THE UNIVERSITY OF OXFORD | THE DEPARTMENT OF STATISTICS

FIXED AND LEARNED REPRESENTATIONS IN EARLY STAGE DRUG DISCOVERY

CANDIDATE NUMBER: 1047400

In Fulfillment of Assessment for
'Topics in Computational Biology'

May 2, 2021

ABSTRACT

Der Abstract fasst die zentralen Inhalte der Arbeit zusammen. Eine Wertung oder Interpretation erfolgt nicht. Dies hilft, sich einen groben Überblick über Fragestellung, Vorgehen und Ergebnisse zu verschaffen. Bestandteil sollen die Teile a) Hintergrundinformationen, Fragestellung, Zielsetzung, Forschungskontext, b) Methoden, c) Ergebnisse und d) Schlussfolgerungen, Anwendungsmöglichkeiten sein. Der Text ist knapp, vollständig und präzise, zudem objektiv und ohne persönliche Wertung. Achten Sie auf eine einfache und verständliche Sprache. Alle genannten Inhalte müssen auch im Hauptteil aufgegriffen werden. Den Inhalt objektiv und ohne persönliche Wertung wiedergeben. Gehen Sie auf die wichtigsten Konzepte, Resultate oder Folgerungen ein. Verwenden Sie keine Zitate und verzichten Sie auf Abkürzungen. In der Regel sind ca. 200 Wörter ausreichend.

CONTENTS

1 Introduction . . . 1
2 Fixed Representations . . . 2
2.1 Descriptors . . . 2
2.2 Fingerprint Vectors . . . 3
3 Learned Representations . . . 6
3.1 Technical Background . . . 6
3.1.1 Molecular Graphs . . . 6
3.1.2 Message Passing Neural Networks . . . 7
3.2 Results . . . 8
3.2.1 Convolutional Neural Networks for Learning Molecular Fingerprints . . . 9
3.3 Directed MPP ? Yang et al. (2019a) . . . 9
3.3.1 Graph Convolutional Neural Networks . . . 9
3.3.2 Graph Attention Networks . . . 9
3.3.3 Attentive FP . . . 9
3.4 Sequence modeling . . . 10
4 Case study . . . 10
4.1 De novo design . . . 10
5 Discussion . . . 10

6 Conclusion 11
References 16
List of Figures 17
List of Tables 18
Appendix 18
A Similarity values for fingerprints 18
B Kostenrechnung 19
C Ergebnisse 19

1 INTRODUCTION

From 2010 to 2020 the amount of data that was processed rose from 1.2 trillion gigabytes to 59 trillion gigabytes - an increase by 5,000% (dat, 2021). This exponential growth has evoked a high demand to leverage these amounts of data to promote scientific discoveries. In particular, it motivated the use of machine learning across all disciplines. Machine learning (ML) refers to a field of study that gives computers the ability to learn without being explicitly programmed. The benefits of this approach are immediate, since it allows computational systems to automatically process and reason about the enormous amounts of data, exceeding task-specific human capabilities considerably. Most recently, deep learning (DL) has emerged as a sub-discipline of machine learning denoting the use of multiple hidden layers in a network. Deep learning models can achieve even better accuracy than standard machine learning architectures given the availability of a substantially greater amount of data.

A great potential of these developments in drug discovery lies within early stages to predict structure-activity relationships (SARs) and structure-property relationships (SPRs). These are grounded on the fundamental assumption that structurally similar molecules have similar activities/properties. For instance, after finding a hit compound in a drug screening campaign researchers would like to understand how its chemical structure can be optimised in order to improve properties like binding affinity, biological responses or physiochemical properties (Lo et al., 2018). Classically, this problem could only be solved through resource-intensive *in vitro* screeening and *in vivo* testing. Early quantitative structure-activity/property relationships (QSAR) models (Hansch & Fujita, 1964), that attempted to solve this problem *in silico*, were limited by a lack of experimental data and the linearity assumption made for modeling (Lo et al., 2018). The introduction of high-throughput screening (HTS) and combinatorial synthesis resulted in a rapid explosion of the availability of data screening 100,000s or more samples per day for a desired biological activity. Ultimately, this led to the development of large databases containing the profiles of millions of chemical substances. How to effectively use these amounts of data for machine and deep learning methods has become a crucial challenge for drug discovery.

In abstract terms, fitting a QSAR/QSPR model amounts to finding a generally non-linear function between a class of molecules and a desired biological activity/property. ML/DL methods solve this problem by learning this function automatically. Pretty much any machine learning methods has been applied in this second step Shen & Nicolaou (2019) and popular examples include support vector machines Heikamp & Bajorath (2013); Zernov et al. (2003), extreme gradient boosting Jiang et al. (2020a); Yang et al. (2019b) and random forest Svetnik et al. (2003). However, the performance of these methods crucially depends on the used mathematical representation of molecules, since it needs to describe the attributes of the molecules necessary to predict the activity/property of concern. Two approaches to design these 'features' have been introduced. On the one hand, fixed representations, like descriptors and fingerprints, map a molecule to a predefined vector space that contains the numerical values of some selected properties of a molecule. The drawback of this approach is that expert knowledge is necessary in order to make a meaningful selection of properties that are useful for learning the relationship to the target activity/property. Furthermore, this selection is inherently biased by the domain knowledge (Merkwirth & Lengauer, 2005). The other, recently emerged, class of representations is given by learned representations. Instead of manually employing a mapping to a fixed space, Recurrent Neural Networks (RNNs) and Graph Neural Networks (GNNs) can be used to learn the space itself such that is contains the most suitable properties for predicting the target values. The typical workflow of designing features and predicting an activity/property is summarised in figure 1.

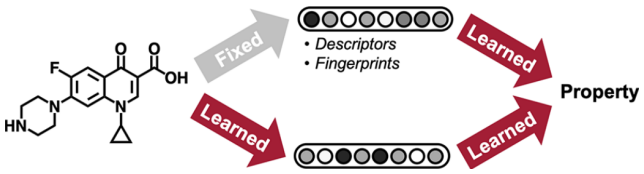

Figure 1: Illustration of the QSAR/SQPR workflow using ML/DL. Reprinted from Yang et al. (2019a).

There are numerous examples highlighting the potential of learned representations (Duvenaud et al., 2015; Li et al., 2019b; Honda et al., 2019). Most prominently, Stokes et al. (2020) achieved a breakthrough in antibiotic discovery when they discovered a new broad-spectrum bactericidal antibiotic 'halicin' after decades of stagnation in the field. They employed a directed-message passing neural network Yang et al. (2019a) at two stages of their work. Firstly, this graph neural network was used to predict growth inhibitory effects against E. Coli. Later, another D-MPNN was used to predict the toxicity of potential candidates. This finding underlines both the potential of learned representations to lead to meaningful discoveries as well as their versatility to be employed at various stages of an early-stage drug discovery workflow.

The goal of this thesis is to present both fixed and learned molecular representations to understand the advantages and drawbacks of either approach. In the following two sections these are introduced, accompanied by technical background information where needed. Consecutively, we exemplify the application of a learned representation using the paper of Stokes et al. (2020). Finally, we discuss both approaches in terms of interpretability, computational costs and accuracy.

The advancements of these methods are further motivated by the enormous costs time and costs connected to drug development. Discovery and development of a new drug can take 5000-10000 compounds to screen and 12-15 years to end up with one approved drug requiring costs of more than $1.3B. Only 2 out of 10 approved and marketed drugs can recover these costs Hecht & Fogel (2009).'

2 Fixed Representations

Mapping molecules to a fixed representation is the classical form of generating a machine-interpretable input for QSAR/QSPR models. As the name suggests these are characterised by a fixed target space that contains a pre-selected choice of local or global information about the molecule. Fixed representations can be broadly separated into two categories: Molecular descriptors and fingerprints. Descriptors are generally characterised by a more holistic representation of the molecule. Fingerprints, on the other hand, are local in nature by aggregating information of subgroups of atoms in a molecule.

2.1 Descriptors

According to Todeschini & Consonni (2008) 'The molecular descriptor is the final result of a logical and mathematical procedure which transforms chemical information encoded within a symbolic representation of a molecule into an useful number or the result of some standardized experiment'. This definition highlights the purpose of a descriptor to generate a numerical representation, such as a vector of numbers, from a symbolic representation.

The expectations for the usefulness of a descriptor vary a lot depending on the application domain but according to Mauri et al. (2016) these typically include

1. invariance to node reorderings,
2. invariance to rotations and translations of the molecule,
3. definition by an umabiguous algorithm,
4. well-defined applicability to molecular structures.

These desiderata are supposed to guarantee that the descriptor always gives the same representations for molecules that are considered the same and is generally applicable to all molecules. Beyond that, common extra requirements concern the inclusion of structural information (according to the fundamental principle of chemistry that different structures possess differeny properties), certain discriminative abilities and degeneracy/continuity, i.e. small structural differences result in small but existing differences in the value of the descriptor.

The variety of different descriptors that have been used for QSAR analysis is enormous and depends highly on the considered application. Any attempt to group descriptors into different categories would be quite arbitrary given the sheer amount of different application domains and descriptors. However, Guha & Willighagen (2013) propose an grouping based on the nature of the structural information that they require: Constitutional, topological, geometric and quantum mechanical descriptors.

.

Constitutional descriptors are the most rudimentary form of descriptors as they do not take into account any spatial information about the molecule but just its basic structural properties. Examples include basic attributes like the molecular weight the number of atoms but also more complex ones such as the sum of atomic van der Waals volumes.

Topological descriptors are based on the connectivity of the atoms in a molecule and encode 2D structural properties using graph invariants of the underlying molecular graphs, i.e. properties that only depend on the abstract mathematical object and not on a particular labeling or ordering of the vertices. Such invariants include the Wiener index Wiener (1947); Nikolić et al. (2001) $W = \frac{1}{2}\sum_{i,j}^{N} d_{ij}$, where N is the number of non-hydrogen atoms and d_{ij} is the edge count of the shortest part between atoms i and j. A drawback of topological descriptors compared with constitutional descriptors is that they often tend to be less interpretable due to the abstract nature of the underlying graph.

Geometric descriptors receive 3D information about the molecule as their input which may be resourceful to obtain from crystallographic data or molecular optimization Mauri et al. (2016). However, they may also come with more information compared to descriptors that receive lower dimensional inputs. Therefore, they are usually employed in domains when this additional information is critical such as when two conformations are compared (TODO rewrite). An example of a geometric descriptor is given by the 3D Wiener Index which extends the 2D case by weighing the edges by their actual length or the gravitation index Katritzky et al. (1996).

Finally, quantum mechanical descriptors are based on quantum mechanical calculations. For instance, they have been used to predict toxicity of molecules in QSAR studies (Reenu & Vikas, 2015; Senior et al., 2011). However, their tendency to require high computational costs impediment their application ot large scale virtual screenings.

Note that these categories are a non-exhaustive classification of descriptors and many others exist such as auto-correlation descriptors (Broto et al., 1984) (TODO one more?). We conclude that descriptors are a popular method to represent molecules as they are a flexible means to encode the properties that are relevant to the particular application domain. However, this comes also with a downside as the performance of the application may heavily depend on the choice of descriptors and this selection is by no means a trivial task.

2.2 Fingerprint Vectors

All descriptors considered so far are derived from performing mathematical computations on the underlying structure and give a holistic representation of the substances considered. Fingerprint vectors on the other hand are given as bit vectors that indicate the presence or absence of a local property and are thus local in nature. Two classes of fingerprints can be distinguished (Shen & Nicolaou, 2019): Dictionary-based and hash-based fingerprints. Dictionary-based fingerprints such as Molecular ACCess System (MACCS) are computed by encoding each position of the vector as the presence or absence of structural property from a pre-defined dictionary. However, these can be very sparse if arbitrarily large vectors are used leading to an inefficient representation. To overcome this sparsity hash-based fingerprints have been introduced that employ a hashing algorithm to combine the different substructures into a unique bit-vector. These substructures can be enumerated linearly by iterating over all edges in a molecular graph (day, 2021) or in a circular manner as for extended connectivity fingerprints (ECFPs).

Since most of the recent studies that explore fixed and learned representations for drug discovery use ECFPs for baseline results (Li et al., 2017; Stokes et al., 2020; Wu et al., 2018), we choose to demonstrate their technical details in the following.

Extended Connectivity Fingerprints (ECFPs) were first introduced by the software Pipeline Pilot in 2000 and then described in detail by Rogers & Hahn (2010). The origin of this representation goes back to Morgan (1965) who introduced the Morgan algorithm on which ECFPs are based. This is why they are also often referred to as Morgan fingerprints. This algorithm assigns numerical values to each atom by an iterative process that does not depend on a specific numbering of the atoms. It is depicted in Algorithm 1.

ECFPs adapt this algorithm by stopping the while-loop after a predefined number of steps rather than until completion and storing the intermediate values. We outline each part of the full algorithm in detail in the following paragraphs.

Algorithm 1: Morgan Algorithm TODO check with paper

Data: Molecular graph
Result: unique node ordering
Assign each atom the value 1;
while *not done* **do**
  for *atom in atoms* **do**
    Update value by the sum of the values from the neighbouring atoms;
  end
  if *number of different values does not change* **then**
    break;
  end
end

In the first step every non-hydrogen atom is assigned an integer identifier that can be chosen arbitrarily as long as it is independent of the node ordering, e.g. the atom's mass or atomic number. Rogers & Hahn (2010) choose a 32 bit integer value as an identifier that results from hashing the properties used in the Daylight atomic invariants rule (Weininger et al., 1989). A set A is created containing the initial identifiers of all the atoms. Then, for each atom we add the atom's own identifier and that of its immediate neighbouring atoms together with their bond order to an array (ordered by the atoms' identifiers and the order of the attaching bonds). These values are then hashed to get a single-integer identifier which overrides the initial identifier that the atom was assigned. The updated identifiers are added to the set A if there are no two structurally equal identifiers in the set. Two identifiers are considered structurally equal if after an equal number of iterations they encode the same substructure of the molecule. This may occur for example for the nitrogen and oxygen atoms at the top and right of the structure shown in Figure 2. After two iterations they both encode the same substructure consisting of the two carbon atoms, the oxygen atom and the nitrogen atom. To avoid this information redundancy only one of the corresponding hashes is added.

The first step is repeated n times using the updated identifiers of each atom as the the initial identifiers for the next step. After the completion of the n steps, numerically equal values are removed from the set A to arrive at the final ECFP. (fixed length?)

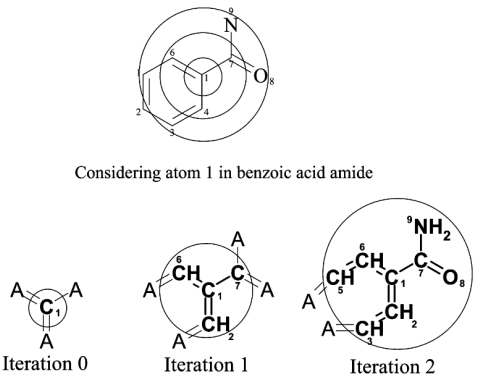

Figure 2: Illustration of the iterative updating in the computation of the ECFPs. In this example the atom type is used as an identifier. In iteration 0 the middle atom' identifier only represents the information about its own type. After the first iteration it has aggregated the information from its immediate neighbors and after the second iteration the represented substructure has grown even further. Reprinted from Rogers & Hahn (2010).

We clearly see ECFP's local nature. It manages to generate a global representation by using only local operations thereby implicitly encoding the molecule's structure. This is opposed to molecular descriptors discussed in the previous subsection that are based on global properties.

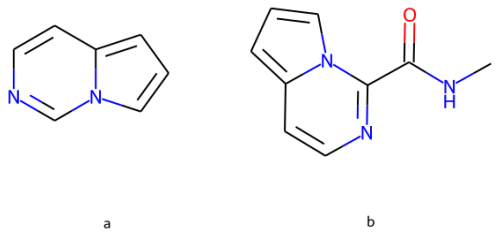

Figure 3: Molecular graphs corresponding to the SMILES strings 'c1nccc2n1ccc2' and 1CNC(=O)c1nccc2cccn12'.

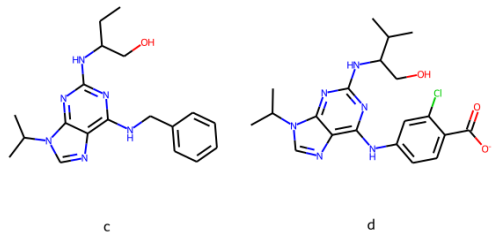

Figure 4: Molecular graphs corresponding to the SMILES strings 'CCC(CO)Nc1nc(NCc2ccccc2)c2ncn(C(C)C)c2n1' and 'CC(C)C(CO)Nc1nc(Nc2ccc(C(=O)[O-])c(Cl)c2)c2ncn(C(C)C)c2n1"

In the literature ECFP fingerprints are usually used with $n = 2$ which is referred to as ECFP4 (4 being the maximum diameter of substructures considered). To understand the importance of this parameters, we compare the predicted structural similarity of two pairs of molecules in Figure 3 and 4 for $n = 1, 2, 3$. Furthermore, we list the similarity scores of Atom-Pair fingerprints (Carhart et al., 1985) that are considered more suitable for representing larger molecules since they aggregate information of all pairs of atoms seperated by an arbitrary distance. The results are described in Table 1. The source code for this experiment can be found in the Appendix A. We choose a pair of smaller molecules and one of larger molecules to understand if their size has any impact on the similarity scores. As expected, the predicted similarity drops for either pair with an increasing n since larger and more dissimilar substructures are taken into account. However, the drop is substantially more significant for molecules a & b. This may be because for the first pair the proportion of dissimilar parts is larger for greater n relative to the second pair. We remark that this hyperparameter appears to play an important role for when ECFPs are used as the input features for machine learning techniques. We can interpret n as a regularisation parameter that penalises structurally too dissimilar molecules to be assigned too similar properties by a machine learning algorithm.

	Morgan Fingerprints			
Molecules	**ECFP2**	**ECFP4**	**ECFP6**	**APFP**
a & b	56.25%	46.15%	34.29%	50.88%
c & d	68.66%	58.71%	52.86%	54.47%

Table 1: Sørensen-Dice similarity values Sorensen (1948); Dice (1945) using different fingerprints for molecules in Figure 3 and Figure 4 respectively

Other circular fingerprints can be obtained by selecting different identifiers such as FCFPs (Functional Class Fingerprints) that is based on the pharmacophore role of the atoms in a molecule (Rogers & Hahn (2010)), SCFPs (Clark et al., 1989) or LCFPs (Ghose et al., 1998). The choice of the identifier is ultimately responsible for the discriminative abilities of the fingerprint. So expert knowledge is needed to make a meaningful decision.

As with numerical descriptors fingerprints are also a powerful mean to represent molecules in form of a fixed-size array. But a similar drawback as to numerical descriptors is that the best fingerprints depend strongly on the considered data set which again is non-trivial to find.

3 Learned Representations

One of the major drawbacks of using a fixed representation as inputs for ML methods in drug discovery is that the performance of the respective method is dependent on an a priori selection of features and therefore biased by expert knowledge Merkwirth & Lengauer (2005). For descriptors this choice is given by the selection of properties to be represented by the descriptor. Molecular fingerprints require this selection in the form of the identifier that is used to initialise the atom's values. This manual feature design means that a significant inductive bias is imposed and the resulting method can only perform as well as the feature selection allows. An idea to remedy this problem is given by stepping away from a fixed target feature space and instead use deep learning to learn the space itself. Specifically, two kinds of model have been proposed. Graph Neural Networks operate directly on a molecular graph and extract structural information combined with the features most relevant to a property of interest.

In this section we first give a technical description of the basic Graph Neural Network framework. This is accompanied by high-level explanations. Furthermore, we list the most recent extensions to this framework in the literature and briefly mention their application.

There are several anticipated benefits of using a learned representation (Shen & Nicolaou, 2019):

1. It does not result in large, sparse representations as fingerprints.
2. It provides a level of interpretability through (Duvenaud et al., 2015) (read)
3. Attention algorithms can be adopted to f (Li et al., 2019a; Xiong et al., 2020)
4. It could improce predictive performance on large data sets. (Yang et al., 2019a)

3.1 Technical Background

3.1.1 Molecular Graphs

Molecular graphs are a convenient means to represent molecules in two dimensions. Formally a graph is defined as a tuple of sets $G = (V, E)$, where V are the vertices of the graph and E are the edges. Any edge $e \in E$ is uniquely identified by a pair of vertices (v_1, v_2), $v_1, v_2 \in V$ that it connects. In a molecular graph the vertices are given by the atoms and edges represent bonds between atoms. An example of a molecular graph is given in Figure 5. We also note that the number of edges, i.e. the edge *multiplicity*, may differ. This corresponds to the bond order in the molecule, i.e. the difference between the number of bonds and anti-bonds between two atoms, as introduced by Pauling (1947).

In computers, graphs are represented by a matrix - most commonly by their adjacency matrix A. The entries of this matrix are given by

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge from } v_i \text{ to } v_j \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Note that for an undirected graph, like a molecular graph, the adjacency matrix is always symmetric. In order to represent a graph by its adjacency matrix, we need to make a non-canonical choice of ordering the nodes. This is inconvenient for molecular graphs since these do not possess any kind of ordering and hence this representation is not well-defined.

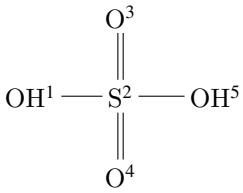

Figure 5: Molecular graph of sulfuric acid.

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \end{array}$$

Figure 6: Adjacency matrix of the molecular graph representing sulfuric acid given the node ordering.

Figure 6 shows the adjacency matrix corresponding to the graph in Figure 5. The ordering of the vertices is indicated by superscripts. If we assumed a different ordering of the vertices this would results in a permutation of the rows and columns of the adjacency matrix. As we will see, this is a common problem for Graph Neural Network which is attempted to be solve by the introduction of an *inductive bias* devising algorithms that give the same results regardless of a permutation of the matrix.

In order to represent information about molecules beyond the connection of its atom, the adjacency matrix is complemented with two more matrices - a node feature matrix and an edge feature matrix. These contain additional information about each atom and bond in a molecular graph. The node feature matrix has the same number of rows as the adjacency matrix, where row i corresponds to the feature values for node i. The number of columns may vary depending on the number of features that are chosen to be encoded. An example feature matrix is shown is Figure 7. Finally, the edge feature matrix contains one row for every edge in the graph, where row i corresponds to edge i (TODO edge ordering?) and again the number of columns may vary depending on the number of features, see Figure 8.

$$\begin{array}{c} \begin{matrix} O & S & 0H & 1H \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \end{array}$$

Figure 7: Example feature matrix of the graph in Figrue 5. The first two columns encode the atom type and the last two columns are a one-hot encoding of the number of implicit hydrogen atoms.

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{matrix} (1,2) \\ (2,3) \\ (2,4) \\ (2,5) \end{matrix} \end{array}$$

Figure 8: Example edge feature matrix of the graph in Figure 5. The choses features represent a one-hot encoding of the bond type.

While the graphical representation allows for the representation of complex 3D information of molecules, there are some drawbacks of working directly on the graph level. First, not all molecules can be represented as graphs (David et al., 2020) such as those that contain bonds that cannot be explained by valence bond theory. Second, graphs are not a suitable means of depicting molecules whose arrangement of molecules change over time as this would require a reordering of the adjacency matrix every time. Finally, graphs are neither very compact nor easy to process. The adjacency matrix alone has a memory requirement quadratic in the number of atoms in the molecule and depending on the amount of atomic and bond information that is to be encoded the feature matrices might get even bigger. As opposed to this, a linear representation as a single string allows for using substantially less memory while being simultaneously easier to store and process by algorithms. Therefore, graphs are usually used as the basis of more compact representations that we are going to depict in the following subsections.

3.1.2 Message Passing Neural Networks

Convolutional Neural Networks (cite) have achieved remarkable success at learning representations of grid-like structures such as images. The idea to generalise these frameworks to less regular structures like graphs motivated the introduction of many Graph Convolutional Neural Networks (GCNNs) as in (Li et al., 2015; Duvenaud et al., 2015; Kearnes et al., 2016; Schütt et al., 2017). An attempt to unify all these approaches in a general framework was made by Gilmer et al. (2017) introducing

Message Passing Neural Networks (MPNNs). In the following we will outline how MPNNs work and mention how they restore the previous approaches.

MPNNs combine edge and node properties of a graph together with an implicit encoding of the structure. This is achieved through a similar aggregation step as for circular fingerprints in which a node update its own feature vector by combining it with the aggregated information from its neighbours. The difference is that a weighting of the features can be learned. As an input they require a graph represented by its adjacency matrix and the node and edge feature matrices that encode the properties. They output a feature vector for the full graph.

An entire forward pass of an MPNN can be divided into two phases: The message passing phase that runs for T time steps and a consecutive readout phase. Each node stores information about its own features and those of its local environment in a hidden state vector $\boldsymbol{h}_v^t \in \mathbb{R}^L$. $\boldsymbol{h}_v^0$ is initialised with the node's feature vector $\boldsymbol{x}_v$. For each time step during the first phase any node receives 'messages' about its neighbours' hidden states and then updates its own hidden state based on that. Specifically, this can be described as the two equations

$$\boldsymbol{m}_v^{t+1} = \sum_{w \in N(v)} M_t(\boldsymbol{h}_v^t, \boldsymbol{h}_w^t, \boldsymbol{e}_{vw}) \tag{2}$$

$$\boldsymbol{h}_v^{t+1} = U_t(\boldsymbol{h}_v^t, \boldsymbol{m}_v^{t+1}) \tag{3}$$

where $\boldsymbol{m}_v^t$ is the 'message' node v receives at time t which is composed of the sum of the message functions M_t from its immediate neighbours that can depend on their own hidden state $\boldsymbol{h}_w^t$, the neighbour's hidden state $\boldsymbol{h}_w^t$ and features of the edge connecting them.

After T time steps, any node v has now received information about any node w that are at most T edges away. This is because after the first step w's neighbors receive information about w's hidden state which is in turn incorporated in their own hidden state. In the next iteration, w's neighbours pass their hidden state, incorporating information about w's hidden state, to their own neighbours. This way, information about w's hidden state is propagated through the graph and after T iterations, v receives this information. This idea is illustrated in Figure 9.

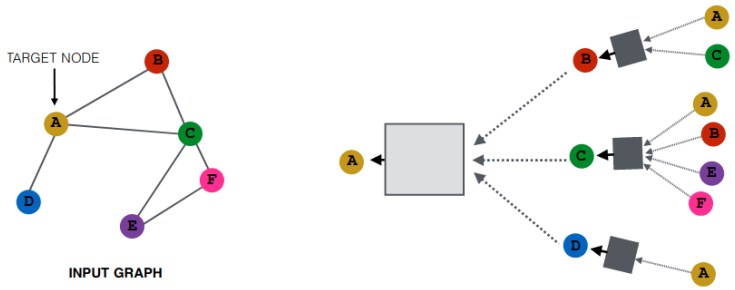

Figure 9: Illustration of the message passing in a MPNN. Reprinted from Hamilton et al. (2018).

The consecutive readout phase now computes a feature vector for the whole graph as given in equation 4

$$\hat{\boldsymbol{y}} = R(\boldsymbol{h}_1^T, \ldots, \boldsymbol{h}_{|V|}^T) \tag{4}$$

Different choices for the functions M_t, U_t and R restore different Graph Neural Networks proposed in the literature. All of them have in common that they are differentiable and learned through backpropagation. Furthermore, R must be permutation-invariant in order for the MPNN to be insensitive to the node ordering.

3.2 Results

We illustrate how MPNNs recover two former architectures of Graph Neural Networks. For more detail, we refer to Gilmer et al. (2017).

3.2.1 CONVOLUTIONAL NEURAL NETWORKS FOR LEARNING MOLECULAR FINGERPRINTS

The method presented by Duvenaud et al. (2015) was one of the first to challenge the state-of-the art approach of using circular fingerprints for the prediction of molecular properties. They noticed that the mechanism used for circular fingerprints, i.e. applying the same operation locally everywhere, was analogous to that of convolutional neural networks. This motivated the idea of creating a differentiable fingerprint that could be learned through backpropagation. To implement this, they went ahead to replace every non-differentiable operation of circular fingerprints by a differentiable analog.

Duvenaud et al. (2015). Here, the message function M_t is the same across all time steps and given by

$$M(\boldsymbol{h}_v, \boldsymbol{h}_w, \boldsymbol{e}_{vw}) = (\boldsymbol{h}_w, \boldsymbol{e}_{vw}),$$

where $(\cdot, \cdot)$ denotes concatentation. The update and readout functions are given by

$$U_t(\boldsymbol{h}_v^t, \boldsymbol{m}_v^{t+1}) = \sigma(\boldsymbol{H}_t^{\deg(v)} \boldsymbol{m}_v^{t+1})$$

which includes learnable parameters as given by the matrices $\boldsymbol{H}_t^k$ for all time steps t and node degrees k. σ denotes the sigmoid activation function. Finally, the readout function is given by

$$R(\boldsymbol{h}_1^T, \ldots, \boldsymbol{h}_{|V|}^T) = f\left(\sum_{v,t} \text{softmax}(\boldsymbol{W}_t \boldsymbol{h}_v^t)\right)$$

with learnable matrices $\boldsymbol{W}_t$ for all time steps t and a neural network f.

3.3 DIRECTED MPP ? YANG ET AL. (2019A)

3.3.1 GRAPH CONVOLUTIONAL NEURAL NETWORKS

These belong to the most popular classes of Graph Neural Networks and was proposed by (Kipf & Welling, 2016). A detailed derivation of the message and update functions can be found on Gilmer et al. (2017). The resulting functions are given by:

$$M_t(\boldsymbol{h}_v^t, \boldsymbol{h}_w^t) = \sum_{w \in N(v) \cup \{v\}} (\deg(v) \deg(w))^{-1/2} \boldsymbol{h}_w^t$$

The update function at time t is given by:

$$U_t(\boldsymbol{h}_v^t, \boldsymbol{m}_v^{t+1}) = \sigma(\boldsymbol{W}^t \boldsymbol{m}^{t+1})$$

with a trainable matrix $\boldsymbol{W}^t$ and a non-linear activation function σ, e.g. ReLU. This can be thought of as a generalisation of the MPNN framework that includes self-loops in the message passing step.

3.3.2 GRAPH ATTENTION NETWORKS

Graph Attention Networks proposed by Veličković et al. (2018) extend GCNs by replacing the normalisation constants $\deg(v) \deg(w)$ in the aggregation step by a learned attention score

$$M_t(\boldsymbol{h}_v^t, \boldsymbol{h}_w^t) = \sum_{w \in N(v)} \alpha_{vw}^t \boldsymbol{h}_w^t.$$

The attention score weighes the information from neighbours according to how important they are. Details about how α_{vw}^t is computed can be retrieved from the paper (Veličković et al., 2018).

3.3.3 ATTENTIVE FP

The most recent state of the art Graph Neural Network architecture was propsed by Xiong et al. (2019). It proceeds similarly to GANs by stacking multiple attention layers together with Gated Recurrent Units (GRUs) to update the nodes' hidden states recursively and allow an atom to focus on its most important neighbors. To generate the final graph embedding, Attentive FP treats the whole molecule as a virtual node that connects to all its atoms. Then, it employs the same architecture as for the individual atoms to learn the molecule's final representation. These GCN, GAN, Attentive FP?

3.4 Sequence modeling

SMILES RNNs, LSTMs? Honda et al. (2019)

4 Case study

As an example for a full MPP workflow, we choose Stokes et al. (2020) who used a Graph Neural Network to predict the growth of E. Coli. Their approach can be divided into three stages. The fist stage concerns the training of the model and a classifier according to Figure 1. The molecular representation was built using a directed-message passing neural network Yang et al. (2019a) and can therefore be classified as a learned representation. Similarly to ECFP fingerprints, (D-)MPNNs can struggle to represent global features of molecules, especially if the number of message passing iterations is greater than the longest path in the molecule as discussed in section 3.1.2. Therefore, the final representation generated by the D-MPNN was augmented with 300 additional molecule-level features. This combined representation was then input in a feed-forward neural network that outputs a number between 0 and 1 as the prediction of the molecule showing growth inhibitory against E. Coli. This whole architecture is trained in an end-to-end fashion such that the D-MPNN can generate a representation that is highly attuned to the desired property. The training of this architecture was performed using a set of 2335 molecules that had been classified as hit or non-hit using 80 % growth inhibition against E. coli BW251113 Zampieri et al. (2017) as a hit cut-off. On the test data this model achieved an AUC-ROC score of 0.896.

In the secong stage, 20 folds of the trained model using different weight initialisations were applied to 6,111 molecules from the Drug Repurposing Hub (Corsello et al., 2017) to predict their probability of growth inhibition against E. Coli. The 20 different results were averaged to arrive at the final prediction scores.

Finally, the best scoring 99 molecules were empirically tested for growth inhibition out of which 51 displayed this property. The resulting 51 molecules were ranked according to their clinical phase of investigation, structural similarity to the training data set and their toxicity that was also predicted using a D-MPNN. This resulted in the discovery ot the broad-spectrum bactericidal antibiotic halicin with a very low structural similarity to its nearest neighbour antibiotic in the training data emphasising the model's capacity to generalise.

This case study shows the versatility and potential of using Graph Neural Network for property prediction in early drug discovery. They could be employed for both prediction of growth inhibitory effects as well as toxicity and resulted in the finding of a new antibiotic after years of stagnation in this field. Stokes et al. (2020) also reported the prediction scores using Morgan fingerprints and various classifier and the rank of the newly discovered antibiotic halicin was lower in all of them ranging between 773-2644 compared to 69 for the D-MPNN approach. Therefore, it could be argued that halcin would not have been found if molecular fingerprints had been used. However, between there is still some correlation among the top scoring molecules. For instance, both the D-MPNN and Morgan fingerprints predict the same highest ranking molecule and the fourth place for D-MPNN is in second place for Morgan fingerprints. The question that remains to be answered is if this is just a correlation of numerical values and halicin being ranked much higher for learned representations is just a fortunate coincidence or if the predictions of GNNs actually carry more physical relevance.

Despite this breakthrough using the GNN approach, Stokes et al. (2020) still emphasise the importance of a combination of *in silico* and empirical investigations.

ADMET study, other properties?

4.1 De novo design

5 Discussion

Artificial intelligence and machine learning are currently one of the most rapidly evolving research areas and the progress in these fields has direct impacts on a great variety of disciplines. In particular, we have hinted at ther potential to revolutionise the entire field of drug discovery coming with significant reductions in time and resources (TODO where? maybe time span to see how little time).

Most recently, a variety of Graph Neural Networks has been introduced as a way to automatize the feature selection for molecular property prediction. Instead of relying on expert knowledge to select the most relevant attributes to be used for a computer-interpretable interpretation, which has been shown to heavily impact the performance of the property prediction (Tian et al., 2012), Graph Neural Network manage to learn a continuous vector representation that is highly attuned to the property of concern.

While many studies report that learned representations are superior to fixed representations in term of the property prediction accuracy for a variety of different applications (Wu et al., 2018; Yang et al., 2019a; Korolev et al., 2020), there is still no consensus on this and others report the dominance of descriptor-based approaches and fingerprints (Mayr et al., 2018; Jiang et al., 2020b). This suggests that there are other relevant factors that influence which approach is better. Since there a substantially more parameters involved in learning a representation compared with using a fixed representation a sufficiently large data set is critical to learned approaches. Something else to take into account is the mode of evaluation. As mentioned by Shen & Nicolaou (2019), the evaluation of model performance is critical to molecular property prediction. This is because unlike images there is no standard to generating ground truth labels for the data. These are usually obtained from experiments and experimental procedures can differ and are subject to human errors. Furthermore, baseline models are often not tuned enough to reach peak performance.(Finally a fundamental assumption of employing and comparing machine different machine learning models is that training and test data are all independently identically distributed. It has been noted that for different molecules this requirement is very hard to verify let alone achieve.) (find source)

In terms of the required computational resources, fixed representations can be computed much quicker than learned approaches

The last aspect to take into consideration is interpretability. Graph Neural Networks like all deep learning algorithms work as a black box. There is no real way to assign any meaning to its final representation in terms of interpreatblity. For descriptor based models on the other hand the SHAP method (Lundberg & Lee, 2017) allows for a way to interpret the final prediction scores by computing the contribution of each input feature that had been selected. Therefore, it enables an understanding of which features turned out to be the most relevant for a particular property.

I personally think that the future of property prediction is within learned molecular representations. While their lack of interpretability is a considerable drawback, there are two major advantages. First, GNNs are able to achieve state-of-the-art performance and they have already successfully used to impel (word?) areas that were stagnating before their introduction (Stokes et al., 2020). While there are still publications reporting better results for descriptor-based approaches, GNN's great potential to be adjusted will probably keep improving their results (phrasing). For example, since the message passing approach may struggle to represent global properties of a graph, a global readout (cite) has been proposed helping overcome this. Secondly, GNNs enable their application to property prediction without having to rely on domain experts that need to select appropriate features. This allows for a wider application across disciplines making GNNs a versatile and promising tool for the future.

6 Conclusion

In this report we have studied fixed and learned molecular representations for molecular property prediction. Two classes of learned representations were introduced, namely descriptor-based approaches and molecular fingerprints. We compared atom-pair descriptors with the most popular Morgan fingerprints to understand hoe they capture similarities between different molecules. We found that .... After that, we introduced molecular graphs and Graph Neural Networks that operate directly on the graph level as an example for a learned representation. Recent advancements for GNNs were outlined with the general message passing framework and more recent improvements through D-MPNNs, Graph Attention Networks and Attentive FP. These highlight the capability of further improvements for GNNs and henceforth their potential in molecular property prediction. Finally, we compared learned representations with fixed representations in terms of accuracy, computational costs and interpretability. Despite fixed approaches being better in terms of the two latter aspects, we hypothesised Graph Neural Networks to be the future molecular prpoerty predictions. On the one hand this was because of their state-of-the-art performance that is probable to be improveed

further due to their flexibility to be extended (wphrasing) and on the other hand due to their wide applicability given that they do not require expert knowledge to be used.

REFERENCES

54 predictions about the state of data in 2021. https://chem.libretexts.org/@go/page/21702, 2021. Retrieved: April 30, 2021.

Daylight chemical information systems. https://www.daylight.com/dayhtml/doc/theory/theory.finger.html, 2021. Retrieved: May 1, 2021.

Mahendra Awale and Jean-Louis Reymond. Polypharmacology browser ppb2: Target prediction combining nearest neighbors with machine learning. *Journal of Chemical Information and Modeling*, 59(1):10–17, 2019. doi: 10.1021/acs.jcim.8b00524. URL https://doi.org/10.1021/acs.jcim.8b00524.

P Broto, G Moreau, and C Vandycke. Molecular structures: perception, autocorrelation descriptor and sar studies: system of atomic contributions for the calculation of the n-octanol/water partition coefficients. *European journal of medicinal chemistry*, 19(1):71–78, 1984.

Raymond E. Carhart, Dennis H. Smith, and R. Venkataraghavan. Atom pairs as molecular features in structure-activity studies: definition and applications. *Journal of Chemical Information and Computer Sciences*, 25(2):64–73, 1985. doi: 10.1021/ci00046a002. URL https://doi.org/10.1021/ci00046a002.

Matthew Clark, Richard D. Cramer III, and Nicole Van Opdenbosch. Validation of the general purpose tripos 5.2 force field. *Journal of Computational Chemistry*, 10(8):982–1012, 1989. doi: https://doi.org/10.1002/jcc.540100804. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.540100804.

Steven Corsello, Joshua Bittker, Zihan Liu, Joshua Gould, Patrick McCarren, Jodi Hirschman, Stephen Johnston, Anita Vrcic, Bang Wong, Mariya Khan, Jacob Asiedu, Rajiv Narayan, Christopher Mader, Aravind Subramanian, and Todd Golub. The drug repurposing hub: A next-generation drug library and information resource. *Nature Medicine*, 23:405–408, 04 2017. doi: 10.1038/nm.4306.

Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12, 09 2020. doi: 10.1186/s13321-020-00460-5.

Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3): 297–302, 1945.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints, 2015.

Arup K Ghose, Vellarkad N Viswanadhan, and John J Wendoloski. Prediction of hydrophobic (lipophilic) properties of small organic molecules using fragmental methods: an analysis of alogp and clogp methods. *The Journal of Physical Chemistry A*, 102(21):3762–3772, 1998.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017. URL http://arxiv.org/abs/1704.01212.

Robert C Glen, Andreas Bender, Catrin H Arnby, Lars Carlsson, Scott Boyer, and James Smith. Circular fingerprints: flexible molecular descriptors with applications from physical chemistry to adme. *IDrugs*, 9(3):199, 2006.

Rajarshi Guha and Egon Willighagen. A survey of quantitative descriptions of molecular structure. *Current Topics in Medicinal Chemistry*, 12:1946–1956, 01 2013. doi: 10.2174/1568026611212180002.

William L. Hamilton, Rex Ying, Jure Leskovec, and Rok Sosic. Representation learning on networks. http://snap.stanford.edu/proj/embeddings-www/, 2018. Retrieved: April 19, 2021.

Corwin Hansch and Toshio Fujita. p -??-?? analysis. a method for the correlation of biological activity and chemical structure. *Journal of the American Chemical Society*, 86, 04 1964. doi: 10.1021/ja01062a035.

David Hecht and Gary Fogel. Computational intelligence methods for admet prediction. *Frontiers in Drug Design and Discovery*, 4, 01 2009.

Kathrin Heikamp and Jürgen Bajorath. Support vector machines for drug discovery. *Expert opinion on drug discovery*, 9, 12 2013. doi: 10.1517/17460441.2014.866943.

Shion Honda, Shoi Shi, and Hiroki R. Ueda. SMILES transformer: Pre-trained molecular fingerprint for low data drug discovery. *CoRR*, abs/1911.04738, 2019. URL http://arxiv.org/abs/1911.04738.

Dejun Jiang, Tailong Lei, Zhe Wang, Chao Shen, Dong-Sheng Cao, and Tingjun Hou. Admet evaluation in drug discovery. 20. prediction of breast cancer resistance protein inhibition through machine learning. *Journal of Cheminformatics*, 12:16, 03 2020a. doi: 10.1186/s13321-020-00421-y.

Dejun Jiang, Zhenxing Wu, Chang-Yu Hsieh, Chen Guangyong, Ben Liao, Zhe Wang, Chao Shen, Dong-Sheng Cao, Jian Wu, and Tingjun Hou. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. 09 2020b. doi: 10.21203/rs.3.rs-79416/v1.

Alan R Katritzky, Lan Mu, Victor S Lobanov, and Mati Karelson. Correlation of boiling points with molecular structure. 1. a training set of 298 diverse organics and a test set of 9 simple inorganics. *The Journal of Physical Chemistry*, 100(24):10400–10407, 1996.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8): 595–608, Aug 2016. ISSN 1573-4951. doi: 10.1007/s10822-016-9938-8. URL http://dx.doi.org/10.1007/s10822-016-9938-8.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

Vadim Korolev, Artem Mitrofanov, Alexandru Korotcov, and Valery Tkachenko. Graph convolutional neural networks as "general-purpose" property predictors: The universality and limits of applicability. *Journal of Chemical Information and Modeling*, 60(1):22–28, 2020. doi: 10.1021/acs.jcim.9b00587. URL https://doi.org/10.1021/acs.jcim.9b00587. PMID: 31860296.

G. Landrum. Rdkit: Open-source cheminformatics. https://www.rdkit.org/docs/index.html, 2006.

Junying Li, Deng Cai, and Xiaofei He. Learning graph-level representation for drug discovery, 2017.

Xiuming Li, Xin Yan, Qiong Gu, Huihao Zhou, Di Wu, and Jun Xu. Deepchemstable: Chemical stability prediction with an attention-based graph convolution network. *Journal of Chemical Information and Modeling*, 59(3):1044–1049, 2019a. doi: 10.1021/acs.jcim.8b00672. URL https://doi.org/10.1021/acs.jcim.8b00672. PMID: 30764613.

Xiuming Li, Xin Yan, Qiong Gu, Huihao Zhou, Di Wu, and Jun Xu. Deepchemstable: Chemical stability prediction with an attention-based graph convolution network. *Journal of chemical information and modeling*, 59(3):1044–1049, 2019b.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

Yu-Chen Lo, Stefano E. Rensi, Wen Torng, and Russ B. Altman. Machine learning in chemoinformatics and drug discovery. *Drug Discovery Today*, 23(8):1538–1546, 2018. ISSN 1359-6446. doi: https://doi.org/10.1016/j.drudis.2018.05.010. URL https://www.sciencedirect.com/science/article/pii/S1359644617304695.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Andrea Mauri, Viviana Consonni, and Roberto Todeschini. *Molecular Descriptors*, pp. 1–29. Springer Netherlands, Dordrecht, 2016. ISBN 978-94-007-6169-8. doi: 10.1007/978-94-007-6169-8_51-1. URL https://doi.org/10.1007/978-94-007-6169-8_51-1.

Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Joerg Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical Science*, 9, 06 2018. doi: 10.1039/C8SC00148K.

Christian Merkwirth and Thomas Lengauer. Automatic generation of complementary descriptors with molecular graph networks. *Journal of Chemical Information and Modeling*, 45(5):1159–1168, 2005. doi: 10.1021/ci049613b. URL https://doi.org/10.1021/ci049613b. PMID: 16180893.

H. L. Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5(2):107–113, 1965. doi: 10.1021/c160017a018. URL https://doi.org/10.1021/c160017a018.

Sonja Nikolić, Nenad Trinajstić, and Milan Randić. Wiener index revisited. *Chemical Physics Letters*, 333(3-4):319–321, 2001.

Linus Pauling. Atomic radii and interatomic distances in metals. *Journal of the American Chemical Society*, 69(3):542–553, 1947. doi: 10.1021/ja01195a024. URL https://doi.org/10.1021/ja01195a024.

Reenu and Vikas. Exploring the role of quantum chemical descriptors in modeling acute toxicity of diverse chemicals to daphnia magna. *Journal of Molecular Graphics and Modelling*, 61: 89–101, 2015. ISSN 1093-3263. doi: https://doi.org/10.1016/j.jmgm.2015.06.009. URL https://www.sciencedirect.com/science/article/pii/S1093326315300176.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t. URL https://doi.org/10.1021/ci100050t. PMID: 20426451.

Kristof T. Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R. Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8(1), Jan 2017. ISSN 2041-1723. doi: 10.1038/ncomms13890. URL http://dx.doi.org/10.1038/ncomms13890.

Samir A Senior, Magdy D Madbouly, et al. Qstr of the toxicity of some organophosphorus compounds by using the quantum chemical and topological descriptors. *Chemosphere*, 85(1):7–12, 2011.

Jie Shen and Christos A. Nicolaou. Molecular property prediction: recent trends in the era of artificial intelligence. *Drug Discovery Today: Technologies*, 32-33:29–36, 2019. ISSN 1740-6749. doi: https://doi.org/10.1016/j.ddtec.2020.05.001. URL https://www.sciencedirect.com/science/article/pii/S1740674920300032. Artificial Intelligence.

Th A Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5:1–34, 1948.

Jonathan M. Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M. Donghia, Craig R. MacNair, Shawn French, Lindsey A. Carfrae, Zohar Bloom-Ackermann, Victoria M. Tran, Anush Chiappino-Pepe, Ahmed H. Badran, Ian W. Andrews, Emma J. Chory, George M. Church, Eric D. Brown, Tommi S. Jaakkola, Regina Barzilay, and James J. Collins. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.e13, 2020. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2020.01.021. URL https://www.sciencedirect.com/science/article/pii/S0092867420301021.

Vladimir Svetnik, Andy Liaw, Christopher Tong, John Culberson, Robert Sheridan, and Bradley Feuston. Random forest: A classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43:1947–58, 11 2003. doi: 10.1021/ci034160g.

Sheng Tian, Junmei Wang, Youyong Li, Xiaojie Xu, and Tingjun Hou. Drug-likeness analysis of traditional chinese medicines: Prediction of drug-likeness using machine learning approaches. *Molecular pharmaceutics*, 9:2875–86, 06 2012. doi: 10.1021/mp300198d.

Roberto Todeschini and Viviana Consonni. *Handbook of molecular descriptors*, volume 11. John Wiley & Sons, 2008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

David Weininger, Arthur Weininger, and Joseph L. Weininger. Smiles. 2. algorithm for generation of unique smiles notation. *Journal of Chemical Information and Computer Sciences*, 29(2):97–101, 1989. doi: 10.1021/ci00062a008. URL https://doi.org/10.1021/ci00062a008.

Harry Wiener. Structural determination of paraffin boiling points. *Journal of the American chemical society*, 69(1):17–20, 1947.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning, 2018.

Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Zhong Feisheng, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, H. Jiang, and Mingyue Zheng. Pushing the boundaries of molecular representation for drug discovery with graph attention mechanism. *Journal of Medicinal Chemistry*, 63, 08 2019. doi: 10.1021/acs.jmedchem.9b00959.

Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, 2020. doi: 10.1021/acs.jmedchem.9b00959. URL https://doi.org/10.1021/acs.jmedchem.9b00959. PMID: 31408336.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019a. doi: 10.1021/acs.jcim.9b00237. URL https://doi.org/10.1021/acs.jcim.9b00237. PMID: 31361484.

Zi-Yi Yang, Zhi-Jiang Yang, Jie Dong, Liang-Liang Wang, Liu-Xia Zhang, Jun-Jie Ding, Xiao-Qin Ding, Ai-Ping Lu, Ting-Jun Hou, and Dong-Sheng Cao. Structural analysis and identification of colloidal aggregators in drug discovery. *Journal of Chemical Information and Modeling*, 59 (9):3714–3726, 2019b. doi: 10.1021/acs.jcim.9b00541. URL https://doi.org/10.1021/acs.jcim.9b00541. PMID: 31430151.

Mattia Zampieri, Michael Zimmermann, Manfred Claassen, and Uwe Sauer. Nontargeted metabolomics reveals the multilevel response to antibiotic perturbations. *Cell Reports*, 19(6): 1214–1228, 2017. ISSN 2211-1247. doi: https://doi.org/10.1016/j.celrep.2017.04.002. URL https://www.sciencedirect.com/science/article/pii/S2211124717304618.

Vladimir V. Zernov, Konstantin V. Balakin, Andrey A. Ivaschenko, Nikolay P. Savchuk, and Igor V. Pletnev. Drug discovery using support vector machines. the case studies of drug-likeness, agrochemical-likeness, and enzyme inhibition predictions. *Journal of Chemical Information and Computer Sciences*, 43(6):2048–2056, 2003. doi: 10.1021/ci0340916. URL https://doi.org/10.1021/ci0340916. PMID: 14632457.

List of Figures

1 Illustration of the QSAR/SQPR workflow using ML/DL. Reprinted from Yang et al. (2019a). 1

2 Illustration of the iterative updating in the computation of the ECFPs. In this example the atom type is used as an identifier. In iteration 0 the middle atom' identifier only represents the information about its own type. After the first iteration it has aggregated the information from its immediate neighbors and after the second iteration the represented substructure has grown even further. Reprinted from Rogers & Hahn (2010). 4

3 Molecular graphs corresponding to the SMILES strings 'c1nccc2n1ccc2' and 1CNC(=O)c1nccc2cccn12'. 5

4 Molecular graphs corresponding to the SMILES strings 'CCC(CO)Nc1nc(NCc2ccccc2)c2ncn(C(C)C)c2n1' and 'CC(C)C(CO)Nc1nc(Nc2ccc(C(=O)[O-])c(Cl)c2)c2ncn(C(C)C)c2n1" 5

5 Molecular graph of sulfuric acid. 7

6 Adjacency matrix of the molecular graph representing sulfuric acid given the node ordering. 7

7 Example feature matrix of the graph in Figrue 5. The first two columns encode the atom type and the last two columns are a one-hot encoding of the number of implicit hydrogen atoms. 7

8 Example edge feature matrix of the graph in Figure 5. The choses features represent a one-hot encoding of the bond type. 7

9 Illustration of the message passing in a MPNN. Reprinted from Hamilton et al. (2018). 8

List of Tables

1 Sørensen-Dice similarity values Sorensen (1948); Dice (1945) using different fingerprints for molecules in Figure 3 and Figure 4 respectively 5

Appendix

A Similarity values for fingerprints

Used RDKit(Landrum, 2006) implementation. Note that this library implements Morgan fingerprints which use the same algorithms as the one proposed in (Rogers & Hahn, 2010) but with a different hashing function

[7]:
```
from rdkit import Chem
from rdkit.Chem import Draw
from rdkit.Chem.Draw import IPythonConsole
from rdkit.Chem.Draw import rdMolDraw2D
from rdkit.Chem import rdDepictor
from rdkit.Chem.AtomPairs import Pairs
from rdkit import DataStructs

# rdDepictor.SetPreferCoordGen(True)
from IPython.display import SVG
from rdkit.Chem import AllChem
```

[16]:
```
m1s = [Chem.MolFromSmiles("c1nccc2n1ccc2"), Chem.
↪MolFromSmiles("CNC(=O)c1nccc2cccn12")]
m2s = [Chem.
↪MolFromSmiles("CC(C)(O1)C[C@@H](O)[C@@]1(O2)[C@@H](C)[C@@H]3CC=C4[C@]3(C2)C(=O)C[C@H]5[C@H]
↪Chem.
↪MolFromSmiles("CC(C)C(CO)CC(C)C(CO)Nc1nc(Nc2ccc(C(=O)[O-])c(Cl)c2)c2ncn(C(C)C)c2n1")]
```

[17]:
```
img1 = Draw.MolsToGridImage(m1s,molsPerRow=2,subImgSize=(300,300),␣
↪returnPNG=False, legends = ['a', 'b'])
img1.save("test1.png")
img2 = Draw.MolsToGridImage(m2s,molsPerRow=2,subImgSize=(300,300),␣
↪returnPNG=False, legends = ['c', 'd'])
img2.save("test2.png")
```

[18]:
```
### atom pair fingerprints
AP_FP1s = [Pairs.GetAtomPairFingerprint(m) for m in m1s]
AP_FP2s = [Pairs.GetAtomPairFingerprint(m) for m in m2s]
hashdict0 = AP_FP1s[0].GetNonzeroElements()
print(sum( hashdict0.values()) == 36) #number of hash values equals␣
↪number of atom pairs in the first molecule= 9 choose 2
```

```
True
```

[19]:
```
print('Dice Similarity of Atom Pair Fingerprints of molecules m1 and␣
↪m2', DataStructs.DiceSimilarity(AP_FP1s[0],AP_FP1s[1]))
print('Dice Similarity of Atom Pair Fingerprints of molecules m1 and␣
↪m2', DataStructs.DiceSimilarity(AP_FP2s[0],AP_FP2s[1]))
```

```
Dice Similarity of Atom Pair Fingerprints of molecules m1 and m2
0.5087719298245614
Dice Similarity of Atom Pair Fingerprints of molecules m1 and m2
0.21837837837837837
```

[20]:
```
#Morgan fingerprints
for k in range(1,4):
    M_FP1s = [AllChem.GetMorganFingerprintAsBitVect(m,k,nBits=1024) for
↪m in m1s]
    M_FP2s = [AllChem.GetMorganFingerprintAsBitVect(m,k,nBits=1024) for
↪m in m2s]
    print('Dice Similarity of Morgan Fingerprints of a and b using r =
↪' + str(k), DataStructs.DiceSimilarity(M_FP1s[0],M_FP1s[1]))
    print('Dice Similarity of Morgan Fingerprints of c and d using r =
↪' + str(k), DataStructs.DiceSimilarity(M_FP2s[0],M_FP2s[1]))
```

```
Dice Similarity of Morgan Fingerprints of a and b using r = 1 0.5625
Dice Similarity of Morgan Fingerprints of c and d using r = 1
0.20689655172413793
Dice Similarity of Morgan Fingerprints of a and b using r = 2
0.46153846153846156
Dice Similarity of Morgan Fingerprints of c and d using r = 2 0.
↪174496644295302
Dice Similarity of Morgan Fingerprints of a and b using r = 3
0.34285714285714286
Dice Similarity of Morgan Fingerprints of c and d using r = 3 0.
↪1791044776119403
```

[ ]:

B KOSTENRECHNUNG

C ERGEBNISSE