Machine Learning: Science and Technology

PAPER • OPEN ACCESS

Chemistry-informed macromolecule graph representation for similarity computation, unsupervised and supervised learning

To cite this article: Somesh Mohapatra *et al* 2022 *Mach. Learn.: Sci. Technol.* **3** 015028

MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
2 December 2021

REVISED
24 January 2022

ACCEPTED FOR PUBLICATION
11 February 2022

PUBLISHED
21 February 2022

PAPER

Chemistry-informed macromolecule graph representation for similarity computation, unsupervised and supervised learning

Somesh Mohapatra, Joyce An and Rafael Gómez-Bombarelli*

Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA, United States of America

* Author to whom any correspondence should be addressed.

E-mail: rafagb@mit.edu

Keywords: macromolecules, representation learning, similarity computation, graph neural network, attribution analysis

Supplementary material for this article is available online

Abstract

The near-infinite chemical diversity of natural and artificial macromolecules arises from the vast range of possible component monomers, linkages, and polymers topologies. This enormous variety contributes to the ubiquity and indispensability of macromolecules but hinders the development of general machine learning methods with macromolecules as input. To address this, we developed a chemistry-informed graph representation of macromolecules that enables quantifying structural similarity, and interpretable supervised learning for macromolecules. Our work enables quantitative chemistry-informed decision-making and iterative design in the macromolecular chemical space.

1. Introduction

Macromolecules are ubiquitous and indispensable. Biological macromolecules form the basis of life, serving as drivers of survival and growth functions [1]. Synthetic macromolecules have been engineered by humans based on composition [2–4] and topology [5] to design structural components [6, 7], sensors [8], responsive materials [9], drugs [10], digital information storage [11], and much more.

An individual macromolecule is distinguished by the identity and spatial arrangement of its monomers and linkages [12]. Monomer and linkage are functions of atomic composition, connectivity and stereochemistry, while spatial arrangement of the monomers and linkages dictates the topology. Experimentalists and theoreticians have explored a vast chemical space by varying monomers [13, 14], linkages [15], and topologies—both linear [4] and non-linear such as branched [16], star [17], and bottle-brush [18]. As a result of such chemical diversity, representing, comparing, and learning over macromolecules with different monomers, linkages, and topologies emerges as a critical challenge.

Linear biological macromolecules, such as proteins and DNA/RNA, are easily represented as strings of one/three-letter monomer codes. However, machine-readable macromolecule representations, such as hierarchical editing language for macromolecules [19], IUPAC international chemical identifier [20], CurlySMILES [21] (where, SMILES is simplified molecular-input line-entry system) and BigSMILES [22], do not always support non-linear topologies, require a fair amount of customization, and have non-canonical variants. In a recent attempt, glycans, which are non-linear biological macromolecules, were represented as SMILES-like sequences, where groups of monosaccharides were binned into ‘glycowords’ and placed in hierarchical brackets [23].

Likewise, similarity computation for macromolecules has mostly been limited to linear sequences [23–26], leveraging sequence alignment using the Smith–Waterman [27] or Needleman–Wunsch [28] algorithms, and scoring with substitution matrices, such as BLOSUM62 [29] (for proteins) and GLYSUM [23] (for glycans). These substitution matrices are based on evolutionary statistics and cannot be used for non-natural building blocks since they lack a general way to quantify chemical similarity. In the case of non-linear macromolecules, alignment of glycans has been explored using q-grams [30], tree matching methods [31–33], and tree kernels [34]. Unfortunately, these methods are tailored to specific classes of macromolecules and do not extend to the general macromolecular chemical space.

Unsupervised and supervised machine learning (ML) applications to individual macromolecule classes, such as proteins, have been very successful but typically rely on sequence-based representations that are tailored for linear architectures [35, 36]. For instance, unsupervised learning of protein sequences has resulted in functional annotation and identification of sub-families for yet unseen sequences [37], in addition to creating information-rich embeddings for downstream data-poorer property prediction tasks [36, 38]. On the supervised learning front, for artificial polymers, the PolymerGenome and similar works have used hierarchical fingerprints to predict glass transition temperature, dielectric point and other macromolecular properties [39, 40] and there have been attempts to extrapolate macromolecular property by training over monomer input features [41–43]. However, these methods do not extend well to macromolecules with complex topologies, such as glycans or biohybrid sequence-defined polymers, which exhibit non-linear structure and higher levels of monomer and linkage diversity.

In this work, we developed a graph representation for macromolecules. Graphs are a natural and more general macromolecule representation, which can handle linear, branched and cyclic topologies along with any monomer and linkage composition. The representation generalizes ideas of macromolecule similarity, from sequence alignment to structural similarity, between complex topologies. Using chemical similarity between monomers through cheminformatic fingerprints and exact graph edit distances (GEDs) or graph kernels to compare topologies, the representation allows for the quantification of the chemical and structural similarity of two arbitrary macromolecule topologies. To investigate the relationship between chemical similarity and functionality, we used unsupervised learning over similarity vectors obtained from aforementioned similarity computation methods.

Leveraging advancements in ML over graph representations has achieved state-of-the-art results across several fields [44], including chemistry and life sciences, where graph neural networks (GNNs) have become the modern workhorse for molecular property prediction [45–49]. We coupled macromolecule graphs to supervised GNN models to learn structure-property relationships. Further, we used attribution methods compatible with GNNs to highlight how input features are relevant to model predictions of target properties [50, 51].

2. Results and discussion

2.1. Text file system converts macromolecule structure to machine-readable graph

We developed a generalized text file system to convert a macromolecule structure into a machine-readable format (figure 1(A) and SI section 2 available online at stacks.iop.org/MLST/3/015028/mmedia). The text file has three sections—SMILES, MONOMERS, and BONDS, inspired by the PDB file format [52]. Under SMILES, monomer and bond names followed by the stereochemical SMILES are noted. MONOMERS enumerates indices of all nodes numbered from 1 to n, where n is the total number of monomers, followed by the monomer names. Similarly, BONDS lists indices of connections between monomer indices, followed by bond names. In this way, we are able to incorporate complexity from the level of individual atoms to the full macromolecular structure.

For our experiments, the macromolecule text files were then processed into attributed NetworkX graphs [53], with monomers as nodes and bonds as edges (SI section 3). In line with our earlier work where fingerprint-based monomer representations worked effectively for macromolecule property prediction, the monomer and bond molecules were featurized using standard ECFP [35]. The fingerprints capture the atomic connectivity of the monomer/bond molecule as a series of bits using circular atom neighborhoods, for each constituent node or edge of the macromolecule graph, encoding the macromolecules in their native structure with explicit featurization of the stereochemistry and topology. We optimized the radius of the atomic neighborhood and the dimension of the fingerprint, by analyzing the distribution of Tanimoto similarity for the individual monomers and bonds. This system allows us to represent any macromolecule structure, irrespective of monomer and linkage types, and topology, using the same framework. Along with fingerprints, we benchmarked models with discrete one-hot encodings of the monomers, in order to understand the effect of these chemical features.

2.2. Graph similarity measures enable similarity computation for arbitrary macromolecules

To compute the (dis)similarity between macromolecule graphs, we used exact GED and graph kernels (figure 1(B)). GED [54] computes the dissimilarity between two graphs by assigning node and edge insertion/deletion/substitution cost. For insertion and deletion of node/edge, we add a fixed cost to the distance, while for substitution, we multiply a constant cost with the Tanimoto dissimilarity of the molecules being substituted (SI section 4.2). Tanimoto dissimilarity is a metric to compute dissimilarity between two

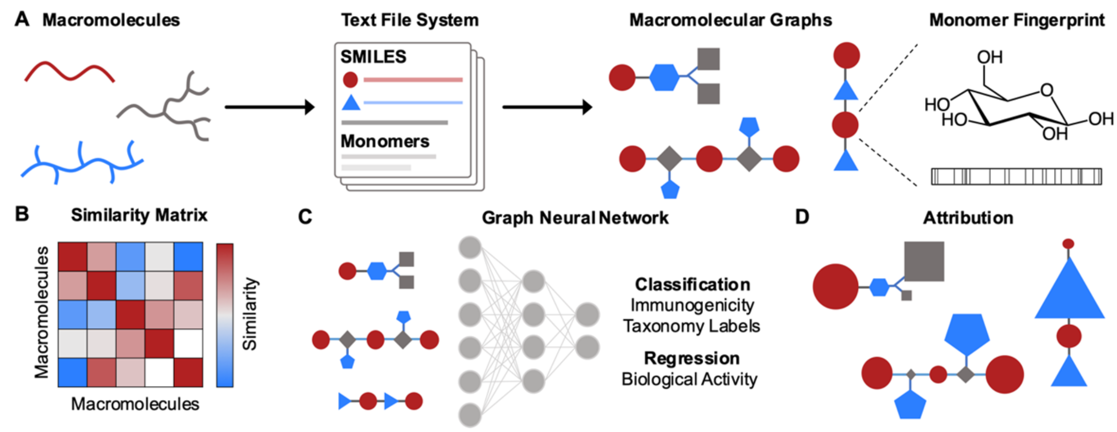


Figure 1. Macromolecules represented as graphs enable similarity computation and interpretable ML. (A) Macromolecular structures are converted into text files. The text files are parsed into network graphs, where monomers are nodes and bonds are edges. Each text file enumerates SMILES for monomers and bonds, node indices corresponding to monomers, and pairs of node indices for bonds. Node and edge attributes are fingerprints of respective molecules. (B) Pair-wise similarity matrix obtained for the library of macromolecules, and dimensionality reduction followed by clustering of similarity vectors corresponding to each macromolecule. (C) GNNs learn over macromolecule graphs for various tasks. (D) Graph attribution provides insight into the model's decision-making process by highlighting the relative importance of different nodes/monomers (denoted as node size) in making a specific prediction.

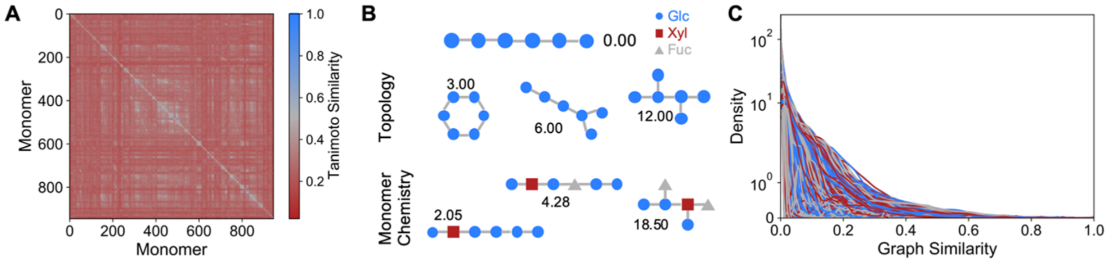


Figure 2. Chemical similarity computation of macromolecules using GEDs and graph kernels. (A) Pairwise chemical similarity between glycan monomers using Tanimoto distance on ECFP fingerprints is typically low. (B) GED dissimilarities between a linear glycan consisting of six glucose monomers with itself and six related glycans with modified topology and monomer chemistry, with node/edge insertion/deletion cost as 3, and node/edge substitution cost as 3 multiplied by the Tanimoto distance. (C) Graph kernel computed similarity vectors overlaid as histograms for 8899 glycans in the curated data set. Similarity vector is the vector of similarities of a single glycan with rest of the library. The vectors were normalized to the maximum in the respective vector.

bit-vectors on a scale of 0 and 1, where self-dissimilarity is zero. This process is analogous to sequence alignment using methods like BLAST [24]. However, instead of scoring using evolutionary statistics-based substitution matrices, such as BLOSUM62, the use of Tanimoto dissimilarity matrices over molecular fingerprints allows us to extend the similarity computation to unnatural monomers. We have demonstrated the similarity computation for a linear glycan with six additional glycans of different topology and/or monomer chemistry, as well as with itself, using Tanimoto chemical similarity matrix (figures 2(A) and (B)).

As the size of both individual graph and dataset increase, computing exact GED becomes computationally untractable, since GED belongs to a class of problems that are non-deterministic polynomial time-hard, otherwise known as NP-hard problems. To scale the similarity computation to large datasets, we used graph kernels, specifically propagation attribute kernels, to obtain approximate similarity matrices [55, 56] (figure 2(C) and SI section 4.3). The propagation attribute kernel method captures local monomer node information and propagates this information along the bond edges, thereby capturing both local and global information, to produce a similarity score. The information flow in the propagation attribute kernel is similar to message passing in graphs, making it an ideal choice for macromolecule graphs represented through featurized node and edge, given the success of graph convolutional NNs for supervised tasks on macromolecule graphs.

2.3. Unsupervised learning separates functional macromolecules into distinct regions

We used dimensionality reduction methods, such as principal component analysis (PCA) [57], t-distributed stochastic neighbor embedding (t-SNE) [58] and uniform manifold approximation and projection (UMAP) [59] in combination with our similarity computations for unsupervised learning (SI section 5).

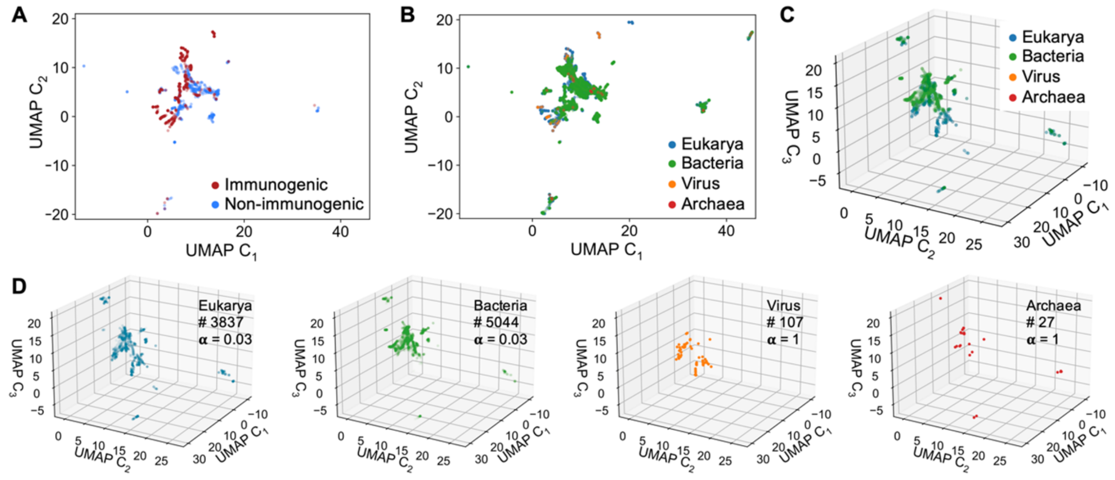


Figure 3. Unsupervised learning clusters glycans by immunogenicity and taxonomy. (A) Two-component UMAP over the macromolecule similarity matrix, colored by immunogenicity, for 1313 glycans with immunogenicity labels. The components of the reduced dimensions are denoted as C_1 and C_2 in the axis labels. Visualization of (B) two and (C) three components UMAP, colored by domain. (D) Glycans corresponding to individual domains are shown, with the text noting the domain, number of glycans, and color transparency (α) of each point on a scale of 0 and 1, where zero is transparent and one is opaque. The color transparency setting is used to appropriately visualize the dense clusters of glycans belonging to eukarya and bacteria domains. The components of the reduced dimensions are denoted as C_1, C_2 and C_3 in the axis labels.

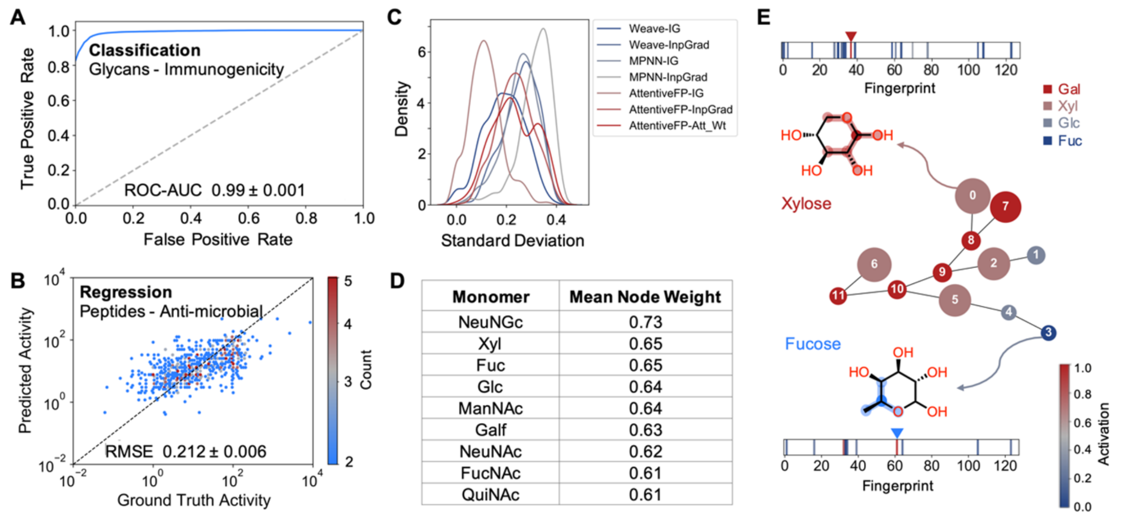
Conventional implementations of dimensionality reduction methods are based on feature vectors, so macromolecule graphs cannot be processed directly. Instead, we used an approach inspired by multidimensional scaling and applied linear and non-linear dimensionality reduction to the similarity matrix obtained using graph kernels [60]. The graph similarity matrices make for a powerful representation, since they encode chemical and topological pairwise similarity across the dataset, without resorting to representing each macromolecule as a vector.

For glycans with immunogenicity labels, we observed that the non-immunogenic and immunogenic glycans are in nearly distinct regions (figure 3(A)). In a similar experiment we colored glycans by domain in two- and three-component UMAP plots (figures 3(B) and (C)). Noting that the taxonomic complexity was not being adequately captured by the two-component plot, we used a three-component plot. In the individual plots, we observed that glycans belonging to eukarya, bacteria, and virus clustered in distinct regions, with the bacteria glycans at the core, eukarya glycans spreading out of the core, and virus glycans at the fringes (figure 3(D)). Benchmarking against PCA and t-SNE, we observed that PCA was able to capture the global structure separating the immunogenic and non-immunogenic glycans, while t-SNE was better at capturing the local structure (SI figures 24 and 25). In contrast, UMAP performed better in separating functional glycans, for both local and global structures.

To check if more components in dimensionality reduction could help in finding distinct clusters, we performed dimensionality reduction for {2, 3, 5, 10, 30, 50} components and used hierarchical density-based clustering of applications with noise, an unsupervised clustering algorithm, to figure out the number of clusters [61]. We found that the numbers of clusters were similar across the different numbers of components, and in the low 400s (SI figure 22). The high number of clusters indicates the diversity of the space, and the differences in graph similarity of glycans with distinct taxonomy. As a further check of the validity of the clustering approach, we plotted the histograms of the glycans assigned to each cluster. Across all the components, the histograms were consistent with the number of glycans in each of them (SI figure 23).

2.4. GNNs predict macromolecule properties with high accuracy

We evaluated five different GNN model architectures to classify glycans by immunogenicity and taxonomy levels and predict the anti-microbial activity of peptides (figures 4(A), (B) and SI section 6). Each model architecture was trained over fingerprint and one-hot node and edge attributes on 60%, validated on 20%, and tested on held-out 20% data set, all determined using random splits. To report final metrics on validation and test datasets, for each model architecture, we averaged values obtained from 25 individual models—top five hyperparameter sets, each trained with five random seeds. Unlike previous implementations with all-atom GNNs [62], we used a hierarchical GNN model trained over monomer nodes and linkage edges, featurized using fingerprints. The hierarchical leads to learning over a coarse-grained representation without trying to learn over an all-atomistic representation. Moreover, this approach enables



Monomer	Mean Node Weight
NeuNGc	0.73
Xyl	0.65
Fuc	0.65
Glc	0.64
ManNAc	0.64
Galf	0.63
NeuNAc	0.62
FucNAc	0.61
QuiNAc	0.61

Figure 4. Supervised learning with attribution analysis enables robust and interpretable property prediction for macromolecule graphs. (A) Mean ROC-AUC curve for classification of immunogenic and non-immunogenic glycans, averaged across all 25 AttentiveFP models (five random seeds, with top five sets of hyperparameters), with the standard deviation shaded in light blue too insignificant to be visible in the graph. (B) Mean predictions of anti-microbial activity of peptides, averaged across all 25 Weave models (five random seeds, with top five sets of hyperparameters). (C) Distribution of normalized node weights for different model architecture—attribution methods. Standard deviation of node weights obtained from 25 different model implementations—top five hyperparameter sets and five random seeds—were pooled across all immunogenic glycans to calculate the distribution for each model architecture and attribution combination. (D) Top monomers contributing to glycan immunogenicity, arranged in descending order of average node weight. Monomers which occur at least 20 times in the entire data set have been included in the table. The node weights have been obtained using AttentiveFP–IG combination. (E) Attribution analysis using AttentiveFP model architecture and integrated gradients attribution method for a representative glycan shows that xylose (Xyl) contributes the most to immunogenicity. In xylose, the substructure centered on the anomeric carbon, C1, is shown to be the key contributor. A similar substructure analysis has been shown for fucose. In the glycan graph, the size of the node corresponds to the importance, and the color corresponds to the monomer. In the fingerprint, only the indices with positive contribution to immunogenicity have been visualized, with the color corresponding to the normalized importance.

attribution analysis at the level of chemical substructures, which is more intuitive than weighted importance of atoms in a large macromolecule.

We obtained receiver operating characteristic-area under curve (ROC-AUC) greater than 0.95 on the held-out test data set, for all glycan immunogenicity and taxonomy classification tasks (SI tables 3 and 4). Against results reported in the literature, our models outperformed metrics for classification for four out of eight tasks and achieved comparable results for the remaining four (SI table 5) [62]. We noted that for most tasks the performance of the one-hot-featurized graphs were comparable to the ECFP-featurized graphs.

2.5. Attribution analysis finds features key to the model's decision-making

Graph attribution methods attempt to crack open the black-box supervised GNNs and allow to infer specific features—subgraphs, monomers and chemical moieties—and their impact on the predicted property. The critical features revealed through graph attribution help elucidate the fundamental structure-function relationships that underpin otherwise opaque chemical/biological properties like immunogenicity, postulate very explicit hypotheses that can be validated in the lab, and may help in further design of immunogenic or non-immunogenic scaffolds.

To avoid spurious hypotheses [63] that may occur in NN attribution on molecular models, we chose the optimal combination of GNN architecture and attribution method, following the implementation invariance [64] axiom which proposes that attributions using the same method with different model implementations should be identical. In other words, all implementations of the same model type should attribute similar features for the same predicted property. In an ideal scenario, all attributions over different implementations of a model should be equal, or have a standard deviation of zero.

We evaluated three attribution methods, integrated gradients [64], input × gradients [65] and attention weights. To quantify implementation invariance, we calculated the standard deviation of the node attribution weights across all immunogenic glycans using different combinations of model architecture and attribution method. For attribution of immunogenicity in glycans, we noted that AttentiveFP-integrated gradients had the smallest deviation between implementations and is thus the best choice (figure 4(C)). All further attribution analysis was done using node weights obtained from AttentiveFP-integrated gradients.

Across all immunogenic glycans present in the dataset, N-glycolylneuraminic acid, xylose, and fucose were found to be the key monomers responsible for immunogenicity, consistent with experimental findings and in line with their low prevalence in human glycans (figure 4(D)) [66].

For a single immunogenic glycan, we observed that xylose, followed by galectin, were the monomers that contributed most significantly to immunogenicity (figure 4(E)). Attribution identifies, in addition to the importance of individual nodes, the critical substructures in the monomers that contribute most to immunogenicity, such as the substructure centered on the anomeric carbon of xylose. To assess the sensitivity [64] of attribution analysis, we performed ablation analysis, where we muted individual xylose monomers and then all xylose monomers (SI figure 45). When features for a single xylose monomer were muted, we noted that attributions of other xylose monomers remained unchanged. Similarly, when all xylose monomers were muted, galectin was the key monomer responsible for immunogenicity.

3. Conclusion

This work provides a generalized method for representing macromolecules as hierarchical graphs with molecular fingerprints to capture chemical information which can be used to compute structural similarity between macromolecules with different composition and topology, and perform unsupervised and supervised learning. The unsupervised learning enables visualization of the complex landscape of different classes of macromolecules and understanding of the subtle differences between similar macromolecules. The attribution analysis helps in cracking open the black-box of supervised GNN models, which in turn can help elucidate fundamental design principles of otherwise opaque structure-property relationships and assist with hypothesis generation for future experimental studies. We therefore expect that this toolkit will be used by both experimentalists and computational practitioners in chemistry, biology and materials science for a variety of macromolecule property prediction tasks. Because accurate property prediction is key for design, application or these models could drive design of improved macromolecules by combining directed evolution, Monte Carlo tree search, or similar optimization methods that seek to maximize scores predicted through supervised models.

4. Methods

4.1. Dataset download and pre-processing

4.1.1. Glycans

A dataset of 19 299 glycans was downloaded from GlycoBase (accessed on 2 November 2020) [23]. The file contained GlycoBase ID, sequence, link (N, O, free, or none), species, and immunogenicity information for each glycan. For each glycan sequence string the brackets denote branches, with the point of attachment/bonding of the branch as the monomer immediately after the brackets. The 1st element within the bracket is the monomer most distant from the point of attachment, and the last element within the bracket is the abbreviation of the bond that connects the branch to the original main chain. Nested brackets indicate additional sub-branches off of branches, and multiple sets of brackets next to each other indicate several branches off of the same monomer.

Seven modifications and 152 glycan sequences were found to be erroneous, i.e. they are invalid data, owing to unequal number of opening and closing brackets and dangling branches without specified connectivity, and were thus deleted. Additional glycan sequences were removed due to missing SMILES sequences for a number of monomers. The curated dataset resulted in a total of 19 147 glycans.

4.1.2. Anti-microbial peptides (AMPs)

A dataset of 15 864 AMPs, including 15 450 monomers, 200 multimers, and 214 multi-peptides, from the database of antimicrobial activity and structure of peptides (DBAASP) was downloaded (accessed on 6 October 2020) [67]. Each peptide was represented in the dataset as an individual JavaScript Object Notation (JSON) file containing information about the peptide ID, name, sequence(s), unusual amino acids, connectivity, terminal modifications, complexity, synthesis type, target groups and objects, and target species. The term ‘target species’ is used loosely and actually encompasses both species and sub-classification information such as subspecies, strain for bacteria, and forma speciales for fungi. For each target species for any given peptide, the dataset provides the AMP concentration in units of mostly either M or g ml^{-1} as a function of four unique variables: activity measure, salt type, medium, and CFU. The dataset includes three different types of peptide complexities: monomers, multimers, and multi-peptides. In the DBAASP dataset, monomers consist of a single sequence, multimers between two and four separate sequences connected via interchain bonds, and multi-peptides between two and four separate sequences connected not via covalent bonds but instead weaker intermolecular forces.

The information from each JSON file was combined into a single table and the AMP concentrations for each target species was converted into a numerical value by removing symbols like '>' and '<', taking the average whenever the dataset provides a range, and disregarding uncertainty values. Nine peptide monomer types included in a total of 86 peptides were removed due to ambiguity of the molecular structure, bringing the total number of peptides to 15 778. This condensed dataset was further processed to visualize the distribution of target species data points. For each 'target species', the species name was separated from any sub-classifications (subspecies, strain, forma speciales, serovar, pathovar, biovar, etc).

4.2. Text file to machine-readable graph conversion

A custom parser was developed to convert the macromolecules from the SMILES-MONOMERS-BONDS text files to machine-readable NetworkX graphs with monomers expressed as nodes and bonds expressed as edges. The parser goes through the .txt file line by line, stores the monomer information in a dictionary with keys as integer positions and values as monomer abbreviations, and stores the bond information in a dictionary with keys as tuples containing bond connectivities and values as bond abbreviations. Afterwards, the reader uses NetworkX to add each key in the monomer dictionary as a node and each key in the bond dictionary as an edge, storing the abbreviations as attributes for the corresponding node or edge. The resulting NetworkX graphs include both linear and highly branched architectures. Before using the parser, all glycans and peptides obtained from the various datasets were converted to the standardized text file format.

4.3. Macromolecule graph representation and featurization

The macromolecule was represented as an attributed graph, $G(V, E)$, where V stands for monomers/nodes, and E stands for bonds/edges. Stereochemical extended connectivity fingerprints, generated using RDKit, were used to featurize the monomers and bonds [68, 69]. Radius and number of bits were optimized by calculating mean and standard deviation, and visualizing the distribution of Tanimoto similarity [70] of all monomers in the glycans dataset (SI section 3.1).

4.4. Similarity computation

4.4.1. Exact GEDs

GED is a measure of similarity between two graphs, computed as the cost of transforming one graph into another by basic edit operations, such as insertion, deletion and substitution of nodes and edges [54]. This method was used to calculate dissimilarity which is more intuitive when compared to a baseline case where the edit distance for the macromolecule with itself is zero, and with anything else is greater than zero. GED has three operations—insertion, deletion and substitution—for both nodes and edges. We performed a grid search over combinations of possible node/edge substitution costs and multipliers for node/edge insertion/deletion costs to find an optimal set of values. For node/edge substitution, the Tanimoto distance between the stereochemical fingerprints, a value in the range of 0 and 1, is multiplied by the substitution cost to obtain the edit distance. For node/edge insertion/deletion, a constant value which is the node/edge insertion/deletion cost is added to the edit distance.

The insertion/deletion and substitution costs can be tuned to accurately depict the differences across topology and monomer chemistry of the macromolecules in the dataset. Using a higher insertion/deletion cost than substitution cost would penalize changes in topology more than those in monomer chemistry, and vice-versa. When both insertion/deletion and substitution costs are same, the changes to both topology and monomer chemistry are treated equally. The magnitude of the costs helps in tuning the range of edit distances. We noted that using both insertion/deletion and substitution costs as three provided an optimal intuition for our dataset, accounting for both the changes in chemistry and topology (SI table 1).

4.4.2. Graph kernel

Graph kernel is a kernel function to compute inner product on graph-structured data [71]. The kernel approach results in a similarity matrix for a set of graphs, analogous to GED for a pair of graphs. In this work, we used propagation attribute kernel implemented in GraKeL [55, 56] to compute the $(n \times n)$ similarity matrix. All glycans with labels on at least one taxonomic level were considered for the similarity computation. Each pair of graph similarity was computed for a maximum of 100 iterations. This resulted in 5% of the pairs being assigned a zero similarity (10% of all indices in the similarity matrix are zero). To benchmark against GED, we performed a grid search over hyperparameters of propagation attribute kernel–bin width {1, 3, 10, 100}, and the preserved distance metric on local sensitive hashing {'L1-norm', 'L2-norm'}.

4.5. Unsupervised learning

4.5.1. Uniform manifold approximation and projection (UMAP)

UMAP is a method for dimensionality reduction based on manifold learning and topological data analysis, capturing both local and global structure of the data [59]. This method was used for dimensionality reduction of similarity matrices. Number of neighbors, {2, 4, 8, 16, 32, 64, 128, 256}, was optimized for two component UMAP dimensionality reduction of similarity vectors [59]. By visual inspection, UMAP with 128 neighbors was observed to resolve into optimal size and number of clusters. The subplot showed distinct regions for the immunogenic and non-immunogenic glycans. In addition to two component UMAP, we constructed UMAP for increasing number of components, and analyzed the segmentation at higher components.

4.5.2. t-distributed stochastic neighbor embeddings (t-SNEs)

t-SNE applies a non-linear dimensionality reduction technique that calculates the pairwise similarities between points and minimizes the difference between the similarities in higher and lower dimensions [58]. We benchmarked the dimensionality reduction results obtained from UMAP against a broad range of t-SNE models. For the different models, we varied perplexity as {2, 5, 30, 50, 100}, and number of steps as {500, 1000, 5000}. From the scatter plot, colored by immunogenicity labels, we noted that dimensionality reduction using t-SNE was not able to deduce the differences and cluster the glycans into distinct areas.

4.5.3. Principal component analysis (PCA)

PCA is a linear dimensionality reduction technique, using singular value decomposition to transform the high-dimensional data to low dimensional embeddings [57]. We used two-component PCA, with default hyperparameters, to benchmark the dimensionality reduction of immunogenic glycans.

4.6. Supervised learning

4.6.1. Tasks

For classification, we used the glycans dataset, and classified immunogenicity and eight taxonomy levels (domain, kingdom, phylum, class, order, genus, and species). For regression, we trained over minimum inhibitory concentration for AMPs against *Escherichia coli* and *Staphylococcus aereus*.

4.6.2. Model architectures

Five different model architectures—graph convolutional networks (GCN) [72], Weave [73], message passing neural networks (MPNNs) [74], graph attention networks (GAT) [75], and AttentiveFP [76], as implemented in deep graph library (DGL) LifeSci library [77], were used for classification and regression. GCN uses convolutional aggregation operations over node features in the graph. The Weave model architecture is an extension of GCNs to learn over molecular graphs, thus convolving over both atom/node and bond/edge features. MPNN updates the node features by summing over the node and edge features in the node neighborhood. GAT utilizes self-attention layers to implicitly focus on key node features, unlike GCNs that give equal weight to all node features. AttentiveFP learns both the local neighborhood by propagation of information at the nodes and the non-local information via GAT mechanism. While Weave, MPNN, and AttentiveFP utilize both node and edge features, GCN and GAT only consider node features.

4.6.3. Adapting NetworkX graphs to be trained using DGL

NetworkX graphs were converted into undirected, unweighted, and homogenous DGL graphs [77]. For GCN and GAT model architectures, self-loops were added to the DGL graphs to prevent silent performance regression due to zero-in-degree nodes during training.

4.6.4. Optimization of model

For classification, the optimization was done by minimization of average cross-entropy loss between batches and additional metrics such as F1 score, recall, precision and accuracy were noted. For regression, the optimization was done by minimization of root-mean-squared-error loss on the validation dataset, and additional metrics such as R^2, Pearson's correlation, Spearman's correlation, and mean absolute error were noted. Hyperparameter optimization was carried out for 1000 iterations using SigOpt [78].

4.7. Attribution analysis

4.7.1. Graph attribution methods

Integrated gradients [64] and input × gradients [65] attribution methods were used over weave, AttentiveFP and MPNN, for attribution analysis. Additionally, node attention weights were analyzed for AttentiveFP. The model architecture selection was done to have one of each type of architecture—weave (graph convolution), AttentiveFP (graph attention), and MPNN (message passing).

Integrated gradients interpolate between the input graph and a baseline graph, where all features are zero, and accumulate the gradient values for each node equation (1). The notation follows [50].

$$G_{\mathrm{A}} = (G - G') \int\limits_{\alpha=0}^{1} \frac{\mathrm{d}y\left(G' + \alpha\left(G - G'\right)\right)}{\mathrm{d}G} \mathrm{d}\alpha. \tag{1}$$

Input × gradients attribution is the element-wise product of the input graph and the gradient.

$$G_{\mathrm{A}} = \left(\frac{\mathrm{d}\hat{y}}{\mathrm{d}G}\right)^{\mathrm{T}} G. \tag{2}$$

For attention weights, the node attention weights were obtained by averaging over the attention scores of the adjacent nodes.

For each attribution method, we obtained the node weights by multiplying the positive weights with the input fingerprint vectors:

$$n = \sum_{\mathrm{nodes}} G_{\mathrm{A}}^{+} G. \tag{3}$$

The node weights were normalized to the maximum node weight to obtain the normalized weights.

$$n_{\mathrm{norm}} = \frac{n}{\max\left(n\right)}. \tag{4}$$

4.7.2. Visualization of key substructures

To visualize the responsible substructures in the monomers, we used G_{A} in equation (1), and multiplied the weights of the with the respective monomer fingerprint. This approach resulted in a weights vector with the same size as the fingerprint, with the most positively to the most negatively influencing substructure for the prediction. Using RDKit, we visualized the chemical substructures at different fingerprint indices and mapped it to the weights.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: 10.5281/zenodo.5237237.

Acknowledgments

S M acknowledges funding from MIT-Takeda Fellowship, Novo Nordisk and MIT's Jameel Clinic, J A acknowledges funding from MIT's UROP office. The authors thank Dr Daria Kim and Sarah Antilla from the Department of Chemistry, MIT for valuable comments.

Conflict of interest

R G B is an advisor at Relay Therapeutics and Alphabet Sandbox.

Author contribution

S M developed and trained the computational tools with contributions from J A. R G B supervised the work. All authors contributed to the writing of the manuscript.

Code availability

All the code used for model training and analysis is available at https://github.com/learningmatter-mit/GLAMOUR, and archived in Zenodo repository [79].

ORCID iD

Somesh Mohapatra https://orcid.org/0000-0001-9498-3834

References

[1] Wyman J and Gill S J 1990 *Binding and Linkage: Functional Chemistry of Biological Macromolecules* (Mill Valley, CA: University Science Books)

[2] Rosales A M, Segalman R A and Zuckermann R N 2013 Polypeptoids: a model system to study the effect of monomer sequence on polymer properties and self-assembly *Soft Matter* **9** 8400–14

[3] Lutz J-F, Lehn J-M, Meijer E W and Matyjaszewski K 2016 From precision polymers to complex materials and systems *Nat. Rev. Mater.* **1** 1–14

[4] Lutz J-F, Ouchi M, Liu D R and Sawamoto M 2013 Sequence-controlled polymers *Science* **341** 1238149

[5] Romio M, Trachsel L, Morgese G, Ramakrishna S N, Spencer N D and Benetti E M 2020 Topological polymer chemistry enters materials science: expanding the applicability of cyclic polymers *ACS Macro Lett.* **9** 1024–33

[6] Crosby A J and Lee J 2007 Polymer nanocomposites: the "nano" effect on mechanical properties *Polym. Rev.* **47** 217–29

[7] Boydston A J, Cui J, Lee C U, Lynde B E and Schilling C A 2020 100th anniversary of macromolecular science viewpoint: integrating chemistry and engineering to enable additive manufacturing with high-performance polymers *ACS Macro Lett.* **9** 1119–29

[8] Cichosz S, Masek A and Zaborski M 2018 Polymer-based sensors: a review *Polym. Test.* **67** 342–8

[9] Thompson C B and Korley L T J 2020 100th anniversary of macromolecular science viewpoint: engineering supramolecular materials for responsive applications—design and functionality *ACS Macro Lett.* **9** 1198–216

[10] Sun H and Zhong Z 2020 100th anniversary of macromolecular science viewpoint: biological stimuli-sensitive polymer prodrugs and nanoparticles for tumor-specific drug delivery *ACS Macro Lett.* **9** 1292–302

[11] Lutz J F 2015 Coding macromolecules: inputting information in polymers using monomer-based alphabets *Macromolecules* **48** 4759–67

[12] Hiemenz P C and Lodge T P 2007 *Polymer Chemistry* (Boca Raton, FL: CRC Press)

[13] Cho C Y, Moran E J, Stephans J C, Fodor S P, Adams C L, Sundaram A, Sundaram A, Jacobs J W and Schultz P G 1993 An unnatural biopolymer *Science* **261** 1303–5

[14] Soth M J and Nowick J S 1997 Unnatural oligomers and unnatural oligomer libraries *Curr. Opin. Chem. Biol.* **1** 120–9

[15] Cromm P M, Spiegel J and Grossmann T N 2015 Hydrocarbon stapled peptides as modulators of biological function *ACS Chem. Biol.* **10** 1362–75

[16] Gaynor S G, Edelman S and Matyjaszewski K 1996 Synthesis of branched and hyperbranched polystyrenes *Macromolecules* **29** 1079–81

[17] Gao H and Matyjaszewski K 2006 Synthesis of star polymers by a combination of ATRP and the "click" coupling method *Macromolecules* **39** 4960–5

[18] Johnson J A, Lu Y Y, Burts A O, Lim Y-H, Finn M G, Koberstein J T, Turro N J, Tirrell D A and Grubbs R H 2011 Core-clickable PEG-branch-azide bivalent-bottle-brush polymers by ROMP: grafting-through and clicking-to *J. Am. Chem. Soc.* **133** 559–66

[19] Zhang T, Li H, Xi H, Stanton R V and Rotstein S H 2012 HELM: A Hierarchical Notation Language for Complex Biomolecule Structure Representation *J. Chem. Inf. Model.* **52** 2796–806

[20] Heller S R, McNaught A, Pletnev I, Stein S and Tchekhovskoi D 2015 InChI, the IUPAC international chemical identifier *J. Cheminform.* **7** 23

[21] Drefahl A 2011 CurlySMILES: a chemical language to customize and annotate encodings of molecular and nanodevice structures *J. Cheminform.* **3** 1–7

[22] Lin T-S *et al* 2019 BigSMILES: a structurally-based line notation for describing macromolecules *ACS Cent. Sci.* **5** 1523–31

[23] Bojar D, Powers R K, Camacho D M and Collins J J 2021 Deep-learning resources for studying glycan-mediated host-microbe interactions *Cell Host Microbe* **29** 132–144.e3

[24] Altschul S F, Gish W, Miller W, Myers E W and Lipman D J 1990 Basic local alignment search tool *J. Mol. Biol.* **215** 403–10

[25] Altschul S F *et al* 1997 Gapped BLAST and PSI-BLAST: a new generation of protein database search programs *Nucleic Acids Res.* **25** 3389–402

[26] Boratyn G M, Thierry-Mieg J, Thierry-Mieg D, Busby B and Madden T L 2019 Magic-BLAST, an accurate RNA-seq aligner for long and short reads *BMC Bioinform.* **20** 1–19

[27] Smith T F and Waterman M S 1981 Identification of common molecular subsequences *J. Mol. Biol.* **147** 195–7

[28] Needleman S B and Wunsch C D 1970 A general method applicable to the search for similarities in the amino acid sequence of two proteins *J. Mol. Biol.* **48** 443–53

[29] Eddy S R 2004 Where did the BLOSUM62 alignment score matrix come from? *Nat. Biotechnol.* **22** 1035

[30] Li L, Ching W K, Yamaguchi T and Aoki-Kinoshita K F 2010 A weighted q-gram method for glycan structure classification *BMC Bioinform.* **11** 1–6

[31] Aoki K F, Yamaguchi A, Okuno Y, Akutsu T, Ueda N, Kanehisa M and Mamitsuka H 2003 Efficient tree-matching methods for accurate carbohydrate database queries *Genome Inform.* **14** 134–43

[32] Hosoda M, Akune Y and Aoki-Kinoshita K F 2017 Development and application of an algorithm to compute weighted multiple glycan alignments *Bioinformatics* **33** 1317–23

[33] Coff L, Chan J, Ramsland P A and Guy A J 2020 Identifying glycan motifs using a novel subtree mining approach *BMC Bioinform.* **21** 42

[34] Yamanishi Y, Bach F and Vert J P 2007 Glycan classification with tree kernels *Bioinformatics* **23** 1211–6

[35] Schissel C K *et al* 2021 Deep learning to design nuclear-targeting abiotic miniproteins *Nat. Chem.* **13** 1–9

[36] Alley E C, Khimulya G, Biswas S, AlQuraishi M and Church G M 2019 Unified rational protein engineering with sequence-based deep representation learning *Nat. Methods* **16** 1315–22

[37] Bileschi M L *et al* 2019 Using deep learning to annotate the protein universe *bioRxiv* pp 1–29

[38] Elnaggar A, Heinzinger M, Dallago C and Rihawi G 2020 ProtTrans: towards cracking the language of life ' s code through self-supervised deep learning and high performance computing (arXiv:200706225)

[39] Kim C, Chandrasekaran A, Huan T D, Das D and Ramprasad R 2018 Polymer genome: a data-powered polymer informatics platform for property predictions *J. Phys. Chem.* C **122** 17575–85

[40] Chen L, Pilania G, Batra R, Huan T D, Kim C, Kuenneth C and Ramprasad R 2021 Polymer informatics: current status and critical next steps *Mater. Sci. Eng.* R **144** 100595

[41] St John P C, Phillips C, Kemper T W, Wilson A N, Guan Y, Crowley M F, Nimlos M R and Larsen R E 2019 Message-passing neural networks for high-throughput polymer screening *J. Chem. Phys.* **150** 234111

[42] Qiao B *et al* 2020 Quantitative mapping of molecular substituents to macroscopic properties enables predictive design of oligoethylene glycol-based lithium electrolytes *ACS Cent. Sci.* **6** 1115–28

[43] Lee C-K, Lu C, Yu Y, Sun Q, Hsieh C-Y, Zhang S, Liu Q and Shi L 2021 Transfer learning with graph neural networks for optoelectronic properties of conjugated oligomers *J. Chem. Phys.* **154** 024906

[44] Hamilton W L, Ying R and Leskovec J 2017 Representation learning on graphs: methods and applications (arxiv:1709.05584)

[45] Yang K *et al* 2019 Analyzing learned molecular representations for property prediction *J. Chem. Inf. Model.* **59** 3370–88

[46] Jin W, Barzilay R and Jaakkola T 2020 Hierarchical generation of molecular graphs using structural motifs (arXiv:2002.03230 [cs.LG])

[47] Schütt K T, Arbabzadah F, Chmiela S, Müller K R and Tkatchenko A 2017 Quantum-chemical insights from deep tensor neural networks *Nat. Commun.* **9** 6–13

[48] Schütt K T, Sauceda H E, Kindermans P-J-J, Tkatchenko A and Müller K-R-R 2018 SchNet—a deep learning architecture for molecules and materials *J. Chem. Phys.* **148** 241722

[49] Unke O T and Meuwly M 2019 PhysNet: a neural network for predicting energies, forces, dipole moments, and partial charges *J. Chem. Theory Comput.* **15** 3678–93

[50] Sanchez-Lengeling B *et al* 2020 Evaluating attribution for graph neural networks *Adv. Neural Inf. Process. Syst.* **33** 5898–910

[51] Sanchez-Lengeling B, Wei J N, Lee B K, Gerkin R C, Aspuru-Guzik A and Wiltschko A B Machine learning for scent: learning generalizable perceptual representations of small molecules (arXiv:1910.10685 [stat.ML])

[52] Berman H M 2000 The protein data bank *Nucleic Acids Res.* **28** 235–42

[53] Hagberg A A, Schult D A and Swart P J 2008 Exploring network structure, dynamics, and function using networkx *Proc. 7th Python in Science Conf. (Pasadena, CA)* ed G Varoquaux, T Vaught and J Millman pp 11–15

[54] Abu-Aisheh Z, Raveaux R, Ramel J Y and Martineau P 2015 An exact graph edit distance algorithm for solving pattern recognition problems *4th Int. Conf. on Pattern Recognition Applications and Methods 2015 (Lisbon)* (available at: https://hal.archives-ouvertes.fr/hal-01168816)

[55] Neumann M, Garnett R, Bauckhage C and Kersting K 2016 Propagation kernels: efficient graph kernels from propagated information *Mach. Learn.* **102** 209–45

[56] Siglidis G, Nikolentzos G, Limnios S, Giatsidis C and Vazirgiannis M 2020 GraKeL: a graph kernel library in Python *J. Mach. Learn. Res.* **21** 1–5

[57] Tipping M E and Bishop C M 1999 Mixtures of probabilistic principal component analyzers *Neural Comput.* **11** 443–82

[58] van der Maaten L and Hinton G 2008 Visualizing data using t-SNE *J. Mach. Learn. Res.* **9** 2579–605

[59] McInnes L, Healy J, Saul N and Großberger L 2018 UMAP: uniform manifold approximation and projection *J. Open Source Softw.* **3** 861

[60] Borg I and Groenen P J F 2005 *Modern Multidimensional Scaling: Theory and Applications* (New York: Springer)

[61] McInnes L, Healy J and Astels S 2017 hdbscan: hierarchical density based clustering *J. Open Source Softw.* **2** 205

[62] Burkholz R, Quackenbush J and Bojar D 2021 Using graph convolutional neural networks to learn a representation for glycans *Cell Rep.* **35** 109251

[63] McCloskey K, Taly A, Monti F, Brenner M P and Colwell L J 2019 Using attribution to decode binding mechanism in neural network models for chemistry *Proc. Natl Acad. Sci. USA* **116** 11624–9

[64] Sundararajan M, Taly A and Yan Q 2017 Axiomatic attribution for deep networks *34th Int. Conf. on Machine Learning ICML (4 March)* vol 7 (available at: https://arxiv.org/abs/1703.01365v2) (Accessed 30 December 2021) pp 5109–18

[65] Shrikumar A, Greenside P and Kundaje A 2017 Learning important features through propagating activation differences *34th Int. Conf. on Machine Learning ICML* vol 7 pp 4844–66

[66] Planinc A, Bones J, Dejaegher B, van Antwerpen P and Delporte C 2016 Glycan characterization of biopharmaceuticals: updates and perspectives *Anal. Chim. Acta* **921** 13–27

[67] Pirtskhalava M *et al* 2016 DBAASP v. 2: an enhanced database of structure and antimicrobial/cytotoxic activity of natural and synthetic peptides *Nucleic Acids Res.* **44** D1104–12

[68] Landrum G 2006 RDKit: open-source cheminformatics (available at: www.rdkit.org)

[69] Rogers D and Hahn M 2010 Extended-connectivity fingerprints *J. Chem. Inf. Model.* **50** 742–54

[70] Rogers D J, Tanimoto T T and Computer A 1960 Program for classfying plants *Science* **132** 1115–8

[71] Borgwardt K, Ghisu E, Llinares-López F, Bray L O and Rieck B 2020 Graph kernels (arXiv:2011.03854v2 [cs.LG])

[72] Kipf T N and Welling M 2019 Semi-supervised classification with graph convolutional networks *5th Int. Conf. on Learning Representations ICLR 2017—Conf. Track Proc.* pp 1–14

[73] Kearnes S, McCloskey K, Berndl M, Pande V and Riley P F 2016 Molecular graph convolutions: moving beyond fingerprints *J. Comput. Aided Mol. Des.* **30** 595–608

[74] Gilmer J, Schoenholz S S, Riley P F, Vinyals O and Dahl G E 2017 Neural message passing for quantum chemistry (arXiv:1704.01212)

[75] Velickovic P, Cucurull G, Casanova A, Romero A, Lio P and Bengio Y 2017 Graph attention networks (arXiv:1710.10903 [stat.ML]) pp 1–12

[76] Xiong Z *et al* 2020 Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism *J. Med. Chem.* **63** 8749–60

[77] Wang M *et al* 2019 Deep graph library: a graph-centric, highly-performant package for graph neural networks (arXiv:1909.01315) pp 1–18

[78] Clark S and Hayes P 2019 SigOpt webpage (available at: https://sigopt.com) (Accessed 20 September 2006)

[79] Mohapatra S 2021 Learningmatter-mit/GLAMOUR: v0.1 (available at: https://zenodo.org/record/5237237) (Accessed 23 August 2021)