PCCP

ROYAL SOCIETY OF CHEMISTRY

PERSPECTIVE

Cite this: *Phys. Chem. Chem. Phys.*, 2020, **22**, 4343

A review of mathematical representations of biomolecular data

Duc Duy Nguyen, [a] Zixuan Cang [a] and Guo-Wei Wei *[abc]

Received 3rd December 2019,
Accepted 17th January 2020

DOI: 10.1039/c9cp06554g

rsc.li/pccp

Recently, machine learning (ML) has established itself in various worldwide benchmarking competitions in computational biology, including Critical Assessment of Structure Prediction (CASP) and Drug Design Data Resource (D3R) Grand Challenges. However, the intricate structural complexity and high ML dimensionality of biomolecular datasets obstruct the efficient application of ML algorithms in the field. In addition to data and algorithm, an efficient ML machinery for biomolecular predictions must include structural representation as an indispensable component. Mathematical representations that simplify the biomolecular structural complexity and reduce ML dimensionality have emerged as a prime winner in D3R Grand Challenges. This review is devoted to the recent advances in developing low-dimensional and scalable mathematical representations of biomolecules in our laboratory. We discuss three classes of mathematical approaches, including algebraic topology, differential geometry, and graph theory. We elucidate how the physical and biological challenges have guided the evolution and development of these mathematical apparatuses for massive and diverse biomolecular data. We focus the performance analysis on protein–ligand binding predictions in this review although these methods have had tremendous success in many other applications, such as protein classification, virtual screening, and the predictions of solubility, solvation free energies, toxicity, partition coefficients, protein folding stability changes upon mutation, *etc.*

I Introduction

Recently, Google's DeepMind has caught the world's breath in winning the 13th Critical Assessment of Structure Prediction (CASP13) competition using its latest artificial intelligence (AI) system, AlphaFold.[1] The goal of the CASP is to develop and recognize the state-of-the-art technology in constructing protein three-dimensional (3D) structures from protein sequences, which are abundantly available nowadays. While many people were surprised by the power of AI when AlphaGo beat humans for the first time in the highly intelligent Go game a few years ago, it was not clear whether AI could tackle scientific challenges. Since CASP has been regarded as one of the most important challenges in computational biophysics, AlphaFold's dominant win of 25 out of 43 contests ushers in a new era of scientific discovery.

The algorithms underpinning the ALphaFold's AI system are machine learning (ML), including deep learning (DL). Indeed, ML is one of the most transformative technologies in history. The combination of big data and ML has been referred to as both the "fourth industrial revolution"[2] and the "fourth paradigm of science".[3] However, this two-element combination may not work very well for biological science, particularly, biomolecular systems because of the intricate structural complexity and the intrinsic high dimensionality of biomolecular datasets.[4] For example, a typical human protein–drug complex has so many possible configurations that even if a computer enumerates one possible configuration per second, it would still take longer than the universe has existed to reach the right configuration. The chemical and pharmacological spaces of drugs are so large that even all the world's computers put together do not have enough power for automated *de novo* drug design due to additional requirements in solubility, partition coefficient, permeability, clearance, toxicity, pharmacokinetics, pharmacodynamics, *etc.*

An appropriate low-dimensional representation of biomolecular structures is required[4–9] to translate the complex structural information into machine learning feature vectors or mathematical representations as shown in Fig. 1 and 2. As a result, various machine learning algorithms, particularly relatively simple ones without complex internal structures, can work efficiently and robustly with biomolecular data.

Descriptors or fingerprints are indispensable even for small molecules – they play a fundamental role in quantitative

[a] *Department of Mathematics, Michigan State University, MI 48824, USA. E-mail: weig@msu.edu*
[b] *Department of Biochemistry and Molecular Biology, Michigan State University, MI 48824, USA*
[c] *Department of Electrical and Computer Engineering, Michigan State University, MI 48824, USA*

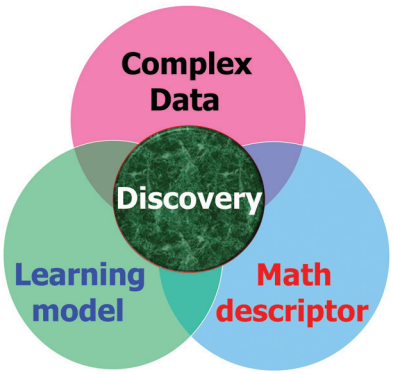

Fig. 1 Illustration of essential elements for machine learning (ML) based discovery from complex biomolecular data.

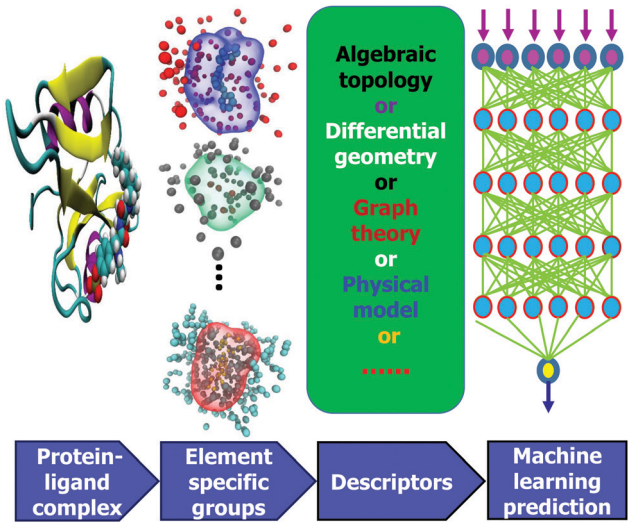

Fig. 2 Illustration of descriptor-based learning processes.

structure–activity relationship (QSAR) and quantitative structure–property relationship (QSPR) analysis, virtual screening, similarity-based compound search, target molecule ranking, drug absorption, distribution, metabolism, and excretion (ADME) prediction, and other drug discovery processes. Molecular descriptors are property profiles of a molecule, usually in the form of vectors with each vector component indicating the existence, the degree or the frequency of a certain structure feature.[10–12] Various descriptors have been developed in the past few decades.[13–15] Most of them are 2D ones that can be extracted from molecular simplified molecular-input line-entry system (SMILES) strings without 3D structure information. High dimensional descriptors have also been developed to utilize 3D molecular structures and other chemical and physical information.[16] There are four main categories of 2D descriptors: (1) substructure keys-based fingerprints, (2) topological or path-based fingerprints, (3) circular fingerprints, and (4) pharmacophore fingerprints. Substructure keys-based fingerprints, such as the molecular access system (MACCS),[17] are bit strings representing the presence of certain substructures or fragments from a given list of structural keys in a molecule. Topological or path-based descriptors, *e.g.*, FP2,[18] Daylight[19] and electro-topological state (Estate),[20] are designed to analyze all the fragments of a molecule following a (usually linear) path up to a certain number of bonds, and then hashing every one of these paths to create fingerprints. Circular fingerprints, such as the extended-connectivity fingerprint (ECFP),[13] are also hashed topological fingerprints but rather than looking for paths in a molecule, they record the environment of each atom up to a pre-defined radius. Pharmacophore fingerprints include the relevant features and interactions needed for a molecule to be active against a given target, including 2D-pharmacophore,[21] 3D-pharmacophore[21] and extended reduced graph (ERG)[22] fingerprints as examples.

However, typically designed for 2D SMILES strings, the aforementioned small-molecular descriptors do not work well for macromolecules that have complex 3D structures. The complexity of biomolecular structure, function, and dynamics often makes the structural representation inconclusive, inadequate, inefficient and sometimes intractable. These challenges call for innovative design strategies for the representation of macromolecules.

Popular molecular mechanics models use bonded terms for describing covalent bond interactions and non-bonded terms for representing long-range electrostatic and van der Waals effects. As a result, the early effort has been focused on exploring related *physical descriptors* to account for hydrogen bonds, electrostatic effects, van der Waals interactions, hydrophilicity, and hydrophobicity. These descriptors have been applied to many macromolecular systems, such as protein–protein interaction hot spots.[6,7,23,24] Similar physical descriptors in terms of van der Waals interactions, Coulomb interactions, electrostatic potentials, electrostatic binding free energies, reaction field energies, surface areas, volumes, *etc.*, were applied by us to predictions of protein–ligand affinity[25] and solvation free energy.[26,27] However, the major limitation of physical descriptors is that they highly depend on existing molecular force fields, such as radii, partial charges, polarizability, dielectric constants, and van der Waals well depth, and thus could inherit errors from upstream physical models. As a result, these descriptors are often not as competitive as the state-of-the-art force-field-free models based on advanced mathematics.[9,28]

Topology analyzes space, connectivity, dimension, and transformation. Topology offers the highest level of abstraction and thus could provide an efficient tool for tackling high-dimensional biological data.[30–32] However, topology typically oversimplifies geometric information. Persistent homology is a new branch of algebraic topology that is able to bridge geometry and topology.[31,33,34] This approach has been applied to macromolecular analysis.[35–45] Nonetheless, it neglects critical chemical/biological information when it is directly applied to complex biomolecular structures. Recently, we have introduced element-specific persistent homology to retain critical biological information during the topological abstraction, rendering a potentially revolutionary representation for biomolecular data (see Fig. 2).[46–49]

Graph theory studies the modeling of pairwise relations between vertices or nodes.[50] Geometric graphs admit geometric

objects as graph nodes while algebraic graphs utilize algebraic techniques to study the relations between nodes. Both geometric graph theory and algebraic graph theory have been widely applied to biomolecular systems.[8,51–53] For example, spectral graph theory has been used to represent protein C_α atoms as an elastic mass-and-spring network in the Gaussian network model (GNM)[54] and the anisotropic network model (ANM).[55] Extremal graph theory concerns unavoidable patterns and structures in graphs with a given density or distribution. It has potential applications to chromosome packing and Hi-C data. However, most graph theory methods suffer from the neglecting of critical biological information and non-covalent interactions, and sometimes, inappropriate distance metrics for biomolecular interactions. In the past few years, we have developed weighted graphs,[56–62] multiscale graphs,[60,63] and colored graphs[64,65] for modeling biomolecular systems. These new graph theory methods are found to be some of the most powerful representations of macromolecules.[64–66]

How biomolecules assume complex structures and intricate shapes and why biomolecular complexes admit convoluted interfaces between different parts can be naturally described by differential geometry, a mathematical subject drawing on differential calculus, integral calculus, algebra, and differential equations to study problems in geometry or differentiable manifolds. Einstein used this approach to formulate his general theory of relativity. Curve and curvature analysis has been applied to the shape analysis of molecular surfaces[67] and protein folding trajectories.[68,69] In the past two decades, we have developed a variety of differential geometry models for biomolecular surface analysis,[70–75] solvation modeling,[76–85] ion-channel study,[80–82,86,87] protein binding pocket detection,[88] and protein–ligand binding affinity prediction.[89] Differential geometry-based representations are able to offer a high-level abstraction of macromolecular structures.[89]

We have pursued differential geometry, algebraic topology, graph theory and other mathematical methods, such as de Rham–Hodge theory,[90,91] for modeling, analysis and characterization of biomolecular systems for nearly two decades. Using these representations, we have studied a number of biomolecular systems and problems, including macromolecular electrostatics, implicit solvent models, ion channels, protein flexibility, geometric analysis, surface modeling, and multiscale analysis. Our mathematical representations have evolved and improved over time. In 2015, we proposed one of the first integrations of persistent homology and machine learning and applied this new approach to protein classification. Since then, we have demonstrated the superiority of our mathematical representations over other existing methods in a wide variety of other applications, including the predictions of protein thermal fluctuations,[59,60,63,65] toxicity,[92] protein–ligand binding affinity,[25,47,64,66,89] mutation-induced protein stability changes,[46,48] solvation,[26,27,79,93] solubility,[94] partition coefficients[94] and virtual screening.[49] As shown in Fig. 3, the aforementioned mathematical approaches have enabled us to win many contests in D3R Grand Challenges, a worldwide competition series in computer-aided drug design.[28]

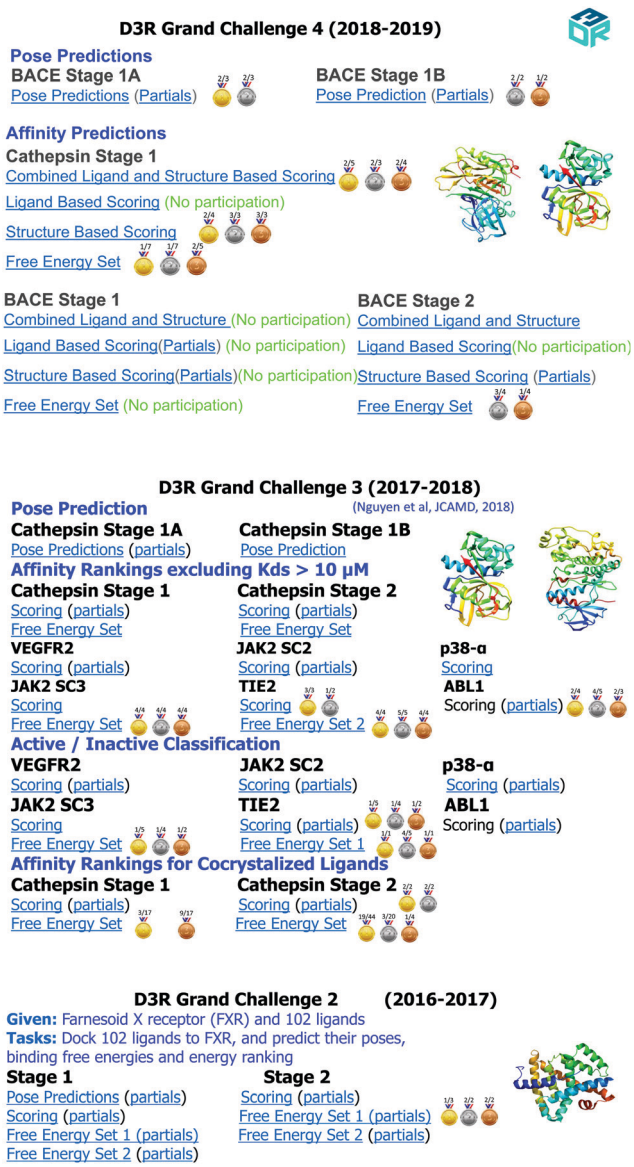

Fig. 3 Wei team's performance in D3R Grand Challenges 2, 3 and 4,[28,29] community-wide competition series in computer-aided drug design, with components addressing blind predictions of pose-prediction, affinity ranking, and binding free energy. The gold medal, silver medal, and bronze medal label the contest where our prediction was ranked 1st, 2nd, and 3rd, respectively. The numbers (*a*/*b*) right beside each medal, say the gold medal, imply that we have *a* predictions that were ranked 1st and there was a total of *b* submissions sharing the first position. "No Participation" is placed in the contests that we unintendedly did not participate due to the inconsistent announcement from the D3R organizer.

Due to the abstract nature of mathematical representations and the fact that our results are scattered over a large number of subjects and topics it is difficult for a researcher who has no formal training in mathematics to use these methods. Therefore, there is a pressing need to elucidate these methods in physical terms, provide simplified representations, and interpret their working principles. To this end, we provide a review of our mathematical representations. Our goal is to offer a coherent description of these methods for protein–ligand

binding interactions so that the reader can better understand how to use advanced mathematics for describing macromolecules and their interaction complexes.

Like small molecular descriptors, macromolecular representations, once designed, can be applied to different tasks in principle. However, many different types of applications require specially designed macromolecular representations. For example, in protein *B*-factor prediction, one deals with atomic properties, while in predicting protein stability changes upon mutation, solubility, *etc.* one considers molecular properties. Additionally, in protein–ligand binding affinity predictions, one deals with the properties of protein–ligand complexes. Therefore, different mathematical representations are required to tackle atomic, molecular, and molecular complex properties. Another complication is due to different systems. For example, representations for the binding affinity of protein–ligand interactions should differ from those for the binding affinity of protein–protein interactions or protein–nucleic acid interactions. The other hindrance arises from specific tasks. For example, in protein classification, one deals with secondary structures and needs to design macromolecular representations to capture secondary structural differences. In general, macromolecules and their interactive complexes inherently contain multiscale, multiphysics, multi-dynamics, and multifunction. Their descriptions can vary from cases to cases. We cannot cover all possible situations in this review.

Biologically, protein–ligand binding interactions are tremendously important for living organisms. Ligand–receptor agonist binding is known to initiate a vast variety of molecular and/or cellular processes, from transmitter-mediated signal transduction, hormone or growth factor regulated metabolic pathways, stimulus-initiated gene expression, enzyme production, to cell secretion. Therefore, the understanding of protein–ligand binding interactions is a central issue in biological sciences, including drug design and discovery. Despite much research in the past, the molecular mechanism of protein–ligand binding interactions is still elusive. A prevalent view is that protein–ligand binding is initiated through protein–ligand molecular recognition, synergistic cooperation, and conformational changes. Computationally, the prediction of protein–ligand binding affinity is sufficiently challenging. Consequently, we focus on mathematical representations for protein–ligand binding affinity predictions to illustrate their design and application in the present review.

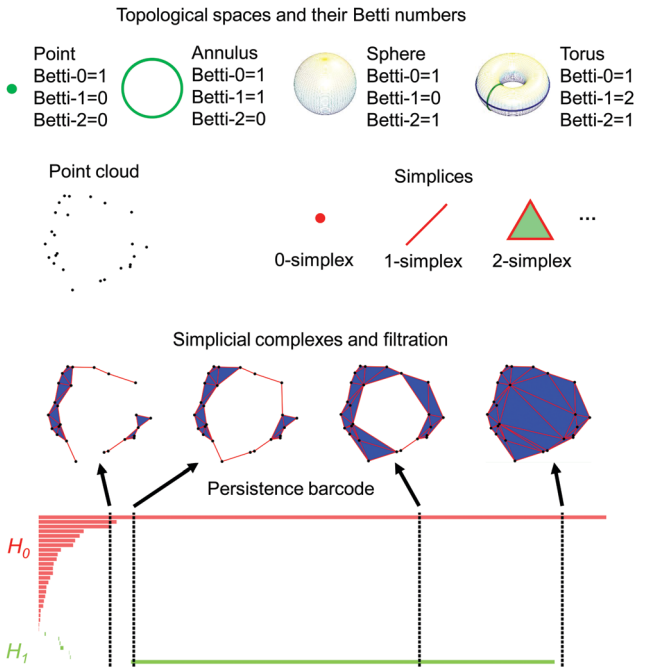

Fig. 4 Topological representation of point clouds *via* persistent homology. Top panel: The Betti numbers for some objects. Middle panel: Many datasets are represented as a point cloud and the simplices are the building blocks for constructing a simplicial complex to topologically characterize the point cloud. Bottom panel: The persistence barcode of the point cloud and some example simplicial complexes at different stages of the filtration.

II Methods

In this section, we briefly review three classes of mathematical representations, *i.e.*, representations constructed from algebraic topology, graph theory, and differential geometry.

II.A Algebraic topology-based methods

II.A.1 Background. Topology dramatically simplifies geometric complexity.[23,30–32,95–98] The study of topology deals with the connectivity of different components in space and characterizes independent entities, rings, and higher dimensional faces within the space.[99] For example, simplicial homology, a type of algebraic topology, concerns the identification of topological invariants from a set of discrete node coordinates such as atomic coordinates in a protein. For a given (protein) configuration, independent components, rings, and cavities are topological invariants and their numbers are called Betti-0, Betti-1, and Betti-2, respectively, see Fig. 4. To study topological invariants in a discrete dataset, simplicial complexes are constructed by gluing simplices under various settings, such as the Vietoris–Rips (VR) complex, Čech complex or alpha complex. Specifically, a 0-simplex is a vertex, a 1-simplex an edge, a 2-simplex a triangle, and a 3-simplex a tetrahedron, as illustrated in Fig. 4. Algebraic groups built on these simplicial complexes are used in simplicial homology to systematically compute various Betti numbers. There is also a cubical complex[99] built upon volumetric data, including those from biomolecules.[44]

However, conventional topology or homology is truly free of metrics or coordinates, and thus retains too little geometric information to be practically useful. Persistent homology is a relatively new branch of algebraic topology that embeds multiscale geometric information into topological invariants to achieve a topological description of geometric details.[31,33] It creates a sequence of topological spaces of a given object by varying a filtration parameter, such as the radius of a ball or the level set of a surface function as shown in Fig. 4. As a result, persistent homology can capture topological structures continuously over a range of spatial scales. Unlike commonly used computational homology which results in truly metric free representations,

persistent homology embeds essential geometric information into topological invariants, *e.g.*, topological representations or barcodes[100] shown in Fig. 4, so that "birth" and "death" of isolated components, circles, rings, voids or cavities can be monitored at all geometric scales by topological measurements. A schematic illustration of our persistent homology-based machine learning predictions is given in Fig. 6. Key concepts are briefly discussed below. More mathematical details can be found in the literature,[31] including ours.[37,38]

Simplicial complex. A simplicial complex is a topological space consisting of vertices (points), edges (line segments), triangles, and their high dimensional counterparts. Based on the simplicial complex, simplicial homology can be defined and used to analyze topological invariants. The essential building blocks of a geometry induced simplicial complex are simplices. Specifically, let v_0, v_1, v_2,...,v_k be $k + 1$ affinely independent points; a (geometric) k-simplex $\sigma^k = \{v_0,v_1,v_2,\ldots,v_k\}$ is the convex hull of these points in $\mathbb{R}^N$ ($N \geq k$), and can be expressed as

$$\sigma^k = \left\{ \lambda_0 v_0 + \lambda_1 v_1 + \cdots + \lambda_k v_k \middle| \sum_{i=0}^{k} \lambda_i = 1; 0 \leq \lambda_i \leq 1,\ \ i = 0,1,\ldots,k \right\}.$$

An i-dimensional face of σ^k is defined as the convex hull formed by the subset of $i + 1$ vertices from σ^k ($k \geq i$). Geometrically, a 0, 1, 2, and 3-simplex corresponds to a vertex, an edge, a triangle, and a tetrahedron, respectively. A simplicial complex K is a finite set of simplices such that any face of a simplex from K is also in K and the intersection of any two simplices in K is either empty or a face of both. The underlying space $|K|$ is a union of all the simplices of K, *i.e.*, $|K| = \cup_{\sigma \in K}\sigma$.

Homology. The basic algebraic structures, chain groups, are defined for simplicial complexes so that homology can be characterized. A k-chain $[\sigma^k]$ is a formal sum $\sum_i \alpha_i \sigma_i^k$ of k-simplices σ_i^k. The coefficients α_i are often chosen in an algebraic field (typically, $\mathbb{Z}_2$). The set of all k-chains of the simplicial complex K together with addition operation forms an abelian group $C_k(K, \mathbb{Z}_2)$. The homology of a topological space is represented also by a series of abelian groups, constructed based on these spaces of chains connected by boundary operators. The boundary operator on chains ∂_k:$C_k \to C_{k-1}$ is defined by linear extension from the boundary operators on simplices. The boundary of a k-simplex $\sigma^k = \{v_0,v_1,v_2,\ldots,v_k\}$ is defined to be the alternating sum of its codimension-1 faces, $\partial_k\sigma^k = \sum_{i=0}^{k} (-1)^i \{v_0, v_1, \ldots, \hat{v}_i, \ldots, v_k\}$, where $\{v_0,v_1,\ldots,\hat{v}_i,\ldots,v_k\}$ is the $(k-1)$-simplex excluding v_i from the vertex set. A key property of the boundary operator is that $\partial_{k-1}\partial_k = \varnothing$ and $\partial_0 = \varnothing$. The k-cycle group Z_k and the k-boundary group B_k are the subgroups of C_k defined as $Z_k = \mathrm{Ker}\ \partial_k = \{c \in C_k | \partial_k c = \varnothing\}$, $B_k = \mathrm{Im}\ \partial_{k+1} = \{\partial_{k+1} c | c \in C_{k+1}\}$.

An element in the k-th cycle group Z_k (or the k-th boundary group B_k) is called a k-cycle (or the k-boundary, resp.). As the boundary of a boundary is always empty $\partial_{k-1}\partial_k = \varnothing$, one has $B_k \subseteq Z_k \subseteq C_k$. Topologically, a k-cycle is a union of k dimensional loops (or closed membranes). The k-th homology group H_k is the quotient group generated by the k-cycle group Z_k and k-boundary group B_k: $H_k = Z_k/B_k$. Two k-cycles are called homologous if they differ by a k-boundary element. From the fundamental theorem of finitely generated abelian groups, the k-th homology group H_k can be expressed as a direct sum, $H_k = Z \oplus \cdots \oplus Z \oplus Z_{p_1} \oplus \cdots \oplus Z_{p_n} = Z^{\beta_k} \oplus Z_{p_1} \oplus \cdots \oplus Z_{p_n}$, where β_k, the rank of the free subgroup, is the k-th Betti number. Here Z_{p_i} is torsion subgroup with torsion coefficients $\{p_i | i = 1, 2,\ldots,n\}$, powers of prime numbers. The Betti number can be simply calculated by $\beta_k = \mathrm{rank}\ H_k = \mathrm{rank}\ Z_k - \mathrm{rank}\ B_k$. The geometric interpretations of Betti numbers in $\mathbb{R}^3$ are as follows: β_0 represents the number of isolated components, β_1 is the number of independent one-dimensional loops (or circles), and β_2 describes the number of independent two-dimensional voids (or cavities). Together, the Betti numbers $\{\beta_0,\beta_1,\beta_2,\ldots\}$ describe the intrinsic topological property of a system.

Persistent homology. For a simplicial complex K, a filtration is defined as a nested sequence of subcomplexes, $\varnothing = K^0 \subseteq K^1 \subseteq \cdots \subseteq K^m = K$. Generally speaking, abstract simplicial complexes generated from a filtration give a multiscale topological representation of the original space, from which related homology groups can be evaluated to reveal topological features. Specifically, upon passing the previous sequence to homology, we obtain a sequence of vector spaces connected by homomorphisms: $H_*(K^0) \to H_*(K^1) \to \cdots \to H_*(K^m)$. Following this sequence of homology groups, sometimes new homology classes are created (*i.e.*, without pre-image under the map $H_*(K^i) \to H_*(K^{i+1})$), and sometimes certain homology classes are destroyed (*i.e.*, they have trivial image under $H_*(K^j) \to H_*(K^{j+1})$). The concept of persistence is introduced to measure the "lifetime" of such homological features. The results can be summarized in the *persistence barcodes* (or equivalently *persistence diagrams*), consisting of a set of intervals $[x,y)$ with the beginning and ending values representing the birth and death of homology classes. The introduction of filtration is of essential importance and directly leads to the invention of persistent homology. Generally speaking, abstract simplicial complexes generated from a filtration give a multiscale representation of the corresponding topological space, from which related homology groups can be evaluated to reveal topological features. Furthermore, the concept of persistence is introduced for long-lasting topological features. However, we have shown that short-lived topological features are also important for biomolecular systems.[37] The p-persistent of the k-th homology group, K^i, is

$$H_k^{i,p} = Z_k^i / (B_k^{i+p} \cap Z_k^i). \quad (1)$$

Through the study of the persistence pattern of these topological features, the so-called persistent homology is capable of capturing the intrinsic properties of the underlying space solely from a discrete point set.

Filtration. Given a set of discrete sample points, there are different ways to construct simplicial complexes. Typical

constructions are based on the intersection patterns of the set of expanding balls centered at the sample points, such as the Čech complex, (Vietoris–)Rips complex and alpha complex.[101,102] The corresponding topological invariants, *e.g.*, the Betti numbers, could be different depending on the choice of simplicial complexes. A common filtration for a set of atomistic data of a macromolecule is constructed by enlarging a common atomic radius r from 0. As the value of r increases, the solid balls will grow and new simplices can be defined through the overlaps among the set of balls. In Fig. 4, we illustrate this process by a set of points. In Fig. 5, we demonstrate the persistent homology analysis of different aspects of a protein–ligand complex using the barcode representation.

II.A.2 Challenge. Conventional topology and homology are independent of metrics or coordinates and thus retain too little geometric information to be practically useful in most biomolecular systems. While persistent homology incorporates more geometric information, it typically treats all atoms in a macromolecule indifferently, which fails to recognize detailed chemical, physical, and biological information.[35,36] We introduced persistent homology as a quantitative tool for analyzing biomolecular systems.[37–42,44,45] In particular, we introduced one of the first topology-based machine learning algorithms for protein classification in 2015.[43] We further introduced element specific persistent homology, *i.e.*, element-induced topology, to deal with massive and diverse bimolecular datasets.[43,45–48] Moreover, we introduced multi-level persistent homology to extract non-covalent-bond interactions.[49] Furthermore, physics-embedded persistent homology was proposed to incorporate physical laws into topological invariants.[49] These new topological tools are potentially revolutionary for complex biomolecular data analysis.[9]

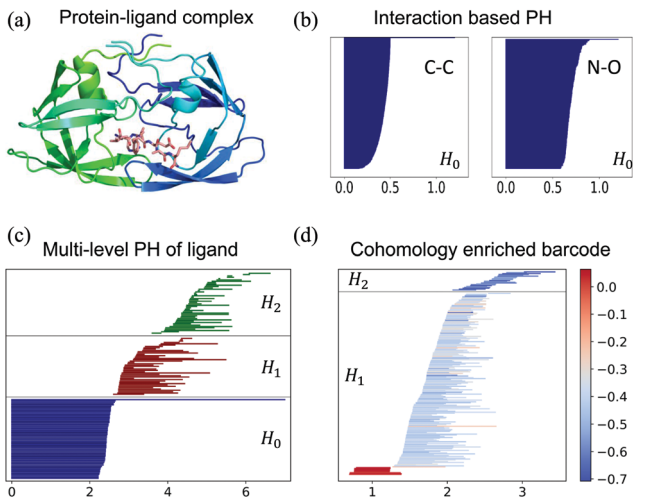

Fig. 5 Topological fingerprints addressing different aspects of the protein–ligand complex. (a) The example protein–ligand complex (PDB: 1A94). (b) The H_0 barcodes from Rips filtration based on the Coulomb potential for carbon–carbon and nitrogen–oxygen interactions between the protein and ligand. (c) The multi-level persistent homology characterization of the ligand revealing the non-covalent intramolecular interaction network. (d) The enriched barcode *via* persistent cohomology for atomic partial charges as the non-geometric information.

II.A.3 Element specific persistent homology. Many types of interactions exist in a protein–ligand complex, for example, hydrophobic effects, hydrogen bonds, and electrostatics. Due to the mechanisms of these interactions, they happen under different geometric distances. Persistent homology, when applied to all the atoms, however, will mostly capture the interactions among nearest neighbors and hinder the detection of long-range interactions. Additionally, it does not distinguish the difference between different element types and their combinations and thus, neglects important chemistry and biology. Element specific persistent homology provides a simple yet effective solution to these issues. Instead of computing persistent homology for the whole molecule once, we perform persistent homology computations on a collection of subsets of atoms. For example, persistent homology on only carbon atoms characterizes the hydrophobic interaction network and the hydrogen bond interactions can be described by persistent homology on the set of nitrogen and oxygen atoms. Although different types of interactions have different characteristics, they may also influence each other. This encourages the iteration over all combinations of atom types which may result in large computation cost. Fortunately, as Vietoris–Rips filtration is often used to characterize the interaction networks, we only need to generate the filtered simplicial complex once for all atoms and perform persistent homology computation on the subcomplexes of the filtered simplicial complex.

II.A.4 Multi-level persistent homology. The Vietoris–Rips complex based only on pairwise distance is a widely used realization of filtration. When directly feeding the Euclidean distance between atoms to Rips complex construction, the interactions of interest such as electrostatic interactions can be flushed away by covalent bonds which usually have shorter lengths. This motivates us to incorporate a simple yet effective strategy to recover these important interactions by masking the original Euclidean distance matrix. Specifically, we keep only the entries corresponding to the interaction of interest and set every other entry to infinity in the distance matrix. For example, we set distances between atoms from the same component (protein or ligand) to infinity to focus on the interactions between the protein and ligand. This strategy was found especially useful when dealing with ligands alone which often have a much simpler structure than the proteins or the protein–ligand complexes. We call this approach to small molecules multi-level persistent homology of level n where we set the distance between two atoms to infinity if the shortest path between them through the covalent bond network is at most of the length n. This treatment has led to powerful predictive tools in tasks only explicitly involving small molecules.[49,92]

II.A.5 Physics-embedded persistent homology. All the topological methods discussed above are force-field-free approaches. In other words, they depend only on atomic coordinates and types without the need for molecular force field information. However, despite being insufficient, non-unique, and subject to errors, many biophysical models offer important approximations to the ground truth of biological science and reflect some of our best understandings of the biological world. Therefore, it is

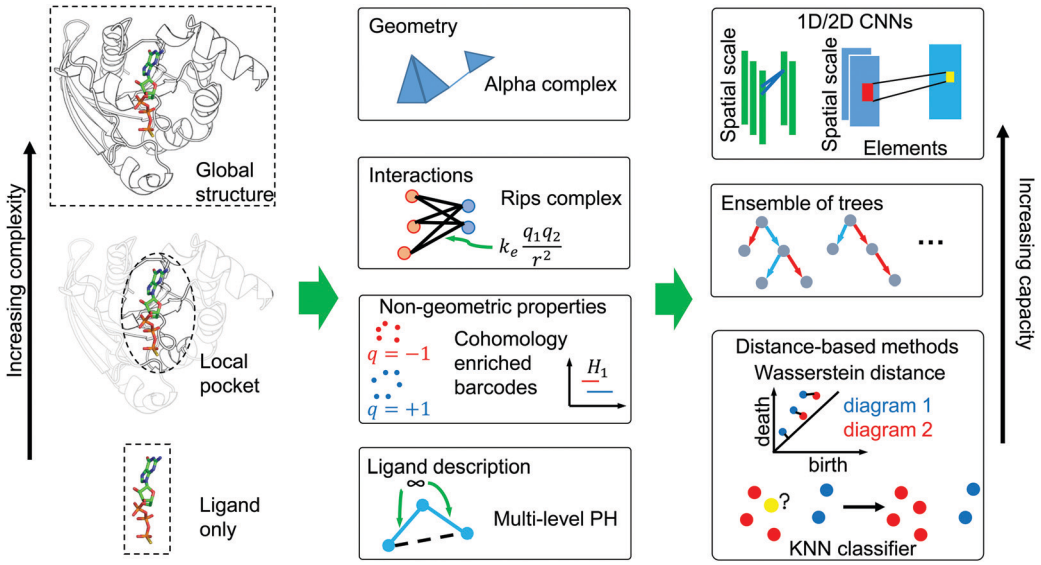

Fig. 6 Workflow of topology based protein–ligand binding affinity prediction. In multi-level persistent homology, the distance between covalent bonds are set to ∞ to avoid their disturbance to the topological representation of non-covalent bonds.

crucial to develop the so-called "physics-embedded" topology which incorporates physical models into topological invariants.

We are particularly interested in physical models that quantify the interaction strengths and directions. To characterize electrostatic interactions, we can construct a Rips filtration based on the Coulomb potential,

$$F_{\text{ele}}(i,j) = \frac{1}{1+\exp(-cq_iq_j/d_{ij})}, \tag{2}$$

where the filtration value $F_{\text{ele}}(i,j)$ for the edge between atoms i and j depends on their partial charges q_i and q_j and their geometric distance d_{ij}.[49] The part due to the Coulomb potential in eqn (2) can be substituted by other models, such as the van der Waals potential. We can also use cubical persistent homology[103] to characterize the charge density as volumetric data, for example, the one estimated from point charges,

$$\mu_c(\mathbf{r}) = \sum_i q_i \exp(-\|\mathbf{r}-\mathbf{r}_i\|/\eta_i), \tag{3}$$

where $\mathbf{r}_i$ is the position of atom i and η_i is a characteristic bond-length parameter.

In a more general setting, there often are available properties defined on the simplices in the simplicial complex representing the protein–ligand complex. The interaction strength characterized by physical models as in eqn (2) is indeed a property defined on the 1-simplices (edges). There are also various properties given on the 0-simplices (nodes/atoms) including atomic weight, atomic radii, and partial charges. Another way of incorporating these properties into the topological representation is to attach additional attributes to the persistence barcodes obtained through geometric filtration. We developed a method called enriched barcode through cohomology theory.[104] The usage of cohomology has led to efficient algorithms[105] as well as richer representations.[106] We are unable to elaborate on the details of cohomology here and the interested reader is referred to the aforementioned references.

Consider a persistence barcode $\{[b_i,d_i)\}_i \in I$ of dimension k obtained by a geometric based filtration of the molecular system, for example, the Vietoris–Rips filtration built upon the Euclidean distance between atoms in space. Let $K(x;k)$ be the set of k-simplices of the simplicial complex in the corresponding filtration with the filtration parameter x. Our goal is to annotate each persistence pair $[b_i,d_i)$ in the barcode with the non-geometric information provided by $f:K(\infty,k) \to \mathbb{R}$. We proposed to embed such non-geometric information *via* cohomology.[104] Specifically, for an $x \in [b_i,d_i)$, let $\omega_{i,x}$ be a real k-cocycle lifted from the representative cocycle from the persistent (co)homology computation.[106] A smoothed cocycle $\bar{\omega}_{i,x} = \bar{\alpha} + \omega_{i,x}$ can be obtained by solving the following problem,

$$\bar{\alpha} = \underset{\alpha \in C^{k-1}(K(x),\mathbb{R})}{\arg\min} \left\| \mathcal{L}(\omega_{i,x} + \mathrm{d}\alpha) \right\|_2^2, \tag{4}$$

where $\left(C^{k-1}(K(x),\mathbb{R})\right.$ is the real $(k-1)$-cochain on $K(x)$, d is the coboundary operator, and $\mathcal{L}$ is a Laplacian operator. This smoothed representative k-cocycle $\bar{\omega}$ annotates the simplices with weights which can be used to describe the non-geometric information on this persistence pair,

$$f_i^*(x) = \sum_{\sigma \in K(x;k)} f(\sigma)|\bar{\omega}_{i,x}(\sigma)| \Big/ \sum_{\sigma \in K(x;k)} |\bar{\omega}_{i,x}(\sigma)|. \tag{5}$$

Intuitively, this obtained function $f_i^*:[b_i,d_i) \to \mathbb{R}$ describes the average value of f near the k-dimensional hole associated with the persistence pair $[b_i,d_i)$. We call this object enriched barcode $\{\{[b_i,d_i),f_i^*\}\}_{i \in I}$.[104] In practice, we only compute for several filtration values in the interval or even only one such as the midpoint of each persistence pair.

II.A.6 From topological invariants to machine learning algorithms. While persistent homology already significantly reduces the complexity of the molecular system description, directly feeding it to machine learning algorithms can cause too

many model parameters compared to the moderate size of available data in this field. Also, the outputs of persistent homology are similar to unstructured point clouds. Additional processing is needed to integrate persistent homology characterization with machine learning models.

In the application to biomolecular structure description, prior knowledge is available on the approximate distance ranges for different interactions. Therefore, we first divide an interval $[0,D]$ where D is the longest range among the interactions of interest into bins. We then count the number of events in each bin, namely, (1) birth of persistence pairs, (2) death of persistence pairs, and (3) overlaps of bars with the bins. These approaches result in a 1-dimensional image-like feature tensor with three channels which can be fed into a 1-dimensional convolutional neural network or any other machine learning model that accepts structured features. Prior knowledge on the spatial range of different types of interactions can guide the decision of bin endpoints. We have also found similar performance with uniform partitioning. Another way of vectorization is to statistically describe the unstructured persistence barcodes, for example, the mean value and standard deviation of birth, death, and bar lengths.

The Wasserstein distance between the resulting persistence barcodes also works well with distance-based methods, such as k-nearest-neighbor-based regression and classification or k-means clustering. This approach was found effective especially when the objects are moderately complex. It has been successfully applied to ligand-based tasks.[49]

In general applications of integrating persistent homology with machine learning, the persistence barcodes can become sparse and available field knowledge might be insufficient to guide the vectorization. In this case, a neural network layer with each neuron learning a kernel function can automatically vectorize the barcodes. Specifically, one neuron in such a layer is a function that takes the persistence barcode $\mathcal{B} = \{[b_i, d_i]\}_{i\in I}$ and output a number,

$$\mathcal{N}(\mathcal{B};\Theta) = \sum_{i\in I} \phi(|b_i - \mu_b|, |d_i - \mu_d|; \Theta), \qquad (6)$$

where ϕ is a distance-based kernel function with learnable parameters Θ and the center (μ_b,μ_d). This layer can be the first layer in a neural network for supervised learning. This layer can also be used as the first layer of an autoencoder that tries to reconstruct the persistence barcodes controlled by the Wasserstein metric. On the other hand, kernel density estimators with a fixed number of kernels can also be used as a vectorization tool. Specifically, a kernel density estimator with n_k kernels each of which has n_p parameters to optimize can turn a persistence barcode into a feature vector of size $n_k \times n_p$. Treatment such as truncated kernels might be needed to take care of the nature of persistence barcodes that the points are only in the upper left part of the first quadrant.

II.B Differential geometry-based methods

II.B.1 Background. Differential geometry has a long history of development in mathematics and has been consistently studied since the 18th century. Nowadays, many differential geometry branches have been created from the Riemannian geometry, differential topology, to Lie groups. As a result, differential geometry has been used in various interdisciplinary fields including physics, chemistry, economics, and computer vision. In 2005, we unfolded a curvature-based model to generate biomolecular surfaces.[70] In the following years, we successfully formulated a Laplace–Beltrami operator based minimal molecular surface (MMS) for macromolecular systems.[71,72,107] This approach is applied to multiscale solvation modeling in which the molecular surfaces are described *via* the differential geometry of surfaces. Specifically, the solute molecule is still described in microscopic detail while the solvent is treated as a macroscopic continuum to reduce a large number of degrees of freedom.[76–79,83,84] Differential geometry-based multiscale models incorporate molecular dynamics, elasticity and fluid flow to further couple the discrete macromolecular and continuum solvent domains.[80–82,86,87] In the past few years, we have improved the computational efficiency of the geometric modeling by incorporating the differential geometry based multiscale paradigms in Lagrangian[73,74] and Eulerian representations.[75,108]

Differential geometry-based multiscale models have been used for solvation free energy prediction[79,93] and ion channel transport analysis[80–82,87,109] to demonstrate their model efficiency in comparison with atomistic scale models.

Another type of application of differential geometry in biomolecular systems is to utilize curvatures to characterize the macromolecular surface landscape and further infer chemical and biological properties. For example, the minimum and maximum curvatures are combined with the surface electrostatic potential to detect both positively charged and negatively charged protein binding sites.[75,108]

The other type of application of differential geometry in molecular science is to carry out curvature-based solvation free energy prediction.[85] In this approach, the total Gaussian, mean, minimum, and maximum curvatures of a molecule are computed for a molecule and correlated with its solvation free energy.

II.B.2 Challenge. Differential geometry based multiscale models bridge the discrete and continuum descriptions and enable physical interpretation of molecular mechanisms. Curvature-based modeling of biomolecular binding sites and solvation free energy reveals macromolecular interactive landscapes. These methods are designed as physical models to enhance our understanding of biomolecular systems. However, they have limited capability in predicting massive and diverse datasets due to their dependence on physical models such as the Poisson–Boltzmann equation or the Poisson–Nernst–Planck equation or their excessive reduction of geometric shape information, *i.e.*, a molecular-level average of local curvatures. Indeed, physical models depend on force field parameters which are subject to errors. Meanwhile, molecular-level descriptions are too coarse-gained for large datasets. In contrast, atomistic descriptions not only involve too much detail but also are not scalable for molecules with different sizes in a large dataset. As a result, machine learning algorithms cannot be effectively implemented.

To overcome these obstacles, we have designed new differential geometry-based models to extract element-level geometric information which automatically leads to scalable machine learning representations. Additionally, effort has been devoted to encoding intermolecular and intramolecular non-covalent interactions. Therefore, these novel models can be handily applied for diverse molecular and biomolecular datasets, including protein–ligand binding analysis and prediction.

II.B.3 Multiscale discrete-to-continuum mapping. Biomolecular datasets provide atomic coordinate and type information. To facilitate differential geometry modeling, this discrete representation is transformed into a continuum one by the so-called discrete-to-continuum mapping. In a given biomolecule or molecule with N atoms, denote $\mathbf{r}_j \in \mathbb{R}^3$ and q_j the position of the jth atom and its partial charge, respectively. For any point $\mathbf{r}$ in three-dimensional space, a discrete-to-continuum mapping[56,59,62] defines the molecular number/charge density as the following

$$\rho(\mathbf{r}, \{\eta_k\}, \{w_k\}) = \sum_{j=1}^{N} w_j \Phi\Big(\big\|\mathbf{r} - \mathbf{r}_j\big\|; \eta_j\Big), \tag{7}$$

In particular, the density ρ indicates the molecular number density when $w_j = 1$, and represents the molecular charge density when $w_j = q_j$. In addition, η_j describes characteristic distances, $\|\cdot\|$ is the second norm, and Φ with C^2 property satisfies the following admissibility conditions

$$\Phi(\|\mathbf{r} - \mathbf{r}_j\|;\eta_j) = 1, \quad \text{as} \quad \|\mathbf{r} - \mathbf{r}_j\| \to 0, \tag{8}$$

$$\Phi(\|\mathbf{r} - \mathbf{r}_j\|;\eta_j) = 0, \quad \text{as} \quad \|\mathbf{r} - \mathbf{r}_j\| \to \infty. \tag{9}$$

In principle, the density function can accept all radial basis functions (RBFs) as well as C^2 delta sequence of the positive type examined in this work.[110] In practice, the generalized exponential functions

$$\Phi\big(\big\|\mathbf{r}_i - \mathbf{r}_j\big\|; \eta_{kk'}\big) = \mathrm{e}^{-\left(\|\mathbf{r}_i - \mathbf{r}_j\|/\eta_{kk'}\right)^{\kappa}}, \quad \kappa > 0; \tag{10}$$

and generalized Lorentz functions

$$\Phi\big(\big\|\mathbf{r}_i - \mathbf{r}_j\big\|; \eta_{kk'}\big) = \frac{1}{1 + \left(\big\|\mathbf{r}_i - \mathbf{r}_j\big\|/\eta_{kk'}\right)^{\nu}}, \quad \nu > 0. \tag{11}$$

seem to be the most optimal choice for the biomolecular datasets.[56,59] Here power parameters κ and ν vary for different datasets and are systemically selected.

To generate the multiscale representation for $\rho(\mathbf{r},\{\eta_j\},\{w_j\})$, one can vary different values for scale parameters $\{\eta_j\}$. A published work[42] has shown that the molecular number density eqn (7) is an efficient representation for molecular surfaces. Unfortunately, such a molecular-level description has a small role in the predictive models for massive data.

II.B.4 Element interactive densities. To handle the diverse molecular or biomolecular datasets, we have upgraded differential geometry representations with an emphasis on non-covalent intramolecular interactions in a molecule and intermolecular interactions in complexes, such as protein–ligand, protein–nucleic acid, and protein–protein complexes. Also, our differential geometry features can characterize the geometric information at element-specific interactions and are scalable despite a wide range of molecular sizes.

To accurately encode the physical and biological information in the differential geometry representations, we describe the molecular interactions at the element-level in a systematic manner. For instance, in the protein–ligand datasets, the intermolecular interactions are decomposed into element-level descriptions based on the commonly occurring element type in proteins and ligands. Typically, protein structures usually consist of H, C, N, O, S, and ligand structures often include H, C, N, O, S, P, F, Cl, Br, I. That results in 50 element-level intermolecular descriptions. In practice, hydrogen atoms are missing in most Protein Data Bank (PDB) datasets for proteins. Therefore, we do not include them in our models for macromolecules or for both proteins and ligands. Finally, we end up with 40 or 36 element-specific groups to express the intermolecular interactions in the protein–ligand complexes. This element-specific approach can be straightforwardly carried out in other interactive systems in chemistry, biology and materials science. For example, in protein–protein interactions, one can similarly arrive at a total of 16 element-level descriptions for practical use.

In a given molecule, based on the most frequently appearing element types included in the set $\mathcal{C} = \{\mathrm{H}, \mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{S}, \mathrm{P}, \mathrm{F}, \mathrm{Cl}, \ldots\}$, we collect N atoms. For each jth atom in that collection, we label it as $\{(\mathbf{r}_j,\alpha_j,q_j)\}$. Here α_j is the element type of the jth atom, and $\alpha_j = \mathcal{C}_k$ indicates the kth element type in set $\mathcal{C}$.

Before defining the element interactive density, we have to designate the non-covalent interactions between two element types $\mathcal{C}_k$ and $\mathcal{C}_{k'}$. Such interactions can be represented by correlation kernel Φ

$$\begin{aligned}\{&\Phi(||\mathbf{r}_i - \mathbf{r}_j||;\eta_{kk'})|\alpha_i = \mathcal{C}_k, \alpha_j = \mathcal{C}_{k'};\\ &i,j = 1,2,\ldots,N; ||\mathbf{r}_i - \mathbf{r}_j|| > r_i + r_j + \sigma\},\end{aligned} \tag{12}$$

where r_i and r_j are the atomic radii of the ith and jth atoms, respectively, and σ is the mean value of the standard deviations of all r_i and r_j in the dataset. The inequality constraint $\|\mathbf{r}_i - \mathbf{r}_j\| > r_i + r_j + \sigma$ serves the purpose of excluding the covalent forces.

Given a point $\mathbf{r}$ in $\mathbb{R}^3$, we define the element interactive density induced by the pairwise interaction between two chemical element types $\mathcal{C}_k$ and $\mathcal{C}_{k'}$

$$\begin{aligned}\rho_{kk'}(\mathbf{r},\eta_{kk'}) = \sum_j w_j \Phi(||\mathbf{r} - \mathbf{r}_j||;\eta_{kk'}), \quad \mathbf{r} \in D_k, \alpha_j = \mathcal{C}_{k'};\\ ||\mathbf{r}_i - \mathbf{r}_j|| > r_i + r_j + \sigma, \forall \alpha_i \in \mathcal{C}_k; k \neq k',\end{aligned} \tag{13}$$

where D_k is the so-called atomic-radius-parametrized van der Waals domain given by the union of all the balls with centers at the C_k atomic positions with the corresponding atomic radius r_k. In other words, if $B(\mathbf{r}_i,r_i)$ is denoted as a ball with a center $\mathbf{r}_i$ and a radius r_i, D_k can be expressed as

$$D_k := \cup_{\mathbf{r}_i,\alpha_i=\mathcal{C}_k} B(\mathbf{r}_i, r_k). \tag{14}$$

Note that element interactive density represented in (13) is only good for $k \neq k'$. When density is calculated based on the

interactions between the same element types, *i.e.* $k \neq k'$, each C_k atom will belong to the atomic-radius-parametrized van der Waals domain and element interactive density representation. To this end, we define such density formulation as the following

$$\rho_{kk}(\mathbf{r},\eta_{kk}) = \sum_j w_j \Phi(||\mathbf{r}-\mathbf{r}_j||;\eta_{kk}), \quad \mathbf{r} \in D_k^i, \alpha_i = C_k; \alpha_j = C_k; ||\mathbf{r}_i - \mathbf{r}_j|| > 2r_j + \sigma, \tag{15}$$

in which domain D_k^i is just a single ball $B(\mathbf{r}_i,r_i)$, and the density function ρ_{kk} is evaluated at all D_k^i.

The element interactive density ρ_{kk} is the linear combination of correlation kernel Φ of pairs of element types. Consequently, the smoothness of ρ_{kk} is the same as that of Φ. Moreover, by changing a level constant c, one can attain a family of element interactive manifolds as

$$\rho_{kk'}(\mathbf{r},\eta_{kk'}) = c\rho_{\max}, \quad 0 \leq c \leq 1 \quad \text{and} \quad \rho_{\max} = \max\{\rho_{kk'}(\mathbf{r},\eta_{kk'})\}. \tag{16}$$

Fig. 7 illustrates a few element interactive manifolds.

II.B.5 Element interactive curvatures

Differential geometry of differentiable manifolds. We here describe the geometric information calculation on a differential manifold. Consider U being an open subset of $\mathbb{R}^n$ with its closure compact,[72,86,111] we are interested in a C^2 immersion $\mathbf{f}$: $U \to \mathbb{R}^{n+1}$. Given a vector $\mathbf{u} = (u_1,u_2,\ldots,u_n) \in U$, we express the Jacobian matrix with respect to $\mathbf{u}$ as

$$\mathbf{Df} = (X_1, X_2, \cdots, X_n), \quad X_i = \frac{\partial \mathbf{f}}{\partial u_i}, \ i = 1,2,\ldots,n. \tag{17}$$

The first fundamental form is written in the metric tensor with its coefficients $g_{ij} = \langle X_i, X_j \rangle$, where $\langle , \rangle$ is the Euclidean inner product in $\mathbb{R}^n$, $i, j = 1, 2, \ldots, n$.

We define the unit normal vector *via* the Gauss map

$$\mathbf{N}\text{: } U \to R^{n+1} \tag{18}$$

$$(u_1,u_2,\ldots,u_n) \mapsto X_1 \times X_2 \cdots \times X_n / \|X_1 \times X_2 \cdots \times X_n\|, \tag{19}$$

where "$\times$" denotes the cross product. If we denote $\perp_{\mathbf{u}}\mathbf{f}$ the normal space of $\mathbf{f}$ at point $\mathbf{X} = \mathbf{f}(\mathbf{u})$, then $\mathbf{N}(\mathbf{u}) \in \perp_{\mathbf{u}}\mathbf{f}$. In addition, one can form a second fundamental form *via* the means of the normal vector $\mathbf{N}$ and tangent vector X_i:

$$II(X_i, X_j) = (h_{ij})_{i,j=1,2,\ldots n} = \left(\left\langle -\frac{\partial \mathbf{N}}{\partial u_i}, X_j \right\rangle \right)_{ij}. \tag{20}$$

Then, the Gaussian curvature K and the mean curvature H are determined as the following

$$K = \frac{\mathrm{Det}(h_{ij})}{\mathrm{Det}(g_{ij})}, \quad H = \frac{1}{n} h_{ij} g^{ji}. \tag{21}$$

The Einstein summation convention is used in the curvature expressions and $(g^{ij}) = (g_{ij})^{-1}$.

Element interactive curvatures. With element interactive manifolds defined *via* element interactive density $\rho(\mathbf{r})$ described in (16) and the expressions in (21), one can further formulate the representations for the Gaussian curvature (K) and the

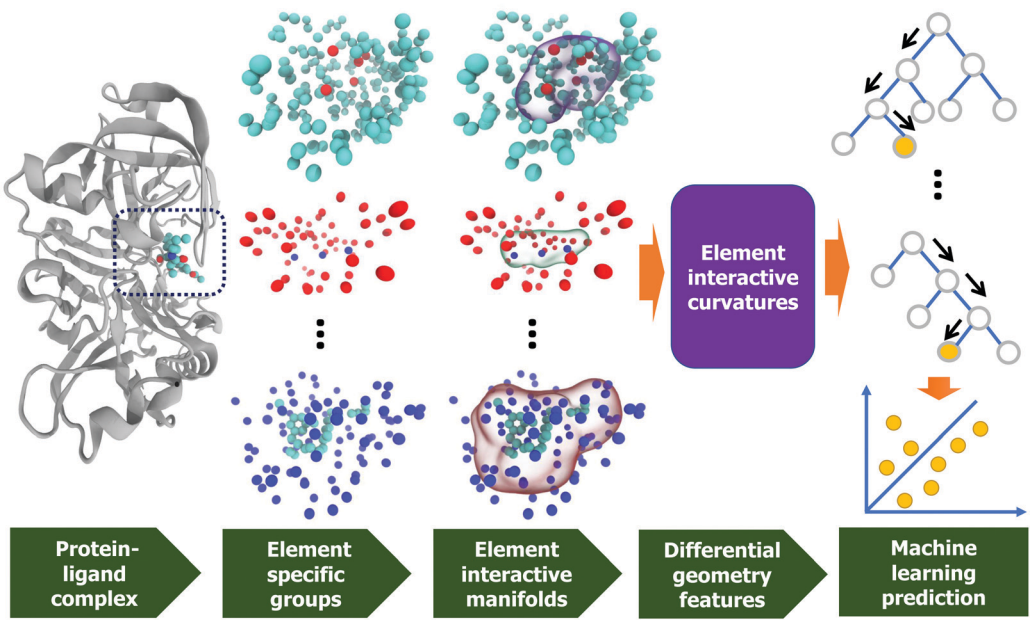

Fig. 7 Illustration of the DG-GL strategy for a complex with PDBID:5QCT (first column). The second column presents the different element specific groups including OC, CN, and CH, respectively, from top to bottom. The third column depicts the element interactive manifolds for the corresponding element specific groups. A predictive model in the last column integrates the differential geometry features (fourth column) with the machine learning algorithm.

mean curvature (H) as the following[75,112]

$$K = \frac{1}{g^2}\Big[2\rho_x\rho_y\rho_{xz}\rho_{yz} + 2\rho_x\rho_z\rho_{xy}\rho_{yz} + 2\rho_y\rho_z\rho_{xy}\rho_{xz} - 2\rho_x\rho_z\rho_{xz}\rho_{yy} - 2\rho_y\rho_z\rho_{xx}\rho_{yz} - 2\rho_x\rho_y\rho_{xy}\rho_{zz} + {\rho_z}^2\rho_{xx}\rho_{yy} + {\rho_x}^2\rho_{yy}\rho_{zz} + {\rho_y}^2\rho_{xx}\rho_{zz} - {\rho_x}^2{\rho_{yz}}^2 - {\rho_y}^2{\rho_{xz}}^2 - {\rho_z}^2{\rho_{xy}}^2\Big], \quad (22)$$

and

$$H = \frac{1}{2g^{\frac{3}{2}}}\Big[2\rho_x\rho_y\rho_{xy} + 2\rho_x\rho_z\rho_{xz} + 2\rho_y\rho_z\rho_{yz} - ({\rho_y}^2 + {\rho_z}^2)\rho_{xx} - ({\rho_x}^2 + {\rho_z}^2)\rho_{yy} - ({\rho_x}^2 + {\rho_y}^2)\rho_{zz}\Big], \quad (23)$$

where $g = {\rho_x}^2 + {\rho_y}^2 + {\rho_z}^2$.

In addition, the minimum curvature ($\kappa_{\min}$) and maximum curvatures ($\kappa_{\max}$) can be evaluated based on the Gaussian and mean curvature values

$$\kappa_{\min} = H - \sqrt{H^2 - K}, \quad \kappa_{\max} = H + \sqrt{H^2 - K}. \quad (24)$$

It is noted that in the curvature representations in (22)–(24), the derivatives of the density function can be analytically calculated. For convenience, we denote the curvatures associated with the density function $\rho_{kk'}(\mathbf{r},\eta_{kk'})$ as $K_{kk'}(\mathbf{r},\eta_{kk'})$, $H_{kk'}(\mathbf{r},\eta_{kk'})$, $\kappa_{kk',\min}(\mathbf{r},\eta_{kk'})$, and $\kappa_{kk',\max}(\mathbf{r},\eta_{kk'})$. In practical use, the element interactive curves are only evaluated at the atomic positions in a given molecule or biomolecule structure. Notice that, due to the variant sizes in different biomolecular structures, numbers of selected atoms for the curvature evaluations vary. To achieve element-level geometry information, we propose the element interactive mean curvature as the following

$$H^{\mathrm{EI}}_{kk'}(\eta_{kk'}) = \sum_i H_{kk'}(\mathbf{r}_i,\eta_{kk'}), \quad \mathbf{r}_i \in D_k; k \neq k' \quad (25)$$

and

$$H^{\mathrm{EI}}_{kk}(\eta_{kk}) = \sum_i H_{kk}(\mathbf{r}_i,\eta_{kk}), \quad \mathbf{r}_i \in D^i_k, D^i_k \subset D_k. \quad (26)$$

The other element-level interactive curvatures for Gaussian curvature $\left(K^{\mathrm{EI}}_{kk'}(\eta_{kk'})\right)$, minimum curvature $\left(\kappa^{\mathrm{EI}}_{kk',\min}(\eta_{kk'})\right)$, and maximum curvature $\left(\kappa^{\mathrm{EI}}_{kk',\max}(\eta_{kk'})\right)$ are defined in a similar manner.

II.B.6 Differential geometry based geometric learning (DG-GL)

Geometric learning. In our differential geometry based geometric learning (DG-GL) model, we incorporate the geometric representations such as element-level interactive curvatures with advanced machine learning algorithms to form powerful predictive models. Given a training set $\{\mathcal{X}_i\}_{i\in I}$, in which $\mathcal{X}_i$ is the input data for the ith molecule and I is the set of the molecular indices in the training data, we denote $\mathbf{F}(\mathcal{X}_i;\zeta)$ as differential geometric functions encoding the input structures $\mathcal{X}_i$ *via* the given hyperparameter set ζ into aforementioned DG descriptions. Our DG-GL model learns the training set $\{\mathcal{X}_i\}_{i\in I}$ by minimizing the following loss functions

$$\min_{\zeta,\theta} \sum_{i\in I} L(\mathbf{y}_i, \mathbf{F}(\mathcal{X}_i;\zeta);\theta), \quad (27)$$

in which L is the loss function, $\mathbf{y}_i$ is the target label of molecule $\mathcal{X}_i$, and θ is the set of parameters of a selected machine learning algorithm. It is worth noting that the DG representation encoded in $\mathbf{F}$ does not depend on the type of learning task. Therefore, our DG-GL models can adapt any regressor or classifier models such as linear regression, support vector machine, random forest, gradient boosting trees, artificial neural networks, and convolutional neural networks. Besides the machine learning hyperparameters, the kernel parameters in the encoding DG function $\mathbf{F}$ need to be optimized for a specific learning algorithm and a particular training set $\{\mathcal{X}_i\}$.

In the validation, we only utilize the gradient boosting trees (GBTs) even though the other advanced machine learning models including convolutional neural networks can be incorporated with minimal effort. The general framework of the DG-GL model is depicted in Fig. 7. The GBTs in the DG-GL score are employed *via* the gradient boosting regression module in the scikit-learn v0.19.1 package with the following hyperparameters: n_estimators = 10 000, max_depth = 7, min_samples_split = 3, learning_rate = 0.01, loss = ls, subsample = 0.3, max_features = sqrt for all experiments.

Model parametrization. In our differential geometry-based approach, we calculate the element interactive curvatures (EICs) of type C based on kernel α with parameters (δ,τ). We denote such a model as $\mathrm{EIC}^{C}_{\alpha,\delta,\tau}$. Here, $C \in \{K,H,k_{\min},k_{\max}\}$ and α = E and α = L indicate generalized exponential and generalized Lorentz kernels, respectively. In addition, δ refers to the kernel order and is denoted as κ if α = E or ν if α = L. Another kernel parameter is τ defined by the following relationship

$$\eta_{kk'} = \tau(\bar{r}_k + \bar{r}_{k'}) \quad (28)$$

where $\bar{r}_k$ and $\bar{r}_{k'}$ stand for the van der Waals radii of element type k and element type k', respectively. These kernel parameters are selected *via* a 5-fold cross-validation on a specific training set with the range of τ and δ varying from 0.5 to 6 with an increment of 0.5. Moreover, we are interested in high values of power order, $\delta \in \{10,15,20\}$, which accounts for the ideal low-pass filter (ILF).[63] These parameter ranges are also listed in Table 1.

To enable the multiscale descriptions in differential geometry representation, we employ multiple kernels to evaluate the EICs. For instance, if two kernels with the following

Table 1 The ranges of DG-GL hyperparameters for 5-fold cross-validations

Parameter	Domain
τ	$\{0.5,1.0,\ldots,6\}$
δ	$\{0.5,1.0,\ldots,6\} \cup \{10,15,20\}$
C	$\{K,H,k_{\min},k_{\max}\}$

parameters (α_1,δ_1,τ_1) and (α_2,δ_2,τ_2) are utilized, our EIC model can be written as $\mathrm{EIC}^{C_1C_2}_{\alpha_1,\delta_1,\tau_1;\alpha_2,\delta_2,\tau_2}$.

In a protein–ligand complex, we are interested in 4 commonly occurring protein atom types {C,N,O,S}, and 10 commonly occurring ligand atom types {H,C,N,O,F,P,S,Cl,Br,I}. That results in a total of 40 different combinations. With a set of calculated atomic pairwise curvatures, we construct 10 statistical features, namely, sum, the sum of absolute values, minimum, the minimum of absolute values, maximum, the maximum of absolute values, mean, the mean of absolute values, standard deviation, and the standard deviation of absolute values. In total, we attain 400 features for the current differential geometry-based models.

II.C Graph theory-based methods

II.C.1 Background. Graph theory is one of the most popular subjects in discrete mathematics. In graph theory, the information inputs are represented in graph structures formed by vertices that are connected by edges and/or high-dimensional simplexes. Different ways are there to interpret the graph result in different graph theories such as geometric graph theory, algebraic graph theory, and topological graph theory. In geometric graph study, the graph information is extracted based on the geometric objects drawn in the Euclidean plane.[113] If there are algebraic methods involved in graph structure processing, that approach belongs to algebraic graph theory. There are two common approaches to this branch. The first one is to use linear algebra to study the spectrum of various types of matrices representing the graph including the adjacency matrix and Laplacian matrix.[114] Another approach relies on group theory, especially automorphism groups[115] and geometric group theory,[116] for the study of graphs. Unlike the aforementioned graph theories, the algebraic graph theory considers graphs as topological spaces by associating with different types of simplicial complexes such as abstract simplicial complexes[117] and Whitney complexes.[118]

Due to the natural representations for structured information, graph theory finds enormous applications in various fields including computer science, linguistics, physics, chemistry, biology, and social sciences. Particularly in chemical and biological studies, graph theory is commonly used since molecular structures always feature graph information in which vertices illustrate atoms and graph edges represent bonds. Indeed, graph-based approaches have been utilized to describe chemical datasets[119–124] as well as biomolecular datasets.[54,125–130] In addition, one can make use of graph representations to uncover the connectivity of different components of a molecule such as centrality,[131–133] contact map,[54,134] and topological index.[123,135] Moreover, graph extracting representations can be employed in chemical analysis[52,120,121] and biomolecular modeling.[136] Particularly, some research groups have invested their efforts to carry out the graph-based representation to model protein flexibility and long-time dynamics such as the normal-mode analysis (NMA)[137–140] and elastic network model (ENM).[54,55,141–144]

II.C.2 Challenge. Due to the richness in geometric interpretations, graph theory-based approaches have shown their efficiency in the qualitative and descriptive models. However, oversimplified representations and the lack of physical and biological detailed information may render graph theory-based approaches less attractive in the quantitative analysis. For instance, in the Gaussian network model (GNM),[54,142,145] the use of the spectrum of the Laplacian matrix is quite efficient to decompose the flexible and rigid regions and domains of proteins but its fluctuation predictions on protein C_α atoms were not reliable with the Pearson correlation coefficient as low as 0.6 for three datasets.[146] To predict the mutations in proteins, the graph-based mCSM method was not competent as physical and knowledge-based or topological fingerprint-based models.[46,147]

The poor performances of the aforementioned graph theory-based models on quantitative tasks are due to the lack of three main components in our point of view. Firstly, these graph theory-based structures do not provide the information at the chemical element level. Consequently, these models treat different element types equally which results in inadequate coded information from the original structures. Secondly, non-covalent interactions between two atoms are overlooked in many graph edges which cause the unphysical representations for most molecular and biomolecular data. Finally, the edges in the many graph-based models express the connectivity between a pair of atoms based on the number of covalent bonds between these two atoms, which inaccurately describe many interactions that depend on the Euclidean distance.

To address the aforementioned issues in graph based-modeling, we have developed the weighted graphs, termed as the flexibility–rigidity index (FRI), to predict the *B*-factor of protein atoms. In our FRI model, the graph edges were formulated by the radial basis functions (RBFs)[58–60,62] which properly describe the interaction strengths between two atoms in the equilibrium structures. The original FRI was upgraded to multiscale FRI[60,63] for capturing the multiscale interactions in biological structures. Specifically, the graph in the multiscale FRI model is allowed to have multiple edges formed by RBFs with careful selections of scaled and power parameters. Although our FRI models have outperformed the GNM in *B*-factor predictions, they provide only coarse-grained molecular-level descriptions. To overcome this limitation, we have proposed graph coloring based methods with vertices colored differently based on the corresponding element types. Consequently, we ended up with various element-specific subgraphs taking care of different types of physical interactions, such as hydrophilic, hydrophobic, hydrogen bonds.[64,65] As a result, the predicted accuracy for protein *B*-factors by our multiscale weighted colored graphs is over 40% higher than GNM models.[65] The success of multiscale weighted colored graph models on *B*-factor prediction encouraged us to design graph-based scoring functions to predict protein–ligand binding affinities. The protein–ligand binding mechanism is more complex than the protein *B*-factor. Therefore, it requires sophisticated graph-based models to accurately encode the physical and biological properties to unveil its molecular mechanism. The development of such graphs is described in the following sections.

II.C.3 Multiscale weighted colored geometric subgraphs. In this section, we discuss general graph representations for a

molecule or biomolecule. Graph-based representations are systematic, scalable, and straightforward applied not only to the predictions of protein–ligand binding affinity but also for various bioactivities such as toxicity, solvation, solubility, partition coefficients, mutation-induced protein folding stability changes, and protein–nucleic acid interactions. In a given molecule or biomolecule in a dataset, we denote a graph $\mathcal{G}$ to represent a subset of its N atoms. The set of its vertices $\mathcal{V}$ consists of coordinates and chemical element types of atoms, defined as

$$\mathcal{V} = \{(\mathbf{r}_j, \alpha_j)|\mathbf{r}_j \in \mathbb{R}^3; \alpha_j \in \mathcal{C}; j = 1, 2, \ldots, N\}, \tag{29}$$

where $\mathbf{r}_j$ is the 3D position of the jth atom, and α_j is its element type which belongs to a predefined set of commonly occurring chemical element types as introduced in Section II.B.4. To accomplish meaningful encoded physical and biological information in the graph, graph edges have to express the non-covalent interactions. Moreover, to accommodate for the interactions between k element atoms and k' element type atoms, we consider a set of graph edges $\mathcal{E}_{kk'}$ represented by RBFs as the following

$$\begin{aligned}\mathcal{E}_{kk'} = \{&\Phi(||\mathbf{r}_i - \mathbf{r}_j||; \eta_{kk'})|\alpha_i = \mathcal{C}_k, \alpha_j = \mathcal{C}_{k'}; \\ &i, j = 1, 2, \ldots, N; ||\mathbf{r}_i - \mathbf{r}_j|| > r_i + r_j + \sigma\},\end{aligned} \tag{30}$$

where $\|\mathbf{r}_i - \mathbf{r}_j\|$ accounts for the Euclidean distance between the ith and jth atoms, r_i and r_j are the atomic radii of ith and jth atoms, respectively. Moreover, σ is the mean value of the standard deviations of all atomic radii belonging to element types $\mathcal{C}_k$ and $\mathcal{C}_{k'}$ in the dataset. The exclusion of the covalent interactions are portrayed in this inequality $\|\mathbf{r}_i - \mathbf{r}_j\| > r_i + r_j + \sigma$. Φ is a predefined RBF representing a graph weight and has the following properties[56,59]

$$\Phi(||\mathbf{r}_i - \mathbf{r}_j||; \eta_{kk'}) = 1, \text{ as } ||\mathbf{r}_i - \mathbf{r}_j|| \to 0 \quad \text{and} \tag{31}$$

$$\Phi(||\mathbf{r}_i - \mathbf{r}_j||; \eta_{kk'}) = 0 \text{ as } ||\mathbf{r}_i - \mathbf{r}_j|| \to \infty, \quad \alpha_i = \mathcal{C}_k, \alpha_j = \mathcal{C}_{k'}, \tag{32}$$

where $\eta_{kk'}$ is a characteristic distance between the atoms. We now achieve the weight colored subgraphs (WCS) $\mathcal{G}(\mathcal{V}, \mathcal{E}_{kk'})$ or denote $\mathcal{G}_{kk'}$ for short.

In principle, our WCS $\mathcal{G}(\mathcal{V}, \mathcal{E}_{kk'})$ can adopt any RBFs. In practice, the generalized exponential functions (10) and generalized Lorentz functions (11) seem to be the most optimal choice for the biomolecular datasets.[56,59] Here power parameters κ and ν vary for different datasets and are systemically selected. To illustrate WCS of a given molecule, we use the uracil compound ($C_4H_4N_2O_2$) as an example. Fig. 8 depicts WCS for nitrogen and oxygen atoms ($\mathcal{G}_{NO}$). To elicit the geometrical invariants of WCS formed by element types $\mathcal{C}_k$ and $\mathcal{C}_{k'}$, we propose a collective representation at the element level as follows

$$\begin{aligned}\mathrm{RI}^{\mathrm{G}}(\eta_{kk'}) = \sum_i \mu_i^{\mathrm{G}}(\eta_{kk'}) = \sum_i \sum_j \Phi(||\mathbf{r}_i - \mathbf{r}_j||; \eta_{kk'}), \quad \alpha_i = \mathcal{C}_k, \\ \alpha_j = \mathcal{C}_{k'}; ||\mathbf{r}_i - \mathbf{r}_j|| > r_i + r_j + \sigma,\end{aligned} \tag{33}$$

where $\mu_i^{\mathrm{G}}(\eta_{kk'})$ which is a geometric subgraph centrality for the ith atom has been developed in our previous work for protein B-factor predictions.[65] The summation over $\mu_i^{\mathrm{G}}(\eta_{kk'})$ in eqn (33) gives rise to WCS rigidity between element types $\mathcal{C}_k$ and $\mathcal{C}_{k'}$. In fact, $\mu_i^{\mathrm{G}}(\eta_{kk'})$ is the generalized form of our successful rigidity index model for protein–ligand binding affinity prediction in the previous work.[64] It is noticed that the WCS for the protein–ligand system is bipartite since each of its edges presents the interaction between one atom in the protein and another protein in the ligand. With that design, a variety of physical and biological properties such as electrostatics, van der Waals interactions, hydrogen bonds, polarization, hydrophilicity, and hydrophobicity can be successfully encoded in our WCS representations.

To exhibit the intermolecular and intramolecular properties, one can vary the characteristic distance $\eta_{kk'}$ to set up multiscale weighted colored subgraphs (MWCS). To methodically attain multiscale graph-based molecular and biomolecular representations in a collective and scalable manner, one can aptly select groups of pairwise element interactions k and k', the choice of subgraph weights Φ and their parameters.

II.C.4 Multiscale weighted colored algebraic subgraphs. In this section, we present another approach to extract the meaningful representation for biomolecules from their WCS. This scheme depends on the algebraic graph or spectral graph formulations. Since geometric and algebraic approaches handle the graph information differently, these two kinds of subgraphs will be expected to encode the physical and biological information in varied aspects. In the algebraic graph theory, matrices are utilized to represent a given subgraph. Two of the most common ones are the Laplacian matrix and the adjacency matrix.

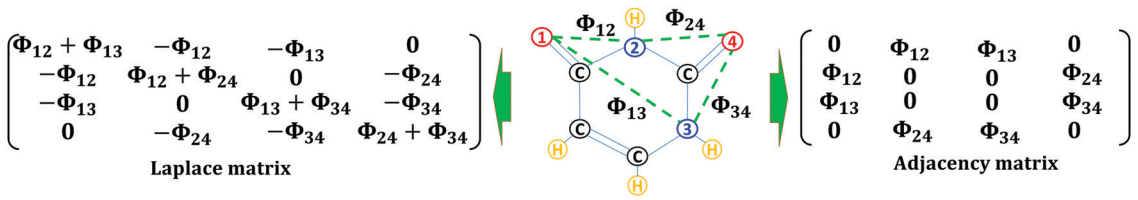

Fig. 8 Illustration of weighted colored subgraph $\mathcal{G}_{NO}$ (left), its Laplacian matrix (middle), and adjacency matrix (right) for uracil molecules ($C_4H_4N_2O_2$). Graph vertices, namely oxygen (*i.e.*, atoms 1 and 4) and nitrogen (*i.e.*, atoms 2 and 3), are labeled in red and blue colors, respectively. Here, graph edges (*i.e.*, Φ_{ij}) are labeled by green-dashed lines which are not covalent bonds. Here, Φ_{ij} are distance-weighted edges. Note that there are 9 other nontrivial subgraphs for this molecule (*i.e.*, $\mathcal{G}_{CC}, \mathcal{G}_{CN}, \mathcal{G}_{CO}, \mathcal{G}_{CH}, \mathcal{G}_{NN}, \mathcal{G}_{NH}, \mathcal{G}_{OO}, \mathcal{G}_{OH}, \mathcal{G}_{HH}$).

Multiscale weighted colored Laplacian matrix. Considering a weighted colored subgraph $\mathcal{G}(\mathcal{V},\mathcal{E}_{kk'})$ defined at eqn (29) and (30), we construct the following weighted colored Laplacian matrix $L(\eta_{kk'}) = (L_{ij}(\eta_{kk'}))$ describing the interaction between element types $\mathcal{C}_k$ and $\mathcal{C}_{k'}$

$$L_{ij}(\eta_{kk'}) = \begin{cases} -\Phi(||\mathbf{r}_i - \mathbf{r}_j||;\eta_{kk'}) & \text{if } i\neq j, \alpha_i = \mathcal{C}_k, \alpha_j = \mathcal{C}_{k'} \text{ and } \left\|\mathbf{r}_i - \mathbf{r}_j\right\| > r_i + r_j + \sigma; \\ -\sum\limits_j L_{ij} & \text{if } i = j. \end{cases} \tag{34}$$

For illustration, we explicitly formulate the Laplacian matrix of the WCS $\mathcal{G}_{\mathrm{NO}}$ for the uracil molecule ($C_4H_4N_2O_2$) in Fig. 8. It is obvious to learn that all eigenvalues of our element-level WCS Laplacian matrix are nonnegative due to its symmetric, diagonally dominant, and positive-semidefinite properties. Moreover, every row sum and column sum of $L(\eta_{kk'})$ is zero. In consequence, its first eigenvalue is 0. The second smallest eigenvalue of $L(\eta_{kk'})$ is so-called algebraic connectivity (also known as a Fiedler value) which approximates the sparest cut of a graph. With a given WCS $\mathcal{G}(\mathcal{V},\mathcal{E}_{kk'})$ one can easily see that its geometrical invariant proposed at eqn (33) is fully recovered in the trace of its Laplacian matrix $L(\eta_{kk'})$

$$\mathrm{RI}^{\mathrm{G}}(\eta_{kk'}) = \mathrm{Tr}\, L(\eta_{kk'}), \tag{35}$$

where Tr is the trace.

In the algebraic graph, we are interested in using the eigenvalue and eigenvector information to extract the graph invariants. To this end, we denote $\lambda_j^{\mathrm{L}}, j = 1, 2,\ldots$ and $\mathbf{u}_j^{\mathrm{L}}, j = 1, 2,\ldots$ the eigenvalues and eigenvectors of $L(\eta_{kk'})$. The element-level molecular representations of the Laplacian matrix $L(\eta_{kk'})$ is proposed as the following

$$\mathrm{RI}^{\mathrm{L}}(\eta_{kk'}) = \sum_i \mu_i^{\mathrm{L}}(\eta_{kk'}), \tag{36}$$

where $\mu_i^{\mathrm{L}}(\eta_{kk'})$ is a so-called atomic representation for the ith atom ($\mathbf{r}_i, \alpha_i = \mathcal{C}_k$)

$$\mu_i^{\mathrm{L}}(\eta_{kk'}) = \sum_l (\lambda_l^{\mathrm{L}})^{-1}\left[\mathbf{u}_l^{\mathrm{L}}(\mathbf{u}_l^{\mathrm{L}})^{\mathrm{T}}\right]_{ii}, \tag{37}$$

where T is the transpose. It is noted that $\mu_i^{\mathrm{L}}(\eta_{kk'})$ is the atomic representation of the generalized GNM.[54,63] Therefore, it can be directly utilized to capture atomic properties such as protein B-factors. Moreover, the element-level invariant of the Laplacian matrix can be enriched *via* the statistical information of $\mu_i^{\mathrm{L}}(\eta_{kk'})$ values, namely, sum, mean, maximum, minimum and standard deviation.

Another way to extract the invariant representation from the WCS Laplacian matrix is the direct use of nontrivial eigenvalues $\{\lambda_j^{\mathrm{L}}\}_{j=2,3,\ldots}$. Also, the statistical analysis of those eigenvalues can be incorporated to form a feature vector to characterize element-level information of the molecule and biomolecule.

Multiscale weighted colored adjacency matrix. By setting all diagonal entities of the Laplacian matrix to be 0, we end up with an adjacency matrix with simpler representation but still preserve the essential properties of the original molecular structures. With a given WCS $\mathcal{G}_{kk'}$, the adjacency matrix $A(\eta_{kk'}) = (A_{ij}(\eta_{kk'}))$ is given as

$$A_{ij}(\eta_{kk'}) = \begin{cases} \Phi(||\mathbf{r}_i - \mathbf{r}_j||;\eta_{kk'}) & \text{if } i\neq j, \alpha_i = \mathcal{C}_k, \alpha_j = \mathcal{C}_{k'} \text{ and } \parallel \mathbf{r}_i - \mathbf{r}_j \parallel > r_i + r_j + \sigma; \\ 0 & \text{if } i = j. \end{cases} \tag{38}$$

Since the adjacency matrix defined in (38) is undirected, $A(\eta_{kk'})$ is symmetric. Thus, all the eigenvalues of it are real. Moreover, due to being a bipartite graph, for each eigenvalue λ, its opposite $-\lambda$ is also an eigenvalue of $A(\eta_{kk'})$. In consequence, only positive eigenvalues are used in the molecular representation. For the sake of illustration, Fig. 8 illustrates the adjacency matrices for the weighted colored subgraph G_{NO} in the uracil molecule ($C_4H_4N_2O_2$). It can be seen from the Perron-Frobenius theorem that the spectral radius of $A(\eta_{kk'})$, denoted as $\rho(A)$, is bounded by the range of the diagonal elements of the corresponding Laplacian matrix

$$\min_i \sum_j A_{ij} \leq \rho(A) \leq \max_i \sum_j A_{ij}. \tag{39}$$

It is easy to see that all elements in the Laplacian matrix belong to [0,1] and depends on the scale parameter $\eta_{kk'}$. In a characteristic scale range for capturing hydrogen bonds or van der Waals interactions, the Laplacian matrix has many zeros. However, the scale parameter $\eta_{kk'}$ can be very huge in electrostatic and hydrophobic interactions,[47] which results in many elements in the Laplacian matrix being nearly 1. In that particular situation, the spectral radius of the adjacency matrix $A(\eta_{kk'})$ is bounded by $n - 1$, where n is the number of atoms in WCS $\mathcal{G}_{kk'}$.

Similarly to the approach of forming feature representation for the Laplacian matrix, all positive eigenvalues $\{\lambda_j^{\mathrm{A}}\}$, and their statistical information such as sum, mean, maximum, minimum, and standard deviation are included in element-level molecular representations. If we define $\{\mathbf{u}_j^{\mathrm{A}}\}$ as the eigenvectors corresponding to eigenvalues $\{\lambda_j^{\mathrm{A}}\}$, then the atomic representations can be attained as

$$\mu_i^{\mathrm{A}}(\eta_{kk'}) = \sum_j \left[Q\Lambda Q^{-1}\right]_{ij}, \tag{40}$$

where $Q = [\mathbf{u}_1^{\mathrm{A}}\mathbf{u}_2^{\mathrm{A}}\ldots\mathbf{u}_n^{\mathrm{A}}]$ is composed of n linearly independent eigenvectors of $A(\eta_{kk'})$; thus Q is invertible. Moreover, Λ is a diagonal matrix with each diagonal element Λ_{ii} being the eigenvalue $\{\lambda_i^{\mathrm{A}}\}$. Unfortunately, the formulation given in eqn (40) is very computationally expensive due to the involvement of the inverse-matrix calculation.

In general, the methods regarding the analysis of eigenvalues and eigenvectors often pose a great challenge for sustaining an efficient computation strategy. Fortunately, the construction of WCS enables us to design a less-expensive computational model due to two facts. First, the protein–ligand binding site only involves a small region of the whole complex structure. Second, WCS only admits the specific element types in the matrix construction, which further reduces the size of matrices for eigenvalue and eigenvector calculations. As a result, these

facts offer an efficient spectral graph-based model for protein–ligand affinity analysis.

II.C.5 Graph-based learning models

Graph learning. The eigenvalue related information obtained from the algebraic graph approach is incorporated with machine learning algorithms to form predicting models for molecular and biomolecular properties. Depending on the nature of each learning task, regressor or classifier algorithms will be utilized. To illustrate the learning process, we denote $\mathcal{X}_i$ the ith structure in the training data and denote $\mathbf{G}(\mathcal{X}_i;\zeta)$ a function representing the graph information of sample $\mathcal{X}_i$ with respect to kernel parameters ζ. Generally, during the training process, machine learning models will minimize the following loss

$$\min_{\zeta,\theta} \sum_{i\in I} \mathcal{L}(\mathbf{y}_i, \mathbf{G}(\mathcal{X}_i;\zeta);\theta), \tag{41}$$

where $\mathcal{L}$ is the loss function, and $\mathbf{y}_i$ indicates the training labels. In addition, θ is the machine learning parameter. In principle, the set of parameters θ will be optimized for a specific training set and the choice of a machine learning algorithm. With the current graph presentations, one can make use of advanced machine learning models such as random forest (RF), gradient boosting trees (GBTs), deep learning neural networks to minimize the loss function $\mathcal{L}$. To illustrate the performance of our graph-based model, we employ GBTs for a balance between accuracy and complexity. The flow chart of the proposed model is illustrated in Fig. 9.

All the experiments in this graph learning task are carried out by the Gradient Boosting Regressor module implemented in the scikit-learn v0.19.1. The detailed parameters are given as n_estimators = 10 000, max_depth = 7, min_samples_split = 3, learning_rate = 0.01, loss = ls, subsample = 0.3, and max_features = sqrt. That parameter selection is nearly optimal and is the same for all calculations.

Model parametrization. Avoiding the wording, this notation $\mathrm{AGL}^{\mathcal{M}}_{\Omega,\delta,\tau}$ represents the AGL-Score features encoded based on the interactive matrix type along with kernel type Ω and kernel parameters δ and τ. Furthermore, $\mathcal{M}=\mathrm{Adj}$, $\mathcal{M}=\mathrm{Lap}$, and $\mathcal{M}=\mathrm{Inv}$ represent the adjacent matrix, the Laplacian matrix, and the pseudoinverse of the Laplacian matrix, respectively. In the kernel type notation, $\Omega=\mathrm{E}$ and $\Omega=\mathrm{L}$, respectively, indicate the generalized exponential kernel and generalized Lorentz kernels. Since the kernel order notation depends on the specific kernel type, we denote $\delta=\kappa$ if $\Omega=\mathrm{E}$, and $\delta=\nu$ if $\Omega=\mathrm{L}$. Lastly, the scale factor τ implicitly implies the expression $\eta_{kk'}=\tau(\bar{r}_k+\bar{r}_{k'})$, in which $\bar{r}_k$ and $\bar{r}_{k'}$ are the van der Waals radii of element type k and element type k', respectively.

In the multiscale representation for the AGL-Score, we naturally extend the single-scale notation. Only at most two different kernels are carried out in the AGL-Score model, and the resulting model is denoted as $\mathrm{AGL}^{\mathcal{M}_1,\mathcal{M}_2}_{\Omega_1,\delta_1,\tau_1;\Omega_2,\delta_2,\tau_2}$.

To achieve the optimal parameter selection in the AGL-Score's kernels, we perform 5-fold cross-validation (CV) on the training data of the benchmark. Ideally, one needs to revise the machine learning model for different problem settings. To demonstrate the robustness of our graph-based features, we only train the AGL-Score's parameters on the CASF-2007 benchmark with a training data size of 1105 complexes. Similar to our previous work, we select the range of the graph-based model's hyperparameters as demonstrated in Table 2. The ranges of AGL's kernel parameters are selected similarity to the ones in DG-GL

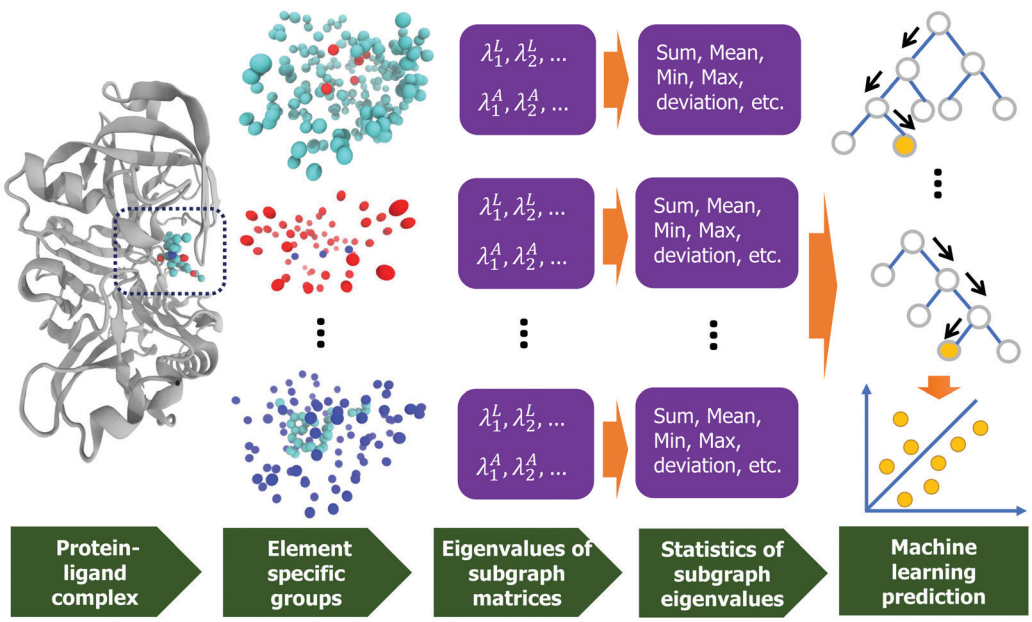

Fig. 9 A paradigm of the graph-based approach. The first column is the complex input with PDBID 5QCT. The second column illustrates the element-specific groups in the binding site. The third column presents the eigenvalues of the corresponding weighted colored graph Laplacian and adjacency matrices in the second column. The statistics of these eigenvalues are calculated in the fourth column. The final column forms a gradient boosting trees model using these eigenvalues.

Table 2 The ranges of AGL hyperparameters for 5-fold cross-validations

Parameter	Domain
τ	{0.5,1.0,...,6}
δ	{0.5,1.0,...,6}∪{10,15,20}
$\mathcal{M}$	{Adj,Lap,Inv}

models discussed in Section II.B.6. For the CASF benchmark datasets, we take into account 4 atom types in protein, namely, {C,N,O,S}, and 10 atom types in the ligand, namely, {H,C,N,O,F,P,S,Cl,Br,I}, which results in 40 different atom-pairwise combinations. Due to having opposite eigenvalues in the adjacency matrix, we only consider its positive eigenvalues. Moreover, the statistical properties of these eigenvalues such as sum, minimum (*i.e.*, the Fiedler value for Laplacian matrices or the half band gap for adjacency matrices), maximum, mean, median, standard deviation, and variance are collected. Moreover, the number of distinct eigenvalues, as well as the summation of their second power, are calculated. Finally, we form a representation vector of 360 features.

II.D Machine learning algorithms

It is generally true that our mathematical representations can be paired with any machine learning model. However, the devil is in the details: different machine learning algorithms respond differently to data size, representation dimension, representation noise, representation correlation, representation amplitude, and representation distribution. Therefore, it is useful to design learning-model adapted mathematical representations.

In the past few years, we have integrated various mathematical representations with a variety of machine learning algorithms, namely, *k*-nearest neighbors (KNNs),[26,49] learning to rank (LR),[25,27] support vector machine (SVM),[43] gradient boosted decision trees (GBDT),[46,47] random forest (RF),[64,92,94] extra-trees (ET),[49] deep artificial neural network (ANN),[92,94] deep convolutional neural network (CNN),[48,49] multitask ANN,[92,94] multitask CNN,[48] and generative networks.[148]

Due to the extensive variability in the possible types of biological tasks and machine learning algorithms for potentially many data conditions, it is very challenging to provide an exhaustive list of fully optimized representations for a specific combination of biological tasks, learning algorithms and datasets. Nevertheless, one can explore near-optimal representations to each potential combination of biological task, learning model, and dataset and select appropriate mathematical representations with suitable parameters. Using topological representations as an example, we outline the construction of a few topological learning strategies. In general, kNNs are very simple and are used to facilitate optimal transport approaches, such as Wasserstein metrics. However, their results might not be the optimal.[49] LR algorithms can be quite accurate,[25,27] but their training is quite time-consuming. Ensemble methods, such as RF, GBDT, and ET, are relatively accurate and efficient.[49,64,92,94] In particular, RF should be the method of choice for a new problem due to its fewer parameters and robustness. Due to its accuracy and robustness, the RF method is often used to rank the feature importance. Utilizing a few more parameters, GBDT can typically improve RF's predictions after a more intensive parameter search.

Ensemble methods and deep CNNs can be very accurate and robust against overfitting originated from large machine learning dimensions by shrinkage and dropout techniques, respectively.[46,47] Therefore, they can be used to examine a large number of representations. It is worth noting that none of these methods works well when the statistics of the test set differs much from that of the training set. When training datasets are sufficiently large, deep learning methods can be more accurate but might involve a very expensive training because of multiple layers of neurons.[48,49,92,94] Transfer learning or multitask learning can be used to improve the prediction of small datasets when they are coupled to a large dataset that shares similar statistics and the same representation structure.[48,92,94]

Intrinsically low-dimensional representations based on advanced mathematics can be constructed for complex learning models involving multiple neural networks, such as domain adaptation, active learning, recurrent neural network, long short term memory, autoencoder, generative adversarial networks, and various reinforcement learning algorithms.

III Datasets and evaluation metrics

III.A Datasets

In this review, we illustrate our models against three commonly used drug-discovery related benchmark datasets, namely, CASF-2007,[149] CASF-2013,[150] and CASF-2016.[151] These benchmarks are collected in the PDBbind database and have been used to evaluate the general performance of a scoring function on a diverse set of protein–ligand complexes (Table 3).

Note that for docking power and screening power assessments, additional data information is given for CASF-2007[149] and CASF-2013[150,152] as described in the next section.

III.B Evaluation metrics

In the drug-design related benchmark, a scoring function (SF) is often validated based on four common metrics, namely, scoring power, ranking power, docking power, and screening power.[149,152] The following sections briefly offer introductions for these matrices and the associated datasets.

III.B.1 Scoring power. This metric measures how good a scoring function is in predicting affinities that linearly correlate with the experimental data. To this end, the standard Pearson's correlation coefficient (R_p) is employed

$$R_p = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2}\sqrt{\sum (y_i - \bar{y})^2}}, \quad (42)$$

Table 3 Summary of PDBbind datasets used in the present work

	Training set complexes	Test set complexes
CASF-2007 benchmark	1105	195
CASF-2013 benchmark	3516	195
CASF-2016 benchmark	3772	285

where x_i and y_i are, respectively, the predicted binding affinity and experimental data for the ith complex. The average of all predicted and experimental values are denoted as $\bar{x}$ and $\bar{y}$, respectively. All three benchmark datasets, CASF-2007, CASF-2013, and CASF-2016, were used to evaluate the scoring power of our models.

III.B.2 Ranking power. In this assessment, the ability to ranking the binding affinity of complexes in the same cluster is stressed.[149,152] Two benchmarks, CASF-2007 and CASF-2013, were used to test our AGL-Score's ranking power. Both these datasets have 65 different protein targets, and each protein has three binding distinct ligands. There are two different levels of assessments. The first is a high-level success measurement which determines whether the affinities of three ligands in each cluster are correctly ranked. The other assessment is the so-called low-level success measurement which determines whether a scoring function can identify the ligand with the highest binding affinity in its cluster. The score in this assessment is calculated by the percentage of successful ranking in a given benchmark.

The above-mentioned ranking power evaluation may not be robust since there are only three ligands in each cluster used to determine the order ranking. Thus, the real performance of the scoring function in virtual screening cannot be transferable. Moreover, more accurate statistical information can be attained by Kendall's tau or Spearman correlation coefficient as used in D3R Grand Challenges.[153]

III.B.3 Docking power. This metric is used to determine the ability of a scoring function in discriminating the ''native'' pose from the docking software-generated structures.[149] To determine the native pose, one used the root-mean-square deviation (RMSD) between that structure and the true binding pose. If its RMSD is less than 2 Å, that pose is classified as native. Each ligand in the CASF-2007 benchmark has 100 generated structures using four docking software, namely, GOLD,[154,155] Surflex,[156,157] FLexX[158] and LigandFit.[159] In CASF-2013, there are still 100 software-generated structures for each ligand but from three docking software, namely, GOLD v5.1 (https://www.ccdc.cam.ac.uk), Surflex-Dock provided in SYBYL v8.1 (https://www.certara.com/), and MOE v2011 (https://www.chemcomp.com/). It is noted that RMSD formulation in CASF-2007 is different from one in CASF-2013. Specifically, RMSD in CASF-2007 used a standard representation but property-matched RMSD ($RMSD^{PM}$) is employed in CASF-2013.[150,152] The use of new RMSD formulation is due to the incorrect values reported by the standard RMSD on the symmetric structures. It is worth mentioning that each ligand can have more than one ''native'' structure in the benchmark. Thus, if a scoring function is able to detect any native poses, one can regard it as a successful task. The number of ligands whose ''native'' poses are precisely selected defines the docking power of the method.

III.B.4 Screening power. This assessment relates to the scoring function's capability in the differentiation of a target protein's true binders from unbinding structures. The CASF-2013 benchmark is used in this assessment. This dataset consists of 65 different protein classes. In each protein class, at least three ligands bind to that target. The true binder that has the highest experimental binding affinity is regarded as the best true binder. In this assessment, there are two different kinds of measurements. The first type concerns the enrichment factor (EF) in x% top-ranked candidates:

$$\mathrm{EF}_{x\%} = \frac{\text{Number of true binders among } x\% \text{ top-ranked candidates}}{\text{Total number of true binders of the given target protein}}. \quad (43)$$

In this measure, top-ranked candidates are the ligands with high binding affinities predicted by the scoring function. The screen power is determined by the average of all EF values over 65 targets in the benchmark.

The second type of screening power is the success rate which concerns the best true binder identification. The percentage of identifying the best binders for 65 receptors from x% top-ranked candidates yields the value of the success rate.

IV Results and discussion

In this section, we review the scoring power, ranking power, docking power and screening power of the discussed mathematical models on the three benchmark sets including CASF-2007, CASF-2013, and CASF-2016.

IV.A Hyperparameter optimization

To achieve optimal hyperparameters among the possible combinations listed in Tables 1 and 2 for our models, 5-fold cross validation-based grid search strategies are taken into account. For each CASF benchmark, the training data excluding the corresponding data are employed for the aforementioned grid search. As a result, the best EIC models in the differential based approach are $\mathrm{EIC}^{\mathrm{HH}}_{\mathrm{E},2,1;\mathrm{E},3,3}$ and $\mathrm{EIC}^{\mathrm{HH}}_{\mathrm{L},3.5,0.5;\mathrm{L},3.5,2}$ for CASF-2007. In CASF-2013, two optimal models are $\mathrm{EIC}^{\mathrm{HH}}_{\mathrm{E},1.5,5;\mathrm{E},3.5,3}$ (R_p = 0.771) and $\mathrm{EIC}^{\mathrm{HH}}_{\mathrm{L},4.5,2.5;\mathrm{L},5.5,5}$. The selected hyperparameters found in CASF-2013 are also employed in CASF-2016. In AGL-Score models, we find that the following hyperparameters attain the highest cross-validation scores for all the CASF benchmarks: $\mathrm{AGL}^{\mathrm{Adj}}_{\mathrm{E},6,2.5;\mathrm{E},4,2}$ and $\mathrm{AGL}^{\mathrm{Adj}}_{\mathrm{L},3.5,1.5;\mathrm{L},15,0.5}$. Noting that the consensus models, which are achieved by the mean of predictions of two associated models, will further lift the accuracy. Therefore, they are included in our experiments.

IV.B Performance and discussion

IV.B.1 Scoring power. In this task, we measure the Pearson correlation coefficient (R_p) between predicted affinity by our models, namely, TopBP, EIC-Score, and AGL-Score and experimental values on CASF-2007, CASF-2013, and CASF-2016. The optimal hyperparameters for AGL-Score which are chosen based on the procedure described in Section IV.A are $\mathrm{AGL}^{\mathrm{Adj}}_{\mathrm{E},6,2.5;\mathrm{E},4,2}$ and $\mathrm{AGL}^{\mathrm{Adj}}_{\mathrm{L},3.5,1.5;\mathrm{L},15,0.5}$. To validate the scoring power of AGL-Score models on CASF-2007, we train the two aforementioned models on that benchmark's training set consisting of 1105 samples after excluding 195 complexes in the test set. To reduce the variance in our results, we perform 50 prediction tasks of AGL-Score models at different random seeds. The final reported

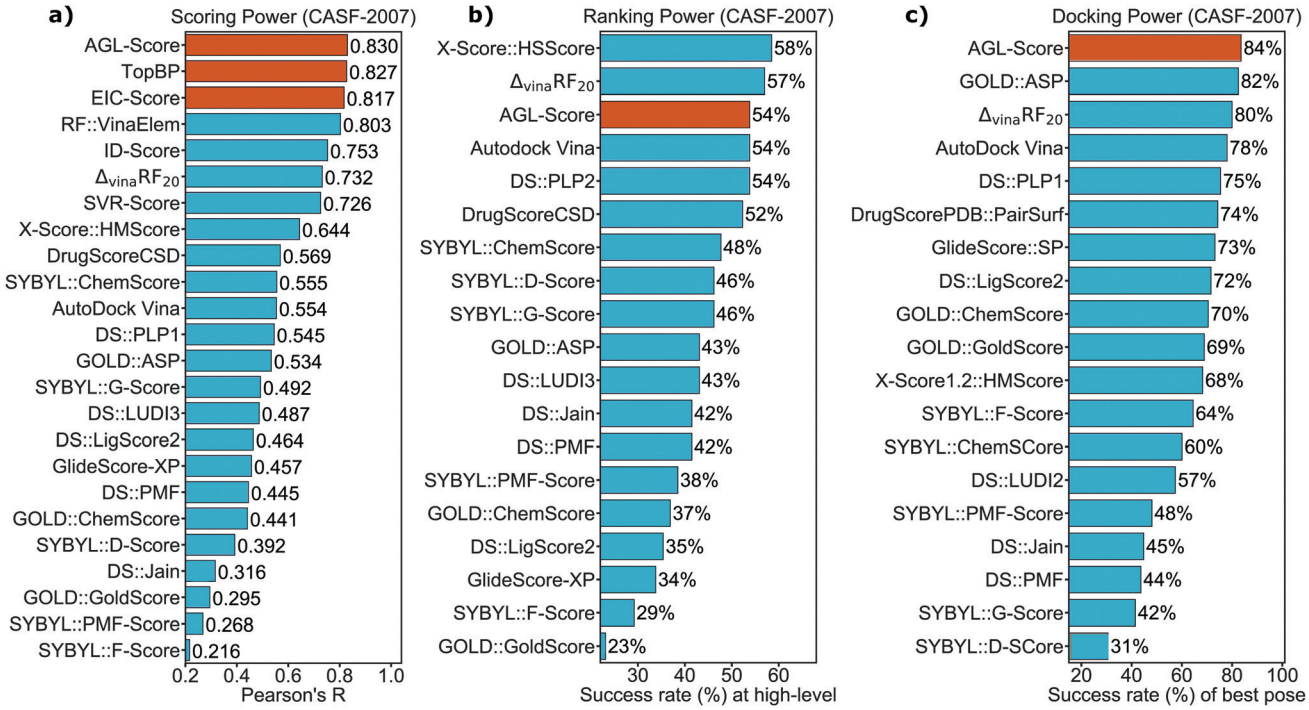

Fig. 10 The performances on different evaluation metrics of various scoring functions on the CASF-2007 benchmark. (a) Scoring power ranked by the Pearson correlation coefficient, (b) ranking power assessed by the high-level success measurement, and (c) docking power measured by the rate of successfully identifying the "native" pose from 100 poses for each ligand. Our developed models, namely, TopBP,[49] EIC-Score,[89] and AGL-Score[66] are colored in orange, and other scoring functions[48,89,149,160–163] are colored in teal.

affinity is defined by averaging all the predicted values at different runs. Similarly, we also train the optimal models of DG-GL, *i.e.* $EIC^{HH}_{E,2,1;E,3,3}$ and $EIC^{HH}_{L,3.5,0.5;L,3.5,2}$, and topology based models (TopBP) on 1105 complexes of CASF-2007. To compare the accuracy of our models with other state-of-the-art models, Fig. 10a provides a comprehensive list of various scoring functions published in the literature.[149,160–163] It is encouraging to see that all our models are at the top positions. Particularly, AGL-Score is the best model with $R_p = 0.830$, followed by TopBP with $R_p = 0.827$ and EIC-Score with $R_p = 0.817$.

To predict the affinity labels of the test set consisting of 195 complexes in the CASF-2013 benchmark, we train the TopBP, EIC-Score, and AGL-Score models with optimal parameters selected in Section IV.A on CASF-2013's training set having 3516 samples. We also provide a list of various scoring functions' performances on this benchmark as illustrated in Fig. 11a. The data from that figure reveals that our TopBP is ranked in the first place with a Pearson correlation coefficient value $R_p = 0.808$, followed by AGL-Score with its $R_p = 0.792$. Our differential geometry-based model is in the third place with $R_p = 0.774$. The fourth place in the ranking table is PLEC-nn,[165] a deep learning network model.

Similar to the training procedure on the first two benchmarks, in the last one, *i.e.* CASF-2016, the structures of our three models are learned from the training set ($N = 3772$) of this benchmark. Fig. 12 compares R_p of numerous scoring functions on the CASF-2016. Consistently, our models still achieve the highest correlation values with $R_p = 0.861$, $R_p = 0.835$, and $R_p = 0.825$ for TopBP, AGL-Score, and EIC-Score, respectively. It is worth noting that all top models in this benchmark are machine learning-based scoring functions, namely, K_{DEEP},[166] Pafnucy,[167] and PLEC-nn.[165] These models predict the energies for the test set of 290 complexes which is the PDBbind v2016 core set (Table 4). Our topology-based model, TopBP, was able to outperform our other methods because it used convolutional neural networks whereas AGL-Score and EIC-Score were based on gradient boosted decision trees.

IV.B.2 Ranking power. In this assessment, the predicted binding energies are used to determine the rank of the ligands binding to the same target. We evaluated the ranking power of three AGL-Score models, namely generalized exponential kernel model $AGL^{Adj}_{E,6,2.5;E,4,2}$ and generalized Lorentz kernel model $AGL^{Adj}_{L,3.5,1.5;L,15,0.5}$, and the consensus one. The result reveals that the generalized exponential kernel model produces the best performances on both CASF-2007 and CASF-2013 benchmarks. Therefore, it is the representative model of the AGL-Score on this measurement. Fig. 10b reports the ranking power of various scoring functions on CASF-2007. In this benchmark, our AGL-Score is ranked the third on high-level success with a rate of 54%, and is behind $\Delta_{vina}RF_{20}$ (success rate = 57%)[163] and d X-Score::HSScore (success rate = 58%).[149] Surprisingly, our graph-based model achieves the best success rate in CASF-2013 with the rate being 60%, followed by X-ScoreHM with the success rate as high as 59%. Since the ranking power performance depends on the predicted affinities used for the scoring power, one can see there is a correlation between these two

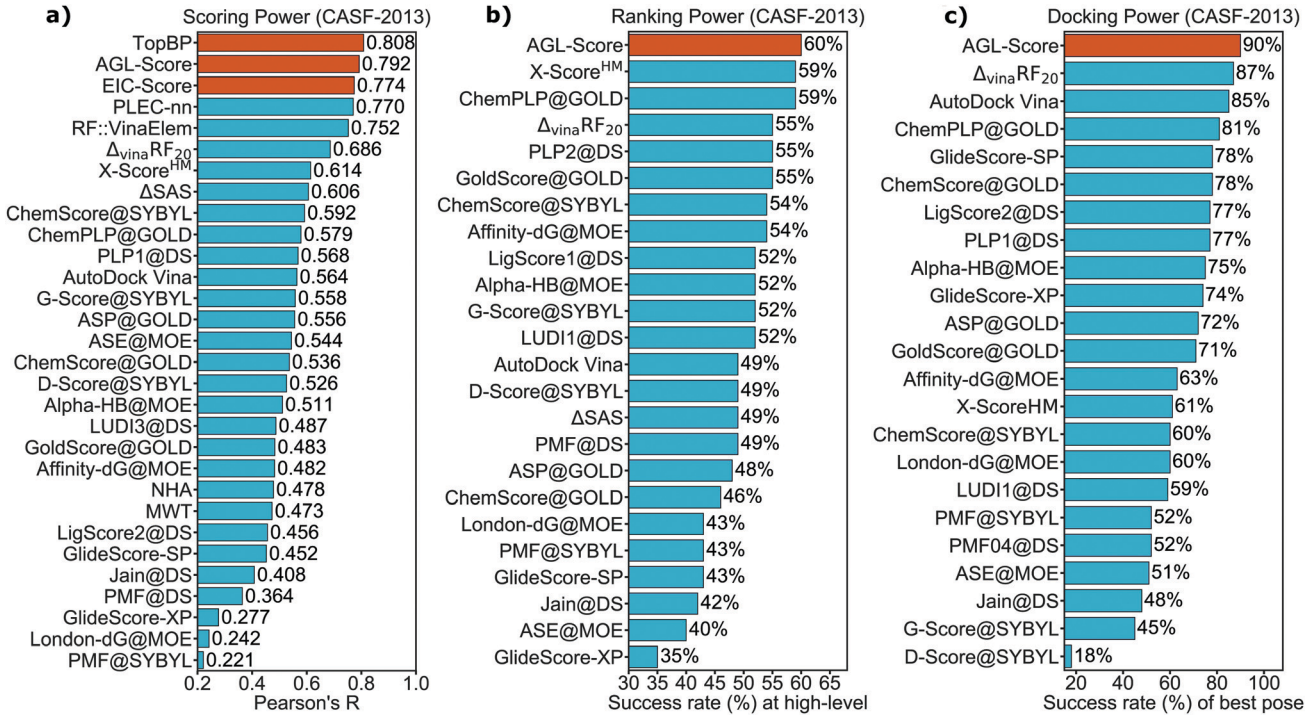

Fig. 11 The performances on different evaluation metrics of various scoring functions on the CASF-2013 benchmark. (a) Scoring power ranked by the Pearson correlation coefficient, (b) ranking power assessed by the high-level success measurement, and (c) docking power measured by the rate of successfully identifying the "native" pose from 100 poses for each ligand. Our developed models, namely, TopBP,[49] EIC-Score,[89] and AGL-Score[66] are colored in orange, and other scoring functions[89,150,163–165] are colored in teal.

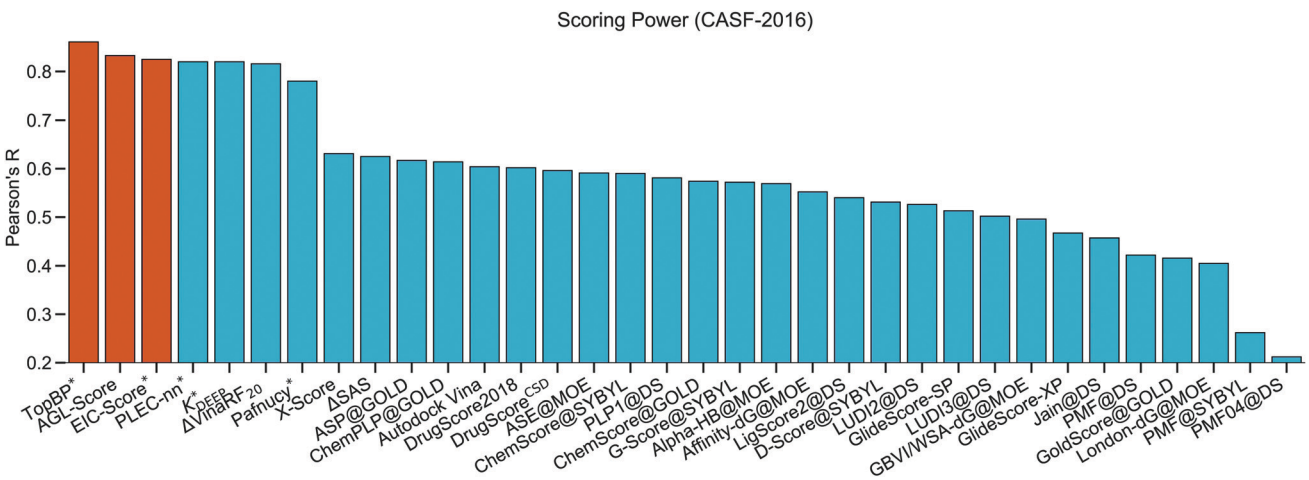

Fig. 12 The Pearson correlation coefficient of various scoring functions on CASF-2016. Our developed models, namely, TopBP,[49] EIC-Score,[89] and AGL-Score[66] are colored in orange. The performances of other models that are in teal are taken from ref. 48, 89, 151 and 165–167 Our TopBP is the best model with R_p = 0.861 and RMSE = 1.65 kcal mol^{-1}. Our AGL-Score is the second best model, with R_p = 0.833 and RMSE = 1.733 kcal mol^{-1}. The third-ranked scoring function is still our model, EIC-Score, with R_p = 0.825 and RMSE = 1.767 kcal mol^{-1}. Note that scoring functions marked with * use PDBbind v2016 core set (N = 290).

Table 4 Discrepancy information between PDBbind v2016 core set and CASF-2016 test set

	PDBID
Complexes in CASF-2016 but not in PDBbind v2016 core set	1g2k
Complexes in PDBbind v2016 core set but not in CASF-2016	4mrw, 4mrz, 4msn, 5c1w, 4msc, 3cyx

assessments. However, our AGL-Score is the only model that is ranked in the top three places in these metrics for both CASF-2007 and CASF-2013 benchmarks.

IV.B.3 Docking power. This docking power examines the ability of a scoring function in the discrimination between ''native'' and ''non-native'' poses. To build a robust machine learning-based model for this task, it is natural to include the diverse conformers with different ranges of root-mean-squared deviation (RMSD) to target experimental structure. Therefore, to create a satisfactory training data set for our AGL-Score model, we carry out GOLD v5.6.3[155] to set up a training set of 1000 poses for a given target ligand and its corresponding receptor. The parameters in the GOLD software are chosen as the following autoscale = 1.5, early_termination = 0, and gold_fitfunc_path = plp. The total of computer-generated structures for both CASF-2007 and CASF-2013 benchmarks is 365 000 poses which are fed to the AGL-Score for the learning process. Interested readers can download this structure information from our online server https://weilab.math.msu.edu/AGL-Score.

In considering benchmarks, each target ligand has 100 generated structures. To identify its ''native'' poses, we retrain single exponential kernel AGL-Score $AGL^{Adj}_{E,6,2.5}$ on 1000 poses generated by docking software for that specific ligand. The single model is used here to save the calculation and training time. The accuracy and robustness of our AGL-Score model on the docking power is illustrated in Fig. 10c and 11c for CASF-2007 and CASF-2013, respectively. In both benchmarks, our graph-based model is ranked in the first place. Specifically, on CASF-2007, the success rate of the AGL-Score model is 84%, the second and third best models are GOLD::ASP (82%)[149] and $\Delta_{vina}RF_{20}$ (80%),[163] respectively. On CASF-2013, the success rate of our method is higher with the rate being 90%, while $\Delta_{vina}RF_{20}$[163] and Autodock Vina[163] only attain 87% and 85%, respectively.

The training data of the AGL-Score model for this assessment is provided by the docking software GOLD with ChemPLP as a scoring function type (ChemPLP@GOLD). It is interesting to see how this scoring function performs on the same benchmark. The ChemPLP@GOLD model achieves the success rates of 67% and 82% for CASF-2007 and CASF-2013, respectively. These values are much lower than those of our model (84% and 90%). These comparisons confirm that our AGL-Score indeed upgrades the accuracy of the existing docking software by correctly extracting the real physical and biological properties of a biomolecular structure.

Scoring power and docking power are two very different measurement metrics. The first one concerns the affinity with the training data based on the experimental information. The latter targets the geometric validation involving artificial data. Consequently, it is not an easy task to accomplish state-of-the-art performances on both evaluations.[168–170] According to our observation, the most commonly used docking software is reliable in identifying the ''native'' structures but inadequate in the binding energy prediction. For instance, GOLD with ASP as a scoring function (ASP@GOLD) performs quite well on the docking power with the success rate being 82% in CASF-2007. However, ASP@GOLD's performance on the scoring power does not meet the satisfactory accuracy with $R_p = 0.534$. In contrast, the machine learning-based scoring functions often display an opposite impression. For example, RF-IChem[169] is a machine learning model and attains a higher Pearson correlation coefficient on the scoring power ($R_p = 0.791$), as expected. Unfortunately, due to the lack of appropriate training data and too simple representations for accurately encoding physical and biological information of a molecule, RF-IChem has difficulty in detecting the ''native'' pose with the success rate as low as 30%. Recently, a machine learning-based model named $\Delta_{vina}RF_{20}$ was developed by Wang and Zhang[163] with a purpose of improving the accuracy of random-forest based scoring functions on various evaluations. Indeed, $\Delta_{vina}RF_{20}$ offers an excellent success rate (80%) on the docking power of CASF-2007 but still shows a respectable precision on binding affinity prediction with $R_p = 0.732$. Nevertheless, the Pearson correlation coefficient of the $\Delta_{vina}RF_{20}$ is far behind the elite models such as TNet-BP ($R_p = 0.826$).[48] Our graph based-model, AGL-Score, not only has a great accomplishment on the docking power (success rate = 84% in CASF-2007) as $\Delta_{vina}RF_{20}$, but also performs similarly to TNet-BP on the scoring power ($R_p = 0.83$ in CASF-2007). These results again reinforce the ability of the AGL-Score in capturing the crucial interactions in molecular and biomolecular structures.

IV.B.4 Screening power. In this assessment, we verify the ability of the AGL-Score in picking up the true binders for 65 different protein classes in the CASF-2013 benchmark. The power metric concerns the active and inactive ones of 195 ligands for a specific class of proteins rather than the estimation of the binding affinity for an experimental complex or ''native'' conformer identification. Therefore, to effectively carry out the machine learning scoring function on this take, one needs to construct appropriate training data tailoring the active/inactive classification purpose. To this end, our training data consist of docking software-generated poses and corresponding energies. The 3D structures of 195 ligands binding to a specific target are also created by the docking program and their energies are estimated by our AGL-Score model. The predicted true binders are identified based on their predicted affinities.

Our training set for AGL-Score on this screen power test is based on the PDBbind v2015 refined set excluding the core set in that database. Besides these experimental structures, we generate the non-binder structures for each target protein by using Autodock Vina.[171] Specifically, we use that docking software to dock all ligands in the PDBbind v2015 refined set without the inclusion of the core-set compounds to the interested receptor. Here are the parameters of Autodock Vina we use in this procedure: exhaustiveness = 10, num_modes = 10, and energy_range = 3. For each docking run, the pose associated with the highest predicted affinity by Autodock Vina is kept.

To preserve the consistency in the energy unit, all the Autodock Vina scores in kcal mol^{-1} are converted to pK_d units *via* a constant factor −1.3633.[25] Ligands in the PDBbind v2015 refined set which do not bind to a target protein are designated as decoys.[150,152] To conserve the physical and biological sense, the Autodock Vina predicted energies of those decoys cannot be

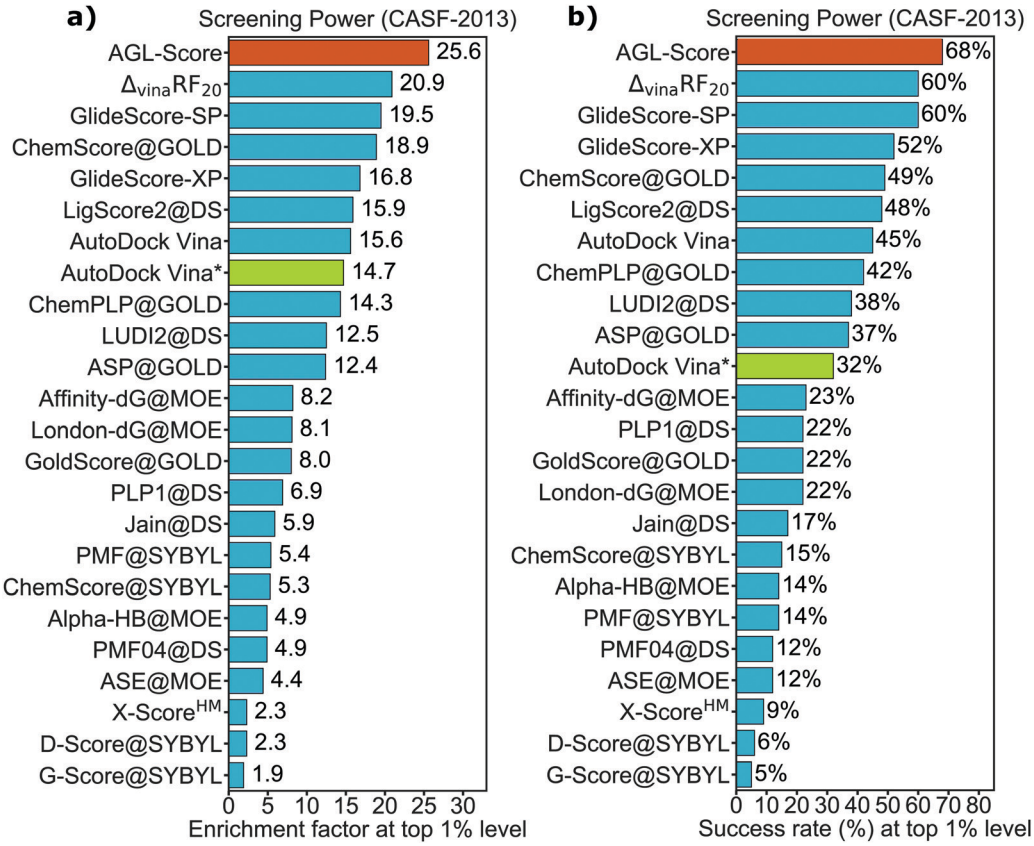

Fig. 13 The performances of various scoring functions on the screening power for CASF-2013 benchmark based on (a) enrichment factor and (b) success rate at the top 1% level. The orange bar indicates our graph-based models.[66] The green bar represents the results of Autodock Vina carried out in our lab. The teal bars express the performances of other models ref. 150 and 163.

higher than the lowest energies among the ligands experimentally bound to that target protein. To this end, we constrain the decoy energies by the lower bound of the true binders. The generated structures, as well as the energy labels of the decoys used in the AGL-Score training process, are publicly available at https://weilab.math.msu.edu/AGL-Score.

The AGL-Score model we used in this screening power is AGL-Score $AGL^{Adj}_{E,6,2.5}$. Fig. 13 plots the performance of the AGL-Score along with numerous scoring functions reported in the literature.[150,163] It is an encouragement to see that our AGL-Score achieves the top performance on enrichment factor (EF) and success rate at the top 1% level in the CASF-2013 benchmark. The EF of the AGL-Score is 25.6 followed by $\Delta_{vina}RF_{20}$ (EF = 20.9)[163] and GlideScore-SP (EF = 19.5).[150] Moreover, the success rate of our graph-based model is 68% followed by $\Delta_{vina}RF_{20}$ and GlideScore-SP that both attain 60%.

Since the partial training data of our AGL-Score model is generated by Autodock Vina, it is interesting to see the accuracy of that docking software carried out in our lab on this assessment. The Autodock Vina's performances are much lower than the graph-based model. Specifically, the docking software attains EF as low as 14.7 while AGL-Score produces EF as high as 25.6. In the success rate metric, Autodock Vina's accuracy is only 32% which is far from AGL-Score's rate at 68%. Since the published work[163] already reported Vina's screen power tests, to avoid any confusion we plot our experiments on the Vina software as green bars in Fig. 13. The unsatisfactory results of the Autodock Vina on the screen power further reinforce the accurately encoded physical and biological information in our graph-based model rather than the dependence on training quality.

The screening power validation is an important metric in virtual screening in drug design. Since this assessment strictly requires meaningful molecular representations and an appropriate training set, large numbers of machine learning-based scoring functions with simple features and irrelevant training data often perform poorly on this metric despite the promising accuracy on the scoring power. For instance, RF@ML[170] is a machine learning model using Random Forest for the prediction but its features simply count the number of intermolecular contacts between two atom types. In fact, RF@ML produces an acceptable correlation (R_p = 0.704) on 164 complexes in the PDBbind v2013 dataset. However, RF@ML's accuracies of screen power are the worst among the models listed in Fig. 13. In contrast, our AGL-Score model with superior feature representations and training data insight has achieved the top places in both scoring and screening powers.

IV.C Online servers

In the past few years, a few online servers have been developed for the predictions of protein–ligand binding affinities (RI-Score,

TML-BP, and TML-BP), protein stability changes upon mutation (TML-MP, and TML-MP), molecular toxicity (TopTox), partition coefficient and aqueous solubility (TopP-S), and protein flexibility (FRI).

V Concluding remarks

Artificial Intelligence (AI), including machine learning (ML), has had tremendous impacts on science, engineering, technology, healthcare, security, finance, education, and industry, to name just a few. However, the development of ML algorithms for macromolecular systems is hindered by their intricate structural complexity and associated high ML dimensionality. In the past few years, we have addressed these challenges by three classes of mathematical techniques based on algebraic topology, differential geometry, and graph theory. These mathematical apparatuses are enormously effective for macromolecular structural simplification and ML dimensionality reduction. By integrating with advanced ML algorithms, we have demonstrated that our mathematical approaches give rise to the best prediction in D3R Grand Challenges, a worldwide competition series in computer-aided drug design,[28,29] as well as many other physical, chemical and biological datasets. Nonetheless, our methods and results were scattered over a number of papers. In this review, we provide a systematic and coherent narration of our state-of-the-art algebraic topology, differential geometry, and graph theory-based methods. Emphasis is given to the physical and biological challenge-guided evolution of these mathematical approaches. Although our mathematical methods can be paired with various machine learning algorithms for a wide variety of chemical, physical, and biological systems, we focus on protein–ligand binding analysis and prediction in the present review.

Fueled by the fast advances in ML and the availability of biological datasets, recent years have witnessed rapid growth in the development of advanced mathematical tools in the realm of molecular biology and biophysics. In most of history, mathematics has been the driving force for natural science. Indeed, mathematics is the underpinning for every aspect of modern physics, from electrodynamics, thermodynamics, statistical mechanics, quantum mechanics, solid state physics, quantum field theory, to the general theory of relativity. In the past century, mathematics and physics have been mutually beneficial. Similarly, mathematics will become an indispensable part of biological sciences shortly. Currently, algebraic topology, differential geometry, graph theory, group theory, differential equations, algebra, and combinatorics have been widely applied to biological science. Many other advanced mathematical subjects, such as algebraic geometry and low dimensional manifolds will soon find their applications to biological science.

The next generation of AI and ML technologies will be designed to understand the rules of life and reveal the physical and molecular mechanics of biomolecular systems. Such a development will bring tremendous benefits to health sciences, including drug discovery. Mathematics will play a paramount role in future AI and ML technologies. On the one hand, the mathematical theory will contribute to the foundation of AL and the design principle of ML. On the other hand, new mathematical representations will be developed to enable the automatic discovery of scientific laws and principles.[172] New mathematical representations will be made physically interpretable so that machine learning predictions from these representations can reveal new molecular mechanisms. A generation of new mathematical representations will be made adaptive to future AI technology. Mathematical representations will be systematically validated and optimized on a vast variety of existing datasets.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was supported in part by NSF Grants DMS-1721024, DMS-1761320, and IIS1900473, NIH grants GM126189 and GM129004, Bristol-Myers Squibb, and Pfizer. We thank Dr Kaifu Gao for his contribution to our team's pose prediction in D3R Grand Challenge 4.

References

1 M. AlQuraishi, *Bioinformatics*, 2019, **35**, 4862–4865.
2 K. Schwab, *The fourth industrial revolution*, Currency, 2017.
3 A. Agrawal and A. Choudhary, *APL Mater.*, 2016, **4**, 053208.
4 K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev and A. Walsh, *Nature*, 2018, **559**, 547.
5 S. Brandt, F. Sittel, M. Ernst and G. Stock, *J. Phys. Chem. Lett.*, 2018, **9**, 2144–2150.
6 S. J. Darnell, L. LeGault and J. C. Mitchell, *Nucleic Acids. Res.*, 2008, **36**, W265–W269.
7 B. Huang and O. A. Von Lilienfeld, *J. Chem. Phys.*, 2016, **145**, 161102.
8 R. Winter, F. Montanari, F. Noé and D.-A. Clevert, *Chem. Sci.*, 2019, **10**, 1692–1701.
9 G. Wei, D. Nguyen and Z. Cang, *System and methods for machine learning for drug design and discovery*, *US Pat. App.*, 16/372239, 2019.
10 H. Geppert, M. Vogt and J. Bajorath, *J. Chem. Inf. Model.*, 2010, **50**, 205–216.
11 K. Roy and I. Mitra, *Curr. Comput.-Aided Drug Des.*, 2012, **8**, 135–158.
12 M. Tareq Hassan Khan, *Curr. Drug Metab.*, 2010, **11**, 285–295.
13 D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.
14 Y.-C. Lo, S. E. Rensi, W. Torng and R. B. Altman, *Drug Discovery Today*, 2018, 1538–1546.
15 A. Cereto-Massagué, M. J. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé and G. Pujadas, *Methods*, 2015, **71**, 58–63.

16 J. Verma, V. M. Khedkar and E. C. Coutinho, *Curr. Top. Med. Chem.*, 2010, **10**, 95–115.
17 J. L. Durant, B. A. Leland, D. R. Henry and J. G. Nourse, *J. Chem. Inf. Comput. Sci.*, 2002, **42**, 1273–1280.
18 N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch and G. R. Hutchison, *J. Cheminf.*, 2011, **3**, 33.
19 D. Toolkit, Inc.: Aliso Viejo, CA, 2007.
20 L. H. Hall and L. B. Kier, *J. Chem. Inf. Comput. Sci.*, 1995, **35**, 1039–1045.
21 G. Landrum *et al.*, *RDKit: Open-source cheminformatics*, 2006.
22 N. Stiefl, I. A. Watson, K. Baumann and A. Zaliani, *J. Chem. Inf. Model.*, 2006, **46**, 208–220.
23 O. N. A. Demerdash, M. D. Daily and J. C. Mitchell, *PLoS Comput. Biol.*, 2009, **5**, e1000531.
24 J. Lu, C. Wang and Y. Zhang, *J. Chem. Theory Comput.*, 2019, 4113–4121.
25 B. Wang, Z. Zhao, D. D. Nguyen and G. W. Wei, *Theor. Chem. Acc.*, 2017, **136**, 55.
26 B. Wang, Z. Zhao and G. W. Wei, *J. Chem. Phys.*, 2016, **145**, 124110.
27 B. Wang, C. Wang, K. D. Wu and G. W. Wei, *J. Comput. Chem.*, 2018, **39**, 217–232.
28 D. D. Nguyen, Z. Cang, K. Wu, M. Wang, Y. Cao and G.-W. Wei, *J. Comput.-Aided Mol. Des.*, 2019, **33**, 71–82.
29 D. D. Nguyen, K. Gao, M. Wang and G.-W. Wei, *J. Comput.-Aided Mol. Des.*, 2019, DOI: 10.1007/s10822-019-00237-5.
30 T. Schlick and W. K. Olson, *Science*, 1992, **257**, 1110–1115.
31 A. Zomorodian and G. Carlsson, *Discrete Comput. Geom.*, 2005, **33**, 249–274.
32 D. W. Sumners, *Proc. Symp. Appl. Math.*, 1992, 39–72.
33 H. Edelsbrunner, D. Letscher and A. Zomorodian, Proceedings 41st Annual Symposium on Foundations of Computer Science, 2000, pp. 454–463.
34 A. Zomorodian and G. Carlsson, *Comput. Geom.*, 2008, **41**, 126–148.
35 Y. Yao, J. Sun, X. H. Huang, G. R. Bowman, G. Singh, M. Lesnick, L. J. Guibas, V. S. Pande and G. Carlsson, *J. Chem. Phys.*, 2009, **130**, 144115.
36 M. Gameiro, Y. Hiraoka, S. Izumi, M. Kramar, K. Mischaikow and V. Nanda, *Jpn. J. Ind. Appl. Math.*, 2014, **32**, 1–17.
37 K. L. Xia and G. W. Wei, *Int. J. Numer. Methods Biomed. Eng.*, 2014, **30**, 814–844.
38 K. L. Xia, X. Feng, Y. Y. Tong and G. W. Wei, *J. Comput. Chem.*, 2015, **36**, 408–422.
39 K. L. Xia and G. W. Wei, *Int. J. Numer. Methods Biomed. Eng.*, 2015, **31**, e02719.
40 K. L. Xia and G. W. Wei, *J. Comput. Chem.*, 2015, **36**, 1502–1520.
41 K. L. Xia, Z. X. Zhao and G. W. Wei, *J. Comput. Biol.*, 2015, **22**, 1–5.
42 K. L. Xia, Z. X. Zhao and G. W. Wei, *J. Chem. Phys.*, 2015, **143**, 134103.
43 Z. X. Cang, L. Mu, K. Wu, K. Opron, K. Xia and G.-W. Wei, *Mol. Based Math. Biol.*, 2015, **3**, 140–162.
44 B. Wang and G. W. Wei, *J. Comput. Phys.*, 2016, **305**, 276–299.
45 B. Liu, B. Wang, R. Zhao, Y. Tong and G. W. Wei, *J. Comput. Chem.*, 2017, **38**, 446–466.
46 Z. X. Cang and G. W. Wei, *Bioinformatics*, 2017, **33**, 3549–3557.
47 Z. X. Cang and G. W. Wei, *Int. J. Numer. Methods Biomed. Eng.*, 2018, **34**(2), e2914.
48 Z. X. Cang and G. W. Wei, *PLoS Comput. Biol.*, 2017, **13**(7), e1005690.
49 Z. X. Cang, L. Mu and G. W. Wei, *PLoS Comput. Biol.*, 2018, **14**(1), e1005929.
50 F. R. K. Chung, *AMS*, 1997.
51 R. Twarock and N. Jonoska, *J. Phys. A: Math. Theor.*, 2008, **41**, 304043.
52 D. Janezic, A. Milicevic, S. Nikolic and N. Trinajstic, *Graph-theoretical matrices in chemistry*, CRC Press, 2015.
53 Z. Li, N. Omidvar, W. S. Chin, E. Robb, A. Morris, L. Achenie and H. Xin, *J. Phys. Chem. A*, 2018, **122**, 4571–4578.
54 I. Bahar, A. R. Atilgan and B. Erman, *Folding Des.*, 1997, **2**, 173–181.
55 A. R. Atilgan, S. R. Durrell, R. L. Jernigan, M. C. Demirel, O. Keskin and I. Bahar, *Biophys. J.*, 2001, **80**, 505–515.
56 K. L. Xia, K. Opron and G. W. Wei, *J. Chem. Phys.*, 2013, **139**, 194109.
57 K. L. Xia and G. W. Wei, *Chaos*, 2014, **24**, 013103.
58 K. Xia and G.-W. Wei, 2016, arXiv preprint arXiv:1612.01735.
59 K. Opron, K. L. Xia and G. W. Wei, *J. Chem. Phys.*, 2014, **140**, 234105.
60 K. Opron, K. L. Xia and G. W. Wei, *J. Chem. Phys.*, 2015, **142**, 211101.
61 K. Opron, K. L. Xia, Z. Burton and G. W. Wei, *J. Comput. Chem.*, 2016, **37**, 1283–1295.
62 D. D. Nguyen, K. L. Xia and G. W. Wei, *J. Chem. Phys.*, 2016, **144**, 234106.
63 K. L. Xia, K. Opron and G. W. Wei, *J. Chem. Phys.*, 2015, **143**, 204106.
64 D. D. Nguyen, T. Xiao, M. L. Wang and G. W. Wei, *J. Chem. Inf. Model.*, 2017, **57**, 1715–1721.
65 D. Bramer and G. W. Wei, *J. Chem. Phys.*, 2018, **148**, 054103.
66 D. Nguyen and G.-W. Wei, *J. Chem. Inf. Model.*, 2019, **59**, 3291–3304.
67 B. S. Duncan and A. J. Olson, *Biopolymers*, 1993, **33**, 231–238.
68 H. Sun, H. Ferhatosmanoglu, M. Ota and Y. Wang, *BMC Bioinf.*, 2008, **9**, 344.
69 T. K. Dey, F. Fan and Y. Wang, Proc. 29th Annu. Sympos. Comput. Geom. (SoCG), 2013, pp. 425–434.
70 G. W. Wei, Y. H. Sun, Y. C. Zhou and M. Feig, 2005, arXiv:math-ph/0511001v1, pp. 1–11.
71 P. W. Bates, G. W. Wei and S. Zhao, arXiv:q-bio/0610038v1, 2006, [q-bio.BM].
72 P. W. Bates, G. W. Wei and S. Zhao, *J. Comput. Chem.*, 2008, **29**, 380–391.
73 X. Feng, K. Xia, Y. Tong and G.-W. Wei, *Int. J. Numer. Methods Biomed. Eng.*, 2012, **28**, 1198–1223.
74 X. Feng, K. L. Xia, Y. Y. Tong and G. W. Wei, *J. Comput. Chem.*, 2013, **34**, 2100–2120.

75 K. L. Xia, X. Feng, Y. Y. Tong and G. W. Wei, *J. Comput. Phys.*, 2014, **275**, 912–936.
76 Z. Chen, N. A. Baker and G. W. Wei, *J. Comput. Phys.*, 2010, **229**, 8231–8258.
77 Z. Chen, N. A. Baker and G. W. Wei, *J. Math. Biol.*, 2011, **63**, 1139–1200.
78 Z. Chen and G. W. Wei, *J. Chem. Phys.*, 2011, **135**, 194108.
79 Z. Chen, S. Zhao, J. Chun, D. G. Thomas, N. A. Baker, P. B. Bates and G. W. Wei, *J. Chem. Phys.*, 2012, **137**, 084101.
80 D. Chen, Z. Chen and G. W. Wei, *Int. J. Numer. Methods Biomed. Eng.*, 2012, **28**, 25–51.
81 D. Chen and G. W. Wei, *J. Chem. Phys.*, 2012, **136**, 134109.
82 G.-W. Wei, Q. Zheng, Z. Chen and K. Xia, *SIAM Rev.*, 2012, **54**, 699–754.
83 M. Daily, J. Chun, A. Heredia-Langner, G. W. Wei and N. A. Baker, *J. Chem. Phys.*, 2013, **139**, 204108.
84 D. Thomas, J. Chun, Z. Chen, G. W. Wei and N. A. Baker, *J. Comput. Chem.*, 2013, **24**, 687–695.
85 D. D. Nguyen and G. W. Wei, *J. Comput. Chem.*, 2017, **38**, 24–36.
86 G. W. Wei, *Bull. Math. Biol.*, 2010, **72**, 1562–1622.
87 G. W. Wei, *J. Theor. Comput. Chem.*, 2013, **12**, 1341006.
88 R. Zhao, Z. Cang, Y. Tong and G.-W. Wei, *Bioinformatics*, 2018, **34**(17), i830–i837.
89 D. D. Nguyen and G.-W. Wei, *Int. J. Numer. Methods Biomed. Eng.*, 2019, **35**, e3179.
90 R. Zhao, M. Desbrun, G. Wei and Y. Tong, *ACM Trans. Graph.*, 2019, **38**, 181.
91 R. Zhao, M. Wang, Y. Tong and G.-W. Wei, 2019, arXiv preprint arXiv:1908.00572.
92 K. Wu and G. W. Wei, *J. Chem. Inf. Model.*, 2018, **58**, 520–531.
93 B. Wang and G. W. Wei, *J. Chem. Phys.*, 2015, **143**, 134119.
94 K. Wu, Z. Zhao, R. Wang and G. W. Wei, *J. Comput. Chem.*, 2018, **39**, 1444–1454.
95 I. K. Darcy and M. Vazquez, *Biochem. Soc. Trans.*, 2013, **41**, 601–605.
96 C. Heitsch and S. Poznanovic, *Discrete and Topological Models in Molecular Biology*, 2014, ch. 7, pp. 145–166.
97 B. DasGupta and J. Liang, *Models and Algorithms for Biomolecules and Molecular Networks*, John Wiley & Sons, 2016.
98 X. Shi and P. Koehl, *Far East J. Appl. Math.*, 2011, **50**, 1–34.
99 T. Kaczynski, K. Mischaikow and M. Mrozek, *Computational Homology*, Springer-Verlag, 2004.
100 G. Carlsson, A. Zomorodian, A. Collins and L. J. Guibas, *Int. J. Shape Model.*, 2005, **11**, 149–187.
101 A. Tausz, M. Vejdemo-Johansson and H. Adams, *JavaPlex: A research software package for persistent (co)homology*, Software available at http://code.google.com/p/javaplex, 2011.
102 K. Mischaikow and V. Nanda, *Discrete Comput. Geom.*, 2013, **50**, 330–353.
103 M. Allili, K. Mischaikow and A. Tannenbaum, 2001 International Conference on Image Processing, vol II, Proceedings, 2001, pp. 173–176.
104 Z. X. Cang and G. W. Wei, 2018, arXiv:1807.11120 [q-bio.QM].
105 U. Bauer, Software available at https://github.com/Ripser/ripser, 2017.
106 V. De Silva, D. Morozov and M. Vejdemo-Johansson, *Discrete Comput. Geom.*, 2011, **45**, 737–759.
107 P. W. Bates, Z. Chen, Y. H. Sun, G. W. Wei and S. Zhao, *J. Math. Biol.*, 2009, **59**, 193–231.
108 L. Mu, K. Xia and G. Wei, *J. Comput. Appl. Math.*, 2017, **313**, 18–37.
109 D. Chen and G. W. Wei, *Commun. Comput. Phys.*, 2013, **13**, 285–324.
110 G. W. Wei, *J. Phys. A: Math. Gen.*, 2000, **33**, 8577–8596.
111 K. Wolfgang, *Differential Geometry: Curves-Surface-Manifolds*, American Mathematical Society, 2002.
112 O. Soldea, G. Elber and E. Rivlin, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, 265–278.
113 J. Pach, *Erdös Centennial*, Springer, 2013, pp. 465–484.
114 C. Godsil and G. F. Royle, *Algebraic graph theory*, Springer Science & Business Media, 2013, vol. 207.
115 L. Babai, *Handbook of combinatorics*, 1996, vol. 2, pp. 1447–1540.
116 P. de La Harpe, *Topics in geometric group theory*, University of Chicago Press, 2000.
117 B. Korte, L. Lovász and R. Schrader, *Greedoids*, Springer Science & Business Media, 2012, vol. 4.
118 F. Larrión, V. Neumann-Lara and M. A. Pizaña, *Discrete Math.*, 2002, **258**, 123–135.
119 A. T. Balaban, *Chemical Applications of Graph Theory*, Academic Press, 1976.
120 N. Trinajstic, *Chemical graph theory*, Boca Raton, 1983.
121 H. P. Schultz, *J. Chem. Inf. Comput. Sci.*, 1989, **29**, 227–228.
122 L. R. Foulds, *Graph theory applications*, Springer Science & Business Media, 2012.
123 P. J. Hansen and P. C. Jurs, *J. Chem. Educ.*, 1988, **65**, 574.
124 A. Ozkanlar and A. E. Clark, *J. Comput. Chem.*, 2014, **35**, 495–505.
125 L. Di Paola and A. Giuliani, *Curr. Opin. Struct. Biol.*, 2015, **31**, 43–48.
126 A. A. Canutescu, A. A. Shelenkov and R. L. Dunbrack, *Protein Sci.*, 2003, **12**, 2001–2014.
127 G. A. Ryslik, Y. Cheng, K.-H. Cheung, Y. Modis and H. Zhao, *BMC Bioinf.*, 2014, **15**, 86.
128 D. J. Jacobs, A. J. Rader, L. A. Kuhn and M. F. Thorpe, *Proteins*, 2001, **44**, 150–165.
129 S. Vishveshwara, K. Brinda and N. Kannan, *J. Theor. Comput. Chem.*, 2002, **1**, 187–211.
130 Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing and V. Pande, *Chem. Sci.*, 2018, **9**, 513–530.
131 M. Newman, *Networks: An Introduction*, Oxford University Press, Inc., USA, 2010.
132 A. Bavelas, *J. Acoust. Soc. Am.*, 1950, **22**, 725–730.
133 A. Dekker, *J. Soc. Struct.*, 2005, **6**, 31.
134 L. W. Yang and C. P. Chng, *Bioinf. Biol. Insights*, 2008, **2**, 25–45.
135 H. Hosoya, *Bull. Chem. Soc. Jpn.*, 1971, **44**, 2332–2339.
136 A. Angeleska, N. Jonoska and M. Saito, *Discrete Appl. Math.*, 2009, **157**, 3020–3037.

137 N. Go, T. Noguti and T. Nishikawa, *Proc. Natl. Acad. Sci. U. S. A.*, 1983, **80**, 3696–3700.
138 M. Tasumi, H. Takenchi, S. Ataka, A. M. Dwidedi and S. Krimm, *Biopolymers*, 1982, **21**, 711–714.
139 B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. States, S. Swaminathan and M. Karplus, *J. Comput. Chem.*, 1983, **4**, 187–217.
140 M. Levitt, C. Sander and P. S. Stern, *J. Mol. Biol.*, 1985, **181**, 423–447.
141 P. J. Flory, *Proc. R. Soc. London, Ser. A*, 1976, **351**, 351–378.
142 I. Bahar, A. R. Atilgan, M. C. Demirel and B. Erman, *Phys. Rev. Lett.*, 1998, **80**, 2733–2736.
143 K. Hinsen, *Proteins*, 1998, **33**, 417–429.
144 F. Tama and Y. H. Sanejouand, *Protein Eng.*, 2001, **14**, 1–6.
145 Q. Cui and I. Bahar, *Normal mode analysis: theory and applications to biological and chemical systems*, Chapman and Hall/CRC, 2010.
146 J. K. Park, R. Jernigan and Z. Wu, *Bull. Math. Biol.*, 2013, **75**, 124–160.
147 L. Quan, Q. Lv and Y. Zhang, *Bioinformatics*, 2016, **32**, 2936–2946.
148 C. Grow, K. Gao, D. D. Nguyen and G.-W. Wei, *Commun. Inf. Syst.*, 2019, **19**, 241–277.
149 T. Cheng, X. Li, Y. Li, Z. Liu and R. Wang, *J. Chem. Inf. Model.*, 2009, **49**, 1079–1093.
150 Y. Li, L. Han, Z. Liu and R. Wang, *J. Chem. Inf. Model.*, 2014, **54**, 1717–1736.
151 M. Su, Q. Yang, Y. Du, G. Feng, Z. Liu, Y. Li and R. Wang, *J. Chem. Inf. Model.*, 2018, 895–913.
152 Y. Li, M. Su, Z. Liu, J. Li, J. Liu, L. Han and R. Wang, *Nat. Protoc.*, 2018, **13**, 666.
153 Z. Gaieb, C. D. Parks, M. Chiu, H. Yang, C. Shao, W. P. Walters, M. H. Lambert, N. Nevins, S. D. Bembenek and M. K. Ameriks, *et al.*, *J. Comput.-Aided Mol. Des.*, 2019, **33**, 1–18.
154 G. Jones, P. Willett and R. C. Glen, *J. Mol. Biol.*, 1995, **245**, 43–53.
155 G. Jones, P. Willett, R. C. Glen, A. R. Leach and R. Taylor, *J. Mol. Biol.*, 1997, **267**, 727–748.
156 A. N. Jain, *J. Med. Chem.*, 2003, **46**, 499–511.
157 A. N. Jain, *J. Comput. Aided Mol. Des.*, 2007, **21**, 281–306.
158 M. Rarey, B. Kramer, T. Lengauer and G. Klebe, *J. Mol. Biol.*, 1996, **261**, 470–489.
159 C. M. Venkatachalam, X. Jiang, T. Oldfield and M. Waldman, *J. Mol. Graphics Modell.*, 2003, **21**, 289–307.
160 P. J. Ballester and J. B. O. Mitchell, *Bioinformatics*, 2010, **26**, 1169–1175.
161 G.-B. Li, L.-L. Yang, W.-J. Wang, L.-L. Li and S.-Y. Yang, *J. Chem. Inf. Model.*, 2013, **53**, 592–600.
162 H. Li, K.-S. Leung, M.-H. Wong and P. J. Ballester, *Mol. Inf.*, 2015, **34**, 115–126.
163 C. Wang and Y. Zhang, *J. Comput. Chem.*, 2017, **38**, 169–177.
164 H. Li, K.-S. Leung, M.-H. Wong and P. J. Ballester, *Molecules*, 2015, **20**, 10947–10962.
165 M. Wójcikowski, M. Kukiełka, M. Stepniewska-Dziubinska and P. Siedlecki, *Bioinformatics*, 2019, **35**, 1334–1341.
166 J. Jiménez, M. Skalic, G. Martnez-Rosell and G. De Fabritiis, *J. Chem. Inf. Model.*, 2018, **58**, 287–296.
167 M. M. Stepniewska-Dziubinska, P. Zielenkiewicz and P. Siedlecki, *Bioinformatics*, 2018, **1**, 9.
168 D. Plewczynski, M. Łaźniewski, R. Augustyniak and K. Ginalski, *J. Comput. Chem.*, 2011, **32**, 742–755.
169 J. Gabel, J. Desaphy and D. Rognan, *J. Chem. Inf. Model.*, 2014, **54**, 2807–2815.
170 M. A. Khamis and W. Gomaa, *Eng. Appl. Artif. Intell.*, 2015, **45**, 136–151.
171 O. Trott and A. J. Olson, *J. Comput. Chem.*, 2010, **31**, 455–461.
172 M. Schmidt and H. Lipson, *Science*, 2009, **324**, 81–85.