Generative Models for Chemical Structures

Dr. David White[†] and Dr. Richard C. Wilson[*,‡]

Software Technologies Research Group, University of Bamberg, Germany, and Department of Computer Science, University of York, UK

E-mail: wilson@cs.york.ac.uk

Abstract

We apply recently developed techniques for pattern recognition to construct a generative model for chemical structure. This approach can be viewed as ligand based *de novo* design. We construct a statistical model describing the structural variations present in a set of molecules which may be sampled to generate new structurally-similar examples. We prevent the possibility of generating chemically invalid molecules, according to our implicit hydrogen model, by projecting samples onto the nearest chemically valid molecule. By populating the input set with molecules that are active against a target, we show how new molecules may be generated that will likely also be active against the target.

1 Introduction

In this article we detail a method of generating new chemical structures that offers a way to explore the *chemical space*[1,2] near a set of structurally similar molecules. This type of exploration of chemical space often occurs in drug discovery but is difficult to perform effectively due to a

[*]To whom correspondence should be addressed
[†]Software Technologies Research Group, University of Bamberg, Germany
[‡]Department of Computer Science, University of York, UK

combinatorial explosion in the search space. Our approach produces new chemical structures according to a statistical model of structural variation which is learnt from the structures present in an *input set* of molecules. Therefore, molecules generated using this model will be drawn from the distribution of molecular structures present in the input set. This is in contrast to virtual screening methods where the molecules are selected on the basis of a number of desirable properties.

Nevertheless, when the desired properties are structural in nature, as is the case in drug discovery, our method can produce novel molecules which exhibit the same structural traits. Given that the effectiveness of a drug is largely dependent on its structure,[3,4] we expect that if the input set is populated with molecules that are effective against a target, then the molecules we generate will have a high probability of also being effective against the target. Thus, the approach could provide a method for lead identification.

The approach draws on ideas from *de novo* design techniques[5] which are automated methods for producing chemical structures with certain properties. In reality, no one method is exhaustive and a number of lead identification techniques are often used in tandem such as high throughput screening (of molecules[6] and fragments[7]) and database searching.[8] In terms of drug discovery, the properties required of a molecule are threefold. Firstly, the shape of the intended active site determines the overall structure of the molecule. Secondly, it must be possible to synthesize the molecule and thirdly, the molecule must be druglike.[9]

De novo methods for ligand design have been in development for more than 15 years and have their roots in computer assisted chemical elucidation systems[10] such as CHEMICS[11] or CONCOA.[12] Initially, methods only implemented the first requirement of the three outlined above, that is, imposing structural constraints on the generated ligands.

Early approaches built the ligands inside the active site starting from *seed* atoms. By joining either atoms (GenStar[13]) or fragments (GroupBuild[14]) to the seeds, a complete ligand could be produced that was complementary to the shape of the active site. LigBuilder[15] used a similar approach but the process was controlled by a genetic algorithm. GROW[16] used an incremental approach and avoided some of the complications of fragment assembly by only joining amino

acids. BUILDER[17] searched a database of structures which were then superimposed in the active site. Ligands were found by tracing paths through the superpositions.

While all the above approaches construct the ligands with 3D constraints in place, 2D systems have also been proposed that make use of conversion algorithms to produce low energy 3D conformations from 2D descriptions during the process. In DBMAKER[18] and BOOMSLANG[19] generated molecules are described using SMILES strings and then converted into 3D structures using CONCORD. MOLMAKER[20] uses a graph theoretic approach to enumerate all graphs with certain sets of vertex degrees. These graphs are then attributed with atom and edge labels and thus transformed into molecules.

More recently, attention has been paid to the requirement that *de novo* methods should produce ligands that have a realistic synthesis routes. For example, DREAM++[21] produces new ligands through user specified chemical reactions which are then checked for compatibility with the active site. TOPAS[22] uses a large fragment library derived from current drugs and a restricted set of reactions to produce new structures. A template structure is presented to the system from which a number of new molecules are produced. An evolutionary algorithm is employed in which the best molecule from the current generation becomes the new template structure for the next generation. SYNOPSIS[23] also employs a cyclic generation procedure where new molecules are added to a pool that is used to construct later molecules. SPROUT[24] offers a three step approach to the problem; first a skeleton is constructed that represents the structure of the active site. Next, fragments with generic atom types are substituted into the skeleton and finally, real atoms are assigned to the generic atoms.

The third requirement of druglikeness is generally only assessed in *de novo* methods through a simple filtering step based on a rule of thumb (e.g. Lipinski's "Rule of Five"[25]). However, a recent approach by Kutchukian et al.[26] allows druglike molecules to be assembled according to the class of molecules the system is trained on, such as drugs or natural products.

In this work we focus on the first requirement of generating drug candidates, that is, structural requirements. By representing molecules as relational graphs we propose a method of constructing

a generative model for chemical structure. Since we will use only a set of active molecules as input, rather than information on the target receptor, our approach can be viewed as ligand based *de novo* structure generation.

In the pattern recognition domain we are often faced with the task of classification, that is, determining which class a particular sample belongs to. We view the problem of chemical structure generation as the reverse of classification; given a particular class of molecules, generate samples that belong to that class. We will therefore construct a statistical model of the data and sample from it to generate new examples. This process is well-understood when the data naturally resides in a vectorial form. However, we make use of relational graphs to represent 2D chemical structures[27] and it is difficult to define the necessary statistical quantities such as the mean and variance of a set of graphs.[28,29]

There have been several proposed solutions[30–34] to this problem for generic graphs. One approach[32] is to construct a canonical representation of each graph using a graph alignment algorithm[35] and then embed[36–39] each ordered graph in a vector space. A model is then constructed over the embeddings of the input graphs and thus generated samples appear in vectorial form. A reconstruction step[40] is necessary to recover a graph from a generated vector. For some types of graphs this reconstruction step is the inverse mapping of the initial vectorization. However, in other cases a more complex reconstruction step can be required to recover an almost-correct sample into a valid representation. For example, if the model was trained with discrete graphs but allows vectors representing weighted graphs to be sampled, then a reconstruction step such as projection onto the nearest discrete graph is required. We term such generated samples that are invalid with respect to the trained data *quasi-graphs* (or *quasi-molecules* if the data represents chemical structures).

The situation is similar when chemical structures are considered. The model might be trained on valid chemical structures, but the possibility exists that quasi-molecules could be generated. This is seen in both computer assisted structure elucidation and computer assisted molecule design algorithms, whether the generated structure does not adhere to valence laws, has overlapping atoms when a 3D representation is considered or perhaps is simply invalid in terms of not having a

realistic synthesis route using current techniques.

In our approach we ensure that generated chemical structures are valid with respect to an implicit hydrogen model. We allow the possibility of sampling chemically invalid structures but correct these by projection onto the nearest valid structure chosen from a set of molecules (termed the *projection set*) which are constructed using a simple statistical fragment linking technique. More complex fragment assembly techniques are available[26,41] but are out of the scope of our research. The subset of projection molecules that are mapped to from generated quasi-molecules are termed the *generated set*. To allow construction of the model using standard methods[42] we employ a vectorial representation. This means that all molecules, quasi or otherwise, reside in the same vector space and this brings the added benefit of not requiring complex measures of graph similarity[43] to perform the projection step.

The key idea is that although the set of projection molecules do not represent a sample from the input distribution, they are similar enough such that if we select a subset intelligently then those graphs in the subset are acceptable as samples. On the other hand, the set of quasi-molecules we create are *drawn directly* from the distribution of input molecules. We therefore find the subset of projection molecules that are suitable as samples by projecting a true sample from the input distribution (a generated quasi-molecule) onto the most similar "correct" molecule from the projection set. In this way we can approximate sampling the input distribution without the possibility of generating a molecule that is invalid with respect to our implicit hydrogen model.

Of course, we are designing molecules in 2D while the actual protein-ligand interactions take place in 3D. Therefore, there will be situations where a promising ligand does not bind as effectively as expected. However, for a given 2D representation there are only a limited number of valid 3D conformations that must be investigated. Indeed, 2D methods have been shown to be successful in the literature.[18–20]

We assess the effectiveness of our method by using two data sets from the Directory of Useful Decoys (DUD).[44] The first contains molecules that are complementary with an active site on the COX2 protein and the second considers the EGFR protein. With the generated set of molecules to

hand, we perform a docking of each generated molecule with the selected receptor on the protein. There are many tools available (FlexX,[45] DOCK,[46] LigandFit[47] and GOLD[48]) for performing docking and we use Fred.[49] To provide comparative results we also perform a virtual docking of the entire input set, a random sample of molecules from the projection set, a subset of projection molecules determined using a fingerprint similarity method, and a random sample of molecules from a large set of decoy molecules for that particular receptor. To provide an evaluation measure that does not utilize docking scores, we test the ability of our method to generate known binders and compare this to a similar experiment based on fingerprint descriptors.

2 Method

We begin by giving an overview of our method, please consult Figure 1 for a diagrammatic description.

The first step is the construction of the projection set. To accomplish this we make use of the fragments present in the molecules of the input set. With the functional groups and carbon scaffolds to hand, we construct a simple statistical fragment model which allows the generation of new chemical structures. This procedure is discussed in Section 2.2.

We then turn our attention to the problem of performing alignment and computing the similarity of molecules. To model the distribution of the input molecules we will require an alignment of the molecules in the input set. This will allow us to construct a vectorial representation of each molecule and estimate a distribution on this vector set. Therefore, generated quasi-molecules will be of a vectorial form.

However, to compute the best mapping between the generated quasi-molecules and the projection molecules we will require a measure of similarity. Due to the vectorial form of the generated quasi-molecules, this must be computed in the vector space. Thus, the molecules of the projection set must also be embedded in the same vector space.

The result of this is that we require a global alignment over the molecules of the input and

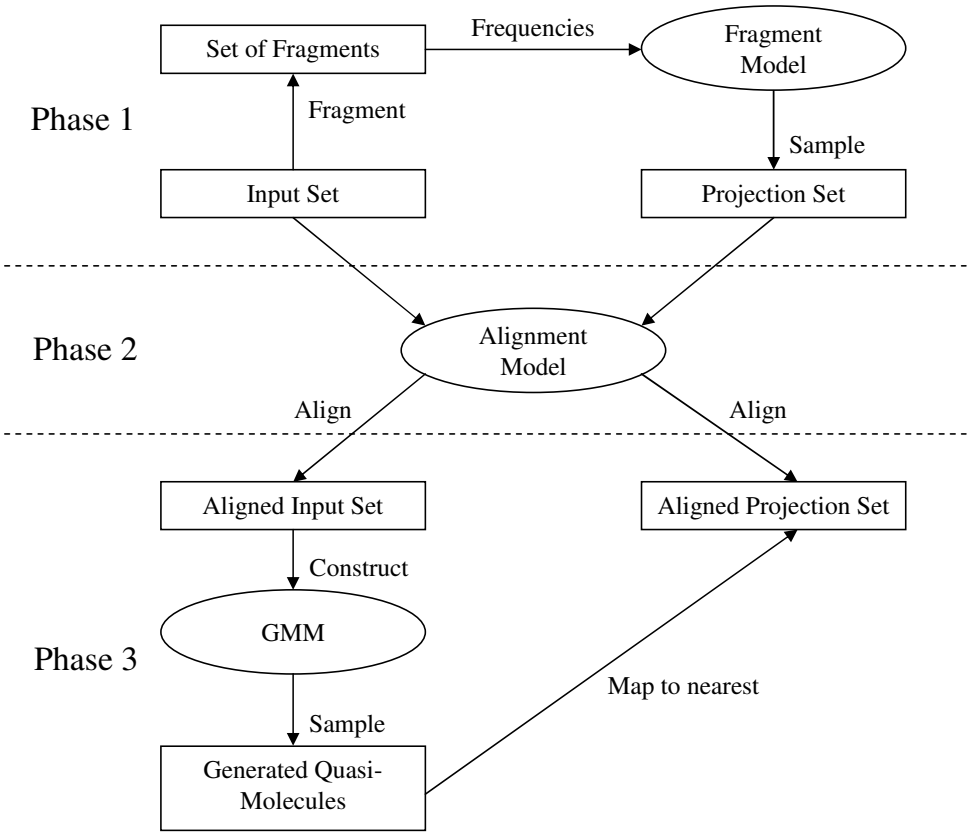


Figure 1: An overview of our method for generating chemical structure.

projection sets. Due to the size of the projection set this step must be as efficient as possible while still producing high quality alignments. We describe our solution to this problem in Section 2.3.

The final step is to construct a model of the structural variations present in the input set. To do this we fit a Gaussian mixture model (GMM) over the vectorized set of input molecules. To estimate the parameters of the GMM accurately we work in a dimensionality-reduced version of the vector space. This is computed from the principle modes of variation present in the input and projection sets. New generated quasi-molecules are sampled from the GMM and then each is mapped to the most similar molecule from the projection set. The subset of projection molecules mapped to in this way are the set of generated molecules. This process is discussed in Section 2.4.

2.1 Molecule Representation

We represent molecules as attributed relational graphs. Formally, a graph G is defined to be a tuple $(\mathcal{V}, \mathcal{E}, w_{\mathcal{V}}, w_{\mathcal{E}})$ which has vertex set $\mathcal{V}$, edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, weight function on the vertices $w_{\mathcal{V}} : \mathcal{V} \to [0,1]$ and weight function on the edges $w_{\mathcal{E}} : \mathcal{E} \to (0,1]$.

We map chemical structures to graphs by using the weight functions $w_{\mathcal{V}}$ and $w_{\mathcal{E}}$ to label the vertices with atoms and the edges with bonds. Beginning from a set of *input molecules* represented by relational graphs and denoted by the set $\mathcal{S}$, we construct an *adjacency matrix* $\mathbf{S}_k$ for each graph $S_k \in \mathcal{S}$. The adjacency matrix of a graph is defined as follows:

$$S_k(u,v) = \begin{cases} w_{\mathcal{V}}(u) & \text{if } u = v \\ w_{\mathcal{E}}(u,v) & \text{if } u \neq v \text{ and } (u,v) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Weights are assigned to an atom ($w_{\mathcal{V}}$) on the basis of the number of valence electrons it possesses. We consider two atoms to be similar if they have similar valence configurations. The edge weights ($w_{\mathcal{E}}$) are assigned by bond strength, i.e. a double bond has a different weight than a single bond. We do not distinguish between different types of bonds (i.e. H-O and H-N bonds have the same weight) and all weights are scaled to the required numerical ranges given above. We adopt an implicit hydrogen model in which all Hydrogen atoms are removed from the input molecules. These can be added back later when complete molecules are required.

However, adjacency matrices constructed through the above process are unsuitable for forming the model. This is because there is no ordering on the vertices of the input molecules; the same vertex in different molecules may represent different structural information. This problem is solved in the *alignment phase* of our approach which places the vertices of all input molecules in a *canonical order*.

The graph represented by an *aligned adjacency matrix* may be transformed into a *vectorial rep-*

resentation by stacking the columns of the matrix to form a vector[1]. To ensure the vector describing each molecule is the same length, the aligned adjacency matrices are padded to accommodate the size of the largest molecule. If the number of vertices in the largest molecule is n, then the padded aligned adjacency matrices are of size $n \times n$ and the vectorial representations are of length n^2. For all results reported in this article, $n = 40$.

2.2 Constructing the Set of Projection Molecules

The fundamental basis of our method is to generate new graphs (quasi-molecules) according to a generative model of the graphs, and then use a set of valid molecules to locate a similar molecule to the generated graph which has a valid chemical structure. We therefore require a method of generating chemical structures in a particular region of graph-space. We term this set of molecules the *projection set* with the symbol $\mathcal{P}$ since these graphs will form the range of the projection function that maps a generated quasi-molecule to a valid molecule.

The first step is to segment each graph in $\mathbb{S}$ into its constituent fragments. We then construct a model describing the probability with which each fragment appears in the input molecules. Finally, we use this model to construct the graphs in the projection set by repeatedly combining fragments identified in the first step with probabilities computed in the second step.

2.2.1 Segmenting the Input Molecules

Graph segmentation is a well known topic in pattern recognition and computer vision and there are many algorithms that perform well at the task. Perhaps the best known algorithm is Shi & Malik's Normalized Cut.[50] However, for the purposes of segmenting chemical structures, a more specialized method is required that considers the chemistry of the molecule as well as the structure of the graph.

Drug discovery often proceeds by making small changes to drug-like molecules by altering

[1] Although both representations describe a vector space, when we come to construct the model it is more intuitive to think of graphs as vectors than matrices.

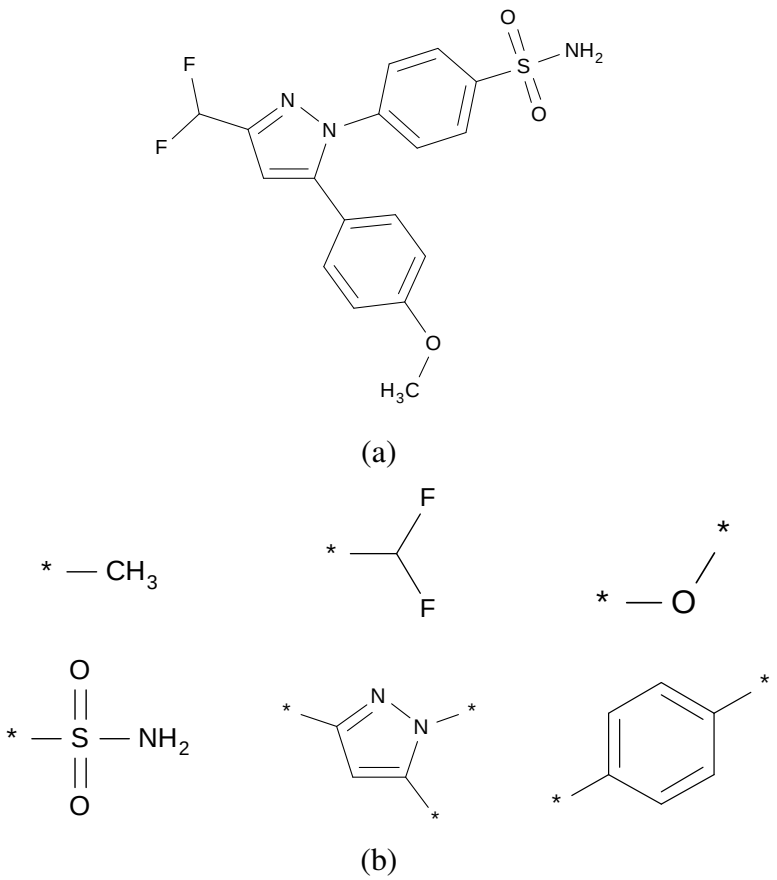


Figure 2: The fragments resulting from running Chomp on the molecule (a) are shown in (b).

their functional groups, in an effort to make the molecule bind more tightly to an active site or improve other desirable properties. Since the set of projection molecules will be built from the fragments identified in this step we would like the way we segment the molecules to reflect the way a molecule might be improved during the process of lead refinement. In practice this means that we wish to decompose molecules into functional groups and scaffolds.

The tool *Chomp*[51] provides the functionality we require. It takes a molecule file as a parameter and returns the set of fragments contained in that molecule. It is possible to specify specific fragmentation rules by providing a set of SMARTS rules as a parameter, however, in our experiments the default fragmentation rules were used. The result of applying Chomp to a molecule is seen in Figure 2. By applying Chomp to each molecule in the input set we can compute a function $f(S_i, F_j)$ that gives the number of occurrences of a fragment F_j in an input molecule S_i.

2.2.2 Constructing the Projection Molecules

With the set of fragments contained in each input molecule to hand, we can build the model describing the distribution of fragments in the input set. Using the fragment function f from the previous step, we compute a probability distribution $\mathrm{P}(F_i, j)$ which is an estimate of the probability that fragment F_i occurs j times in an input molecule.

The construction of each projection molecule then proceeds as follows; first, the distribution $\mathrm{P}(F_i, j)$ is sampled for each F_i. The result of this is an *indicator vector*. An element i of this vector indicates whether fragment F_i should appear 0,1,2,... times in the new projection molecule. We use the term indicator vector here as it is not always possible to use all fragments described by the indicator vector in the construction of a molecule due to structural constraints.

The next step is to join the fragments. Starting with the fragment which has the largest number of connections, we iteratively join the fragments at random connection points, in descending order of the number of connection points in the fragment. This ordering ensures that as many fragments from the indicator vector as possible can be joined, but in some cases, the connection points may run out before all fragments are joined.

Note that although small active molecules are usually preferred in the context of drug discovery, we prefer to try to use all fragments contained in the indicator vector. This results in projection molecules that are similar in size to the input molecules. The reason behind this is that the projection molecules are used as the mapping targets of the generated quasi-molecules. Since the quasi-molecules are constructed by analyzing the input molecules, mapping targets that are more similar to those in the input set are preferred and will result in better mappings.

Although constructing projection molecules through the above process usually results in useful mapping targets, it can still be helpful to remove obviously useless targets at this stage. This can be done with a filter based on a coarse threshold on the average molecule size in the input set.

To summarize, the key aim of the process described in this section is to generate a number of different molecules which are reasonably structurally similar to the input set and can be used for the projection step of our algorithm. It is important to note at this point that, while these molecules

are structurally similar to the input set, this generation process does not have the finesse of our generative model of structure which is described later in Section 2.4. The idea is that we use the generative model of structure to prune the projection set down onto a much smaller and more structurally interesting set. We show that this is indeed the case in the experimental section.

Finally, we note that there are many other possible ways that the projection set may be constructed. For example, the set could be mined from a large catalogue of molecules using a simple similarity measure. Alternatively, a more complex fragmentation/combination process could be used that models chemistry more accurately. Furthermore, the joining process proposed in this section could be improved by testing different attachment locations and using a heuristic to choose the best.

2.3 Aligning the Molecules

In order to compare graph structures in a consistent way, it is first necessary to compute a canonical ordering for the vertices and then apply this ordering to the vertices of each graph. This process can be accomplished by performing *graph alignment*. The result of applying graph alignment to the graphs of the input and projection sets is that the vectorial descriptions computed for each graph will now reside in the same region of vector space. Thus, these vectorial representations will now be suitable for forming the basis of a model. Furthermore, since generated quasi-graphs will also reside in this space, the mappings to projection graphs can be easily computed. We make use of the Gold & Rangarajan[35] graph alignment algorithm to compute all alignments.

However, it is not possible to align the molecules using a single representative due to the large diversity in the projection set (and to a lesser extent the input set). Specifically, problems occur when aligning structures of different size magnitudes. Therefore, we propose a hierarchical solution to the alignment problem.

In our solution we use all molecules in the input set as representatives since we expect these to be good examples of the structures present in the projection set. We construct a hierarchical alignment over the molecules of the input set. This ensures that each alignment step takes place

between similar molecules where possible. Once the input set is aligned, we consider each projection molecule in turn by first aligning it to the input molecule S_k that it is most similar to. This projection molecule can then be placed in global alignment by applying the same alignment steps that were used for molecule S_k.

The computational complexity of this procedure is dominated by computing the alignments, i.e. applying the Gold & Rangarajan algorithm[35] to a pair of graphs. Thus, the following complexity descriptions are given in terms of the number of alignments that must be computed.

The success of our approach is directly related to how many molecules it is feasible to have in the projection set, and therefore, the addition of a new projection molecule should have low computational complexity. This is indeed the case as our hierarchical alignment only requires an amount of work linear in the size of the input set; the complexity of aligning the whole projection set is $O(|\mathcal{S}||\mathcal{P}|)$. Adding in the work to compute the hierarchical alignment of the input set we require $O(|\mathcal{S}|^2+|\mathcal{S}||\mathcal{P}|)$ alignments operations in total. In contrast, a full pairwise alignment requires $O((|\mathcal{S}|+|\mathcal{P}|)^2)$ alignment operations.

2.3.1 Hierarchical Alignment of the Input Set

To compute the hierarchical alignment of the input molecules we must compute the distances[2] between all pairs of graphs in the input set. We record the distance between input graph S_i and S_j in the distance matrix $D(S_i,S_j)$. We construct the hierarchical alignment tree by computing the set of graphs that will be present at each level of the tree, starting from the deepest level first: d. Using the matrix $\mathbf{D}$ we approximately solve the *stable roommates problem* which involves finding the set of pairings such that the total sum of the distances of the pairings is minimized. In each pairing the smaller graph will be aligned to the larger one and then the larger one moves up to the next level of the tree (at depth $d-1$).

At depth $d-1$ of the tree we have approximately $\frac{|\mathcal{S}|}{2}$ graphs and we remake the distance matrix $\mathbf{D}$ to include only graphs present at this depth. We recursively solve this problem until we are left

[2]The distances are computed from the graph alignments.

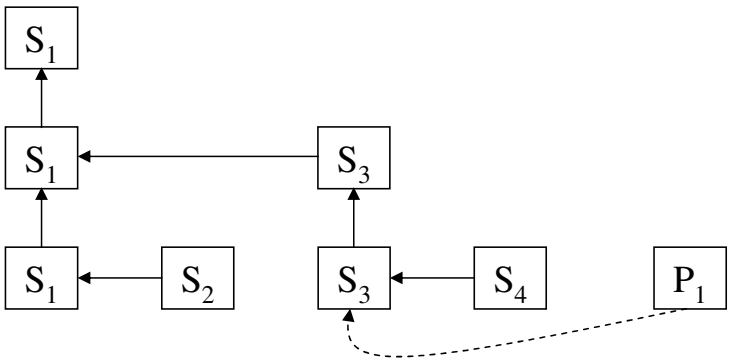


Figure 3: An example pairwise hierarchial alignment tree.

with one graph that is the root of the tree, which by definition must be the largest graph in the input set.

Figure 3 shows an example alignment tree. Consider the alignment of graph S_3, it is the larger of its pair (S_3, S_4) at the bottom level of the tree and thus no alignment is applied at this level. At the next level it is the smaller of the pair (S_1, S_3) and it is therefore aligned to S_1. We are now at the root of the tree so no more alignments are necessary.

2.3.2 Hierarchical Alignment of the Projection Set

Applying the alignment to the projection graphs is very similar. We require the computation of a new distance matrix $\mathbf{D}'$ of size $|\mathcal{S}|$ by $|\mathcal{P}|$ where the distance between input graph $S_i \in \mathcal{S}$ and projection graph $P_k \in \mathcal{P}$ is stored in $D'(S_i, P_k)$. We can then find the input graph S_i closest to projection graph P_k. We align P_k with S_i and then the alignment proceeds up the tree as is the case for input graphs.

As an example consider the alignment of P_1 in Figure 3. We find that it is most similar to S_3 so it is initially aligned to this graph. Then, to place it in global alignment, the sequence of alignments for S_3 (described above) are applied.

2.4 Generating New Molecules

The key element of our method is a statistical generative model which can generate new graph examples from the same distribution as the input set. We produce a set of generated graphs by sampling from a Gaussian mixture model (GMM) which is constructed over the vectorized aligned adjacency matrices of the input molecules. We call the vectors sampled from this model *quasi-molecules* since, while they drawn from the distribution of input molecules, they are not legitimate molecular graphs. The quasi-molecules are then projected onto the projection set $\mathcal{P}$ of nearby molecules to find a valid molecular structure.

2.4.1 Fitting the GMM to the Input Data

The distribution of input molecules is too complex to model using a single multivariate distribution. Therefore, we make use of a GMM. However, the model parameters of a GMM are very difficult to estimate in a high dimensional space; such as the one in which the vectorised input and projection molecules reside. Hence, an important part of this step is the construction of an additional vector space.

This is computed by applying a principle component analysis (PCA) to the combined set of vectorised input and projection molecules. Thus, the key structural variations from both sets are captured in this vector space.

To put this formally, each aligned adjacency matrix $\mathbf{S}_k$ that represents an input molecule, and each aligned adjacency matrix $\mathbf{P}_k$ that represents a projection molecule are transformed into vectors $\mathbf{s}_k$ and $\mathbf{p}_k$ respectively. The vectorisation process is performed by simply stacking the columns of the adjacency matrices. With the vectors to hand we compute the PCA transform which results in a projection matrix $\mathbf{\Phi}$ of eigenvectors and a mean vector $\boldsymbol{\mu}$. We select the top t components of the projection matrix resulting in a matrix $\tilde{\mathbf{\Phi}}$ which is $\mathbf{\Phi}$ truncated after t columns. The value chosen for t is dependent on the amount of sample data available. It represents a tradeoff between the complexity of finding the model parameters for the GMM and the amount of variance in the input and projection molecules which is explained by the PCA space.

Both the input and projection vectors may now be projected into this new space. Note that the prime symbol represents vectors in which reside in the computed PCA space.

$$\mathbf{s}'_k = \tilde{\mathbf{\Phi}}^T(\mathbf{s}_k - \boldsymbol{\mu}) \quad (2)$$

$$\mathbf{p}'_k = \tilde{\mathbf{\Phi}}^T(\mathbf{p}_k - \boldsymbol{\mu}) \quad (3)$$

We are now ready to fit a GMM to the input molecules. These are described by the set of vectors $\{\mathbf{s}'_1, \mathbf{s}'_2, ..., \mathbf{s}'_{|\mathbb{S}|}\}$. We use the algorithm proposed by Figueiredo & Jain[42] to estimate the GMM parameters. The algorithm is also capable of estimating the number of components in the mixture model. The output of the algorithm is a k-component multivariate normal distribution.

Using a single multivariate normal distribution $p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ we can define a k-component multivariate normal distribution as follows:

$$p_k(\mathbf{x}) = \sum_{i=1}^{k} p(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) w_i \quad (4)$$

The mixing weight w_i is the probability that a sample from the GMM would be drawn from component i. As such the sum of the mixing weights must equal one $\sum_{i=1}^{k} w_i = 1$ for $i \in \{1, ..., k\}$: $w_i \geq 0$. Therefore, a component i of the GMM is represented by the triple $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, w_i)$ which denotes the mean, covariance and weight respectively.

In order to sample from component i of the GMM we require the set of eigenvalues and eigenvectors from covariance matrix $\boldsymbol{\Sigma}_i$, $\boldsymbol{\Lambda}_i$ and $\boldsymbol{\Phi}_i$ respectively. The eigenvectors describe the principle components of variance in the subset of vectors that make up this component of the GMM and the eigenvalues describe the variance of each principle component.

2.4.2 Sampling from the GMM

To sample from the GMM with distribution $p_k(\mathbf{x})$ we first choose which component i should be sampled. The probability that component i is sampled is simply its weight w_i. To sample from

component i we generate a parameter vector $\mathbf{b}$. This is produced by sampling from a number of 1D normal distributions with zero mean and variance determined by the diagonal values in $\mathbf{\Lambda}_i$. If $\mathcal{N}(Mean, Variance)$ is a 1D normal distribution then the parameter vector is computed as such:

$$\mathbf{b}(j) \sim \mathcal{N}(0, \mathbf{\Lambda}_i(j,j)) \tag{5}$$

We can then project the parameter vector $\mathbf{b}$ on the components of the distribution $\mathbf{\Phi}_i$ and add the mean $\boldsymbol{\mu}_i$ to compute a vectorial representation of a new quasi-graph $\mathbf{g}'_k$.

$$\mathbf{g}'_k = \boldsymbol{\mu}_i + \mathbf{\Phi}_i \mathbf{b} \tag{6}$$

2.4.3 Mapping the Samples to the Projection Set

We are now in a position to compute the nearest projection graph to each generated quasi-graph. We define a function ρ that maps a generated quasi-graph to the graph from the projection set at minimum Euclidean distance.

$$\rho : \mathbf{g}'_k \rightarrow P_j \text{ where } j = \underset{i}{\operatorname{argmin}} \operatorname{dist}(\mathbf{p}'_i, \mathbf{g}'_k) \tag{7}$$

Therefore, the final multiset of generated molecules $G_k \in \mathcal{G}$ is found by applying ρ to each generated quasi-molecule, $G_k = \rho(\mathbf{g}''_k)$. A multiset is used to allow repetitions of projection molecules since it is likely that a repeated molecule will indicate a statistically interesting molecule to explore.

3 Results

We make use of a number of experimental methods to evaluate our generative model of chemical structure. We begin with a visual inspection of the generated molecules. Next, we analyze the various molecule distributions by visualizing the first two dimensions of the PCA space. We can also visualize mappings between sets of graphs by overlaying them on the distribution plots. The

alignment process can provide information on the applicability of the molecules in the projection set, so we include histograms depicting what percentage of projection molecules map to each input molecule.

Using these results we will assess the performance of the hierarchical alignment step, the applicability of the projection molecules as targets for the mapping of the generated quasi-molecules, the suitability of fitting a GMM and the quality of the final generated molecules. We can further assess the quality of the generated molecules by applying them to our evaluation domain which is the task of docking molecules to active sites in proteins.

By using a set of molecules that dock with high affinity to the selected active site as input to our method, we hope to generate molecules that also dock with high affinity. Although we do not expect to replicate all of the pharmacological properties that the molecules in the input set exhibit, we do expect that the generated molecules will perform well in the docking test. We compute all docking scores using Fred[49] and construct 3D representations of our generated molecules using Omega.[52]

It is well known that the docking results can be misleading[53] and sometimes not indicative of real-world binding affinity. Therefore, we include an experiment that is based on generating known binders and does not rely on docking scores. We begin by removing some of the molecules from the input set and adding, or hiding, these in the projection set. The enrichment rate is then the ratio of the fraction of hidden input molecules (a) in the generated set $\mathcal{G}$ to the fraction of hidden input molecules (b) in the projection set $\mathcal{P}$.

$$EnrichmentRate = \frac{\frac{a}{|\mathcal{G}|}}{\frac{b}{|\mathcal{P}|}} \tag{8}$$

To provide comparative results, we compute fingerprint descriptors for the molecules and use these to rank projection molecules by similarity to the input molecules. We use FP2 Daylight style fingerprints which are computed using OpenBabel[54] and folded to 512 bits. For each projection molecule, the mean similarity to all input molecules is found using the Tanimoto coefficient and it is this mean value that is used to compute the ranking.

We would like to compare this enrichment result to an enrichment rate obtained using a fingerprint method. However, as the fingerprint method produces a ranking of projection molecules, rather than a subset, it is not directly comparable. To provide a level of comparison, we take the average number of molecules produced by our method over 10 runs; i.e. the mean size of the generated set. Let us call this number $|\bar{\mathcal{G}}|$. We then search in the top $|\bar{\mathcal{G}}|$ molecules as ranked by the fingerprint method for hidden input molecules and use this as the value for a in Eq. (8).

We form our test sets from the Directory of Useful Decoys[44] as follows. From the set of active ligands we remove any duplicates that may be present (the duplicates would manifest themselves as conformers of another active ligand and, since we are not using any 3D information in our process, they are superfluous). We then pre-process the ligands to remove all hydrogen atoms. Next, we use FredTool[49] to setup the active site using a pre-docked ligand. Using this information we perform a preliminary docking of the set of active ligands and select all the top performing ligands above a threshold. These sets of ligands form the input to our model.

3.1 Results from the COX2 Data Set

Our first data set is based on molecules that are active against the COX2 protein. We begin by giving examples of the molecules used as input. Figure 4 shows the nine most effective binding molecules from the data set. Molecule S_n has the n-th highest binding affinity. Figure 5(a) shows molecule S_1 docked in the active site. It is for this site that we aim to generate molecules that bind with high affinity.

The details for the results in this section are as follows. Using the preprocessing procedure described above, 39 input molecules were selected from the DUD data set. The projection set contained 500 projection molecules which were constructed by joining 45 unique fragments discovered in the input molecules using the procedure described in Section 2.2. The top 5 principle components were selected (parameter $t = 5$) to form the PCA space resulting in a total of 50% of the variance being explained. Using the GMM modeling the distribution of input molecules, 200 vectors were sampled that represent the generated quasi-molecules. These quasi-molecules were

mapped to the structures in the projection set resulting in a subset of 80 projection molecules on average being selected to form the final generated set. When using Omega, the average number of conformers for a molecule from the generated set was 50. For the enrichment experiments, 10 molecules were removed from the input set and hidden in the projection set resulting in the an input set containing 29 molecules.

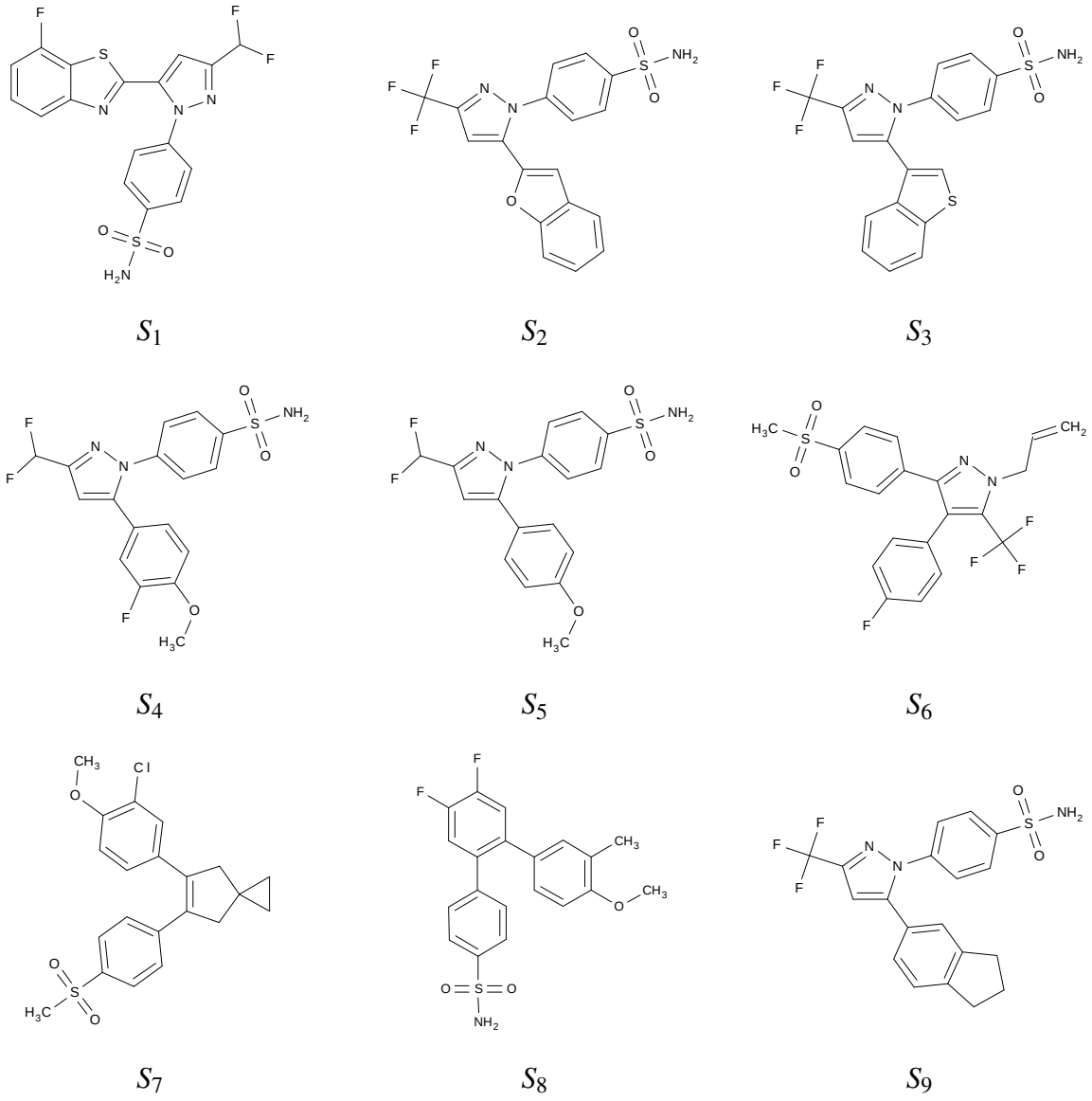
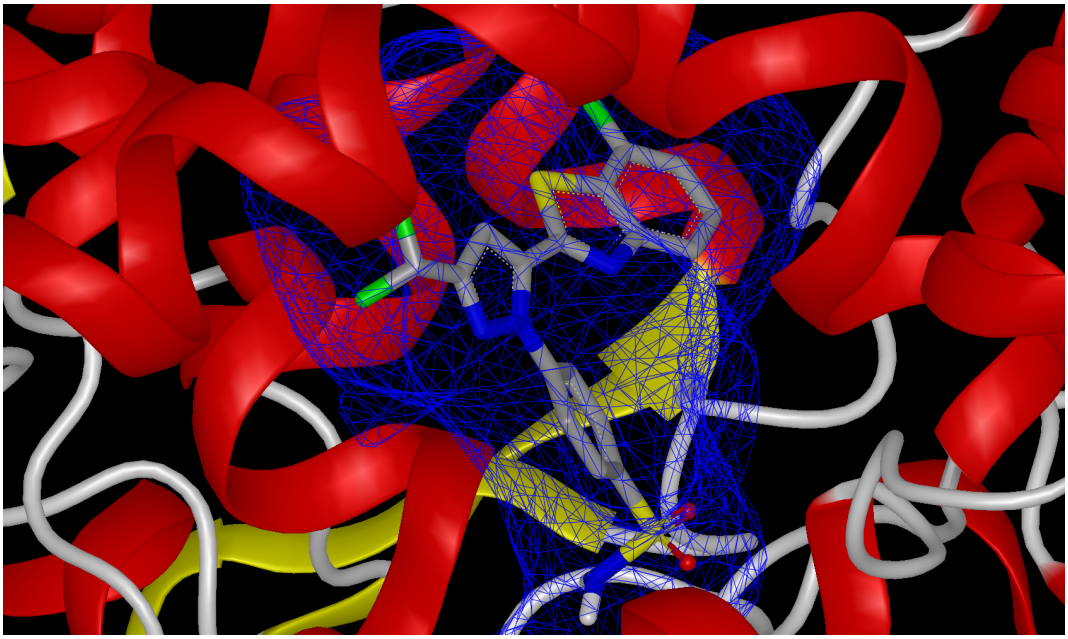
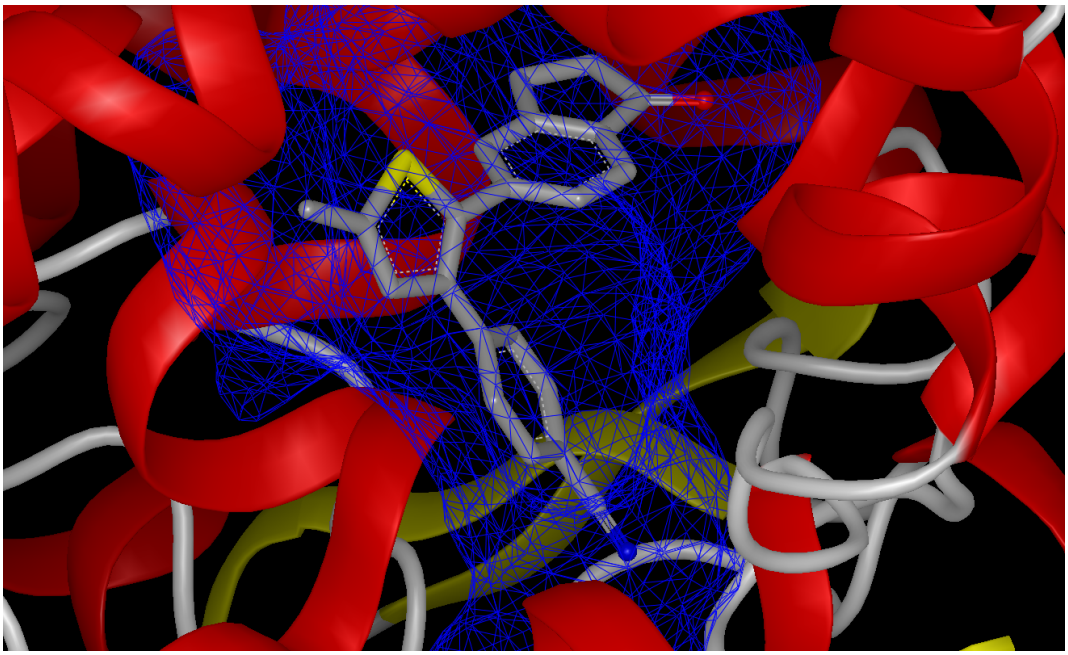


Figure 4: Nine molecules from the COX2 data set. Molecule S_n has the n-th highest binding affinity.

Figure 6 shows the top 6 highest-binding molecules that we have generated from this data set.

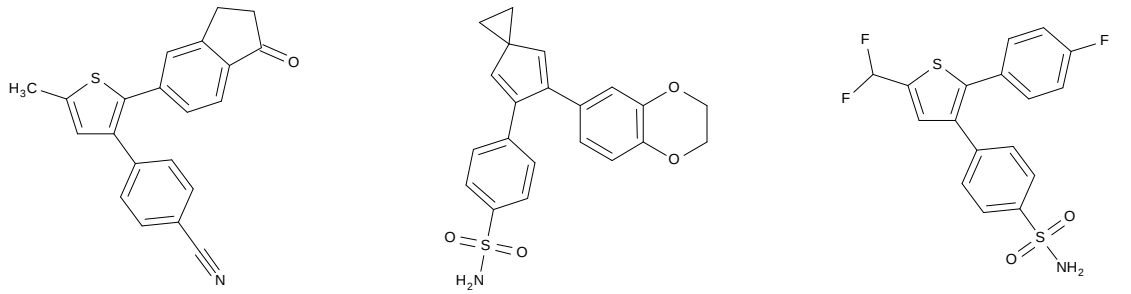


(a)



(b)

Figure 5: The active site of interest in the COX2 protein. In subfigure (a) molecule S_1 from Figure 4 (the molecule from the input set with highest binding affinity) is shown in its optimal pose as computed by Fred. In subfigure (b) molecule G_1 from Figure 6 (the molecule from the generated set with highest binding affinity) is shown in its optimal pose. Note the similarity of the computed poses.



G_1 G_2 G_3

G_4 G_5 G_6

Figure 6: Six molecules we have generated using our approach from the COX2 data set. Molecule G_n has the n-th highest binding affinity.

Note the similarities of structure observed between the input molecules shown in Figure 4 and the molecules shown in Figure 6. Molecule G_1 is shown docked in the active site in Figure 5(b). Note that in molecule G_6 there is a non-existent functional group. This is an artifact of our fragment joining process combined with the removal of Hydrogen atoms.

We now show PCA plots allowing us to view the various distributions constructed throughout the process. The projection in Figure 7(a) shows the input molecules as red plus signs and the generated vectors from the GMM as crosses. In this case the estimation algorithm used two normal distributions to fit the data and these are indicated by ellipses. In this plot vectors have been sampled spanning the whole space of the input molecules and in areas where the input molecules are clustered, so are the generated vectors.

Figure 7(b) shows just the input and projection molecules. The color of each circle indicates the binding affinity of the projection molecule. The hierarchial alignments clearly have a large impact on the distances between molecules in the vector space and this can be clearly seen in the PCA plot. Notice the large cluster of input molecules in the middle right of the plot. This has arisen since the distances between alignments of these molecules are all relatively low values.

We can now begin to evaluate one of the main goals of this research: are the set of projection molecules close enough to the input distribution to effectively allow generated graphs to be mapped to them? In other words, do we have diversity in the projection graphs and are these diverse regions near clusters of input molecules? By a visual inspection of the PCA plot in Figure 7(b) we can see that this requirement has been suitably fulfilled. Another important aspect is that the projection molecules we constructed in the process are distinct from the input molecules, i.e. we are not just duplicating the input set. We have tested this and found no duplication of the input set in the projection set.

We can assess the effectiveness of the projection set further by considering the alignment step. Ideally, we would like approximately the same number of projection graphs to align to each input graph. This would indicate that we have constructed projection molecules that are close in distance to the whole range of input molecules. The histogram in Figure 8 shows the percentage of

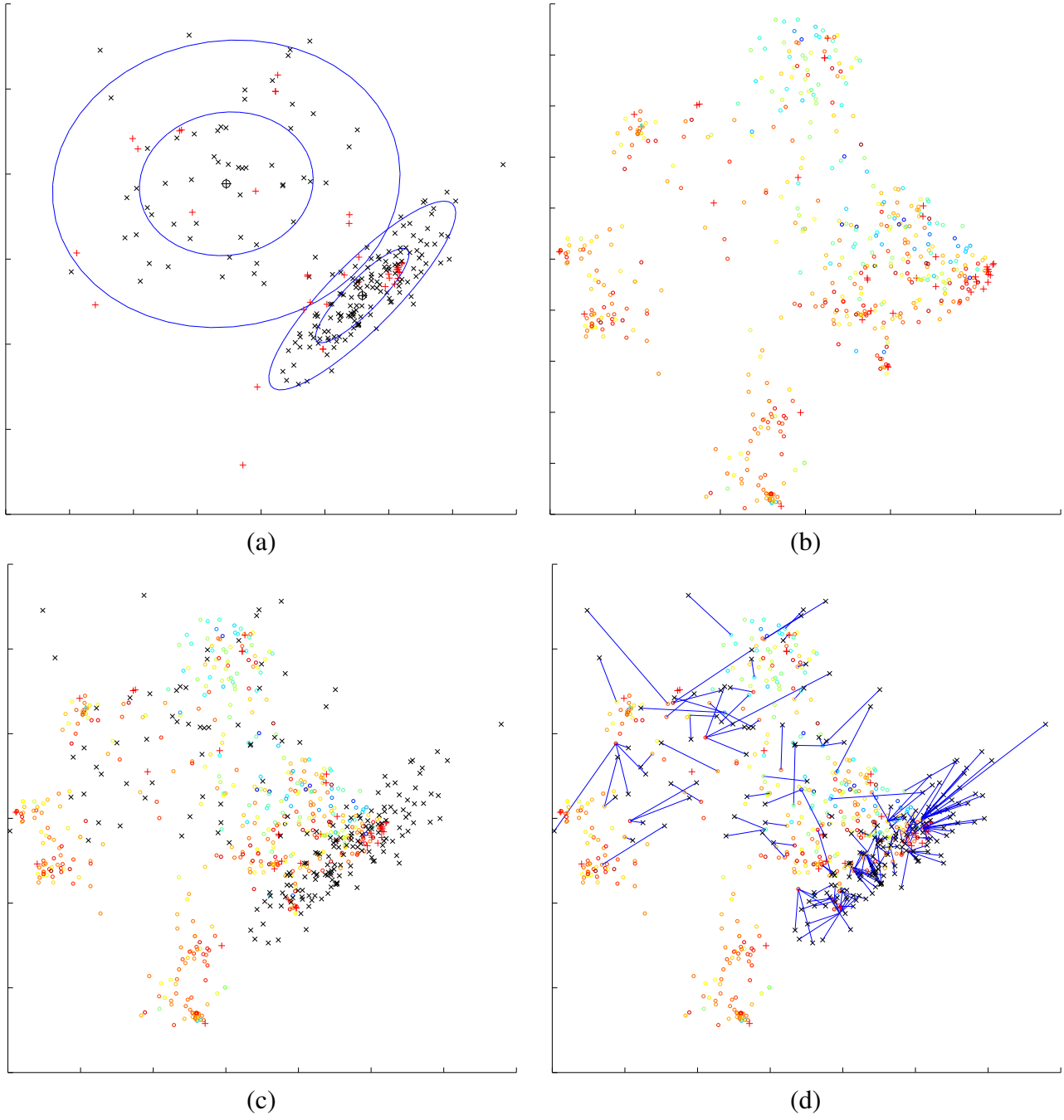


Figure 7: PCA projections of the input, projection and generated sets using the COX2 data set. Subfigure (a) shows in the input set (plus signs) and the generated quasi-molecules (crosses) in the PCA space. Ellipses are also drawn indicating the normal distributions chosen by the GMM. Subfigure (b) shows the input set (plus signs) and the projection set (colored circles). In (c), the generated quasi-molecules (crosses) are also projected into this space. Finally, in (d) the chosen mappings between projection molecules and generated quasi-molecules are shown. The color of each projection molecule marker indicates the docking value, blue is low affinity and red is high affinity.

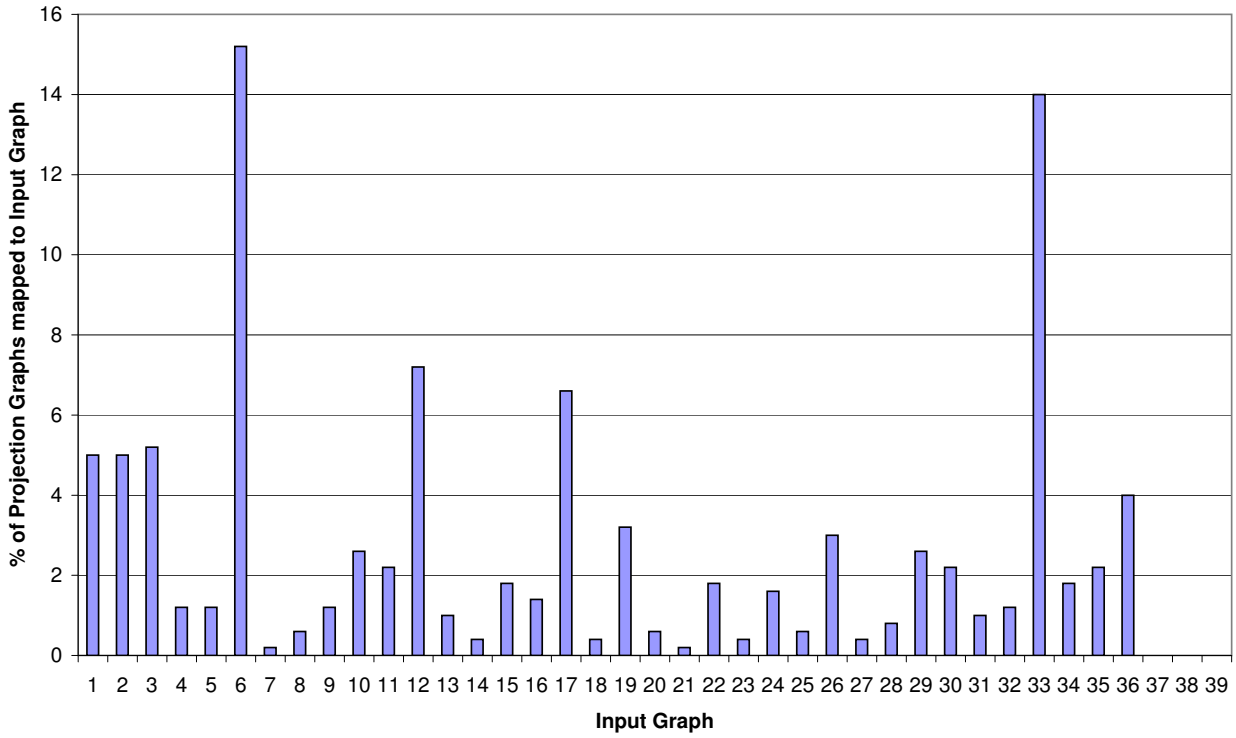


Figure 8: A histogram showing the percentage of projection graphs that align to a specific input graph in the hierarchical alignment step.

projection molecules that map to an input molecule. The large discrepancies of input graphs 6 (the largest) and 33 (the smallest) can be explained by the projection molecules that are significantly smaller and significantly larger than those in the input distribution. It is likely that a projection molecule falling outside the expected number of atoms will be aligned to either of these graphs.

We now consider the generated quasi-molecules both in the distribution they form and the mappings to the projection molecules. In Figure 7(c) we augment the previous plot with crosses to indicate generated quasi-molecules and, in Figure 7(d), we overlay mappings between generated quasi-molecules and projection molecules.

We see that the generated quasi-molecules cover the distribution of input molecules well, especially near the cluster on the right side of the plot. The remainder of the space is sparsely populated with input molecules and correspondingly the generated quasi-molecules also sparsely cover this space. The mixture of normal distributions seems an appropriate way to model this data. The mappings are appropriate, however, it is important to note that we are only visualizing 2 dimensions of

a larger space and even moving to 3 dimensions makes the mappings appear significantly better.

Near the cluster of input graphs on the right of the plot there are not enough projection molecules and thus many generated quasi-molecules are mapped to the same projection molecule. This indicates that there is not enough diversity in this area of chemical space. However, this could be resolved by populating the projection set with more molecules.

We now describe the results of applying this set of generated graphs to the evaluation domain, that is, docking them to the identified active site on the COX2 protein. As mentioned above, when using Omega, the average number of conformers for a molecule from the generated set is 50. This is a good result as there is plenty of conformational space for Fred to explore to find a high scoring docking pose. On the other hand, there is not too much conformational flexibility as to render the 2D description of the molecules unimportant.

Figure 9 shows the docking scores for five sets of 39 molecules sorted by score (each set is limited to 39 since this is the number of molecules in the input set). Unsurprisingly, the highest scoring set is the input set. The second highest scoring set is the set of generated molecules. Due to the stochastic nature of our approach, we report mean docking scores over 10 trials. The next highest scoring set are the projection molecules selected by the fingerprint approach as described in this section's introduction. Note that the docking values are lower than those of the generated set and they tail off sharply at the end.

To provide further comparison we include the scores of two additional sets. The first is a random sample of 39 molecules from the set of decoys supplied with the DUD data set for the COX2 protein. This shows, as we would expect, that choosing a random sample of chemical structures results in a set of molecules that do not dock well with the active site.

The second data set is a sample of 39 molecules from the projection set (again, due to the stochastic approach, averaged over 10 trials). With this set we are showing that taking the projection set by itself is not enough to produce a set of molecules that dock with high affinity. In other words, the subset of the projection set that is produced by our method is superior.

The results of the enrichment experiments described in the introduction are as follows: our ap-

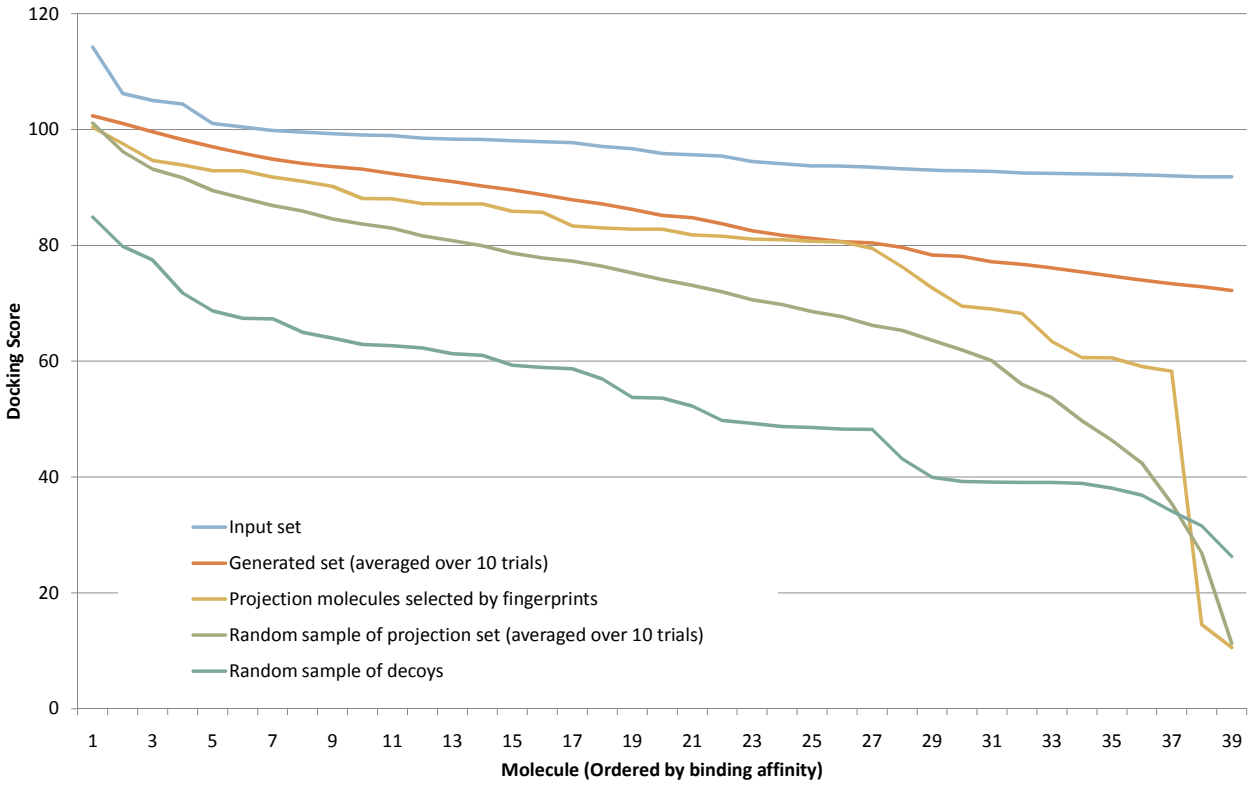


Figure 9: Docking scores for the COX2 data set. The first line represents the docking scores of the input molecules, and the second represents the docking scores of the generated molecules. Only the best 39 generated molecules are included. Next, the top 39 molecules selected by the fingerprint method are given. Finally, a random sample of 39 molecules from the projection and decoy sets are included. Within each set the molecules are sorted by docking score.

proach achieved an enrichment rate of 2.72 and the fingerprint approach achieved 3.44. To ensure that neither method received any bias in which input molecules were removed, the experiments were repeated with 10 different sets of input molecules being removed. Due to the stochastic nature of our approach, for each subset of input molecules that were removed, the experiment was performed 10 times and the results averaged.

The fingerprint method outperforms our approach slightly; however, the docking scores for our approach are superior. This shows that our approach finds structurally interesting molecules which are good binders (according to the docking scores) but are often different from those in the input set. In contrast, the fingerprint method is effective at selecting the hidden input molecules but is not able to find such a diverse set with high binding affinity.

3.2 Results from the EGFR Data Set

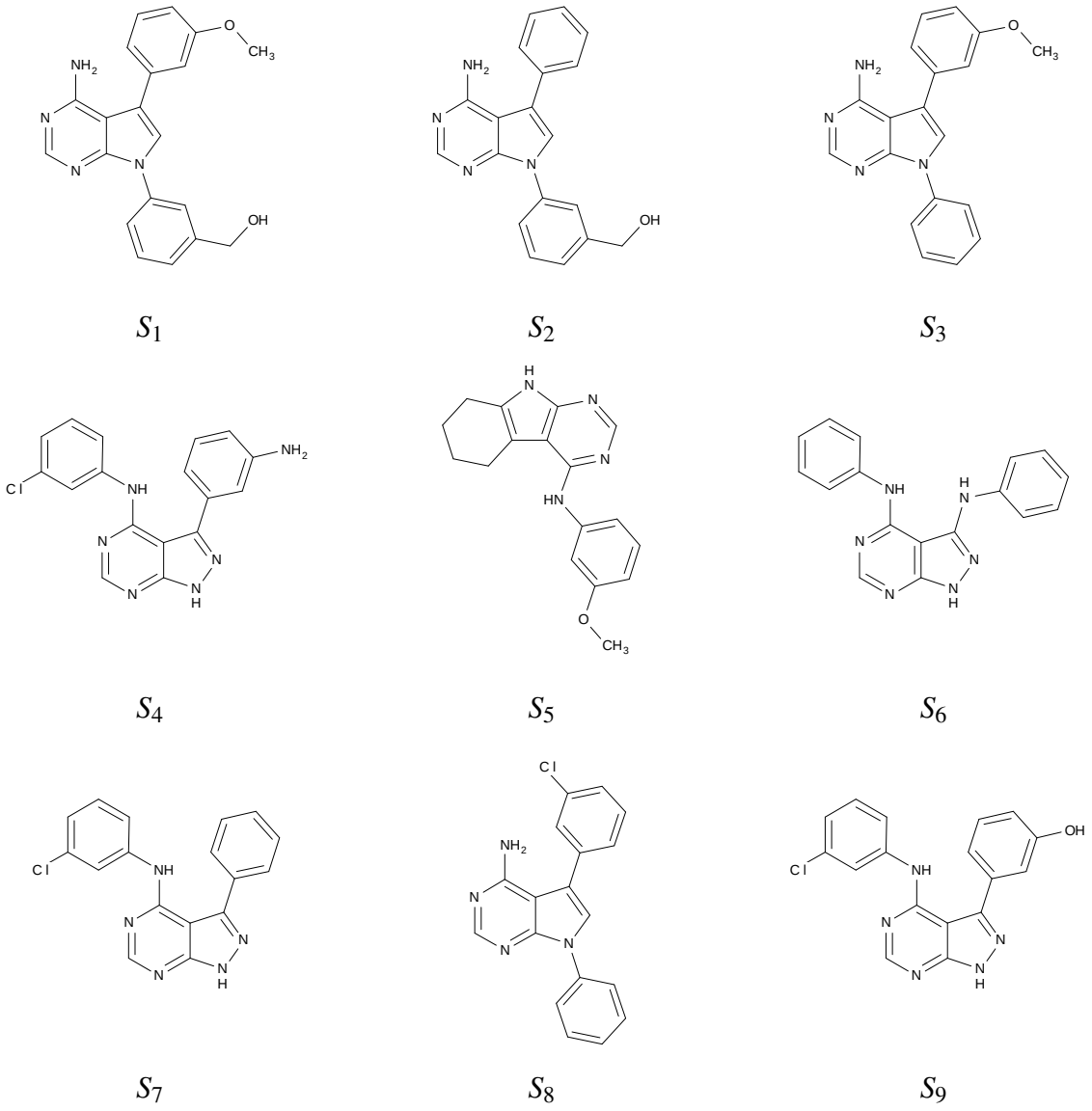
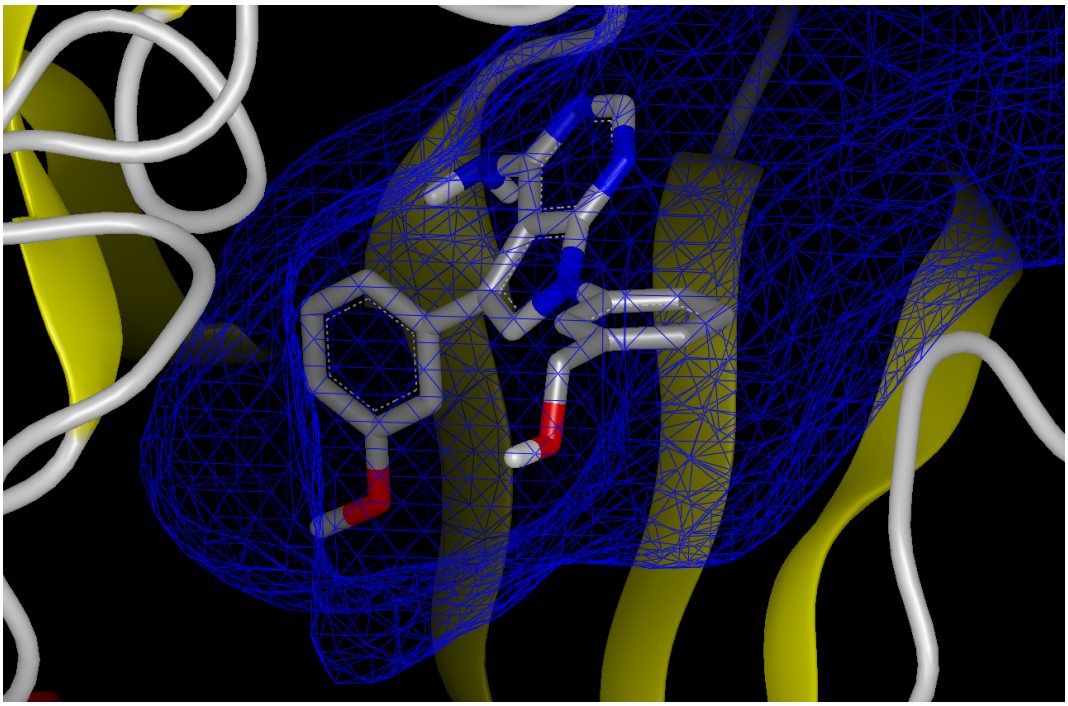
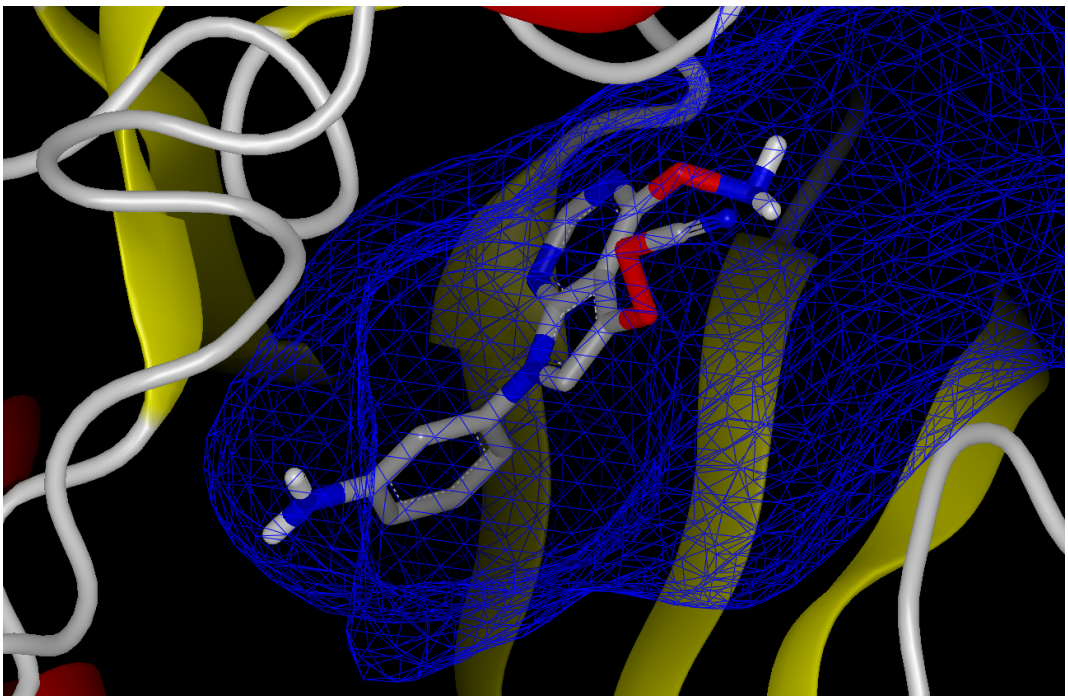


Figure 10: Nine molecules from the EGFR data set. Molecule S_n has the n-th highest binding affinity.

We will present the results from the EGFR data set in the same format as those from the COX2 data set and therefore descriptions of the mechanisms of providing results will be brief in this section. Figure 10 shows 9 different molecules from the input set. Again, molecule S_n has the n-th highest binding affinity for the EGFR active site. Figure 11(a) shows the molecule from the input set with highest affinity docked in the active site. It can be seen that the shape of the active



(a)



(b)

Figure 11: The active site of interest in the EGFR protein. In subfigure (a) molecule S_1 from Figure 10 (the molecule from the input set with highest binding affinity) is shown in its optimal pose as computed by Fred. In subfigure (b) molecule G_1 from Figure 12 (the molecule from the generated set with highest binding affinity) is shown in its optimal pose.

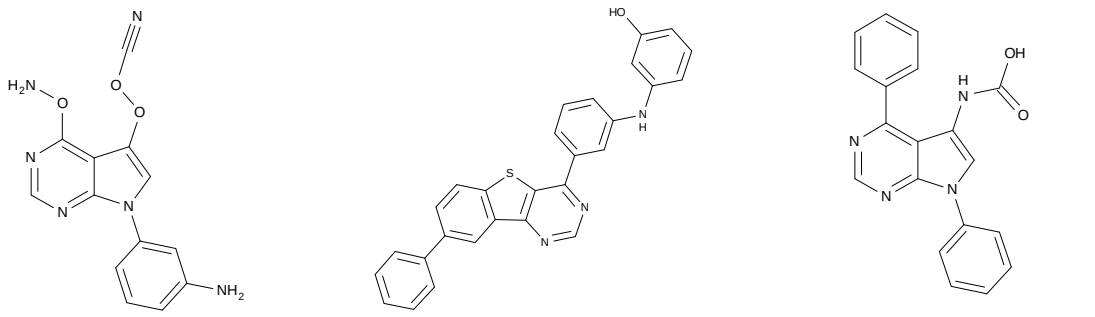
site differs considerably from the active site on the COX2 protein. This is unsurprising of course, however it's more general shape will allow a larger set of molecules to dock successfully than the more specific shape of the COX2 active site. We will examine this issue more closely when we discuss the docking results.

The details for the results in this section are as follows. Using the preprocessing procedure described above, 110 input molecules were selected from the DUD data set. The projection set contained 1000 projection molecules which were constructed by joining 72 unique fragments discovered in the input molecules using the procedure described in Section 2.2. The top 5 principle components were selected (parameter $t = 5$) to form the PCA space resulting in a total of 40% of the variance being explained. Using the GMM modeling the distribution of input molecules, 400 vectors were sampled that represent the generated quasi-molecules. These quasi-molecules were mapped to the structures in the projection set resulting in a subset of 192 projection molecules on average being selected to form the final generated set. When using Omega, the average number of conformers for a molecule from the generated set was 65. For the enrichment experiments, 30 molecules were removed from the input set and hidden in the projection set resulting in an input set containing 80 molecules.

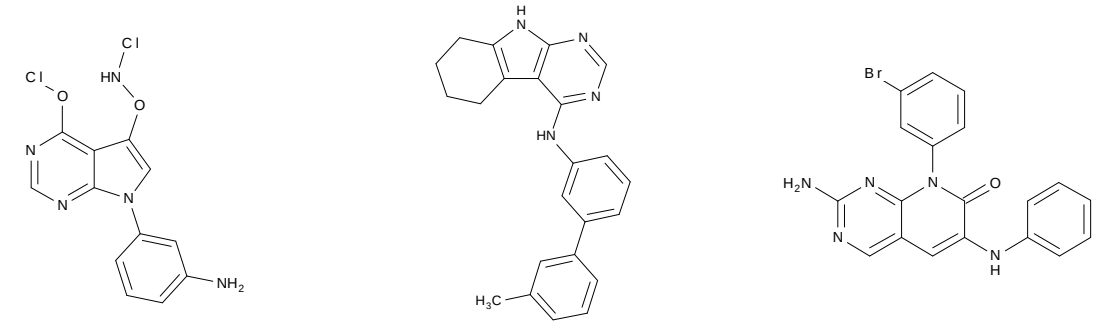
Figure 12 shows the top 6 scoring molecules we have generated using this data set. Some of these molecules are very similar to those in the top scoring set of input molecules (Figure 10). Specifically G_1, G_3, G_4 and G_6 have the same bulky center structure with two smaller functional groups attached. Note that G_6 has been drawn differently due to the optimization based drawing algorithm. Molecule G_1 can be seen docked in the active site in figure Figure 11(b).

We will now show PCA plots beginning with Figure 13(a). This figure shows the input molecules and generated quasi-molecules in PCA space. Also shown are the components of the GMM which accurately represent the distribution of input molecules.

In Figure 13(b) it can be seen that we have successfully generated projection molecules covering the whole of the input space, and they are also clustered in the regions where the input molecules are clustered. The only issue here is that we have generated a large number of re-



G_1 G_2 G_3



G_4 G_5 G_6

Figure 12: Six molecules we have generated using our approach from the EGFR data set. Molecule G_n has the n-th highest binding affinity.

dundant projection molecules in the bottom left of the plot which reduces the efficiency of this projection set.

As in the COX2 results we can use a histogram to assess whether the projection graphs are being generated over the whole space of the input graphs. Figure 14 shows the percentage of projection graphs that are mapped to a single input graph. Again we see quite uniform results with the exceptions being input graphs 15, 83 and 100 which have significantly more graphs mapped to them for the same reason as those of the COX2 data set.

In Figure 13(c) we show the generated quasi-molecules overlayed on top of the input and projection sets. The generated quasi-molecules span the space of input molecules well and although there are some outliers, there are not a significant number of them. Finally, in Figure 13(d) we show the matches between generated quasi-molecules and the molecules of the projection set.

We now discuss the docking results which are given in Figure 15. It is immediately apparent

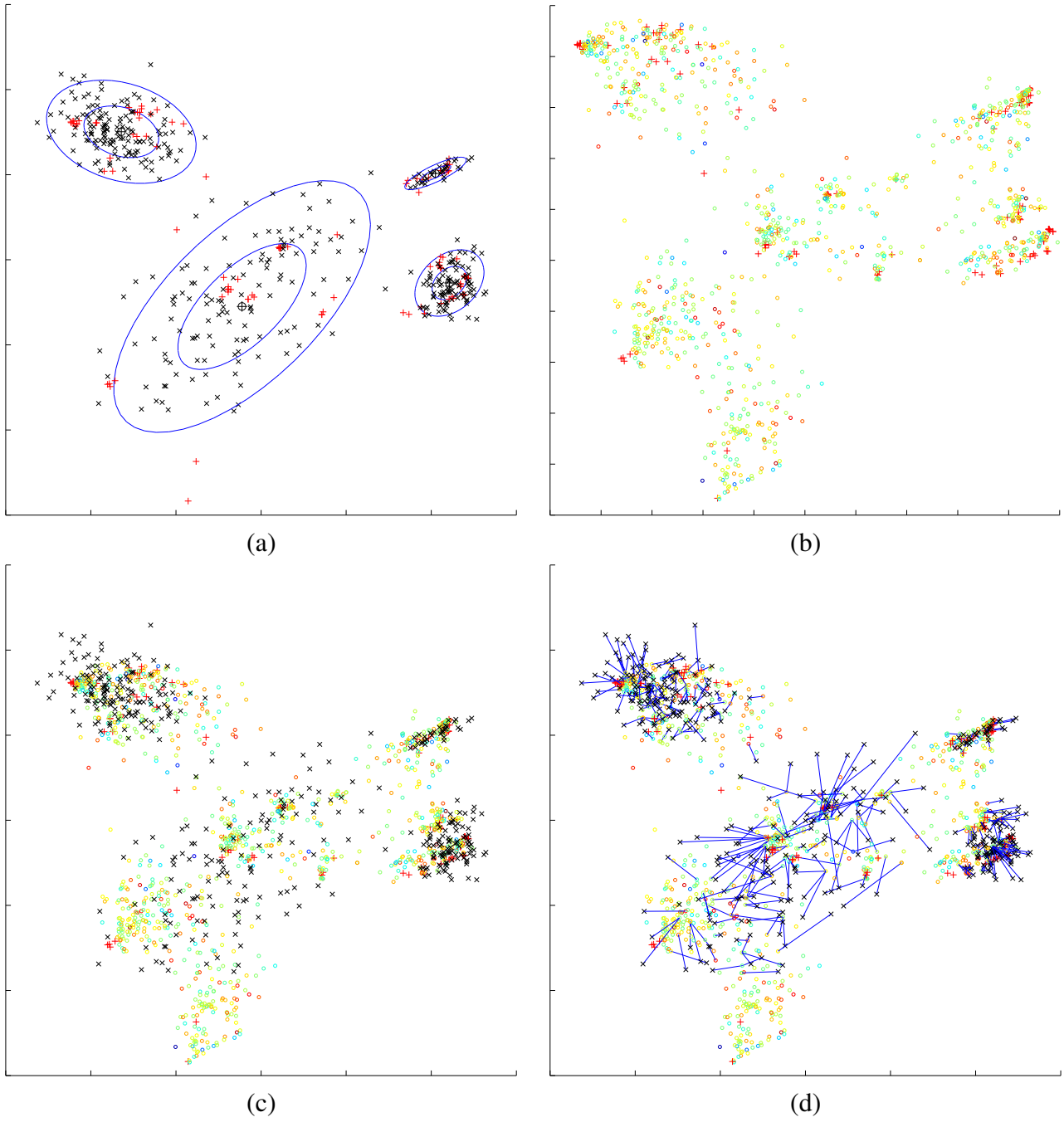


Figure 13: PCA projections of the input, projection and generated sets using the EGFR data set. Subfigure (a) shows in the input set (plus signs) and the generated quasi-molecules (crosses) in the PCA space. Ellipses are also drawn indicating the normal distributions chosen by the GMM. Subfigure (b) shows the input set (plus signs) and the projection set (colored circles). In (c), the generated quasi-molecules (crosses) are also projected into this space. Finally, in (d) the chosen mappings between projection molecules and generated quasi-molecules are shown. The color of each projection molecule marker indicates the docking value, blue is low affinity and red is high affinity.

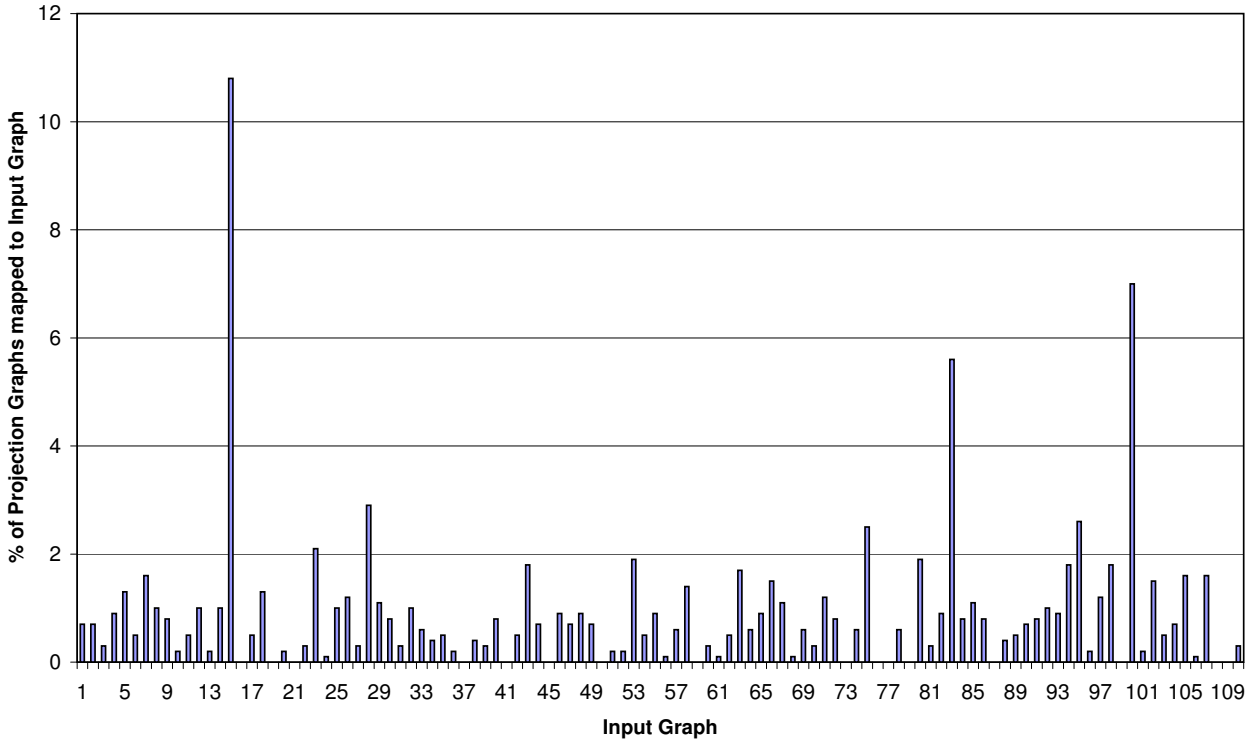


Figure 14: A histogram showing the percentage of projection graphs that align to a specific input graph in the hierarchical alignment step.

that the set of generated molecules docks well with this active site. As discussed earlier this is due to the more generic shape of this active site when compared to the COX2 active site. While the generated set appears to perform very well in comparison to the input set it is important to remember that the molecules of the input set have additional constraints imposed such as selectiveness of binding, druglikeness and valid synthesis routes. For example, ligands identified in the lead hopping process that bind with a great number of active sites must be discarded since they will interfere with the function of proteins other than the required target. We postulate that many of the generated molecules would display the same problem, however this is out of the scope of our research.

This view is reinforced when we consider the performance of the sample of decoys compared to the performance of the decoys in the COX2 active site. Recall from the COX2 docking results (Figure 9) that the performance of the decoys dropped off very quickly. In other words, we identified a few decoys that bound well but the majority performed poorly. Contrast this with the

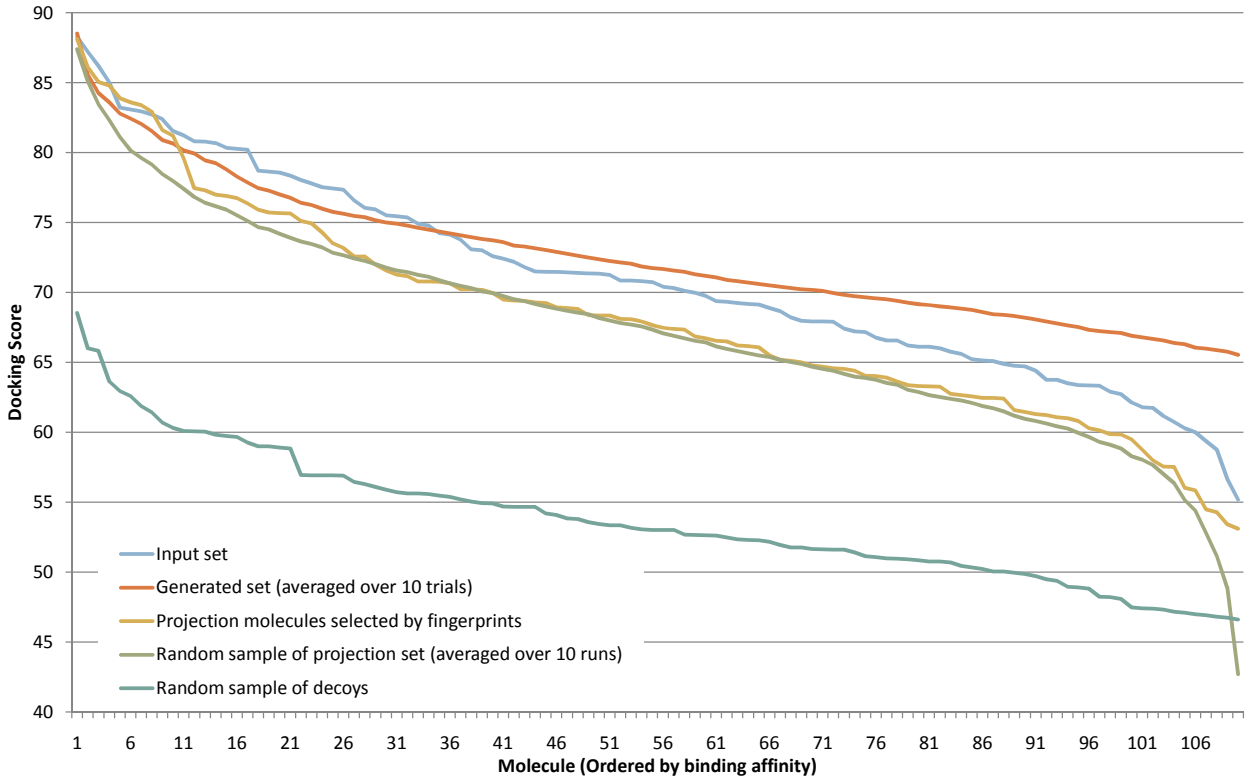


Figure 15: Docking scores for the EGFR data set. The first line represents the docking scores of the input molecules, and the second represents the docking scores of the generated molecules. Only the best 110 generated molecules are included. Next, the top 110 molecules selected by the fingerprint method are given. Finally, a random sample of 110 molecules from the projection and decoy sets are included. Within each set the molecules are sorted by docking score. Note that the docking score axis does not begin at 0.

performance of the decoys in the EGFR active site and we see that the decoys for the EGFR protein perform much better and do not tail off quickly in the same way as the COX2 decoys.[3] This suggests that it is much easier for a random molecule to bind to the EGFR active site than the COX2 active site. The same result is seen when we consider docking a random sample of the projection set.

The fingerprint approach produces acceptable results but the docking scores are lower than those of the generated set. Moreover, they tail off sharply towards the end of the docking scores.

The enrichment results are as follows for this data set: our approach achieved an enrichment rate of 1.85 while the fingerprint method managed to achieve a rate of 4.43. Again our method

[3]The sets of decoys are different for the two targets.

performs worse in this experiment for the reasons discussed as the end of Section 3.1. However, as the projection set is much larger for this data set the difference is exaggerated. This can be seen both in the docking results where our method outperforms the fingerprint method by a larger margin than in the COX2 data set, and in the enrichment results where the opposite is true. These results thus reinforce the conclusions drawn at the end of Section 3.1.

4 Conclusion

In this article we have proposed a method for constructing a generative model of chemical structure. The task of generating chemical structure is significantly harder than that of generating generic graphs due to the additional set of constraints governing what constitutes a valid chemical structure. The challenges this presents forces us to look at new ways of generating relational data through a projection step. In terms of generating drugs, we are only considering structural constraints in this work. The analysis of a synthesis route for generated molecules or the acceptability of molecules as drug candidates are not considered beyond the possibility of a simple filtering step for druglike properties.

The proposed method works by a three stage process. The first step involves the construction of a projection set of valid chemical structures near those of the input set in chemical space. The second step computes a global alignment of the input and projection sets that allows a vectorial description of each molecule to be produced. This makes the final step possible which consists of (a) constructing of the model, (b) sampling from the model and (c) mapping samples to the molecules of the projection set.

This method allows us to approximate sampling valid structures from the distribution of input molecules. The approximation arises from the step that maps true samples to molecules from the projection set. While this additional mapping step does introduce a small amount of error into our method we do not believe this to be significant and indeed we have shown that the molecules we generate are of high quality for the task they are designed for.

Furthermore, this error can be reduced by increasing the number of molecules in the projection set and therefore reducing the average distance of the mapping between a quasi-molecule and a molecule from the projection set. Another effect of this is that if these additional projection molecules are similar to those of the input set, then this allows the chemical space spanned by the input molecules to be explored in more detail. The hierarchical global alignment algorithm ensures that only a linear amount of work is required for each added projection molecule.

We have provided results for two different data sets. Using these data sets we have checked that (a) the molecules of the projection set are similar to those in the input set and are therefore suitable to be used as the range of our projection function, (b) the hierarchical alignment step results in ordered vectors that accurately describe the structures present in the molecules and (c) the molecules in the input set are well represented by a GMM.

Furthermore, the docking results show that if the input set is populated with molecules that are active against a target, then generated molecules are usually also successful at interacting with that target. This was true for active sites on both the COX2 and EGFR proteins when evaluated in a virtual docking environment. This result was reinforced when compared to the docking scores of a random sample of the projection set and a random sample of a decoy set. To provide some comparative results we also included the results of a fingerprint similarity method and showed that our method outperforms it at generating new molecules according to the docking scores.

To include an experiment that did not depend on the docking scores we considered the ability of the approach to generate known binders. These results were compared to those of the fingerprint method which outperformed our approach. However, as our method selects a more diverse and structurally interesting set of molecules than the fingerprint method, as evidenced by the docking results, it is less likely that it will generate known binders. It is the ability of our approach to select novel binders that hampers its chance of reproducing known binders.

Acknowledgement

The authors thank OpenEye Scientific Software (Santa Fe, NM) for the use of their tools under the academic licence agreement.

References

(1) Oprea, T. I. Chemical space navigation in lead discovery. *Curr. Opin. Chem. Biol.* **2002**, *6*, 384–389.

(2) Lipinski, C.; Hopkins, A. Navigating chemical space for biology and medicine. *Nature* **2004**, *432*, 855–861.

(3) Johnson, M. A.; Maggiora, G. M. *Concepts and applications of molecular similarity*; Wiley, New York, 1990.

(4) Patterson, D. E.; Cramer, R. D.; Ferguson, A. M.; Clark, R. D.; Weinberger, L. E. Neighborhood behavior: A useful concept for validation of "molecular diversity" descriptors. *J. Med. Chem.* **1996**, *39*, 3049–3059.

(5) Schneider, G.; Fechner, U. Computer-based de novo design of drug-like molecules. *Nat. Rev. Drug Discovery* **2005**, *4*, 649–663.

(6) Hertzberg, R.; Pope, A. High-throughput screening: new technology for the 21st century. *Curr. Opin. Chem. Biol.* **2000**, *4*, 445–451.

(7) Rees, D. C.; Congreve, M.; Murray, C. W.; Carr, R. Fragment-based lead discovery. *Nat. Rev. Drug Discovery* **2004**, *3*, 660–672.

(8) Miller, M. Chemical database techniques in drug discovery. *Nat. Rev. Drug Discovery* **2002**, *1*, 220–227.

(9) Ajay, A.; Walters, W. P.; Murcko, M. A. Can we learn to distinguish between "drug-like" and "nondrug-like" molecules? *J. Med. Chem.* **1998**, *41*, 3314–3324.

(10) Munk, M. E. Computer-based structure determination: then and now. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 997–1009.

(11) Funatsu, K.; Miyabayashi, N.; Sasaki, S. Further development of structure generation in the automated structure elucidation system CHEMICS. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 18–28.

(12) Christie, B. D.; Munk, M. E. Structure generation by reduction: a new strategy for computer-assisted structure elucidation. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 87–93.

(13) Rotstein, S.; Murcko, M. GenStar: a method for de novo drug design. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 23–43.

(14) Rotstein, S.; Murcko, M. GroupBuild: a fragment-based method for de novo drug design. *J. Med. Chem.* **1993**, *36*, 1700–1710.

(15) Wang, R.; Gao, Y.; Lai, L. LigBuilder: a multi-purpose program for structure-based drug design. *J. Mol. Model.* **2000**, *6*, 498–516.

(16) Moon, J.; Howe, W. Computer design of bioactive molecules: a method for receptor-based de novo ligand design. *Proteins: Struct. Funct. Genet.* **1991**, *11*, 314–328.

(17) Roe, D.; Kuntz, I. BUILDER v. 2: Improving the chemistry of a de novo design strategy. *J. Comput.-Aided Mol. Des.* **1995**, *9*, 269–282.

(18) Ho, C.; Marshall, G. DBMAKER: A set of programs to generate three-dimensional databases based upon user-specified criteria. *J. Comput.-Aided Mol. Des.* **1995**, *9*, 65–86.

(19) Cosgrove, D.; Kenny, P. BOOMSLANG: A program for combinatorial structure generation. *J. Mol. Graphics* **1996**, *14*, 1–5.

(20) Clark, D.; Firth, M.; Murray, C. MOLMAKER: de novo generation of 3D databases for use in drug design. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 137–145.

(21) Makino, S.; Ewing, T.; Kuntz, I. DREAM++: flexible docking program for virtual combinatorial libraries. *J. Comput.-Aided Mol. Des.* **1999**, *13*, 513–532.

(22) Schneider, G.; Lee, M.; Stahl, M.; Schneider, P. De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 487–494.

(23) Vinkers, H.; de Jonge, M.; Daeyaert, F.; Heeres, J.; Koymans, L.; van Lenthe, J.; Lewi, P.; Timmerman, H.; Van Aken, K.; Janssen, P. Synopsis: synthesize and optimize system in silico. *J. Med. Chem.* **2003**, *46*, 2765–2773.

(24) Boda, K.; Johnson, A. P. Molecular complexity analysis of de novo designed ligands. *J. Med. Chem* **2006**, *49*, 5869–5879.

(25) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.* **1997**, *23*, 3–25.

(26) Kutchukian, P.; Lou, D.; Shakhnovich, E. FOG: Fragment Optimized Growth Algorithm for the de Novo Generation of Molecules Occupying Druglike Chemical Space. *J. Chem. Inf. Model.* **2009**, *49*, 1630–1642.

(27) Balaban, A. T. Applications of Graph Theory in Chemistry. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 334–343.

(28) Jiang, X.; Münger, A.; Bunke, H. On Median Graphs: Properties, Algorithms and Applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1144–1151.

(29) Ferrer, M.; Valveny, E.; Serratosa, F. Median graphs: A genetic approach based on new theoretical properties. *Pattern Recognit.* **2009**, *42*, 2003–2012.

(30) Luo, B.; Wilson, R. C.; Hancock, E. R. A Linear Generative Model for Graph Structure. *Lect. Notes Comput. Sci.* **2005**, *3434*, 54–62.

(31) Xiao, B.; Hancock, E. R. A Spectral Generative Model for Graph Structure. *Lect. Notes Comput. Sci.* **2006**, *4109*, 173–181.

(32) White, D.; Wilson, R. C. Spectral Generative Models for Graphs. In *Proceedings of the 14th International Conference on Image Analysis and Processing (ICIAP 2007)*, Modena, Italy, 2007; IEEE Computer Society; pp 35–42.

(33) White, D.; Wilson, R. C. Parts Based Generative Models for Graphs. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008)*, Tampa, Florida, USA, 2008; IEEE Computer Society; pp 1–4.

(34) Torsello, A.; Dowe, D. L. Learning a Generative Model for Structural Representations. *Lecture Notes in Artifical Intelligence* **2008**, *5360*, 573–583.

(35) Gold, S.; Rangarajan, A. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **1996**, *18*, 377–388.

(36) Caelli, T.; Kosinov, S. An Eigenspace Projection Clustering Method for Inexact Graph Matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 515–519.

(37) Shokoufandeh, A.; Dickinson, S. J.; Siddiqi, K.; Zucker, S. W. Indexing using a Spectral Coding of Topological Structure. In *1999 Conference on Computer Vision and Pattern Recognition (CVPR 99)*, Ft. Collins, CO, USA, 1999; IEEE Computer Society; pp 2491–2497.

(38) Wilson, R. C.; Hancock, E. R.; Luo, B. Pattern Vectors from Algebraic Graph Theory. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1112–1124.

(39) Riesen, K.; Neuhaus, M.; Bunke, H. Graph Embedding in Vector Spaces by Means Of Prototype Selection. *Lect. Notes Comput. Sci.* **2007**, *4538*, 383–393.

(40) White, D.; Wilson, R. C. Mixing Spectral Representations of Graphs. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006)*, Hong Kong, China, 2006; IEEE Computer Society; pp 140–144.

(41) Porquet, A.; Duval, J. F. L.; Buffle, J. Random Computer Generation of 3D Molecular Structures: Theoretical and Statistical Analysis. *Macromol. Theory Simul.* **2006**, *15*, 147–162.

(42) Figueiredo, M.; Jain, A. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 381–396.

(43) Sanfeliu, A.; Fu, K. S. A Distance Measure Between Attributed Relational Graphs for Pattern Recognit. *IEEE Transactions on Systems, Man and Cybernetics* **1983**, *13*, 353–362.

(44) Huang, N.; Shoichet, B. K.; Irwin, J. J. Benchmarking sets for molecular docking. *J. Med. Chem.* **2006**, *49*, 6789–6801.

(45) Rarey, M.; Kramer, B.; Lengauer, T.; Klebe, G. A fast flexible docking method using an incremental construction algorithm. *J. Mol. Biol.* **1996**, *261*, 470–489.

(46) Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; Ferrin, T. E. A geometric approach to macromolecule-ligand interactions. *J. Mol. Biol.* **1982**, *161*, 269–288.

(47) Venkatachalam, C. M.; Jiang, X.; Oldfield, T.; Waldman, M. LigandFit: a novel method for the shape-directed rapid docking of ligands to protein active sites. *J. Mol. Graphics Modell.* **2003**, *21*, 289–307.

(48) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Development and validation of a genetic algorithm for flexible docking. *J. Mol. Biol.* **1997**, *267*, 727–748.

(49) *FRED (Fast Rigid Exhaustive Docking)*, version 2.2.3; OpenEye Scientific Software: Santa Fe, NM, 2009. `http://www.eyesopen.com/products/applications/fred.html` (accessed May 27, 2010).

(50) Shi, J.; Malik, J. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.

(51) *Chomp*, version 1.1.1; OpenEye Scientific Software: Santa Fe, NM, 2009. http://www.eyesopen.com/docs/brood/1.1.2/html/manual/node30.html (accessed May 27, 2010).

(52) *Omega*, version 2.3.0; OpenEye Scientific Software: Santa Fe, NM, 2009. http://www.eyesopen.com/products/applications/omega.html (accessed May 27, 2010).

(53) Warren, G. L.; Andrews, C. W.; Capelli, A. M.; Clarke, B.; LaLonde, J.; Lambert, M. H.; Lindvall, M.; Nevins, N.; Semus, S. F.; Senger, S.; Tedesco, G.; Wall, I. D.; Woolven, J. M.; Peishoff, C. E.; Head, M. S. A critical assessment of docking programs and scoring functions. *J. Med. Chem* **2006**, *49*, 5912–5931.

(54) *The Open Babel Package*, version 2.2.3. http://openbabel.sourceforge.net (accessed May 27, 2010).

Graphical TOC Entry