

GRAPH EDIT DISTANCE DETERMINATION IN DRUG-LIKE CHEMICAL SPACES

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims priority to United States Provisional Patent Application No. 63/492,200 entitled “GRAPH EDIT DISTANCE DETERMINATION IN DRUG-LIKE CHEMICAL SPACES,” filed March 24, 2023, which is hereby incorporated by reference.

[0002] This application also claims priority to United States Provisional Patent Application No. 63/492,208 entitled “IMPROVED BINDING SITE MODELS,” filed March 24, 2023, which is hereby incorporated by reference.

TECHNICAL FIELD

[0003] This application is directed to determining estimates for graph edit distance between compounds.

BACKGROUND

[0004] Figure 3A illustrates graph edit distance and how it is a measure of chemical similarity. A molecule can be represented as a graph in which atoms of the molecule are represented as nodes in the graph and bonds of the molecule are represented by edges in the graph. Graph edits comprise six elementary operations: (i) node addition, (ii) node deletion, (iii) node substitution or relabeling, (iv) edge addition, (v) edge deletion, and (vi) edge substitution or relabeling. Each of these six elementary operations can be associated with a cost of the operation. A path defines a sequence of elementary operations that will transform one graph to another. The summation of the cost of each of the elementary operations in a path defines the cost of the path. The graph edit distance is the path with the minimum cost. Thus, if there are two or more (sets of elementary operations) that can transform one graph to another, the graph edit distance is defined as the path with the lowest cost. An example of this is shown in Figure 3A, where a pyridine is transformed to an aniline. While there may be multiple paths to perform the transformation, the path that has the lowest cost would be the graph edit distance. For the transformation of pyridine to aniline, the path with the least costs is an N to C substitution, addition of an N atom, and addition of a C-N bond. This will

transform the graph for pyridine to the graph for aniline. If each of these three elementary operations had a cost of one, the sum of the costs for the transition from pyridine to aniline, the graph edit distance, would be 3.

[0005] Graph edit distance can be compared to other molecule similarity metrics such as fingerprint methods and maximum common substructures (MCS)-based methods. *See, Zhang et al., 2015, "Design of chemical space networks using a Tanimoto similarity variant based upon maximum common substructures," J. Comput. Aided Des. 29, 937, which is hereby incorporated by reference.* Fingerprint-based methods can be oversensitive to small changes in the molecule, particularly in instances where the change is in the middle of the molecule. Thus a small change in the middle of the molecule will lead to a large value for the fingerprint metric whereas the same change on the periphery of the molecule does not lead to a larger value for the fingerprint metric. By contrast, moving a small change from the periphery of a molecule to the middle of the molecule can reduce an MCS-metric by fifty percent. GED is sensitive to molecule changes regardless of where the change happens. Thus, GED produces similar values for small changes at the middle of the molecule and the periphery of the molecule. For example, if a carbon is changed to a nitrogen, the GED value is always 1, regardless of where in the molecule the change was made.

[0006] GED and MCS metrics are chemically intuitive, in that they track well with the actual chemical changes needed to transform one molecule to another whereas fingerprint methods typically are not. GED is chemically intuitive in the sense that it defines the transformations that take one molecule to another. MCS is chemically intuitive in a different sense. MCS says how much of the two molecules is common.

[0007] The drawback with GED is it is very computationally expensive to calculate. In fact, exact GED computation has been characterized as an NP-complete computational task. The computational expense scales unfavorably with molecular weight of the compounds being compared. For instance, for compounds with a molecular weight of 350 to 500 Daltons, a GED calculation is intractably expensive for large-scale usage in most drug discovery applications such as QSAR modeling, identifying chemically relevant analogs, *etc.* of a compound. Because of the computational expense, substitutes for exact GED computations, which approximate or estimate exact GED have been developed. One such GED approximation algorithm is called the Hungarian algorithm. *See, Bougleux et al., 2017, "A Hungarian Algorithm for Error-Correcting Graph Matching," Lecture Notes in Computer Science, Conference: International Workshop on Graph-Based Representations in Pattern*

Recognition (GbR), Anacapri, Capri, Italy, LNCS 10310, which is hereby incorporated by reference. The Hungarian algorithm scales favorably with molecular weight. However, the GED approximations produced by the Hungarian algorithm have unsatisfactory mean square error relative to true GED.

[0008] Another GED approximation approach is SimGNN. *See*, Bai *et al.* 2019, ACM International Conference on Web Search and Data Mining, which is hereby incorporated by reference. SimGNN is a general program for computing graph similarities, including similarities between compound graphs. SimGNN was trained on AIDS, LINUX program dependency graphs, and IMDB ego-network datasets. Of these three datasets, only the AIDS dataset pertains to drug like systems. As such, SimGNN addresses a general graph edit problem and can be used with graphs much larger than molecules. Moreover, much of what SimGNN was trained on is not representative of drug-like chemical spaces. As discussed in further detail below SimGNN GED approximations, like those produced by the Hungarian algorithm, have unsatisfactory mean square error when applied to drug-like applications.

[0009] Thus, given the above background what is needed in the art are computer systems and methods that perform improved GED approximations.

SUMMARY

[0010] The present disclosure addresses the problems identified in the background. Systems and methods for estimating graph edit distance (GED) between compounds are provided. A first graph representing a first compound comprises a plurality of nodes and a plurality of edges. Atoms of the first compound are represented by the nodes and bonds of the first compound are represented by the edges of the first graph. A second graph representing a second compound also comprises a plurality of nodes and a plurality of edges. Atoms of the second compound are represented by the nodes and bonds of the second compound are represented by edges of the second graph. The first graph is inputted into a model to generate a first isometry preserving latent embedding. The second graph is inputted into a second model to generate a second isometry preserving latent embedding. An estimate (GED) between the two compounds is provided using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

[0011] *A. Graph based embodiments.* Accordingly, one aspect of the present disclosure provides system and methods for estimating graph edit distance between a first compound and a second compound.

[0012] A first graph is obtained that represents the first compound. The first graph comprises a first plurality of nodes and a first plurality of edges. Each atom of the first compound is represented by a node in the first plurality of nodes. Each bond of the first compound is represented by an edge in the first plurality of edges.

[0013] In some embodiments, a second graph is obtained representing the second compound. The second graph comprises a second plurality of nodes and a second plurality of edges. Each atom of the second compound is represented by a node in the second plurality of nodes. Each bond of the second compound is represented by an edge in the second plurality of edges.

[0014] In some such embodiments, the first compound and the second compound have an exact graph edit distance of between 1 and 10.

[0015] In some such embodiments the first compound and the second compound have an exact graph edit distance of between 1 and 15.

[0016] In some embodiments the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons, less than 1000 Daltons, less than 2000 Daltons, less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons.

[0017] In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of between 400 Daltons and 10000 Daltons.

[0018] The first graph is inputted into a model comprising a plurality of parameters. The model applies the plurality of parameters of the model to the first graph (*e.g.*, through at least 10,000 computations, such as processor instructions) to generate as first output from the model a first isometry preserving latent embedding that represents the first compound.

[0019] The second graph is inputted into the model. The second model applies the plurality of parameters to the second graph (*e.g.*, through at least 10,000 computations, such as processor instructions) to generate as second output from the model a second isometry preserving latent embedding that represents the second compound.

[0020] In some embodiments, the model is a message passing network.

- [0021] In some embodiments, the model is a graph neural network.
- [0022] In some embodiments, the model is a message passing graph neural network having between 3 and 8 layers.
- [0023] In some embodiments, the model is a message passing graph neural network consisting of 5 layers.
- [0024] In some embodiments, the model comprises 500,000 or more parameters, or 1×10^6 or more parameters.
- [0025] In some embodiments, the model performs at least 100,000 computations to compute the first isometry preserving latent embedding, and the model performs at least 100,000 computations to compute the second isometry preserving latent embedding.
- [0026] The estimate of the graph edit distance between the first compound and the second compound is provided using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.
- [0027] In some embodiments, the difference is an L2-norm difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding. In some embodiments, the first latent space is a first vector comprising a first plurality of vector values, the second latent space is a second vector comprising a second plurality of vector values, and the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the first vector and the second vector.
- [0028] In some embodiments, a determination is made as to whether the second compound is a chemically relevant analog of the first compound when the estimate of the graph edit distance falls below a threshold value (*e.g.*, less than 5, less than 4, less than 3, or less than 2).
- [0029] In some embodiments, the obtaining the first graph, obtaining the second graph, inputting the first graph, inputting the second graph, and providing the approximate GED is repeated for each respective unique pair of compounds in a database of compounds, where a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of estimated graph edit distances. The plurality of estimated graph edit distances are then clustered into a plurality of clusters by evaluating the plurality

of distances with a criterion function. Then, compounds are selected from one or more clusters in the plurality of clusters for evaluation in a drug discovery program.

[0030] In some embodiments, the estimate of the graph edit distance is used to build a quantitative structure activity relationship model that includes the first compound and the second compound.

[0031] In some embodiments, the estimate of the graph edit distance is used to determine an activity cliff with respect to a drug target.

[0032] Another aspect of the present disclosure provides a method for estimating a graph edit distance between a first compound and a second compound. The method comprises (A) obtaining a first graph representing the first compound, the first graph comprising a first plurality of nodes and a first plurality of edges. Each atom of the first compound is represented by a node in the first plurality of nodes and each bond of the first compound is represented by an edge in the first plurality of edges. The method further comprises (B) obtaining a second graph representing the second compound. The second graph comprises a second plurality of nodes and a second plurality of edges. Each atom of the second compound is represented by a node in the second plurality of nodes and each bond of the second compound is represented by an edge in the second plurality of edges. The method further comprises (C) inputting the first graph into a model comprising a plurality of parameters. The model applies the plurality of parameters to the first graph (*e.g.*, through at least 10,000 computations, such as processor instructions) to generate as first output from the model a first isometry preserving latent embedding that represents the first compound. The method further comprises (D) inputting the second graph into the model. The model applies the plurality of parameters to the second graph (*e.g.*, through at least 10,000 computations) to generate as second output from the model a second isometry preserving latent embedding that represents the second compound. The method further comprises (E) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

[0033] Another aspect of the present disclosure provides a non-transitory computer readable storage medium. The non-transitory computer readable storage medium stores instructions, which when executed by a computer system, cause the computer system to perform a method for estimating a graph edit distance between a first compound and a second

compound. The method comprises (A) obtaining a first graph representing the first compound. The first graph comprises a first plurality of nodes and a first plurality of edges. Each atom of the first compound is represented by a node in the first plurality of nodes and each bond of the first compound is represented by an edge in the first plurality of edges. The method further comprises (B) obtaining a second graph representing the second compound. The second graph comprises a second plurality of nodes and a second plurality of edges. Each atom of the second compound is represented by a node in the second plurality of nodes and each bond of the second compound is represented by an edge in the second plurality of edges. The method further comprises (C) inputting the first graph into a model comprising a plurality of parameters. The model applies the plurality of parameters to the first graph (*e.g.*, through at least 10,000 computations) to generate as first output from the model a first isometry preserving latent embedding that represents the first compound. The method further comprises (D) inputting the second graph into the model. The model applies the plurality of parameters to the second graph (*e.g.*, through at least 10,000 computations) to generate as second output from the model a second isometry preserving latent embedding that represents the second compound. The method further comprises (E) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

[0034] *B. Compound representation embodiments.* Another aspect of the present disclosure provides a computer system for estimating a graph edit distance between a first compound and a second compound. The computer system comprises one or more processors and memory addressable by the one or more processors. The memory stores at least one program for execution by the one or more processors. The at least one program comprises instructions for (A) inputting a first representation of the first compound into a model comprising a plurality of parameters. The model applies the plurality of parameters to the first representation of the first compound to generate, as first output from the model, a first isometry preserving latent embedding that represents the first compound. The at least one program further comprises instructions for (B) inputting a second representation of the second compound into the model. The model applies the plurality of parameters to the second representation of the second compound to generate, as second output from the model, a second isometry preserving latent embedding that represents the second compound. The at least one program further comprises instructions for (C) providing the estimate of the graph

edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

[0035] In some embodiments the difference is an L2-norm difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding. In some such embodiments, the first isometry preserving latent embedding is a first vector comprising a first plurality of vector values, the second latent isometry preserving embedding is a second vector comprising a second plurality of vector values, and the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the first vector and the second vector, wherein the model is trained such that the L2-norm of the difference between the first and second vector is an estimate of the GED between the first and the second molecule.

[0036] In some embodiments, the at least one program further comprises instructions for determining that the second compound is a chemically relevant analog of the first compound when the estimate of the graph edit distance falls below a threshold value.

[0037] In some embodiments, the at least one program further comprises instructions for repeating the inputting (A), inputting (B), and providing (C) for each respective unique pair of compounds in a database of compounds, where a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of graph edit distances. In some such embodiments, the plurality of graph edit distances is clustered into a plurality of clusters by evaluating the plurality of distances with a criterion function and compounds from one or more clusters in the plurality of clusters are selected for evaluation in a drug discovery program.

[0038] In some embodiments the at least one program further comprises instructions for using the estimate of the graph edit distance to build a quantitative structure activity relationship model that includes the first compound and the second compound.

[0039] In some embodiments, the at least one program further comprises instructions for using the estimate of the graph edit distance to determine an activity cliff with respect to a drug target.

[0040] In some embodiments, the model is a graph convolutional neural network.

[0041] In some embodiments, the model is a graph neural network (message passing network).

[0042] In some embodiments, the model is a message passing graph neural network consisting of between 3 and 8 layers.

[0043] In some embodiments, the model is a message passing graph neural network consisting of 5 layers.

[0044] In some embodiments, the model is an autoencoder. In some such embodiments, the autoencoder is a variational autoencoder, an adversarial autoencoder, a regularized autoencoder, a sparse autoencoder, or a metric learning autoencoder.

[0045] In some embodiments, the first representation is a first fingerprint of the first compound, and the second representation is a second fingerprint of the second compound.

[0046] In some embodiments, the first representation is a first plurality of descriptors of the first compound, and the second representation is a second plurality of descriptors of the second compound.

[0047] In some embodiments, the first representation is a first three-dimensional structural representation of the first compound, and the second representation is a second three-dimensional structural representation of the second compound.

[0048] In some embodiments, the first representation is a first tensor-based representation of the first compound, and the second representation is a second tensor-based representation of the second compound.

[0049] In some embodiments, the first representation is a first string representation of the first compound, and the second representation is a second string representation of the second compound.

[0050] In some embodiments, model comprises 500,000 or more parameters. In some embodiments, the model comprises 1×10^6 or more parameters.

[0051] In some embodiments, the first compound and the second compound each satisfy two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

[0052] In some embodiments, the first compound and the second compound have an exact graph edit distance of between 1 and 10. In some embodiments, the first compound and the second compound have an exact graph edit distance of between 1 and 15.

[0053] In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons. In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than 1000 Daltons. In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than 2000 Daltons. In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons. In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of between 300 Daltons and 1500 Daltons.

[0054] In some embodiments, the model performs at least 100,000 computations to compute the first isometry preserving latent embedding, and the model performs at least 100,000 computations to compute the second isometry preserving latent embedding.

[0055] Another aspect of the present disclosure provides a method for estimating a graph edit distance between a first compound and a second compound. The method comprises (*e.g.*, at a computer system comprising one or more processors memory addressable by the one or more processors), (A) inputting a first representation of the first compound into a model comprising a plurality of parameters, wherein the model applies the plurality of parameters to the first representation of the first compound to generate as first output from the model a first isometry preserving latent embedding that represents the first compound. The method further comprises (*e.g.*, at a computer system comprising one or more processors memory addressable by the one or more processors) (B) inputting a second representation of the second compound into the model, wherein the model applies the plurality of parameters to the second representation of the second compound to generate as second output from the model a second isometry preserving latent embedding that represents the second compound. The method further comprises (*e.g.*, at a computer system comprising one or more processors memory addressable by the one or more processors) (C) providing the estimate of the graph edit distance between the first compound and the second compound using a difference

between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

[0056] Another aspect of the present disclosure provides a non-transitory computer readable storage medium. The non-transitory computer readable storage medium stores instructions, which when executed by a computer system, cause the computer system to perform a method for estimating a graph edit distance between a first compound and a second compound. The method comprises (A) inputting a first representation of the first compound into a model comprising a plurality of parameters. The model applies the plurality of parameters to the first representation of the first compound to generate as first output from the model a first isometry preserving latent embedding that represents the first compound. The method further comprises (B) inputting a second representation of the second compound into the model. The model applies the plurality of parameters to the second representation of the second compound to generate as second output from the model a second isometry preserving latent embedding that represents the second compound. The method further comprises (C) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

BRIEF DESCRIPTION OF THE DRAWINGS

[0057] In the drawings, embodiments of the systems and methods of the present disclosure are illustrated by way of example. It is to be expressly understood that the description and drawings are only for the purpose of illustration and as an aid to understanding, and are not intended as a definition of the limits of the systems and methods of the present disclosure.

[0058] FIG. 1 illustrates a computer system in accordance with some embodiments of the present disclosure.

[0059] FIGS. 2A, 2B, 2C, and 2D illustrate methods for estimating a graph edit distance between a first compound and a second compound in accordance with some embodiments of the present disclosure in which optional elements are indicated by dashed line boxes.

[0060] FIG. 3A illustrates graphic edit distance and how is it a measure of chemical similarity in accordance with the prior art.

[0061] FIG. 3B illustrates desirable properties of a training set for training a model to approximate graph edit distance in accordance with some embodiments of the present disclosure.

[0062] FIG. 4 illustrates statistics for training and test drug-like datasets and training and test fragment datasets for training a model to approximate graph edit distance in accordance with some embodiments of the present disclosure.

[0063] FIG. 5 shows the performance of a prior art model SimGNN after training on a fragment dataset and drug-like dataset of the present disclosure in accordance with some embodiments of the present disclosure.

[0064] FIG. 6 shows the desired formal properties of a model for approximating graph edit distance, and how the prior art model SimGNN does not have some of these formal properties.

[0065] FIG. 7 is a schematic view of a system for approximating graph edit distance in accordance with some embodiments of the present disclosure.

[0066] FIG. 8 illustrates the performance, in terms of means squared error, of a system in accordance with the present disclosure, MolGEDNet, compared to the prior art model SimGNN at approximating graph edit distance against the test datasets of Figure 4.

[0067] FIG. 9 illustrates the performance, in terms of Spearman's ρ and Kendall's τ_B , of a system in accordance with the present disclosure, MolGEDNet, compared to the prior art model SimGNN at ranking graph edit distances using approximate graph edit distances, versus ranking graph edit distances using true graph edit distances for the test datasets of Figure 4.

[0068] FIG. 10 illustrates optimizing a system in accordance with the present disclosure by dimensioning a model with respect to graph embedding size, in terms of edge embedding size and node embedding size using mean squared error, Spearman's ρ and Kendall's τ_B across the test datasets of Figure 4 in accordance with some embodiments of the present disclosure.

[0069] FIG. 11 illustrates optimizing a system in accordance with the present disclosure by optimizing dropout between layers of the model during model hyperparameter training in accordance with some embodiments of the present disclosure.

[0070] FIG. 12 illustrates optimizing a system in accordance with the present disclosure by optimizing the number of layers of the model, in accordance with some embodiments of the present disclosure.

[0071] FIG. 13 illustrates the performance, in terms of mean squared error, Spearman's ρ and Kendall's τ_B , of a model in accordance with the present disclosure, MolGEDNet, after hyperparameter training for the test datasets of Fig. 4, in accordance with the present disclosure.

[0072] FIG. 14 stratifies graph edit distance error produced by the hyperparameter optimized model of Fig. 13, as a function of true graph edit distance, across the drug-like test dataset of FIG. 4 in accordance with an embodiment of the present disclosure.

[0073] FIG. 15 illustrates example pairs of compounds in the drug-like test dataset of FIG. 4 that each have a true graph edit distance of one, and the absolute error in approximate graph edit distance provided by the hyperparameter optimized system of FIG. 13 for these pairs of compounds.

[0074] FIG. 16 specifies the betweenness centrality of the node that differs in each the graphs for the example pairs of compounds of FIG. 15.

[0075] FIG. 17 illustrates the run time performance of the hyperparameter optimized model of Figure 13 as a function of the average molecular weight of compounds being compared in the drug-like test set of FIG. 14, compared to the Hungarian algorithm, in accordance with an embodiment of the present disclosure.

[0076] FIG. 18 illustrate details of the system of FIG. 7 in accordance with some embodiments of the present disclosure.

[0077] FIG. 19 illustrates how a message passing graph neural network that is hyperparameter optimized in accordance with Figure 13 performs node, edge, and graph updates with these optimal hyperparameters in accordance with an embodiment of the present disclosure.

[0078] FIG. 20 illustrates the message passing graph neural network details in accordance with an embodiment of the present disclosure.

[0079] FIG. 21 illustrates the difference between chemical and exact graph edit distance in accordance with the prior art.

[0080] FIG. 22 illustrates a computer system in accordance with some embodiments of the present disclosure.

[0081] FIGS. 23A, 23B, and 23C illustrate methods for training a model to predict potency of a test compound against an orthosteric site of a target protein, in accordance with some embodiments of the present disclosure.

[0082] FIG. 24 illustrates a compound selection flow.

[0083] FIG. 25 illustrates a time table and stages of releasing pretrained structure based models that serve as training data for binding site models.

[0084] FIG. 26 illustrates benchmarks that serve to ensure that binding models are learning true protein binding features.

[0085] FIG. 27 further illustrates benchmarks that serve to ensure that binding models are learning true protein binding features.

[0086] FIG. 28 illustrates making use of AlphaFold2 to obtain models of underrepresented therapeutically relevant protein classes.

[0087] FIG. 29A illustrates the dilemma that most activity measurements for compounds against targets cannot be reliably mapped to a specific binding site on the target.

[0088] FIG. 29B illustrates a way of determining which binding site on a target compounds bind to by first clustering compounds that bind to a given target into a plurality of clusters.

[0089] FIG. 30 illustrates mapping compounds whose binding sites to a given target are known to the clusters of Fig. 29A and 29B and the identification of clusters that are consistent in that all compounds mapped to such clusters map to the same binding site on the given target.

[0090] FIGS. 31A and 31B illustrate using known structures of targets to select models of targets whose structures have not been resolved and determining the sites on such targets to which particular compounds bind.

[0091] FIG. 32 illustrates how training binding site models using multi-site labeling of compounds against targets deconvolves structure activity relationship data in accordance with an embodiment of the present disclosure.

[0092] FIGS. 33A and 33B illustrate how training binding site models using multi-site labeling of compounds against targets deconvolves structure activity relationship data in accordance with an embodiment of the present disclosure.

[0093] Like reference numerals refer to corresponding parts throughout the several views of the drawings.

DETAILED DESCRIPTION

[0094] *Introduction.* The present disclosure provides systems and methods for estimating graph edit distance (GED) between compounds. A first graph representing a first compound comprises a plurality of nodes and a plurality of edges. Atoms of the first compound are represented by the nodes and bonds of the first compound are represented by the edges of the first graph. A second graph representing a second compound also comprises a plurality of nodes and a plurality of edges. Atoms of the second compound are represented by the nodes and bonds of the second compound are represented by edges of the second graph. The first graph is inputted into a model to generate a first isometry preserving latent embedding. The second graph is inputted into the model to generate a second isometry preserving latent embedding. An estimate (GED) between the two compounds is providing using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

[0095] Reference will now be made in detail to embodiments, examples of which are illustrated in the accompanying drawings. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present disclosure. However, it will be apparent to one of ordinary skill in the art that the present disclosure may be practiced without these specific details. In other instances, well-known methods, procedures, components, circuits, and networks have not been described in detail so as not to unnecessarily obscure aspects of the embodiments.

[0096] Unless defined otherwise, all technical and scientific terms used herein have the same meaning as is commonly understood by one of skill in the art to which this disclosure belongs. All patents and publications referred to herein are incorporated by reference in their entireties.

[0097] Plural instances may be provided for components, operations or structures described herein as a single instance. Finally, boundaries between various components,

operations, and data stores are somewhat arbitrary, and particular operations are illustrated in the context of specific illustrative configurations. Other forms of functionality are envisioned and may fall within the scope of the implementation(s). In general, structures and functionality presented as separate components in the example configurations may be implemented as a combined structure or component. Similarly, structures and functionality presented as a single component may be implemented as separate components. These and other variations, modifications, additions, and improvements fall within the scope of the implementation(s).

[0098] It will also be understood that, although the terms first, second, *etc.* may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first subject could be termed a second subject, and, similarly, a second subject could be termed a first subject, without departing from the scope of the present disclosure. The first subject and the second subject are both subjects, but they are not the same subject.

[0099] The terminology used in the present disclosure is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used in the description of the invention and the appended claims, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term “and/or” as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms “comprises” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[00100] As used herein, the term “if” may be construed to mean “when” or “upon” or “in response to determining” or “in accordance with a determination” or “in response to detecting,” that a stated condition precedent is true, depending on the context. Similarly, the phrase “if it is determined (that a stated condition precedent is true)” or “if (a stated condition precedent is true)” or “when (a stated condition precedent is true)” may be construed to mean “upon determining” or “in response to determining” or “in accordance with a determination” or “upon detecting” or “in response to detecting” that the stated condition precedent is true, depending on the context.

[00101] Furthermore, when a reference number is given an “ i^{th} ” denotation, the reference number refers to a generic component, set, or embodiment. For instance, a cellular-component termed “cellular-component i ” refers to the i^{th} cellular-component in a plurality of cellular-components.

[00102] The foregoing description included example systems, methods, techniques, instruction sequences, and computing machine program products that embody illustrative implementations. For purposes of explanation, numerous specific details are set forth in order to provide an understanding of various implementations of the inventive subject matter. It will be evident, however, to those skilled in the art that implementations of the inventive subject matter may be practiced without these specific details. In general, well-known instruction instances, protocols, structures and techniques have not been shown in detail.

[00103] The foregoing description, for purpose of explanation, has been described with reference to specific implementations. However, the illustrative discussions below are not intended to be exhaustive or to limit the implementations to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The implementations are chosen and described in order to best explain the principles and their practical applications, to thereby enable others skilled in the art to best utilize the implementations and various implementations with various modifications as are suited to the particular use contemplated.

[00104] In the interest of clarity, not all of the routine features of the implementations described herein are shown and described. It will be appreciated that, in the development of any such actual implementation, numerous implementation-specific decisions are made in order to achieve the designer’s specific goals, such as compliance with use case- and business-related constraints, and that these specific goals will vary from one implementation to another and from one designer to another. Moreover, it will be appreciated that such a design effort might be complex and time-consuming, but nevertheless be a routine undertaking of engineering for those of ordering skill in the art having the benefit of the present disclosure.

[00105] Some portions of this description describe the embodiments of the invention in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are commonly used by those skilled in the data processing arts to convey the substance of their work effectively to others skilled in the art.

These operations, while described functionally, computationally, or logically, are understood to be implemented by computer programs or equivalent electrical circuits, microcode, or the like.

[00106] The language used in the specification has been principally selected for readability and instructional purposes, and it may not have been selected to delineate or circumscribe the inventive subject matter. It is therefore intended that the scope of the invention be limited not by this detailed description, but rather by any claims that issue on an application based hereon. Accordingly, the disclosure of the embodiments of the invention is intended to be illustrative, but not limiting, of the scope of the invention.

[00107] In general, terms used in the claims and the specification are intended to be construed as having the plain meaning understood by a person of ordinary skill in the art. Certain terms are defined below to provide additional clarity. In case of conflict between the plain meaning and the provided definitions, the provided definitions are to be used.

[00108] Any terms not directly defined herein shall be understood to have the meanings commonly associated with them as understood within the art of the invention. Certain terms are discussed herein to provide additional guidance to the practitioner in describing the compositions, devices, methods and the like of aspects of the invention, and how to make or use them. It will be appreciated that the same thing may be said in more than one way. Consequently, alternative language and synonyms may be used for any one or more of the terms discussed herein. No significance is to be placed upon whether or not a term is elaborated or discussed herein. Some synonyms or substitutable methods, materials and the like are provided. Recital of one or a few synonyms or equivalents does not exclude use of other synonyms or equivalents, unless it is explicitly stated. Use of examples, including examples of terms, is for illustrative purposes only and does not limit the scope and meaning of the aspects of the invention herein.

[00109] For the avoidance of doubt, it is intended herein that particular features (for example integers, characteristics, values, uses, diseases, formulae, compounds or groups) described in conjunction with a particular aspect, embodiment or example of the disclosure are to be understood as applicable to any other aspect, embodiment or example described herein unless incompatible therewith. Thus such features may be used where appropriate in conjunction with any of the definition, claims or embodiments defined herein. All of the features disclosed in this specification (including any accompanying claims, abstract and

drawings), and/or all of the steps of any method or process so disclosed, may be combined in any combination, except combinations where at least some of the features and/or steps are mutually exclusive. The disclosure is not restricted to any details of any disclosed embodiments. The disclosure extends to any novel one, or novel combination, of the features disclosed in this specification (including any accompanying claims, abstract and drawings), or to any novel one, or any novel combination, of the steps of any method or process so disclosed.

[00110] *Definitions.*

[00111] As used herein, the terms “compound” and “molecule” are used interchangeably.

[00112] As used herein, the term “about” means that dimensions, sizes, formulations, parameters, shapes and other quantities and characteristics are not and need not be exact, but may be approximate and/or larger or smaller, as desired, reflecting tolerances, conversion factors, rounding off, measurement error and the like, and other factors known to those of skill in the art. In general, a dimension, size, formulation, parameter, shape or other quantity or characteristic is “about” or “approximate” whether or not expressly stated to be such. It is noted that embodiments of very different sizes, shapes and dimensions may employ the described arrangements.

[00113] As used herein the term “fingerprint” as in a fingerprint of a compound is a digital digest of the compound. Nonlimiting examples of such a digital digest include Daylight fingerprints, a BCI fingerprint, an ECFC4 fingerprint, an ECFP4 fingerprint, an EcFC fingerprint, an MDL fingerprint, an atom pair fingerprint (APFP fingerprint), a topological torsion fingerprint (TTFP) fingerprint, a UNITY 2D fingerprint, an RNNS2S fingerprint, a GraphConv fingerprint. *See* Franco, 2014, “The Use of 2D fingerprint methods to support the assessment of structural similarity in orphan drug legislation,” J. Cheminform 6, p. 5, and Rensi and Altman, 2017, “Flexible Analog Search with Kernel PCA Embedded Molecule Vectors,” Computational and Structural Biotechnology Journal, doi:10.1016/j.csbj.2017.03.003, each of which is hereby incorporated by reference. *See also* Raymond and Willett, 2002, “Effectiveness of graph-based and fingerprint-based similarity measures for virtual screening of 2D chemical structure databases,” Journal of Computer-Aided Molecular Design 16, 59-71, and Franco *et al.*, 2014, “The use of 2D fingerprint methods to support the assessment of structural similarity in orphan drug legislation” Journal of chemoinformatics 6(5), each of which is hereby incorporated by reference.

[00114] As used interchangeably herein, the term “classifier”, “model”, algorithm, “regressor”, and/or “classifier” refers to a machine learning model or algorithm. In some embodiments, a model is an unsupervised learning. In some embodiments, a model is supervised machine learning. Nonlimiting examples of supervised learning include, but are not limited to, logistic regression, neural networks, support vector machines, Naive Bayes algorithms, nearest neighbor algorithms, random forest algorithms, decision trees, boosted trees, multinomial logistic regression, linear models, linear regression, GradientBoosting, mixture models, hidden Markov models, Gaussian NB algorithms, linear discriminant analysis, or any combinations thereof. In some embodiments, a model is a multinomial classifier algorithm. In some embodiments, a model is a 2-stage stochastic gradient descent (SGD) model. In some embodiments, a model is a deep neural network (*e.g.*, a deep-and-wide sample-level model). In some embodiments, a classifier or model of the present disclosure has 25 or more, 100 or more, 1000 or more, 10,000 or more, 100,000 or more or 1×10^6 or more parameters and thus the calculations of the model cannot be mentally performed.

[00115] Moreover, as used herein, the term “parameter” refers to any coefficient or, similarly, any value of an internal or external element (*e.g.*, a weight and/or a hyperparameter) in an algorithm, model, regressor, and/or classifier that can affect (*e.g.*, modify, tailor, and/or adjust) one or more inputs, outputs, and/or functions in the algorithm, model, regressor and/or classifier. For example, in some embodiments, a parameter refers to any coefficient, weight, and/or hyperparameter that can be used to control, modify, tailor, and/or adjust the behavior, learning, and/or performance of an algorithm, model, regressor, and/or classifier. In some instances, a parameter is used to increase or decrease the influence of an input (*e.g.*, a feature) to an algorithm, model, regressor, and/or classifier. As a nonlimiting example, in some embodiments, a parameter is used to increase or decrease the influence of a node (*e.g.*, of a neural network), where the node includes one or more activation functions. Assignment of parameters to specific inputs, outputs, and/or functions is not limited to any one paradigm for a given algorithm, model, regressor, and/or classifier but can be used in any suitable algorithm, model, regressor, and/or classifier architecture for a desired performance. In some embodiments, a parameter has a fixed value. In some embodiments, a value of a parameter is manually and/or automatically adjustable. In some embodiments, a value of a parameter is modified by a validation and/or training process for an algorithm, model, regressor, and/or classifier (*e.g.*, by error minimization and/or

backpropagation methods). In some embodiments, an algorithm, model, regressor, and/or classifier of the present disclosure includes a plurality of parameters. In some embodiments, the plurality of parameters is n parameters, where: $n \geq 2$; $n \geq 5$; $n \geq 10$; $n \geq 25$; $n \geq 40$; $n \geq 50$; $n \geq 75$; $n \geq 100$; $n \geq 125$; $n \geq 150$; $n \geq 200$; $n \geq 225$; $n \geq 250$; $n \geq 350$; $n \geq 500$; $n \geq 600$; $n \geq 750$; $n \geq 1,000$; $n \geq 2,000$; $n \geq 4,000$; $n \geq 5,000$; $n \geq 7,500$; $n \geq 10,000$; $n \geq 20,000$; $n \geq 40,000$; $n \geq 75,000$; $n \geq 100,000$; $n \geq 200,000$; $n \geq 500,000$, $n \geq 1 \times 10^6$, $n \geq 5 \times 10^6$, or $n \geq 1 \times 10^7$. As such, the algorithms, models, regressors, and/or classifiers of the present disclosure cannot be mentally performed. In some embodiments n is between 10,000 and 1×10^7 , between 100,000 and 5×10^6 , or between 500,000 and 1×10^6 . In some embodiments, the algorithms, models, regressors, and/or classifier of the present disclosure operate in a k -dimensional space, where k is a positive integer of 5 or greater (*e.g.*, 5, 6, 7, 8, 9, 10, *etc.*). As such, the algorithms, models, regressors, and/or classifiers of the present disclosure cannot be mentally performed.

[00116] *Neural networks.* In some embodiments, the model is a neural network (*e.g.*, a convolutional neural network and/or a residual neural network). Neural networks, also known as artificial neural networks (ANNs), include convolutional and/or residual neural networks (deep learning models). In some embodiments, neural networks are machine learning models that are trained to map an input dataset to an output dataset, where the neural network includes an interconnected group of nodes organized into multiple layers of nodes. For example, in some embodiments, the neural network architecture includes at least an input layer, one or more hidden layers, and an output layer. In some embodiments, the neural network includes any total number of layers, and any number of hidden layers, where the hidden layers function as trainable feature extractors that allow mapping of a set of input data to an output value or set of output values. In some embodiments, a deep learning model is a neural network including a plurality of hidden layers, *e.g.*, two or more hidden layers. In some instances, each layer of the neural network includes a number of nodes (or “neurons”). In some embodiments, a node receives input that comes either directly from the input data or the output of nodes in previous layers, and performs a specific operation, *e.g.*, a summation operation. In some embodiments, a connection from an input to a node is associated with a parameter (*e.g.*, a weight and/or weighting factor). In some embodiments, the node sums up the products of all pairs of inputs, x_i , and their associated parameters. In some embodiments, the weighted sum is offset with a bias, b . In some embodiments, the output of a node or neuron is gated using a threshold or activation function, f , which, in some instances, is a

linear or non-linear function. In some embodiments, the activation function is, for example, a rectified linear unit (ReLU) activation function, a Leaky ReLU activation function, or other function such as a saturating hyperbolic tangent, identity, binary step, logistic, arcTan, softsign, parametric rectified linear unit, exponential linear unit, softPlus, bent identity, softExponential, Sinusoid, Sine, Gaussian, or sigmoid function, or any combination thereof.

[00117] In some implementations, the parameters (weighting factors, bias values, and/or threshold values, or other computational parameters) of the neural network, are “taught” or “learned” in a training phase using one or more sets of training data. For example, in some implementations, the parameters are trained using the input data from a training dataset and a gradient descent or backward propagation method so that the output value(s) that the ANN computes are consistent with the examples included in the training dataset. In some embodiments, the parameters are obtained from a back propagation neural network training process.

[00118] Any of a variety of neural networks are suitable for use in accordance with the present disclosure. Examples include, but are not limited to, feedforward neural networks, radial basis function networks, recurrent neural networks, residual neural networks, convolutional neural networks, residual convolutional neural networks, and the like, or any combination thereof. In some embodiments, the machine learning makes use of a pre-trained and/or transfer-learned ANN or deep learning architecture. In some implementations, convolutional and/or residual neural networks are used, in accordance with the present disclosure.

[00119] For instance, a deep neural network model includes an input layer, a plurality of individually parameterized (*e.g.*, weighted) convolutional layers, and an output scorer. The parameters (*e.g.*, weights) of each of the convolutional layers as well as the input layer contribute to the plurality of parameters (*e.g.*, weights) associated with the deep neural network model. In some embodiments, at least 50 parameters, at least 100 parameters, at least 1000 parameters, at least 2000 parameters or at least 5000 parameters are associated with the deep neural network model. As such, deep neural network models require a computer to be used because they cannot be mentally solved. In other words, given an input to the model, the model output needs to be determined using a computer rather than mentally in such embodiments. *See*, for example, Krizhevsky *et al.*, 2012, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 2*, Pereira, Burges, Bottou, Weinberger, eds., pp. 1097-1105, Curran Associates, Inc.; Zeiler,

2012 “ADADELTA: an adaptive learning rate method,” CoRR, vol. abs/1212.5701; and Rumelhart *et al.*, 1988, “Neurocomputing: Foundations of research,” ch. Learning Representations by Back-propagating Errors, pp. 696-699, Cambridge, MA, USA: MIT Press, each of which is hereby incorporated by reference.

[00120] Neural networks, including convolutional neural networks, suitable for use as models are disclosed in, for example, Vincent *et al.*, 2010, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” J Mach Learn Res 11, pp. 3371-3408; Larochelle *et al.*, 2009, “Exploring strategies for training deep neural networks,” J Mach Learn Res 10, pp. 1-40; and Hassoun, 1995, Fundamentals of Artificial Neural Networks, Massachusetts Institute of Technology, each of which is hereby incorporated by reference. Additional example neural networks suitable for use as models are disclosed in Duda *et al.*, 2001, *Pattern Classification*, Second Edition, John Wiley & Sons, Inc., New York; and Hastie *et al.*, 2001, *The Elements of Statistical Learning*, Springer-Verlag, New York, each of which is hereby incorporated by reference in its entirety. Additional example neural networks suitable for use as models are also described in Draghici, 2003, *Data Analysis Tools for DNA Microarrays*, Chapman & Hall/CRC; and Mount, 2001, *Bioinformatics: sequence and genome analysis*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York, each of which is hereby incorporated by reference in its entirety.

[00121] *Support vector machines.* In some embodiments, the model is a support vector machine (SVM). SVMs suitable for use as models are described in, for example, Cristianini and Shawe-Taylor, 2000, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge; Boser *et al.*, 1992, “A training algorithm for optimal margin classifiers,” in Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, ACM Press, Pittsburgh, Pa., pp. 142-152; Vapnik, 1998, *Statistical Learning Theory*, Wiley, New York; Mount, 2001, *Bioinformatics: sequence and genome analysis*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, N.Y.; Duda, *Pattern Classification*, Second Edition, 2001, John Wiley & Sons, Inc., pp. 259, 262-265; and Hastie, 2001, *The Elements of Statistical Learning*, Springer, New York; and Furey *et al.*, 2000, *Bioinformatics* 16, 906-914, each of which is hereby incorporated by reference in its entirety. When used for classification, SVMs separate a given set of binary labeled data with a hyper-plane that is maximally distant from the labeled data. For certain cases in which no linear separation is possible, SVMs work in combination with the technique of ‘kernels’, which automatically realizes a non-linear mapping to a feature space. The hyper-plane found by the SVM in

feature space corresponds, in some instances, to a non-linear decision boundary in the input space. In some embodiments, the plurality of parameters (*e.g.*, weights) associated with the SVM define the hyper-plane. In some embodiments, the hyper-plane is defined by at least 10, at least 20, at least 50, or at least 100 parameters and the SVM model requires a computer to calculate because it cannot be mentally solved.

[00122] *Naïve Bayes models.* In some embodiments, the model is a Naive Bayes model. Naïve Bayes models suitable for use as models are disclosed, for example, in Ng *et al.*, 2002, “On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes,” *Advances in Neural Information Processing Systems*, 14, which is hereby incorporated by reference. A Naive Bayes model is any model in a family of “probabilistic models” based on applying Bayes’ theorem with strong (naïve) independence assumptions between the features. In some embodiments, they are coupled with Kernel density estimation. *See*, for example, Hastie *et al.*, 2001, *The elements of statistical learning: data mining, inference, and prediction*, eds. Tibshirani and Friedman, Springer, New York, which is hereby incorporated by reference.

[00123] *Nearest neighbors.* In some embodiments, a model is nearest neighbors. In some implementations, nearest neighbor models are memory-based and include no model to be fit. For nearest neighbors, given a query point x_0 (a test subject), the k training points $x_{(r)}$, $r = 1, \dots, k$ (here the training subjects) closest in distance to x_0 are identified and then the point x_0 is classified using the k nearest neighbors. In some embodiments, Euclidean distance in feature space is used to determine distance as $d_{(i)} = \|x_{(i)} - x_{(0)}\|$. Typically, when the nearest neighbor model is used, the abundance data used to compute the linear discriminant is standardized to have mean zero and variance 1. In some embodiments, the nearest neighbor rule is refined to address issues of unequal class priors, differential misclassification costs, and feature selection. Many of these refinements involve some form of weighted voting for the neighbors. For more information on nearest neighbor analysis, *see* Duda, *Pattern Classification*, Second Edition, 2001, John Wiley & Sons, Inc; and Hastie, 2001, *The Elements of Statistical Learning*, Springer, New York, each of which is hereby incorporated by reference.

[00124] A k -nearest neighbor model is a non-parametric machine learning method in which the input consists of the k closest training examples in feature space. The output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive

integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor. See, Duda *et al.*, 2001, *Pattern Classification*, Second Edition, John Wiley & Sons, which is hereby incorporated by reference. In some embodiments, the number of distance calculations needed to solve the k-nearest neighbor model is such that a computer is used to solve the model for a given input because it cannot be mentally performed.

[00125] *Random forest, decision tree, and boosted trees.* In some embodiments, the model is a decision tree. Decision trees suitable for use as models are described generally by Duda, 2001, *Pattern Classification*, John Wiley & Sons, Inc., New York, pp. 395-396, which is hereby incorporated by reference. Tree-based methods partition the feature space into a set of rectangles, and then fit a model (like a constant) in each one. In some embodiments, the decision tree is random forest regression. For example, one specific model is a classification and regression tree (CART). Other specific decision tree models include, but are not limited to, ID3, C4.5, MART, and Random Forests. CART, ID3, and C4.5 are described in Duda, 2001, *Pattern Classification*, John Wiley & Sons, Inc., New York, pp. 396-408 and pp. 411-412, which is hereby incorporated by reference. CART, MART, and C4.5 are described in Hastie *et al.*, 2001, *The Elements of Statistical Learning*, Springer-Verlag, New York, Chapter 9, which is hereby incorporated by reference in its entirety. Random Forests are described in Breiman, 1999, "Random Forests--Random Features," Technical Report 567, Statistics Department, U.C. Berkeley, September 1999, which is hereby incorporated by reference in its entirety. In some embodiments, the decision tree model includes at least 10, at least 20, at least 50, or at least 100 parameters (*e.g.*, weights and/or decisions) and requires a computer to calculate because it cannot be mentally solved.

[00126] *Regression.* In some embodiments, the model uses a regression formula. In some embodiments, the regression formula is determined using any type of regression. For example, in some embodiments, the regression is logistic regression. In some embodiments, the regression is logistic regression with lasso, L2 or elastic net regularization. In some embodiments, those extracted features that have a corresponding regression coefficient that fails to satisfy a threshold value are pruned (removed from) consideration. In some embodiments, a generalization of the logistic regression model that handles multicategory responses is used as the model. Logistic regression is disclosed in Agresti, *An Introduction to Categorical Data Analysis*, 1996, Chapter 5, pp. 103-144, John Wiley & Son, New York, which is hereby incorporated by reference. In some embodiments, the model makes use of a regression model disclosed in Hastie *et al.*, 2001, *The Elements of Statistical Learning*,

Springer-Verlag, New York. In some embodiments, the logistic regression model includes at least 10, at least 20, at least 50, at least 100, or at least 1000 parameters (*e.g.*, weights) and requires a computer to calculate because it cannot be mentally solved.

[00127] *Linear discriminant analysis models.* In some embodiments the model is linear discriminant analysis (LDA), normal discriminant analysis (NDA), or discriminant function analysis. In some embodiments such models are a generalization of Fisher's linear discriminant, a method used in statistics, pattern recognition, and machine learning to find a linear combination of features that characterizes or separates two or more classes of objects or events. In some embodiments, the resulting combination is used as the model (linear model) in some embodiments of the present disclosure.

[00128] *Mixture model and Hidden Markov model.* In some embodiments, the model is a mixture model, such as that described in McLachlan *et al.*, Bioinformatics 18(3):413-422, 2002. In some embodiments, in particular, those embodiments including a temporal component, the model is a hidden Markov model such as described by Schliep *et al.*, 2003, Bioinformatics 19(1):i255-i263.

[00129] *Ensembles of models and boosting.* In some embodiments, an ensemble (two or more) of models is used. In some embodiments, a boosting technique such as AdaBoost is used in conjunction with many other types of learning algorithms to improve the performance of the model. In this approach, the output of any of the models disclosed herein, or their equivalents, is combined into a weighted sum that represents the final output of the boosted model. In some embodiments, the plurality of outputs from the models is combined using any measure of central tendency known in the art, including but not limited to a mean, median, mode, a weighted mean, weighted median, weighted mode, *etc.* In some embodiments, the plurality of outputs is combined using a voting method. In some embodiments, a respective model in the ensemble of models is weighted or unweighted.

[00130] As used herein, the term "latent space" refers to a multi-dimensional space where data points are represented after being encoded by a model. While such data points are multi-dimensional they are of lower dimension than the input to the model. As used herein, the latent space has a lower dimensionality compared to the original input space of the mode, allowing for more efficient storage and computation.

[00131] As used herein, the term "latent embedding" refers to the embedding of data points by a model into a lower-dimensional space.

[00132] As used herein, the term “vector” is an enumerated list of elements, such as an array of elements, where each element has an assigned meaning. As such, the term “vector” as used in the present disclosure is interchangeable with the term “tensor.” As an example, if a vector comprises the abundance counts, in a plurality of cells, for a respective cellular constituent, there exists a predetermined element in the vector for each one of the plurality of cells. For ease of presentation, in some instances a vector may be described as being one-dimensional. However, the present disclosure is not so limited. A vector of any dimension may be used in the present disclosure provided that a description of what each element in the vector represents is defined (*e.g.*, that element 1 represents abundance count of cell 1 of a plurality of cells, *etc.*).

[00133] *Example computer system*

[00134] Figure 1 illustrates a computer system 100 for estimating a graph edit distance between a first compound and a second compound.

[00135] Referring to Figure 1, in typical embodiments, computer system 100 comprises one or more computers. For purposes of illustration in Figure 1, the computer system 100 is represented as a single computer that includes all of the functionality of the disclosed computer system 100. However, the present disclosure is not so limited. The functionality of the computer system 100 may be spread across any number of networked computers and/or reside on each of several networked computers and/or virtual machines. One of skill in the art will appreciate that a wide array of different computer topologies is possible for the computer system 100 and all such topologies are within the scope of the present disclosure.

[00136] Turning to Figure 1 with the foregoing in mind, the computer system 100 comprises one or more processing units (CPUs and/or GPUs) 59, a network or other communications interface 84, a user interface 78 (*e.g.*, including an optional display 82 and optional keyboard 80 or other form of input device), a memory 92 (*e.g.*, random access memory, persistent memory, or combination thereof), one or more magnetic disk storage and/or persistent devices 90 optionally accessed by one or more controllers 88, one or more communication busses 12 for interconnecting the aforementioned components, and a power supply 79 for powering the aforementioned components. To the extent that components of memory 92 are not persistent, data in memory 92 can be seamlessly shared with non-volatile memory 90 or portions of memory 92 that are non-volatile / persistent using known computing techniques such as caching. Memory 92 and/or memory 90 can include mass

storage that is remotely located with respect to the central processing unit(s) 59. In other words, some data stored in memory 92 and/or memory 90 may in fact be hosted on computers that are external to computer system 100 but that can be electronically accessed by the computer system 100 over an Internet, intranet, or other form of network or electronic cable using network interface 84. In some embodiments, the computer system 100 makes use of models that are run from the memory associated with one or more graphical processing units in order to improve the speed and performance of the system. In some alternative embodiments, the computer system 100 makes use of models that are run from memory 92 rather than memory associated with a graphical processing unit.

[00137] In some embodiments, the computer system 100 presents media to a user through the display 82. Examples of media presented by the display 82 include one or more images, a video, audio (*e.g.*, waveforms of an audio sample), or a combination thereof. In typical embodiments, the one or more images, the video, the audio, or the combination thereof is presented by the display 82 through a client application (*e.g.*, a client application hosted by the computer system 100, a client application accessible through the computer system 100, etc.). In some embodiments, the audio is presented through an external device (*e.g.*, speakers, headphones, input/output (I/O) subsystem, *etc.*) that receives audio information from the computer system 100 and presents audio data based on this audio information. In some embodiments, the user interface 78 also includes an audio output device, such as speakers or an audio output for connecting with speakers, earphones, or headphones.

[00138] The memory 92 includes high-speed random access memory, such as DRAM, SRAM, DDR RAM, or other random access solid state memory devices, and optionally also includes non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. The memory 92 may optionally include one or more storage devices remotely located from the CPU(s) 59. The memory 92, or alternatively the non-volatile memory device(s) within memory 92, includes a non-transitory computer readable storage medium. Access to memory 92 by other components of the computer system 100, such as the CPU(s) 59, is, optionally, controlled by a controller 88. In some embodiments, the memory 92 include mass storage that is remotely located with respect to the CPU(s) 59. In other words, some data stored in the memory 93 may in fact be hosted on devices that are external to the computer system 100, but that can be electronically accessed by the computer system 100 over an Internet, intranet, or other form of network or electronic cable using network interface 84.

[00139] The memory 92 of the computer system 100 stores:

- an optional operating system 30 (*e.g.*, ANDROID, iOS, DARWIN, RTXC, LINUX, UNIX, OS X, WINDOWS, or an embedded operating system such as VxWorks)) that includes procedures for handling various basic system services;
- an approximate graph editing distance (GED) computation module 32 for spatial data evaluation module 36 for estimating a graph edit distance between a first compound and a second compound;
- an identity of a first compound 34;
- a graph 36 of the first compound 34, the graph comprising a plurality of nodes (38-1, ..., 38-N, where N is a positive integer of 2 or greater), and a plurality of edges (40-1, ..., 40-M, where M is a positive integer of 2 or greater), each edge 40 identifying a source node (*e.g.*, 42-1-1) and a destination node 42-1-2) in the plurality of nodes, where each atom of the first compound 36 is represented by a node 38 in the plurality of nodes and each bond of the first compound is represented by an edge 40 in the plurality of edges;
- an identity of a second compound 44;
- a graph 46 of the second compound 44, the graph 46 comprising a plurality of nodes (48-1, ..., 48-X, where X is a positive integer of 2 or greater), and a plurality of edges (50-1, ..., 50-Y, where Y is a positive integer of 2 or greater), each edge 50 identifying a source node (*e.g.*, 52-1-1) and a destination node 52-1-2) in the plurality of nodes, where each atom of the second compound 46 is represented by a node 48 in the plurality of nodes and each bond of the second compound is represented by an edge 50 in the plurality of edges; and
- a training library 56 comprising pairs of training compounds 58-1, ..., 58-Q, where Q is a positive integer of 2 or greater, each pair of training compounds associated with a true graph edit distance 60.

[00140] In some implementations, one or more of the above identified data elements or modules of the computer system 100 are stored in one or more of the previously mentioned memory devices, and correspond to a set of instructions for performing a function described above. The above identified data, modules or programs (*e.g.*, sets of instructions) need not be implemented as separate software programs, procedures or modules, and thus various subsets of these modules may be combined or otherwise re-arranged in various implementations. In some implementations, the memory 92 and/or 90 (and optionally 52) optionally stores a

subset of the modules and data structures identified above. Furthermore, in some embodiments the memory 92 and/or 90 (and optionally 52) stores additional modules and data structures not described above.

[00141] *Example methods*

[00142] Now that a system for estimating a graph edit distance between a first compound and a second compound has been disclosed, methods for performing such characterization is detailed with reference to Figure 2 and discussed below.

[00143] *Block 200.* Referring to block 200, systems and methods for estimating graph edit distance between a first compound 34 and a second compound 44 are provided.

[00144] *Block 202.* Referring to block 202, a first graph 36 is optionally obtained that represents the first compound. The first graph comprises a first plurality of nodes and a first plurality of edges. Each atom of the first compound is represented by a node 38 in the first plurality of nodes. Each bond of the first compound is represented by an edge 40 in the first plurality of edges.

[00145] *Block 204.* Referring to block 204, a second graph 46 is optionally obtained representing the second compound 44. The second graph 46 comprises a second plurality of nodes and a second plurality of edges. Each atom of the second compound is represented by a node 48 in the second plurality of nodes. Each bond of the second compound is represented by an edge 50 in the second plurality of edges.

[00146] *Block 206.* Referring to block 206, in some embodiments, the first compound and the second compound each satisfy two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5. *See*, Lipinski, 1997, Adv. Drug Del. Rev. 23, 3, which is hereby incorporated herein by reference in its entirety. In some embodiments, the first compound and the second compound each satisfy one or more criteria in addition to Lipinski's Rule of Five. For example, in some embodiments, the first compound and the second compound each have five or fewer aromatic rings, four or fewer aromatic rings, three or fewer aromatic rings, or two or fewer aromatic rings.

[00147] *Blocks 208-210.* Referring to block 208, in some embodiments, the first compound and the second compound have an exact graph edit distance of between 1 and 10. Referring to block 210, in some embodiments, the first compound and the second compound

have an exact graph edit distance of between 1 and 15. In some embodiments, it is not known what the exact graph edit distance is between the first and second compound.

[00148] *Block 212-214.* Referring to block 212, in some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons, less than 1000 Daltons, less than 2000 Daltons, less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons. Referring to block 214, in some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of between 400 Daltons and 10000 Daltons.

[0001] In some embodiments the first compound and the second compound each have a molecular weight of at least 10 Da, at least 20 Da, at least 50 Da, at least 100 Da, at least 200 Da, at least 500 Da, at least 1 kDa, at least 2 kDa, at least 3 kDa, at least 5 kDa, at least 10 kDa, at least 20 kDa, at least 30 kDa, at least 50 kDa, at least 100 kDa, or at least 500 kDa. In some embodiments, the first compound and the second compound each have a molecular weight of no more than 1000 kDa, no more than 500 kDa, no more than 100 kDa, no more than 50 kDa, no more than 10 kDa, no more than 5 kDa, no more than 2 kDa, no more than 1 kDa, no more than 500 Da, no more than 300 Da, no more than 100 Da, or no more than 50 Da. In some embodiments, the first compound and the second compound each have a molecular weight of from 10 Da to 900 Da, from 50 Da to 1000 Da, from 100 Da to 2000 Da, from 1 kDa to 10 kDa, from 5 kDa to 500 kDa, or from 100 kDa to 1000 kDa. In some embodiments, the first compound and the second compound each have a molecular weight that falls within another range starting no lower than 10 Daltons and ending no higher than 1000 kDa.

[0002] In some embodiments, the first compound and the second are each a small molecule. For instance, in some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than approximately 1000 Daltons (*e.g.*, less than 900 Daltons).

[00149] *Block 216.* Referring to block 216, the first graph, or some other form of representation of the first compound, is inputted into a model comprising a plurality of parameters. The model applies the plurality of parameters to the first graph (*e.g.*, through at least 10,000 computations) to generate as first output from the model a first isometry preserving latent embedding that represents the first compound.

[00150] As illustrated in Figure 10, in some embodiments the first isometry preserving latent embedding has 64 dimensions, 96 dimensions, 128 dimensions, 192 dimensions, or 256 dimensions. In some embodiments, the first isometry preserving latent embedding has between 64 and 1024 dimensions. In some embodiments, the first isometry preserving latent embedding has between 64 and 512 dimensions.

[00151] In some embodiments, the model (*e.g.* model 702 of Figure 7) comprises more than 400 parameters, more than 1000 parameters, more than 2000 parameters, more than 5000 parameters, more than 10,000 parameters, more than 100,000 parameters, or more than 1×10^6 parameters.

[00152] In some embodiments the model comprises 10,000 or more parameters, 100,000 or more parameters, or 1×10^6 or more parameters. In some embodiments, the model comprises 1000 parameters. In some embodiments, the model consists of between 10 and 10 million parameters. In some embodiments, the model comprises at least 10, at least 50, at least 100, at least 500, at least 1000, at least 2000, at least 5000, at least 10,000, at least 20,000, at least 50,000, at least 100,000, at least 200,000, at least 500,000, at least 1 million, at least 2 million, at least 3 million, at least 4 million or at least 5 million parameters. In some embodiments, the model consists of no more than 8 million, no more than 5 million, no more than 4 million, no more than 1 million, no more than 500,000, no more than 100,000, no more than 50,000, no more than 10,000, no more than 5000, no more than 1000, or no more than 500 parameters. In some embodiments, the first model consists of from 10 to 5000, from 500 to 10,000, from 10,000 to 500,000, from 20,000 to 1 million, or from 1 million to 5 million parameters. In some embodiments, the number of parameters in the model falls within another range starting no lower than 10 parameters and ending no higher than 8 million parameters.

[00153] In some embodiments, the amount of information in the first graph, or other form of representation of the first compound, that inputted into the model 702 coupled with the number of the parameters of the model results in the performance of more than 10,000 computations, more than 100,000 computations, more than 1×10^6 computations, more than 5×10^6 computations, or more than 1×10^7 computations to calculate the first isometry preserving latent embedding.

[00154] In some embodiments, compounds are represented in forms other than graphs for inputting into a model of the present disclosure. For instance, in some embodiments,

compounds are represented as fingerprints, and it is the fingerprint of a compound that is inputted into a model in order to produce the isometry preserving latent embedding of the molecule. Nonlimiting examples of fingerprint representations of compounds have been given in the definitions section above.

[00155] In some embodiments, compounds are represented as descriptors, and it is the descriptor of a compound that is inputted into a model in order to produce the isometry preserving latent embedding of the compound. Descriptors include, but are not limited to, physicochemical descriptors, topological descriptors, electronic descriptors, geometric descriptors, functional group counts, and toxicological and environmental descriptors, and any combinations thereof.

[00156] Physicochemical descriptors include, but are not limited to, molecular weight (the sum of the atomic weights of all atoms in a compound), LogP, penalized logP, compound acid dissociation constant (pKA), compound acid dissociation constant (kA), compound topological polar surface area (TPSA), polar surface area (PSA) (the surface area of a molecule occupied by polar atoms and functional groups, often used to predict membrane permeability and bioavailability), solubility (predictions of a compound's solubility in different solvents or aqueous solutions), and drug likeness. *See*, for example, Prasanna and Doerksen *et al.*, 2009, "Topological Polar Surface Area: A Useful Descriptor in 2D-QSAR," *Current Medicinal Chemistry* 16(1), pp. 21-41, which is hereby incorporated by reference

[00157] LogD is a distribution constant of the lipophilicity of a compound. In some embodiments LogD is calculated from the chemical structure of a compound (cLogD). Generally, cLogD refers to the calculated distribution constant of the lipophilicity of a molecule. This is typically determined for a compound in an aqueous phase, which can be adjusted to a specific pH using a buffer. This parameter can serve as an estimate of a compound's overall lipophilicity, a value that influences the behavior of the compound in a range of biological processes relevant to a drug discovery, such as solubility, permeability through biological membranes, hepatic clearance, lack of selectivity and/or non-specific toxicity. In some instances, lipophilicity can be used as an estimate of druglikeness. *See*, Cambridge MedChem Consulting, 2019, "Lipophilicity," available online at cambridgemedchemconsulting.com/resources/physiochem/logD.html, which is hereby incorporated herein by reference in its entirety.

[00158] A logP is a measure of compound hydrophilicity. The partition coefficient, abbreviated P, is defined as a particular ratio of the concentrations of a solute between the two solvents (a biphasic of liquid phases), specifically for un-ionized solutes, and the logarithm of the ratio is thus log P. As used here, following convention, the numerator of this ratio is concentration of the compound in octanol and the denominator is concentration of the compound in water.

[00159] Penalized logP is a logP score that also accounts for ring size and synthetic accessibility. *See*, for example, Ertl, 2009, "Estimation of synthetic accessibility score of drug-like molecules," J. Cheminform, which is hereby incorporated by reference.

[00160] In some embodiments the pK_a is estimated using the structure of a compound. *See*, for example, Seybold and Shields, 2015, "Computational estimation of pK_a values," WIREs Comput Mol Sci 5, pp. 290–297, which is hereby incorporated by reference.

[00161] Compound topological polar surface area (TPSA) makes use of functional group contributions based on a large database of structures, is a measure of the polar surface area of a compound that avoids the need to calculate compound three-dimensional structure or to decide which is the relevant biological conformation or conformations. *See*, for example, Prasanna and Doerksen *et al.*, 2009, "Topological Polar Surface Area: A Useful Descriptor in 2D-QSAR," Current Medicinal Chemistry 16(1), pp. 21-41, which is hereby incorporated by reference.

[00162] Topological descriptors include, but are not limited to, atom and bond counts (the number of atoms and bonds in a compound), molecular connectivity indices (graph-based indices representing structural connectivity, such as the Wiener index, Balaban index, and Zagreb indices), and molecular diameter (measures of the size or compactness of a molecule, often computed as the maximum distance between any two atoms in the molecule). *See* Balaban and Ivanciuc, 1999, *Historical Development of Topological Indices*, CRC press, 1st Edition, which is hereby incorporated by reference.

[00163] Electronic descriptors include, but are not limited to, electronegativity (a measure of an atom's tendency to attract electrons in a chemical bond), ionization potential (the energy required to remove an electron from an atom or molecule), and electron density (distribution of electron charge within a molecule, often represented using molecular orbitals or electron density maps).

[00164] Geometric descriptors include, but are not limited to, bond lengths and angles (measurements of the lengths and angles between atoms in a molecule), torsional angles (measures of rotation around single bonds in a molecule), as well as surface area and volume (geometric properties characterizing the shape and size of a molecule).

[00165] Functional group count descriptors include, but are not limited to, counts of specific functional groups or substructures within a compound, such as hydroxyl groups, aromatic rings, or carboxylic acid groups.

[00166] Toxicological and environmental descriptors include, but are not limited to, toxicophores (substructural features associated with toxicity or adverse effects in compounds), environmental fate descriptors (predictions of a compound's behavior in the environment, including biodegradability, persistence, and bioaccumulation potential).

[00167] QED scores are described in Bickerton *et al.*, 2012, "Quantifying the chemical beauty of drugs," *Nature Chemistry* 4(2), pp. 90-98, which is hereby incorporated by reference.

[00168] Nonlimiting examples of descriptors of compounds are disclosed in Seko, 2017, "Representation of compounds for machine-learning prediction of physical properties," *Phys. Rev. B* 95, 144110, which is hereby incorporated by reference.

[00169] In some embodiments, compounds are represented as three-dimensional structural representations, and it is the three-dimensional structural representation of a compound that is inputted into a model in order to produce the isometry preserving latent embedding of the molecule. Nonlimiting examples of three-dimensional structural representations of compounds include, but are not limited to pharmacophore models (hypothetical representations of the spatial arrangement of chemical features, such as hydrogen bond donors, acceptors, aromatic rings, *etc.* of a compound), conformational ensembles (representations of a compound's multiple conformations or energetically accessible states), and quantum mechanical calculations produced by, for example, *ab initio* and density functional theory (DFT) methods that compute the electronic structure and energy of molecules, which can inform their 3D geometry.

[00170] In some embodiments, compounds are represented as tensor-based representations, and it is the tensor-based representation of a compound that is inputted into a model in order to produce the isometry preserving latent embedding of the molecule. Nonlimiting examples of tensor-based representations of compounds are disclosed in Huo

and Rupp, 2022, “Unified representation of molecules and crystals for machine learning,” Machine Learning 3, 045017, which is hereby incorporated by reference.

[00171] In some embodiments, compounds are represented as a string representation, and it is the string representation of a compound that is inputted into a model in order to produce the isometry preserving latent embedding of the molecule. Nonlimiting examples of string representations of compounds include, but are not limited to, Simplified Molecular Input Line Entry System (SMILES; *see*, for example, Honda *et al.*, 2019, “SMILES Transformer: Pre-trained Molecular Fingerprint for Low Data Drug Discovery,” arXiv:1911.04738), SMiles ARbitrary Target Specification (SMARTS; *see*, for example, SMARTS - A Language for Describing Molecular Patterns,” 2022 on the Internet at daylight.com/dayhtml/doc/theory/theory.smarts.html (accessed Dec. 2020), DeepSMILES (*see*, for example, O’Boyle and Dalke, 2018, “DeepSMILES: an adaptation of SMILES for use in machine-learning of chemical structures,” Preprint at ChemRxiv. <https://doi.org/10.26434/chemrxiv.7097960.v1>), and the self-referencing embedded string (SELFIES) (Krenn *et al.*, 2022, “SELFIES and the future of molecular string representations,” Patterns 3(10), pp. 1-27) format, which is hereby incorporated by reference.

[00172] *Block 218.* Referring to block 218, the second graph, or other form of representation of the first compound, is inputted into the model. While Figure 7 shows two models, 702 and 704, in typical embodiments models 702 and 704 are in fact the same model. That is, the first graph (or other form of representation) of the first compound is inputted into model 702 to get the first isometry preserving latent embedding and the second graph (or other form of representation) of the second compound is then inputted into model 702 to the second isometry preserving latent embedding. Two models 702 / 704 are depicted in Figure 7 to better illustrate the comparison of the model output. In some embodiments, the amount of information in the second graph (or other form of representation of the second compound) inputted into the model coupled with the number of the parameters of the model results in the performance of more than 10,000 computations, more than 100,000 computations, more than 1×10^6 computations, more than 5×10^6 computations, or more than 1×10^7 computations to calculate the second isometry preserving latent embedding.

[00173] As used herein, in some embodiments, a first isometry preserving latent embedding and a second isometry preserving latent embedding refer to a latent embedding that is produced by a model that adheres, within a threshold value, to the symmetry and distance-to-self requirements illustrated in Figure 6.

[00174] As used herein, in some embodiments, a first isometry preserving latent embedding and a second isometry preserving latent embedding refer to a latent embedding that is produced by a model that adheres, within a threshold value, to the symmetry property $|\hat{d}(x, y) - \hat{d}(y, x)|^2$ illustrated in Figure 6 across a test dataset of compounds such as the one used to compute the statistics illustrated in Figure 6. In some embodiments, this threshold value is 0.5 or less, 0.4 or less, 0.3 or less, or 0.2 or less. In some embodiments, this threshold value is a value between 0.1 and 0.5.

[00175] As used herein, in some embodiments, a first isometry preserving latent embedding and a second isometry preserving latent embedding refer to a latent embedding that is produced by a model that adheres, within a threshold value, to the distance-to-self property $|\hat{d}(x, x)|^2$ illustrated in Figure 6 across a test dataset of compounds such as the one used to compute the statistics illustrated in Figure 6. In some embodiments, this threshold value is 1.4 or less, 1.3 or less, 1.2 or less, 1.1 or less, 1.0 or less, 0.9 or less, 0.8 or less, 0.7 or less, 0.6 or less, or 0.5 or less. In some embodiments, this threshold value is a value between 0.1 and 1.5.

[00176] *Block 220-222.* Referring to block 220, in some embodiments, the model is a message passing network. Referring to block 222, in some embodiments, the model is a graph neural network.

[00177] In some embodiments, the model is any model that takes a representation of a compound and outputs an isometry preserving latent embedding that represents the compound.

[00178] Isometry preserving latent embeddings are those that maintain the geometric properties of the input space in the latent space. In some embodiments the model used to produce such isometry preserving latent embeddings is an autoencoder.

[00179] In some embodiments, the model is a variational autoencoder (VAE). VAEs are generative models that learn to encode and decode data points probabilistically. They are trained to maximize the likelihood of generating the input data and simultaneously minimize the divergence between the learned latent distribution and a prior distribution (usually Gaussian). By encouraging the latent space to follow a specific distribution (*e.g.*, Gaussian), VAEs implicitly aim to preserve local structure and distance relationships in the input space, which can lead to isometry preservation to some extent. See, for example, Kingma and Max,

2019, *Foundations and Trends in Machine Learning*, 12(4), ISSN 1935-8237, which is hereby incorporated by reference.

[00180] In some embodiments, the model is an adversarial autoencoder (AAE). AAEs combine the principles of variational autoencoders with adversarial training. In addition to reconstructing input data accurately, they also train a discriminator to distinguish between samples from the true latent distribution and samples from the encoder's distribution. The adversarial training component encourages the encoder to produce latent embeddings that are indistinguishable from samples drawn from the prior distribution, which can lead to improved isometry preservation.

[00181] In some embodiments, the model is a regularized autoencoder. Various regularization techniques can be applied to standard autoencoders to encourage isometry preservation in the latent space. For example, adding a penalty term to the loss function that penalizes deviations from a desired structure or distribution in the latent space. In some embodiments, as described in the examples section an L2 norm regularization is applied to the autoencoder to preserve isometry.

[00182] In some embodiments, the model is a sparse autoencoder, a denoising autoencoder (see, for example, Zheng, 2019, *Stacked Denoising Auto-Encoder for Short-Term Load Forecasting: Deep Learning with Stacked Denoising Auto-Encoder Algorithm*, Lambert Academic Publishing), or a contractive autoencoder. In some embodiments, such autoencoders indirectly encourage isometry preservation by promoting robust representations that capture salient features of the input data.

[00183] In some embodiments, the model is a metric learning autoencoder. As used herein, a metric learning autoencoder is explicitly designed to learn embeddings that preserve specific distance metrics or relationships in the input space. These models incorporate loss functions that directly encourage isometry preservation, such as triplet loss or contrastive loss. By optimizing these loss functions during training, the autoencoder learns to map similar input samples to nearby points in the latent space, thereby preserving local distances and geometric structure.

[00184] In some embodiments, the model is a graph neural network (GNN), also known as message passing network. See Baskin *et al.*, 1997, "A neural device for searching direct correlations between structures and properties of chemical compounds," *Journal of chemical information and computer sciences*, 37(4):715–721; ; Sperduti & Starita, 1997, "Supervised

neural networks for the classification of structures,” IEEE Transactions on Neural Networks 8(3), pp. 714–735; Scarselli *et al.*, 2008, “The graph neural network model,” IEEE Transactions on Neural Networks 20(1), pp. 61–80; and Gilmer *et al.*, 2017, “Message Passing Neural Networks,” In: Schütt *et al.* (eds), Machine Learning Meets Quantum Physics, Lecture Notes in Physics 968, Springer, Cham, each of which is hereby incorporated by reference.

[00185] In some embodiments, the model is a graph convolutional neural network (GCN). Generally, the message passing of GNNs is computed with nonlinear neural networks, which can incur a tremendous computational cost. In contrast, the GCN (Kipf and Welling, 2017) is a considerable simplification of a GNN, that uses a linear message passing scheme expressed as

$$\mathbf{H}_{\text{out}} = \sigma(\hat{\mathbf{A}}\mathbf{H}_{\text{in}}\mathbf{W})$$

where $\mathbf{H}_{\text{in}}$ ($\mathbf{H}_{\text{out}}$) is an input (output) feature of the l^{th} layer, $\hat{\mathbf{A}}$ is a renormalized adjacency matrix with self-loops, and $\mathbf{W}$ is a trainable weight. A GCN, among the variants of GNNs, has considerable computational efficiency compared to message passing of GNN computed with nonlinear neural networks. In some embodiments, a GCN is graph representation-invariant, as long as the initialization is carefully designed. It is also inductive, since for any unseen graph, the node embedding following the GCN operation can be computed. GCN is a node embedding model that belongs to the family of neighbor aggregation based models. Its core operation, graph convolution, operates on the representation of a node, which is denoted as

$$\text{conv}(\mathbf{u}_n) = f_1\left(\sum_{m \in \mathcal{N}(n)} \frac{1}{\sqrt{d_n d_m}} \mathbf{u}_m \mathbf{W}_1^{(l)} + \mathbf{b}_1^{(l)}\right)$$

where $\mathcal{N}(n)$ is the set of the first-order neighbors of node n plus n itself, d_n is the degree of node n plus 1, $\mathbf{W}_1^{(l)} \in \mathbb{R}^{D^l \times D^{l+1}}$ is the weight matrix associated with the l -th GCN layer, $\mathbf{b}_1^{(l)} \in \mathbb{R}^{D^{l+1}}$ is the bias, and $f_1(\cdot)$ is an activation function such as $\text{ReLU}(x) = \max(0, x)$. Intuitively, the graph convolution operation aggregates the features from the first-order neighbors of the node. See Bai *et al.*, 2019, “” ACM International Conference on Web Search and Data Mining, February 11–15, 2019, Melbourne, VIC, Australia, which is hereby incorporated by reference. In some embodiments, a graph of a compound is encoded using one-hot encoding based on node types (*e.g.*, all the nodes with carbon type share the same

one-hot encoding vector), so even if the node ids are permuted, the aggregation results would be the same. For graphs with unlabeled nodes, every node has the same label, resulting in the same constant number as the initialized representation. In some embodiments the GCN has 2, layers, 3 layers, 4 layers, 5 layers, 6 layers, or more than 7 layers.

[00186] Nonlimiting examples of GCNs are disclosed in Behler Parrinello, 2007, “Generalized Neural-Network Representation of High Dimensional Potential-Energy Surfaces,” *Physical Review Letters* 98, 146401; Chmiela et al., 2017, “Machine learning of accurate energy-conserving molecular force fields,” *Science Advances* 3(5):e1603015; Schütt et al., 2017, “SchNet: A continuous-filter convolutional neural network for modeling quantum interactions,” *Advances in Neural Information Processing Systems* 30, pp. 992-1002; Feinberg et al., 2018, “PotentialNet for Molecular Property Prediction,” *ACS Cent. Sci.* 4, 11, 1520–1530; and Stafford *et al.*, 2022, “AtomNet PoseRanker: Enriching Ligand Pose Quality for Dynamic Proteins in Virtual High Throughput Screens,” *J. Chem Inf Model* 62, pp. 1178–1189, each of which is hereby incorporated by reference.

[00187] In some embodiments, a message passing network is used for the model rather than a GCN as illustrated in Figure 18. In some embodiments, a message passing network is used for the model than a GCN as illustrated in Figure 19. In some embodiments, a message passing network is used for the model than a GCN as illustrated in Figure 20.

[00188] In some embodiments, the model is any encoder that takes a graph of a compound, or other form of representation of a compound, and outputs a vector that represents the compound.

[00189] *Blocks 224-228.* Referring to block 224, in some embodiments, the model is a message passing graph neural network having between 3 and 8 layers. Referring to block 226, in some embodiments, the model is a message passing graph neural network consisting of 5 layers. Referring to block 228, in some embodiments, the model comprises 500,000 or more parameters, or 1×10^6 or more parameters.

[00190] *Block 230.* Referring to block 230, in some embodiments, the model performs at least 100,000 computations to compute the first isometry preserving latent embedding, and the model performs at least 100,000 computations to compute the second isometry preserving latent embedding.

[00191] *Blocks 232-234.* Referring to block 232, in some embodiments, the estimate of the graph edit distance is between the first compound and the second compound is provided

as an L2-norm difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding. Referring to block 234, in some embodiments, the first isometry preserving latent embedding is a first vector comprising a first plurality of vector values, the second isometry preserving latent embedding is a second vector comprising a second plurality of vector values, and the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the first vector and the second vector.

[00192] Advantageously, by using the L2-norm difference of the output of the first and the second output of the model as the GED estimate, all the properties of distance metrics outlined at the top of Figure 6, other than positivity, are satisfied.

[00193] *Block 236.* One practical application / technical effect of GED is to find chemically relevant analogs. For instance, consider the case where the first compound is a known lead compound in a drug discovery effort. In this instance, one or more databases of compounds can be searched for compounds that are chemically relevant analogs by computing the GED distance between the first compound and each compound in the one or more databases of compounds. One example of such a database of compounds is the PubChem database (*see*, Kim *et al.*, 2021, “PubChem in 2021: new data content and improved web interfaces,” *Nucleic Acids Research* 8(49)(D1), pp. D1388-D1395, which is hereby incorporated by reference). Another example of such a database is the ZINC database (*see*, Irwin *et al.*, 2020, “ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery,” *J. Chem. Inf. Model* 60(12), 6065-6073, which is hereby incorporated by reference). Another example of such a database is the ChEMBL database. ChEMBL is a manually curated database of bioactive molecules with drug-like properties. *See*, Gaulton *et al.*, 2019, “ChEMBL: towards direct deposition of bioassay data,” *Nucleic Acids Res.* 47(D1), D930-D940, which is hereby incorporated by reference. Additional examples of such databases include those of commercially available molecules such as MCULE (Kiss *et al.*, 2012, “Http://Mcule.Com: A Public Web Service for Drug Discovery,” *J. Cheminformatics* 4 (1), p.17.) and ENAMINE (Irwin *et al.*, 2016, “Docking Screens for Novel Ligands Conferring New Biology,” *J. Med. Chem.* 59 (9), pp. 4103–4120), each of which is hereby incorporated by reference.

[00194] In some embodiments, the GED distance between the first compound and each of 10 or more compounds in one or more databases of compounds is computed in such an effort.

[00195] In some embodiments, the GED distance between the first compound and each of 100 or more compounds in one or more databases of compounds is computed in such an effort.

[00196] In some embodiments, the GED distance between the first compound and each of 1000 or more compounds in one or more databases of compounds is computed in such an effort.

[00197] In some embodiments, the GED distance between the first compound and each of 10,000 or more compounds in one or more databases of compounds is computed in such an effort.

[00198] In some embodiments, the GED distance between the first compound and each of 100,000 or more compounds in one or more databases of compounds is computed in such an effort.

[00199] In some embodiments, the GED distance between the first compound and each of 1×10^6 , 2×10^6 , 3×10^6 , 4×10^6 , 5×10^6 , 6×10^6 , 7×10^6 , 8×10^6 , or 9×10^6 , or more compounds in one or more databases of compounds are computed in such an effort.

[00200] In some embodiments, the GED distance between the first compound and each of 1×10^7 , 2×10^7 , 3×10^7 , 4×10^7 , 5×10^7 , 6×10^7 , 7×10^7 , 8×10^7 , or 9×10^7 , or more compounds in one or more databases of compounds are computed in such an effort.

[00201] In the drug discovery process, a lead compound is a molecule having biological activity that is likely to be useful, but may require modifications to better fit the target or to have better pharmacological properties. Those compounds that have less than a threshold estimated GED (*e.g.*, less than 10, less than 9, less than 8, less than 7, less than 6, less than 5, less than 4, less than 3, or less than 2) can then be evaluated as chemically relevant analog of the first compound and thus evaluated to see if they better fit the target or have better pharmacological properties. As such, referring to block 236, in some embodiments, a determination is made as to whether the second compound is a chemically relevant analog of the first compound when the estimate of the GED falls below a threshold value. In some embodiments, this threshold value is any real number between 2 and 10 inclusive of 2 and 10. In some embodiments the threshold value is 2, 3, 4, 5, 6, 7, 8, 9 or 10. In some embodiments the threshold value is 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, or 9.5.

[00202] *Block 238.* Another practical application / technical effect of GED is to ensure diversity in a chemical library, or among chemicals selected for a drug discovery program.

For instance, one or more databases of compounds can be clustered using GED as the distance metric to form clusters and then compounds from each of the resulting clusters can be selected for evaluation in the drug discovery program.

[00203] One example of such a database of compounds is the PubChem database (*see*, Kim *et al.*, 2021, “PubChem in 2021: new data content and improved web interfaces,” Nucleic Acids Research 8(49)(D1), pp. D1388-D1395, which is hereby incorporated by reference). Another example of such a database is the ZINC database (*see*, Irwin *et al.*, 2020, “ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery,” J. Chem. Inf. Model 60(12), 6065-6073, which is hereby incorporated by reference). Another example of such a database is the ChEMBL database. ChEMBL is a manually curated database of bioactive molecules with drug-like properties. *See*, Gaulton *et al.*, 2019, “ChEMBL: towards direct deposition of bioassay data,” Nucleic Acids Res. 47(D1), D930-D940, which is hereby incorporated by reference. Additional examples of such databases include those of commercially available molecules such as MCULE (Kiss *et al.*, 2012, “Http://Mcule.Com: A Public Web Service for Drug Discovery,” J. Cheminformatics 4 (1), p.17.) and ENAMINE (Irwin *et al.*, 2016, “Docking Screens for Novel Ligands Conferring New Biology,” J. Med. Chem. 59 (9), pp. 4103–4120), each of which is hereby incorporated by reference.

[00204] Accordingly, referring to block 238, in some embodiments, the optional obtaining of the first graph (block 202), the optional obtaining the second graph (block 204), the inputting the first graph, or other form of representation of the first compound (block 216), the inputting the second graph, or other form of representation of the second compound (block 218), and providing the approximate GED is repeated for each respective unique pair of compounds in a database of compounds, where a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of graph edit distances. The plurality of graph edit distances are then clustered into a plurality of clusters by evaluating the plurality of distances with a criterion function. Then, compounds are selected from one or more clusters in the plurality of clusters for evaluation in a drug discovery program. In some embodiments, compounds are selected from 2, 3, 4, 5, 6, 7, 8, 9, or 10 or more different clusters for the drug discovery program. In some embodiments, 10 or more, 100 or more, 1000 or more, 10,000 or more, 100,000 or more or 1×10^6 or more compounds are selected from each cluster in the one or more clusters.

[00205] In some embodiments, 100 or more unique pairs of compounds are evaluated for block 238. In some embodiments, 1000 or more unique pairs of compounds are evaluated for block 238. In some embodiments, 10,000 or more unique pairs of compounds are evaluated for block 238. In some embodiments, 100,000 or more unique pairs of compounds are evaluated for block 238. In some embodiments, 1×10^6 or more unique pairs of compounds are evaluated for block 238.

[00206] In some embodiments, 1×10^6 , 2×10^6 , 3×10^6 , 4×10^6 , 5×10^6 , 6×10^6 , 7×10^6 , 8×10^6 , or 9×10^6 , or more unique pairs of compounds are evaluated for block 238. In some embodiments, 1×10^7 , 2×10^7 , 3×10^7 , 4×10^7 , 5×10^7 , 6×10^7 , 7×10^7 , 8×10^7 , or 9×10^7 , or more unique pairs of compounds are evaluated for block 238. In some embodiments, 1×10^8 , 2×10^8 , 3×10^8 , 4×10^8 , 5×10^8 , 6×10^8 , 7×10^8 , 8×10^8 , or 9×10^8 , or more unique pairs of compounds are evaluated for block 238.

[00207] In some embodiments, the clustering is unsupervised clustering. In some embodiments, the clustering is supervised clustering. Clustering algorithms suitable for use in accordance with block 348 are described, for example, at pages 211-256 of Duda and Hart, *Pattern Classification and Scene Analysis*, 1973, John Wiley & Sons, Inc., New York, (hereinafter “Duda 1973”) which is hereby incorporated by reference in its entirety. As an illustrative example, in some embodiments, the clustering problem is described as one of finding natural groupings in a dataset. To identify natural groupings, two issues are addressed. First, a way to measure similarity (or dissimilarity) between two samples is determined. This metric (*e.g.*, similarity measure) is used to ensure that the samples in one cluster are more like one another than they are to samples in other clusters. Second, a mechanism for partitioning the data into clusters using the similarity measure is determined. One way to begin a clustering investigation is to define a distance function and to compute the matrix of distances between all pairs of samples in the training set. If distance is a good measure of similarity, then the distance between reference entities in the same cluster is significantly less than the distance between the reference entities in different clusters. However, in some implementations, clustering does not use a distance metric. For example, in some embodiments, a nonmetric similarity function $s(x, x')$ is used to compare two latent embeddings x and x' . In some such embodiments, $s(x, x')$ is a symmetric function whose value is large when x and x' are somehow “similar.” Once a method for measuring “similarity” or “dissimilarity” between latent embeddings has been selected, clustering uses a criterion function that measures the clustering quality of any partition of the data. Partitions

of the plurality of pairs of compounds that extremize the criterion function are used to cluster the data. Particular exemplary clustering techniques contemplated for use in the present disclosure include, but are not limited to, hierarchical clustering (agglomerative clustering using a nearest-neighbor algorithm, farthest-neighbor algorithm, the average linkage algorithm, the centroid algorithm, or the sum-of-squares algorithm), k-means clustering, fuzzy k-means clustering algorithm, and Jarvis-Patrick clustering. In some embodiments, the clustering includes unsupervised clustering (*e.g.*, with no preconceived number of clusters and/or no predetermination of cluster assignments).

[00208] *Block 240.* Another practical application / technical effect of GED is to build quantitative structure activity relationship (QSAR) models. QSAR models are computational or mathematical models that attempt to find a significant correlation between molecular structure and molecular activity. With the huge amount of available data about chemical compounds and their reactivity, there is a need for computational tools to reduce the drug synthesis and test cycle execution times. These tools are important if activity data are to be analyzed and new models created for virtual screening techniques. For QSAR models, the GED distance serves as a basis for quantifying differences in molecule structure. *See*, for example, Garcia-Hernandez, 2020, “Learning the Edit Costs of Graph Edit Distance Applied to Ligand-Based Virtual Screening,” *Current Topics in Medicinal Chemistry* 20, 1582-1592, which is hereby incorporated by reference. Accordingly, referring to block 240, in some embodiments, the estimate of the GED is used to build a quantitative structure activity relationship model that includes the first and second compound.

[00209] In some embodiments the QSAR is built in relation to a first compound that is a known lead compound in a drug discovery effort. In this instance, one or more databases of compounds can be searched for compounds that are within a predetermined GED of the first compound and then the QSAR is built using these compounds.

[00210] One example of such a database of compounds is the PubChem database (*see*, Kim *et al.*, 2021, “PubChem in 2021: new data content and improved web interfaces,” *Nucleic Acids Research* 49(D1), pp. D1388-D1395, which is hereby incorporated by reference). Another example of such a database is the ZINC database (*see*, Irwin *et al.*, 2020, “ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery,” *J. Chem. Inf. Model* 60(12), 6065-6073, which is hereby incorporated by reference). Another example of such a database is the ChEMBL database. ChEMBL is a manually curated database of bioactive molecules with drug-like properties. *See*, Gaulton *et al.*, 2019, “ChEMBL: towards

direct deposition of bioassay data,” Nucleic Acids Res. 47(D1), D930-D940, which is hereby incorporated by reference. Additional examples of such databases include those of commercially available molecules such as MCULE (Kiss *et al.*, 2012, “[Http://Mcule.Com](http://Mcule.Com): A Public Web Service for Drug Discovery,” J. Cheminformatics 4 (1), p.17.) and ENAMINE (Irwin *et al.*, 2016, “Docking Screens for Novel Ligands Conferring New Biology,” J. Med. Chem. 59 (9), pp. 4103–4120), each of which is hereby incorporated by reference.

[00211] As such in some embodiments, the GED distance between the first compound and each of 10, 100, 1,000, 10,000, 100,000, 1×10^6 or 1×10^7 or more compounds in one or more databases of compounds is computed and those compounds that are within a threshold estimated GED of the first compound are used for determining QSAR. In some embodiments the threshold estimated GED is less than 10, less than 9, less than 8, less than 7, less than 6, less than 5, less than 4, less than 3, or less than 2. In some embodiments, the threshold estimated GED is any real number between 2 and 10 inclusive of 2 and 10. In some embodiments the threshold estimated GED is 2, 3, 4, 5, 6, 7, 8, 9 or 10. In some embodiments the threshold estimated GED is 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, or 9.5. In some embodiments, between ten percent and ninety percent of the compounds evaluated for estimated GED with respect to the first compound in the one or more databases is within the threshold estimated GED. In some embodiments, between twenty percent and eighty percent of the compounds evaluated for estimated GED with respect to the first compound in the one or more databases is within the threshold estimated GED. In some embodiments, at least 20, 30, 40, 50, or 60 percent of the compounds evaluated for estimated GED with respect to the first compound in the one or more databases is within the threshold estimated GED. In some embodiments, less than 90, 80, 70, 60, or 50 percent of the compounds evaluated for estimated GED with respect to the first compound in the one or more databases is within the threshold estimated GED.

[00212] *Block 242.* Another practical application / technical effect of GED is to define activity cliffs. An activity cliff arises when a limited number of changes, such as one or two small changes, to a molecule are made and the activity of the derived molecules falls an order of magnitude relative to the original molecule. Knowledge of activity cliffs is an important aspect in structure activity analysis. See, for example, Wassermann, *et al.*, 2010, “Activity Landscape Representations for Structure-Activity Relationship Analysis, J. Med. Chem. 53, 8209-8233, which is hereby incorporated by reference. Accordingly, referring to block 242,

in some embodiments, the estimate of the GED is used to determine an activity cliff with respect to a drug discovery program.

[00213] *Additional embodiments*

[00214] Another aspect of the present disclosure provides a computer system for estimating a graph edit distance between a first compound and a second compound. The computer system comprises one or more processors and memory addressable by the one or more processors. The memory stores at least one program for execution by the one or more processors. The at least one program comprises instructions for (A) inputting a first representation of the first compound into a model comprising a plurality of parameters. The model applies the plurality of parameters to the first representation of the first compound to generate, as first output from the model, a first latent embedding that represents the first compound. The at least one program further comprises instructions for (B) inputting a second representation of the second compound into the model. The model applies the plurality of parameters to the second representation of the second compound to generate, as second output from the model, a second latent embedding that represents the second compound. The at least one program further comprises instructions for (C) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first latent embedding and the second latent embedding.

[00215] In some embodiments the difference is an L2-norm difference between the first latent embedding and the second latent embedding. In some such embodiments, the first latent embedding is a first vector comprising a first plurality of vector values, the second latent embedding is a second vector comprising a second plurality of vector values, and the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the first vector and the second vector. In some such embodiments, the model is trained such that the L2-norm of the difference between the first and second vector is an estimate of the GED between the first and the second molecule.

[00216] In some embodiments, the at least one program further comprises instructions for determining that the second compound is a chemically relevant analog of the first compound when the estimate of the graph edit distance falls below a threshold value.

[00217] In some embodiments, the at least one program further comprises instructions for repeating the inputting (A), inputting (B), and providing (C) for each respective unique pair

of compounds in a database of compounds, where a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of graph edit distances. In some such embodiments, the plurality of graph edit distances is clustered into a plurality of clusters by evaluating the plurality of distances with a criterion function and compounds from one or more clusters in the plurality of clusters are selected for evaluation in a drug discovery program.

[00218] In some embodiments the at least one program further comprises instructions for using the estimate of the graph edit distance to build a quantitative structure activity relationship model that includes the first compound and the second compound.

[00219] In some embodiments, the at least one program further comprises instructions for using the estimate of the graph edit distance to determine an activity cliff with respect to a drug target.

[00220] In some embodiments, the model is a graph convolutional neural network.

[00221] In some embodiments, the model is a graph neural network (message passing network).

[00222] In some embodiments, the model is a message passing graph neural network consisting of between 3 and 8 layers.

[00223] In some embodiments, the model is a message passing graph neural network consisting of 5 layers.

[00224] In some embodiments, the model is an autoencoder. In some such embodiments, the autoencoder is a variational autoencoder, an adversarial autoencoder, a regularized autoencoder, a sparse autoencoder, or a metric learning autoencoder.

[00225] In some embodiments, the first representation is a first fingerprint of the first compound, and the second representation is a second fingerprint of the second compound.

[00226] In some embodiments, the first representation is a first plurality of descriptors of the first compound, and the second representation is a second plurality of descriptors of the second compound.

[00227] In some embodiments, the first representation is a first three-dimensional structural representation of the first compound, and the second representation is a second three-dimensional structural representation of the second compound.

[00228] In some embodiments, the first representation is a first tensor-based representation of the first compound, and the second representation is a second tensor-based representation of the second compound.

[00229] In some embodiments, the first representation is a first string representation of the first compound, and the second representation is a second string representation of the second compound.

[00230] In some embodiments, model comprises 500,000 or more parameters. In some embodiments, the model comprises 1×10^6 or more parameters.

[00231] In some embodiments, the first compound and the second compound each satisfy two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

[00232] In some embodiments, the first compound and the second compound have an exact graph edit distance of between 1 and 10. In some embodiments, the first compound and the second compound have an exact graph edit distance of between 1 and 15.

[00233] In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons. In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than 1000 Daltons. In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than 2000 Daltons. In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons. In some embodiments, the first compound and the second compound are each an organic compound having a molecular weight of between 300 Daltons and 1500 Daltons.

[00234] In some embodiments, the model performs at least 100,000 computations to compute the first latent embedding, and the model performs at least 100,000 computations to compute the second latent embedding.

[00235] In some embodiments the first latent embedding is a first isometry preserving latent embedding. In some embodiments the second latent embedding is a second isometry preserving latent embedding.

[00236] Another aspect of the present disclosure provides a method for estimating a graph edit distance between a first compound and a second compound. The method comprises (*e.g.*, at a computer system comprising one or more processors memory addressable by the one or more processors), (A) inputting a first representation of the first compound into a model comprising a plurality of parameters. The model applies the plurality of parameters to the first representation of the first compound to generate as first output from the model a first latent embedding that represents the first compound. The method further comprises (*e.g.*, at a computer system comprising one or more processors memory addressable by the one or more processors) (B) inputting a second representation of the second compound into the model. The model applies the plurality of parameters to the second representation of the second compound to generate as second output from the model a second latent embedding that represents the second compound. The method further comprises (*e.g.*, at a computer system comprising one or more processors memory addressable by the one or more processors) (C) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first latent embedding and the second latent embedding. In some embodiments the first latent embedding is a first isometry preserving latent embedding. In some embodiments the second latent embedding is a second isometry preserving latent embedding. In some embodiments the difference is an L2-norm difference between the first latent embedding and the second latent embedding.

[00237] Another aspect of the present disclosure provides a non-transitory computer readable storage medium. The non-transitory computer readable storage medium stores instructions, which when executed by a computer system, cause the computer system to perform a method for estimating a graph edit distance between a first compound and a second compound. The method comprises (A) inputting a first representation of the first compound into a model comprising a plurality of parameters. The model applies the plurality of parameters to the first representation of the first compound to generate as first output from the model a first latent embedding that represents the first compound. The method further comprises (B) inputting a second representation of the second compound into the model. The model applies the plurality of parameters to the second representation of the second compound to generate as second output from the model a second latent embedding that represents the second compound. The method further comprises (C) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first latent embedding and the second latent embedding. In some

embodiments the first latent embedding is a first isometry preserving latent embedding. In some embodiments the second latent embedding is a second isometry preserving latent embedding. In some embodiments the difference is an L2-norm difference between the first latent embedding and the second latent embedding.

[00238] *Training a model to predict potency of a test compound against a target protein*

[00239] Successful drug discovery campaigns rely on identifying biologically active lead molecules that are chemically distinct from known compounds for the disease target. This is especially challenging when there is little or no nearby ligand data available, as is the case with novel targets, or when novel scaffolds are distant in chemical space. Structure-based, virtual High-Throughput Screening (vHTS) methods are designed to overcome this challenge, by identifying novel compounds with predicted activity from vast chemical libraries, for example, MCULE (Kiss *et al.*, [Http://McuLe.Com](http://McuLe.Com): A Public Web Service for Drug Discovery, J. Cheminformatics 2012, 4 (1), P17. or ENAMINE (Irwin and Shoichet, 2016, Docking Screens for Novel Ligands Conferring New Biology, J. Med. Chem. 59(9), 4103–4120).

As such, virtual screening is a tool for identifying novel inhibitors of proteins with known structures. *See*, for example, Wallach *et al.*, 2015, “AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery,” doi: 29 10.48550/arxiv.1510.02855; Lyu *et al.*, 2019, “Ultra large library docking for discovering new chemotypes,” Nature 566:7743 566, 20 224–229. doi: 10.1038/s41586-019-0917-9; and Bender *et al.*, 2021, “A practical guide to large-scale docking,” Nature Protocols 16:10 16, 39 4799–4832, each of which is incorporated by reference. Conventional, structure-based virtual high throughput screening approaches use an empirical- or force-field-based scoring function to dock ligands to mostly rigid receptors and rank compounds. *See*, (Trott and Olson, 2010, “AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading,” J Comput Chem 31, 455–461, which is hereby incorporated by reference. Docking to structures that deviate from the ligand-bound state can result in inaccurate predictions of the bound complex and poor compound ranking. For example, it is often difficult to recover active compounds when docking against ligand-free experimental structures (*e.g.*, an apo state), or when the cognate ligand is small. *See*, for example, Abagyan *et al.*, 2010, “Recipes for the selection of experimental protein conformations for virtual screening,” J Chem Inf Model 50, 32 186–193, which is hereby incorporated by reference. Another setback is that experimentally

derived structures are unavailable for many protein targets with disordered or flexible domains. AlphaFold (AF) has the potential to accelerate drug discovery thanks to accurate structure prediction for such proteins. *See*, Jumper *et al.*, 2024, “Highly accurate protein structure prediction with AlphaFold,” *Nature*, doi: 41 10.1038/s41586-021-03819-2, which is hereby incorporated by reference. However, these are still just rigid structures, and their utility will be limited if they do not represent bound-like structures. *See*, for example, Vijayan *et al.*, 2015, “Conformational analysis of the DFG-out kinase motif and biochemical profiling of structurally validated type II inhibitors,” *J Med Chem* 58, 466–479; and Wankowicz *et al.*, 2022, “Ligand binding remodels protein side chain conformational heterogeneity,” *Elife* 11. doi: 10.7554/ELIFE.74114, each of which is hereby incorporated by reference.

[00240] Using vHTS methods that apply deep learning to structure-based drug discovery provides a path for pharmaceutical companies to identify promising new compounds. These methods are likely to have a larger impact in research areas that have been challenging for pharmaceutical companies, such as studies focused on identifying compounds for target proteins with multiple binding sites. However, using these computational methods requires large quantities of well-annotated data regarding compounds and the protein sites they bind. The problem is made more challenging because of the presence of allosteric sites in many protein targets. Allosteric sites are alternate positions on proteins outside of the orthosteric (active) site that can modulate the protein’s function in some way. Compounds and proteins are typically matched using measurements of binding activity. While this is suitable for cases where the majority of compounds being tested bind to a single protein site, there are many protein targets where various compounds can bind to one of multiple possible sites. In such instances, incorrect binding site information used in training predictive models leads to poor model performance. Unfortunately, since most compound activity measurements are not mapped directly to the exact binding sites on the protein, there is not enough data annotated at the site level available in public databases such as the Protein Data Bank (PDB) or ChEMBL to train models directed at alternate sites.

[00241] Given the above, what is needed are improved methods for training models that predict the potency of a target compound. One aspect of the present disclosure addresses this shortcoming by assigning binding sites to training compounds known to bind to a target proteins that have an orthosteric site and one or more allosteric sites by comparing such compounds to reference compounds for which the binding site is known. In this way, more

accurate poses of such training compounds are realized because the training compounds are now placed in the binding site based on the chemical intuition provided by the reference compounds. Using training data of this type to train binding models, it has been discovered that such binding models trained on such multisite data provides improved potency prediction.

[00242] Figure 22 illustrates a computer system 2200 for potency prediction of a test compound against a target protein. Referring to Figure 22, in typical embodiments, computer system 2200 comprises one or more computers. For purposes of illustration in Figure 22, the computer system 2200 is represented as a single computer that includes all of the functionality of the disclosed computer system 2200. However, the present disclosure is not so limited. The functionality of the computer system 2200 may be spread across any number of networked computers and/or reside on each of several networked computers and/or virtual machines. One of skill in the art will appreciate that a wide array of different computer topologies is possible for the computer system 2200 and all such topologies are within the scope of the present disclosure.

[00243] Turning to Figure 22 with the foregoing in mind, the computer system 2200 comprises one or more processing units (CPUs) 2259, a network or other communications interface 2284, a user interface 2278 (*e.g.*, including an optional display 2282 and optional keyboard 2280 or other form of input device), a memory 2292 (*e.g.*, random access memory, persistent memory, or combination thereof), one or more magnetic disk storage and/or persistent devices 2290 optionally accessed by one or more controllers 2288, one or more communication busses 2212 for interconnecting the aforementioned components, and a power supply 2279 for powering the aforementioned components. To the extent that components of memory 2292 are not persistent, data in memory 2292 can be seamlessly shared with non-volatile memory 2290 or portions of memory 2292 that are non-volatile / persistent using known computing techniques such as caching. Memory 2292 and/or memory 2290 can include mass storage that is remotely located with respect to the central processing unit(s) 2259. In other words, some data stored in memory 2292 and/or memory 2290 may in fact be hosted on computers that are external to computer system 2200 but that can be electronically accessed by the computer system 2200 over an Internet, intranet, or other form of network or electronic cable using network interface 2284. In some embodiments, the computer system 2200 makes use of models 2248 that are run from the memory associated with one or more graphical processing units in order to improve the speed

and performance of the system. In some alternative embodiments, the computer system 2200 makes use of models 2248 that are run from memory 2292 rather than memory associated with a graphical processing unit.

[00244] The memory 2292 of the computer system 2200 stores:

- an optional operating system 2230 that includes procedures for handling various basic system services;
- a potency prediction computation module 2232 for training a model 2248 to predict potency of a test compound against a target protein;
- training data 2234, the training data including data for a plurality of training compounds, where each respective training compound 2236 in the plurality of training compounds is labeled with (i) a respective protein target 2240 in a plurality of protein targets, (ii) a binding site for the respective protein target 2242, and (iii) a potency against the protein target 2238;
- the atomic structure of each target protein 2246 in a plurality of proteins; and
- and a binding site model 2248 for predicting potency of a test compound against a target protein.

[00245] In some embodiments, one or more of the above identified data elements or modules of the computer system 2200 are stored in one or more of the previously mentioned memory devices, and correspond to a set of instructions for performing a function described above. The above identified data, modules or programs (*e.g.*, sets of instructions) need not be implemented as separate software programs, procedures or modules, and thus various subsets of these modules may be combined or otherwise re-arranged in various implementations. In some implementations, the memory 2292 and/or 2290 optionally stores a subset of the modules and data structures identified above. Furthermore, in some embodiments the memory 2292 and/or 2290 stores additional modules and data structures not described above.

[00246] Now that a system for training a model 2248 to predict potency of a test compound against a target protein has been disclosed, methods for performing such characterization is detailed with reference to Figure 23 and discussed below.

[00247] *Block 2300.* Referring to block 2300, in some embodiments, systems and methods for training a model to predict potency of a test compound against a target protein are provided. In some embodiments, this model is part of a compound selection workflow.

As a nonlimiting example, in some embodiments, the systems and methods of the present disclosure are used to train model 2402 of the compound selection workflow of Figure 24.

[00248] *Block 2302.* Referring to block 2302, a dataset comprising data for a plurality of training compounds is obtained. Each training compound is labeled with a respective protein target in a plurality of protein targets, a binding site for the respective protein target and a potency against the protein target. In some embodiments the potency is expressed by the model as a calculated pKi. In some embodiments, the potency is expressed by the model as a calculated IC₅₀, EC₅₀, K_d, K_i, or pK_i of the corresponding compound against the target protein. Such measures of potency are referred to as binding coefficients and are generally described in Huser ed., 2006, "High-Throughput-Screening in Drug Discovery," Methods and Principles in Medicinal Chemistry 35; and Chen ed., 2019, *A Practical Guide to Assay Development and High-Throughput Screening in Drug Discovery*, CRC Press, Boca Raton, each of which is hereby incorporated by reference.

[00249] One nonlimiting source of such compounds is ChEMBL. ChEMBL is a manually curated database of bioactive molecules with drug-like properties. *See*, Gaulton *et al.*, 2019, "ChEMBL: towards direct deposition of bioassay data," Nucleic Acids Res. 47(D1), D930-D940, which is hereby incorporated by reference.

[00250] *Block 2304.* Referring to block 2304, in some embodiments, each training compound satisfies two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5. *See*, Lipinski, 1997, Adv. Drug Del. Rev. 23, 3, which is hereby incorporated herein by reference in its entirety. In some embodiments, a training compound satisfies one or more criteria in addition to Lipinski's Rule of Five. For example, in some embodiments, a training compound has five or fewer aromatic rings, four or fewer aromatic rings, three or fewer aromatic rings, or two or fewer aromatic rings.

[00251] *Blocks 2306-2308.* Referring to block 2306, in some embodiments, each training compound is an organic compound having a molecular weight of less than 500 Daltons, less than 1000 Daltons, less than 2000 Daltons, less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons. Referring to block 208, in some embodiments, each training compound is an organic compound having a molecular weight of between 400 Daltons and 10000 Daltons.

[00252] In some embodiments, each training compound has a molecular weight of at least 10 Da, at least 20 Da, at least 50 Da, at least 100 Da, at least 200 Da, at least 500 Da, at least 1 kDa, at least 2 kDa, at least 3 kDa, at least 5 kDa, at least 10 kDa, at least 20 kDa, at least 30 kDa, at least 50 kDa, at least 100 kDa, or at least 500 kDa. In some embodiments, each training compound has a molecular weight of no more than 1000 kDa, no more than 500 kDa, no more than 100 kDa, no more than 50 kDa, no more than 10 kDa, no more than 5 kDa, no more than 2 kDa, no more than 1 kDa, no more than 500 Da, no more than 300 Da, no more than 100 Da, or no more than 50 Da. In some embodiments, each training compound has a molecular weight of from 10 Da to 900 Da, from 50 Da to 1000 Da, from 100 Da to 2000 Da, from 1 kDa to 10 kDa, from 5 kDa to 500 kDa, or from 100 kDa to 1000 kDa. In some embodiments, each training compound has a molecular weight that falls within another range starting no lower than 10 Daltons and ending no higher than 1000 kDa.

[00253] In some embodiments, each training compound is a small molecule. For instance, in some embodiments, each training compound is an organic compound having a molecular weight of less than approximately 1000 Daltons (*e.g.*, less than 900 Daltons).

[00254] *Block 2310.* Referring to block 2310, in some embodiments, the plurality of training compounds comprises 100 or more, 1000 or more, 2000 or more, 10,000 or more, 100,000 or more, or 1×10^6 or more training compounds. In some embodiments, the plurality of training compounds is between 500 training compounds and 1×10^7 training compounds. In some embodiments, the plurality of training compounds is between 1000 training compounds and 1×10^6 training compounds. In some embodiments, the plurality of training compounds is between 10,000 training compounds and 100,000 training compounds.

[00255] *Block 2312.* Referring to block 2312, in some embodiments, each of the protein targets in the plurality of protein targets is represented by one or more training compounds in the plurality of training compounds and the plurality of protein targets comprises 10 or more protein targets or 100 or more protein targets. In other words, in some embodiments, for each respective protein target in the plurality of protein targets, there is at least one training compound that has been docked to the protein target for purposes of generating a pose for training the model.

[00256] In some embodiments, each of the protein targets in the plurality of protein targets is represented by 5 or more compounds in the plurality of compounds and the plurality of protein targets comprises 10 or more protein targets or 100 or more protein targets. In

other words, in some embodiments, for each respective protein target in the plurality of protein targets, there is at least five training compounds that have each been docked to the protein target for purposes of generating five or more poses for training the model.

[00257] In some embodiments, each of the protein targets in the plurality of protein targets is represented by 50 or more compounds in the plurality of compounds and the plurality of protein targets comprises 10 or more protein targets or 100 or more protein targets.

[00258] In some embodiments, each of the protein targets in the plurality of protein targets is represented by 100 or more compounds in the plurality of compounds and the plurality of protein targets comprises 10 or more protein targets or 100 or more protein targets.

[00259] In some embodiments, each of the protein targets in the plurality of protein targets is represented by 500 or more compounds in the plurality of compounds and the plurality of protein targets comprises 10 or more protein targets or 100 or more protein targets.

[00260] In some embodiments, each of the protein targets in the plurality of protein targets is represented by 1000, 10,000 or 100,000 or more compounds in the plurality of compounds and the plurality of protein targets comprises 10 or more protein targets or 100 or more protein targets.

[00261] *Block 2314.* Referring to block 2314, in some embodiments, at least 5 percent, 10 percent, or 15 percent of the plurality of protein targets comprise multiple binding sites. Identification of a binding site for training compounds binding to proteins is important in order to obtain the correct pose information for such training compounds docked to the target protein. For instance, referring to Fig. 29A, if a training compound is docked to the binding site depicted in the left side of the figure, when the training compound actually binds to the binding site depicted in the right side of the figure, the pose data for the training compound that is provided to the model for purposes of training the model will be incorrect. One way to determine the true binding site of such training compounds is to cocluster them with compounds having known binding sites based on compound fingerprints, such as ECFP4, as illustrated in Figure 29B. As depicted in Figures 29B and 30, when each of the compounds with known binding sites in each cluster map to the same binding site, there is confidence

that remaining compounds in the cluster also bind to the same binding site even in the absence of structural data showing this is the case.

[00262] As illustrated in Figure 31A, tools such as AF2 can extend the mapping assignments to protein targets for which experimental structural data of the protein target, such as x-ray crystallography data, has not yet been elucidated. AF2 is described in Jumper *et al.*, 2021, “Highly accurate protein structure prediction with AlphaFold,” *Nature* 596, pp. 583–589; and Tunyasuvunakool *et al.*, 2021, “Highly accurate protein structure prediction for the human proteome,” *Nature* 596, 590–596, each of which is hereby incorporated by reference. With reference to Figure 31A, a protein target bound to a test or training compound is referred to as ligand-bound or HOLO conformation, and a protein target without a bound test or training compound is referred to as a ligand-free or APO conformation. As indicated in Figure 31A, in one approach to expanding the number of target proteins that can be used for training the model beyond those whose structures have been experimentally determined, through means such as x-ray crystallography or nuclear magnetic resonance, and indexed in the Protein Data Bank (PDB) database accessed through BLASTp, AF2 structures are generated from the UniProt protein identifiers. In some embodiments, those generated structures having less than 30% sequence identity to the top BLASTp hit from a BLASTp search of the Protein Databank or have no HOLO homolog or structure are discarded. Blastp is described in Altschul, *et al.*, 1997, “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25(17), pp. 3389-402; and Zhang and Freddolino, 2023, “A large-scale assessment of sequence database search tools for homology-based protein function prediction,” <https://doi.org/10.1101/2023.11.14.567021>, each of which is hereby incorporated by reference. UniProt is described in UniProt: the Universal Protein Knowledgebase in 2023, *Nucleic Acids Res.* 51:D523–D531 (2023), which is hereby incorporated by reference. The PDB database is described in Berman *et al.*, 2000, “The Protein Data Bank (2000)” *Nucleic Acids Research* 28: 235-242, which is hereby incorporated by reference. Referring to Figure 31A, in some embodiments, a HOLO template for a target protein generated by AF2 is considered average when it has between 30 percent and 70 percent sequence identity to the top BLASTp hit from a query of the PDB database with the UniProt sequence of the target protein. In some embodiments, a HOLO template for a target protein generated by AF2 is considered good when it has between 70 and 95 percent sequence identity to the top BLASTp hit from a query of the PDB database with the UniProt sequence of the target protein. In some embodiments, a HOLO template for

a target protein generated by AF2 is considered resolved when it has greater than 95 percent sequence identity to the top BLASTp hit from a query of the PDB database with the UniProt sequence of the target protein.

[00263] *Block 2315.* Referring to block 2315, a model is trained using the training dataset. For each respective training compound in the plurality of training compounds, one or more poses of the respective training compound in the binding site of the target protein identified in the dataset for the training compound are each inputted into the model and evaluated against the actual known potency of the respective training compound.

[00264] In some embodiments, this is done with the command line parameters --exhaustiveness 384 --energy_range 99999 --num_modes 64 --mc_steps 3 --minimize_iters 40 --accurate_line --approximation linear --autobox_add 2.0 --seed 42 as described in Stafford *et al.*, 2022, “AtomNet PoseRanker: Enriching Ligand Pose Quality for Dynamic Proteins in Virtual High-Throughput Screens,” J. Chem. Inf. Model 62, p. 1178-1189, which is hereby incorporated by reference. In such approaches, a bounding box is drawn around the designated binding site in the target protein and the training compound is docked to the portion of the target protein contained within this bound box. Figure 8A illustrates the bounding box of two different binding sites of a target protein. In some embodiments, the conformation of the training compound is obtained using a program such as RDKit. *See*, Landrum, G. RDKit: Open-Source Cheminformatics. In some such embodiments, a SMILES description of the training compound is UFF energy minimized using RDKit to obtain the conformations of the training compound that are docked into the binding site. *See also* Meller *et al.*, 2023, “Discovery of a cryptic pocket in the AI-predicted structure of PPM1D phosphatase explains the binding site and potency of its allosteric inhibitors,” bioRxiv [Preprint], 2023 Mar 24:2023.03.22.533829. doi: 10.1101/2023.03.22.533829, for information on how a pose of a compound in a binding site of a protein is constructed in some embodiments.

[00265] In some embodiments, for each pose of a respective training compound in a binding site, a plurality of descriptors is generated for input into the model. In some embodiments, each heavy atom (non-hydrogen atom) is described with its corresponding SYBYL atom type indicating its chemical environment, generated, for example, by converting to Mol2 format using OpenBabel. *See* O’Boyle *et al.*, 2011, “Open Babel: An Open Chemical Toolbox. J. Cheminf. 3, 33, which is hereby incorporated by reference. In some embodiments, protons are not represented explicitly in the descriptor sets, and all

metals are a single type. In some embodiments, the target protein binding site and the training compound are kept as distinct entities and their atom types are encoded separately.

[00266] Errors in the predicted potency of the respective training compounds are back-propagated through the parameters of the model of the systems of the present disclosure (*e.g.*, model 2248 of Figure 22, model 2402 of Figure 24). In some embodiments, the model is trained against the errors in the estimated potency made by the model, in view of the actual potency for the training compound. One way to perform such training is to use stochastic gradient descent with the AdaDelta adaptive learning method (Zeiler, 2012 “ADADELTA: an adaptive learning rate method,” CoRR, vol. abs/1212.5701, which is hereby incorporated by reference), and the back propagation algorithm provided in Rumelhart *et al.*, 1988, “Neurocomputing: Foundations of research,” ch. Learning Representations by Back-propagating Errors, pp. 696-699, Cambridge, MA, USA: MIT Press, which is hereby incorporated by reference. However, any back-propagation method can be used for training.

[00267] *Block 2316.* Referring to block 2316, a test chemical compound is obtained.

[00268] *Block 2318.* Referring to block 218, in some embodiments, the test compound satisfies two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5. *See*, Lipinski, 1997, Adv. Drug Del. Rev. 23, 3, which is hereby incorporated herein by reference in its entirety. In some embodiments, the test compound satisfies one or more criteria in addition to Lipinski's Rule of Five. For example, in some embodiments, the test compound has five or fewer aromatic rings, four or fewer aromatic rings, three or fewer aromatic rings, or two or fewer aromatic rings.

[00269] *Blocks 2320-2322.* Referring to block 2320, in some embodiments, the test compound is an organic compound having a molecular weight of less than 500 Daltons, less than 1000 Daltons, less than 2000 Daltons, less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons. Referring to block 2322, in some embodiments, the test compound is an organic compound having a molecular weight of between 400 Daltons and 10000 Daltons.

[00270] *Block 2324.* Referring to block 2324, the test chemical compound is posed against a binding site of the target protein to obtain a test pose. Poses have been described above in conjunction with training compounds. Additional examples of poses are described

in United States Patent Nos. 11,080,570, 10,546,237; 10,482,355, 10,002,312, 9,379,059; and United States Provisional Patent Application No. 63/336,841, entitled “Characterization of Interactions Between Compounds and Polymers Using Pose Ensembles,” filed April 29, 2022, each of which is hereby incorporated by reference.

[00271] *Block 2325.* Referring to block 2325, the test pose is inputted into the model to predict potency of the test compound against the binding site. As illustrated in Figures 32, 33A, and 33B, advantageously, a model trained on the multi-site data of the present disclosure shows improved performance at predicting the potency of compounds against the target protein. In particular, Figure 32 shows that the area under curve (AUC) for the receiver operating characteristic curve (ROC) for the model trained with the binding site specific data of the present disclosure versus a static assumption that each training compound binds to the orthosteric site of the target protein is improved. This improvement is even observed for the primary binding site (orthosteric site, active site, binding site 1 in Figure 32). Figure 33A shows improvement in the mean square error of the model upon training with a plurality of protein targets that have been augmented with the AF2 generated protein targets described herein with reference to Figure 31A (af2_holo on the Y-axis of Fig. 33A) versus training with a plurality of protein targets that have not been augmented with the AF2 generated protein targets (holo on the Y-axis of Fig. 33A). Figure 33B shows improvement in the TauB upon training with a plurality of protein targets that have been augmented with the AF2 generated protein targets described herein with reference to Figure 31A (af2_holo on the Y-axis of Fig. 33B) versus training with a plurality of protein targets that has not been augmented with the AF2 generated protein targets (holo on the Y-axis of Fig. 33B).

[00272] *Block 2326.* Referring to block 2326, in some embodiments, the model comprises a first plurality of parameters. In some embodiments, the model applies the plurality of parameters to the pose through at least 10,000 computations to generate the predicted potency. In some embodiments, the model applies the plurality of parameters to the pose through at least 100,000 computations or at least 1×10^6 computations to generate the predicted potency.

[00273] *Block 2328.* Referring to block 2328, in some embodiments, the model is a graph neural network. In some embodiments the model is a SE(3)-equivariant graph attention transformer. In some embodiments, the model is a message passing neural network. Message passing neural networks are described, for example, in Gilmer *et al.*, 2020, “Message Passing Neural Networks,” In: Schütt *et al.* (eds), Machine Learning Meets

Quantum Physics, Lecture Notes in Physics 968, Springer, Cham, each of which is hereby incorporated by reference.

[00274] In some embodiments, the model is a graph convolutional neural network. Nonlimiting examples of graph convolutional neural networks are disclosed in Behler Parrinello, 2007, “Generalized Neural-Network Representation of High Dimensional Potential-Energy Surfaces,” *Physical Review Letters* 98, 146401; Chmiela *et al.*, 2017, “Machine learning of accurate energy-conserving molecular force fields,” *Science Advances* 3(5):e1603015; Schütt *et al.*, 2017, “SchNet: A continuous-filter convolutional neural network for modeling quantum interactions,” *Advances in Neural Information Processing Systems* 30, pp. 992-1002; Feinberg *et al.*, 2018, “PotentialNet for Molecular Property Prediction,” *ACS Cent. Sci.* 4, 11, 1520–1530; and Stafford *et al.*, 2022, “AtomNet PoseRanker: Enriching Ligand Pose Quality for Dynamic Proteins in Virtual High Throughput Screens,” *J. Chem Inf Model* 62, pp. 1178–1189, each of which is hereby incorporated by reference.

[00275] In some embodiments, a final readout layer of the model extracts relevant features from training compound and protein target embeddings independently and passes them through a final multilayer perceptron.

[00276] In some embodiments, the model is, or is similar to, the GRAPHite architecture described in Stafford *et al.*, 2022, “AtomNet PoseRanker: Enriching Ligand Pose Quality for Dynamic Proteins in Virtual High Throughput Screens,” *J. Chem Inf Model* 62, pp. 1178–1189). The GRAPHite architecture is a directed Graph Convolutional Network (GCN) comprised of four graph convolutional layers. In some embodiments, the first two layers include both ligand and receptor features, whereas the last two layers are ligand-only. In some embodiments, nodes in the graph represent ligand and receptor atoms. In some embodiments, only receptor atoms within 7 Å of any ligand atom are used as part of the graph. In some embodiments, edges were defined by atoms within 4 Å of each other and edge weights are distance-dependent. In some embodiments, the final layer is sum-pooled into an embedding. In some embodiments, this embedding is then passed through two (independent) multilayer perceptrons to predict two outputs: the ANPR pose quality score, and the Vina docking score. In some embodiments, those outputs are then concatenated to the embedding and passed through a third multilayer perceptron that outputs the predicted potency in the form of a pKi.

[00277] *Block 2330.* Referring to block 2330, in some embodiments, the model comprises 500,000 or more parameters, or 1×10^6 or more parameters. In some embodiments, the model comprises more than 400 parameters, more than 1000 parameters, more than 2000 parameters, more than 5000 parameters, more than 10,000 parameters, more than 100,000 parameters, or more than 1×10^6 parameters. In some embodiments, the amount of information in the pose inputted into the model coupled with the number of the parameters of the model results in the performance of more than 10,000 computations, more than 100,000 computations, more than 1×10^6 computations, more than 5×10^6 computations, or more than 1×10^7 computations to calculate the predicted potency.

[00278] *Examples*

[00279] *Example 1: Building Training and Test datasets.* With reference to Figure 3B, to identify methods that provided GED approximations, training and test datasets were created. One desirable property for the train/test split was to create a train/test split of drug-like molecules/fragments with similar GED distributions. Another desirable property for the train/test split was to require that train/test pairs be sufficiently dissimilar to ensure that performance generalizes to unseen chemical spaces (rather than model overlearning). To create a training and test dataset with these properties, compounds were randomly taken from ChEMBL 30. ChEMBL is a manually curated database of bioactive molecules with drug-like properties. See, Gaulton *et al.*, 2019, "ChEMBL: towards direct deposition of bioassay data," Nucleic Acids Res. 47(D1), D930-D940, which is hereby incorporated by reference. The randomly selected compounds were then filtered for drug-likeness. One method for computing a drug-likeness score is a quantitative estimate of drug-likeness (QED), Bickerton *et al.*, 2012, "Quantifying the chemical beauty of drugs," Nat. Chem. 4, pp. 90–98, which is hereby incorporated by reference. Compounds were screened to have a QED drug-likeness score of 0.5 or greater. Further, the compounds were constrained to have a graph edit distance of less than 10. Further still, the compounds were stratified on GED so that all GED values between 0 and 10 were uniformly sampled. Finally, the compounds were split into a train/test set using Taylor-Butina clustering using the extended connectivity fingerprint, up to four bonds (ECFP4) of each compound. As shown in Figure 3B, all pairs in a given cluster is either placed in the training set or the test to ensure that there is no pair of compounds in the test set that is sufficiently similar to a pair of compounds in the training set such that a model could simply memorize pairs in the training set and extend this memorization to the pairs in

the test set, rather than learning true approximations for GED distance, even for chemical pairs not seen by the model.

[00280] The above describes construction of the drug-like training/test dataset. Referring to Figure 4, the training drug-like dataset had 45,000 compounds while the test dataset had 4500 compounds. GED was computed for the compounds using commercial software based on chemical GED. Figure 21 illustrates the difference between chemical and exact graph edit distance. As seen in Figure 21, the chemical graph edit distance considers chemical changes whereas the exact GED considers the number of elementary changes described above. The fragment dataset was created by randomly taking drug-like molecules and cutting them up into fragments. The fragments were then filtered using the same criteria used to filter for the chemical dataset (QED greater than 0.5, GED less than or equal to 10, stratified by GED), with the exception that exact GED was computed for the fragments using Depth-first GED rather than chemical GED. Depth-first GED is described in Abu-Aisheh *et al.*, 2015, 4th International Conference on Pattern Recognition Applications and Methods, Lisbon Portugal, which is hereby incorporated by reference. Figure 4 further illustrates the distribution of compounds as a function of number of heavy atoms in the training and test datasets. Each respective pair of compounds in the training and test dataset in the fragment dataset summarized in Figure 4 has the training label of exact GED between the respective pair compounds. Each respective pair of compounds in the training and test dataset in the drug-like dataset summarized in Figure 4 has the training label of chemical GED between the respective pair of compounds.

[00281] *Example 2: Testing SimGnn on the test and training datasets.* Referring to Figure 5, the performance of a model in accordance with the prior art (SimGNN, Bai *et al.* 2019, ACM International Conference on Web Search and Data Mining) after training on the fragment dataset and the drug-like dataset of Figure 4 respectively is illustrated in terms of MSE, Spearman's ρ and Kendall's τ_B . Here, Spearman's ρ and Kendall's τ_B is the ability to rank pairs of compounds in the test dataset based on estimated GED such that the ranking matches the rankings of the test dataset based on the true GED. Unfortunately, the performance statistics of Figure 5 show that even when training on the highly curated datasets of Figure 4, the performance of SimGNN is not satisfactory. Figure 6 addresses the basis for the shortcomings of SimGNN. As noted in Figure 6, GED is a distance metric and therefore any model that approximates GED should seek to satisfy the properties of distance metrics. In detail, the distance to self should be zero. So, a model that estimates the distance between

two identical molecules should return a GED of zero. Further, there should be symmetry, so that the GED from the graph of molecule A to molecule B should equal the GED from the graph of molecule B to molecule A. Further, the estimated GED should be a positive value greater than zero when the two graphs (two compounds) being compared are not identical. Finally, the model should obey the triangle inequality theorem. Evaluating the compounds of the drug-like test set on the SimGNN model after training on the drug-like training set of Figure 4, it is seen that SimGNN has symmetry violations of 0.67 and distance to self-violation 1.52. In other words, the GED approximation for molecule A to molecule B deviates from molecule B to molecule A by 0.67 across the drug-like test set of Figure 4 while the distance to self across the drug-like test set, which should be zero, is in fact 1.52.

[00282] *Example 3: Building an improved approximate GED computation module 32.*

To address the shortcomings identified in Figure 6, the system summarized in Figure 7 was built. Figure 7 illustrates how a first graph G_i 36 representing the first compound is obtained. The first graph comprises a first plurality of nodes and a first plurality of edges. Each atom of the first compound is represented by a node 38 in the first plurality of nodes and each bond of the first compound is represented by an edge 40 in the first plurality of edges. Further a second graph G_j 46 representing a second compound 44 is obtained. The second graph comprises a second plurality of nodes and a second plurality of edges. Each atom of the second compound is represented by a node 48 in the second plurality of nodes and each bond of the second compound is represented by an edge 50 in the second plurality of edges. The first graph is inputted into model 702 comprising a plurality of parameters. Model 702 applies the plurality of parameters to the first graph to generate, as first output from the model 702, a first latent embedding h_i that represents the first compound. The second graph is inputted into model 704 (which is, in fact model 702, but is shown separately for purposes of illustration only). Model 704 applies the plurality of parameters to the second graph to generate, as second output from the model 704, a second latent embedding h_j that represents the second compound. Then, the estimate of the graph edit distance between the first compound and the second compound is determined as the L2-norm difference between the first latent embedding and the second latent embedding. In the case where the first latent embedding is a first vector and the second latent embedding is a second vector, the L2-norm difference (between the first vector and the second vector) is taken as the square root of the sum of the squared difference in corresponding vector element values between corresponding

elements in the first vector and the second vector. Figure 18 illustrates the mathematical specification of Figure 7.

[00283] Model 702 / 704 is a message passing neural network in this example. Message passing neural networks are described, for example, in Gilmer *et al.*, 2020, “Message Passing Neural Networks,” In: Schütt *et al.* (eds), Machine Learning Meets Quantum Physics, Lecture Notes in Physics 968, Springer, Cham, each of which is hereby incorporated by reference.

[00284] *Example 4: Training a system in accordance with the present disclosure.* By the training model 702 / 704 such that the L2-norm difference of the first and second output of the model is equal to the training GED estimates, all the properties of distance metrics outlined at the top of Figure 6, other than positivity, are satisfied. In some embodiments, the positivity axiom breaks down in instances where model 702 / 704 cannot distinguish the two graphs.

[00285] The system 700 illustrated in Figure 7 outputs an estimated graph edit distance (e.g., the L2-norm difference of the output of model 702 / 704). The estimated GEDs assigned by the systems of the present disclosure for each respective pair of training compounds in the fragment dataset or the drug-like dataset summarized in Figure 4, were then compared to the actual GEDs for the pairs of compounds. In the case of the fragmented dataset, the actual GEDs were the exact GEDs as summarized in Figure 4. In the case of the drug-like dataset, the actual GEDs were taken to be the chemical GEDs as summarized in Figure 4. The difference in the estimated GED produced by model 700 and the actual GEDs for the training set were considered errors made by the systems of the present disclosure (e.g., model 700). To further train the model 700, these errors were back-propagated through the parameters of the model 702 / 704. In some embodiments, system 700 was trained against the errors in the estimated GED made by system 700, in view of the actual GED for the training set. One way to perform such training is to use stochastic gradient descent with the AdaDelta adaptive learning method (Zeiler, 2012 “ADADELTA: an adaptive learning rate method,” CoRR, vol. abs/1212.5701, which is hereby incorporated by reference), and the back propagation algorithm provided in Rumelhart *et al.*, 1988, “Neurocomputing: Foundations of research,” ch. Learning Representations by Back-propagating Errors, pp. 696-699, Cambridge, MA, USA: MIT Press, which is hereby incorporated by reference. However, any back-propagation method can be used for training.

[00286] Figure 8 shows the performance of system 700 of Figure 7, which is termed MolGEDNet. It is seen from Figure 8 that MolGEDNet significantly outperformed SimGNN on mean squared error on both the fragment dataset and the drug-like dataset of Figure 4. Figure 9 shows the performance of system 700 of Figure 7, which is termed MolGEDNet. It is seen from Figure 9 that MolGEDNet significantly outperforms SimGNN on Spearman's ρ and Kendall's τ_B on both the fragment dataset and the drug-like dataset of Figure 4.

[00287] *Example 5: Hyperparameter tuning of the system in accordance with the present disclosure.* With the superior metrics of MolGEDNet over SimGNN established as illustrated in Figures 8 and 9 and discussed in Example 4, work was done to tune the hyperparameters of MolGEDNet as illustrated in Figure 10. For Figure 10, as in Figures 8 and 9, model 702 / 704 was a message passing graph neural network. As illustrated in Figure 10, it was found that dimensioning model 702 / 704 with respect to latent embedding size, in terms of edge embedding size and node embedding size, was optimal when a size of 128 is used. To obtain the results illustrated in Figure 10, the system 700 was trained on the drug-like training dataset of Figure 4 and mean squared error, Spearman's ρ and Kendall's τ_B was then calculated across the drug-like test dataset of Figure 4. As illustrated in Figure 10 it was found that dimensioning model 702 / 704 with respect to graph embedding size, in terms of edge embedding size and node embedding size, did not significantly affect performance, in terms of Spearman's ρ and Kendall's τ_B for all the dimensions tested, from 64 through 256.

[00288] Figure 11 illustrates the model performance of MolGEDNet when different dropout percentages were considered. Here, dropout references to the percentage of output in each layer (connections between layer i and layer $i+1$) that were dropped in model training. Thus, for a dropout of 0.1, ten percent of the outputs of each layer were randomly selected and dropped during training rather than passing these output on to subsequent layers within models 702 and 704. As Figure 11 illustrates, the addition of dropout did not improve model performance (here measured by mean squared error). To obtain the results illustrated in Figure 11, the system 700 was trained on the drug-like training dataset of Figure 4 with the different dropout values illustrated in Figure 11 and mean squared error was then calculated across the drug-like test dataset of Figure 4 with no dropout (regardless of the amount of dropout that was used in training).

[00289] Figure 12 illustrates the model performance of MolGEDNet when model 702 / 704 had varying numbers of layers. For Figure 12, model 702 / 704 was a message passing graph neural network and what was being varied was the number of layers in the model 702 /

704 (4, 5, 6, or 7 layers). As illustrated in Figure 12 it was found that model 702 / 704 with five layers was optimal for the test datasets of Figure 4. To obtain the results illustrated in Figure 12, the system 700 was trained on the drug-like training dataset of Figure 4 and mean squared error, Spearman's ρ and Kendall's τ_B was then calculated across the drug-like test dataset of Figure 4. As illustrated in Figure 12 it was found that dimensioning model 702 / 704 with respect to number of layers did not significantly affect performance, in terms of Spearman's ρ and Kendall's τ_B for all configurations tested, from 4 layers through 7 layers.

[00290] Referring to Figure 13, system 700 was trained on the drug-like training dataset of Figure 4 using the optimal hyperparameters 128 variable dimensions, 5 layers, and no dropout for messaging passing graph neural network 702 / 704. Figure 19 illustrates how model 702 / 704 performs node, edge, and graph updates with these optimal hyperparameters. Figure 20 illustrates the details of the message passing neural network used to generate the performance data of Figure 13. The trained system was then used to compute mean squared error, Spearman's ρ and Kendall's τ_B across the drug-like test dataset of Figure 4. As seen in Figure 13, with these optimal hyperparameters, system 700 had a mean squared error of 1.27, Spearman's ρ of 0.94, and Kendall's τ_B of 0.81.

[00291] *Example 6: Examining sources of error in the hyperparameter optimized system of the present disclosure.* Figure 14 examines the source of the error for the hyperparameter optimized system 700 of Figure 13 / Example 5. Figure 14 shows that the disclosed model 702 / 704 tends to overestimate GED when true graph edit distance was less than five or six, and tended to underestimate GED when true graph edit distance was greater six.

[00292] To determine where in the compounds error is arising from, pairs of compounds in the test set that had a true GED of exactly 1 were examined. Examples of such pairs are shown in Figure 15 along with the estimated GED given by the disclosed model. The examples in the left-hand box in Figure 15 are pairs that had large errors whereas the examples in the right hand box in Figure 15 are pairs that have small errors. From this, it is seen that large errors occur when edits are in the center of the compound whereas high accuracy is achieved when edits are in the periphery of the compound. In Figure 16, the betweenness centrality of the nodes requiring change in each of the pairs of Figure 15 is given. Betweenness centrality is a measure of centrality in a graph based on shortest paths. For every pair of vertices in a connected graph, there exists at least one shortest path between the vertices such that either the number of edges that the path passes through (for unweighted graphs) or the sum of the weights of the edges (for weighted graphs) is minimized. The

betweenness centrality for each vertex is the number of these shortest paths that pass through the vertex. *See*, Freeman, 1977, "A set of measures of centrality based on betweenness," *Sociometry*, 40 (1): 35–41, which is hereby incorporated by reference. Nodes that have a high betweenness centrality are near the center of a graph whereas nodes that are on the periphery of the graph has a low betweenness centrality. What is seen is that higher error in GED estimation arises for the disclosed model 702 / 704 when the betweenness centrality of the node that must be substituted in a given pair of compounds is high.

[00293] *Example 7: Run times of the hyperparameter optimized system of the present disclosure.* Referring to Figure 17, the average run time as a function of the average molecular weight of the two compounds being compared was given for the disclosed model 702 / 704 (MolGEDNet, with the optimal hyperparameters set forth in Example 5), exact methods, and the Hungarian algorithm. It is seen that MolGEDNet performs very well in terms of speed, almost fifty times faster than the Hungarian approximation algorithm, while at the same time much more accurate than the Hungarian approximation algorithm.

CONCLUSION

[00294] The foregoing description, for purposes of explanation, has been described with reference to specific implementations. However, the illustrative discussions above are not intended to be exhaustive or to limit the implementations to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The implementations were chosen and described in order to best explain the principles and their practical applications, to thereby enable others skilled in the art to best utilize the implementations and various implementations with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A computer system for estimating a graph edit distance between a first compound and a second compound, the computer system comprising:

one or more processors; and

memory addressable by the one or more processors, the memory storing at least one program for execution by the one or more processors, the at least one program comprising instructions for:

(A) obtaining a first graph representing the first compound, the first graph comprising a first plurality of nodes and a first plurality of edges, wherein each atom of the first compound is represented by a node in the first plurality of nodes and each bond of the first compound is represented by an edge in the first plurality of edges;

(B) obtaining a second graph representing the second compound, the second graph comprising a second plurality of nodes and a second plurality of edges, wherein each atom of the second compound is represented by a node in the second plurality of nodes and each bond of the second compound is represented by an edge in the second plurality of edges;

(C) inputting the first graph into a model comprising a plurality of parameters, wherein the model applies the plurality of parameters to the first graph through at least 10,000 computations to generate as first output from the model a first isometry preserving latent embedding that represents the first compound;

(D) inputting the second graph into the model, wherein the model applies the plurality of parameters to the second graph through at least 10,000 computations to generate as second output from the model a second isometry preserving latent embedding that represents the second compound; and

(E) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

2. The computer system of claim 1, wherein the difference is an L2-norm difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

3. The computer system of claim 2, wherein

the first isometry preserving latent embedding is a first vector comprising a first plurality of vector values;

the second isometry preserving latent embedding is a second vector comprising a second plurality of vector values; and

the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the first vector and the second vector, wherein the model is trained such that the L2-norm of the difference between the first and second vector is an estimate of the GED between the first and the second molecule.

4. The computer system of any one of claims 1-3, wherein the at least one program further comprises instructions for determining that the second compound is a chemically relevant analog of the first compound when the estimate of the graph edit distance falls below a threshold value.

5. The computer system of any one of claims 1-4, wherein the at least one program further comprises instructions for:

repeating the obtaining (A), obtaining (B), inputting (C), inputting (D), and providing (E) for each respective unique pair of compounds in a database of compounds, wherein a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of graph edit distances;

clustering the plurality of graph edit distances into a plurality of clusters by evaluating the plurality of distances with a criterion function; and

selecting compounds from one or more clusters in the plurality of clusters for evaluation in a drug discovery program.

6. The computer system of any one of claims 1-5, wherein the at least one program further comprises instructions for using the estimate of the graph edit distance to build a quantitative structure activity relationship model that includes the first compound and the second compound.

7. The computer system of any one of claims 1-6, wherein the at least one program further comprises instructions for using the estimate of the graph edit distance to determine an activity cliff with respect to a drug target.
8. The computer system of any one of claims 1-7, wherein the model is a graph convolutional neural network.
9. The computer system of any one of claims 1-7, wherein the model is a graph neural network (message passing network).
10. The computer system of any one of claims 1-7, wherein the model is a message passing graph neural network consisting of between 3 and 8 layers.
11. The computer system of any one of claims 1-7, wherein the model is a message passing graph neural network consisting of 5 layers.
12. The computer system of any one of claims 1-7, wherein the model is an autoencoder.
13. The computer system of claim 12, wherein the autoencoder is a variational autoencoder, an adversarial autoencoder, a regularized autoencoder, a sparse autoencoder, or a metric learning autoencoder.
14. The computer system of any one of claims 1-13, wherein the model comprises 500,000 or more parameters.
15. The computer system of any one of claims 1-13, wherein the model comprises 1×10^6 or more parameters.
16. The computer system of any one of claims 1-15, wherein the first compound and the second compound each satisfy two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

17. The computer system of any one of claims 1-16, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 10.
18. The computer system of any one of claims 1-16, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 15.
19. The computer system of any one of claims 1-18, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons.
20. The computer system of any one of claims 1-18, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 1000 Daltons.
21. The computer system of any one of claims 1-18, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 2000 Daltons.
22. The computer system of any one of claims 1-18, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons.
23. The computer system of any one of claims 1-18, wherein first compound and the second compound are each an organic compound having a molecular weight of between 300 Daltons and 1500 Daltons.
24. The computer system of any one of claims 1-23, wherein
the model performs at least 100,000 computations to compute the first isometry preserving latent embedding, and
the model performs at least 100,000 computations to compute the second isometry preserving latent embedding.

25. A method for estimating a graph edit distance between a first compound and a second compound, the method comprising:

at a computer system comprising one or more processors memory addressable by the one or more processors:

(A) obtaining a first graph representing the first compound, the first graph comprising a first plurality of nodes and a first plurality of edges, wherein each atom of the first compound is represented by a node in the first plurality of nodes and each bond of the first compound is represented by an edge in the first plurality of edges;

(B) obtaining a second graph representing the second compound, the second graph comprising a second plurality of nodes and a second plurality of edges, wherein each atom of the second compound is represented by a node in the second plurality of nodes and each bond of the second compound is represented by an edge in the second plurality of edges;

(C) inputting the first graph into a model comprising a plurality of parameters, wherein the model applies the plurality of parameters to the first graph through at least 10,000 computations to generate as first output from the model a first isometry preserving latent embedding that represents the first compound;

(D) inputting the second graph into the model, wherein the model applies the plurality of parameters to the second graph through at least 10,000 computations to generate as second output from the model a second isometry preserving latent embedding that represents the second compound; and

(E) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

26. The method of claim 25, wherein the difference is an L2-norm difference between the first latent embedding and the second latent embedding.

27. The method of claim 26, wherein

the first isometry preserving latent embedding is a first vector comprising a first plurality of vector values;

the second isometry preserving latent embedding is a second vector comprising a second plurality of vector values; and

the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the

first vector and the second vector, wherein the model is trained such that the L2-norm of the difference between the first and second vector is an estimate of the GED between the first and the second molecule.

28. The method of any one of claims 25-27, wherein the at least one program further comprises instructions for determining that the second compound is a chemically relevant analog of the first compound when the estimate of the graph edit distance falls below a threshold value.

29. The method of any one of claims 25-28, wherein the method further comprises:

repeating the obtaining (A), obtaining (B), inputting (C), inputting (D), and providing (E) for each respective unique pair of compounds in a database of compounds, wherein a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of graph edit distances;

clustering the plurality of graph edit distances into a plurality of clusters by evaluating the plurality of distances with a criterion function; and

selecting compounds from one or more clusters in the plurality of clusters for evaluation in a drug discovery program.

30. The method of any one of claims 25-29, the method further comprising using the estimate of the graph edit distance to build a quantitative structure activity relationship model for that includes the first compound and the second compound.

31. The method of any one of claims 25-30, wherein the method further comprises using the estimate of the graph edit distance to determine an activity cliff with respect to a drug target.

32. The method of any one of claims 25-31, wherein the model is a graph convolutional neural network.

33. The method of any one of claims 25-31, wherein the model is a graph neural network (message passing network).

34. The method of any one of claims 25-31, wherein the model is a message passing graph neural network consisting of between 3 and 8 layers.
35. The method of any one of claims 25-31, wherein the model is a message passing graph neural network consisting of 5 layers.
36. The method of any one of claims 25-31, wherein the model is an autoencoder.
37. The method of claim 36, wherein the autoencoder is a variational autoencoder, an adversarial autoencoder, a regularized autoencoder, a sparse autoencoder, or a metric learning autoencoder.
38. The method of any one of claims 25-38, wherein the model comprises 500,000 or more parameters.
39. The method of any one of claims 25-38, wherein the model comprises 1×10^6 or more parameters.
40. The method of any one of claims 25-39, wherein the first compound and the second compound each satisfy two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.
41. The method of any one of claims 25-40, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 10.
42. The method of any one of claims 25-40, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 15.
43. The method of any one of claims 25-42, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons.

44. The method of any one of claims 25-42, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 1000 Daltons.
45. The method of any one of claims 25-42, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 2000 Daltons.
46. The method of any one of claims 25-42, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons.
47. The method of any one of claims 25-42, wherein first compound and the second compound are each an organic compound having a molecular weight of between 300 Daltons and 1500 Daltons.
48. The method of any one of claims 25-47, wherein
the model performs at least 100,000 computations to compute the first isometry preserving latent embedding, and
the model performs at least 100,000 computations to compute the second isometry preserving latent embedding.
49. A non-transitory computer readable storage medium, wherein the non-transitory computer readable storage medium stores instructions, which when executed by a computer system, cause the computer system to perform a method for estimating a graph edit distance between a first compound and a second compound, the method comprising:
(A) obtaining a first graph representing the first compound, the first graph comprising a first plurality of nodes and a first plurality of edges, wherein each atom of the first compound is represented by a node in the first plurality of nodes and each bond of the first compound is represented by an edge in the first plurality of edges;
(B) obtaining a second graph representing the second compound, the second graph comprising a second plurality of nodes and a second plurality of edges, wherein each atom of

the second compound is represented by a node in the second plurality of nodes and each bond of the second compound is represented by an edge in the second plurality of edges;

(C) inputting the first graph into a model comprising a plurality of parameters, wherein the model applies the plurality of parameters to the first graph through at least 10,000 computations to generate as first output from the model a first isometry preserving latent embedding that represents the first compound;

(D) inputting the second graph into the model, wherein the model applies the plurality of parameters to the second graph through at least 10,000 computations to generate as second output from the model a second isometry preserving latent embedding that represents the second compound; and

(E) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

50. The non-transitory computer readable storage medium of claim 45, wherein the difference is an L2-norm difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

51. The non-transitory computer readable storage medium of claim 50, wherein

the first isometry preserving latent embedding is a first vector comprising a first plurality of vector values;

the second isometry preserving latent embedding is a second vector comprising a second plurality of vector values; and

the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the first vector and the second vector, wherein the model is trained such that the L2-norm of the difference between the first and second vector is an estimate of the GED between the first and the second molecule.

52. The non-transitory computer readable storage medium of any one of claims 49-51, wherein the at least one program further comprises instructions for determining that the second compound is a chemically relevant analog of the first compound when the estimate of the graph edit distance falls below a threshold value.

53. The non-transitory computer readable storage medium of any one of claims 49-52, wherein the method further comprises:

repeating the obtaining (A), obtaining (B), inputting (C), inputting (D), and providing (E) for each respective unique pair of compounds in a database of compounds, wherein a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of graph edit distances;

clustering the plurality of graph edit distances into a plurality of clusters by evaluating the plurality of distances with a criterion function; and

selecting compounds from one or more clusters in the plurality of clusters for evaluation in a drug discovery program.

54. The non-transitory computer readable storage medium of any one of claims 49-53, the method further comprising using the estimate of the graph edit distance to build a quantitative structure activity relationship model for that includes the first compound and the second compound.

55. The non-transitory computer readable storage medium of any one of claims 49-54, wherein the method further comprises using the estimate of the graph edit distance to determine an activity cliff with respect to a drug target.

56. The non-transitory computer readable storage medium of any one of claims 49-55, wherein the model is a graph convolutional neural network.

57. The non-transitory computer readable storage medium of any one of claims 49-55, wherein the model is a graph neural network (message passing network).

58. The non-transitory computer readable storage medium of any one of claims 49-55, wherein the model is a message passing graph neural network consisting of between 3 and 8 layers.

59. The non-transitory computer readable storage medium of any one of claims 49-55, wherein the model is a message passing graph neural network consisting of 5 layers.

60. The non-transitory computer readable storage medium of any one of claims 59-55, wherein the model is an autoencoder.
61. The non-transitory computer readable storage medium of claim 60, wherein the autoencoder is a variational autoencoder, an adversarial autoencoder, a regularized autoencoder, a sparse autoencoder, or a metric learning autoencoder.
62. The non-transitory computer readable storage medium of any one of claims 49-61, wherein the model comprises 500,000 or more parameters.
63. The non-transitory computer readable storage medium of any one of claims 49-61, wherein the model comprises 1×10^6 or more parameters.
64. The non-transitory computer readable storage medium of any one of claims 49-63, wherein the first compound and the second compound each satisfy two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.
65. The non-transitory computer readable storage medium of any one of claims 49-64, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 10.
66. The non-transitory computer readable storage medium of any one of claims 49-64, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 15.
67. The non-transitory computer readable storage medium of any one of claims 49-66, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons.
68. The non-transitory computer readable storage medium of any one of claims 49-66, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 1000 Daltons.

69. The non-transitory computer readable storage medium of any one of claims 49-66, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 2000 Daltons.

70. The non-transitory computer readable storage medium of any one of claims 49-66, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons.

71. The non-transitory computer readable storage medium of any one of claims 49-66, wherein first compound and the second compound are each an organic compound having a molecular weight of between 300 Daltons and 1500 Daltons.

72. The non-transitory computer readable storage medium of any one of claims 49-71, wherein

the model performs at least 100,000 computations to compute the first isometry preserving latent embedding, and

the model performs at least 100,000 computations to compute the second isometry preserving latent embedding.

73. A computer system for estimating a graph edit distance between a first compound and a second compound, the computer system comprising:

one or more processors; and

memory addressable by the one or more processors, the memory storing at least one program for execution by the one or more processors, the at least one program comprising instructions for:

(A) inputting a first representation of the first compound into a model comprising a plurality of parameters, wherein the model applies the plurality of parameters to the first representation of the first compound to generate, as first output from the model, a first isometry preserving latent embedding that represents the first compound;

(B) inputting a second representation of the second compound into the model, wherein the model applies the plurality of parameters to the second representation of the

second compound to generate, as second output from the model, a second isometry preserving latent embedding that represents the second compound; and

(C) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

74. The computer system of claim 73, wherein the difference is an L2-norm difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

75. The computer system of claim 73, wherein

the first isometry preserving latent embedding is a first vector comprising a first plurality of vector values;

the second latent isometry preserving embedding is a second vector comprising a second plurality of vector values; and

the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the first vector and the second vector, wherein the model is trained such that the L2-norm of the difference between the first and second vector is an estimate of the GED between the first and the second molecule.

76. The computer system of any one of claims 73-75, wherein the at least one program further comprises instructions for determining that the second compound is a chemically relevant analog of the first compound when the estimate of the graph edit distance falls below a threshold value.

77. The computer system of any one of claims 73-76, wherein the at least one program further comprises instructions for:

repeating the inputting (A), inputting (B), and providing (C) for each respective unique pair of compounds in a database of compounds, wherein a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of graph edit distances;

clustering the plurality of graph edit distances into a plurality of clusters by evaluating the plurality of distances with a criterion function; and

selecting compounds from one or more clusters in the plurality of clusters for evaluation in a drug discovery program.

78. The computer system of any one of claims 73-77, wherein the at least one program further comprises instructions for using the estimate of the graph edit distance to build a quantitative structure activity relationship model that includes the first compound and the second compound.

79. The computer system of any one of claims 73-78, wherein the at least one program further comprises instructions for using the estimate of the graph edit distance to determine an activity cliff with respect to a drug target.

80. The computer system of any one of claims 73-79, wherein the model is a graph convolutional neural network.

81. The computer system of any one of claims 73-79, wherein the model is a graph neural network (message passing network).

82. The computer system of any one of claims 73-79, wherein the model is a message passing graph neural network consisting of between 3 and 8 layers.

83. The computer system of any one of claims 73-79, wherein the model is a message passing graph neural network consisting of 5 layers.

84. The computer system of any one of claims 73-79, wherein the model is an autoencoder.

85. The computer system of claim 84, wherein the autoencoder is a variational autoencoder, an adversarial autoencoder, a regularized autoencoder, a sparse autoencoder, or a metric learning autoencoder.

86. The computer system of any one of claims 73-85, wherein
the first representation is a first fingerprint of the first compound, and

the second representation is a second fingerprint of the second compound.

87. The computer system of any one of claims 73-85, wherein
the first representation is a first plurality of descriptors of the first compound, and
the second representation is a second plurality of descriptors of the second compound.
88. The computer system of any one of claims 73-85, wherein
the first representation is a first three-dimensional structural representation of the first compound, and
the second representation is a second three-dimensional structural representation of the second compound.
89. The computer system of any one of claims 73-85, wherein
the first representation is a first tensor-based representation of the first compound, and
the second representation is a second tensor-based representation of the second compound.
90. The computer system of any one of claims 73-85, wherein
the first representation is a first string representation of the first compound, and
the second representation is a second string representation of the second compound.
91. The computer system of any one of claims 73-90, wherein the model comprises 500,000 or more parameters.
92. The computer system of any one of claims 73-90, wherein the model comprises 1×10^6 or more parameters.
93. The computer system of any one of claims 73-92, wherein the first compound and the second compound each satisfy two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

94. The computer system of any one of claims 73-93, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 10.
95. The computer system of any one of claims 73-93, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 15.
96. The computer system of any one of claims 73-95, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons.
97. The computer system of any one of claims 73-96, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 1000 Daltons.
98. The computer system of any one of claims 73-96, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 2000 Daltons.
99. The computer system of any one of claims 73-96, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons.
100. The computer system of any one of claims 73-96, wherein first compound and the second compound are each an organic compound having a molecular weight of between 300 Daltons and 1500 Daltons.
101. The computer system of any one of claims 73-100, wherein
the model performs at least 100,000 computations to compute the first isometry preserving latent embedding, and
the model performs at least 100,000 computations to compute the second isometry preserving latent embedding.

102. A method for estimating a graph edit distance between a first compound and a second compound, the method comprising:

at a computer system comprising one or more processors memory addressable by the one or more processors:

(A) inputting a first representation of the first compound into a model comprising a plurality of parameters, wherein the model applies the plurality of parameters to the first representation of the first compound to generate as first output from the model a first isometry preserving latent embedding that represents the first compound;

(B) inputting a second representation of the second compound into the model, wherein the model applies the plurality of parameters to the second representation of the second compound to generate as second output from the model a second isometry preserving latent embedding that represents the second compound; and

(C) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

103. The method of claim 102, wherein the difference is an L2-norm difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

104. The method of claim 103, wherein

the first isometry preserving latent embedding is a first vector comprising a first plurality of vector values;

the second isometry preserving latent embedding is a second vector comprising a second plurality of vector values; and

the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the first vector and the second vector, wherein the model is trained such that the L2-norm of the difference between the first and second vector is an estimate of the GED between the first and the second molecule.

105. The method of any one of claims 102-104, wherein the method further comprises determining that the second compound is a chemically relevant analog of the first compound when the estimate of the graph edit distance falls below a threshold value.

106. The method of any one of claims 102-105, wherein the method further comprises:
- repeating the inputting (A), inputting (B), and providing (C) for each respective unique pair of compounds in a database of compounds, wherein a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of graph edit distances;
 - clustering the plurality of graph edit distances into a plurality of clusters by evaluating the plurality of distances with a criterion function; and
 - selecting compounds from one or more clusters in the plurality of clusters for evaluation in a drug discovery program.
107. The method of any one of claims 102-106, wherein the method further comprises using the estimate of the graph edit distance to build a quantitative structure activity relationship model that includes the first compound and the second compound.
108. The method of any one of claims 102-107, wherein the method further comprises using the estimate of the graph edit distance to determine an activity cliff with respect to a drug target.
109. The method of any one of claims 102-108, wherein the model is a graph convolutional neural network.
110. The method of any one of claims 102-108, wherein the model is a graph neural network (message passing network).
111. The method of any one of claims 102-108, wherein the model is a message passing graph neural network consisting of between 3 and 8 layers.
112. The method of any one of claims 102-111, wherein the model is a message passing graph neural network consisting of 5 layers.
113. The method of any one of claims 102-111, wherein the model is an autoencoder.

114. The method of claim 113, wherein the autoencoder is a variational autoencoder, an adversarial autoencoder, a regularized autoencoder, a sparse autoencoder, or a metric learning autoencoder.
115. The method of any one of claims 102-114, wherein
the first representation is a first fingerprint of the first compound, and
the second representation is a second fingerprint of the second compound.
116. The method of any one of claims 102-114, wherein
the first representation is a first plurality of descriptors of the first compound, and
the second representation is a second plurality of descriptors of the second compound.
117. The method of any one of claims 102-114, wherein
the first representation is a first three-dimensional structural representation of the first compound, and
the second representation is a second three-dimensional structural representation of the second compound.
118. The method of any one of claims 102-114, wherein
the first representation is a first tensor-based representation of the first compound, and
the second representation is a second tensor-based representation of the second compound.
119. The method of any one of claims 102-114, wherein
the first representation is a first string representation of the first compound, and
the second representation is a second string representation of the second compound.
120. The method of any one of claims 102-119, wherein the model comprises 500,000 or more parameters.
121. The method of any one of claims 102-119, wherein the model comprises 1×10^6 or more parameters.

122. The method of any one of claims 102-121, wherein the first compound and the second compound each satisfy two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

123. The method of any one of claims 102-122, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 10.

124. The method of any one of claims 102-123, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 15.

125. The method of any one of claims 102-124, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons.

126. The method of any one of claims 102-124, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 1000 Daltons.

127. The method of any one of claims 102-124, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 2000 Daltons.

128. The method of any one of claims 102-124, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons.

129. The method of any one of claims 102-124, wherein first compound and the second compound are each an organic compound having a molecular weight of between 300 Daltons and 1500 Daltons.

130. The method of any one of claims 102-129, wherein

the model performs at least 100,000 computations to compute the first latent embedding, and

the model performs at least 100,000 computations to compute the second latent embedding.

131. A non-transitory computer readable storage medium, wherein the non-transitory computer readable storage medium stores instructions, which when executed by a computer system, cause the computer system to perform a method for estimating a graph edit distance between a first compound and a second compound, the method comprising:

(A) inputting a first representation of the first compound into a model comprising a plurality of parameters, wherein the model applies the plurality of parameters to the first representation of the first compound to generate as first output from the model a first isometry preserving latent embedding that represents the first compound;

(B) inputting a second representation of the second compound into the model, wherein the model applies the plurality of parameters to the second representation of the second compound to generate as second output from the model a second isometry preserving latent embedding that represents the second compound; and

(C) providing the estimate of the graph edit distance between the first compound and the second compound using a difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

132. The non-transitory computer readable storage medium of claim 131, wherein the difference is an L2-norm difference between the first isometry preserving latent embedding and the second isometry preserving latent embedding.

133. The non-transitory computer readable storage medium of claim 132, wherein

the first isometry preserving latent embedding is a first vector comprising a first plurality of vector values;

the second isometry preserving latent embedding is a second vector comprising a second plurality of vector values; and

the L2-norm difference between the first vector and the second vector is taken as the square root of the sum of the squared difference in corresponding vector values between the first vector and the second vector, wherein the model is trained such that the L2-norm of the

difference between the first and second vector is an estimate of the GED between the first and the second molecule.

134. The non-transitory computer readable storage medium of any one of claims 131-133, wherein the method further comprises determining that the second compound is a chemically relevant analog of the first compound when the estimate of the graph edit distance falls below a threshold value.

135. The non-transitory computer readable storage medium of any one of claims 131-134, wherein the method further comprises:

repeating the inputting (A), inputting (B), and providing (C) for each respective unique pair of compounds in a database of compounds, wherein a first compound in the respective unique pair of compounds is the first compound and a second compound in the respective unique pair of compounds is the second compound, thereby forming a plurality of graph edit distances;

clustering the plurality of graph edit distances into a plurality of clusters by evaluating the plurality of distances with a criterion function; and

selecting compounds from one or more clusters in the plurality of clusters for evaluation in a drug discovery program.

136. The non-transitory computer readable storage medium of any one of claims 131-135, wherein the method further comprises using the estimate of the graph edit distance to build a quantitative structure activity relationship model that includes the first compound and the second compound.

137. The non-transitory computer readable storage medium of any one of claims 131-136, wherein the method further comprises using the estimate of the graph edit distance to determine an activity cliff with respect to a drug target.

138. The non-transitory computer readable storage medium of any one of claims 131-137, wherein the model is a graph convolutional neural network.

139. The non-transitory computer readable storage medium of any one of claims 131-137, wherein the model is a graph neural network (message passing network).

140. The non-transitory computer readable storage medium of any one of claims 131-137, wherein the model is a message passing graph neural network consisting of between 3 and 8 layers.

141. The non-transitory computer readable storage medium of any one of claims 131-137, wherein the model is a message passing graph neural network consisting of 5 layers.

142. The non-transitory computer readable storage medium of any one of claims 131-137, wherein the model is an autoencoder.

143. The non-transitory computer readable storage medium of claim 142, wherein the autoencoder is a variational autoencoder, an adversarial autoencoder, a regularized autoencoder, a sparse autoencoder, or a metric learning autoencoder.

144. The non-transitory computer readable storage medium of any one of claims 131-143, wherein

the first representation is a first fingerprint of the first compound, and
the second representation is a second fingerprint of the second compound.

145. The non-transitory computer readable storage medium of any one of claims 131-143, wherein

the first representation is a first plurality of descriptors of the first compound, and
the second representation is a second plurality of descriptors of the second compound.

146. The non-transitory computer readable storage medium of any one of claims 131-143, wherein

the first representation is a first three-dimensional structural representation of the first compound, and
the second representation is a second three-dimensional structural representation of the second compound.

147. The non-transitory computer readable storage medium of any one of claims 131-143, wherein

the first representation is a first tensor-based representation of the first compound, and the second representation is a second tensor-based representation of the second compound.

148. The non-transitory computer readable storage medium of any one of claims 131-143, wherein

the first representation is a first string representation of the first compound, and the second representation is a second string representation of the second compound.

149. The non-transitory computer readable storage medium of any one of claims 131-148, wherein the model comprises 500,000 or more parameters.

150. The non-transitory computer readable storage medium of any one of claims 131-148, wherein the model comprises 1×10^6 or more parameters.

151. The non-transitory computer readable storage medium of any one of claims 131-150, wherein the first compound and the second compound each satisfy two or more rules, three or more rules, or all four rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

152. The non-transitory computer readable storage medium of any one of claims 131-151, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 10.

153. The non-transitory computer readable storage medium of any one of claims 131-151, wherein the first compound and the second compound have an exact graph edit distance of between 1 and 15.

154. The non-transitory computer readable storage medium of any one of claims 131-153, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 500 Daltons.

155. The non-transitory computer readable storage medium of any one of claims 131-153, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 1000 Daltons.

156. The non-transitory computer readable storage medium of any one of claims 131-153, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 2000 Daltons.

157. The non-transitory computer readable storage medium of any one of claims 131-153, wherein the first compound and the second compound are each an organic compound having a molecular weight of less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons.

158. The non-transitory computer readable storage medium of any one of claims 131-153, wherein first compound and the second compound are each an organic compound having a molecular weight of between 300 Daltons and 1500 Daltons.

159. The non-transitory computer readable storage medium of any one of claims 131-153, wherein

the model performs at least 100,000 computations to compute the first latent embedding, and

the model performs at least 100,000 computations to compute the second latent embedding.

160. A computer system for training a model to predict a potency of a test compound against a target protein, the computer system comprising:

one or more processors; and

memory addressable by the one or more processors, the memory storing at least one program for execution by the one or more processors, the at least one program comprising instructions for:

obtaining a dataset comprising data for a plurality of training compounds, wherein each respective training compound in the plurality of training compounds is labeled with (i) a respective protein target in a plurality of protein targets, (ii) a binding site for the respective protein target, and (iii) a potency against the protein target, and wherein at least 5 percent of

the plurality of protein targets comprise multiple binding sites and wherein at least 5 percent of the plurality of training compounds specify an allosteric site of a corresponding protein target in the plurality of protein targets; and

training a model using the dataset, wherein for each respective training compound in the plurality of training compounds, a pose of respective training compound in the binding site of the target protein identified in the dataset for the training compound is inputted into the model and evaluated against the potency of the respective training compound.

161. The computer system of claim 160, wherein the at least one program further comprises instructions for:

obtaining a test chemical compound;
posing the test chemical compound against a binding site of the target protein to obtain a test pose; and
inputting the test pose into the model thereby obtaining the prediction of the potency of the test compound against the target protein.

162. The computer system of claim 161, wherein the target protein is in the plurality of protein targets.

163. The computer system of claim 161, wherein the target protein is other than any protein in the plurality of protein targets.

164. The computer system of any one of claims 160-163, wherein each training compound in the plurality of training compounds satisfies two or more rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

165. The computer system of any one of claims 160-163, wherein each training compound in the plurality of training compounds satisfies three or more rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

166. The computer system of any one of claims 160-163, wherein each training compound in the plurality of training compounds has (i) not more than five hydrogen bond donors, (ii) not

more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

167. The computer system of any one of claims 160-166, wherein each training compound in the plurality of training compounds is an organic compound having a molecular weight of less than 500 Daltons.

168. The computer system of any one of claims 160-166, wherein each training compound in the plurality of training compounds is an organic compound having a molecular weight of less than 1000 Daltons.

169. The computer system of any one of claims 160-166, wherein each training compound in the plurality of training compounds is an organic compound having a molecular weight of less than 2000 Daltons.

170. The computer system of any one of claims 160-166, wherein each training compound in the plurality of training compounds is an organic compound having a molecular weight of less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10000 Daltons, or less than 20000 Daltons.

171. The computer system of any one of claims 160-166, wherein each training compound in the plurality of training compounds is an organic compound having a molecular weight of between 350 Daltons and 10000 Daltons.

172. The computer system of any one of claims 160-171, wherein the plurality of training compounds comprises 1000 or more training compounds.

173. The computer system of any one of claims 160-171, wherein the plurality of training compounds comprises 2000 or more training compounds.

174. The computer system of any one of claims 160-171, wherein the plurality of training compounds comprises 10,000 or more training compounds.

175. The computer system of any one of claims 160-174, wherein each of the protein targets in the plurality of protein targets is represented by one or more compounds in the plurality of compounds and the plurality of protein targets comprises 10 or more protein targets.

176. The computer system of any one of claims 160-174, wherein each of the protein targets in the plurality of protein targets is represented by one or more compounds in the plurality of compounds and the plurality of protein targets comprises 100 or more protein targets.

177. The computer system of any one of claims 160-174, wherein each of the protein targets in the plurality of protein targets is represented by 10 or more compounds in the plurality of compounds and the plurality of protein targets comprises 10 or more protein targets.

178. The computer system of any one of claims 160-177, wherein at least 10 percent of the plurality of protein targets comprise multiple binding sites.

179. The computer system of any one of claims 160-177, wherein at least 15 percent of the plurality of protein targets comprise multiple binding sites.

180. The computer system of claim 161, wherein the test compound satisfies two or more rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

181. The computer system of claim 161, wherein the test compound satisfies three or more rules of the Lipinski's rule of Five: (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

182. The computer system of claim 161, wherein the test compound has (i) not more than five hydrogen bond donors, (ii) not more than ten hydrogen bond acceptors, (iii) a molecular weight under 500 Daltons, and (iv) a LogP under 5.

183. The computer system of claim 161, wherein the test compound is an organic compound having a molecular weight of less than 500 Daltons.

184. The computer system of claim 161, wherein the test compound is an organic compound having a molecular weight of less than 1000 Daltons.

185. The computer system of claim 161, wherein the test compound is an organic compound having a molecular weight of less than 2000 Daltons, less than 4000 Daltons, less than 6000 Daltons, less than 8000 Daltons, less than 10,000 Daltons, or less than 20,000 Daltons.

186. The computer system of claim 161, wherein the test compound is an organic compound having a molecular weight of between 350 Daltons and 10,000 Daltons.

187. The computer system of claim 161, wherein
the model comprises a plurality of parameters, and
the model applies the plurality of parameters to the test pose through at least 10,000 computations to generate the predicted potency of the test compound.

188. The computer system of any one of claims 160-187, wherein the model is a graph neural network.

189. The computer system of any one of claims 160-188, wherein the model comprises 500,000 or more parameters.

190. The computer system of any one of claims 160-189, wherein the model comprises 1×10^6 or more parameters.

191. The computer system of any one of claims 160-190, wherein the potency against the protein target is a pKi.

192. The computer system of any one of claims 160-90, wherein the potency against the protein target is an IC50, EC50, Kd, or KI.

193. The computer system of any one of claims 160-192, wherein the potency of the test compound against the target protein is a pKi.

194. The computer system of any one of claims 160-192, wherein the potency of the test compound against the target protein is an IC₅₀, EC₅₀, K_d, or K_I.

195. A method for training a model to predict a potency of a test compound against a target protein, the method comprising:

at a computer system comprising one or more processors memory addressable by the one or more processors:

obtaining a dataset comprising data for a plurality of training compounds, wherein each respective training compound in the plurality of training compounds is labeled with (i) a respective protein target in a plurality of protein targets, (ii) a binding site for the respective protein target, and (iii) a potency against the protein target, and wherein at least 5 percent of the plurality of protein targets comprise multiple binding sites and wherein at least 5 percent of the plurality of training compounds specify an allosteric site of a corresponding protein target in the plurality of protein targets; and

training a model using the dataset, wherein for each respective training compound in the plurality of training compounds, a pose of respective training compound in the binding site of the target protein identified in the dataset for the training compound is inputted into the model and evaluated against the potency of the respective training compound.

196. The method of claim 195, the method further comprising:

obtaining a test chemical compound;

posing the test chemical compound against a binding site of the target protein to obtain a test pose; and

inputting the test pose into the model thereby obtaining the prediction of the potency of the test compound against the target protein.

197. A non-transitory computer readable storage medium, wherein the non-transitory computer readable storage medium stores instructions, which when executed by a computer system, cause the computer system to perform a method for training a model to predict potency of a test compound against a target protein, the method comprising:

obtaining a dataset comprising data for a plurality of training compounds, wherein each respective training compound in the plurality of training compounds is labeled with (i) a respective protein target in a plurality of protein targets, (ii) a binding site for the respective protein target, and (iii) a potency against the protein target, and wherein at least 5 percent of

the plurality of protein targets comprise multiple binding sites and wherein at least 5 percent of the plurality of training compounds specify an allosteric site of a corresponding protein target in the plurality of protein targets; and

training a model using the dataset, wherein for each respective training compound in the plurality of training compounds, a pose of respective training compound in the binding site of the target protein identified in the dataset for the training compound is inputted into the model and evaluated against the potency of the respective training compound.

198. The non-transitory computer readable storage medium of claim 197, the method further comprising:

obtaining a test chemical compound;

posing the test chemical compound against a binding site of the target protein to obtain a test pose; and

inputting the test pose into the model thereby obtaining the prediction of the potency of the test compound against the target protein.