Memetic Computing (2024) 16:519–536
https://doi.org/10.1007/s12293-024-00414-6

REVIEW

Molecular representations in bio-cheminformatics

Thanh-Hoang Nguyen-Vo[1,4] · Paul Teesdale-Spittle[2] · Joanne E. Harvey[3] · Binh P. Nguyen[4]

Received: 20 May 2024 / Accepted: 17 June 2024 / Published online: 20 July 2024

Abstract
Molecular representations have essential roles in bio-cheminformatics as they facilitate the growth of machine learning applications in numerous sub-domains of biology and chemistry, especially drug discovery. These representations transform the structural and chemical information of molecules into machine-readable formats that can be efficiently processed by computer programs. In this paper, we present a comprehensive review, providing readers with diverse perspectives on the strengths and weaknesses of well-known molecular representations, along with their respective categories and implementation sources. Moreover, we provide a summary of the applicability of these representations in de novo molecular design, molecular property prediction, and chemical reactions. Besides, representations for macromolecules are discussed with highlighted pros and cons. By addressing these aspects, we aim to offer a valuable resource on the significant role of molecular representations in advancing bio-cheminformatics and its related domains.

Keywords Molecular representation · Molecular property prediction · Molecular fingerprint · Language models · Graph neural networks · Drug discovery

1 Molecular representations for machine learning

Molecular representations and features play an essential role in machine learning applications in the domains of chemistry, drug discovery, and materials science. These representations convert the structural and chemical information of molecules into a format that can be efficiently processed by computational models. In recent years, several reviews on these representations have been published [1–3] to give readers different perspectives on the pros and cons of known representations as well as how they are categorized. Despite their valuable information, these works need to be updated with recent advances. Building partially on previous reviews and incorporating updated information and a holistic understanding of these representations, we conducted a comprehensive review of molecular representations for machine learning in bio-cheminformatics. Within the scope of this study, we focus on those that are commonly used in de novo molecular design and Quantitative Structure–Activity Relationship (QSAR) modeling. In this section, we introduce various molecular representations that are classified into six groups: *string-based*, *property-based*, *molecular fingerprints*, *language model-based*, *graph-based*, and *others* based on their characteristics.

1.1 String-based representations

String-based representations include all types that describe molecular bonds and structures using special symbols (e.g., '//', '@'), alphabet letters (e.g., 'H', 'C'), or any other non-

✉ Binh P. Nguyen
binh.p.nguyen@vuw.ac.nz

Thanh-Hoang Nguyen-Vo
hoang.nguyen@weltec.ac.nz

Paul Teesdale-Spittle
paul.teesdale-spittle@vuw.ac.nz
https://people.wgtn.ac.nz/paul.teesdale-spittle

Joanne E. Harvey
joanne.harvey@vuw.ac.nz

1 School of Innovation, Design, and Technology, Wellington Institute of Technology, 21 Kensington Avenue, Lower Hutt 5012, New Zealand

2 School of Biological Sciences, Victoria University of Wellington, Kelburn Parade, Wellington 6012, New Zealand

3 School of Chemical and Physical Sciences, Victoria University of Wellington, Kelburn Parade, Wellington 6012, New Zealand

4 School of Mathematics and Statistics, Victoria University of Wellington, Kelburn Parade, Wellington 6012, New Zealand

numeric forms. One of the most widely used string-based molecular representations is the Simplified Molecular-input Line-Entry System (SMILES) [4] (Fig. 1). The SMILES representation of a molecule is a compact textual notation that encodes its molecular structure, where atoms are represented by chemical symbols (e.g., '*S*' for '*Sulfur*', '*O*' for' *Oxygen*') and bonds are represented by special symbols (e.g., '-' for a *single bond*, '=' for a *double bond*, and ':' for an *aromatic bond*). SMILES has found extensive applications in cheminformatics and drug discovery due to its simplicity and ease of use. The SMILES notation follows a set of predefined rules and syntax, facilitating the conversion between molecular structures and textual representations. It enables the storage, retrieval, and manipulation of molecular information in databases and machine learning workflows. However, the direct input of SMILES representations into machine learning models is not ideal without being transformed into corresponding numeric forms (e.g., one-hot encoding) [5]. It has also been observed that SMILES syntax is redundant, as multiple SMILES strings can represent the same compound. Besides SMILES, there are several other string-based representations, such as the International Chemical Identifier (InChI) [6], InChI Key [6], and SYBYL Line Notation [7]. SELFIES (Self-Referencing Embedded Strings), introduced by Krenn et al. [8], is a more advanced, unique, and concise string-based representation developed with rules for molecular reconstruction.

Most string-based representations are supported by the RDKit library [9], open-source software for cheminformatics. SMARTS and SMIRKS are two specific representations used for structural pattern searching and chemical reaction description, respectively. While SMARTS is supported by RDKit, SMIRKS, a hybrid representation based on SMILES and SMARTS, can be generated using Ambit-SMIRKS [10]. SLN is versatile and used for expressing chemical structures, conducting searches, and describing chemical reactions in 3D chemical structures, whereas SMILES is specifically designed for representing 2D structures. Ambit-SLN [11] facilitates the processing of SLN conversion. Table 1 summarizes tools and software that support translation from one string-based representation to another.

1.2 Property-based representations

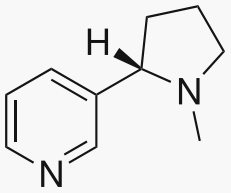
Property-based representations of molecules are numerical vectors or matrices that carry information on theoretically-derived molecular properties and characteristics (Fig. 2). Molecular descriptors, which are typical property-based representations, can be either continuous or categorical values computed based on the 2D or 3D structures of molecules [12]. These descriptors provide a quantitative representation of molecular structures, which can be directly used in machine learning tasks, exploratory data analysis, or structural similarity assessment.

Molecular descriptors cover a wide range of physicochemical properties, including topological, geometrical, electrostatic, and quantum-chemical properties. Many molecular descriptor sets (e.g., Chemopy, CDK, etc.) are defined by different groups of properties. Numerous non-commercial libraries [9, 13, 14], software [15], and web servers [16] support the computation of these descriptors. In addition to molecular descriptors, electrostatically computed matrices, such as the Coulomb matrix, the Ewald sum matrix, and the Sine matrix, also serve as property-based representations [14]. However, these matrices are expressed in a similar pattern to the adjacency matrix of the molecular graph. Table 2 summarizes tools and software that support property-based representations.

1.3 Molecular fingerprints

Molecular fingerprints, also known as *chemical fingerprints* or simply *fingerprints*, are numerical expressions that indicate the presence or absence of specific substructures. The fingerprint vector contains information about substructural patterns within a molecule (Fig. 3). The diversity of fragmentation methods for substructural hashing creates a variety of fingerprints. While most fingerprint vectors are binary, others are substructure-count vectors. A binary fingerprint reader scans the molecular structure and, upon detecting a substructure, counts it as '*one (1)*', ignoring any subsequent occurrences of the same substructure. In contrast, substructure-count fingerprint readers count all occurrences of repetitive substructures, highlighting differences in the frequency of substructures. Fingerprint vectors are useful for similarity searches [20] and various machine learning tasks, except for de novo molecular design. The high computational cost of reconstructing a molecule from its fingerprint is the primary barrier to the applications of fingerprints in this field. Furthermore, these reconstruction methods often lack precision. E-State and Extended-Connectivity are typical examples of binary fingerprinting tools found in CDK and PubChem [15]. Klekota-Roth, AtomPairs2D, and Substructure fingerprints support both substructure-count and binary forms [15]. Since a fingerprint vector is simply a binary vector of annotated substructures, users can customize their fingerprints by defining which substructures should be detected. NC-MFP [21] is an example of a fingerprint customized for natural compounds. Similar to molecular descriptors, fingerprint vectors can be easily computed using different non-commercial libraries [9, 13], software [15], and web servers [16].

Table 3 provides information on tools and software that support typical molecular fingerprints. ChemDes [16] can currently be used to compute 59 commonly used types of fin-



IUPAC Name: 3-[(2S)-1-methylpyrrolidin-2-yl]pyridine

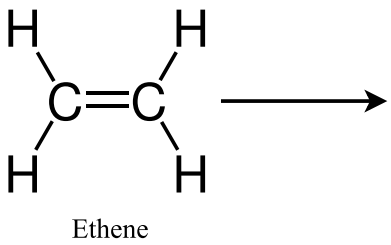
InChI Key: SNICXCGAKADSCV-JTQLQIEISA-N

SMILES: CN1CCC[C@H]1C1=CN=CC=C1

Fig. 1 An example of string-based representations (Compound: *Nicotine*)

Table 1 Tools and software that support string-based representations

String type	Short term	Tool/software
Simplified Molecular-input Line-Entry System	SMILES	RDKit [9]
International Chemical Identifier	InChI	RDKit [9]
International Chemical Identifier Key	InChI Key	RDKit [9]
SMILES arbitrary target specification	SMARTS	RDKit [9]
Hybrid of SMILES and SMARTS	SMIRKS	Ambit-SMIRKS [10]
SYBYL line notation	SLN	Ambit-SLN [11]



	H	H	C	C	H	H
H	0.5	0.3	2.9	1.5	0.2	0.2
H	0.3	0.5	2.9	1.5	0.2	0.2
C	2.9	2.9	36.9	14.3	1.5	1.5
C	1.5	1.5	14.3	36.9	2.9	2.9
H	0.2	0.2	1.5	2.9	0.5	0.3
H	0.2	0.2	1.5	2.9	0.3	0.5

Coulomb matrix

Fig. 2 An example of property-based representations. The Ethene (C_2H_4)'s Coulomb matrix is constructed from N^2 (where $N = 6$) Coulomb potential values ($M_{i,j}^{Coulomb}$) which are computed based on the pairwise inter-atomic distance between any pair of constitutional atoms as follows: $M_{i,j}^{Coulomb} = \begin{cases} 0.52 \times Z_i^{2.4} \; \forall \, i = j \\ \frac{Z_i \times Z_j}{|R_i - R_j|} \; \forall \, i \neq j \end{cases}$, where Z and $|R_i - R_j|$ are the atomic number and the Euclidean distance between atoms i and j, respectively

Table 2 Tools and software that support property-based representations

Descriptor set	Number of features	Tool/software
RDKit descriptors	196	RDkit [9], ChemDes [16]
ChemoPy descriptors	1135	ChemoPy [17], ChemDes [16]
CDK descriptors	275	CDK [18], ChemDes [16]
Pybel descriptors	24	Pybel [19], ChemDes [16]
BlueDesc descriptors	174	ChemDes [16]
PaDEL descriptors	1875	PaDEL [15], ChemDes [16], Mordred [13]
Coulomb matrix	n/a	DScribe [14]
Ewald sum matrix	n/a	DScribe [14]
Sine matrix	n/a	DScribe [14]

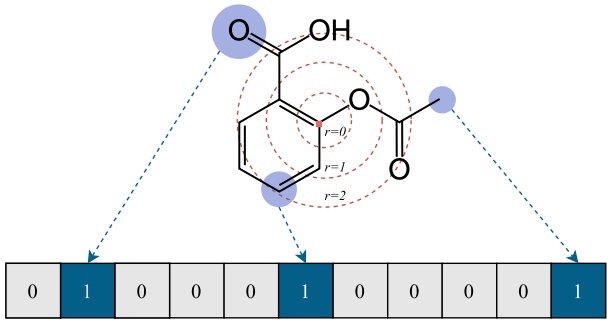


Fig. 3 An example of structure-based representations. Aspirin's Morgan fingerprints are a binary vector in which '*one (1)*' and '*zero (0)*' indicate the '*presence*' and '*absence*' of a defined substructure, respectively. A set of Morgan substructures is determined by the number of selected bits (e.g., 1024, 2048) and radius (r). The size of the substructure is associated with the radius

gerprints. Among these, MACCS (166 bits), PubChem (881 bits), Morgan (1024 $\times$ n bits), and Substructure (307 bits) are frequently used for molecular featurization. The Klekota-Roth fingerprint (4860 bits), introduced by Klekota and Roth [22], creates high-dimensional sparse vectors, whereas the E-State fingerprint (79 bits) generates low-dimensional vectors. The number of bits and the radius (r) can be adjusted for the Extended-Connectivity Fingerprint (ECFP) [23], a more generalized and adaptable version of the Morgan fingerprint. In the Morgan fingerprint, the radius indicates the size of circular substructures; for example, a radius of 2 indicates that each substructure is composed of two atoms. Varying the number of bits and the radius results in different fingerprints. ECFP is frequently followed by a number indicating the chosen diameter (twice the radius), such as ECFP2, ECFP4, and ECFP6, which correspond to radius of 1, 2, and 3 atoms away, respectively. The performance of a downstream task is often influenced by the selected number of bits. It is important to choose a sufficiently large number of bits to cover the most essential substructures in the chemical set, but an excessively large number can result in sparse vectors that slow down computation. Although there is no strict rule for selecting the number of bits, researchers commonly set it to be a multiple of 512 (e.g., 1024, 2048). Additionally, ECFP is used to create language model-based representations, such as Mol2vec [21] and NPBERT [24]. An ECFP fingerprint can also be converted to an indexing vector and then transformed into an embedding matrix for a molecular property prediction task [25, 26]. The Natural Compound-Molecular Fingerprint (NC-MFP) [27], a fingerprint customized for natural compounds, is not readily available as a module. Developing and reimplementing NC-MFP (10,016 bits) is challenging due to the numerous unconnected processing stages and software required. Menke et al. [27] trained a deep neural network to encode the Natural Product Fingerprint (NPFP) vectors, demonstrating that NC-MFP was less effective compared to NPFP in downstream tasks.

1.4 Language model-based representations

Language model-based representations are continuous vectors or matrices created by '*molecular encoders*' (Fig. 4). Molecular encoders are pre-trained models developed using a large set of molecules. During training, these molecular encoders learn the structural patterns and characteristics of molecules to map them to corresponding continuous forms, which are expected to be convertible back to their original structures. Examples of molecular encoders include Mol2vec [28], ChemBERTa [29], and NPBERT [24]. Most molecular encoders are developed using language models, where each molecule (defined by a specific set of substructures) is treated as a '*sentence*', and its substructures are treated as '*words*'. A '*valid molecule*' is analogous to a '*meaningful sentence*', emphasizing the importance of the order of substructures. The molecular encoders learn the '*grammar of molecules*' to create a vector space capable of effectively encoding any inputted molecule. The inputs for molecular encoders can include index vectors, one-hot vectors, graph-based matrices, or any other form readable by the model. The quality of the vector space depends on the volume of training data, the architecture used, and the training strategies. These language model-based representations are then used as inputs for downstream machine learning tasks. The use of continuous representations enables more efficient optimization through gradient descent and other brute-force methods [3].

Table 4 presents tools and software that support language model-based representations. Mol2vec [28], a pre-trained model, was the first molecular encoder to convert molecules into corresponding language model-based features. It draws inspiration from Word2vec [30], a method for word embedding. These encoders are trained using language models and vast sources of data. For developing Mol2vec, Jaeger et al. [28] used nearly 20 million chemical structures as training samples, initially translated into ECFP vectors with 2048 bits and radii of 0 and 1. The Mol2vec encoder was trained with two approaches: Continuous Bag-of-Words (CBOW) and Skip-gram, resulting in two embedding sizes for molecules: 100- and 300-dimensional continuous vectors. Motivated by Mol2vec and aided by advanced deep learning architectures, various molecular encoders have been constructed for specific purposes. Examples include SMILES-BERT [31], MolBERT [32], ChemBERTa [29, 33], NPBERT [24], and FP-BERT [34], all developed using the Bidirectional Encoder Representations from Transformers (BERT) architecture [35]. Currently, BERT is one of the most robust Transformer architectures, employing self-supervised learning methods. SMILES-BERT, MolBERT, and ChemBERTa are designed to learn the syntax of SMILES for encoding

Table 3 Tools and software that support molecular fingerprints

Fingerprint	No. of bits	Tool/software
CDK	1024	CDK [18], PaDEL [15]
CDK Extended	1024	CDK [18], PaDEL [15]
CDK Graph-only	1024	CDK [18], PaDEL [15]
E-State	79	RDkit [9], PaDEL [15]
MACCS	166	PaDEL [15], ChemDes [16]
PubChem	881	PaDEL [15], ChemDes [16]
Substructure	307	PaDEL [15], ChemDes [16]
Substructure-count	307	PaDEL [15], ChemDes [16]
Klekota-Roth	4860	PaDEL [15], ChemDes [16]
Klekota-Roth-count	4860	PaDEL [15], ChemDes [16]
Morgan	flexible	RDkit [9], ChemDes [16]
NC-MFP	10,016	Seo et al. [21]
NPFP	n/a	Menke et al. [27]

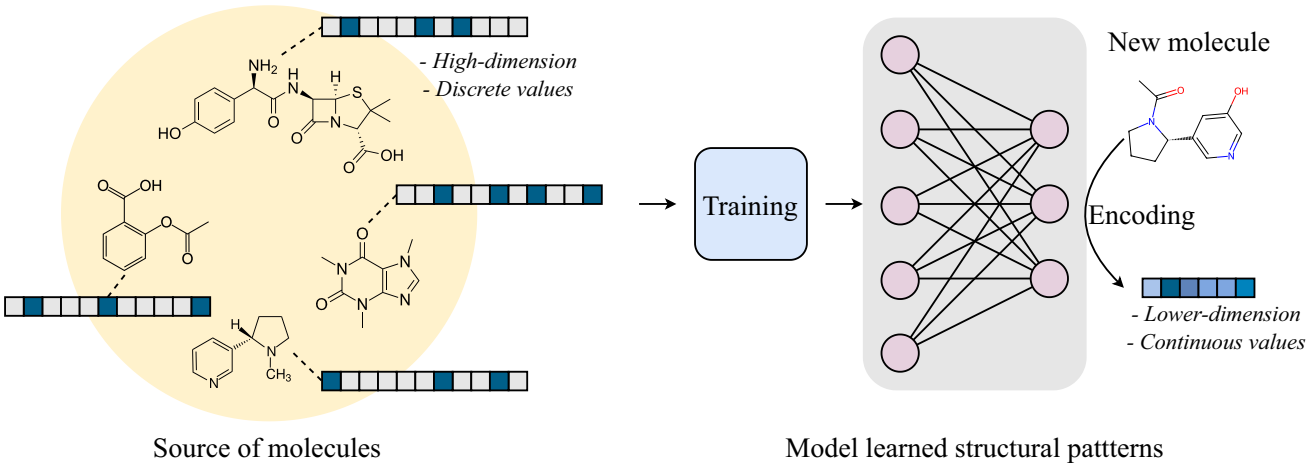


Fig. 4 An example of language model-based representation

input SMILES strings of molecules. The training datasets for SMILES-BERT, MolBERT, and ChemBERTa contained approximately 18 million, 1.6 million, and 77 million molecular structures, respectively. NPBERT and FP-BERT are trained to learn the ECFP fingerprints of the substructures according to their appearance orders in the molecule. The NPBERT training dataset was enriched with 250k structures of natural products and about 1.9 million ordinary chemical data points. FP-BERT was trained with roughly 2.0 million compounds.

Besides using SMILES, ChemBERTa [29, 33] has another version trained with SELFIES. Similarly, SELFormer [36] was designed to create representations from SELFIES using RoBERTa [37], a robustly optimized BERT variant. ChemFormer [38] was constructed using the Bidirectional and Auto-Regressive Transformer (BART) [39] architecture. Contrary to BERT-based models, BART-based models prioritize the correction of sequences that have been altered with random tokens instead of using masked language modeling in their pre-training phase. MoLFormer [40] was developed using the RoFormer [41] architecture, an enhanced Transformer version with rotary position embedding. X-MOL [42], a large-scale molecular encoder, was trained using a Transformer architecture with 12 pairs of Encoder-Decoder. MolMap [43] learned 1,456 molecular descriptors and 16,204-bit fingerprints from about 8.5 million molecules using a dual-path convolutional neural network [44] to create 3D fingerprint maps of size $37 \times 36 \times 3$.

Language model-based representations can also be generated from graph-based encoders. The Hierarchical Molecular Graph Self-supervised Learning (HiMol) encoder [45] uses three levels of molecular graph information: node, motif, and graph. Initially, the input molecular graph (atom node-level) is fragmented into motifs to create motif-level nodes before adding a graph-level node. These three levels of a molecular graph's features are learned by an encoder to create three corresponding representation levels. FunQG [46] is a molecular encoder trained with Quotient Graphs of Functional groups.

Instead of using traditional molecular graphs constructed by a network of nodes, Hajiabolhassan et al. [46] considered each functional group as a specific node, resulting in more informative graphs. However, their representation learning is most useful for encoding heavy molecules with complex structures, as small molecules typically consist of a limited number of functional groups.

1.5 Graph-based representations

Graph-based representations provide graphical expressions of the structural connectivity of molecules. In these representations, atoms are considered '*nodes*' or '*vertices*', and '*intramolecular bonds*' are considered '*edges*' (Fig. 5). Thus, a molecule can be viewed as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, defined by a set of nodes (atoms) $\mathcal{V}$ and a set of edges (bonds) $\mathcal{E}$, where $\mathcal{E} \subseteq v_i, v_j \mid v_i, v_j \in \mathcal{V}\ and\ v_i \neq v_j$. Employing molecular graphs helps to extract valuable information on molecular connectivity, such as substructures, symmetry, and functional groups, to predict possible molecular properties (e.g., toxicity, solubility) or to explain the origins of these properties (e.g., alert structures). To be processed by machine learning models, a molecule is transformed into a *node matrix*', which is an adjacency matrix indicating connections among all atoms within the molecule. In addition to the node matrix, novel graph-based neural networks utilize additional matrices indicating node or edge attributes to enhance learning efficiency. For molecular graphs, the '*edge attribute matrix*' provides information on the types of bonds between atom pairs, while the '*node attribute matrix*' includes additional molecular characteristics (e.g., element, orbital hybridization, charge status). To facilitate the learning process with graph-based representations, a number of graph-based deep learning architectures have been developed and continue to evolve, fully exploiting the potential of molecular graphs [47]. Numerous graph-based representations have been derived from molecular graphs, such as graph-embedding features [48–50]. The graph-based representation shown in Fig. 5 is just one of many possible graphs. The node order in the adjacency matrix can change depending on the graph traversal algorithm used. A single molecule can have multiple graph representations tailored for specific tasks. Some examples can be found in [51, 52].

While graphs are inherently 2D data structures with no spatial relationships between elements, they can effectively encode 3D information and stereochemical details by incorporating such data into the node and edge features. Graph representations have significant advantages over linear notations due to their ability to naturally encode 3D information and the interpretability of all molecular subgraphs. However, there are also some disadvantages to using molecular graph representations for certain applications. Molecular graphs are inadequate for representing certain types of molecules, particularly those with delocalized bonds, polycentric bonds, ionic bonds, or metal-metal bonds. Organometallic compounds, for instance, cannot be effectively described by molecular graphs due to their complex bonding schemes. Hypergraphs offer a solution for handling multi-valent bonds by representing edges as sets of atoms, but their use is not widespread. Additionally, for molecules with constantly changing 3D structures, a single static graph representation is not meaningful and could hinder problem-solving. A significant challenge with graph-based representations is their lack of compactness, both in memory usage and size. Representing a molecular graph requires complex data structures that are harder to search than compact linear representations. As a graph grows larger, its memory requirements increase significantly. In contrast, linear notations provide more compact and memory-efficient molecular representations, making them easier to use for identity searches, though less effective for substructure searches.

Table 5 summarizes tools and software that support graph representations. Initially, DeepChem [53] was launched as a community project focusing on the applications of deep learning in chemistry and drug discovery. Over the years, the project has expanded to encompass a broader range of applications in molecular science. It now provides an open-source Python library with useful modules for processing multiple molecular representations, including graphs. DGL-LifeSci, developed by Li et al. [54], is another open-source Python library that supports deep learning on graphs in life sciences. Surge, created by McKay et al. [55], is a quick command-line tool for generating molecular graphs from SMILES. However, it struggles to process complex aromatic structures.

1.6 Other representations

In addition to the five types of molecular representations mentioned earlier, other formats, such as '*3D voxelized*' and '*image-based*' representations (Fig. 6), can also be employed for machine learning tasks [56]. However, their applications are somewhat restricted due to unaddressed limitations. The 3D voxelized representation creates 3D arrays that often exhibit high sparsity and dimensionality, but this method lacks invariant information regarding molecular rotation, translation, and permutation [57–59]. Conversely, image-based representations typically convert most small molecules into 2D images. Drawing on the success of Google's Inception-ResNet [60]. Goh et al. developed Chemception [61], a specialized approach for molecular embedding. Building on the Chemception concept, Bjerrum et al. [62] introduced another molecular encoder capable of creating five-band molecular images, offering more comprehensive information for downstream machine learning tasks. Table 6 lists the tools and software used in computing these alternative representations.

Table 4 Tools and software that support language model-based representations

Encoder	Source
Mol2vec [28]	github.com/samoturk/mol2vec
SMILES-BERT [31]	github.com/uta-smile/SMILES-BERT
MolBERT [32]	github.com/BenevolentAI/MolBERT
ChemBERTa [29]	github.com/seyonechithrananda/bert-loves-chemistry
ChemBERTa-2 [29, 33]	github.com/seyonechithrananda/bert-loves-chemistry
NPBERT [24]	github.com/mldlproject/2021-NPBERT-Antimalaria
FP-BERT [34]	github.com/fanganpai/fp2bert
SELFormer [36]	github.com/HUBioDataLab/SELFormer
Chemformer [38]	github.com/MolecularAI/Chemformer
MoLFormer [40]	github.com/IBM/molformer
X-MOL [42]	github.com/bm2-lab/x-mol
MolMap [43]	github.com/shenwanxiang/bidd-molmap
HiMol [45]	github.com/ZangXuan/HiMol
FunQG [46]	github.com/hhaji/funqg

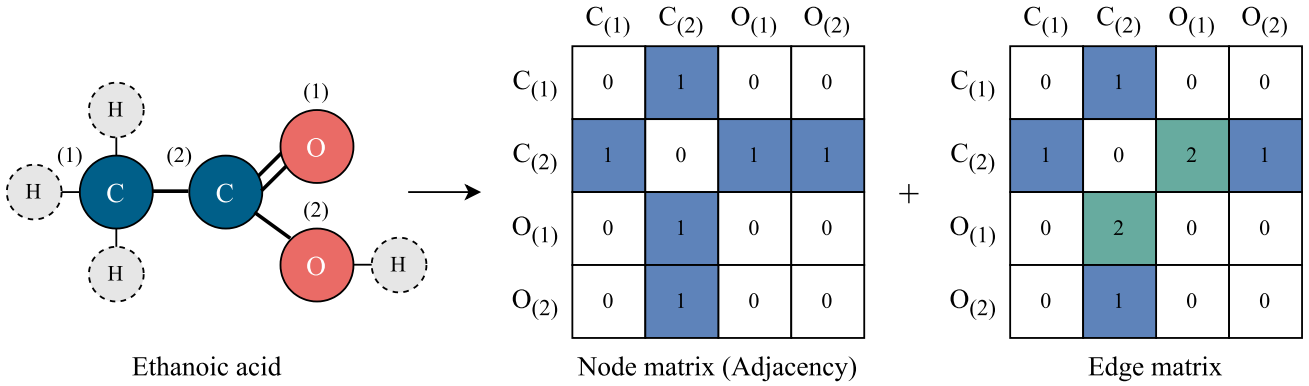


Fig. 5 An example of the graph representation. The node and edge matrices of ethanoic acid (CH_3COOH) are generated based on the connectivities (bonds) among atoms and their bond types (e.g., single, double, triple). Only heavy atoms (excluding hydrogen) are considered when creating these node and edge matrices

Table 5 Tools and software that support graph-based representations

Tool/software	Source
DeepChem [53]	https://deepchem.io
DGL-LifeSci [54]	github.com/awslabs/dgl-lifesci
Surge [55]	https://structuregenerator.github.io

1.7 De novo molecular design and property prediction

The variety of molecular representations provides researchers with numerous options for creating new computational frameworks. No study has conclusively shown that one representation is consistently superior, as model performance depends on many factors, including data volume, learning strategies, and the characteristics of the molecules. Molecular descriptors and fingerprints might be more appropriate for small datasets because they can be quickly computed and are compatible with traditional machine learning models. However, using these representations often requires feature engineering and selection. Additionally, they are restricted to property prediction tasks because they are uninvertible. In contrast, string-based representations are primarily used for de novo molecular design due to their invertibility, while graph-based representations are well-suited for handling large datasets with deep learning models, removing the need for feature engineering. Language model-based representations are particularly effective for exploratory data analysis of molecular structures and property prediction tasks. Their continuous nature allows for more efficient optimization of the learning process compared to other types, such as one-hot matrices and binary vectors. Additionally, as learnable representations, language model-based representations can be customized to distinguish between different classes of

Table 6 Tools and software that support other representations

Tool/software	Source
gnina [63]	github.com/gnina
DeepChem [53]	https://deepchem.io
OctSurf [64]	github.uconn.edu/mldrugdiscovery/OctSurf
LiGAN [65]	github.com/mattragoza/liGAN
RDKit [9]	https://www.rdkit.org

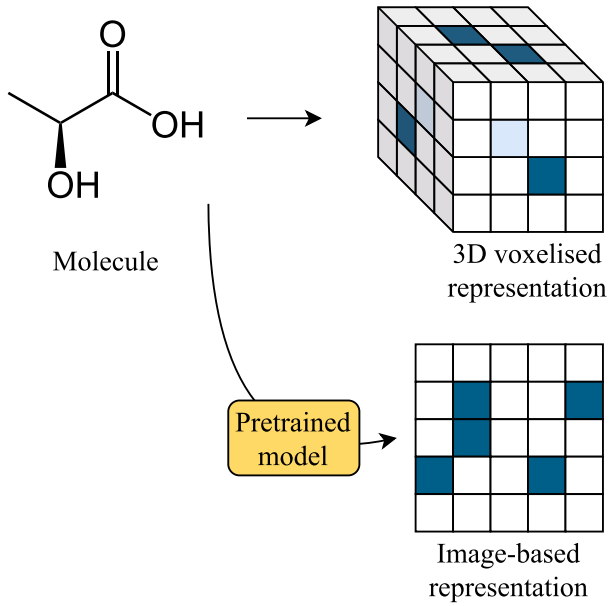


Fig. 6 An example of the other representations

molecules, potentially improving the model's performance. Table 7 summarizes molecular representations and their applicable tasks.

2 Representations for structural preservation

Representations for structural preservation are responsible for holding information about the atoms, bonds, connectivity, and coordinates of a molecule. They contain header information, atom information, bond connections, and types, followed by sections for more complex information.

2.1 Connection table

While graphs are fundamental for molecular representation, their connectivity matrices are not compact and scale quadratically with the number of atoms. The connection table (Ctab) provides a more structured format, comprising six parts: *Counts line*, *Atom block*, *Bond block*, *Atom list block*, *Structural text descriptor block*, and *Properties block*. The Counts line offers an overview of the structure by specifying the number of atoms, bonds, atom lists, and chirality presence, along with the version (V2000 or V3000). The Atom block lists atom identities, atomic symbols, mass differences, charges, stereochemistry, and associated hydrogens, often treating hydrogens implicitly to reduce size. The Bond block details atom connectivity and bond types, including bond order. These core blocks form the basis of the Ctab, which is extensible to include additional properties. Connection tables have become standard for handling chemical structural information due to their backward compatibility and widespread use, particularly in Molfile formats. Notably, connection tables are not file formats themselves but serve as the foundational structure for chemical table files (CTfiles).

2.2 The Molfile format

The Molfile (or CTfile) family utilizes connection tables to represent molecular structures. These formats were first developed by MDL Information Systems (MDL), later acquired by Symyx Technologies, and are now known as BIOVIA [66]. The CTfile format is released in an open format that requires users to register to download the specifications. CTfiles are highly extensible, leading to the creation of a series of widely adopted file formats for transferring chemical information. The connection table (Ctab) is encapsulated within the Molfile format, which can be further integrated into a structure-data (SD) file, including both structural information and additional property data for multiple molecules. Similarly, the Reaction file (RXNfile) [67] describes individual reactions, while the Reaction-Data (RDfile) [66] stores either reactions or molecules along with their associated data. The Reaction Query file (RGfile) [68] is designed for handling queries, and the Extended Data file (XDfile) [67], which is XML-based, facilitates the transfer of structures or reactions along with their metadata. Further information on these file types and their structures is available in MDL documentation and cheminformatics textbooks. Although Molfiles themselves contain rich structural information, they are not directly suitable for training machine learning models in their raw forms. Therefore, they need to be pre-processed and converted into a machine-readable format (e.g., molecular fingerprints, descriptors). Figure 7 visualizes the key features of these CTfiles.

Table 7 Eligible tasks for different molecular representations

Representation	Invertibility		Task	
	Yes	No	Property prediction	De novo molecular design
String-based	✓		✓	✓
Property-based		✓	✓	
Structure-based		✓	✓	
Language model-based	✓		✓	✓
Graph-based	✓		✓	
3D voxelised	✓		✓	✓
Image-based		✓	✓	

3 Representations for chemical reactions

Chemical reactions, which involve the transformation of one set of molecules into another under specific conditions, have been extensively documented, with around 127 million reactions recorded to date [69]. Recently, there has been renewed interest in developing models to predict reaction outcomes, synthetic routes, and analyze reaction networks [70]. While traditional graphical representations of reactions are common, they are not easily machine-readable. Thus, various machine-readable reaction data exchange formats (e.g., RXNfiles, RDfiles) have been developed. These formats are essential for applications in computer-aided synthesis design and autonomous discovery, accommodating the complexities and limitations of different molecular representations.

3.1 SMILES Reaction Kinetics Scheme

SMILES, which describes ordinary text-based molecular structures, has been extended to include the SMILES Reaction Kinetics Scheme (SMIRKS), a notation developed by Daylight Chemical Information Systems for describing generic chemical reaction transformations. SMIRKS extends both SMILES and SMARTS. While SMILES is used to represent specific molecules and SMARTS to define molecular patterns or substructures, SMIRKS is specifically designed to encode reaction transformations, identifying which atoms and bonds change during a reaction.

In Reaction SMILES, reactants, agents, and products are represented as SMILES strings, separated by '>' or '≫'. Atom mappings, which connect reactants to products, are included, but additional information like reaction centers or conditions is not supported. Other formats, such as RXNfiles and RDfiles, can store this additional metadata. SMIRKS describes generic reaction transformations by specifying reaction centers and changes in bonds and atoms. It combines features of SMILES and SMARTS, requiring specific rules for application, such as the correspondence of mapped atoms and explicit hydrogens in reactants and products. SMIRKS are then converted into reaction graphs for further use.

3.2 Reaction InChI

Reaction InChI (RInChI) [71, 72], developed between 2008 and 2018, provides a unique, order-invariant identifier for chemical reactions to aid reproducibility and consistency in reaction representation. Unlike Reaction SMILES, RInChI uses InChIs for individual molecules and tracks *structureless* entities when InChIs cannot be generated. RInChI includes information about equilibrium, unbalanced, or multi-step reactions, and employs a layering system to describe distinct aspects of the reaction, such as solvents, catalysts, and reaction direction. This makes it particularly useful for identifying practically identical reactions conducted under specific conditions. An extension, ProcAuxInfo [73], allows for the storage of metadata like yields and reaction conditions. While RInChI can identify duplicate reactions and efficiently indexing and searching reaction data, it lacks equivalents to SMARTS or SMIRKS, limiting its use for substructure searches and encoding generic transformations.

As a standardized textual identifier for chemical reactions, RInChI facilitates the sharing and indexing of chemical reaction information by encoding the reactants, products, and, optionally, the agents involved. The RInChI system is designed to provide a unique and machine-readable representation of chemical reactions, making it easier to search for, retrieve, and exchange reaction data across different databases and platforms. It includes details about the reaction participants and can also capture information about the reaction conditions, ensuring consistency and interoperability in cheminformatics and related fields.

3.3 Other representations

Varnek et al. developed the Condensed Graph of Reactions (CGR) [74] to encode molecular structures in a matrix, identifying fragment occurrences and highlighting changes in atoms and bonds between reactants and products. This method was inspired by Fujita's concept of imaginary transition states. CGRtools [75] was developed to support CGR.

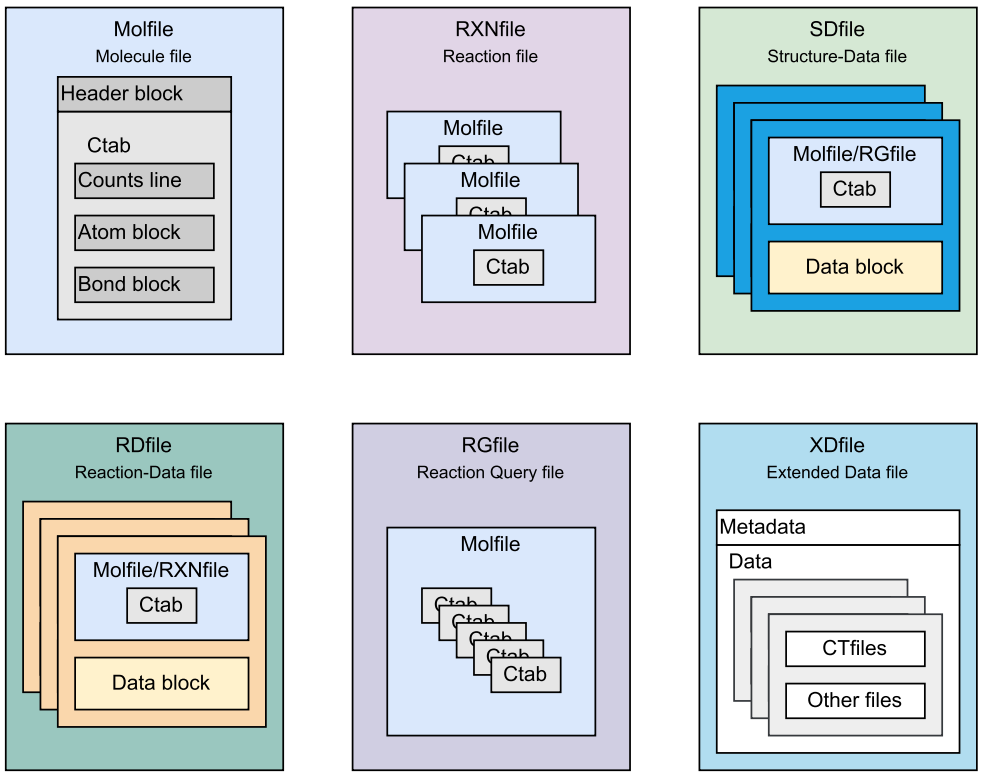


Fig. 7 The MDL family of CTfiles are created based on the connection tables (Ctab). The connection table is specified by atom and bond blocks that describe the atoms and their corresponding connectivity. Molfiles and RXNfiles are used to describe single molecules and reactions, respectively. SDfiles and RDfiles store a series of structures or reactions and associated data. RGfiles are used to handle reaction queries. XDfiles are used for transferring structure or reaction data using the XML format

The Bond-Electron (BE) matrix [76], proposed by Dugundji and Ugi, represents reactions in a matrix format. It has been employed by the EROS software [77] and the WODCA system [78] for reaction classification. The BE-matrix is an $N \times N$ where N is the number of atoms in a molecule; diagonal entries denote free valence electrons, while off-diagonal entries indicate bond orders. Reactions are represented by an "*R-matrix*" that records bond changes, with positive values for bond formation and negative values for bond breakage. Adding the R-matrix to the reactant's BE-matrix yields the product's BE-matrix, providing an alternative way to represent reaction centers and illustrating the integration of detailed information into matrix representations.

Hierarchical Organization of Reactions through Attribute and Condition Education (HORACE) [79] utilizes a machine learning algorithm to classify chemical reactions, notable for its hierarchical reaction description. It captures both specific reaction instances and abstract reaction types using three abstraction levels. At the base level, it describes the partial order of atom types, establishing a hierarchy based on atom similarity. The next level characterizes molecules using functional groups, linking them to the reaction center. The top level specifies physicochemical properties, which describe the functional aspects of the corresponding structures. This hierarchical model provides a more comprehensive depiction of chemical reactions than purely structural approaches like SMILES.

Saller et al. introduced the InfoChem CLASSIFY algorithm [80], a method for reaction representation that has significantly influenced the development of rule-based synthesis planning methods [81, 82]. This approach identifies the reaction center by detecting atoms that change their implicit hydrogens, valency, π-electrons, atomic charges, or have bonds made or broken, mapping equivalent atoms in reactants and products. However, determining the reaction center is a key challenge [83–85]. To address this issue, the maximum common substructure (MCS) between reactants and products is first identified. Once found, hash codes for atoms in the reaction center are calculated using a modified Morgan algorithm [86], incorporating a wide range of properties such as atom type, valence, hydrogen count, π-electrons, aromaticity, and formal charges. These hash codes are then summed across reactants and one product to yield a unique reaction center representation. This description can be extended to include adjacent atoms for varying specificity: the reaction center alone provides a broad description, adding alpha atoms

gives a medium description, and including further adjacent atoms results in a narrower, more specific description. These hash codes facilitate reaction classification and are used in later synthetic planning tools.

The concept of reaction fingerprints involves using binary vectors to capture the structural changes occurring in the reaction center. This method constructs fingerprints (e.g., ECFP variant [23]) and computes the difference between product and reactant vectors, optionally including agents. Patel et al. first discussed reaction vectors [87], which were later utilized in de novo design and classification approaches [88]. Schneider et al. employed difference fingerprints with the atom-pair variant to develop a prediction framework for classifying 50 reaction types [89]. While reaction fingerprints offer an alternative method to traditional reaction center detection and representation, they struggle with convertibility into reaction graphs, and handling stereochemistry remains an ongoing research topic [90]. Colley et al. developed RDChiral [90], an RDKit-based wrapper for managing stereochemistry in retrosynthetic template extraction and future approaches.

4 Representations for macromolecules

4.1 Peptides and proteins

Peptides and proteins are both constructed from amino acids (AAs). A single AA is characterized by an amine (-NH_2) group, a carboxyl (-COOH) group, and a distinct side chain. AAs are typically denoted by either a one-letter symbol or a three-letter abbreviation [91]. Although the Latin alphabet is sufficient to represent the 20 AAs in the genetic code, more symbols are required to represent the large number of naturally occurring AAs.

Peptides are biological sequences of 2 to 50 amino acids (AAs) that connect to each other via peptide bonds. These sequences get involved in diverse biological activities, ranging from antibiotics to biological modulators. In 1994, Siani et al. developed the CHUCKLES method [92] to create SMILES for polymers based on their sequences and vice versa, facilitating Forward Translation (FT) in cheminformatics. The CHUCKLES method uses a lookup table that maps monomer sequences to their corresponding SMILES, with atoms involved in monomer bonds removed. This approach is suitable for oligomeric peptides and is integrated into BIOPEP-UWM [93]. CHORTLES [94], an upgraded version of CHUCKLES, was then created to deal with oligomeric mixtures.

Hierarchical Editing Language for Macromolecules (HELM) [95, 96] and the Self-Contained Sequence Representation (SCSR) [97] are two prominent notations for describing a high variety of macromolecules. While HELM utilizes SMILES, SCSR uses the v3000 Molfiles. Conversion between these two types can be done by BIOVIA's toolkit. Pfizer developed HELM under the auspices of the Pistoia Alliance to represent macromolecules composed of diverse structures (e.g., peptides, antibodies). Initially, HELM could only process molecules with well-defined structures, but the introduction of HELM2 expanded its capabilities to handle polymer mixtures and free-form annotations. HELM uses streamlined CHUCKLES and graphs to represent monomers in simple polymers and complex polymers, respectively. Its structure hierarchy reflects the granularity of the components: complex polymer, simple polymer, monomer, and atom. HELM is widely used by numerous pharmaceutical companies, public databases (e.g., ChEMBL), software (e.g., ChemDraw, ChemAxon), and toolkits (e.g., RDKit, Biomolecule Toolkit) [98].

4.2 Glycans

Glycans, or carbohydrates, refer to polymers such as oligosaccharides and polysaccharides that are built of multiple monosaccharides (monomers). These macromolecules play crucial roles in most biological processes, including cell-cell communication, immune response, and protein stabilization. In drug discovery, glycans are of particular interest for their potential as receptors, small-molecule glycomimetics, therapeutic glycopeptides, and vaccines. Oligosaccharides and polysaccharides are polymers that are composed of more than 3 and 20 monomers, respectively.

Glycan databases are essential for carbohydrate research, typically using monosaccharide-based notations to record structures [99–102]. However, these notations are inadequate for analyzing glycan-protein interactions, which require atom-based representations. To address this, several tools have been developed to translate monosaccharide-based notations into atom-based formats. The Web3 Unique Representation of Carbohydrate Structures (WURCS) [103] was created to provide a linear, unique notation compatible with the semantic web, integrating bioinformatics and cheminformatics features. The latest version of WURCS [104], used by GlyTouCan [105], the International Glycan Structure Repository, encodes the main carbon backbone of monosaccharide residues, backbone modifications, and linkage information, while also handling unspecified structures. Despite its widespread adoption in databases, WURCS remains unsupported by most cheminformatics software. Besides, other independent representations have been proposed to tackle specific issues [106].

4.3 Polymeric drugs

Polymers are used to deliver drug molecules. However, several polymers with therapeutic activities and are used as

bioactive agents in treatments, known as polymeric drugs. The BigSMILES [107] syntax was recently created to encode diverse polymer structures, including homopolymers, random and block co-polymers, and complex connectivity types (e.g., linear, ring, and branched). The stochastic units of these polymers are marked by curly brackets, with repeated units separated by commas within the brackets. Since BigSMILES notation has not yet supported canonicalization, several canonicalization methods have been proposed to eliminate multiplicity [108]. There are currently no practical applications for this notation, but its prospective applications in drug discovery modeling are promising.

5 Discussion

5.1 Representations for machine learning

Property-based representations are continuous or discrete numeric features computed by software or libraries. These features, such as molecular descriptors, can be used for various molecular prediction tasks, including solubility, bioactivity, and toxicity prediction. When using these features with distance-based algorithms (e.g., k-Nearest Neighbors) or linear algorithms (e.g., Logistic Regression, Support Vector Machines), data normalization is often required to ensure that all features contribute equally to the model. This normalization step helps improve the performance and convergence of these algorithms. In contrast, when tree-based algorithms (e.g., Random Forest, Extremely Randomized Trees, Gradient Boosting Machines) are employed, data normalization can be omitted. Tree-based methods inherently handle features with different scales and are robust to varying feature distributions. This makes them particularly advantageous for dealing with heterogeneous datasets where feature scaling might be challenging or unnecessary. Furthermore, property-based representations can be combined with ensemble methods to enhance prediction accuracy and robustness. By leveraging multiple algorithms, ensemble methods can capture a broader range of patterns and relationships within the data, leading to improved model performance. These representations can also be integrated with feature selection techniques to identify the most informative features, reducing dimensionality and potentially enhancing computational efficiency and interpretability.

Unlike property-based representations, most molecular fingerprints are binary features with lengths that vary depending on the type used. Since these features are binary, data scaling is not necessary. However, distance-based machine learning algorithms are generally unsuitable for molecular fingerprints due to the lack of robust distance metrics for binary vectors. This limitation is especially applicable when handling unbalanced datasets. For example, the Synthetic Minority Over-sampling Technique (SMOTE) [109] is not suitable for molecular fingerprints because it relies on computing distances and using interpolation, which are not suitable to binary data. Tree-based algorithms are more appropriate for molecular fingerprints because they can effectively manage binary features. Those algorithms are capable of capturing complex relationships and interactions within the binary features without the need for distance metrics. Some advanced tree-based algorithms (e.g., eXtreme Gradient Boosting [110], LightGBM [111], and CatBoost [112]) are proficient in the management of unbalanced datasets and can integrate feature importance metrics to identify the most pertinent binary features. Ensemble learning techniques can also be used to improve the performance of tree-based algorithms [113]. Furthermore, the representation of molecular fingerprints can be optimized by integrating tree-based methods with feature selection and extraction techniques, thereby reducing dimensionality and enhancing computational efficiency.

Language model-based representations are continuous numeric features generated by molecular encoders, which are pre-trained neural networks that map the substructural information of molecules into vectors of continuous values, known as molecular embeddings. Since each molecule is represented by a fixed-length vector, data scaling is unnecessary. For a given encoder, these molecular embeddings are generated based on a learnable distribution. As medium-dimensional continuous vectors or matrices, these embeddings are highly compatible with distance-based, linear, and tree-based models. Additionally, molecular decoders can reconstruct the corresponding molecular structures from the embeddings. Depending on their configuration, molecular decoders may translate the embeddings into either identical or slightly different structures. This reconstructability is crucial for de novo molecular design. Combining molecular generative models with one or more pre-defined networks for property prediction results in property-directed molecule generation systems. These systems generate molecules with desired properties through a multi-objective optimization process. Essentially, the molecular encoder learns a distribution, forming a chemical vector space. The embeddings created within this space can then be transformed into valid molecules with the desired properties.

5.2 Representations for chemical reactions

The SMIRKS, RInChI, and other representations for chemical reactions each have strengths and weaknesses. SMIRKS, an extension of the SMILES notation, excels in simplicity and can encode complex reaction rules in a text-based format, making it accessible for computational applications. However, its simplicity can be a drawback when dealing with intricate reactions or stereochemistry. On the other

hand, RInChI, a reaction-specific version of the InChI system, offers a more standardized and detailed representation, capturing precise information about reactants, products, and conditions. This standardization aids in data sharing and interoperability but can be cumbersome to generate and interpret due to its complexity. Additionally, other representations, like reaction graphs, provide a visual and intuitive depiction of chemical reactions, highlighting connectivity and transformations between molecules. While these are beneficial for education and initial analysis, they may lack the depth and precision needed for advanced computational modeling. Ultimately, the choice of representation depends on the specific needs of the task, balancing ease of use, detail, and computational efficiency.

5.3 Representations for macromolecules

Representations for macromolecules offer advantages over purely atomic-based notations in developing modified drug peptides. Replacing natural L-amino acids (L-AA) with D-amino acids (D-AA), for instance, can enhance a peptide's oral bioavailability. HELM simplifies these modifications by providing readability at the polymer level, whereas SMILES operates at the atomic level. These approaches advance the integration of cheminformatics and bioinformatics. However, translation errors between biological and chemical peptide notations have been confirmed, and solutions have been proposed to address them.

5.4 Limitations and challenges

Despite being essential in bioinformatics, cheminformatics, and drug discovery, molecular representations face several limitations and challenges. The molecular world is vast and complex, with many aspects still unknown to humans. Molecules exhibit a wide range of structures, from simple linear chains to highly complex branched and ring structures. Large molecules, especially those with intricate 3D configurations, are often inadequately represented by most current methods. Macromolecules, such as proteins and polymers, present additional difficulties due to their long chains, bulky structures, and significant molecular weights, complicating the processes of featurization and encoding. String-based representations, such as SMILES or InChI, offer simplified expressions for all molecules but may fail to accurately capture stereochemistry or conformational details. Graph-based representations include connectivity information but still struggle to represent 3D conformations. Because single bonds can rotate, a molecule can exist in multiple conformations, known as *conformers*. While conformational information is often ignored in some modeling tasks, it can be incorporated into the main graphs as node attributes. Representing molecules with full information on their chiral centers and stereoisomers requires substantial computational resources and specialized tools or software. Current cheminformatics toolkits and libraries can support property-based representations for small or medium-sized molecules but may be slow to process complex structures or unable to compute the physicochemical properties of large molecules. Language model-based representations play crucial roles in various tasks, including property-directed molecule generation, QSAR modeling, and other downstream machine learning tasks. The effectiveness of these representations largely depends on how the pre-trained molecular encoder is developed and can vary across different tasks. Table 8 summarizes all types of molecular representations, highlighting their advantages and disadvantages.

5.5 Future directions and emerging trends

Emerging trends and innovative molecular representations are transforming cheminformatics, particularly in drug discovery, by addressing the limitations of traditional methods. Recently, advanced graph-based deep learning architectures have been developed to tackle challenges in molecular property prediction, de novo molecular design, and representation learning. The introduction of Message Passing Neural Networks (MPNNs) and their learning mechanisms has significantly influenced the development of other deep learning architectures for molecular graphs [114]. Modern graph-based neural networks now incorporate not only connectivity information but also data on molecular structures, substructures, conformation, and properties. Additionally, quantum molecular graphs have emerged as promising alternatives for representing molecules based on quantum mechanical properties and wave functions [115–117]. The rise of transformers and self-attention mechanisms has spurred the development of novel language model-based representations, which can customize the structural patterns of groups of molecules [118]. Quantum computing has made significant progress in recent years, driven by advances in both hardware and algorithms. The potential applications of quantum computing in drug discovery have been extensively discussed [119–121]. While opinions on the practical benefits of quantum computing vary, most computational scientists agree that it can save time and effort by substantially accelerating modeling processes. This acceleration allows for the production of larger models with high generalizability in a shorter time. Quantum computing is also expected to enhance the processing of larger molecular graphs and speed up training and prediction phases. Moreover, pre-trained networks for language model-based representations can be trained on a significantly larger number of molecules than existing models, further enhancing their utility and effectiveness in cheminformatics and drug discovery.

Table 8 All types of molecular representations with highlighted advantages and disadvantages

Task	Representation	Advantages	Disadvantages
Machine Learning	String-based	Simple expression	May fail to capture 3D stereochemistry and conformational details
	Property-based	Effectively work on small medium-sized molecules	May fail to compute features for large and complex molecules
	Molecular fingerprints	Fast computation on capture substructural presence or absence	No connectivity detail provided. Inconvertible
	Language model-based	Adapt to many tasks, especially optimization of molecular properties	Effectiveness of presentations highly depends on how the encoders are trained
	Graph and graph-based	Capture and highlight connectivity details	High computational cost. Struggle in representing 3D conformations
	Others	Depends on particular cases	Depends on particular cases
Structural preservation	Connection table and Molfile	Store information about the atoms, bonds, connectivity, and coordinates of a molecule	Cannot directly used for training machine learning models in their raw forms
Chemical reaction	SMIRKS, RInChI, and others	Store information on chemical reactions from reactants to products following complex reaction rules	May lack the depth and precision needed for advanced computational modeling
Macromolecules	Peptides, proteins, glycans, and polymeric drugs	Featurize and represent large and complex macromolecules	Limited approaches and requires further research to improve these representations

6 Conclusion

The role of molecular representations is pivotal since they provide a variety of methods for converting complex chemical structures into numerical formats that can be efficiently processed and analyzed. The selection of representation may significantly impact the outcomes of downstream tasks, with an appropriate balance between capturing relevant structural information and computational efficiency. Molecular representations facilitate various tasks, including similarity searches, virtual screening, and machine learning. In the future, the ceaseless development of more efficient molecular representations will help improve the power of computational approaches and unlock novel directions in cheminformatics and drug discovery.

Author Contributions T.-H.N.-V.: Conceptualization, Formal analysis, Visualization, Writing—Original Draft. P.T.S.: Validation, Funding acquisition, Writing—Review & Editing, Supervision. J.E.H.: Validation, Funding acquisition, Writing—Review & Editing. B.P.N.: Conceptualization, Validation, Funding acquisition, Writing—Review & Editing, Supervision.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions The work was supported in part by the Faculty Strategic Research Grant (FSRG) Numbers 410132 and 411494 at Victoria University of Wellington (VUW) and the Endeavour Fund (Smart Ideas) from the New Zealand Ministry of Business, Innovation and Employment (MBIE) under contract VUW RTVU2301.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. David L, Thakkar A, Mercado R, Engkvist O (2020) Molecular representations in AI-driven drug discovery: a review and practical guide. J Cheminform. https://doi.org/10.1186/s13321-020-00460-5
2. Raghunathan S, Priyakumar UD (2021) Molecular representations for machine learning applications in chemistry. Int J Quantum Chem 122:7. https://doi.org/10.1002/qua.26870
3. Wigh DS, Goodman JM, Lapkin AA (2022) A review of molecular representation in the age of machine learning. WIREs Comput Mol Sci. https://doi.org/10.1002/wcms.1603
4. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 28(1):31–36. https://doi.org/10.1021/ci00057a005
5. Hirohara M, Saito Y, Koda Y, Sato K, Sakakibara Y (2018) Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. BMC Bioinform. https://doi.org/10.1186/s12859-018-2523-5
6. Heller SR, McNaught A, Pletnev I, Stein S, Tchekhovskoi D (2015) InChI, the IUPAC international chemical identifier. J Cheminform. https://doi.org/10.1186/s13321-015-0068-4
7. Homer RW, Swanson J, Jilek RJ, Hurst T, Clark RD (2008) SYBYL line notation (SLN): a single notation to represent chemical structures, queries, reactions, and virtual libraries. J Chem Inf Model 48(12):2294–2307. https://doi.org/10.1021/ci7004687
8. Krenn M, Häse F, Nigam A, Friederich P, Aspuru-Guzik A (2020) Self-referencing embedded strings (SELFIES): a 100% robust molecular string representation. Mach Learn Sci Technol 1(4):045024. https://doi.org/10.1088/2632-2153/aba947
9. Landrum G et al (2022) RDKit: open-Source Cheminformatics Software (Release 2022.03.2). https://doi.org/10.5281/zenodo.591637. http://www.rdkit.org
10. Kochev N, Avramova S, Jeliazkova N (2018) Ambit-SMIRKS: a software module for reaction representation, reaction search and structure transformation. J Cheminform. https://doi.org/10.1186/s13321-018-0295-6
11. Kochev N, Jeliazkova N, Tancheva G (2021) Ambit-SLN: an open source software library for processing of chemical objects via SLN linear notation. Mol Inform 40(11):2100027. https://doi.org/10.1002/minf.202100027
12. Todeschini R, Consonni V (2009) Molecular descriptors for chemoinformatics, vol 1. Wiley, Germany. https://doi.org/10.1002/9783527628766
13. Moriwaki H, Tian Y-S, Kawashita N, Takagi T (2018) Mordred: a molecular descriptor calculator. J Cheminform 10:1. https://doi.org/10.1186/s13321-018-0258-y
14. Himanen L, Jäger MOJ, Morooka EV, Canova FF, Ranawat YS, Gao DZ, Rinke P, Foster AS (2020) DScribe: library of descriptors for machine learning in materials science. Comput Phys Commun 247:106949. https://doi.org/10.1016/j.cpc.2019.106949
15. Yap CW (2010) PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem 32(7):1466–1474. https://doi.org/10.1002/jcc.21707
16. Dong J, Cao D-S, Miao H-Y, Liu S, Deng B-C, Yun Y-H, Wang N-N, Lu A-P, Zeng W-B, Chen AF (2015) ChemDes: an integrated web-based platform for molecular descriptor and fingerprint computation. J Cheminform 7:1. https://doi.org/10.1186/s13321-015-0109-z
17. Cao D-S, Xu Q-S, Hu Q-N, Liang Y-Z (2013) ChemoPy: freely available python package for computational biology and chemoinformatics. Bioinformatics 29(8):1092–1094. https://doi.org/10.1093/bioinformatics/btt105
18. Willighagen EL, Mayfield JW, Alvarsson J, Berg A, Carlsson L, Jeliazkova N, Kuhn S, Pluskal T, Rojas-Chertó M, Spjuth O, Torrance G, Evelo CT, Guha R, Steinbeck C (2017) The chemistry development kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. J Cheminform. https://doi.org/10.1186/s13321-017-0220-4
19. O'Boyle NM, Morley C, Hutchison GR (2008) Pybel: a python wrapper for the OpenBabel cheminformatics toolkit. Chem Cent J 2(1):66. https://doi.org/10.1186/1752-153x-2-5
20. Cereto-Massagué A, Ojeda MJ, Valls C, Mulero M, Garcia-Vallvé S, Pujadas G (2015) Molecular fingerprint similarity search in virtual screening. Methods 71:58–63. https://doi.org/10.1016/j.ymeth.2014.08.005

21. Seo M, Shin HK, Myung Y, Hwang S, No KT (2020) Development of natural compound molecular fingerprint (NC-MFP) with the dictionary of natural products (DNP) for natural product-based drug development. J Cheminform. https://doi.org/10.1186/s13321-020-0410-3
22. Klekota J, Roth FP (2008) Chemical substructures that enrich for biological activity. Bioinformatics 24(21):2518–2525. https://doi.org/10.1093/bioinformatics/btn479
23. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. J Chem Inf Model 50(5):742–754. https://doi.org/10.1021/ci100050t
24. Nguyen-Vo T-H, Trinh QH, Nguyen L, Do TTT, Chua MCH, Nguyen BP (2021) Predicting antimalarial activity in natural products using pretrained bidirectional encoder representations from transformers. J Chem Inf Model 62(21):5050–5058. https://doi.org/10.1021/acs.jcim.1c00584
25. Nguyen-Vo T-H, Nguyen L, Do N, Le PH, Nguyen T-N, Nguyen BP, Le L (2020) Predicting drug-induced liver injury using convolutional neural network and molecular fingerprint-embedded features. ACS Omega 5(39):25432–25439. https://doi.org/10.1021/acsomega.0c03866
26. Nguyen-Vo T-H, Trinh QH, Nguyen L, Nguyen-Hoang P-U, Nguyen T-N, Nguyen DT, Nguyen BP, Le L (2021) iCYP-MFE: identifying human cytochrome P450 inhibitors using multitask learning and molecular fingerprint-embedded encoding. J Chem Inf Model 62(21):5059–5068. https://doi.org/10.1021/acs.jcim.1c00628
27. Menke J, Massa J, Koch O (2021) Natural product scores and fingerprints extracted from artificial neural networks. Comput Struct Biotechnol J 19:4593–4602. https://doi.org/10.1016/j.csbj.2021.07.032
28. Jaeger S, Fulle S, Turk S (2018) Mol2vec: unsupervised machine learning approach with chemical intuition. J Chem Inf Model 58(1):27–35. https://doi.org/10.1021/acs.jcim.7b00616
29. Chithrananda S, Grand G, Ramsundar B (2020) ChemBERTa: large-scale self-supervised pretraining for molecular property prediction. arXiv. https://doi.org/10.48550/ARXIV.2010.09885
30. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. arXiv. https://doi.org/10.48550/ARXIV.1310.4546
31. Wang S, Guo Y, Wang Y, Sun H, Huang J (2019). SMILES-BERT. ACM. https://doi.org/10.1145/3307339.3342186
32. Fabian B, Edlich T, Gaspar H, Segler M, Meyers J, Fiscato M, Ahmed M (2020) Molecular representation learning with language models and domain-relevant auxiliary tasks. arXiv. https://doi.org/10.48550/ARXIV.2011.13230
33. Ahmad W, Simon E, Chithrananda S, Grand G, Ramsundar B (2022) ChemBERTa-2: towards chemical foundation models. arXiv. https://doi.org/10.48550/ARXIV.2209.01712
34. Wen N, Liu G, Zhang J, Zhang R, Fu Y, Han X (2022) A fingerprints based molecular property prediction method using the BERT model. J Cheminform 14:1. https://doi.org/10.1186/s13321-022-00650-3
35. Devlin J, Chang M-W, Lee K, Toutanova K (2018) BERT: pre-training of deep bidirectional transformers for language understanding. arXiv. https://doi.org/10.48550/ARXIV.1810.04805
36. Yüksel A, Ulusoy E, Ünlü A, Doğan T (2023) SELFormer: molecular representation learning via SELFIES language models. arXiv. https://doi.org/10.48550/ARXIV.2304.04662
37. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) RoBERTa: a robustly optimized bert pretraining approach. arXiv. https://doi.org/10.48550/ARXIV.1907.11692
38. Irwin R, Dimitriadis S, He J, Bjerrum EJ (2022) Chemformer: a pre-trained transformer for computational chemistry. Mach Learn Sci Technol 3(1):015022. https://doi.org/10.1088/2632-2153/ac3ffb
39. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2019) BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv. https://doi.org/10.48550/ARXIV.1910.13461
40. Ross J, Belgodere B, Chenthamarakshan V, Padhi I, Mroueh Y, Das P (2022) Large-scale chemical language representations capture molecular structure and properties. Nat Mach Intell 4(12):1256–1264. https://doi.org/10.1038/s42256-022-00580-7
41. Su J, Lu Y, Pan S, Murtadha A, Wen B, Liu Y (2021) RoFormer: enhanced transformer with rotary position embedding. arXiv. https://doi.org/10.48550/ARXIV.2104.09864
42. Xue D, Zhang H, Chen X, Xiao D, Gong Y, Chuai G, Sun Y, Tian H, Wu H, Li Y, Liu Q (2022) X-MOL: large-scale pre-training for molecular understanding and diverse molecular analysis. Sci Bull 67(9):899–902. https://doi.org/10.1016/j.scib.2022.01.029
43. Shen WX, Zeng X, Zhu F, Wang Y, Qin C, Tan Y, Jiang YY, Chen YZ (2021) Out-of-the-box deep learning prediction of pharmaceutical properties by broadly learned knowledge-based molecular representations. Nat Mach Intell 3(4):334–343. https://doi.org/10.1038/s42256-021-00301-6
44. Chen Y, Li J, Xiao H, Jin X, Yan S, Feng J (2017) Dual path networks, vol 30
45. Zang X, Zhao X, Tang B (2023) Hierarchical molecular graph self-supervised learning for property prediction. Commun Chem. https://doi.org/10.1038/s42004-023-00825-5
46. Hajiabolhassan H, Taheri Z, Hojatnia A, Yeganeh YT (2023) FunQG: molecular representation learning via quotient graphs. J Chem Inf Model 63(11):3275–3287. https://doi.org/10.1021/acs.jcim.3c00445
47. Zhang S, Tong H, Xu J, Maciejewski R (2019) Graph convolutional networks: a comprehensive review. Comput Soc Netw. https://doi.org/10.1186/s40649-019-0069-y
48. Narayanan A, Chandramohan M, Venkatesan R, Chen L, Liu Y, Jaiswal S (2017) graph2vec: learning distributed representations of graphs. arXiv. https://doi.org/10.48550/ARXIV.1707.05005
49. Ji Z, Shi R, Lu J, Li F, Yang Y (2022) ReLMole: molecular representation learning based on two-level graph similarities. J Chem Inf Model 62(22):5361–5372. https://doi.org/10.1021/acs.jcim.2c00798
50. Fang X, Liu L, Lei J, He D, Zhang S, Zhou J, Wang F, Wu H, Wang H (2022) Geometry-enhanced molecular representation learning for property prediction. Nat Mach Intell 4(2):127–134. https://doi.org/10.1038/s42256-021-00438-4
51. Vinh T, Trinh QH, Nguyen L, Nguyen-Vo T-H, Nguyen BP (2024) Predicting cardiotoxicity of molecules using attention-based graph neural network. J Chem Inf Model 64(6):1816–1827. https://doi.org/10.1021/acs.jcim.3c01286
52. Nguyen-Vo T-H, Do TTT, Nguyen BP (2024) An effective ensemble deep learning framework for blood-brain barrier permeability prediction. In: Proceedings of the IEEE conference on artificial intelligence (CAI 2024), Singapore
53. Ramsundar B, Eastman P, Walters P, Pande V, Leswing K, Wu Z (2019) Deep learning for the life sciences. O'Reilly Media, USA
54. Li M, Zhou J, Hu J, Fan W, Zhang Y, Gu Y, Karypis G (2021) DGL-LifeSci: an open-source toolkit for deep learning on graphs in life science. ACS Omega 6(41):27233–27238. https://doi.org/10.1021/acsomega.1c04017
55. McKay BD, Yirik MA, Steinbeck C (2022) Surge: a fast open-source chemical graph generator. J Cheminform 14(1):66. https://doi.org/10.1186/s13321-022-00604-9
56. Elton DC, Boukouvalas Z, Fuge MD, Chung PW (2019) Deep learning for molecular design—a review of the state of the

art. Mol Syst Des Eng 4(4):828–849. https://doi.org/10.1039/c9me00039a
57. Kuzminykh D, Polykovskiy D, Kadurin A, Zhebrak A, Baskov I, Nikolenko S, Shayakhmetov R, Zhavoronkov A (2018) 3d molecular representations based on the wave transform for convolutional neural networks. Mol Pharm 15(10):4378–4385. https://doi.org/10.1021/acs.molpharmaceut.7b01134
58. Amidi A, Amidi S, Vlachakis D, Megalooikonomou V, Paragios N, Zacharaki EI (2018) EnzyNet: enzyme classification using 3d convolutional neural networks on spatial representation. PeerJ 6:4750. https://doi.org/10.7717/peerj.4750
59. Skalic M, Jiménez J, Sabbadin D, Fabritiis GD (2019) Shape-based generative modeling for de novo drug design. J Chem Inf Model 59(3):1205–1214. https://doi.org/10.1021/acs.jcim.8b00706
60. Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017) Inception-v4, inception-ResNet and the impact of residual connections on learning. In: Proceedings of the AAAI conference on artificial intelligence, vol 31, no. 1. https://doi.org/10.1609/aaai.v31i1.11231
61. Goh GB, Siegel C, Vishnu A, Hodas NO, Baker N (2017) Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models. arXiv. https://doi.org/10.48550/ARXIV.1706.06689
62. Bjerrum E, Sattarov B (2018) Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders. Biomolecules 8(4):131. https://doi.org/10.3390/biom8040131
63. Ragoza M, Hochuli J, Idrobo E, Sunseri J, Koes DR (2017) Protein–ligand scoring with convolutional neural networks. J Chem Inf Model 57(4):942–957. https://doi.org/10.1021/acs.jcim.6b00740
64. Liu Q, Wang P-S, Zhu C, Gaines BB, Zhu T, Bi J, Song M (2021) OctSurf: efficient hierarchical voxel-based molecular surface representation for protein–ligand affinity prediction. J Mol Graph Model 105:107865. https://doi.org/10.1016/j.jmgm.2021.107865
65. Ragoza M, Masuda T, Koes DR (2022) Generating 3d molecules conditional on receptor binding sites with deep generative models. Chem Sci 13(9):2701–2713. https://doi.org/10.1039/d1sc05976a
66. Dalby A, Nourse JG, Hounshell WD, Gushurst AKI, Grier DL, Leland BA, Laufer J (1992) Description of several chemical structure file formats used by computer programs developed at molecular design limited. J Chem Inf Comput Sci 32(3):244–255. https://doi.org/10.1021/ci00007a012
67. Delannée V, Nicklaus MC (2020) Reactioncode: format for reaction searching, analysis, classification, transform, and encoding/decoding. J Cheminform 12:1. https://doi.org/10.1186/s13321-020-00476-x
68. Cosgrove DA, Green KM, Leach AG, Poirrette A, Winter J (2012) A system for encoding and searching Markush structures. J Chem Inf Model 52(8):1936–1947. https://doi.org/10.1021/ci3000387
69. Warr WA (2014) A short review of chemical reaction database systems, computer-aided synthesis design, reaction prediction and synthetic feasibility. Mol Inform 33(6–7):469–476. https://doi.org/10.1002/minf.201400052
70. Coley CW, Eyke NS, Jensen KF (2020) Autonomous discovery in the chemical sciences part 2: outlook. Angewandte Chemie Int Ed 59(52):23414–23436. https://doi.org/10.1002/anie.201909989
71. Grethe G, Goodman JM, Allen CH (2013) International chemical identifier for reactions (RInChI). J f Cheminform. https://doi.org/10.1186/1758-2946-5-45
72. Grethe G, Blanke G, Kraut H, Goodman JM (2018) International chemical identifier for reactions (RInChI). J Cheminform. https://doi.org/10.1186/s13321-018-0277-8
73. Jacob P-M, Lan T, Goodman JM, Lapkin AA (2017) A possible extension to the Rinchi as a means of providing machine readable process data. J Cheminform 9:1. https://doi.org/10.1186/s13321-017-0210-6
74. Varnek A, Fourches D, Hoonakker F, Solov'ev VP (2005) Substructural fragments: an universal language to encode reactions, molecular and supramolecular structures. J Comput Aided Mol Des 19(9–10):693–703. https://doi.org/10.1007/s10822-005-9008-0
75. Nugmanov RI, Mukhametgaleev RN, Akhmetshin T, Gimadiev TR, Afonina VA, Madzhidov TI, Varnek A (2019) Cgrtools: python library for molecule, reaction, and condensed graph of reaction processing. J Chem Inf Model 59(6):2516–2521. https://doi.org/10.1021/acs.jcim.9b00102
76. Dugundji J, Ugi I (2023) An algebraic model of constitutional chemistry as a basis for chemical computer programs. Springer, Berlin, pp 19–64. https://doi.org/10.1007/bfb0051317
77. Gasteiger J, Jochum C (2023) EROS A computer program for generating sequences of reactions. Springer, Berlin, pp 93–126. https://doi.org/10.1007/bfb0050147
78. Gasteiger J, Ihlenfeldt WD (2023) The WODCA system. Springer, Berlin, pp 57–65. https://doi.org/10.1007/978-3-642-75430-2_7
79. Rose JR, Gasteiger J (1994) HORACE: an automatic system for the hierarchical classification of chemical reactions. J Chem Inf Comput Sci 34(1):74–90. https://doi.org/10.1021/ci00017a010
80. Kraut H, Eiblmaier J, Grethe G, Löw P, Matuszczyk H, Saller H (2013) Algorithm for reaction classification. J Chem Inf Model 53(11):2884–2895. https://doi.org/10.1021/ci400442f
81. Bøgevig A, Federsel H-J, Huerta F, Hutchings MG, Kraut H, Langer T, Löw P, Oppawsky C, Rein T, Saller H (2015) Route design in the 21st century: the icsynth software tool as an idea generator for synthesis prediction. Organ Process Res Dev 19(2):357–368. https://doi.org/10.1021/op500373e
82. Segler MHS, Preuss M, Waller MP (2018) Planning chemical syntheses with deep neural networks and symbolic AI. Nature 555(7698):604–610. https://doi.org/10.1038/nature25978
83. Raymond JW, Willett P (2002) J Comput Aided Mol Des 16(7):521–533. https://doi.org/10.1023/a:1021271615909
84. Ehrlich H, Rarey M (2011) Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. WIREs Comput Mol Sci 1(1):68–79. https://doi.org/10.1002/wcms.5
85. Chen WL, Chen DZ, Taylor KT (2013) Automatic reaction mapping and reaction center detection. WIREs Comput Mol Sci 3(6):560–593. https://doi.org/10.1002/wcms.1140
86. Morgan HL (1965) The generation of a unique machine description for chemical structures—a technique developed at chemical abstracts service. J Chem Doc 5(2):107–113. https://doi.org/10.1021/c160017a018
87. Patel H, Bodkin MJ, Chen B, Gillet VJ (2009) Knowledge-based approach to de novo design using reaction vectors. J Chem Inf Model 49(5):1163–1184. https://doi.org/10.1021/ci800413m
88. Ghiandoni GM, Bodkin MJ, Chen B, Hristozov D, Wallace JEA, Webster J, Gillet VJ (2019) Development and application of a data-driven reaction classification model: comparison of an electronic lab notebook and medicinal chemistry literature. J Chem Inf Model 59(10):4167–4187. https://doi.org/10.1021/acs.jcim.9b00537
89. Schneider N, Lowe DM, Sayle RA, Landrum GA (2015) Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. J Chem Inf Model 55(1):39–53. https://doi.org/10.1021/ci5006614
90. Coley CW, Green WH, Jensen KF (2019) RDChiral: an RDKit wrapper for handling stereochemistry in retrosynthetic template extraction and application. J Chem Inf Model 59(6):2529–2537. https://doi.org/10.1021/acs.jcim.9b00286

91. Walter (1984) Nomenclature and symbolism for amino acids and peptides (Recommendations 1983). Pure Appl Chem **56**(5):595–624. https://doi.org/10.1351/pac198456050595
92. Siani MA, Weininger D, Blaney JM (1994) CHUCKLES: a method for representing and searching peptide and peptoid sequences on both monomer and atomic levels. J Chem Inf Comput Sci 34(3):588–593. https://doi.org/10.1021/ci00019a017
93. Minkiewicz Iwaniak (2019) Darewicz: Biopep-uwm database of bioactive peptides: current opportunities. Int J Mol Sci 20(23):5978. https://doi.org/10.3390/ijms20235978
94. Siani MA, Weininger D, James CA, Blaney JM (1995) CHORTLES: a method for representing oligomeric and template-based mixtures. J Chem Inf Comput Sci 35(6):1026–1033. https://doi.org/10.1021/ci00028a012
95. Zhang T, Li H, Xi H, Stanton RV, Rotstein SH (2012) HELM: a hierarchical notation language for complex biomolecule structure representation. J Chem Inf Model 52(10):2796–2806. https://doi.org/10.1021/ci3001925
96. Milton J, Zhang T, Bellamy C, Swayze E, Hart C, Weisser M, Hecht S, Rotstein S (2017) HELM software for biopolymers. J Chem Inf Model 57(6):1233–1239. https://doi.org/10.1021/acs.jcim.6b00442
97. Chen WL, Leland BA, Durant JL, Grier DL, Christie BD, Nourse JG, Taylor KT (2011) Self-contained sequence representation: bridging the gap between bioinformatics and cheminformatics. J Chem Inf Model 51(9):2186–2208. https://doi.org/10.1021/ci2001988
98. Pistoia Alliance (2024) HELM project. https://www.pistoiaalliance.org/projects/current-projects/helm/. Accessed 19 May 2024
99. Bohne-Lang A, Lang E, Förster T, Lieth C-W (2001) LINUCS: linear notation for unique description of carbohydrate sequences. Carbohyd Res 336(1):1–11. https://doi.org/10.1016/s0008-6215(01)00230-0
100. Herget S, Ranzinger R, Maass K, Lieth C-Wvd (2008) GlycoCT—a unifying sequence format for carbohydrates. Carbohydr Res 343(12):2162–2171. https://doi.org/10.1016/j.carres.2008.03.011
101. Ranzinger R, Kochut KJ, Miller JA, Eavenson M, Lütteke T, York WS (2017) GLYDE-II: the glycan data exchange format. Perspect Sci 11:24–30. https://doi.org/10.1016/j.pisc.2016.05.013
102. Toukach PV, Egorova KS (2019) New features of carbohydrate structure database notation (csdb linear), as compared to other carbohydrate notations. J Chem Inf Model 60(3):1276–1289. https://doi.org/10.1021/acs.jcim.9b00744
103. Tanaka K, Aoki-Kinoshita KF, Kotera M, Sawaki H, Tsuchiya S, Fujita N, Shikanai T, Kato M, Kawano S, Yamada I, Narimatsu H (2014) WURCS: the web3 unique representation of carbohydrate structures. J Chem Inf Model 54(6):1558–1566. https://doi.org/10.1021/ci400571e
104. Matsubara M, Aoki-Kinoshita KF, Aoki NP, Yamada I, Narimatsu H (2017) WURCS 2.0 update to encapsulate ambiguous carbohydrate structures. J Chem Inf Mode 57(4):632–637. https://doi.org/10.1021/acs.jcim.6b00650
105. Tiemeyer M, Aoki K, Paulson J, Cummings RD, York WS, Karlsson NG, Lisacek F, Packer NH, Campbell MP, Aoki NP, Fujita A, Matsubara M, Shinmachi D, Tsuchiya S, Yamada I, Pierce M, Ranzinger R, Narimatsu H, Aoki-Kinoshita KF (2017) GlyTouCan: an accessible glycan structure repository. Glycobiology 27(10):915–919. https://doi.org/10.1093/glycob/cwx066
106. Bojar D, Camacho DM, Collins JJ (2020) Using natural language processing to learn the grammar of glycans. Cold Spring Harbor Laboratory https://doi.org/10.1101/2020.01.10.902114
107. Lin T-S, Coley CW, Mochigase H, Beech HK, Wang W, Wang Z, Woods E, Craig SL, Johnson JA, Kalow JA, Jensen KF, Olsen BD (2019) BigSMILES: a structurally-based line notation for describing macromolecules. ACS Cent Sci 5(9):1523–1531. https://doi.org/10.1021/acscentsci.9b00476
108. Lin T-S, Rebello NJ, Lee G-H, Morris MA, Olsen BD (2022) Canonicalizing bigsmiles for polymers with defined backbones. ACS Polym Au 2(6):486–500. https://doi.org/10.1021/acspolymersau.2c00009
109. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16(1):321–357
110. Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. https://doi.org/10.1145/2939672.2939785
111. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y (2017) LightGBM: a highly efficient gradient boosting decision tree. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, vol 30. Curran Associates, Inc
112. Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A (2017) CatBoost: unbiased boosting with categorical features. arXiv. https://doi.org/10.48550/ARXIV.1706.09516
113. Nguyen L, Nguyen Vo T-H, Trinh QH, Nguyen BH, Nguyen-Hoang P-U, Le L, Nguyen BP (2022) iANP-EC: identifying anticancer natural products using ensemble learning incorporated with evolutionary computation. J Chem Inf Model 62(21):5080–5089. https://doi.org/10.1021/acs.jcim.1c00920
114. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017) Neural message passing for quantum chemistry. arXiv. https://doi.org/10.48550/ARXIV.1704.01212
115. Balasubramanian K, Gupta SP (2019) Quantum molecular dynamics, topological, group theoretical and graph theoretical studies of protein–protein interactions. Curr Top Med Chem 19(6):426–443. https://doi.org/10.2174/1568026619666190304152704
116. Kneiding H, Lukin R, Lang L, Reine S, Pedersen TB, De Bin R, Balcells D (2023) Deep learning metal complex properties with natural quantum graphs. Digit Discov 2(3):618–633. https://doi.org/10.1039/d2dd00129b
117. Yan G, Wu H, Yan J (2023) Quantum 3D graph learning with applications to molecule embedding, vol 202, pp 39126–39137
118. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. arXiv. https://doi.org/10.48550/ARXIV.1706.03762
119. Cao Y, Romero J, Aspuru-Guzik A (2018) Potential of quantum computing for drug discovery. IBM J Res Dev 62(6):6–1620. https://doi.org/10.1147/jrd.2018.2888987
120. Batra K, Zorn KM, Foil DH, Minerali E, Gawriljuk VO, Lane TR, Ekins S (2021) Quantum machine learning algorithms for drug discovery applications. J Chem Inf Model 61(6):2641–2647. https://doi.org/10.1021/acs.jcim.1c00166
121. Blunt NS, Camps J, Crawford O, Izsák R, Leontica S, Mirani A, Moylett AE, Scivier SA, Sünderhauf C, Schopf P, Taylor JM, Holzmann N (2022) Perspective on the current state-of-the-art of quantum computing for drug discovery applications. J Chem Theory Comput 18(12):7001–7023. https://doi.org/10.1021/acs.jctc.2c00574