THE UNIVERSITY OF OXFORD | THE DEPARTMENT OF STATISTICS

GRAPH NEURAL NETWORKS FOR LEARNING MOLECULAR REPRESENTATIONS IN DRUG DISCOVERY

CANDIDATE NUMBER: 1047400

In Fulfillment of Assessment for
'Topics in Computational Biology'

May 2, 2021

CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

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 analyse 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 if a substantially greater amount of data is available.

A field that has seen a particularly high interest in employing machine and deep learning technologies is drug discovery. Discovery and development of a new drug can take 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). These figures put a great emphasis on making this process less resource-intensive promoting the use of machine learning. One of the main applications of ML for drug discovery lies in early stages that are concerned with target and hit identification as well as lead optimisation. Here, ML is used in quantitative structure-activity relationships (QSAR) and quantitative structure-property relationships (QSPR) models to predict properties and activities of potential drug candidates. For instance, after finding a hit compound researchers would like to understand how its chemical structure can be optimised in order to improve properties like binding affinity, biological responses or physio-chemical properties (Lo et al., 2018).

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 methods solve this problem by learning this function from existing input/output pairs and almost any popular machine learning methods has been applied for QSAR analysis Shen & Nicolaou (2019). 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). Since ML methods cannot operate on molecular structures directly, a suitable mathematical representation of molecules is needed. Finding and selecting this representation is referred to as featurization. It is crucial for the performance of ML methods that these features contain all of the information that impact the relationship between molecules and target property. Otherwise, the method may not be able to discern their true relationship. Hence, great emphasis has been put on developing methods for featurization to extract all of the relevant information.

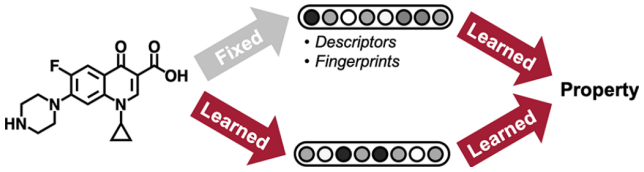

Figure 1: Illustration of the QSAR/QSPR workflow using ML/DL. Reprinted from Yang et al. (2019a).

Recently, Graph Neural Networks (GNNs) emerged as a class of deep learning methods and presented a novel solution to this problem (Duvenaud et al., 2015; Li et al., 2019b; Stokes et al., 2020). Until then, the reigning paradigm was to map molecules to a numerical vector in a fixed, pre-defined space that represents the presumably most important features of the molecule that determine the target property. Expert knowledge is necessary to choose these features and the result of the prediction is ultimately biased by this knowledge and perhaps incorrect if the feature selection was sub-optimal (Wieder et al., 2020). GNNs solve this problem by automatising the selection process and learning the space itself to find the most suitable representation from a pool of given features for a specific task. Figure 1 shows the branching between fixed and learned representations in a property prediction workflow. Not long ago, the potential of employing Graph Neural Networks in drug discovery was

highlighted by the discovery of a new broad-spectrum bactericidal antibiotic 'halicin' after decades of stagnation in the field. Stokes et al. (2020) employed a directed-message passing neural network Yang et al. (2019a) for both target selection to predict growth inhibitory effects against E. Coli. and ADME/T modelling predicting the toxicity of potential candidates.

1.2 Outline

The goal of this thesis is to give an overview of GNNs in the context of learning the input representation of molecules for machine learning prediction tasks in drug discovery. We present their technical background together with an analysis of the results that have been obtained using GNNs. The outline for the thesis is as follows: Firstly, we give a brief overview of other techniques that have been used for featurization. Most prominently, these concern molecular descriptors and fingerprints. Consecutively, the technical background to understand GNNs is presented. This involves a detailed explanation of circular fingerprints motivating the introduction of GNNs, a summary of molecular graphs building the basis for employing GNNs and ultimately an introduction to Message-Passing Neural Networks. These have been introduced as a general technical framework summarising some of the most prominent implementations of GNNs for drug discovery. In the next section we depict the application of GNNs in drug discovery by explaining two studies in detail. Finally, we discuss the advantages an disadvantages of GNNs and give and outlook for future research.

1.3 Overview of methods for featurization

A variety of methods have been used to design the input features for machine learning prediction tasks in drug discovery. Most of them fall into the category of fixed representations characterised by a pre-defined target space. These fixed representations can be broadly separated into molecular descriptors and fingerprints. Additionally, another form of learned representation has been proposed under the name of sequence models. Instead of operating on molecular graphs, they work with linear notations like SMILES (Weininger, 1988) or InCHI (Heller et al., 2015). These can be used to learn a representation by employing Recurrent Neural Networks (RNNs) or long short-term memory (LSTM) cells. Furthermore, Honda et al. (2019) introduced a SMILES transformer to predict molecular properties. We will not discuss these sequence models in more detail in this report.

Descriptors. As the name suggests, (numerical) descriptors represent molecules by describing their properties. 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 a useful number or the result of some standardized experiment'. As this definition suggests descriptors can be given by all kinds of properties that represent chemical information about the molecule. This makes them a straightforward, yet versatile means to encode a molecule mathematically. Critically, these properties need to be 'useful'. Some basic requirements for the usefulness of a descriptor are outlined by Mauri et al. (2016) concerning for example

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

However, these requirements ultimately depend on the application domain (Jiang et al., 2020b). This highlights a drawback of the descriptor approach since their usefulness as molecular representations for property prediction is constrained by problem-specific knowledge (Shen & Nicolaou, 2019).

Due to the enormous amount of different descriptors that have been proposed there are a lot of ways to categorise them. One attempt is based on the nature of the structural information that they require (Guha & Willighagen, 2013) and classifies them as constitutional, topological, geometric and quantum mechanical descriptors. Constitutional descriptors are the most rudimentary form of descriptors not taking into account any spatial information about the molecule. Quantum mechanical descriptors are among the most complex descriptors and their high computational requirements can make them unsuitable for large-scale screenings.

Popular examples of descriptors for QSAR models include

- the Wiener index (Wiener, 1947; Nikolić et al., 2001)
- the coulomb matrix (Rupp et al., 2012) or
- symmetry functions (Behler & Parrinello, 2007).

Fingerprint Vectors. Descriptors are often 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) keys (Durant et al., 2002) 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 edge segments up to a given length in a molecular graph (day, 2021) or in a circular manner as for extended connectivity fingerprints (ECFPs).

ECFPs (Rogers & Hahn, 2010) are among the most popular fingerprints and they are often used as baseline results for the development of new featurizations techniques (Li et al., 2017; Wu et al., 2018; Stokes et al., 2020). Furthermore, they motivated the introduction of Graph Neural Networks that adapt their aggregation process to become differentiable. For this reason, we depict the technical details of ECFPs in section 2.2.

Other circular fingerprints can be obtained from ECFPs by selecting different atom identifiers. This gives rise to fingerprints like FCFPs (Functional Class Fingerprints) that are 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. Expert knowledge is needed to make a meaningful decision (Wieder et al., 2020).

Like numerical descriptors, fingerprints are also a powerful means to represent molecules in form of a fixed-size vector. They differ from descriptors by implicitly encoding the molecular structure. However, they suffer from a similar drawback as their usefulness for QSAR models is dependent on the choice of the atom identifier.

2 Technical Background

2.1 Molecular Graphs

Molecular graphs are a convenient means to represent molecules in two dimensions being the starting point for using various molecular representations such as circular fingerprints or GNNs. 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 2. 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.

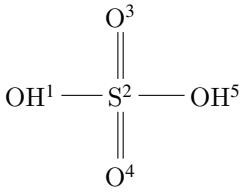

Figure 2: Molecular graph of sulfuric acid.

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

Figure 3: Adjacency matrix of the molecular graph representing sulfuric acid given the node ordering.

Figure 3 shows the adjacency matrix corresponding to the graph in Figure 2. The ordering of the vertices is indicated by superscripts. If we assumed a different ordering of the vertices this would result in a permutation of the rows and columns of the adjacency matrix. For representation methods that use molecular graphs this means that the generated representation needs to be invariant to permutations of the adjacency 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 for a node feature matrix is shown in Figure 4. Finally, the edge feature matrix contains one row for every edge in the graph and the number of columns may vary depending on the number of encoded features, see Figure 5.

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

Figure 4: Example feature matrix of the graph in Figure 2. 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 5: Example edge feature matrix of the graph in Figure 2. The chosen 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. Firstly, 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. Secondly, 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.

2.2 Extended-Connectivity Fingerprints

Extended-Connectivity fingerprints belong to the class of circular fingerprints and are based on a variation of the Morgan algorithm (Morgan, 1965) which is outlined in Algorithm 1. Given a molecular graph, they assign each atom a unique identifier that is based on a selection of properties. Then, this information is propagated from each atom to its neighbours. Contrary to the original

Algorithm 1: Morgan Algorithm

```
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
```

Morgan algorithm, this process is terminated after a pre-defined number of iterations. In the following we detail the generation of ECFPs.

Firstly, 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. This way, each atom updates its own features by incorporating those of its neighbours. 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 they encode the same substructure of the molecule after an equal number of iterations . This may occur for example for the nitrogen and oxygen atoms at the top and right of the structure shown in Figure 6. 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

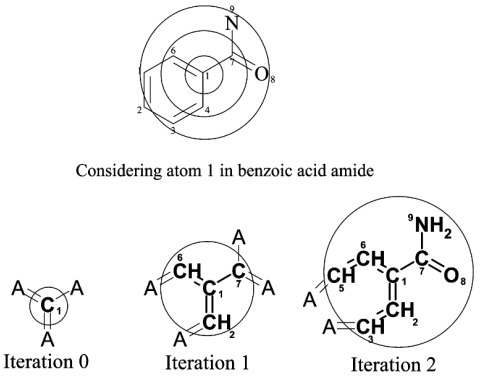

Figure 6: Illustration of the iterative updating in the computation of the ECFPs. In this example the atom type is used as an identifier. In iteration 0 the middle atom' identifier only represents the information about its own type. After the first iteration it has aggregated the information from its immediate neighbors and after the second iteration the represented substructure has grown even further. Reprinted from Rogers & Hahn (2010).

n times using the updated identifiers of each atom as the the initial identifiers for the next step. This way, the identifier of each atom represents substructures of increasing sizes as illustrated in Figure 6. After the completion of the n steps, numerically equal values are removed from the set A and the remaining identifiers define the circular fingerprint. This final set of identifiers can be interpreted in a similar fashion as dictionary-based fingerprints. Each identifier in A corresponds to a bit in a huge

virtual bit string denoting the presence or absence of a particular substructural feature (ecf, 2021). This representation allows for the folding of the string into a bit vector of consistent size, e.g. 1024 bits.

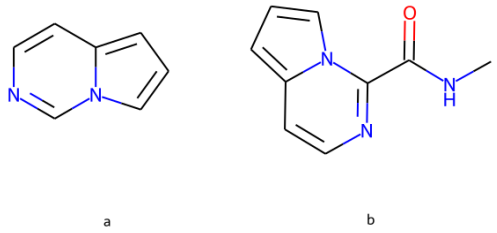

Figure 7: Molecular graphs obtained using the code in Appendix A.

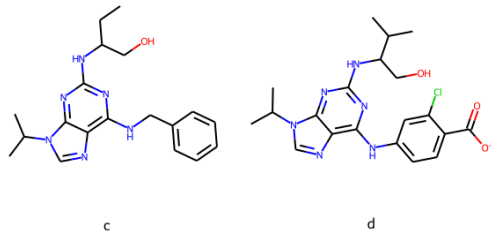

Figure 8: Molecular graphs obtained using the code in Appendix A.

To better understand the aggregation scheme we compare the predicted similarity scores obtained using two pairs of molecules in Figure 7 and 8 for $n = 1, \ldots, 5$. The results are described in Table 1. Note that we adopt the notation used in the literature where the number behind 'ECFP' denotes the maximum diameter of substructures considered. For instance, ECFP4 corresponds to $n = 2$. 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 we expand the size of the substructure that each atom's identifier represents, more dissimilarities between the molecules are discerned as expressed by a drop in the scores. However, the scores decrease more slowly with an increasing n until they eventually plateau. This is because no new distinct substructures are identified after a certain number of iterations. This corresponds to no new structurally unique identifiers being generated in the computation of the fingerprint as outlined above. We conclude that n can be regarded as parameter that determines how precisely two molecules are distinguished. In the literature, ECFP4 fingerprints are the common choice.

Molecules	**ECFP2**	**ECFP4**	**ECFP6**	**ECFP8**	**ECFP10**
a & b	56.25%	46.15%	34.29%	32.43%	32.43%
c & d	68.66%	58.71%	52.86%	46.32%	43.09%

Table 1: Sørensen-Dice similarity values (Sorensen, 1948; Dice, 1945) using different fingerprints for molecules in Figure 7 and Figure 8 respectively

2.3 Message Passing Neural Networks

Convolutional Neural Networks (LeCun et al., 1999) 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 specific GNNs can be restored.

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 updates 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 which allows for emphasising more important features while less important one contribute less to the final representation. As an input MPNNs 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 $\boldsymbol{x}_v$ obtained from the node feature matrix. 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 by 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.

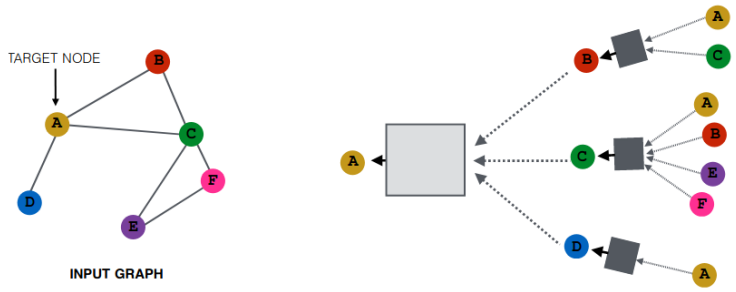

Figure 9: Illustration of the message passing in a MPNN. Reprinted from Hamilton et al. (2018).

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.

Note the parallels to the computation of circular fingerprints as outlined in section 2.2. This is no coincidence as the introduction of one of the earliest GNNs (Duvenaud et al., 2015) was motivated by replacing the steps in the computation of circular fingerprints by differentiable analogs. This will be explained in detail in section 3.1.

3 RESULTS

In this section we study the application of GNNs to drug discovery. Their main motivation lies in overcoming the dependence on an explicit choice of features that are used to compute a fixed representation. For descriptors this choice is given by the selection of properties to be represented by the descriptor. Molecular fingerprints require this selection in form of the identifier that is used to initialise the atom's values. This manual feature design means that a bias is imposed and the resulting method can only perform as well as the feature selection allows (Merkwirth & Lengauer, 2005). GNNs remedy this problem by using deep learning to learn the features most relevant to a property of interest.

We present two applications of GNNs to drug discovery that highlight their potential as state-of-the-art featurization techniques. Anticipated benefits were listed by (Shen & Nicolaou, 2019) and comprise:

1. a compact final representation of the molecule,
2. enhanced interpretability,
3. the possibility to use attention algorithms that allow the model to focus on the most relevant parts of the molecule (Li et al., 2019a; Xiong et al., 2020)
4. an improvement in predictive performance given large enough data sets (Yang et al., 2019a).

We will review these factors in the next section to understand and discuss their validity based on the presented results in this section.

3.1 GNNS FOR THE PREDICTION OF SOLUBILITY, DRUG EFFICACY AND PHOTOVOLTAIC EFFCIENCY

Setup. 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 'neural graph' 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. These adaptations are illustrated by a comparison of both algorithms in Figure 10. Note that invariance to the node ordering is achieved by using a permutation-invariant aggregation function, i.e. summing.

Algorithm 1 Circular fingerprints

Input: molecule, radius R, fingerprint length S
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule
  $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R ▷ for each layer
  for each atom a in molecule
    $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
    $\mathbf{v} \leftarrow [\mathbf{r}_a, \mathbf{r}_1, \dots, \mathbf{r}_N]$ ▷ concatenate
    $\mathbf{r}_a \leftarrow \text{hash}(\mathbf{v})$ ▷ hash function
    $i \leftarrow \text{mod}(r_a, S)$ ▷ convert to index
    $\mathbf{f}_i \leftarrow 1$ ▷ Write 1 at index
Return: binary vector $\mathbf{f}$

Algorithm 2 Neural graph fingerprints

Input: molecule, radius R, hidden weights $H_1^1 \dots H_R^5$, output weights $W_1 \dots W_R$
Initialize: fingerprint vector $\mathbf{f} \leftarrow \mathbf{0}_S$
for each atom a in molecule
  $\mathbf{r}_a \leftarrow g(a)$ ▷ lookup atom features
for $L = 1$ to R ▷ for each layer
  for each atom a in molecule
    $\mathbf{r}_1 \dots \mathbf{r}_N = \text{neighbors}(a)$
    $\mathbf{v} \leftarrow \mathbf{r}_a + \sum_{i=1}^{N} \mathbf{r}_i$ ▷ sum
    $\mathbf{r}_a \leftarrow \sigma(\mathbf{v} H_L^N)$ ▷ smooth function
    $\mathbf{i} \leftarrow \text{softmax}(\mathbf{r}_a W_L)$ ▷ sparsify
    $\mathbf{f} \leftarrow \mathbf{f} + \mathbf{i}$ ▷ add to fingerprint
Return: real-valued vector $\mathbf{f}$

Figure 10: Comparison of the algorithm that generated circular fingerprints with that for generating neural graph fingerprints. Note that the left algorithm uses the interpretation of the atom's hashed identifiers as indices of bits in an array as explained in section 2.2. Reprinted from Duvenaud et al. (2015).

This method for generating neural graph fingerprints is summarised by the message-passing framework presented in section 2.3 using the following message- and readout functions:

The message function M_t is the same across all time steps and given by

$$M(\boldsymbol{h}_v, \boldsymbol{h}_w, \boldsymbol{e}_{vw}) = \frac{1}{|N(v)|}\boldsymbol{h}_v + \boldsymbol{h}_w$$

The update and readout functions are given by

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

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

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

with learnable matrices $\boldsymbol{W}_t$ for all time steps t.

Experiments Duvenaud et al. (2015) applied their proposed architecture to predict the following properties of molecules:

- Aqueous solubility as in (Delaney, 2004),
- efficacy as a drug against a parasite that causes malaria as measured by Gamo et al. (2010),
- photovoltaic efficiency as in (Hachmann et al., 2011).

To obtain the predicted property from the output of the GNN they used a fully connected neural network that receives the output of the GNN and computes the predicted value of the property. This whole architecture is trained in an end-to-end fashion through backpropagation such that the GNN can adapt its weights to favour the contribution of the most relevant features for the desired property. Figure 11 reports the RMSEs of both the GNN and ECFPs as input features for the neural network.

Dataset Units	Solubility [4] log Mol/L	Drug efficacy [5] EC_{50} in nM	Photovoltaic efficiency [8] percent
Predict mean	4.29 ± 0.40	1.47 ± 0.07	6.40 ± 0.09
Circular FPs + linear layer	1.71 ± 0.13	$\mathbf{1.13 \pm 0.03}$	2.63 ± 0.09
Circular FPs + neural net	1.40 ± 0.13	1.36 ± 0.10	2.00 ± 0.09
Neural FPs + linear layer	0.77 ± 0.11	$\mathbf{1.15 \pm 0.02}$	2.58 ± 0.18
Neural FPs + neural net	$\mathbf{0.52 \pm 0.07}$	$\mathbf{1.16 \pm 0.03}$	$\mathbf{1.43 \pm 0.09}$

Figure 11: Comparison of the root-mean-square error on the three data set mentioned above for circular fingerprints and neural fingerprints (GNN). Reprinted from Duvenaud et al. (2015).

Overall, we see that the GNN gives slightly better results than ECFPs. However, the advantage of GNNs is not consistent across all data sets. For predicting drug efficacy the combination of using ECFPs with linear layers in the fully connected network achieves the best scores. However, GNNs perform only slightly worse.

3.2 GNNs FOR ANTIBIOTIC DISCOVERY

The second application of GNNs for drug discovery is concerned with antibiotic discovery. The discovery of new antibiotics is becoming increasingly difficult due to the dereplication problem that the same molecules are discovered over and over (Cox et al., 2017). Given the simultaneous stagnation of the success of existing methods for antibiotic discovery and development of antibiotic-resistant determinants this creates a great urge for new methods to enter the stage.

Stokes et al. (2020) employed a GNN for both target identification and the prediction of toxicity of potential antibiotic candidates. The molecular representation was built using a directed-message passing neural network Yang et al. (2019a). This D-MPNN extends the framework from section 2.3

by considering directed messages associated with bonds instead of atoms. This is motivated by the hope to avoid messages being passed forth and back between two atoms which can lead to noise in the representation.

The full workflow can be separated into three stages. The fist stage concerns the training of the model and a classifier. Since (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 2.3. 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 second 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 halicin would not have been found if molecular fingerprints had been used. However, 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.

4 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. Recently, a variety of Graph Neural Networks has been introduced as a way to automatise 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 Networks manage to learn a continuous vector representation that is highly attuned to the property of concern. After reviewing two applications of GNNs in drug discovery we come back to the anticipated benefits oultined in section 3.

While GNNs do learn a compact representation compared to molecular fingerprints if they are stored as sparse bit-vectors, this is at the downside of substantially higher computational costs. Molecular fingerprints as well as most descriptors can be computed immediately and serve as an off-the-shelf representation of molecules. The higher computational costs of GNNs is not only attributed to the time

that it takes to train the network and adapt the weight. Beyond that, they require resource-intensive hyperparameter tuning in order to find a suitable configuration of weights.

In terms of interpretability, the comparison of fixed representations and GNNs is more level. Methods like studying the activations of the GNN can be used in order to identify the substructures that maximally excite certain feature maps in the GNN as described by Duvenaud et al. (2015). However, these methods are quite costly to perform and ultimately do not help to understand the contribution of the initialised features to the final representations. For molecular fingerprints on the other side, 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 was selected. This might be even more helpful than understanding the correspondence between substructures and feature maps in a GNN.

As another anticipated benefit the use of attention algorithm was mentioned in section 3. While attention algorithms have been introduced as an extension for GNNs (Xiong et al., 2020), the ultimate benefit of GNNs is more fundamental. GNNs are extremely flexible and can be extended very easily. They are a current topic of research even beyond the prediction of chemical properties and general advances of the framework can have immediate impacts on their application in drug discovery. This makes them promising to take over the paradigm in molecular representations as their advances are much more far reaching compared to the development of a new descriptor.

Finally, we compare the predictive accuracy of GNNs to that of molecular fingerprints and descriptors. 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. 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 are 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. Therefore, the question of which method is the state of the art remains to be answered. However, the ongoing research on GNNs is likely to further increase their performance, ultimately leaving fixed representations behind.

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. Firstly, 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.

Extensions ...

5 Conclusion

In this report we have studied the role of GNNs for generating molecular representations that can be used as an input for machine learning methods predicting molecular properties. These turned out to be an extension of molecular fingerprints by making their aggregation differentiable and learning a weighting of the atom identifiers according to their relevance for the target property. Two applications of GNNs for drug discovery were explained in detail. The first by Duvenaud et al. (2015) concerned the prediction of molecular properties like solubility or their efficacy as a drug. The second application by Stokes et al. (2020) was about the application of GNNs in the process of finding a new antibiotic. Finally, we compared learned representations with fixed representations in terms of accuracy, computational costs and interpretability. Despite fixed approaches outperforming GNNs

in terms of their computational costs, we hypothesised Graph Neural Networks to be a key future technology for molecular property predictions. This is due to their performance that is likely to be further improved in the line of ongoing research and +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.

Extended connectivity fingerprint ecfp. https://docs.chemaxon.com/display/docs/extended-connectivity-fingerprint-ecfp.md, 2021. Retrieved: May 2, 2021.

Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98:146401, Apr 2007. doi: 10.1103/PhysRevLett.98.146401. URL https://link.aps.org/doi/10.1103/PhysRevLett.98.146401.

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.

Georgina Cox, Arthur Sieron, Andrew M. King, Gianfranco De Pascale, Andrew C. Pawlowski, Kalinka Koteva, and Gerard D. Wright. A common platform for antibiotic dereplication and adjuvant discovery. *Cell Chemical Biology*, 24(1):98–109, 2017. ISSN 2451-9456. doi: https://doi.org/10.1016/j.chembiol.2016.11.011. URL https://www.sciencedirect.com/science/article/pii/S2451945616304342.

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.

John S Delaney. Esol: estimating aqueous solubility directly from molecular structure. *Journal of chemical information and computer sciences*, 44(3):1000–1005, 2004.

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

Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.

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.

Francisco-Javier Gamo, Laura M Sanz, Jaume Vidal, Cristina De Cozar, Emilio Alvarez, Jose-Luis Lavandera, Dana E Vanderwall, Darren VS Green, Vinod Kumar, Samiul Hasan, et al. Thousands of chemical starting points for antimalarial lead identification. *Nature*, 465(7296):305–310, 2010.

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.

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.

Johannes Hachmann, Roberto Olivares-Amaya, Sule Atahan-Evrenk, Carlos Amador-Bedolla, Roel S Sánchez-Carrera, Aryeh Gold-Parker, Leslie Vogt, Anna M Brockway, and Alán Aspuru-Guzik. The harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid. *The Journal of Physical Chemistry Letters*, 2(17): 2241–2251, 2011.

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.

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.

Stephen R Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. Inchi, the iupac international chemical identifier. *Journal of cheminformatics*, 7(1):1–34, 2015.

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.

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.

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.

Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pp. 319–345. Springer, 1999.

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.

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.

Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108:058301, Jan 2012. doi: 10.1103/PhysRevLett.108.058301. URL https://link.aps.org/doi/10.1103/PhysRevLett.108.058301.

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.

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.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005. URL https://pubs.acs.org/doi/abs/10.1021/ci00057a005.

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.

Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 2020. ISSN 1740-6749. doi: https://doi.org/10.1016/j.ddtec.2020.11.009. URL https://www.sciencedirect.com/science/article/pii/S1740674920300305.

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, 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

List of Tables

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

[1]:
```
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

from IPython.display import SVG
from rdkit.Chem import AllChem
```

[2]:
```
m1s = [Chem.MolFromSmiles("c1nccc2n1ccc2"), Chem.
↪MolFromSmiles("CNC(=O)c1nccc2cccn12")]
m2s = [Chem.MolFromSmiles("CCC(CO)Nc1nc(NCc2ccccc2)c2ncn(C(C)C)c2n1"),
Chem.MolFromSmiles("CC(C)C(CO)Nc1nc(Nc2ccc(C(=O)" + \
"[O-])c(Cl)c2)c2ncn(C(C)C)c2n1")]
```

[3]:
```
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")
```

[4]:
```
### 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
```

[5]:
```
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
0.5447368421052632
```

[8]:
```
#Morgan fingerprints
for k in range(1,6):
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.
↪6865671641791045
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.
↪5871559633027523
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.
↪5285714285714286
Dice Similarity of Morgan Fingerprints of a and b using r = 4
0.32432432432432434
Dice Similarity of Morgan Fingerprints of c and d using r = 4 0.
↪4634146341463415
Dice Similarity of Morgan Fingerprints of a and b using r = 5
0.32432432432432434
Dice Similarity of Morgan Fingerprints of c and d using r = 5 0.
↪430939226519337
```