. Training folds typically consisted of 10–15 samples.

For all datasets, 3D conformations were generated using LigandScout's iConfGen. All parameters were left at their default values, and the maximum number of generated conformations was set to 25.

Baselines

Cross-validation studies were also carried out for two baseline methods using all ChEMBL datasets. One baseline model utilised the number of pharmacophore features per molecule as input to train a regression model. The second model was trained on physico-chemical descriptors of the input molecules. Only a limited number of descriptors was used due to the small dataset sizes. The following seven descriptors were calculated for the input molecules: number of H-Bond Donors/Acceptors, number of rotatable bonds, molecular weight, number of heavy atoms, cLogP [21], TPSA [22].

Additionally, the applicability domain of the baselines models was defined by calculating the min/max values [23] of the input vectors on the training fold. Test samples were deemed to be out-of-domain if their input vectors either fell below or exceeded the before determined min/max values, respectively.

Machine learning model generation

To ensure a fair comparison, all baseline models were trained using the same machine learning algorithm and the same set of parameters as for the quantitative pharmacophores. Cross-validation studies were carried out using the random forest algorithm [24] implemented in the scikit-learn python package. The parameters 'n_estimators' and 'max_depth' were set to 10 and 3, respectively. The remaining parameters were kept at their defaults. No hyperparameter optimisation was performed for cross-validation studies due to the small datasets. Additional required parameters for the quantitative pharmacophores were set as follows (also the default parameters): fuzzy = True, weightType = distance, mostRigidTemplate = True.

Training the machine learning models of the quantitative pharmacophores on the PHASE dataset employed a protocol for hyperparameter optimisation. The following QPhAR specific parameters were subject to optimisation:

- weightType: [distance, nrOfFeatures, None].
- modelType: [random forests, ridge regression, PLS regression, PCA + ridge regression, PCA + linear regression].
- threshold: (1, 1.5, 2).

The parameters of the machine learning models were set to their default values, except for the random forest algorithm. Here the parameters 'n_estimators' and 'max_depth' were optimised by (10, 15, 20) and (2, 3), respectively.

Quantitative pharmacophore algorithm

The generation of a quantitative pharmacophore model proceeds over five consecutive steps (Figs. 1, 2). At first, a template needs to be chosen (Step 1). The selected template could either be one of the training samples, such as the most rigid molecule, or any other pharmacophore or molecule the user deemed relevant. Selecting the template is highly important and can make or break the algorithm. If a poor template is chosen, the resulting alignment of the training samples might yield a quantitative pharmacophore model of low quality. In the second step, the training set is aligned to the selected template. Once all the samples are aligned, the pharmacophore features of the training samples are clustered (Step 3). In the following post-processing step (Step 4), representative features will be selected or generated for each found cluster. Furthermore, clusters with non-conclusive information get discarded, and only relevant clusters will be kept within the quantitative pharmacophore model. Finally, the remaining pharmacophore features from each cluster are used as input for training a regression machine learning model (Step 5).

Template selection

The quantitative pharmacophore algorithm provides two options for selecting a template: either a pharmacophore is given as input or a set of two molecules. A provided pharmacophore will be directly selected as the template. If two molecules are given, they will be aligned to each other via pharmacophore alignment that also regards the conformational flexibility of the molecules. Once the best alignment between the two molecules is found, a pharmacophore will be generated from the first molecule, which is then used as the template. A reasonable choice as the first molecule is to select the most rigid molecule from the training set.

Alignment

Once a template pharmacophore has been selected, the remaining training set will be aligned to the template using pharmacophore alignment and a greedy optimisation procedure to select the best fitting conformations.

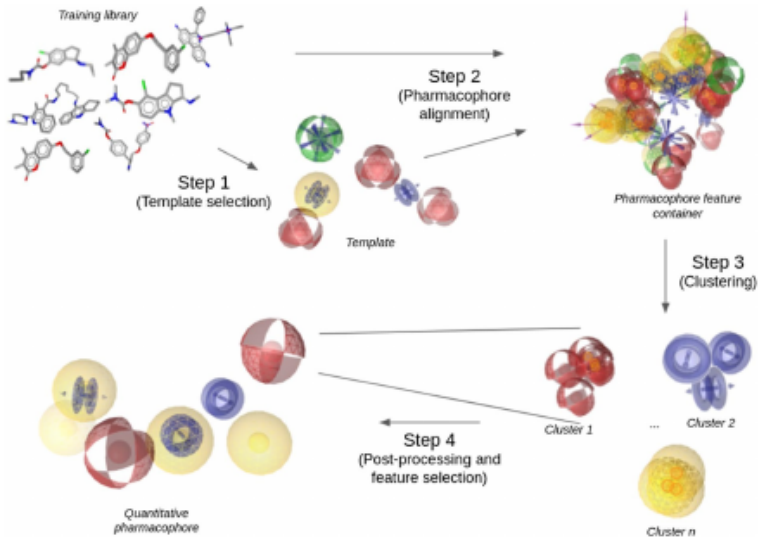

Fig. 1 Schema of generating the quantitative pharmacophore model

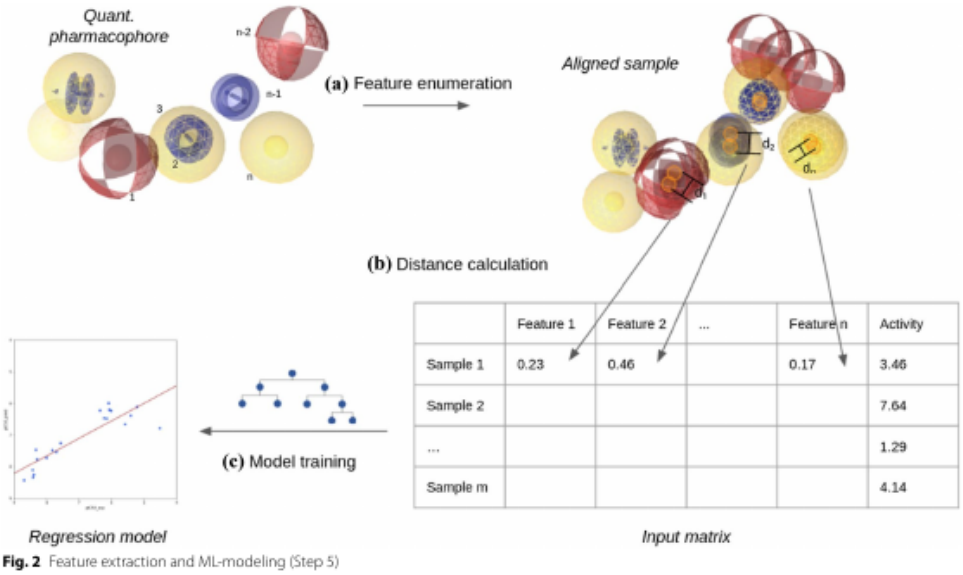

Fig. 2 Feature extraction and ML-modeling (Step 5)

The aligned pharmacophores and their features are stored within a single pharmacophore data structure (container), creating a merged-pharmacophore-like outcome. Each pharmacophore feature is associated with the activity value of the parent pharmacophore. Directed pharmacophore features are converted to undirected spherical pharmacophore features due to the noise introduced by directed features. Each pharmacophore feature type is collected in a separate container, resulting in six distinct containers: hydrophobic (H), aromatic (AR), positive/negative ionisable (PI, NI), H-Bond donor/acceptor (HBD, HBA).

Clustering

A clustering algorithm then processes the collected pharmacophore features. The minimum distance hierarchical clustering algorithm is applied, whereas its cutoff is being treated as a hyperparameter. The default cutoff value is the radius of the pharmacophore features, 1.5 Å. A distance matrix containing the calculated euclidean distances between all features in the container is used as input for the clustering algorithm.

Post-processing

After successful clustering, two post-processing steps are applied. First, representative pharmacophore features are selected for each cluster and then clusters with non-conclusive activity data are removed, keeping only high impact pharmacophore features.

Representative feature selection

Ideally, each cluster can be represented by a single feature. If this is not possible, several features are placed to represent the cluster. The representative feature can either be one of the existing features in the cluster or be the product of merging all features in the cluster. A feature represents another feature if they overlap, meaning their distance is smaller than the radius of these pharmacophore features. The merged feature then inherits all activity values from the features it represents. Additionally, the number of included features is stored. Both these properties will be used in the second post-processing step.

The following options are considered to find representative features:

- Clusters containing a single feature: the cluster is already represented by the feature itself; therefore, no further modifications are necessary.
- Clusters containing multiple features: at first, each feature is probed as a representative feature. If any of the features within a cluster overlap with all other features, that feature is used as the cluster's representative feature. Otherwise, a new feature is created. Three strategies are considered in order to determine the location of the new feature:
  - The feature is placed at the centroid of the cluster. If all features in the cluster overlap with the new feature, they are merged into the new feature and their properties assigned to the merged feature.
  - The new feature is placed in the centre of the cluster. The process of the centroid feature is repeated.
  - If all the aforementioned options fail to yield a representative pharmacophore feature, multiple features will be selected to represent the cluster. This process has no clear solution. Therefore, we apply a greedy iterative algorithm that maximises the number of represented features at each step. As long as non-merged features exist, the feature overlapping with most other features is selected as a representative feature. All properties of merged features are assigned to the selected feature. The just-merged features are then removed from the cluster. This process is repeated until all features in the cluster are assigned to a representative feature.

Removing ambiguous features

At this point, the quantitative pharmacophore consists of several features, each representing a cluster of pharmacophore features from the training set. However, some of these features will not add information to the final model. This is easy to imagine in the simple case of two molecules having the same scaffold but different residues and activities. One of the molecules shows high biological activity, whereas the other molecule is not active. A merged-pharmacophore will contain features representing the scaffold, as well as features representing the different residues. It is clear that the residues' features can explain the activity of these molecules, and the scaffold does not contribute any information. This rationale is applied to the quantitative pharmacophore. Therefore, features containing activity values spanning more than half the distance of global maximum and minimum activity values are removed. They are considered non-conclusive regarding their activity and will not contribute relevant information to the quantitative model. Furthermore, pharmacophore features encountered only once are considered outliers and are removed from the quantitative pharmacophore model too due to missing validation of the feature's importance. Finally, a cleaned, merged-pharmacophore is obtained, serving as a reference model in the machine learning procedure.

ML model and weight selection

Given the training set and the reference merged-pharmacophore, input vectors are generated for each training sample. The vector has the size of the number of pharmacophore features in the reference pharmacophore. Each entry in the vector corresponds to one of the reference pharmacophore features. The vector entries are populated by the inverse euclidean distances between the features from the reference pharmacophore and the aligned training sample features. If these two features do not overlap, the entry is filled with a zero value. It follows that the quantitative pharmacophore algorithm does not consider features from training samples without corresponding features in the reference pharmacophore. This is on purpose and will be explained further in the "Applicability domain" section.

Optionally, weights can be added to the input vectors. If 'weightType' None is chosen, the distance values are converted to binary values, 1 for overlapping features, 0 for non-overlapping features. Specifying no weight type will keep the input vector as it is. The third option is to weigh the input vector by the number of features. Features consisting of a higher number of merged features are weighted heavier than features with fewer data.

Once input features are generated for each aligned training sample, a regression machine learning model is trained on the input vectors and the activity data. The type of the machine learning model is a hyperparameter of the quantitative pharmacophore algorithm. However, it is recommended to use simple algorithms to avoid overfitting. A linear regression model, a ridge regression model, and a heavily restricted random forest model have been applied in this study.

The current version of the algorithm has been implemented in Python using the Chemical Data Processing Toolkit [25] (CDPKit) for the representation and processing of molecule and pharmacophore data. Machine learning models were trained using the scikit-learn package [26]. The code and all datasets are available at https://github.com/StefanKohlbacher/QuantPharmacophore.

Applicability domain

Applicability domains are an essential part of machine learning algorithms [27, 28]. Samples outside the domain of the training data cannot be predicted with high confidence. Moreover, the predictions could be random values not reflecting the true values at all. Therefore, the applicability domain for quantitative pharmacophore models gets defined by two factors. First, suppose a new sample, molecule or pharmacophore cannot be aligned to the template of the quantitative pharmacophore due to missing features or dissimilarities. In that case, the sample is deemed as out-of-domain and will not be predicted by the model. Second, due to the removal of features with non-conclusive activity values, some features in the query sample may not be matched with the quantitative pharmacophore. Furthermore, features not overlapping with any of the model's features will not contribute information to the input vector. Even though the model is missing information in such cases, these non-overlapping features must not be included in the input vector due to missing training data in these regions. Therefore, not the entire sample is out-of-domain, but only certain features within the sample. Features out-of-domain are not included during inference.

Results and discussion

The quantitative pharmacophore model is obtained by first creating a merged-pharmacophore. Data from the merged-pharmacophore and the training set is then used to fit a machine-learning model. Training of the ML-model is carried out with the same dataset as the merged-pharmacophore was created from. Therefore, the dataset is required to have known activity values for each sample. Creating a merged-pharmacophore as the underlying model has several advantages. For one, it keeps the model explainable and straightforward, unlike many other black-box ML algorithms. Second, due to the familiar merged-pharmacophore concept and representation, the model can quickly be adopted by scientists already familiar with such tools. The steep learning-curve allows a medicinal chemist to iterate through ideas quickly.

As mentioned before, there are only a few tools currently available to the scientific community allowing scientists to perform QSAR from pharmacophores. Here we do not directly compare against these methods since the quantitative method described in this paper expands to domains not accessible by previous algorithms. Nevertheless, we show that the quantitative pharmacophore performs similar to the PHASE algorithm on molecule datasets. Furthermore, based on a broad set of commonly used protein-targets for QSAR, we prove that the method shows robust performance over a wide variety of datasets.

PHASE vs. QPhAR

The paper published by Dixon et al. [8] in 2006 describing the PHASE algorithm compares its method against an even earlier published paper [15] using the Hypogen algorithm to predict the activities of a dataset. The quantitative pharmacophore was trained on the same training set, 20 samples, as described in the paper by Debnath et al. [15]. It was then evaluated on the holdout test set containing 57 molecules (originally 58, but one sample had no reported activity value). The reported RMSE and R^2 values on the test set of the PHASE algorithm were 0.822 and 0.407 (Fig. 3A), respectively. In contrast, the quantitative pharmacophore model could achieve an RMSE of 0.85 and an R^2 of 0.365 (Fig. 3B). These two models are

comparable to each other and are expected to yield similar results on new datasets. A student t-Test has further validated the fact that the models perform on par. The student t-Test was performed using scipy's implementation for related samples, as is the case here for predictions from the same test set. The t-Test resulted in a p-value of 0.29. Therefore, the null-hypothesis that the two models perform differently cannot be rejected. To reject the null-hypothesis a p-value of at least 0.05 or lower should be achieved. Figure 5 shows the mean RMSE values of bootstrapping the testset for both the PHASE and the QPhAR models. The 95% confidence intervals obtained from bootstrapping show a significant overlap between the two methods (Fig. 5 Appendix). The distribution of prediction errors is plotted in Fig. 6 (Appendix) for both models. An interesting fact that can be observed is the slightly lower median of the quantitative pharmacophore model compared to PHASE. Although, this does not indicate superiority over PHASE due to the previously stated equality of the two models.

However, it is important to keep in mind that both models were trained and evaluated on molecules, which is not the main focus of the method described here. Prediction of pharmacophores not obtained from molecules is one of the unique strengths of the quantitative pharmacophores. Alignment and prediction of such pharmacophores has not been possible before and prediction of pharmacophores with other methods still relied on molecules for alignment. Therefore, the method described here does not aim to compete against previous algorithms but rather expands the toolbox available to researchers while still providing similar quality results as state-of-the-art methods.

Cross-validation

Besides comparing our method against existing methods, CV was carried out on more than 250 distinct datasets to test the quantitative pharmacophores' general applicability across a wide range of datasets. All training-validation runs used default parameters to gauge the quantitative pharmacophores' effectiveness when used out-of-the box. Simple baseline models were built to demonstrate superior behaviour over standard methods. As baselines, the number of pharmacophore features in the training set was regressed against the activity endpoint. Furthermore, a set of simple physicochemical properties was calculated and used as a second baseline model input. To ensure a fair comparison, all baseline models used the same machine learning algorithm with default parameters as the quantitative pharmacophores. Cross-validation runs were evaluated by calculating the mean RMSE and the standard deviation of the RMSE over the five individual runs (Fig. 4). Additionally, the applicability domain has been investigated for the test set samples. Samples deemed out-of-domain for baseline models were further analysed and compared against in-domain predictions from the quantitative pharmacophores. No sample was found to be out-of-domain for the quantitative pharmacophores.

Mean RMSE values of the CV (80–20 split; Fig. 4A) range from 0.20 to 1.27, with an average over all

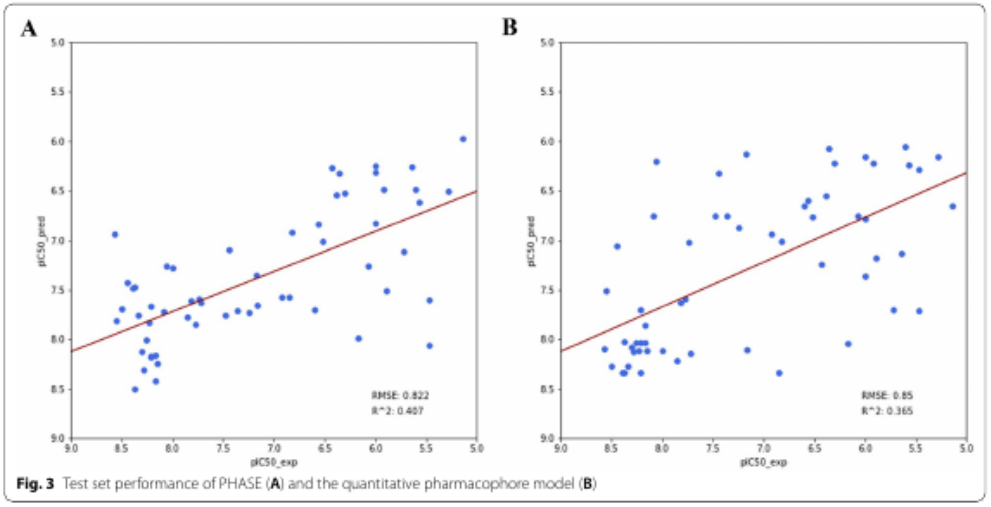

Fig. 3 Test set performance of PHASE (**A**) and the quantitative pharmacophore model (**B**)

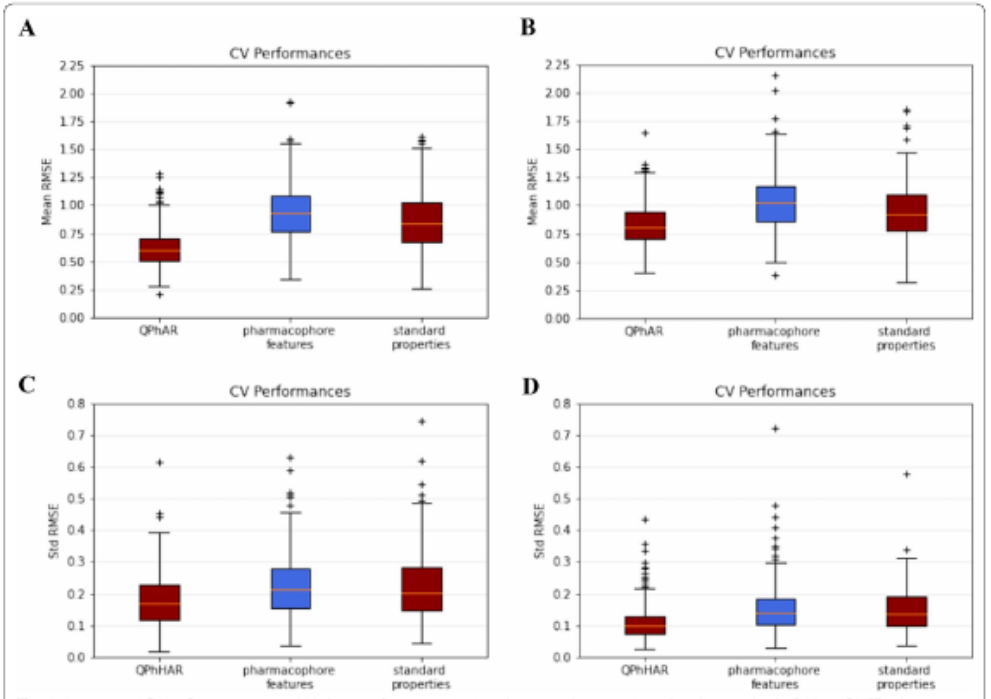

Fig. 4 Aggregated CV performance over > 250 datasets from quantitative pharmacophores and two baseline methods. **A** Mean RMSE values of models in 80–20 split. **B** Mean RMSE values of models in 20–80 split. **C** Mean standard deviation of models in 80–20 split. **D** Mean standard deviation of models in 20–80 split

datasets of 0.62. Generally, an average mean RMSE of 0.62 over all datasets is perceived as quite good, considering that lower RMSE values are a strong indication of modelling the experimental noise in the datasets. With that in mind, the small number of datasets with mean RMSE values of 0.20 from CV are very likely to overfit. On the other hand, the worst RMSE of 1.27 was obtained with default parameters and in depth analysis and model optimisation are likely to sharply increase the model's performance. Along with mean RMSE values, the standard deviation of the model's CV performance was calculated, which on average, was 0.18 across the five folds CV. Further emphasis in the analysis was put on the applicability domain. The applicability domains for the baselines were defined as described in the "Methods" section. The applicability domain for the quantitative pharmacophore is defined by a possible alignment of the test samples to the model. If no alignment with the model can be found, the sample is out-of-domain. During CV runs, no test samples were found to be out-of-domain for the quantitative pharmacophore model. However, out of 8520 test samples from all CV runs, around $^1/_8$ (901) of the test samples were found to be out-of-domain for the physico-chemical properties baseline and ~ 3% (244) for the pharmacophore features baseline. Predictions of these samples were made nevertheless and then further analysed in comparison with the quantitative pharmacophores. Our model could improve on these predictions in 71%, and 72% of all out-of-domain samples for the properties and features baseline, respectively. Figure 7 (Appendix) shows the number of times the quantitative pharmacophore model yielded better predictions than the baselines, when the test-samples were found to be out-of-domain. For a summary of the quantitative pharmacophores and baselines 80–20 CV results see Table 1.

Evaluation of the 20–80 split (Fig. 4B) yielded an average RMSE of 0.83 over all datasets, with a minimum RMSE of 0.41 and a maximum of 1.65. As expected, these models' performance is not as high as the average

performance from models trained on the 80–20 split. These results agree with the general notion that more data will improve a machine learning model's quality. However, considering that the training sets contained only 10–15 samples, the model's performances are respectably high. A valid concern often raised with such low training set sizes is the potential overfitting of the models. Here, we can exclude overfitting since the trained models were evaluated on validation sets four times larger than the training sets. If models would overfit, the performance on the validations sets would be considerably worse and therefore not be in agreement with obtained results. Furthermore, the standard deviation over all datasets of the 20–80 CV is much lower than the 80–20 CV, Fig. 4D, C, respectively. The low standard deviation further strengthens the point that the models are not overfitting any of the CV splits. Nevertheless, the small standard deviation compared to the 80–20 split is surprising since smaller datasets are expected to increase the model's performance variance. Achieving a smaller variance during CV with smaller datasets further boosts confidence in robust quantitative pharmacophores.

As for the 80–20 split, additional analysis was performed for samples classified as out-of-domain. Due to the small training sets, considerably more samples from the test set were found to be out-of-domain for the baseline models as with the 80–20 split. Out of 19,805 test samples in total, > 23% (4602) were deemed as out-of-domain for the properties baseline and ~ 7% (1399) for the features baseline. No test sample was found to be out of domain for the quantitative pharmacophores. 65% of the time predictions were improved by the quantitative pharmacophores (Fig. 8 Appendix). For a summary of the quantitative pharmacophores and baselines 20–80 CV results see Table 2. For a list of all target-proteins used in the CV studies see Table 3.

In direct comparison to the baselines, the quantitative pharmacophore model was superior in ~ 9/10 cases measured by the RMSE of CV (80–20 split as well as 20–80 split). In the 80–20 split CV, the quantitative pharmacophore could improve the mean RMSE by 34% over the pharmacophore features baseline and 27% over the physico-chemical properties baseline. Similar results can be seen on the 20–80 split, where 20% and 12% improvement was achieved, respectively.

Conclusion

Pharmacophores are widely applied in a qualitative manner for hit identification in virtual screening experiments and hardly any information can be found on the quantitative use of pharmacophore models. PHASE and Hypogen, only accessible in commercial packages, currently provide the only two algorithms which allow for quantitative insights on pharmacophore models. Targeting their drawbacks, such as alignment, the requirement of molecules for training, and user-friendliness, we present a novel quantitative pharmacophore generation algorithm for QSAR studies. The algorithm first creates a merged-pharmacophore from a given set of molecules and/or pharmacophores. Information obtained from aligning the training set to the merged-pharmacophore is then used to train a machine-learning model. We performed extensive cross-validation on a large variety of

Table 1 Results for 80–20 CV of quantitative pharmacophores and baselines

	Quantitative pharmacophores	Features baseline	Properties baseline
RMSE (mean)	0.62	0.95	0.86
RMSE (standard deviation)	0.18	0.24	0.24
RMSE (minimum)	0.2	0.34	0.26
RMSE (maximum)	1.27	1.92	1.61
Samples out of domain	0	244 (2.9%)	901 (10.1%)

Table 2 Results for 20–80 CV of quantitative pharmacophores and baselines

	Quantitative pharmacophores	Features baseline	Properties baseline
RMSE (mean)	0.83	1.04	0.94
RMSE (standard deviation)	0.19	0.25	0.25
RMSE (minimum)	0.41	0.39	0.32
RMSE (maximum)	1.65	2.15	1.86
Samples out of domain	0	1399 (7.1%)	4602 (23.2%)

datasets and could show that quantitative pharmacophore models generated by our methods generalise well to many different datasets even without performing hyperparameter optimisation. The trained models achieved a mean RMSE of 0.61 during CV over > 250 datasets. The datasets used for CV resemble sizes typically encountered in SAR settings by medicinal chemists. We could also demonstrate the robustness of our algorithm which is insensitive to small perturbations during training by achieving small variance in RMSE over fivefold CV. Furthermore, on more than 90% of datasets, the generated quantitative pharmacophore models outperformed

Table 3 Target list used for model validation

Target	UniProt ID	CHEMBL ID
Alpha-2a adrenergic receptor	P08913	CHEMBL1867
Tyrosine-protein kinase ABL	P00519	CHEMBL1862
Acetylcholinesterase	P22303	CHEMBL220
Androgen receptor	P10275	CHEMBL1871
Serine/threonine-protein kinase Aurora-A	O14965	CHEMBL4722
Serine/threonine-protein kinase B-raf	P15056	CHEMBL5145
Cannabinoid CB1 receptor	P21554	CHEMBL218
Carbonic anhydrase II	P00918	CHEMBL205
Caspase-3	P42574	CHEMBL2334
Thrombin	P00734	CHEMBL204
Cyclooxygenase-1	P23219	CHEMBL221
Cyclooxygenase-2	P35354	CHEMBL230
Dihydrofolate reductase	P00374	CHEMBL202
Dopamin D2 receptor	P14416	CHEMBL217
Norepinephrine transporter	P23975	CHEMBL222
Epidermal growth factor receptor erbB1	P00533	CHEMBL203
Estrogen receptor alpha	P03372	CHEMBL206
Glucocorticoid receptor	P04150	CHEMBL2034
Glycogen synthase kinase-3 beta	P49841	CHEMBL262
HERG	Q12809	CHEMBL240
Tyrosine-protein kinase JAK2	O60674	CHEMBL2971
Tyrosine-protein kinase LCK	P06239	CHEMBL258
Monoamine oxidase A	P21397	CHEMBL1951
Mu opioid receptor	P35372	CHEMBL233
Vanilloid receptor	Q8NER1	CHEMBL4794

Table 4 Hyperparameters of trained quantitative pharmacophore model on datasets from Debnath et al.

Parameter	Value
Fuzzy	True
Modeltype	RandomForest
Threshold	1
Weighttype	distance
maxDepth (of ML-Model)	3
nEstimators (of ML-Model)	10

Table 5 Predictions on test set of quantitative pharmacophore model

Index	pIC50 exp	pIC50 pred	Index	pIC50 exp	pIC50 pred
0	5.64	7.13	35	8.57	8.10
1	6.6	6.66	37	7.74	7.02
2	6.07	6.75	38	6	6.78
3	6.43	7.24	42	8.17	7.86
4	6.92	6.94	43	6.3	6.22
5	6.52	6.77	44	5.47	7.71
6	6.56	6.60	45	7.17	6.13
7	7.16	8.11	47	6.36	6.08
8	7.77	7.59	48	8.44	7.06
9	7.72	8.14	49	8.15	8.11
10	6	7.36	52	6.85	8.34
11	5.72	7.70	53	8.23	8.11
12	8.09	6.75	54	5.47	6.28
13	8.21	8.04	55	8.37	8.02
14	7.44	6.33	59	7.82	7.63
15	7.48	6.75	61	8.21	7.70
18	8.39	8.34	62	5.57	6.24
19	6.82	7.01	63	8.5	8.28
20	8	8.11	64	8.21	8.34
21	6.17	8.04	65	5.89	7.18
22	7.36	6.75	66	5.6	6.06
24	8.3	8.08	67	8.37	8.34
25	8.55	7.51	68	6	6.16
26	8.06	6.20	69	6.38	6.55
27	8.28	8.13	70	7.24	6.88
28	5.92	6.22	71	7.85	8.22
30	8.26	8.04	72	8.34	8.28
32	8.17	8.04	76	5.14	6.65
34	5.28	6.16			

Table 6 Predictions of quantitative model training set

Index	pIC50 exp	pIC50 pred	Index	pIC50 exp	pIC50 pred
16	7.82	7.48	46	8.29	7.76
17	7.57	7.25	50	5.41	6.38
23	6.09	5.98	51	6.23	6.46
29	7.70	7.53	56	6.07	6.19
31	8.33	7.46	57	6.00	6.23
33	8.00	7.71	58	6.15	6.47
36	4.51	6.78	60	8.40	8.25
39	5.21	6.10	73	5.59	6.65
40	8.43	8.10	74	8.70	8.42
41	8.42	8.34	75	6.36	6.22

tested baselines, thus making our method a reasonable first approach for any researcher looking to get quantitative SAR insights on his data.

Appendix

For a list of target-proteins used for CV studies see Table 3.

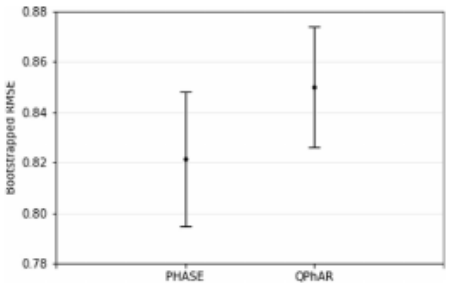

Fig. 5 RMSE values of retrospective bootstrapping including 95% confidence intervals

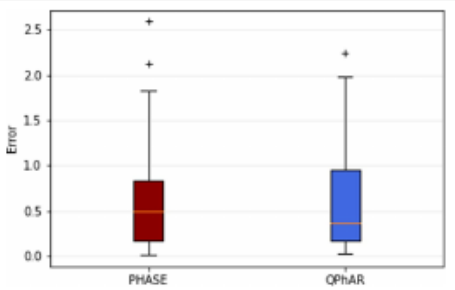

Fig. 6 Error distribution of predictions from PHASE and QPhAR

Quantitative pharmacophore model applied on the dataset from Debnath et al.

A quantitative pharmacophore model was trained on the same dataset as by Dixon et al. for the PHASE algorithm. The hyper-parameters of the best model were as follows (Table 4):

The predictions of the model on the training and test set can be found in the following Tables 5 and 6, respectively.

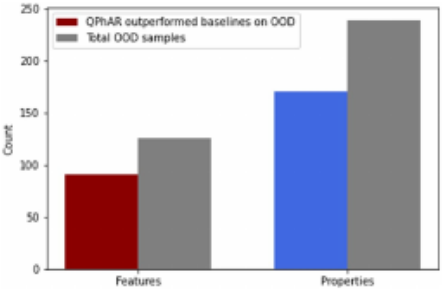

Fig. 7 Number of times QPhAR outperformed baselines for out-of-domain (OOD) samples (80–20 cv split)

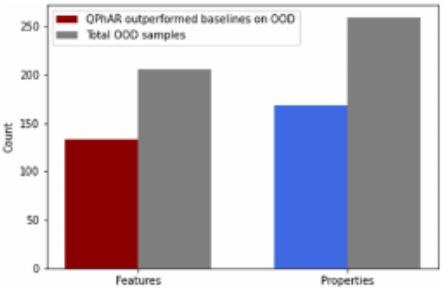

Fig. 8 Number of times QPhAR outperformed baselines for out-of-domain (OOD) samples (20–80 cv split)

Metadata such as dataset sizes, span of activity values, etc. for all datasets used for CV is provided in Additional file 1 (CV 20–80 split) and Additional file 2 (CV 80–20 split).

PHASE vs QPhAR additional plots

Figures 5 and 6 display the bootstrapped confidence intervals as well as the error distributions of the predictions from the PHASE and QPhAR models.

Cross-validation additional plots

Figures 7 and 8 display the frequency of the quant. pharmacophore models outperforming OOD predictions of the baselines.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-021-00537-9.

Additional file 1. Metadata of all datasets used for CV studies with 20–80 split.

Additional file 2. Metadata of all datasets used for CV studies with 80–20 split.

Acknowledgements
We acknowledge the NeuroDeRisk consortium for supporting this project.

Authors' contributions
The method was developed, implemented and validated by SK. The paper was jointly written by TS, TL and SK. All authors read and approved the final manuscript.

Funding
This work was funded by the NeuroDeRisk project [29]. The NeuroDeRisk project has received funding from the Innovative Medicines Initiative 2 [30] Joint Undertaking under Grant Agreement No. 821528. This Joint Undertaking receives support from the European Union's Horizon 2020 [31] research and innovation programme and EFPIA [32].

Availability of data and materials
The datasets supporting the conclusions of this article are available in the Github repository https://github.com/StefanKohlbacher/QuantPharmacophore. Metadata of all datasets used for CV studies are provided in Additional File 1 and Additional File 2 for CV-split 20-80 and CV-split 80-20, respectively.

Declarations

Competing interests
The authors declare no competing interests.

Received: 14 April 2021 Accepted: 21 July 2021
Published online: 09 August 2021

References

1. Hansch C, Maloney PP, Fujita T, Muir RM (1962) Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients. Nature 194:178–180. https://doi.org/10.1038/194178b0
2. Lo Y-C, Rensi SE, Torng W, Altman RB (2018) Machine learning in chemoinformatics and drug discovery. Drug Discov Today 23:1538–1546. https://doi.org/10.1016/j.drudis.2018.05.010
3. Tetko IV, Engkvist O (2020) From big data to artificial intelligence: chemoinformatics meets new challenges. J Cheminformatics 12:74. https://doi.org/10.1186/s13321-020-00475-y
4. Hong H, Chen M, Ng HW, Tong W QSAR Models at the US FDA/NCTR. In: In Silico Methods for Predicting Drug Toxicity; Benfenati, E, Ed; Springer Science+Business Media New York. 2016
5. Rogers D, Hahn M. Extended-Connectivity Fingerprints. J Chem Inf Model. 2010; 50(5):742-754.
6. Golbraikh A, Tropsha A (2003) QSAR modeling using chirality descriptors derived from molecular topology. J Chem Inf Comput Sci 43:144–154. https://doi.org/10.1021/ci025516b
7. Cramer RD, Patterson DE, Bunce JD (1988) Comparative molecular field analysis (CoMFA). 1. Effect of shape on binding of steroids to carrier proteins. J Am Chem Soc 110:5959–5967. https://doi.org/10.1021/ja00226a005
8. Dixon SL, Smondyrev AM, Knoll EH et al (2006) PHASE: a new engine for pharmacophore perception, 3D QSAR model development, and 3D database screening: 1. Methodology and preliminary results. J Comput Aided Mol Des 20:647–671. https://doi.org/10.1007/s10822-006-9087-6
9. Schrödinger [Internet]. [cited 2021 Mar 25]. Available from: https://www.schrodinger.com/
10. Kramer C, Dahl G, Tyrchan C, Ulander J (2016) A comprehensive company database analysis of biological assay variability. Drug Discov Today 21:1213–1221. https://doi.org/10.1016/j.drudis.2016.03.015
11. Hu Y, Stumpfe D, Bajorath J (2017) Recent Advances in Scaffold Hopping. J Med Chem 60:1238–1246. https://doi.org/10.1021/acs.jmedchem.6b01437
12. Li H, Sutter J, Hoffman R. Hypogen: An Automated System for Generating 3D Predictive Pharmacophore Models. In: Pharmacophore Perception, Development and Use in Drug Design; Guner, O, Ed; International University Line: La Jolla, CA,. 2000
13. BIOVIA-Scientific Enterprise Software for Chemical Research, Material Science R&D [Internet]. [cited 2021 Mar 25]. Available from: https://www.3dsbiovia.com/.
14. 3D Design & Engineering Software - Dassault Systèmes®[Internet]. [cited 2021 Mar 25]. Available from: https://www.3ds.com/
15. Debnath AK (2002) Pharmacophore mapping of a Series of 2,4-diamino-5-deazapteridine inhibitors of *Mycobacterium* avium complex dihydrofolate reductase. J Med Chem 45:41–53. https://doi.org/10.1021/jm010360c
16. Poli G, Seidel T, Langer T (2018) Conformational sampling of small molecules with iCon: performance assessment in comparison with OMEGA. Front Chem 6. https://doi.org/10.3389/fchem.2018.00229
17. Wolber G, Langer T (2005) LigandScout: 3-D pharmacophores derived from protein-bound ligands and their use as virtual screening filters. J Chem Inf Model 45:160–169. https://doi.org/10.1021/ci049885e
18. Cortés-Ciriano I, Škuta C, Bender A, Svozil D (2020) QSAR-derived affinity fingerprints (part 2): modeling performance for potency prediction. J Cheminformatics 12:41. https://doi.org/10.1186/s13321-020-00444-5
19. Nowotka M, Gaulton A, Mnedez D, Bento A. P, Hersey A, Leach A. (2017) Using ChEMBL web services for building applications and data processing workflows relevant t. o drug discovery. Expert Opin Drug Discov. 12(8):757-767. https://doi.org/10.1080/17460441.2017.1339032
20. Xu L, Hu O, Guo Y et al (2018) Representative splitting cross validation. Chemom Intell Lab Syst 183:29–35. https://doi.org/10.1016/j.chemolab.2018.10.008
21. Viswanadhan VN, Ghose AK, Revankar GR, Robins RK (1989) Atomic physicochemical parameters for three dimensional structure directed quantitative structure-activity relationships. 4. Additional parameters for hydrophobic and dispersive interactions and their application for an automated superposition of certain naturally occurring nucleoside antibiotics. J Chem Inf Comput Sci 29:163–172. https://doi.org/10.1021/ci00063a006
22. Prasanna S, Doerksen R (2009) Topological polar surface area: a useful descriptor in 2D-QSAR. Curr Med Chem 16:21–41. https://doi.org/10.2174/092986709787002817
23. Jaworska J, Nikolova-Jeliazkova N, Aldenberg T (2005) QSAR applicability domain estimation by projection of the training set in descriptor space: a review. Altern Lab Anim 33:445–459. https://doi.org/10.1177/026119290503300508
24. Breiman L (2001) Random forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324
25. Seidel T (2021) Chemical data processing toolkit, GitHub repository [Internet]. [cited 2021 Mar 19] Available from: https://github.com/aglanger/CDPKit.
26. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
27. Tropsha A, Golbraikh A (2007) Predictive QSAR modeling workflow, model applicability domains, and virtual screening. Curr Pharm Des 13:3494–3504. https://doi.org/10.2174/138161207782794257
28. Tropsha A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inform 29:476–488. https://doi.org/10.1002/minf.201000061
29. NeuroDeRisk (2019) [Internet]. [cited 2021 Mar 17]. Available from: https://neuroderisk.eu/.

30. IMI Innovative Medicines Initiative. [Internet]. [cited 2021 Apr 12]. http://www.imi.europa.eu/.
31. Horizon 2020. [Internet]. Horizon 202 - European Commision. [cited 2021 Mar 17]. Available from: https://ec.europa.eu/programmes/horizon2020/en
32. EFPIA Homepage [Internet]. [cited 2021 Mar 17]. Available from: https://www.efpia.eu/

QPHAR: quantitative pharmacophore activity relationship: method and validation

Stefan M. Kohlbacher, Thierry Langer and Thomas Seidel

Supplementary Information

For a list of target-proteins used for CV studies see Table 3.

Target	UniProt ID	CHEMBL ID
Alpha-2a adrenergic receptor	P08913	CHEMBL1867
Tyrosine-protein kinase ABL	P00519	CHEMBL1862
Acetylcholinesterase	P22303	CHEMBL220
Androgen receptor	P10275	CHEMBL1871
Serine/threonine-protein kinase Aurora-A	O14965	CHEMBL4722
Serine/threonine-protein kinase B-raf	P15056	CHEMBL5145
Cannabinoid CB1 receptor	P21554	CHEMBL218
Carbonic anhydrase II	P00918	CHEMBL205
Caspase-3	P42574	CHEMBL2334
Thrombin	P00734	CHEMBL204
Cyclooxygenase-1	P23219	CHEMBL221
Cyclooxygenase-2	P35354	CHEMBL230
Dihydrofolate reductase	P00374	CHEMBL202
Dopamin D2 receptor	P14416	CHEMBL217
Norepinephrine transporter	P23975	CHEMBL222
Epidermal growth factor receptor erbB1	P00533	CHEMBL203
Estrogen receptor alpha	P03372	CHEMBL206
Glucocorticoid receptor	P04150	CHEMBL2034
Glycogen synthase kinase-3 beta	P49841	CHEMBL262
HERG	Q12809	CHEMBL240

Tyrosine-protein kinase JAK2	O60674	CHEMBL2971
Tyrosine-protein kinase LCK	P06239	CHEMBL258
Monoamine oxidase A	P21397	CHEMBL1951
Mu opioid receptor	P35372	CHEMBL233
Vanilloid receptor	Q8NER1	CHEMBL4794

Table 3: Target list used for model validation

Quantitative pharmacophore model applied on the dataset from Debnath et al.

A quantitative pharmacophore model was trained on the same dataset as by Dixon et al. for the PHASE algorithm. The hyper-parameters of the best model were as follows (Table 4):

Parameter	**Value**
Fuzzy	True
Modeltype	RandomForest
Threshold	1
Weighttype	distance
maxDepth (of ML-Model)	3
nEstimators (of ML-Model)	10

Table 4: Hyperparameters of trained quantitative pharmacophore model on datasets from Debnath et al.

The predictions of the model on training and test set can be found in the following Tables 5 and 6, respectively.

Index	**pIC_{50} exp**	**pIC_{50} pred**	**Index**	**pIC_{50} exp**	**pIC_{50} pred**
0	5.64	7.13	35	8.57	8.10
1	6.6	6.66	37	7.74	7.02
2	6.07	6.75	38	6	6.78
3	6.43	7.24	42	8.17	7.86
4	6.92	6.94	43	6.3	6.22
5	6.52	6.77	44	5.47	7.71

6	6.56	6.60	45	7.17	6.13
7	7.16	8.11	47	6.36	6.08
8	7.77	7.59	48	8.44	7.06
9	7.72	8.14	49	8.15	8.11
10	6	7.36	52	6.85	8.34
11	5.72	7.70	53	8.23	8.11
12	8.09	6.75	54	5.47	6.28
13	8.21	8.04	55	8.37	8.02
14	7.44	6.33	59	7.82	7.63
15	7.48	6.75	61	8.21	7.70
18	8.39	8.34	62	5.57	6.24
19	6.82	7.01	63	8.5	8.28
20	8	8.11	64	8.21	8.34
21	6.17	8.04	65	5.89	7.18
22	7.36	6.75	66	5.6	6.06
24	8.3	8.08	67	8.37	8.34
25	8.55	7.51	68	6	6.16
26	8.06	6.20	69	6.38	6.55
27	8.28	8.13	70	7.24	6.88
28	5.92	6.22	71	7.85	8.22
30	8.26	8.04	72	8.34	8.28
32	8.17	8.04	76	5.14	6.65
34	5.28	6.16			

Table 5: Predictions on test set of quantitative pharmacophore model

Index	pIC_{50} exp	pIC_{50} pred	Index	pIC_{50} exp	pIC_{50} pred
16	7.82	7.48	46	8.29	7.76
17	7.57	7.25	50	5.41	6.38
23	6.09	5.98	51	6.23	6.46
29	7.70	7.53	56	6.07	6.19
31	8.33	7.46	57	6.00	6.23
33	8.00	7.71	58	6.15	6.47
36	4.51	6.78	60	8.40	8.25
39	5.21	6.10	73	5.59	6.65
40	8.43	8.10	74	8.70	8.42
41	8.42	8.34	75	6.36	6.22

Table 6: Predictions of quantitative model training set

Metadata such as dataset sizes, span of activity values, etc. for all datasets used for CV is provided in Additional file 1 (CV 20-80 split) and Additional file 2 (CV 80-20 split).

PHASE vs QPhAR additional plots

Figures 5 and 6 display the bootstrapped confidence intervals as well as the error distributions of the predictions from the PHASE and QPhAR models.

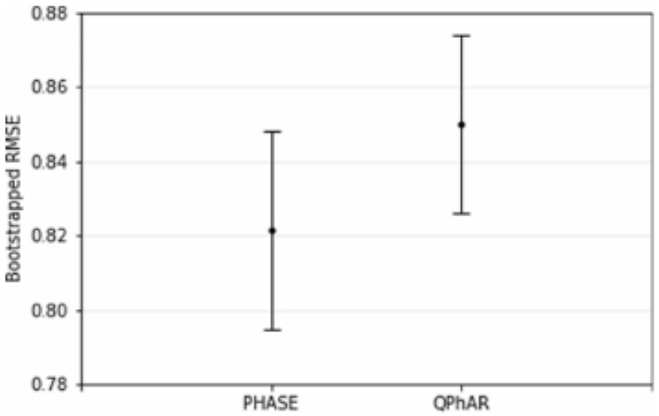

Figure 5: RMSE values of retrospective bootstrapping including 95% confidence intervals

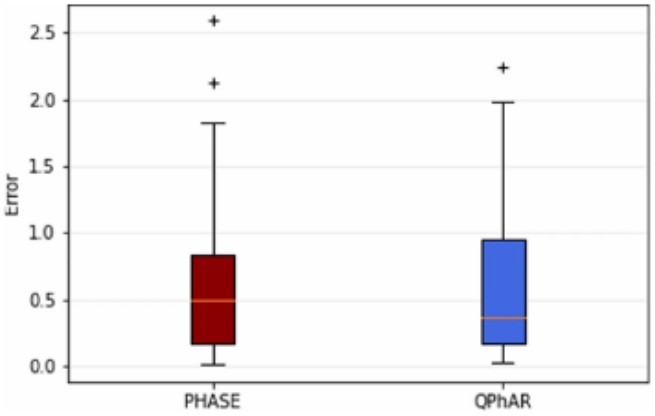

Figure 6: Error distribution of predictions from PHASE and QPhAR

Cross-validation additional plots

Figures 7 and 8 display the frequency of the quant. pharmacophores models outperforming OOD predictions of the baselines.

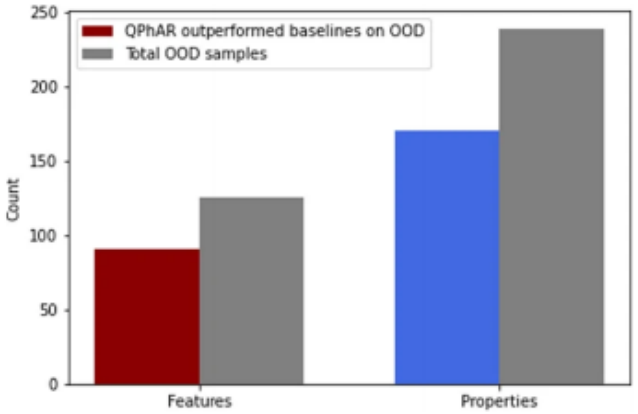

Figure 7: Number of times QPhAR outperformed baselines for out-of-domain (OOD) samples (80-20 cv split)

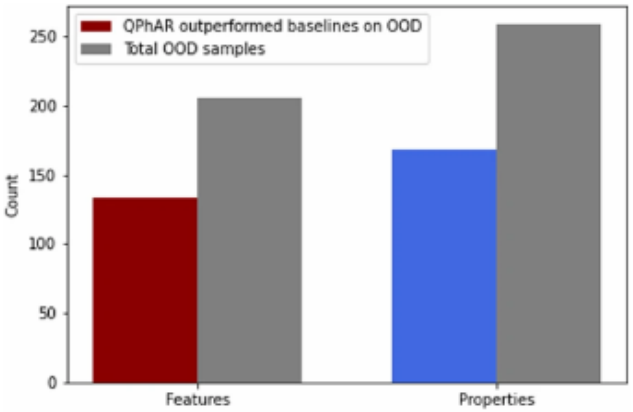

Figure 8: Number of times QPhAR outperformed baselines for out-of-domain (OOD) samples (20-80 cv split)

A new set of KNIME nodes implementing the QPhAR algorithm

Stefan M. Kohlbacher, Thomas Seidel, Gökhan Ibis, Christian Permann, Sharon Bryant, Thierry Langer

Kohlbacher SM, Ibis G, Permann C, Bryant S, Langer T, Seidel T(2022) A new set of KNIME nodes implementing the QPhAR algorithm. Mol Inform. Under Review

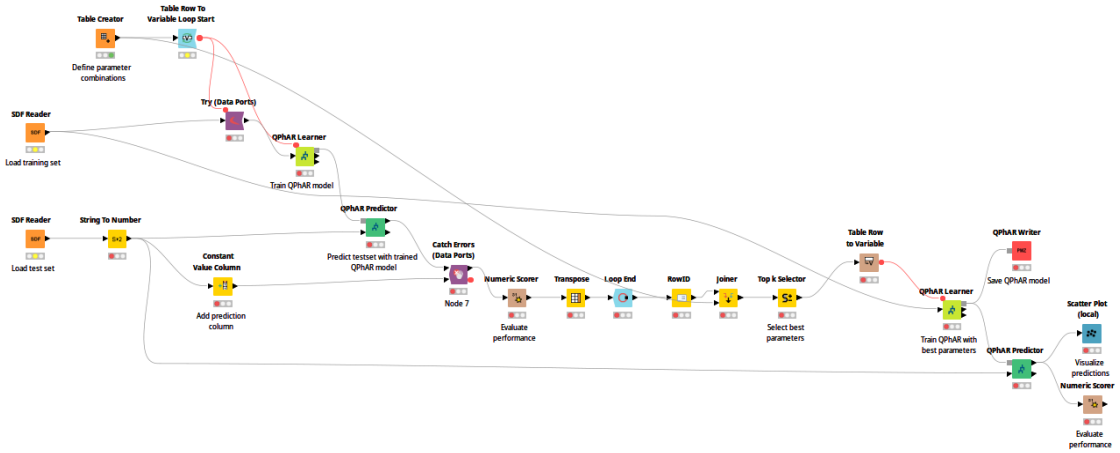

Image 7: Sample KNIME workflow for training a QPhAR model[94].

S. Kohlbacher has developed the KNIME nodes, devised the KNIME workflow, trained the QPhAR models, and wrote the manuscript. G. Ibis and C. Permann contributed to the development of the KNIME nodes. T. Seidel, Sharon Bryant, and T. Langer have supervised the work and revised the manuscript.

molecular informatics
models – molecules – systems

A new set of KNIME nodes implementing the QPhAR algorithm

Journal:	*Molecular Informatics*
Manuscript ID	minf.202200245
Wiley - Manuscript type:	Application Note
Date Submitted by the Author:	13-Oct-2022
Complete List of Authors:	Kohlbacher, Stefan; University of Vienna Ibis, Gökhan; InteLigand GmbH Permann, Christian; University of Vienna; InteLigand GmbH Bryant, Sharon; InteLigand GmbH Langer, Thierry; University of Vienna; InteLigand GmbH Seidel, Thomas; University of Vienna
Keywords:	3D-QSAR, Pharmacophore
Keywords:	pharmacophore modeling, QPhAR, KNIME

molecular informatics

DOI: 10.1002/minf.200((full DOI will be filled in by the editorial staff))

A new set of KNIME nodes implementing the QPhAR algorithm

Stefan M. Kohlbacher[a], Gökhan Ibis[b], Christian Permann[a, b], Sharon Bryant[b], Thierry Langer[a, b] and Thomas Seidel*[a]

Abstract: Dissemination of novel research methods, especially in the form of chemoinformatics software, depends heavily on their ease of applicability for non-expert users with only a little or no programming skills and knowledge in computer science. Visual programming has become widely popular over the last few years, also enabling researchers without in-depth programming skills to develop tailored data processing pipelines using elements from a repository of predefined standard procedures. In this work, we present the development of a set of nodes for the KNIME platform implementing the QPhAR algorithm. We show how the developed KNIME nodes can be included in a typical workflow for biological activity prediction. Furthermore, we present best-practice guidelines that should be followed to obtain high-quality QPhAR models. Finally, we show a typical workflow to train and optimise a QPhAR model in KNIME for a set of given input compounds, applying the discussed best practices.

Keywords: pharmacophores, pharmacophore modeling, QPhAR, KNIME, NeuroDeRisk

Nowadays, *in-silico* methods are an indispensable part of the modern drug discovery toolchain and have shaped the field of computer-aided drug design (CADD). The software tools themselves, however, are often unintuitive and require considerable knowledge in programming or the computer science domain. This presents a huge entry barrier for many lab researchers, preventing them from benefiting from the latest research in chemoinformatics, bioinformatics, or similar domains. In addition to hard-to-use tools, the latest chemoinformatic methods are hardly ever released via a ready-to-use software package.

The Konstanz Information Miner (KNIME)[1] is a platform which aims to reduce a lot of these barriers and facilitate the use of, amongst others, advanced chemoinformatics tools. It provides the user with a graphical interface to build and run complex workflows and pipelines without the need for programming skills. So-called nodes represent clearly defined processing units and are added to a KNIME workflow by simple drag-and-drop. Nodes are available from various software vendors and from the built-in repository maintained by the open source community. Furthermore, the KNIME developers guide[2] allows any researcher with some programming knowledge to develop his own nodes and make them available for broader use by the scientific community if desired. We have previously published a novel method called QPhAR[3] that perceives and models quantitative relationships between biological activities and pharmacophores. The tool has been made available via a docker image that can be retrieved from the author's Github page[4]. Acknowledging the above-made argument that modern chemoinformatic tools lack user-friendliness, we developed, in collaboration with members of the NeuroDeRisk project[5], various KNIME nodes implementing the QPhAR algorithm.

The underlying principle of the QPhAR algorithm is the principle of QSAR[6], which aims to mathematically relate molecular structures, or binding motives, to a chosen endpoint, often the biological activity. In essence, QSAR models are often regression or simple machine learning models. Therefore, they should be treated as such, demanding the datasets to meet certain requirements. While this is common practice in computer sciences developing machine learning algorithms, it is often ignored in the domain of life sciences applying QSAR algorithms.

In this publication, we will present a new set of KNIME nodes implementing the QPhAR algorithm as well as dedicated IO nodes for saving and restoring the generated models. In addition, we will present and discuss best practices for training QSAR models, with a special focus on the QPhAR algorithm. Finally, we provide a typical workflow to train a QPhAR model in KNIME, applying the discussed best practices.

Experimental Section (or Computational Methods)

QPhAR goes KNIME

The original implementation of QPhAR is comprised of a set of Python scripts that were made available for interested users in the form of a docker container. Providing scientific open source software this way is quite common for developments resulting from research projects but causes considerable inconveniences, especially for less experienced end-users. To facilitate

[a] *Division of Pharmaceutical Chemistry, Department of Pharmaceutical Sciences, University of Vienna, Josef-Holaubek-Plart 2, 1090, Vienna, Austria*

[b] *Inte:Ligand G.bH, Mariahilferstraße 74B/11, 1070, Vienna*

Supporting information for this article is available on the WWW under http://dx.doi.org/10.1002/minf.((DOI will be filled in by the editorial staff)).

user-friendliness and applicability, the QPhAR algorithm thus has been re-implemented as a collection of KNIME nodes. This collection provides the following nodes and associated functionalities:

- *QPhAR Learner*

The *QPhAR Learner* (Figure 1 **A**) node implements the main logic to train a QPhAR model from a given dataset. It has one input port and requires that the provided table contains a column storing molecules with 3D atom coordinates. In addition, a column containing the true target values of each compound must be present. The target value data type can be a string or a numeric type that can be converted to a floating point value by the *QPhAR Learner* node at runtime. These columns will be auto-detected, whereas the first column of the corresponding type is selected if multiple columns of the same type exist. The user can also set the columns manually in the node configuration dialogue. The three output ports provide the following data (from top to bottom): i) the trained QPhAR model, ii) the given input data, along with two additional columns containing the predicted target values and a boolean value, indicating whether the compound is within the model's applicability domain or outside, and iii) a table containing all compounds outside the model's applicability domain if the box "Remove out of domain samples" in the configuration dialogue has been checked; otherwise, an empty table will be returned.

The node offers multiple settings specific to the QPhAR algorithm (please refer to the original publication for a description of the QPhAR parameters) and options controlling the model's training. Among the already mentioned general settings ("Activity column" and "Remove out of domain samples"), the "Training size" can be set. By default, this value is set to 0.8, applying internal cross-validation with an 80%-20% training-validation split of the given dataset. This value can be set to 1 if the user prefers to handle cross-validation externally.

- *QPhAR Predictor*

The *QPhAR Predictor* (Figure 1 **B**)is the associated counterpart of the *QPhAR Leaner node*. The *QPhAR Predictor* node has two mandatory input ports. The top input port expects data of a trained QPhAR model that are either provided by a preceding *QPhAR Learner* or *QPhAR Reader* node. The bottom input expects a table that contains a column storing molecules with 3D atom coordinates. As for the *QPhAR Learner* node, the molecule column will be auto-detected. The two output ports are equivalent to the output ports ii) and iii) of the *QPhAR Learner* node: The first port emits the input table with two additional columns containing the model's predictions and the applicability domain indicator. The second port, if desired, provides a table containing all compounds outside the model's applicability domain.

- *QPhAR Reader*

The *QPhAR Reader* node (Figure 1 **C**) implements functionality to load a previously stored QPhAR model from disk. Only models saved by the *QPhAR Writer* node are supported. The node provides the loaded model on its output port which can then be connected to a *QPhAR Predictor* node.

- *QPhAR Writer*

The *QPhAR Writer* node (Figure 1 **D**) is the counterpart of the *QPhAR Reader* node and is used to write a previously trained model to disk. At its input port, the node expects model data as output by the *QPhAR Learner* node.

Figure 1. The set of implemented QPhAR KNIME nodes.

Table 1. Validation metrics of the QPhAR algorithm in KNIME and its original implementation in Python.

Dataset	KNIME		Python	
	RMSE	R^2	RMSE	R^2
Ecel[7]	0.79	0.55	0.41	0.88
Garg[8]	0.75	0.28	0.56	0.67
Krovat[9]	0.82	0.33	0.70	0.50
Mal[10]	0.54	0.36	0.44	0.58

Table 1 contains data about the validated performance of the KNIME nodes compared to the Python models. The KNIME nodes and models were validated on datasets which have been part of extensive validation in the past to ensure a relevant comparison and result. It is acknowledged that the performance of the KNIME nodes is slightly below the performance of the Python models. These differences were investigated and can be attributed to the utilised libraries implementing the random forest model as well as the pharmacophores being generated (the initial QPhAR implementation was done in Python and the KNIME nodes were developed in Java). However, the models are still performing well, and proper validation and tests were carried out to ensure qualitative equality between the implementations.

QPhAR 101 and best practices

User-friendly chemoinformatic tools usually provide reasonable default values for their configuration parameters in order to enable a quick start for users without in-depth knowledge of the applied method. Based on various validation runs, default values for QPhAR parameters were chosen to provide the best initial guess

molecular informatics

for successfully training a model. However, even though training a model with the default values is likely to yield good results, the results will not be the optimum the user could achieve by careful tweaking of the parameters. Therefore, we provide a guide of best practices for training a QPhAR model within KNIME to facilitate the training of a QPhAR model and optimise the obtained results. First, we briefly discuss a few requirements that should ideally be met to train a QPhAR model. Then we will go over various validation metrics that can be used to assess the quality of a QPhAR model. Finally, hyper-parameter optimisation will be discussed, as this is an essential topic for any machine learning model and thus also of relevance for QPhAR.

The following best practices are not only limited to QPhAR but should be applied to any QSAR study. Unfortunately, this is often overlooked. Theoretically, a regression model can be trained for any kind of target value. However, within the context of biological activity prediction, representing one of the most frequent use cases, not every set of activity values will yield a useful model. Given that the experimental error is often as large as 0.5 log units of activity (Ki or IC50), the dataset used for training a QPhAR model should at least span a range of 3 log units of activity to get a meaningful model. Otherwise, the model's statistics might indicate a well-trained model, whereas actually the experimental noise has been modelled.

Furthermore, the distribution of activity values in the training set should be as homogeneous as possible. Otherwise, the task turns into outlier prediction rather than classical regression, which is an entirely different problem. A rough estimation is whether clusters of data points can be identified visually when plotting the sorted activity values of the training set.

Datasets of molecular activity often span multiple orders of magnitude. When calculating the loss, applying these raw activity values as target values for training a machine learning problem can become an issue. For example, the Mean-Squared-Error (MSE) will emphasise samples with larger absolute errors, even if the error is relatively small compared to other samples in the dataset. Transforming these values into the same domain - for example, by calculating the logarithm - will alleviate such problems. The QPhAR algorithm has been designed with these issues in mind, and to obtain the best results, it is strongly recommended to perform such transformations before training a QPhAR model.

Many different types of pharmacophore features can be defined[11], some software even allows the definition of custom pharmacophore feature types. QPhAR only uses the following feature types: H-bond acceptors/donors, hydrophobic groups and aromatic rings, and positive and negative ionisable groups. Therefore, samples containing other chemical feature types, e.g. groups allowing the complexation of metals, should be removed from the training set. These samples are likely to add noise to the training of the QPhAR model without contributing information to the model. Even though these samples will not raise an error, critical information will be missing at training.

Finally, splitting the dataset into a training-, validation-, and test set can be considered a standard procedure carried out in the preparatory phase of machine learning tasks. The "QPhAR Learner" already provides some functionality to split the training set into a training and validation set. The user can determine the ratio with the parameter "training size". Setting this parameter to 1 will not split the dataset, and the model will be trained on the entire dataset given. However, unless handled externally by the user, it is highly recommended to keep this option enabled.

Hyper-parameter optimisation is a bloody double-edged sword. On the one hand, it might drastically improve a machine learning model's performance; on the other hand, if not done correctly, it can quickly lead to model overfitting effects. Therefore, QPhAR only allows users to optimise a selected subset of parameters. In addition, the range of possible values is restricted to avoid the temptation of overfitting. Therefore, it is recommended to optimise the following parameters (same name as in the "QPhAR Learner" settings tab) with values within the specified allowed range:

- "Nr. trees": Number of independent trees contained in the random forest. Values in the interval [1, 1000] are possible, but values from the interval [20, 50] are recommended. For really large datasets, the user might increase the number of trees.
- "Max. depth": Maximal depth for each tree of the random forest model. Values within the interval [1, 100] are possible, whereas values in [3, 5] are recommended.
- "Pharmacophore feature merging threshold": Distance cutoff for the clustering algorithm of the QPhAR algorithm. For clustering, a hierarchical clustering algorithm with maximal distance is used. The default value is set to 4. Values within [0.5, 20] are possible, although values within [2, 5] are advised.

When optimising the model's hyper-parameters, it is recommended to set the "training size" parameter to 1. In that case, the "QPhAR Predictor" node can be used to predict the test set and calculate the final performance from these predictions.

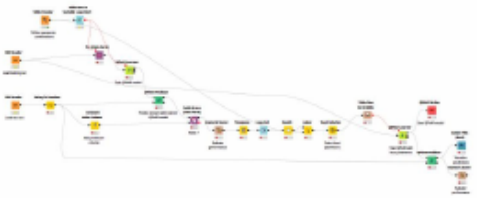

Figure 2. KNIME workflow for training a QPhAR mode, including hyper-parameter optimisation.

molecular informatics

These best practices have been incorporated into an example KNIME workflow (Figure 2) for training a QPhAR model and optimising its hyper-parameters. The workflow and the QPhAR nodes are available for download (https://github.com/StefanKohlbacher/qphar-knime-nodes) and can be used as a starting point for the training of QPhAR models that follow best practices in order to obtain the best results possible.

Furthermore, within the scope of the NeuroDeRisk project, the developed KNIME nodes were applied to build various models for safety and toxicity assessment of the GABA-A transporter. The models are being disseminated and available along with the other data in the Github repository.

Acknowledgements

We acknowledge the NeuroDeRisk consortium for supporting this project.

Conflict of Interest

TL is the co-founder and co-owner of the company InteLigand. The other authors declare no competing interests.

Funding

This work was funded by the NeuroDeRisk project. The NeuroDeRisk project has received funding from the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 821528. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation program, and EFPIA.

References

[1] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, B. Wiswedel, in *Data Anal. Mach. Learn. Appl.* (Eds.: C. Preisach, H. Burkhardt, L. Schmidt-Thieme, R. Decker), Springer, Berlin, Heidelberg, **2008**, pp. 319–326.
[2] "Create a New Java based KNIME Extension," can be found under https://docs.knime.com/latest/analytics_platform_new_node_quickstart_guide/index.html#_introduction, **n.d.**
[3] S. M. Kohlbacher, T. Langer, T. Seidel, *J. Cheminformatics* **2021**, *13*, 57.
[4] S. Kohlbacher, **2022**.
[5] NeuroDeRisk, "NeuroDeRisk - Neurotoxicity De-Risking in Preclinical Drug Discovery," can be found under https://neuroderisk.eu/, **2019**.
[6] C. Hansch, P. P. Maloney, T. Fujita, R. M. Muir, *Nature* **1962**, *194*, 178–180.
[7] A. Ece, F. Sevin, *Med. Chem. Res.* **2013**, *22*, 5832–5843.
[8] D. Garg, T. Gandhi, C. Gopi Mohan, *J. Mol. Graph. Model.* **2008**, *26*, 966–976.
[9] E. M. Krovat, T. Langer, *J. Med. Chem.* **2003**, *46*, 716–726.
[10] Y. Ma, H.-L. Li, X.-B. Chen, W.-Y. Jin, H. Zhou, Y. Ma, R.-L. Wang, *Comput. Biol. Chem.* **2018**, *73*, 1–12.
[11] G. Wolber, T. Seidel, F. Bendix, T. Langer, *Drug Discov. Today* **2008**, *13*, 23–29.

Received: ((will be filled in by the editorial staff))
Accepted: ((will be filled in by the editorial staff))
Published online: ((will be filled in by the editorial staff))

Applications of the novel quantitative pharmacophore activity relationship method QPhAR in virtual screening and lead-optimization

Stefan M. Kohlbacher, Matthias Schmid, Thomas Seidel and Thierry Langer

Kohlbacher SM, Schmid M, Seidel T, Langer T (2022) Applications of the Novel Quantitative Pharmacophore Activity Relationship Method QPhAR in Virtual Screening and Lead-Optimization. Pharmaceuticals 15:1122. https://doi.org/10.3390/ph15091122

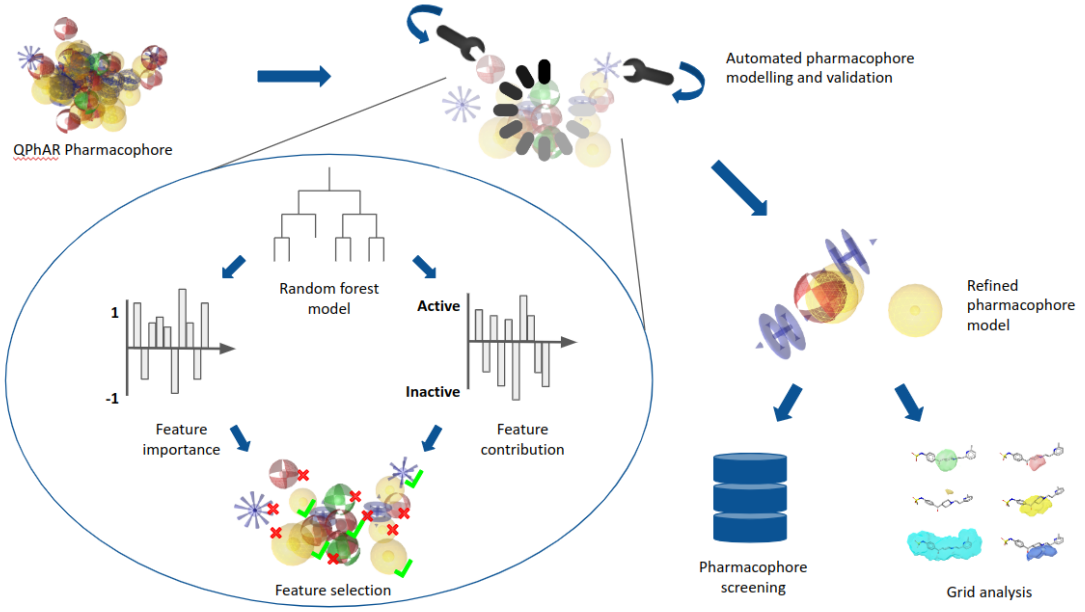

Image 8: Scheme of end-to-end pharmacophore modelling workflow[95]

The initial study has been designed by S. Kohlbacher, T. Seidel, and T. Langer. M. Schmid has collected the QPhAR datasets and trained the corresponding QPhAR models. The algorithm for the extraction of the refined pharmacophore models, as well as the end-to-end workflow of the trained QPhAR models, was developed by S. Kohlbacher. S. Kohlbacher has also developed the method for the 3D activity profiling of lead molecules based on the QPhAR model. S. Kohlbacher carried out the experiments and wrote the manuscript. T. Seidel and T. Langer have supervised the work and have revised the manuscript.

pharmaceuticals

MDPI

Article

Applications of the Novel Quantitative Pharmacophore Activity Relationship Method QPhAR in Virtual Screening and Lead-Optimisation

Stefan Michael Kohlbacher, Matthias Schmid, Thomas Seidel * and Thierry Langer

Division of Pharmaceutical Chemistry, Department of Pharmaceutical Sciences, University of Vienna, Josef-Holaubek-Platz 2, 1090 Vienna, Austria

* Correspondence: thomas.seidel@univie.ac.at; Tel.: +43-1-4277-55051

Abstract: Pharmacophores are an established concept for the modelling of ligand–receptor interactions based on the abstract representations of stereoelectronic molecular features. They became widely popular as filters for the fast virtual screening of large compound libraries. A lot of effort has been put into the development of sophisticated algorithms and strategies to increase the computational efficiency of the screening process. However, hardly any focus has been put on the development of automated procedures that optimise pharmacophores towards higher discriminatory power, which still has to be done manually by a human expert. In the age of machine learning, the researcher has become the decision-maker at the top level, outsourcing analysis tasks and recurrent work to advanced algorithms and automation workflows. Here, we propose an algorithm for the automated selection of features driving pharmacophore model quality using SAR information extracted from validated QPhAR models. By integrating the developed method into an end-to-end workflow, we present a fully automated method that is able to derive best-quality pharmacophores from a given input dataset. Finally, we show how the QPhAR-generated models can be used to guide the researcher with insights regarding (un-)favourable interactions for compounds of interest.

Keywords: pharmacophore; pharmacophore modelling; quantitative pharmacophore; QSAR; machine learning; pharmacophore optimisation; NeuroDeRisk

Citation: Kohlbacher, S.M.; Schmid, M.; Seidel, T.; Langer, T. Applications of the Novel Quantitative Pharmacophore Activity Relationship Method QPhAR in Virtual Screening and Lead-Optimisation. *Pharmaceuticals* **2022**, *15*, 1122. https://doi.org/10.3390/ph15091122

Academic Editor: Osvaldo Andrade Santos-Filho

Received: 5 August 2022
Accepted: 2 September 2022
Published: 8 September 2022

1. Introduction

Pharmacophore modelling was popularised at the turn of the millennium with increasing computational power and its general accessibility for researchers in the field of medicinal chemistry [1–3]. Since then, it has become an integral part of the methodological toolbox for computer-assisted drug discovery and design [4]. In the absence of a crystal structure, ligand-based pharmacophore modelling is often used in combination with the virtual screening of large compound databases in order to identify novel active compounds for a particular target of interest [5]. Even though many drug discovery success stories can be reported [6–8] where pharmacophore-based virtual screening was used as a key technology, the pharmacophore modelling process itself is often tedious, highly complex, error-prone, and relies heavily on the expert knowledge of the researcher. Various unknowns in pharmacophore modelling even often yield completely different results when applying different programs to the same dataset [9,10].

Before the 2000s, Chen et al. [11] proposed a system that analyses a dataset of a few thousand compounds and then generates suggestions for pharmacophore models based on the obtained knowledge. The presented method is a first step toward generating a system that analyses a set of data too complex for humans to fully grasp and present the obtained solutions to the researcher, who merely needs to decide on the best solution. We think machine learning has huge potential in computer-assisted drug discovery to achieve

exactly that; analysing complex data to assist the researcher and offer guidance with the obtained solutions.

Furthermore, Chen et al. pose two arguments contrary to popular heuristics applied in pharmacophore modelling. First, they state that weak or lesser active compounds contain important information for pharmacophore modelling. This argument contrasts the often practised method of selecting a highly active subset of compounds for pharmacophore modelling [5]. Nowadays, this is often considered by adding exclusion volumes to the pharmacophore. The second argument Chen et al. bring forward is that selecting an activity cutoff for active and inactive compounds is highly subjective and not clearly defined. Indeed, the cutoff may depend on factors such as the available dataset, and multiple experts might independently end up with various cutoffs for a certain dataset. Considering these arguments, the logical next step is the generation of pharmacophores from continuous data without the need for arbitrary choosing activity cutoff values.

In addition to automated pharmacophore modelling, scoring and prioritisation of the obtained hits are not possible with the qualitative nature of pharmacophores. Consensus scoring [12] with multiple models is an often applied first step to solving this problem. Another solution is ranking the obtained hits by an external regression model. Considering that most consensus scoring methods are still qualitative in their nature and regression represents a different type of model, a combination of these two would be ideal for ranking the obtained hits. Eventually, this results in a method that prioritises hits with a previously validated pharmacophore model by assigning continuous activity values to the compounds. Combined with an automated approach to generate pharmacophore models from a given dataset containing only a few compounds, a researcher could quickly generate a prioritised list of hits for biological testing in the drug discovery campaign.

In this paper, we present a novel method for automated pharmacophore modelling given a previously trained and validated QPhAR [13] model. We show that it outperforms the commonly applied heuristics for pharmacophore model refinement and can reliably generate a set of three-dimensional (3D) pharmacophores that show high discriminatory power in the virtual screening process. Combined with the training of a QPhAR model, we propose a fully automated workflow for generating a QPhAR model from a set of given compounds, deriving a classification-performance optimised pharmacophore (in the following referred to as 'refined' pharmacophore), using the pharmacophore for the virtual screening of molecule databases, and finally ranking the obtained hits by their predictions made with the QPhAR model. In addition, we highlight a method to visualise the expected changes in the activity of a compound when introducing certain pharmacophore features. The expected activity changes are displayed in a grid around the investigated compound, guiding the researcher with highlighted regions of favourable and unfavourable interactions. The proposed method and workflow aim at the analysis of, for human researchers, usually non-obvious information contained in ligand datasets and the presentation of this information in an easy-to-comprehend way. The expert user can then engage in decision-making based on the presented results of the performed analyses.

2. Results and Discussion

We conducted a case study on the hERG K+ channel using the dataset from Garg et al. [14] and the correspondingly trained QPhAR model. First, we will discuss the process of generating a refined pharmacophore and its comparison against established baseline methods. Second, on the basis of the hERG example, we describe how the information provided by a QPhAR model can be utilized in a fully automated end-to-end pharmacophore modelling workflow. Finally, we will close the discussion with a few examples of how QPhAR can be used to guide a medicinal chemist to further insights after a set of compounds has been selected from an obtained virtual screening hit list.

2.1. Generation of a Refined Pharmacophore for Virtual Screening

For each dataset investigated, we have applied the devised algorithm to extract refined pharmacophore features from the QPhAR model. The pharmacophore can be generated directly from the model without the requirement of additional data. Therefore, all molecules contained in the datasets can be used to evaluate the generated pharmacophore. Nevertheless, it makes sense to keep the training-test split of each dataset for a final validation of the selected models on the test set. Following this strategy, the generated pharmacophores were evaluated on the training set, ranked by their F_β-score and $F_{Specificity}$-score, and the top five models validated on the provided test set. Figure 1 shows the refined pharmacophore model generated for the dataset obtained from Garg et al. [14].

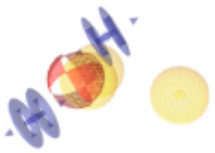

Figure 1. Refined pharmacophore of "Garg et al." [14] dataset.

In contrast to the generation of refined pharmacophores, the generation of shared pharmacophores, the baseline method, requires an input dataset. Shared pharmacophores were chosen as the baseline for two reasons. First, shared feature pharmacophore generation is often employed as the "first-in-line" method when it comes to ligand-based pharmacophore modelling. Second, pharmacophores of highly active compounds are assumed to contain many features of relevance for high compound binding affinity. The baseline models were generated from the n most active compounds in the training set, with n serving as a hyperparameter. These pharmacophores were validated on the training and test set in the same manner. The results and a comparison against the performance of the refined pharmacophores can be found in Table 1.

Table 1. Test performance of the shared pharmacophore baseline models and refined pharmacophores obtained from the corresponding QPhAR models.

Data Source	$F_{Composite}$-Score		QphAR Model Performance	
	Baseline	QphAR	R^2	RMSE
Ece et al. [15]	0.38	0.58	0.88	0.41
Garg et al. [14]	0.00	0.40	0.67	0.56
Ma et al. [16]	0.57	0.73	0.58	0.44
Wang et al. [17]	0.69	0.58	0.56	0.46
Krovat et al. [18]	0.94	0.56	0.50	0.70

The baseline and QphAR-based refined pharmacophores were scored and compared using the $F_{Composite}$-score. The typical metrics used in machine learning, such as accuracy, precision, sensitivity, etc., are not accurately depicting the situation in virtual screening. Scoring pharmacophore models with these metrics would lead to results which might not be considered optimal in this context. Often the objective is to get as many true positives as possible while reducing the number of false positives. The number of false negatives can often be ignored with the reasoning that a missed hit does not consume any resources, whereas false positives will. Accuracy and others are not considering these objectives and put the same emphasis on both numbers. It should be noted that the ROC-AUC score is often used in virtual screening experiments and does reflect the objective much better than accuracy and others. However, due to the ROC-AUC score's non-linearity, we think it

often gives the perception of the results being better than they are. Therefore, we used the F_β-score, $F_{Specificity}$-score, and $F_{Composite}$-score to score the obtained pharmacophore models.

As can be seen in Table 1, the QPhAR-based refined pharmacophores score better than the baseline pharmacophores on the $F_{Composite}$-score, although a dependency on the quality of the QPhAR models can be observed. The lower the performance of the QPhAR model, the less reliable it is in generating a refined pharmacophore. This, however, is not surprising since the quality of the workflow we describe here depends heavily on the trained QPhAR model. Therefore, we advise the user to emphasise training and validating the QPhAR models to increase the model's performance and narrow the confidence interval.

2.2. End-to-End Pharmacophore Modelling

Applying the aforementioned algorithm to generate refined pharmacophores from QPhAR models, we developed a workflow (Figure 2) for fully automated pharmacophore modelling, virtual screening and ranking of the obtained hits. The workflow is completely ligand-based; therefore, only a small set of compounds of ~15–50 ligands with known activity values is required. We will assume IC50 or Ki values here, although theoretically any physicochemical property can be used.

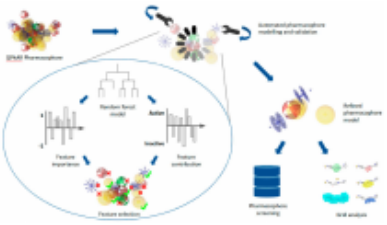

Figure 2. Schematic depiction of end-to-end pharmacophore modelling workflow.

The first step is to prepare and clean a dataset for the target of interest. Here, we use the dataset published by Garg et al. [14] on the widely known hERG K+ channel. The dataset is split into a training and test subset (we adopt the splitting ratio provided in their publication), and a QPhAR model is generated using the training set molecules. The QPhAR model is validated on the before separated test set using cross-validation, leave-one-out analysis, y-scrambling and a paired t-test (results have been published previously [19]). Afterwards, the refined pharmacophore model is generated using the procedure outlined in the methods section. The refined pharmacophore is then validated on the separated test set before being used to screen a database of virtual molecules. We use a filtered version of the Molport database containing ~1.25 million molecules. Since pharmacophore-based virtual screening is only a qualitative method, it is not possible to directly prioritise some compounds from the hit list over others on the basis of particular physicochemical properties of interest (e.g., their IC50 value). Therefore, the next and final step is to score and rank the obtained hit list (14871 molecules, ~1% hit-rate) with the previously trained QPhAR model. The obtained ranked hit list is provided as an SD-file in ascending order of relevance (highest activity value first). The data can be found along with the remaining data in the author's GitHub repository (https://github.com/StefanKohlbacher/qphar-applications).

Even though the entire workflow can be automated from start to end, we recommend including sanity checks at certain key events, such as validating the trained QPhAR model performance and the completed generation of the derived refined pharmacophore. For both steps, we suggest to define key metrics and corresponding values that should be fulfilled before the workflow proceeds.

2.3. Three-Dimensional Pharmacophore Activity Profiling

Finally, the hit-list obtained from the end-to-end pharmacophore modelling workflow will serve as a starting point for medicinal chemists to further optimise compounds in

the hit-to-lead phase of the drug discovery pipeline. Once a few promising compounds have been identified, the main question to be answered is: "What modifications should be introduced to the molecule to improve its affinity, solubility, bioavailability, etc.?" Some of these properties will depend more on the target that is being investigated than others. For example, affinity should always be considered in the context of the structure of the target receptor. We will conclude the end-to-end pharmacophore modelling workflow with a ligand-based approach that guides the medicinal chemist in this process and provides him with insights and ideas for reasonable structure modifications steps.

As explained in detail in the methods section, the QPhAR model may be used to generate 3D-activity grids around a molecule or pharmacophore. Grids can be generated for each feature type present in the QPhAR model and will be split into positive and negative contributions. The positive grids can be interpreted as points in space, where a pharmacophore feature of this type would be beneficial for a higher activity of the given molecule towards the target receptor. Such kind of information is invaluable for any medicinal chemist working on the structural optimisation of lead compounds. It provides the location as well as the type of interaction that potentially improves the sought-after property of an investigated molecule. Negative grid regions, on the other hand, can be interpreted as portions of space where features of a particular type are unfavourable. Any feature of the analysed type in this region is expected to reduce the molecule's activity towards the target and should be avoided, if possible. Unless the model is generated for an anti-target, such as hERG. In such cases, the negative grids might provide the medicinal chemist with ideas on optimising a molecule's structure in a way that helps to avoid binding to the anti-target.

To elaborate on that, we analysed the activity grids of selected known hERG blockers to explore the potential of this method. The blockers were obtained from Perry et al. [20], whereas two of these are discussed in further detail here. Molecules, pharmacophores, as well as generated activity grids, are provided in the data in the author's GitHub repository (https://github.com/StefanKohlbacher/qphar-applications).

Figure 3 shows the generated activity grids for Ibutilide, a known hERG blocker. For each of the six pharmacophore feature types (Aromatic—AR, Hydrophobic—H, H-Bond acceptor—HBA, H-Bond donor—HBD, Positive ionizable—PI, Negative ionizable—NI), a positive and a negative grid was generated. Only the grids relevant to Ibutilide are shown.

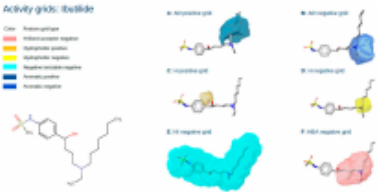

Figure 3. Structure of the known hERG blocker Ibutilide and its respective activity grids: aromatic positive grid (A); aromatic negative grid (B); hydrophobic positive grid (C); hydrophobic negative grid (D); negative ionizable grid (E); hydrogen bond acceptor negative grid (F).

The hERG channel has a well-studied ligand SAR with known distinct binding features that are relevant for high activity. These are two aromatic features, although one is sufficient for strong binders, and a basic nitrogen, forming a y-shaped binding motive [21]. As a rule of thumb, the more hydrophobic a compound is and the lower its pKa, the more likely it will bind to the hERG channel. Figure 3A,B show the grids for aromatic features. Both grids provide information on how the activity of Ibutilide towards hERG is expected to change when introducing a feature (functional group) within the outlined locations. Improvements in activity can be expected when an aromatic feature is introduced at the aliphatic chain neighbouring the basic nitrogen. The positive field shows a clear distinction to locations near the nitrogen, where an aromatic feature would be unfavourable, as seen

in the negative aromatic field, which colocates the basic nitrogen. Introducing an aromatic feature in the aliphatic chain would nicely match the known SAR of the y-shaped binding motive. Furthermore, Figure 3C,D show the activity fields for additional hydrophobic features introduced to Ibutilide. Here, introducing a hydrophobic feature near the phenyl ring (Figure 3C) would yield positive results, as expected due to the fact that hydrophobicity generally increases the affinity to hERG. On the other hand, the negative field in Figure 3D indicates that introducing a hydrophobic feature near or instead of the basic nitrogen would lead to a decrease in expected activity. Again, this agrees with the known SAR, highlighting the central nitrogen's basicity as a crucial binding motive. Similarly, introducing a negative ionisable feature, such as a carboxylic acid, to any location in the molecule increases the pKa, which is known to be unfavourable. This fact can be observed in Figure 3E. Finally, there is the negative field of H-bond acceptor features shown in Figure 3F, which indicates negative expected changes when introduced at or near the basic nitrogen atom. The conclusions obtained from this field are not as clear as those from the other fields. On the one hand, introducing H-bond features, replacing some of the hydrophobic interactions, would decrease the logP, which is roughly equivalent to an increase in pKa and, therefore, unfavourable for binding. On the other hand, H-bond acceptors would be able to interact with external hydrogens in a similar fashion as the positive ionisable group from the basic nitrogen. The strength of this interaction and, therefore, the activity depends heavily on the functional group introduced. Therefore, it is not immediately clear that introducing an HBA feature would result in a negative expected change of activity.

A similar analysis can be made for the molecule E-4031 shown in Figure 4, which is also a known hERG blocker. For subparts Figure 3B–F, the conclusions drawn are the same as those discussed above for Ibutilide, which shows a clear agreement of its 3D activity profile with the known SAR of hERG as reported in the literature [21]. Additionally, Figure 2A shows an activity field for expected negative interactions with H-bond donors colocated with the basic nitrogen atom of E-4031. It follows that replacing the basic nitrogen would be detrimental to activity since the opposing interaction partner is expected to donate a hydrogen atom to the nitrogen, forming an ionic interaction. Opposing such an interaction with another H-bond donor on the side of E-4031 would lead to a loss of this ionic interaction, clearly unfavourable for high-affinity binding to hERG.

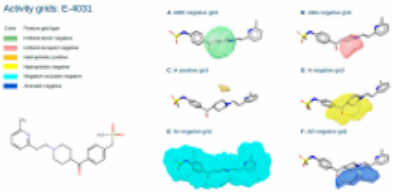

Figure 4. Structure of the known hERG blocker E-4031 and its respective activity grids: hydrogen bond donor negative grid (A); hydrogen bond acceptor negative grid (B); hydrophobic positive grid (C); hydrophobic negative grid (D); negative ionizable grid (E); aromatic negative grid (F).

Overall, Figures 3 and 4 nicely show that an analysis of selected ligands with the QPhAR model derived grids can provide valuable insights for a medicinal chemist and provide him with ideas and even clear directions for optimising the hit or lead molecules. An additional case study based on the dataset from Ece et al. [15,22,23] can be found in the supplementary material.

3. Materials and Methods

The algorithm and workflows described in the following were implemented, unless stated otherwise, in Python 3 using functionality provided by the Chemical Data Processing Toolkit [24].

3.1. Datasets and Training of QPhAR Models

The selection of datasets for quantitative studies is not straightforward and often underestimated. Here, we chose datasets that already have been used in previous validation studies [19] of the QPhAR algorithm. Nevertheless, the datasets were required to fulfill the following criteria:

- A separate training and test set has been defined previously.
- The training set contains between 15 and 30 molecules.
- Activity values for each compound in the dataset were measured in Ki or IC50 values.
- To avoid modelling experimental noise, the associated activity values range by at least three orders of magnitude.

Finally, the activity values have to be somewhat homogeneously distributed over the dataset and not clustered. This requirement has been validated visually.

After filtering, five datasets remained, which were used to evaluate the developed workflows and methods. The datasets are provided for download on the author's Github repository (https://github.com/StefanKohlbacher/qphar-applications). Three-dimensional conformations were calculated for each dataset using LigandScout's iConfGen [25]. Default settings were used with a maximum of 25 output conformations for each molecule. Training and test data were split as described in the publications associated with the datasets [14–18,22,23]. Each compound in each dataset was categorised into active and inactive. As a default, the compounds were ranked by their activity, with the compounds in the top 20th percentile being labelled as active, and the remaining compounds as inactive.

3.2. Screening Baselines

Shared-pharmacophore models were generated and used as baselines in this study. They were generated from a subset of active compounds for each dataset based on typical assumptions made in pharmacophore modelling [5,26]. Whether a compound is considered active or inactive strongly depends on the context of the investigated target and often requires in-depth knowledge about its peculiarities. Usually, values in the range of 1 µM are considered a reasonable threshold for the separation into actives and inactives. The analysed datasets contained compounds ranging from a few nM to a few hundred µM. Therefore, and due to the relatively homogeneous distribution of activity values in the datasets, the cutoff for active compounds was set at the 20th percentile of the dataset. Any compound with activity values below this threshold was considered active, all other compounds inactive. This subset was subsequently used to generate a shared-pharmacophore with LigandScout's [25] command-line tool Espresso.

3.3. Hyperparameter Optimisation

Hyper-parameters were optimised both for the generation of the refined pharmacophore and the shared-pharmacophore baseline (number of most active compounds to use for the generation of the shared-pharmacophore). The following parameters were optimised for the refined pharmacophore:

- Weight features by importance: True, False.
- Set exclusion volumes: True, False.
- Calculate feature contribution from ML (alternatively from QPhAR model): True, False.
- Number of resulting features: [4, 8].

3.4. Refined Pharmacophore Generation Algorithms

In the following, the algorithm to generate a refined pharmacophore from a trained QPhAR model will be explained in detail. The algorithm is based on the assumption that the QPhAR model was trained using a random forest (RF) regressor. Random forest was chosen since it has been shown to be the most promising method to train a QPhAR

model [19]. However, similar conclusions can be derived from other machine learning models, such as linear regression models.

The generation of a refined pharmacophore in the QPhAR context consists of four main steps:

- Determination of feature importance.
- Determination of feature contribution.
- Processing negatively contributing features.
- Selection of features for refined pharmacophore.

3.5. Determination of Feature Importance

Feature importance is derived from the underlying machine learning model of QPhAR via extraction from the random forest model generated by scikit-learn's [27] RF implementation. The feature importance is calculated during the training of the machine learning model and gives insight into the amount of information provided by this feature. The higher the feature's importance, the more information it contains, and the more relevant it is for activity prediction. An analogous concept would be the set of coefficients in a linear regression model.

3.6. Determination of Feature Contribution

In contrast to the feature importance, which is easily obtained, the information on whether a feature contributes positively or negatively to predicted values is not immediately accessible in RF-based models. Within the context of a trained QPhAR model, this information can be obtained directly from the QPhAR pharmacophore without additional information from the machine learning model.

- Feature contribution information derived from the QPhAR pharmacophore model: As explained in the QPhAR publication [13], the QPhAR algorithm associates each newly generated pharmacophore feature with a list of activities. These activities will not only be used to determine the relevance of the feature—whether it is actual information or just adds noise to the model—but also to determine the contribution of a pharmacophore feature to the models' predictions. The mean activity based on the list of associated features is calculated for each feature, resulting in one feature-activity for each pharmacophore feature. Finally, the feature-activities are compared against each other and scaled by their variance. Features with a positive sign of its scaled activity are considered to contribute positively to the prediction of the QPhAR model. Features with a negative sign contribute negatively to the prediction.
- Feature contribution information derived from the RF model: To extract feature contributions from an RF model in a deterministic way, two assumptions are made. First, the data provided to the machine learning model in the QPhAR algorithm represent the pairwise distances between features of the QPhAR model and the pharmacophore to predict. Second, applying the splitting criterion of each node in a tree of the random forest model will yield the left-child node for input values below or equal to the splitting threshold and the right-child node for input values above the splitting threshold. Both these assumptions are ensured by the implementation of the QPhAR algorithm as well as scikit-learn's RF implementation.
- Following this logic, a simple algorithm can be devised to determine whether a feature contributes positively or negatively to the prediction of a sample. For each node in each tree, the node's value is obtained and compared against its neighbouring node. Suppose the left child node has the higher predicted activity. In that case, we can assume that this feature contributes positively to activity since the left child node represents a smaller distance of pairwise pharmacophore features. At the same time, the right child node yields the lower activity prediction, which is associated with a larger distance of pharmacophore feature pairs. On the other hand, if the left child node yields the lower predicted activity, which is associated with a smaller feature pair distance, then the feature can be considered to contribute negatively to activity.

- During this process, the feature-id of each node is obtained, which corresponds to the pharmacophore feature it represents. The value of the feature with the corresponding feature-id is aggregated as the mean value of all nodes that either obtain their value from this feature-id or have a child node that obtains the prediction processing this feature-id. Once all trees and nodes are processed, a value representing the activity is obtained for each pharmacophore feature. These values are scaled as above by their variance. Once again, features with a positive sign are considered to contribute positively to the activity, whereas features with a negative sign are considered to contribute negatively to the activity.

3.7. Processing Negatively Contributing Features

Based on the analysis of feature contribution in the previous step, a post-processing step for negatively contributing features is carried out. The algorithm includes the option to either ignore these features entirely, in which case they are removed from the refined output pharmacophore, or convert them to exclusion volume spheres.

3.8. Selection of Features for the Refined Output Pharmacophore

Finally, the output pharmacophore is created from this list of features with their associated activity values. The features are sorted by their activity contribution values in descending order, resulting in the feature with the most positive contribution in the first place. If feature importance have been obtained from a random forest model in the next-to-last step, the features can optionally be weighted by their feature importance. The first x features are then added to the output pharmacophore, whereas x is a value specified by the user beforehand and the value of the feature is not negative. x is recommended to be a value within the interval [4, 8]. If exclusion volume spheres have been generated in the previous step, these are also added to the refined output pharmacophore based on the sorted list of features.

3.9. 3D Activity Profiling

The activity profile of a sample, pharmacophore or molecule, in 3D space, can be generated with the help of a previously trained QPhAR model. The model should be validated sufficiently before its use and have a narrow confidence interval for high confidence in the model's predictions. The sample of interest is then aligned to the QPhAR model, and the baseline prediction is obtained. A grid is generated with a predetermined interval and some margin extending the sample's size. For each pharmacophore feature type, a probe is placed and moved along the grid. At each point, the current pharmacophore is predicted by the QPhAR model, and the prediction is associated with the location in the grid. Once all grid points are processed, the differences between the predicted grid point values and the previously obtained baseline prediction are calculated. Optionally, the obtained grid of differences can be normalised for better analysis.

The grids were saved in the *.kont format and then loaded into LigandScout alongside the molecules and pharmacophores for analysis. The terms 'activity grids' and 'activity fields' will be used interchangeably in the remainder of this section.

3.10. Metrics

The F1-score, or F-score, is a well-known and often applied metric in machine learning [28] and is defined as the harmonic mean of precision and sensitivity. However, due to the nature of virtual screening, the following scores, derived from the F1-score, will be more suitable for characterising the results of this study.

3.10.1. F_β-Score

The F_β-score [29] is directly derived from the F1-score and weights precision and sensitivity by the factor β. It is calculated by

$$F_\beta = \frac{(1+\beta^2) * precision * recall}{\beta^2 * precision + recall}, \quad (1)$$

The β-value was set to 0.5 for all evaluations in this study.

3.10.2. $F_{Specificity}$-Score

Analogous to the F_β-score, we define the $F_{Specificity}$-score to focus more on the ratio between false positive and false negative hits during virtual screening.

$$F_{Specificity} = \frac{precision * specificity}{precision + specificity}, \quad (2)$$

3.10.3. $F_{Composite}$-Score

We define the $F_{Composite}$-score, which is calculated as the mean of the F_β-score and $F_{Specificity}$-score, as a metric to model the objective of virtual screening.

$$F_{Composite} = \left(F_\beta + F_{Specificty}\right)/2, \quad (3)$$

4. Conclusions

Nowadays, pharmacophore-based methods can be considered indispensable and are an integral part of nearly every modern computer-aided drug design project. A combination of pharmacophore modelling and pharmacophore-based virtual screening is often applied as one of the first filtering techniques to obtain a list of promising compound candidates for biological testing in the hit-finding phase. Despite its popularity, pharmacophore modelling is still a task that heavily relies on the expert knowledge of the researcher. In this study, we presented a method for the generation of pharmacophore models with high discriminatory power from a QPhAR model in a deterministic manner following clear generation guidelines. We showed that the pharmacophores derived by our algorithm are superior to a baseline of ligand-based pharmacophore models generated under the assumption that only active molecules are required to produce good query pharmacophores for virtual screening. Furthermore, we incorporated the presented method into a workflow for end-to-end pharmacophore modelling. This workflow facilitates a fully automated process to train a QPhAR model, generate a query pharmacophore from this QPhAR model, screen a database, and finally rank the obtained hits by relevance using the initial QPhAR model. In a case study using known hERG K+ channel blockers, we have shown that the generated activity fields agree well with the known SAR and can, therefore, provide meaningful insights for medicinal chemists in the hit or lead-optimisation phase.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/ph15091122/s1, Figure S1: Structure of the known CDK2 inhibitor Flavopiridol and its respective activity grids; Figure S2: Structure of the known CDK2 inhibitor Roscovitine and its respective activity grids.

Author Contributions: The method was developed, implemented, and validated by S.M.K. M.S. contributed to research and validation. The paper was jointly written by T.L., T.S. and S.M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the NeuroDeRisk project. The NeuroDeRisk consortium has received funding from the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 821528. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation program, and EFPIA. Open Access Funding by the University of Vienna.

Data Availability Statement: Data are contained within the article and supplementary material.

Acknowledgments: Open Access Funding by the University of Vienna.

Conflicts of Interest: The authors declare no competing interests.

References

1. Caporuscio, F.; Tafi, A. Pharmacophore Modelling: A Forty Year Old Approach and Its Modern Synergies. *Curr. Med. Chem.* **2011**, *18*, 2543–2553. [CrossRef] [PubMed]
2. Böhm, H.-J.; Klebe, G.; Kubinyi, H. *Wirkstoffdesign*; Spektrum Akademischer Verlag: Heidelberg, Germany, 1996.
3. Leach, A.R.; Gillet, V.J.; Lewis, R.A.; Taylor, R. Three-Dimensional Pharmacophore Methods in Drug Discovery. *J. Med. Chem.* **2010**, *53*, 539–558. [CrossRef] [PubMed]
4. Güner, O.F. *Pharmacophore Perception, Development, and Use in Drug Design*; Internat'l University Line: San Diego, CA, USA, 2000; pp. 6–8.
5. Yang, S.-Y. Pharmacophore Modeling and Applications in Drug Discovery: Challenges and Recent Advances. *Drug Discov. Today* **2010**, *15*, 444–450. [CrossRef] [PubMed]
6. Taha, M.O.; Dahabiyeh, L.A.; Bustanji, Y.; Zalloum, H.; Saleh, S. Combining Ligand-Based Pharmacophore Modeling, Quantitative Structure−Activity Relationship Analysis and in Silico Screening for the Discovery of New Potent Hormone Sensitive Lipase Inhibitors. *J. Med. Chem.* **2008**, *51*, 6478–6494. [CrossRef] [PubMed]
7. Kurogi, Y.; Güner, O.F. Pharmacophore Modeling and Three-Dimensional Database Searching for Drug Design Using Catalyst. *Curr. Med. Chem.* **2001**, *8*, 1035–1055. [CrossRef] [PubMed]
8. Schneider, G.; Neidhart, W.; Giller, T.; Schmid, G. "Scaffold-Hopping" by Topological Pharmacophore Search: A Contribution to Virtual Screening. *Angew. Chem. Int. Ed.* **1999**, *38*, 2894–2896. [CrossRef]
9. Vuorinen, A.; Schuster, D. Methods for Generating and Applying Pharmacophore Models as Virtual Screening Filters and for Bioactivity Profiling. *Methods* **2015**, *71*, 113–134. [CrossRef]
10. Mason, J.S.; Good, A.C.; Martin, E.J. 3-D Pharmacophores in Drug Discovery. *Curr. Pharm. Des.* **2001**, *7*, 567–597. [CrossRef]
11. Chen, X.; Rusinko, A., 3rd; Tropsha, A.; Young, S.S. Automated pharmacophore identification for large chemical data sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 887–896. [CrossRef]
12. Baber, J.C.; Shirley, W.A.; Gao, Y.; Feher, M. The Use of Consensus Scoring in Ligand-Based Virtual Screening. *J. Chem. Inf. Model.* **2006**, *46*, 277–288. [CrossRef]
13. Kohlbacher, S.M.; Langer, T.; Seidel, T. QPHAR: Quantitative Pharmacophore Activity Relationship: Method and Validation. *J. Cheminform.* **2021**, *13*, 57. [CrossRef]
14. Garg, D.; Gandhi, T.; Gopi Mohan, C. Exploring QSTR and Toxicophore of HERG K+ Channel Blockers Using GFA and HypoGen Techniques. *J. Mol. Graph. Model.* **2008**, *26*, 966–976. [CrossRef]
15. Ece, A.; Sevin, F. The Discovery of Potential Cyclin A/CDK2 Inhibitors: A Combination of 3D QSAR Pharmacophore Modeling, Virtual Screening, and Molecular Docking Studies. *Med. Chem. Res.* **2013**, *22*, 5832–5843. [CrossRef]
16. Ma, Y.; Li, H.-L.; Chen, X.-B.; Jin, W.-Y.; Zhou, H.; Ma, Y.; Wang, R.-L. 3D QSAR Pharmacophore Based Virtual Screening for Identification of Potential Inhibitors for CDC25B. *Comput. Biol. Chem.* **2018**, *73*, 1–12. [CrossRef]
17. Wang, H.-Y.; Li, L.-L.; Cao, Z.-X.; Luo, S.-D.; Wei, Y.-Q.; Yang, S.-Y. A Specific Pharmacophore Model of Aurora B Kinase Inhibitors and Virtual Screening Studies Based on It. *Chem. Biol. Drug Des.* **2009**, *73*, 115–126. [CrossRef]
18. Krovat, E.M.; Langer, T. Non-Peptide Angiotensin II Receptor Antagonists: Chemical Feature Based Pharmacophore Identification. *J. Med. Chem.* **2003**, *46*, 716–726. [CrossRef]
19. Schmid, M. Validation of the Novel Quantitative Pharmacophore Modeling Algorithm QPhAR. Master's Thesis, University of Vienna, Vienna, Austria, 2022.
20. Perry, M.; Sanguinetti, M.; Mitcheson, J. Revealing the Structural Basis of Action of HERG Potassium Channel Activators and Blockers. *J. Physiol.* **2010**, *588*, 3157–3167. [CrossRef]
21. Garrido, A.; Lepailleur, A.; Mignani, S.M.; Dallemagne, P.; Rochais, C. HERG Toxicity Assessment: Useful Guidelines for Drug Design. *Eur. J. Med. Chem.* **2020**, *195*, 112290. [CrossRef]
22. Zhang, J.; Gan, Y.; Li, H.; Yin, J.; He, X.; Lin, L.; Xu, S.; Fang, Z.; Kim, B.; Gao, L.; et al. Inhibition of the CDK2 and Cyclin A Complex Leads to Autophagic Degradation of CDK2 in Cancer Cells. *Nat. Commun.* **2022**, *13*, 2835. [CrossRef]
23. Ece, A.; Sevin, F. Exploring QSAR on 4-Cyclohexylmethoxypyrimidines as Antitumor Agents for Their Inhibitory Activity of CDK2. *Lett. Drug Des. Discov.* **2010**, *7*, 625–631. [CrossRef]
24. Seidel, T. Chemical Data Processing Toolkit, GitHub Repository. Available online: https://github.com/aglanger/CDPKit (accessed on 19 March 2021).
25. Wolber, G.; Langer, T. LigandScout: 3-D Pharmacophores Derived from Protein-Bound Ligands and Their Use as Virtual Screening Filters. *J. Chem. Inf. Model.* **2005**, *45*, 160–169. [CrossRef] [PubMed]
26. Kutlushina, A.; Khakimova, A.; Madzhidov, T.; Polishchuk, P. Ligand-Based Pharmacophore Modeling Using Novel 3D Pharmacophore Signatures. *Molecules* **2018**, *23*, 3094. [CrossRef] [PubMed]
27. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

28. Taha, A.A.; Hanbury, A. Metrics for Evaluating 3D Medical Image Segmentation: Analysis, Selection, and Tool. *BMC Med. Imaging* **2015**, *15*, 29. [CrossRef]
29. Chinchor, N. MUC-4 Evaluation Metrics. In Proceedings of the 4th Conference on Message Understanding, McLean, VA, USA, 16 June 1992; pp. 22–29.

Applications of the novel quantitative pharmacophore activity relationship method QPhAR in virtual screening and lead-optimization

Stefan M. Kohlbacher, Matthias Schmid, Thomas Seide and Thierry Langer

Supplementary Material

An additional case study for activity profiling has been carried out on the target CDK2 with the model obtained from Ece et al. [15]. The analysed molecules were obtained from Zhang et al. [16]. A few characteristic binding motives and relationships have been identified in previous SAR studies [17]. A few of these insights are:

- An increase in the hydrophobic surface area of the ligand generally leads to reduced biological activity
- H-Bonds to the protein backbone are crucial for binding

If available, further insights on the ligand-protein interaction were obtained from inspecting the co-crystallized ligand in the binding pocket. The activity grids, as well as aligned molecules and pharmacophores for all molecules, can be found on the author's GitHub repository.

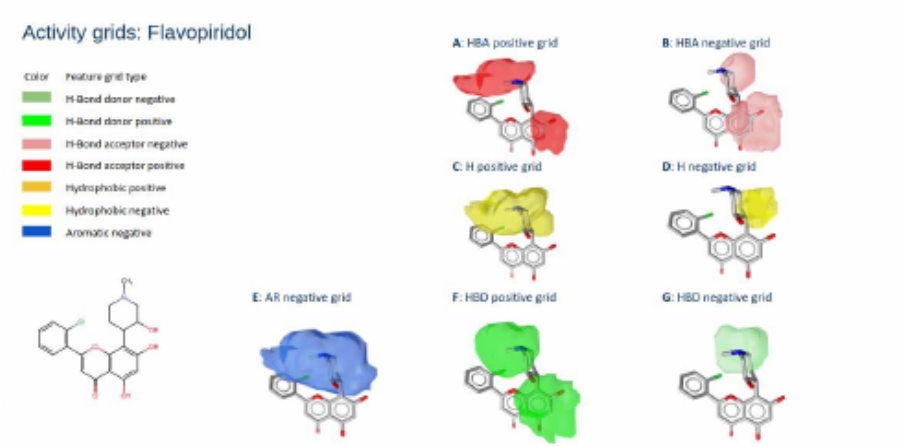

Figure S1. Activity grids for the known CDK2 inhibitor Flavopiridol.

Figure S1 shows the obtained activity grids for Flavopiridol, a strong CDK2 inhibitor, with a reported K_d of 23 nM in the PDB entry of 6GUB. The crystal structure (PDB-Code: 6GUB) was used to gain further insights into favourable interactions and compare these insights against the information provided by the activity grids.
The grids of Figure S1 **A** and **B** show the expected positive and negative regions, respectively. Considering the overall SAR knowledge as well as the supposed position of the molecule in the binding site (Note: The pose of the molecule in the binding site is similar but different from the aligned pose of the molecule to the QPhAR model. These incongruencies between the two conformations and their independent positioning, docking and pharmacophore alignment,

unfortunately, do not allow for a direct comparison of the grids within the binding site.), the positive HBA grids agree very well with the known SAR motives. An interesting case that should be highlighted is the almost overlapping positive and negative field on the right-hand side of the molecule. However, when inspecting the grids in 3D, it becomes clear that the negative fields co-locate with the molecule's atoms, whereas the positive fields are further distant from the molecule, indicating the importance of spatial distance with H-bonding interactions. The hydrophobic fields, **C** and **D**, lead to similar conclusions. The negative field, which in the image is located in front of the molecule, agrees very well with known SAR and the protein binding site. The positive field is slightly ambiguous. On the one hand, it is located on the backside of the molecule near a valine residue of the protein, which agrees with the suggestion of the activity field. However, the size of the positive field is questionable. The negative AR field agrees well with known SAR. Finally, the positive and negative grids for HBDs shown in **F** and **G** agree well with the known SAR as well as the other overlapping fields. For example, on the backside of the molecule, the negative HBD fields and the positive H fields are co-locating the same space. Again, given the molecules supposed orientation in the binding site and the vicinal valine residue, H-Bond donor interactions would be unfavourable in this position. On the other hand, the positive HBD fields indicate favourable expected interactions at the front of the molecule.

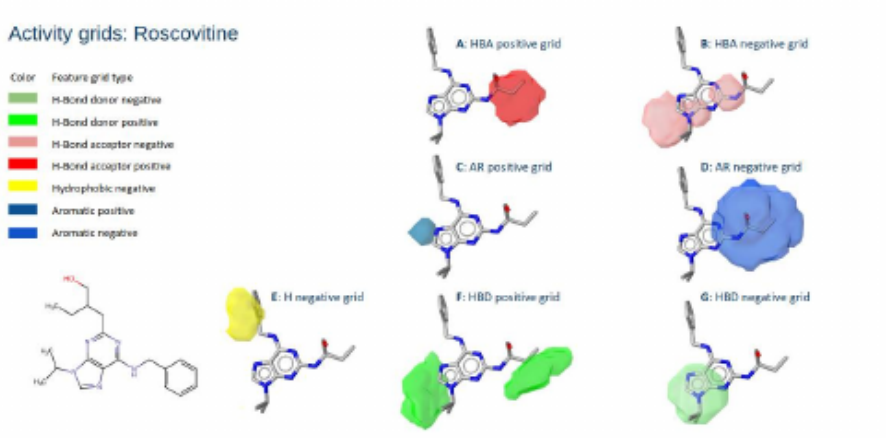

Figure S2: Activity grids for the known CDK2 inhibitor Roscovitine.

Figure S2 shows a similar analysis for the CDK2 inhibitor Roscovitine. Overall, similar conclusions, as were drawn for Flavopiridol, can be obtained for the activity grid analysis of Roscovitine. An interesting fact to highlight is the positive HBD grid. Inspection of the co-crystallized ligand in the binding site (PDB-Code: 3DOG) indicates a missing HBD interaction deep into the pocket. A typical binding motive for kinase inhibitors is the alternating HBD-HBA-HBD interaction of the ligand with the protein backbone. Roscovitine, however, is missing an HBD interaction inside the pocket. Analysing the positive HBD fields leads to the same conclusion, an additional HBD interaction is expected to be favourable for the molecule's activity. The conclusion is further supported by the negative HBA fields, which indicate unfavourable interactions when introducing an HBA feature in this region of the molecule.

III. Concluding discussion

The work conducted in this thesis has focused on developing novel algorithms and applications in the domain of *in-silico* drug discovery, specifically pharmacophores. Despite the broad application of pharmacophores in today's drug discovery campaigns, hardly any research has been conducted in the last decades to further improve pharmacophores themselves. Within the scope of the IMI NeuroDeRisk project, the need for developing next-generation pharmacophores has been identified and addressed in this thesis.

The first study presented, QPHAR: quantitative pharmacophore activity relationship: method and validation, showed that QSAR models based on pharmacophores are a viable alternative to existing QSAR methods while at the same time they have various advantages for the user. Among others, the fuzziness of pharmacophores facilitates a robust training process and generalisation to external datasets. The strengths of quantitative pharmacophore models become especially prevalent in datasets containing various molecular structures and functional groups. Here, the concept of scaffold-hopping can be leveraged to get quantitative insights into previously unseen molecular motives. By abstracting these motives into their corresponding pharmacophore features, high-confidence predictions can be made for these data points. An example would be bioisosterism. The dataset might contain various data points featuring phenyl rings, represented as aromatic pharmacophore features. During inference, data points containing thiophene substructures will be presented to the model. With conventional atom-based QSAR methods, such data points would be considered out-of-domain, and it would not be possible to obtain predictions. However, just like phenyl rings, thiophenes can also be represented by aromatic pharmacophore features. This fact allows the quantitative pharmacophore model to predict these new data points and draw corresponding conclusions. In the early phase of hit discovery, such insights can be extremely valuable to the medicinal chemist and fine-grained optimisation of structures is not yet the focus. Instead, the discovery of novel scaffolds with similar binding motives as one or a few query molecules is guiding the hit discovery phase.

The second study presented, A new set of KNIME nodes implementing the QPhAR algorithm, has focused on the dissemination and delivery of previously developed QPHAR methods. Accessibility and ease of use of chemoinformatic methods have been identified as urgent needs during the work of this thesis. The interdisciplinary approach in the NDR project, as well as collaboration with various other disciplines, have highlighted this need even further. Therefore, the QPhAR algorithm was wrapped and made accessible by a set of KNIME nodes. In combination with the released workflow and guidance on best-practice applications of the QPhAR algorithm, the developed QPhAR methods were made available to researchers in a broad range of domains. KNIME has been chosen due to its popularity in the scientific community and its simple and intuitive approach to data processing. Besides simple training of QPhAR models, the pipeline-like approach of KNIME allows the integration of the QPhAR nodes into larger complex drug discovery workflows. Coupled with a workflow for pharmacophore elucidation, these pharmacophores can be scored in an automatic fashion by a previously trained QPhAR. In a further step of the workflow, a selected pharmacophore will then be used for the virtual screening of large molecule databases. Furthermore, the QPhAR model can immediately be applied to score the obtained hits from the virtual screening, effectively ranking the hits by relevance for further experiments.

The third study, Applications of the novel quantitative pharmacophore activity relationship method QPhAR in virtual screening and lead-optimization, has focused on developing such an automated drug discovery workflow. The aim of the study was the end-to-end integration of QPhAR models based on a single dataset. It was shown that a small set of molecules, serving as the QPhAR model's training set, is sufficient to deduce and model the entire hit-to-lead process. At first, a QPhAR model is built from a subset of the molecules, the training set. The QPhAR model is being validated on the remaining test set. An algorithm was developed to extract a refined pharmacophore from the validated QPhAR model, which can be applied to screen a large database of molecules for hit identification. The obtained hits can immediately be ranked by the QPhAR model to prioritise them for further testing. Finally, the QPhAR model can be applied to gain insights into the SAR of the target of interest. Grids are generated for hit or lead molecules highlighting (dis-)advantageous regions of the molecule. The positive and negative regions of the grids can be interpreted as an expected increase or decrease, respectively, of the molecule's activity if certain pharmacophore features are present in these regions. They will guide a medicinal chemist with insights on which parts of the lead molecules to optimise for increased target activity. Furthermore, training of the QPhAR model is not restricted to the prediction of activity values, although this is the most common use case. With appropriate datasets, a QPhAR model can easily be trained to give insights into various other ADMET properties relevant to the drug discovery process.

An interesting extension, but out of scope for this thesis, is the application of QPhAR models to pharmacophores obtained from binding pockets of proteins. The application of QPhAR has been strictly limited to ligand-based studies. However, getting insights on docked ligands, co-crystallized ligands, and interaction pharmacophores would provide the medicinal chemist with invaluable information in the decision process for lead optimisation. Furthermore, due to the ligand-based nature of the QPhAR algorithm, it is heavily dependent on the alignment algorithm used. This effect was observed during the development of the QPhAR KNIME nodes and should not be underestimated. Therefore the scientific community could also benefit from an in-depth comparison of various alignment methods and their effect on the quality of QPhAR models.

Appendix

QPHAR: quantitative pharmacophore activity relationship: method and validation

Table 1: CV Split 20-80

Target	Assay	NrSamples	MinActivity	MaxActivity
Acetylcholinesterase	CHEMBL3049408	22	4.4	9.52
Acetylcholinesterase	CHEMBL945890	29	4.28	10.7
Acetylcholinesterase	CHEMBL3626447	21	4.75	8.64
Acetylcholinesterase	CHEMBL3047076	72	4.03	7.35
Acetylcholinesterase	CHEMBL644831	33	4.71	8.59
Acetylcholinesterase	CHEMBL1664219	60	5.59	9.48
Acetylcholinesterase	CHEMBL643386	21	5	9.09
Acetylcholinesterase	CHEMBL4146836	35	4.57	9.44
Acetylcholinesterase	CHEMBL1249049	38	4.39	8.15
Acetylcholinesterase	CHEMBL3626446	21	4.62	8.42
Acetylcholinesterase	CHEMBL644105	26	5.58	9.1
Acetylcholinesterase	CHEMBL1041236	29	4.3	8.96
Acetylcholinesterase	CHEMBL3387411	22	4.98	9.14
Acetylcholinesterase	CHEMBL1646125	20	4.62	8.3
Acetylcholinesterase	CHEMBL4198693	24	4.13	7.75
Acetylcholinesterase	CHEMBL641238	28	4.06	8.1
Acetylcholinesterase	CHEMBL3758061	28	5.12	9.09
Acetylcholinesterase	CHEMBL894085	28	4.09	7.98
Acetylcholinesterase	CHEMBL829154	26	4.38	9.49
COX2	CHEMBL769653	20	4.11	8.52
COX2	CHEMBL947228	70	5.24	9.52
COX2	CHEMBL947226	24	5	8.23
COX2	CHEMBL768957	22	4.52	8.46
COX2	CHEMBL655563	29	4	7.33
COX2	CHEMBL762912	27	4	7.85
COX2	CHEMBL1008495	27	4	7.57
COX2	CHEMBL769648	50	4.03	8.77
COX2	CHEMBL3811631	24	4.12	7.7
COX2	CHEMBL3579883	20	4.34	8.2
COX2	CHEMBL1059941	21	5.77	9.6
COX2	CHEMBL3062548	69	5.24	9.52
COX2	CHEMBL1639903	33	4	8.3
COX2	CHEMBL3363266	23	4.02	8.1
DHFR	CHEMBL669873	21	6	10.6

DHFR	CHEMBL669986	76	4.51	8.7
DHFR	CHEMBL2384165	28	4.49	8.77
Estrogen	CHEMBL832786	23	5.78	8.82
Estrogen	CHEMBL4024644	43	5.11	9
Estrogen	CHEMBL1909145	32	4.75	9.89
Estrogen	CHEMBL3776150	24	5.96	9.16
Estrogen	CHEMBL3429911	34	4.97	8.57
Estrogen	CHEMBL4200119	24	5.03	8.7
Estrogen	CHEMBL676818	27	5.45	8.89
Estrogen	CHEMBL3429910	35	5.24	9.82
Estrogen	CHEMBL679891	29	6.25	9.3
Carbonic	CHEMBL3122609	22	5.06	8.25
Carbonic	CHEMBL1646987	29	5.02	8.68
Carbonic	CHEMBL658942	43	6.1	9.3
Carbonic	CHEMBL3879546	25	5.75	9.3
Carbonic	CHEMBL657192	28	5.05	9.15
Carbonic	CHEMBL4017666	23	6.02	9.03
Carbonic	CHEMBL1693045	28	5.02	8.68
Carbonic	CHEMBL657145	44	6.52	9.7
Carbonic	CHEMBL3269990	45	4.55	8.15
Carbonic	CHEMBL3377547	21	5.29	8.36
Carbonic	CHEMBL3366665	21	5.29	8.36
D2	CHEMBL872881	27	5.26	8.7
D2	CHEMBL1061432	20	4.27	7.39
D2	CHEMBL673073	28	4.34	7.41
D2	CHEMBL3801477	25	4.07	8.15
D2	CHEMBL948052	21	6.04	9.34
D2	CHEMBL866800	26	4.17	7.55
D2	CHEMBL3614290	35	6	9.32
D2	CHEMBL1909140	100	4.7	9.31
D2	CHEMBL3089794	24	4.87	7.98
D2	CHEMBL4046973	21	4.85	9.28
D2	CHEMBL3073105	38	5.85	9.72
D2	CHEMBL4046974	22	5.05	9.42
D2	CHEMBL866799	26	4.19	7.39
D2	CHEMBL671368	29	5.1	8.64
erbB1	CHEMBL677204	32	4	10.54
erbB1	CHEMBL1798351	101	4.05	8.02
erbB1	CHEMBL680017	63	4.92	10.6
erbB1	CHEMBL3616083	26	5.28	10.54
erbB1	CHEMBL1686143	30	4.33	9
erbB1	CHEMBL677880	34	4.11	8.4
erbB1	CHEMBL3877875	20	5	9

erbB1	CHEMBL3240253	20	5.26	9
erbB1	CHEMBL679944	48	5.37	10.6
erbB1	CHEMBL1064820	23	4.39	7.52
erbB1	CHEMBL2439336	26	4.47	8.92
erbB1	CHEMBL674019	23	5.11	8.57
erbB1	CHEMBL2066000	25	4.1	7.21
erbB1	CHEMBL925198	23	5.14	8.74
erbB1	CHEMBL680021	43	5.69	10.54
erbB1	CHEMBL3614855	25	5.11	8.14
erbB1	CHEMBL1064829	32	4.36	7.7
erbB1	CHEMBL4052480	21	4.95	9.42
erbB1	CHEMBL3238947	39	5.19	9.05
erbB1	CHEMBL674637	22	5.28	9.35
erbB1	CHEMBL1073378	26	5.21	8.7
Glucocorticoid	CHEMBL941996	43	5.35	9.4
Glucocorticoid	CHEMBL3868161	35	7	10.52
Glucocorticoid	CHEMBL1909150	76	4.38	9.18
Glucocorticoid	CHEMBL679612	44	4.6	8.24
Glucocorticoid	CHEMBL890119	32	5.16	8.47
Ephrin	CHEMBL998745	25	5.51	8.52
Ephrin	CHEMBL1686716	37	5	8.4
Ephrin	CHEMBL1068887	59	4.6	9.9
Ephrin	CHEMBL915413	29	5.14	8.51
Ephrin	CHEMBL945185	27	5.22	8.3
Ephrin	CHEMBL3707854	22	5.19	9
Ephrin	CHEMBL1101480	28	5.45	8.52
Ephrin	CHEMBL1120739	36	5.65	9
Ephrin	CHEMBL932888	33	5.08	8.36
Ephrin	CHEMBL1816473	33	4.85	8.1
Ephrin	CHEMBL1047378	33	5.71	9
COX1	CHEMBL763480	23	4.52	7.7
COX1	CHEMBL1909130	45	4.5	8.35
Glycogen	CHEMBL3413879	26	6.14	9.54
Glycogen	CHEMBL1102983	30	5.09	9.4
Glycogen	CHEMBL1113275	51	4.01	8.22
Glycogen	CHEMBL684976	60	4.39	8
Glycogen	CHEMBL2182637	30	6.16	9.66
Glycogen	CHEMBL928210	36	5.38	8.52
Glycogen	CHEMBL4057839	32	6.04	10
Glycogen	CHEMBL1034478	33	3.38	10.05
Glycogen	CHEMBL2390763	31	6.39	10.89
Glycogen	CHEMBL1249829	20	5.7	9.24
Glycogen	CHEMBL963374	40	5.14	9.64

Glycogen	CHEMBL3097577	25	5.01	8.05
Aurora-A	CHEMBL972583	22	5.14	8.4
Aurora-A	CHEMBL993738	44	5.5	9.22
Aurora-A	CHEMBL1664805	58	4.54	8.7
Aurora-A	CHEMBL3888080	73	4.5	9.1
Aurora-A	CHEMBL3862723	42	4.12	10.4
Aurora-A	CHEMBL2173311	59	4.5	9.1
Aurora-A	CHEMBL856402	41	5.7	9
B-raf	CHEMBL1780874	28	6.07	9.24
B-raf	CHEMBL3888085	55	6.07	9.14
B-raf	CHEMBL1217119	83	4.77	8.7
B-raf	CHEMBL1104358	45	4.16	7.85
B-raf	CHEMBL1067537	64	4.03	8.3
B-raf	CHEMBL1671058	20	5.85	9.89
B-raf	CHEMBL1953548	22	4.6	9
B-raf	CHEMBL3816885	30	4.5	9.3
B-raf	CHEMBL3240201	24	6.07	9.7
B-raf	CHEMBL2341125	47	5.15	8.4
B-raf	CHEMBL3887740	43	5.75	9
B-raf	CHEMBL981002	32	4.49	7.92
B-raf	CHEMBL3804129	36	6	9
B-raf	CHEMBL3804126	37	6.38	9.4
B-raf	CHEMBL1043507	38	6.52	9.52
B-raf	CHEMBL1008210	37	6.01	10.7
B-raf	CHEMBL1937974	35	4.81	9
B-raf	CHEMBL1671059	20	4.12	8.3
B-raf	CHEMBL946282	31	5.05	9
Vanilloid	CHEMBL832934	26	5.07	8.07
Vanilloid	CHEMBL820100	28	5.84	9.52
Vanilloid	CHEMBL828688	20	4.76	9.4
Vanilloid	CHEMBL3705949	137	4.65	8.7
Vanilloid	CHEMBL3706060	41	4.48	7.7
Vanilloid	CHEMBL930279	29	4.82	8.1
Androgen	CHEMBL1960482	23	5.27	9.33
Androgen	CHEMBL891115	42	6	9.7
Androgen	CHEMBL930493	21	6.32	10.3
Androgen	CHEMBL933063	31	5.39	9
Androgen	CHEMBL930492	23	5.63	9.7
Androgen	CHEMBL859226	24	5.34	8.57
Androgen	CHEMBL1169306	41	5.34	8.64
Coagulation	CHEMBL811970	20	4.92	8.4
Coagulation	CHEMBL862522	23	4.7	7.8
Coagulation	CHEMBL1010200	23	4.57	8

Coagulation	CHEMBL832832	63	4.09	8.52
Coagulation	CHEMBL816070	38	4.03	8.33
Coagulation	CHEMBL847937	27	6.8	9.97
Coagulation	CHEMBL814535	22	4.05	8.4
Coagulation	CHEMBL813569	29	6.98	10.82
Coagulation	CHEMBL817243	24	6.25	10.7
Coagulation	CHEMBL860417	24	5.75	9
Coagulation	CHEMBL815806	22	4.38	8.7
Coagulation	CHEMBL814972	22	5.34	10.34
Coagulation	CHEMBL1049043	21	4.77	8.8
Coagulation	CHEMBL884569	35	4.8	10.4
Coagulation	CHEMBL811982	22	4.11	7.6
Coagulation	CHEMBL814951	21	4.22	8.22
Coagulation	CHEMBL1067528	35	4.03	8.4
Coagulation	CHEMBL838818	60	5.15	9.92
Coagulation	CHEMBL845034	23	4.62	8.27
Coagulation	CHEMBL3424914	33	4.12	8.96
Coagulation	CHEMBL813626	41	6.03	10.89
Coagulation	CHEMBL813672	23	5.43	9.09
Coagulation	CHEMBL829646	22	6.82	10.4
Coagulation	CHEMBL812835	20	4.3	7.92
Coagulation	CHEMBL812837	30	5.85	9.57
Coagulation	CHEMBL811969	34	6.33	10
A2a	CHEMBL1259385	22	7.26	10.16
Opiod	CHEMBL1769429	31	5.57	10.21
Opiod	CHEMBL757060	35	5.3	9.05
Opiod	CHEMBL1112210	35	5.5	8.96
Opiod	CHEMBL756359	45	5.37	9.85
Opiod	CHEMBL888956	45	4.22	8.05
Opiod	CHEMBL858889	24	5.96	11
Opiod	CHEMBL754712	22	4.71	8.44
Opiod	CHEMBL3705573	20	5.24	8.87
Opiod	CHEMBL901089	26	6	9.82
Opiod	CHEMBL1050689	40	6.44	10.59
Opiod	CHEMBL3707592	76	5.76	10.4
Opiod	CHEMBL757059	36	5.18	9.05
Opiod	CHEMBL757823	24	6.05	9.19
Opiod	CHEMBL746681	22	5.38	8.82
Opiod	CHEMBL3888044	27	4.85	8.07
Opiod	CHEMBL3888831	88	4.99	9.46
Opiod	CHEMBL2050632	40	6.82	10.21
MAO-A	CHEMBL827909	42	4.19	7.4
MAO-A	CHEMBL2060936	51	4.58	8.25

MAO-A	CHEMBL1648130	38	4.03	7.21
MAO-A	CHEMBL965566	27	4.09	8.42
Caspase	CHEMBL831642	28	4.2	8.4
Caspase	CHEMBL658912	27	4.33	8.6
Caspase	CHEMBL1838643	24	4.09	7.92
Caspase	CHEMBL1012730	27	4.21	8.62
Caspase	CHEMBL658623	25	4.01	7.57
JAK2	CHEMBL3404501	25	5.54	9
JAK2	CHEMBL2395112	22	5.91	8.92
JAK2	CHEMBL1044318	37	6.06	9.7
JAK2	CHEMBL1049517	47	5.35	9.3
JAK2	CHEMBL1011746	23	3.84	7.41
JAK2	CHEMBL2186596	31	6.99	10
JAK2	CHEMBL1936085	21	5.54	10.05
JAK2	CHEMBL2067611	27	5.54	9.19
JAK2	CHEMBL3374493	34	5.5	8.7
JAK2	CHEMBL2438691	32	5.5	9.22
JAK2	CHEMBL1291235	25	4.54	9
CDK2	CHEMBL1250281	25	5.16	9
CDK2	CHEMBL971698	29	3.73	8.52
CDK2	CHEMBL661124	22	5.4	8.68
CDK2	CHEMBL2166172	20	4.21	8.52
CDK2	CHEMBL908827	20	5.2	8.7
Cannabinoid	CHEMBL2187925	27	4.68	8.89
Cannabinoid	CHEMBL2154764	32	5.45	8.61
Cannabinoid	CHEMBL886398	38	5.4	9.05
Cannabinoid	CHEMBL901144	23	5.1	9.1
Cannabinoid	CHEMBL1107820	29	6.13	9.82
Cannabinoid	CHEMBL1011882	62	5.01	9.7
Cannabinoid	CHEMBL2317164	35	4.05	9.07
Cannabinoid	CHEMBL896457	34	5.41	8.77
Cannabinoid	CHEMBL1804296	35	6.21	9.92
Cannabinoid	CHEMBL2210669	25	5.37	8.92
Cannabinoid	CHEMBL887035	29	5.07	8.52
Cannabinoid	CHEMBL901108	30	5.9	9.49
Cannabinoid	CHEMBL832272	36	4.72	8.4
Cannabinoid	CHEMBL932400	32	5.82	9.52
HERG	CHEMBL1676103	146	4	8.66
HERG	CHEMBL1036929	26	4.32	8.6
HERG	CHEMBL2330781	36	4.51	8.39
HERG	CHEMBL2148626	30	5.33	9.12
HERG	CHEMBL3096732	44	4.84	9.4
HERG	CHEMBL1764797	24	4.69	8.7

HERG	CHEMBL3583383	42	4.02	7.3
HERG	CHEMBL827807	49	4	8.82
HERG	CHEMBL766816	29	4.12	8.52
ABL1	CHEMBL3888134	50	5.95	9.3
ABL1	CHEMBL2186853	38	5.67	9.03
ABL1	CHEMBL2353816	50	5.83	10.82
ABL1	CHEMBL923929	25	4.99	8.92
LCK	CHEMBL1113978	53	5.35	8.52
LCK	CHEMBL4145903	26	4.15	9
LCK	CHEMBL913225	34	6.53	11
LCK	CHEMBL841202	44	4.64	8.4
LCK	CHEMBL841204	31	5.7	9.15
LCK	CHEMBL1049046	45	5.57	10.15
LCK	CHEMBL910552	51	4.55	9.7
LCK	CHEMBL864124	29	5.22	8.52
LCK	CHEMBL908547	26	5.06	8.52
LCK	CHEMBL842071	35	5.26	8.7

Table 2: CV Split 80-20

Target	Assay	NrSamples	MinActivity	MaxActivity
Acetylcholinesterase	CHEMBL3049408	22	4.4	9.52
Acetylcholinesterase	CHEMBL945890	29	4.28	10.7
Acetylcholinesterase	CHEMBL3626447	21	4.75	8.64
Acetylcholinesterase	CHEMBL3047076	72	4.03	7.35
Acetylcholinesterase	CHEMBL644831	33	4.71	8.59
Acetylcholinesterase	CHEMBL1664219	60	5.59	9.48
Acetylcholinesterase	CHEMBL643386	21	5	9.09
Acetylcholinesterase	CHEMBL4146836	35	4.57	9.44
Acetylcholinesterase	CHEMBL1249049	38	4.39	8.15
Acetylcholinesterase	CHEMBL3626446	21	4.62	8.42
Acetylcholinesterase	CHEMBL644105	26	5.58	9.1
Acetylcholinesterase	CHEMBL1041236	29	4.3	8.96
Acetylcholinesterase	CHEMBL3387411	22	4.98	9.14
Acetylcholinesterase	CHEMBL1646125	20	4.62	8.3
Acetylcholinesterase	CHEMBL4198693	24	4.13	7.75
Acetylcholinesterase	CHEMBL641238	28	4.06	8.1
Acetylcholinesterase	CHEMBL3758061	28	5.12	9.09
Acetylcholinesterase	CHEMBL894085	28	4.09	7.98
Acetylcholinesterase	CHEMBL829154	26	4.38	9.49
COX2	CHEMBL769653	20	4.11	8.52
COX2	CHEMBL947228	70	5.24	9.52
COX2	CHEMBL947226	24	5	8.23
COX2	CHEMBL768957	22	4.52	8.46

COX2	CHEMBL655563	29	4	7.33
COX2	CHEMBL762912	27	4	7.85
COX2	CHEMBL1008495	27	4	7.57
COX2	CHEMBL769648	50	4.03	8.77
COX2	CHEMBL3811631	24	4.12	7.7
COX2	CHEMBL3579883	20	4.34	8.2
COX2	CHEMBL1059941	21	5.77	9.6
COX2	CHEMBL3062548	69	5.24	9.52
COX2	CHEMBL1639903	33	4	8.3
COX2	CHEMBL3363266	23	4.02	8.1
DHFR	CHEMBL669873	21	6	10.6
DHFR	CHEMBL669986	76	4.51	8.7
DHFR	CHEMBL2384165	28	4.49	8.77
Estrogen	CHEMBL832786	23	5.78	8.82
Estrogen	CHEMBL4024644	43	5.11	9
Estrogen	CHEMBL1909145	32	4.75	9.89
Estrogen	CHEMBL3776150	24	5.96	9.16
Estrogen	CHEMBL3429911	34	4.97	8.57
Estrogen	CHEMBL4200119	24	5.03	8.7
Estrogen	CHEMBL676818	27	5.45	8.89
Estrogen	CHEMBL3429910	35	5.24	9.82
Estrogen	CHEMBL679891	29	6.25	9.3
Carbonic	CHEMBL3122609	22	5.06	8.25
Carbonic	CHEMBL1646987	29	5.02	8.68
Carbonic	CHEMBL658942	43	6.1	9.3
Carbonic	CHEMBL3879546	25	5.75	9.3
Carbonic	CHEMBL657192	28	5.05	9.15
Carbonic	CHEMBL4017666	23	6.02	9.03
Carbonic	CHEMBL1693045	28	5.02	8.68
Carbonic	CHEMBL657145	44	6.52	9.7
Carbonic	CHEMBL3269990	45	4.55	8.15
Carbonic	CHEMBL3377547	21	5.29	8.36
Carbonic	CHEMBL3366665	21	5.29	8.36
D2	CHEMBL872881	27	5.26	8.7
D2	CHEMBL1061432	20	4.27	7.39
D2	CHEMBL673073	28	4.34	7.41
D2	CHEMBL3801477	25	4.07	8.15
D2	CHEMBL948052	21	6.04	9.34
D2	CHEMBL866800	26	4.17	7.55
D2	CHEMBL3614290	35	6	9.32
D2	CHEMBL1909140	100	4.7	9.31
D2	CHEMBL3089794	24	4.87	7.98
D2	CHEMBL4046973	21	4.85	9.28

D2	CHEMBL3073105	38	5.85	9.72
D2	CHEMBL4046974	22	5.05	9.42
D2	CHEMBL866799	26	4.19	7.39
D2	CHEMBL671368	29	5.1	8.64
erbB1	CHEMBL677204	32	4	10.54
erbB1	CHEMBL1798351	101	4.05	8.02
erbB1	CHEMBL680017	63	4.92	10.6
erbB1	CHEMBL3616083	26	5.28	10.54
erbB1	CHEMBL1686143	30	4.33	9
erbB1	CHEMBL677880	34	4.11	8.4
erbB1	CHEMBL3877875	20	5	9
erbB1	CHEMBL3240253	20	5.26	9
erbB1	CHEMBL679944	48	5.37	10.6
erbB1	CHEMBL1064820	23	4.39	7.52
erbB1	CHEMBL2439336	26	4.47	8.92
erbB1	CHEMBL674019	23	5.11	8.57
erbB1	CHEMBL2066000	25	4.1	7.21
erbB1	CHEMBL925198	23	5.14	8.74
erbB1	CHEMBL680021	43	5.69	10.54
erbB1	CHEMBL3614855	25	5.11	8.14
erbB1	CHEMBL1064829	32	4.36	7.7
erbB1	CHEMBL4052480	21	4.95	9.42
erbB1	CHEMBL3238947	39	5.19	9.05
erbB1	CHEMBL674637	22	5.28	9.35
erbB1	CHEMBL1073378	26	5.21	8.7
Glucocorticoid	CHEMBL941996	43	5.35	9.4
Glucocorticoid	CHEMBL3868161	35	7	10.52
Glucocorticoid	CHEMBL1909150	76	4.38	9.18
Glucocorticoid	CHEMBL679612	44	4.6	8.24
Glucocorticoid	CHEMBL890119	32	5.16	8.47
Ephrin	CHEMBL998745	25	5.51	8.52
Ephrin	CHEMBL1686716	37	5	8.4
Ephrin	CHEMBL1068887	59	4.6	9.9
Ephrin	CHEMBL915413	29	5.14	8.51
Ephrin	CHEMBL945185	27	5.22	8.3
Ephrin	CHEMBL3707854	22	5.19	9
Ephrin	CHEMBL1101480	28	5.45	8.52
Ephrin	CHEMBL1120739	36	5.65	9
Ephrin	CHEMBL932888	33	5.08	8.36
Ephrin	CHEMBL1816473	33	4.85	8.1
Ephrin	CHEMBL1047378	33	5.71	9
COX1	CHEMBL763480	23	4.52	7.7
COX1	CHEMBL1909130	45	4.5	8.35

Glycogen	CHEMBL3413879	26	6.14	9.54
Glycogen	CHEMBL1102983	30	5.09	9.4
Glycogen	CHEMBL1113275	51	4.01	8.22
Glycogen	CHEMBL684976	60	4.39	8
Glycogen	CHEMBL2182637	30	6.16	9.66
Glycogen	CHEMBL928210	36	5.38	8.52
Glycogen	CHEMBL4057839	32	6.04	10
Glycogen	CHEMBL1034478	33	3.38	10.05
Glycogen	CHEMBL2390763	31	6.39	10.89
Glycogen	CHEMBL1249829	20	5.7	9.24
Glycogen	CHEMBL963374	40	5.14	9.64
Glycogen	CHEMBL3097577	25	5.01	8.05
Aurora-A	CHEMBL972583	22	5.14	8.4
Aurora-A	CHEMBL993738	44	5.5	9.22
Aurora-A	CHEMBL1664805	58	4.54	8.7
Aurora-A	CHEMBL3888080	73	4.5	9.1
Aurora-A	CHEMBL3862723	42	4.12	10.4
Aurora-A	CHEMBL2173311	59	4.5	9.1
Aurora-A	CHEMBL856402	41	5.7	9
B-raf	CHEMBL1780874	28	6.07	9.24
B-raf	CHEMBL3888085	55	6.07	9.14
B-raf	CHEMBL1217119	83	4.77	8.7
B-raf	CHEMBL1104358	45	4.16	7.85
B-raf	CHEMBL1067537	64	4.03	8.3
B-raf	CHEMBL1671058	20	5.85	9.89
B-raf	CHEMBL1953548	22	4.6	9
B-raf	CHEMBL3816885	30	4.5	9.3
B-raf	CHEMBL3240201	24	6.07	9.7
B-raf	CHEMBL2341125	47	5.15	8.4
B-raf	CHEMBL3887740	43	5.75	9
B-raf	CHEMBL981002	32	4.49	7.92
B-raf	CHEMBL3804129	36	6	9
B-raf	CHEMBL3804126	37	6.38	9.4
B-raf	CHEMBL1043507	38	6.52	9.52
B-raf	CHEMBL1008210	37	6.01	10.7
B-raf	CHEMBL1937974	35	4.81	9
B-raf	CHEMBL1671059	20	4.12	8.3
B-raf	CHEMBL946282	31	5.05	9
Vanilloid	CHEMBL832934	26	5.07	8.07
Vanilloid	CHEMBL820100	28	5.84	9.52
Vanilloid	CHEMBL828688	20	4.76	9.4
Vanilloid	CHEMBL3705949	137	4.65	8.7
Vanilloid	CHEMBL3706060	41	4.48	7.7

Vanilloid	CHEMBL930279	29	4.82	8.1
Androgen	CHEMBL1960482	23	5.27	9.33
Androgen	CHEMBL891115	42	6	9.7
Androgen	CHEMBL930493	21	6.32	10.3
Androgen	CHEMBL933063	31	5.39	9
Androgen	CHEMBL930492	23	5.63	9.7
Androgen	CHEMBL859226	24	5.34	8.57
Androgen	CHEMBL1169306	41	5.34	8.64
Coagulation	CHEMBL811970	20	4.92	8.4
Coagulation	CHEMBL862522	23	4.7	7.8
Coagulation	CHEMBL1010200	23	4.57	8
Coagulation	CHEMBL832832	63	4.09	8.52
Coagulation	CHEMBL816070	38	4.03	8.33
Coagulation	CHEMBL847937	27	6.8	9.97
Coagulation	CHEMBL814535	22	4.05	8.4
Coagulation	CHEMBL813569	29	6.98	10.82
Coagulation	CHEMBL817243	24	6.25	10.7
Coagulation	CHEMBL860417	24	5.75	9
Coagulation	CHEMBL815806	22	4.38	8.7
Coagulation	CHEMBL814972	22	5.34	10.34
Coagulation	CHEMBL1049043	21	4.77	8.8
Coagulation	CHEMBL884569	35	4.8	10.4
Coagulation	CHEMBL811982	22	4.11	7.6
Coagulation	CHEMBL814951	21	4.22	8.22
Coagulation	CHEMBL1067528	35	4.03	8.4
Coagulation	CHEMBL838818	60	5.15	9.92
Coagulation	CHEMBL845034	23	4.62	8.27
Coagulation	CHEMBL3424914	33	4.12	8.96
Coagulation	CHEMBL813626	41	6.03	10.89
Coagulation	CHEMBL813672	23	5.43	9.09
Coagulation	CHEMBL829646	22	6.82	10.4
Coagulation	CHEMBL812835	20	4.3	7.92
Coagulation	CHEMBL812837	30	5.85	9.57
Coagulation	CHEMBL811969	34	6.33	10
A2a	CHEMBL1259385	22	7.26	10.16
Opiod	CHEMBL1769429	31	5.57	10.21
Opiod	CHEMBL757060	35	5.3	9.05
Opiod	CHEMBL1112210	35	5.5	8.96
Opiod	CHEMBL756359	45	5.37	9.85
Opiod	CHEMBL888956	45	4.22	8.05
Opiod	CHEMBL858889	24	5.96	11
Opiod	CHEMBL754712	22	4.71	8.44
Opiod	CHEMBL3705573	20	5.24	8.87

Opiod	CHEMBL901089	26	6	9.82
Opiod	CHEMBL1050689	40	6.44	10.59
Opiod	CHEMBL3707592	76	5.76	10.4
Opiod	CHEMBL757059	36	5.18	9.05
Opiod	CHEMBL757823	24	6.05	9.19
Opiod	CHEMBL746681	22	5.38	8.82
Opiod	CHEMBL3888044	27	4.85	8.07
Opiod	CHEMBL3888831	88	4.99	9.46
Opiod	CHEMBL2050632	40	6.82	10.21
MAO-A	CHEMBL827909	42	4.19	7.4
MAO-A	CHEMBL2060936	51	4.58	8.25
MAO-A	CHEMBL1648130	38	4.03	7.21
MAO-A	CHEMBL965566	27	4.09	8.42
Caspase	CHEMBL831642	28	4.2	8.4
Caspase	CHEMBL658912	27	4.33	8.6
Caspase	CHEMBL1838643	24	4.09	7.92
Caspase	CHEMBL1012730	27	4.21	8.62
Caspase	CHEMBL658623	25	4.01	7.57
JAK2	CHEMBL3404501	25	5.54	9
JAK2	CHEMBL2395112	22	5.91	8.92
JAK2	CHEMBL1044318	37	6.06	9.7
JAK2	CHEMBL1049517	47	5.35	9.3
JAK2	CHEMBL1011746	23	3.84	7.41
JAK2	CHEMBL2186596	31	6.99	10
JAK2	CHEMBL1936085	21	5.54	10.05
JAK2	CHEMBL2067611	27	5.54	9.19
JAK2	CHEMBL3374493	34	5.5	8.7
JAK2	CHEMBL2438691	32	5.5	9.22
JAK2	CHEMBL1291235	25	4.54	9
CDK2	CHEMBL1250281	25	5.16	9
CDK2	CHEMBL971698	29	3.73	8.52
CDK2	CHEMBL661124	22	5.4	8.68
CDK2	CHEMBL2166172	20	4.21	8.52
CDK2	CHEMBL908827	20	5.2	8.7
Cannabinoid	CHEMBL2187925	27	4.68	8.89
Cannabinoid	CHEMBL2154764	32	5.45	8.61
Cannabinoid	CHEMBL886398	38	5.4	9.05
Cannabinoid	CHEMBL901144	23	5.1	9.1
Cannabinoid	CHEMBL1107820	29	6.13	9.82
Cannabinoid	CHEMBL1011882	62	5.01	9.7
Cannabinoid	CHEMBL2317164	35	4.05	9.07
Cannabinoid	CHEMBL896457	34	5.41	8.77
Cannabinoid	CHEMBL1804296	35	6.21	9.92

Cannabinoid	CHEMBL2210669	25	5.37	8.92
Cannabinoid	CHEMBL887035	29	5.07	8.52
Cannabinoid	CHEMBL901108	30	5.9	9.49
Cannabinoid	CHEMBL832272	36	4.72	8.4
Cannabinoid	CHEMBL932400	32	5.82	9.52
HERG	CHEMBL1676103	146	4	8.66
HERG	CHEMBL1036929	26	4.32	8.6
HERG	CHEMBL2330781	36	4.51	8.39
HERG	CHEMBL2148626	30	5.33	9.12
HERG	CHEMBL3096732	44	4.84	9.4
HERG	CHEMBL1764797	24	4.69	8.7
HERG	CHEMBL3583383	42	4.02	7.3
HERG	CHEMBL827807	49	4	8.82
HERG	CHEMBL766816	29	4.12	8.52
ABL1	CHEMBL3888134	50	5.95	9.3
ABL1	CHEMBL2186853	38	5.67	9.03
ABL1	CHEMBL2353816	50	5.83	10.82
ABL1	CHEMBL923929	25	4.99	8.92
LCK	CHEMBL1113978	53	5.35	8.52
LCK	CHEMBL4145903	26	4.15	9
LCK	CHEMBL913225	34	6.53	11
LCK	CHEMBL841202	44	4.64	8.4
LCK	CHEMBL841204	31	5.7	9.15
LCK	CHEMBL1049046	45	5.57	10.15
LCK	CHEMBL910552	51	4.55	9.7
LCK	CHEMBL864124	29	5.22	8.52
LCK	CHEMBL908547	26	5.06	8.52
LCK	CHEMBL842071	35	5.26	8.7

Availability of data and methods

QPHAR: quantitative pharmacophore activity relationship: method and validation

All the data generated and used in this study can be found in the author's GitHub repository: https://github.com/StefanKohlbacher/QuantPharmacophore
The data includes the training and validation set for comparison against the PHASE algorithm. Additionally, the datasets generated for cross-validation are provided as well as some meta-information on these datasets.

The developed QPhAR method is provided as a docker container and can be installed from the repository by executing the *setup.sh* script (docker needs to be successfully installed for the setup to finish without errors). It is recommended to specify a volume on installation that is then shared with the docker container for simplified data exchange between the host's PC and the container. Otherwise, data can easily be transferred in and out of the docker container via docker's *cp* command. Five scripts are provided in the container for training and using the QPhAR method:

- checkTrainingData: Checks whether the training set fulfils a certain set of requirements. This script is provided for the user's convenience to estimate whether the given dataset can be used to successfully train a QPhAR model.
- splitData: A convenience script to split training and test data.
- train: Trains a QPhAR model on a dataset with the specified parameters. When the model is successfully trained it is saved in the specified location.
- gridSearch: Given a set of parameters, this script performs a grid-search over all possible combinations obtained from the given parameters. All models including the predicted validation set and the model's performance are saved in the specified location.
- predict: Given a dataset and the location of a previously trained QPhAR model, this script will predict the activity for a specified dataset. This script can be used to validate the model obtained from the training script. An SD-File containing the predicted values is generated for a set of molecules. For a set of pharmacophores, a JSON file is generated with the pharmacophore's predictions.

A detailed guide to using the scripts can be found in the GitHub repository.

A new set of KNIME nodes implementing the QPhAR algorithm

All the data generated and used in this study can be found in the author's GitHub repository: https://github.com/StefanKohlbacher/qphar-knime-nodes
The data generated and used includes a KNIME workflow that should serve as a template for training a QPhAR model. The workflow implements best practices and commonly applied methods such as hyperparameter optimization and data-splitting and can be used as a template to train a QPhAR model in KNIME. The developed KNIME nodes are available in addition to the workflow. These include the QPhAR Reader, QPhAR Learner, QPhAR Predictor, and QPhAR Writer. They are available for download as part of the NeuroDeRisk toolbox.

During this study, a set of GABA-A QPhAR models were trained and validated. These models are available for download and can be used along with the KNIME nodes, by reading the model with the QPhAR Reader and then predicting a set of molecules with the QPhAR Predictor.

Applications of the novel quantitative pharmacophore activity relationship method QPhAR in virtual screening and lead-optimization

All the data generated and used in this study can be found in the author's GitHub repository:
https://github.com/StefanKohlbacher/qphar-applications
The data provided contains the data for validating the provided method, as well as the supplementary material of the manuscript[95] and a set of scripts to apply the presented method. In order to run the scripts the QPhAR method needs to be installed on the user's machine, as explained in the README.md file of the repository.
Besides scripts for replicating the study, the repository also contains scripts to generate a refined pharmacophore (*makeOptimalPharmacophore.py*), analyse its performance, and finally generate the activity grids as described in the manuscript. Generating the activity scripts for a set of molecules based on a previously trained QPhAR model will yield a set of .kont-files. These can be opened and visualized by LigandScout[8].

List of abbreviations

CADD: computer-assisted drug discovery
VS: virtual screening
SBVS: structure-based virtual screening
LBVS: ligand-based virtual screening
SAR: structure-activity relationship
QSAR: quantitative structure-activity relationship
QPhAR: quantitative pharmacophore-activity relationship
ML: machine learning
RF: random forest
AD: applicability domain
PLS: partial least squares
NDR: NeuroDeRisk
ADMET: absorption distribution metabolism excretion and toxicity

References

1. Pammolli F, Magazzini L, Riccaboni M (2011) The productivity crisis in pharmaceutical R&D. Nat Rev Drug Discov 10:428–438. https://doi.org/10.1038/nrd3405
2. Dickson M, Gagnon JP (2009) The Cost of New Drug Discovery and Development. Discov Med 4:172–179
3. Macarron R, Banks MN, Bojanic D, et al (2011) Impact of high-throughput screening in biomedical research. Nat Rev Drug Discov 10:188–195. https://doi.org/10.1038/nrd3368
4. Moffat JG, Rudolph J, Bailey D (2014) Phenotypic screening in cancer drug discovery — past, present and future. Nat Rev Drug Discov 13:588–602. https://doi.org/10.1038/nrd4366
5. Hopfinger AJ (2002) Computer-assisted drug design. In: ACS Publ. https://pubs.acs.org/doi/pdf/10.1021/jm00147a001. Accessed 23 Jul 2022
6. Wermuth CG, Ganellin CR, Lindberg P, Mitscher LA (1998) Glossary of terms used in medicinal chemistry (IUPAC Recommendations 1998). Pure Appl Chem 70:1129–1143. https://doi.org/10.1351/pac199870051129
7. Hou X, Du J, Liu R, et al (2015) Enhancing the Sensitivity of Pharmacophore-Based Virtual Screening by Incorporating Customized ZBG Features: A Case Study Using Histone Deacetylase 8. J Chem Inf Model 55:861–871. https://doi.org/10.1021/ci500762z
8. Wolber G, Langer T (2005) LigandScout: 3-D Pharmacophores Derived from Protein-Bound Ligands and Their Use as Virtual Screening Filters. J Chem Inf Model 45:160–169. https://doi.org/10.1021/ci049885e
9. Khan HN, Kulsoom S, Rashid H (2012) Ligand based pharmacophore model development for the identification of novel antiepileptic compound. Epilepsy Res 98:62–71. https://doi.org/10.1016/j.eplepsyres.2011.08.016
10. A. Sanders MP, McGuire R, Roumen L, et al (2012) From the protein's perspective: the benefits and challenges of protein structure-based pharmacophore modeling. MedChemComm 3:28–38. https://doi.org/10.1039/C1MD00210D
11. Leach AR, Gillet VJ, Lewis RA, Taylor R (2010) Three-Dimensional Pharmacophore Methods in Drug Discovery. J Med Chem 53:539–558. https://doi.org/10.1021/jm900817u
12. Wolber G, Seidel T, Bendix F, Langer T (2008) Molecule-pharmacophore superpositioning and pattern matching in computational drug design. Drug Discov Today 13:23–29. https://doi.org/10.1016/j.drudis.2007.09.007
13. Kutlushina A, Khakimova A, Madzhidov T, Polishchuk P (2018) Ligand-Based Pharmacophore Modeling Using Novel 3D Pharmacophore Signatures. Molecules 23:3094. https://doi.org/10.3390/molecules23123094
14. Bender A, Glen RC (2004) Molecular similarity: a key technique in molecular informatics. Org Biomol Chem 2:3204–3218. https://doi.org/10.1039/B409813G
15. Langer T, Hoffmann RD (2001) Virtual screening: an effective tool for lead structure discovery? Curr Pharm Des 7:509–527. https://doi.org/10.2174/1381612013397861
16. Seifert MHJ, Wolf K, Vitt D (2003) Virtual high-throughput in silico screening. BIOSILICO 1:143–149. https://doi.org/10.1016/S1478-5382(03)02359-X
17. Phatak SS, Stephan CC, Cavasotto CN (2009) High-throughput and in silico screenings in drug discovery. Expert Opin Drug Discov 4:947–959. https://doi.org/10.1517/17460440903190961
18. Lee H-C, Salzemann J, Jacq N, et al (2006) Grid-Enabled High-Throughput In Silico Screening

Against Influenza A Neuraminidase. IEEE Trans NanoBioscience 5:288–295. https://doi.org/10.1109/TNB.2006.887943
19. Kirchmair J, Distinto S, Schuster D, et al (2008) Enhancing Drug Discovery Through In Silico Screening: Strategies to Increase True Positives Retrieval Rates. Curr Med Chem 15:2040–2053. https://doi.org/10.2174/092986708785132843
20. Lyne PD (2002) Structure-based virtual screening: an overview. Drug Discov Today 7:1047–1055. https://doi.org/10.1016/S1359-6446(02)02483-2
21. Ripphausen P, Nisius B, Bajorath J (2011) State-of-the-art in ligand-based virtual screening. Drug Discov Today 16:372–376. https://doi.org/10.1016/j.drudis.2011.02.011
22. Morris GM, Lim-Wilby M (2008) Molecular Docking. In: Kukol A (ed) Molecular Modeling of Proteins. Humana Press, Totowa, NJ, pp 365–382
23. Karplus M, McCammon JA (2002) Molecular dynamics simulations of biomolecules. Nat Struct Biol 9:646–652. https://doi.org/10.1038/nsb0902-646
24. Krüger DM, Evers A (2010) Comparison of Structure- and Ligand-Based Virtual Screening Protocols Considering Hit List Complementarity and Enrichment Factors. ChemMedChem 5:148–158. https://doi.org/10.1002/cmdc.200900314
25. Chen D, Ranganathan A, IJzerman AP, et al (2013) Complementarity between in Silico and Biophysical Screening Approaches in Fragment-Based Lead Discovery against the A2A Adenosine Receptor. J Chem Inf Model 53:2701–2714. https://doi.org/10.1021/ci4003156
26. Stahura FL, Bajorath J (2005) New Methodologies for Ligand-Based Virtual Screening. Curr Pharm Des 11:1189–1202. https://doi.org/10.2174/1381612053507549
27. Hamza A, Wei N-N, Zhan C-G (2012) Ligand-Based Virtual Screening Approach Using a New Scoring Function. J Chem Inf Model 52:963–974. https://doi.org/10.1021/ci200617d
28. Shirvanyants D, Alexandrova AN, Dokholyan NV (2011) Rigid substructure search. Bioinformatics 27:1327–1329. https://doi.org/10.1093/bioinformatics/btr129
29. Barnard JM (2002) Substructure searching methods: Old and new. In: ACS Publ. https://pubs.acs.org/doi/pdf/10.1021/ci00014a001. Accessed 24 Sep 2022
30. Cereto-Massagué A, Ojeda MJ, Valls C, et al (2015) Molecular fingerprint similarity search in virtual screening. Methods 71:58–63. https://doi.org/10.1016/j.ymeth.2014.08.005
31. Chen B, Harrison RF, Papadatos G, et al (2007) Evaluation of machine-learning methods for ligand-based virtual screening. J Comput Aided Mol Des 21:53–62. https://doi.org/10.1007/s10822-006-9096-5
32. Js M, Ac G, Ej M (2001) 3-D pharmacophores in drug discovery. Curr Pharm Des 7:. https://doi.org/10.2174/1381612013397843
33. Keserű GM, Makara GM (2006) Hit discovery and hit-to-lead approaches. Drug Discov Today 11:741–748. https://doi.org/10.1016/j.drudis.2006.06.016
34. Pickett SD, McLay IM, Clark DE (2000) Enhancing the Hit-to-Lead Properties of Lead Optimization Libraries. J Chem Inf Comput Sci 40:263–272. https://doi.org/10.1021/ci990261w
35. Schneider G, Neidhart W, Giller T, Schmid G (1999) "Scaffold-Hopping" by Topological Pharmacophore Search: A Contribution to Virtual Screening. Angew Chem Int Ed 38:2894–2896. https://doi.org/10.1002/(SICI)1521-3773(19991004)38:19<2894::AID-ANIE2894>3.0.CO;2-F
36. Irwin JJ (2008) Community benchmarks for virtual screening. J Comput Aided Mol Des 22:193–199. https://doi.org/10.1007/s10822-008-9189-4
37. Kontoyianni M, Sokol GS, MCclellan LM (2005) Evaluation of library ranking efficacy in virtual screening. J Comput Chem 26:11–22. https://doi.org/10.1002/jcc.20141

38. Wilton D, Willett P, Lawson K, Mullier G (2003) Comparison of Ranking Methods for Virtual Screening in Lead-Discovery Programs. J Chem Inf Comput Sci 43:469–474. https://doi.org/10.1021/ci025586i
39. Scior T, Bender A, Tresadern G, et al (2012) Recognizing Pitfalls in Virtual Screening: A Critical Review. J Chem Inf Model 52:867–881. https://doi.org/10.1021/ci200528d
40. Wieder M, Garon A, Perricone U, et al (2017) Common Hits Approach: Combining Pharmacophore Modeling and Molecular Dynamics Simulations. J Chem Inf Model 57:365–385. https://doi.org/10.1021/acs.jcim.6b00674
41. Gramatica P (2007) Principles of QSAR models validation: internal and external. QSAR Comb Sci 26:694–701. https://doi.org/10.1002/qsar.200610151
42. Tropsha A (2010) Best Practices for QSAR Model Development, Validation, and Exploitation. Mol Inform 29:476–488. https://doi.org/10.1002/minf.201000061
43. Cherkasov A, Muratov EN, Fourches D, et al (2014) QSAR Modeling: Where Have You Been? Where Are You Going To? J Med Chem 57:4977–5010. https://doi.org/10.1021/jm4004285
44. Hansch C, Maloney PP, Fujita T, Muir RM (1962) Correlation of Biological Activity of Phenoxyacetic Acids with Hammett Substituent Constants and Partition Coefficients. Nature 194:178–180. https://doi.org/10.1038/194178b0
45. Montgomery DC, Peck EA, Vining GG (2012) Introduction to linear regression analysis, 5th ed. Wiley, Hoboken, NJ
46. Dunn WJ, Block JH, Pearlman RS (1986) Partition coefficient: Determination and estimation
47. Consonni V, Todeschini R (2010) Molecular Descriptors. In: Puzyn T, Leszczynski J, Cronin MT (eds) Recent Advances in QSAR Studies: Methods and Applications. Springer Netherlands, Dordrecht, pp 29–102
48. Xue L, Bajorath J (2000) Molecular Descriptors in Chemoinformatics, Computational Combinatorial Chemistry, and Virtual Screening. Comb Chem High Throughput Screen 3:363–372. https://doi.org/10.2174/1386207003331454
49. Mauri A, Consonni V, Todeschini R (2017) Molecular Descriptors. Handb Comput Chem 2065
50. Kubinyi H (1997) QSAR and 3D QSAR in drug design Part 1: methodology. Drug Discov Today 2:457–467. https://doi.org/10.1016/S1359-6446(97)01079-9
51. Hawkins DM (2004) The Problem of Overfitting. J Chem Inf Comput Sci 44:1–12. https://doi.org/10.1021/ci0342472
52. Patani GA, LaVoie EJ (1996) Bioisosterism: A Rational Approach in Drug Design. Chem Rev 96:3147–3176. https://doi.org/10.1021/cr950066q
53. Subbaiah MAM, Meanwell NA (2021) Bioisosteres of the Phenyl Ring: Recent Strategic Applications in Lead Optimization and Drug Design. J Med Chem 64:14046–14128. https://doi.org/10.1021/acs.jmedchem.1c01215
54. Rogers D, Hahn M (2010) Extended-Connectivity Fingerprints. J Chem Inf Model 50:742–754. https://doi.org/10.1021/ci100050t
55. Gedeck P, Rhode B, Bartels C (2006) QSAR - How good is it in practice? Comparison of descriptor sets on an unbiased cross section of corporate data sets. J Chem Inf Model 1924–1936
56. Golbraikh A, Tropsha A (2003) QSAR Modeling Using Chirality Descriptors Derived from Molecular Topology. J Chem Inf Comput Sci 43:144–154. https://doi.org/10.1021/ci025516b
57. Prasanna S, Doerksen R (2009) Topological Polar Surface Area: A Useful Descriptor in 2D-QSAR. Curr Med Chem 16:21–41. https://doi.org/10.2174/092986709787002817
58. Honma M, Kitazawa A, Cayley A, et al (2019) Improvement of quantitative structure–activity

relationship (QSAR) tools for predicting Ames mutagenicity: outcomes of the Ames/QSAR International Challenge Project. Mutagenesis 34:3–16. https://doi.org/10.1093/mutage/gey031
59. Saito H, Osumi M, Hirano H, et al (2009) Technical Pitfalls and Improvements for High-speed Screening and QSAR Analysis to Predict Inhibitors of the Human Bile Salt Export Pump (ABCB11/BSEP). AAPS J 11:581. https://doi.org/10.1208/s12248-009-9137-9
60. Dietterich TG (1990) Machine Learning. Annu Rev Comput Sci 4:255–306. https://doi.org/10.1146/annurev.cs.04.060190.001351
61. Mahesh B (2019) Machine Learning Algorithms -A Review
62. Tarca AL, Carey VJ, Chen X, et al (2007) Machine Learning and Its Applications to Biology. PLOS Comput Biol 3:e116. https://doi.org/10.1371/journal.pcbi.0030116
63. Maglogiannis IG (2007) Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies. IOS Press
64. Mayr A, Klambauer G, Unterthiner T, Hochreiter S (2016) DeepTox: Toxicity Prediction using Deep Learning. Front Environ Sci 3:. https://doi.org/10.3389/fenvs.2015.00080
65. Greene N (2002) Computer systems for the prediction of toxicity: an update. Adv Drug Deliv Rev 54:417–431. https://doi.org/10.1016/S0169-409X(02)00012-1
66. Saxena A, Prasad M, Gupta A, et al (2017) A review of clustering techniques and developments. Neurocomputing 267:664–681. https://doi.org/10.1016/j.neucom.2017.06.053
67. Goldstein BA, Navar AM, Carter RE (2017) Moving beyond regression techniques in cardiovascular risk prediction: applying machine learning to address analytic challenges. Eur Heart J 38:1805–1814. https://doi.org/10.1093/eurheartj/ehw302
68. Verrelst J, Muñoz J, Alonso L, et al (2012) Machine learning regression algorithms for biophysical parameter retrieval: Opportunities for Sentinel-2 and -3. Remote Sens Environ 118:127–139. https://doi.org/10.1016/j.rse.2011.11.002
69. Zhu D, Cai C, Yang T, Zhou X (2018) A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization. Big Data Cogn Comput 2:5. https://doi.org/10.3390/bdcc2010005
70. Schölkopf B, Schölkopf D of the MPI for I in TGP for MLB, Smola AJ, Bach F (2002) Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press
71. Bishop CM (1994) Neural networks and their applications. Rev Sci Instrum 65:1803–1832. https://doi.org/10.1063/1.1144830
72. Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. Psychol Rev 65:386–408. https://doi.org/10.1037/h0042519
73. Svetnik V, Liaw A, Tong C, et al (2003) Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. J Chem Inf Comput Sci 43:1947–1958. https://doi.org/10.1021/ci034160g
74. Belgiu M, Drăguţ L (2016) Random forest in remote sensing: A review of applications and future directions. ISPRS J Photogramm Remote Sens 114:24–31. https://doi.org/10.1016/j.isprsjprs.2016.01.011
75. Breiman L (2001) Random Forests. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324
76. Probst P, Wright MN, Boulesteix A-L (2019) Hyperparameters and tuning strategies for random forest. WIREs Data Min Knowl Discov 9:e1301. https://doi.org/10.1002/widm.1301
77. Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106.

https://doi.org/10.1007/BF00116251
78. Cramer GM, Ford RA, Hall RL (1976) Estimation of toxic hazard—A decision tree approach. Food Cosmet Toxicol 16:255–276. https://doi.org/10.1016/S0015-6264(76)80522-6
79. Raschka S (2020) Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning
80. Roy K, Kar S, Ambure P (2015) On a simple approach for determining applicability domain of QSAR models. Chemom Intell Lab Syst 145:22–29. https://doi.org/10.1016/j.chemolab.2015.04.013
81. Netzeva TI, Worth AP, Aldenberg T, et al (2005) Current Status of Methods for Defining the Applicability Domain of (Quantitative) Structure-Activity Relationships: The Report and Recommendations of ECVAM Workshop 52 ′. Altern Lab Anim 33:155–173. https://doi.org/10.1177/026119290503300209
82. Sahigara F, Mansouri K, Ballabio D, et al (2012) Comparison of Different Approaches to Define the Applicability Domain of QSAR Models. Molecules 17:4791–4810. https://doi.org/10.3390/molecules17054791
83. Jaworska J, Nikolova-Jeliazkova N, Aldenberg T (2005) QSAR Applicability Domain Estimation by Projection of the Training Set in Descriptor Space: A Review. Altern Lab Anim 33:445–459. https://doi.org/10.1177/026119290503300508
84. Krumhansl CL (1978) Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. Psychol Rev 85:445–463. https://doi.org/10.1037/0033-295X.85.5.445
85. Mathea M, Klingspohn W, Baumann K (2016) Chemoinformatic Classification Methods and their Applicability Domain. Mol Inform 35:160–180. https://doi.org/10.1002/minf.201501019
86. Dimitrov S, Dimitrova G, Pavlov T, et al (2005) A Stepwise Approach for Defining the Applicability Domain of SAR and QSAR Models. J Chem Inf Model 45:839–849. https://doi.org/10.1021/ci0500381
87. Li H, Sutter J, Hoffman R (2000) HypoGen: An Automated System for Generating 3D Predictive Pharmacophore Models. In: Pharmacophore Perception, Development and Use in Drug Design; Guner, O., Ed.; International University Line: La Jolla, CA,
88. (2019) BIOVIA - Scientific Enterprise Software for Chemical Research, Material Science R&D. https://www.3dsbiovia.com/. Accessed 11 Jul 2019
89. Dixon SL, Smondyrev AM, Knoll EH, et al (2006) PHASE: a new engine for pharmacophore perception, 3D QSAR model development, and 3D database screening: 1. Methodology and preliminary results. J Comput Aided Mol Des 20:647–671. https://doi.org/10.1007/s10822-006-9087-6
90. Schrödinger. https://www.schrodinger.com/. Accessed 25 Mar 2021
91. Geladi P, Kowalski BR (1986) Partial least-squares regression: a tutorial. Anal Chim Acta 185:1–17. https://doi.org/10.1016/0003-2670(86)80028-9
92. NeuroDeRisk (2019) NeuroDeRisk - Neurotoxicity De-Risking in Preclinical Drug Discovery. https://neuroderisk.eu/. Accessed 17 Mar 2021
93. Kohlbacher SM, Langer T, Seidel T (2021) QPHAR: quantitative pharmacophore activity relationship: method and validation. J Cheminformatics 13:57. https://doi.org/10.1186/s13321-021-00537-9
94. Kohlbacher SM, Ibis G, Permann C, et al (2022) A new set of KNIME nodes implementing the QPhAR algorithm. Mol Inform. Under Review

95. Kohlbacher SM, Schmid M, Seidel T, Langer T (2022) Applications of the Novel Quantitative Pharmacophore Activity Relationship Method QPhAR in Virtual Screening and Lead-Optimisation. Pharmaceuticals 15:1122. https://doi.org/10.3390/ph15091122

Abstract

As drug development becomes increasingly complex, more efficient discovery methods and tools are required. CADD has become an integral part of every drug discovery process, facilitating the search for novel drugs from a vast space of available molecules. Pharmacophores, virtual screening, machine learning, and QSAR are just a few tools and methods commonly applied in the drug discovery process. However, hardly any research has been conducted on combining these concepts and methods. The work done for this thesis explored the application of pharmacophores in QSAR studies. It could be shown that QPhAR, quantitative pharmacophore activity relationship, is a viable alternative to conventional QSAR methods leveraging concepts such as scaffold-hopping to increase the model's confidence and expand the model's applicability domain. Building on the newly developed concept of QPhAR, it is shown that well-trained QPhAR models can facilitate the pharmacophore modelling process and enable the modern medicinal chemist in his role as a decision maker. For this, a method was developed to extract pharmacophores from QPhAR models that show strong performance in distinguishing between higher and less active molecules. The obtained pharmacophores can then be used as a query for virtual pharmacophore screening to obtain a set of hit compounds, which can be ranked by the initially obtained QPhAR model. Finally, the QPhAR model can be used to visualise regions of promising hit or lead compounds and guide the medicinal chemist with insights for further optimisations.

Zusammenfassung

Da die Arzneimittelentwicklung immer komplexer wird, sind effizientere Entdeckungsmethoden und -werkzeuge erforderlich. CADD ist zu einem integralen Bestandteil jedes Arzneimittelentdeckungsprozesses geworden und erleichtert die Suche nach neuartigen Arzneimitteln aus einem riesigen Bereich verfügbarer Moleküle. Pharmakophore, virtuelles Screening, maschinelles Lernen und QSAR sind nur einige Werkzeuge und Methoden, die häufig im Arzneimittelforschungsprozess eingesetzt werden. Die Kombination dieser Konzepte und Methoden ist jedoch kaum erforscht. Die Arbeit dieser Dissertation untersuchte die Anwendung von Pharmakophoren in QSAR-Studien. Es konnte gezeigt werden, dass QPhAR, quantitative Pharmacophore Aktivitäts Beziehung, eine praktikable Alternative zu herkömmlichen QSAR-Methoden ist, wobei Konzepte wie Scaffold-Hopping genutzt werden, um das Vertrauen in die Vorhersagekraft des Modells zu erhöhen und dessen Anwendungsbereich zu erweitern. Aufbauend auf dem neu entwickelten Konzept von QPhAR wurde gezeigt, dass gut trainierte QPhAR-Modelle den Pharmakophor-Modellierungsprozess erleichtern und den modernen medizinischen Chemiker in seiner Rolle als Entscheidungsträger unterstützen können. Dafür wurde eine Methode entwickelt, um Pharmakophore aus QPhAR-Modellen zu extrahieren, die eine sehr gute Leistung bei der Unterscheidung zwischen höher und weniger aktiven Molekülen liefern. Die erhaltenen Pharmakophore können für virtuelles Pharmakophor-Screening verwendet werden, um eine Reihe von Trefferverbindungen zu erhalten, die dann nach dem ursprünglich erhaltenen QPhAR-Modell gereiht werden. Weiters kann das QPhAR-Modell verwendet werden, um vielversprechenden Hit- oder Lead-Verbindungen zu analysieren und auf Basis der erhaltenen Resultate dem medizinischen Chemiker wichtige Erkenntnisse für weitere Optimierungen zu liefern.

Publication list

- Kohlbacher SM, Ionasz V-S, Ielo L, Pace V (**2019**) *The synthetic versatility of the Tiffeneau–Demjanov chemistry in homologation tactics*. Monatshefte Für Chem - Chem Mon 150:2011–2019. https://doi.org/10.1007/s00706-019-02514-3
- Montanari F, Knasmüller B, Kohlbacher S, et al (**2020**) *Vienna LiverTox Workspace—A Set of Machine Learning Models for Prediction of Interactions Profiles of Small Molecules With Transporters Relevant for Regulatory Agencies.* Front Chem 7:899. https://doi.org/10.3389/fchem.2019.00899
- Wieder O, Kohlbacher S, Kuenemann M, et al (**2020**) *A compact review of molecular property prediction with graph neural networks.* Drug Discov Today Technol 37:1–12. https://doi.org/10.1016/j.ddtec.2020.11.009
- Kohlbacher SM, Langer T, Seidel T (**2021**) *QPHAR: quantitative pharmacophore activity relationship: method and validation.* J Cheminformatics 13:57. https://doi.org/10.1186/s13321-021-00537-9
- Kohlbacher SM, Schmid M, Seidel T, Langer T (**2022**) *Applications of the Novel Quantitative Pharmacophore Activity Relationship Method QPhAR in Virtual Screening and Lead-Optimisation.* Pharmaceuticals 15:1122. https://doi.org/10.3390/ph15091122
- Seidel T, Permann C, Wieder O, et al (**2022**) *High-Quality Conformer Generation with CDPKit/CONFORT: Algorithm and Performance Assessment*. Under Review
- Kohlbacher SM, Ibis G, Permann C, Bryant S, Langer T, Seidel T (**2022**) *A new set of KNIME nodes implementing the QPhAR algorithm.* Molecular Informatics. Under Review