Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Multi-Alignment of Molecules using a semi-flexible Core Structure

Submitted by

Christoph Noack, Niklas Paulicks, Marius Rüve
Malte Schokolowski, Mike Trzaska, Jule Würfel

May 12, 2024

Abstract

Molecular alignment plays a crucial role in drug design, particularly in scenarios where information about the structure of a receptor protein is unavailable. In the absence of three-dimensional data for a ligand-receptor complex, the superposition of bioactive molecules serves as a valuable approach for predicting pharmacophores, binding modes and the three-dimensional structure of binding pockets. Frequently, bioactive molecules within one compound series share a common core structure, which can serve as a foundation for multiple molecule superposition. Despite the significance of this strategy, to our knowledge, there exists no implementation utilizing it.
In response to this need, we have developed CoAler (**Co**re **Al**ign**er**), a tool capable of efficiently calculating multi-alignments for dozens of molecules from scratch within minutes on a standard desktop computer.

Contents

Chapter 1.

1 Introduction

When only a set of bioactive ligands for a receptor is known, these molecules can be superimposed to estimate their binding modes. This can aid in deriving pharmacophores and discovering new drug candidates. Typically, these molecules exhibit certain structural similarities when binding to the same binding site. Leveraging this fact, one can assume that a common core structure of the molecules resides roughly at the same location within the binding pocket and thus can be used as a reference point when superimposing the molecules.

In this report, we present the software CoAler (**Co**re **Al**ign**er**) that computes superpositions of molecule sets that share a common core structure. CoAler determines the core structure dynamically, samples the conformational space of the input molecules, and computes an alignment based on pairwise comparisons of the molecule conformations. Several heuristic optimization approaches were used to enable CoAler to superimpose diverse molecule ensembles. The common core structure of the whole ensemble is hereby used as a starting point for the alignment, while in the course of the optimization, CoAler searches for bigger core structures that are shared by subsets of the ensemble.

The tool was tested on a set of close-to-real-world ligands that were gained from the data produced by aligning sequentially close protein binding pockets in [1]. Overall, CoAler shows good performance for a sizable amount of the molecule ensembles it was tested on, where it produced alignments that often diverged only 2-3Å on average from the benchmark ligands. For singular ensembles that for instance did not exhibit a common core structure that was easy to compute, the devised strategy seems not optimal though.

Chapter 2.

2 Theoretical Foundations

2.1. Tanimoto Coefficient

The Tanimoto coefficient, also known as the Jaccard index, serves as a measure in cheminformatics for evaluating chemical or molecular similarity through fingerprints. These fingerprints are binary or count-based vectors that encapsulate the presence, absence, or count of specific structural motifs within molecules. The calculation of the Tanimoto similarity involves dividing the number of features shared by two molecules by the total number of unique features in either molecule, formulated as 2.1, where A and B represent the feature sets of the molecules in comparison. This coefficient, which varies between 0 (no similarity) and 1 (maximum similarity), is widely applied in the realms of drug discovery and chemical database searches to pinpoint molecules with analogous characteristics, infer biological activities, and organize compounds into clusters or categories based on their structural attributes. [2, 3]

$$T_{sim}(A, B) = \frac{\mid A \cap B \mid}{\mid A \cup B \mid} = \frac{\mid A \cap B \mid}{\mid A \mid + \mid B \mid - \mid A \cap B \mid} \tag{2.1}$$

$$T_{dist}(A, B) = 1 - T_{sim}(A, B) \tag{2.2}$$

2.2. Shape-Based Tanimoto

The shape-based similarity assessment leverages three-dimensional structural information of molecules, comparing their volumetric overlap to quantify similarity. In this context, the Tanimoto coefficient is calculated with the same equation 2.1, by dividing the volume of intersection between two molecular shapes by the volume of their union, offering a metric that ranges from 0 (no overlap) to 1 (complete overlap). Such an approach is particularly valuable in drug design, where the spatial arrangement of molecules can significantly influence their biological activity by affecting the fit within a target's binding site. [4]

2.3. Rigid Body Alignment

In Euclidean space, geometric objects like molecules can be modeled as a set of points. Furthermore, a mapping from one such set to another subset of the Euclidean space, while preserving the Euclidean distances between points, is called a rigid body alignment. This means such a mapping can be composed of rotations and translations [5].

The following sections describe the Root Mean Squared Distance (RMSD) and the Kabsch algorithm, which are used for benchmarking in Chapter 4.

2.3.1. Root Mean Squared Distance (RMSD)

For two molecules, a typically used quantitative measure of similarity is the RMSD [6]. Equation 2.3 describes the calculation of the RMSD for two sets of points, which represent the atom coordinates of the two molecules, where $X = \{x_1, ..., x_n\}$ and $Y = \{y_1, ..., y_n\}$ [6, 7]. After an alignment of the molecules, this equation changes to $RMSD(T(X), Y)$, where $T(X)$ denotes the rigid transformation of the coordinates of molecule X [7]. The unit of measurement for the RMSD is the angstrom (Å) [6].

$$RMSD(X, Y) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} ||x_i - y_i||^2} \tag{2.3}$$

2.3.2. Kabsch Algorithm

The Kabsch algorithm computes a rigid body transformation for two sets of points, representing molecules in our case, in 3D Euclidean space, consisting of a 3×3 rotation matrix and a translation vector, and returns the RMSD of those sets [7]. Point sets in the 3D Euclidean space are represented as $N \times 3$ matrices, where N is the number of points and each row contains the coordinates of a single point. Prior to computing the optimal rotation matrix R, the points of the input sets X and Y are translated, such that the centroid of each set coincides with the origin of the coordinate system [8]. Algorithm 2 explicates this process for one input set. In the next step, the covariance matrix C of the input sets X and Y is computed, followed by the derivation of the rotation matrix R using singular value decomposition (SVD) of the covariance matrix C [7, 8]. Algorithm 3 outlines the process of calculating the rotation matrix. The overall RMSD calculation using the Kabsch algorithm is presented by Algorithm 1 [7, 8].

Algorithm 1 Kabsch algorithm for RMSD calculation [7, 8].

Input: $N \times 3$ matrices X and Y where N is the number of points
Output : Root-mean-square distance

procedure KabschRMSD
    $X \leftarrow KabschTranslationBasedOnCentroid(X)$
    $Y \leftarrow KabschTranslationBasedOnCentroid(Y)$
    $R \leftarrow KabschCalculateRotationMatrix(X, Y)$
    return $RMSD(X * R, Y)$ ▷ Eq. 2.3

Algorithm 2 Kabsch algorithm: Translation based on centroid [8].

Input: $N \times 3$ matrix W
Output : Translated matrix W

procedure KabschTranslation
    $points \leftarrow$ interpret rows of W as a point
    $centroid \leftarrow CalculateCentroid(points)$
    for $p_i \in points$ **do**
        $p_i \leftarrow p_i - centroid$
    $W \leftarrow$ save $points$ in rows of W
    return W

Algorithm 3 Kabsch algorithm: Calculation of the rotation matrix [7, 8].

Input: $N \times 3$ matrices X and Y where N is the number of points
Output : The optimal rotation matrix R

procedure KabschCalculateRotationMatrix
  $C \leftarrow X^T * Y$ ▷ Calculation of the covariance matrix
  $U * \Sigma * V^T \leftarrow$ singular value decomposition of C
  $\{s_1, .., s_d\} \leftarrow$ diagonal of Σ
  if $det(U * V) < 0$ **then**
    $s_d \leftarrow -1 * s_d$
  $\Sigma' \leftarrow$ diagonal $\{s_1, .., s_d\}$
  $R \leftarrow V * \Sigma' * U^T$
  return R

2.4. Core Detection

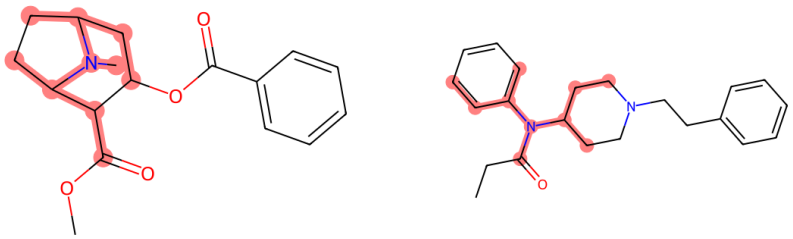
As described in the introduction 1, we decided to detect the common structure (core) of a given set of molecules. For the detection, we used Maximum Common Substructure (MCS) and Bemis-Murcko Scaffolds [9], which are described in the following.

2.4.1. Maximum Common Substructure

For two molecules, the largest substructure that appears in both molecules, is the MCS [10]. An important component of structurally related drugs is likely to be the largest common substructure, which is why the MCS serves as a metric for both chemical similarity searching and activity predictions [10]. In the following, a formal introduction for the MCS of two graphs is provided.

Isomorphic Pair of Graphs Two graphs are isomorphic if there exists a bijective mapping from the vertices in the one graph to the other graph while preserving adjacency [11]. Equation 2.4 provides a formal definition of isomorphic graphs, where $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ represent two graphs with node sets V_i and edge sets E_i. Additionally, $f : V_1 \rightarrow V_2$ denotes a bijective mapping [11].

$$\forall a, b \in V_1, (a, b) \in E_1 \Leftrightarrow (f(a), f(b)) \in E_2 \tag{2.4}$$



(a) (b)

Figure 2.1.: MCS of Cocaine and Fentanyl using RDKit.

Induced Subgraph For a graph $G_1 = (V_1, E_1)$, $G_s = (V_s, E_s)$ is an induced subgraph, if $V_s \subseteq V_1$ and $E_s = \{(a, b) \in E_1 | a, b \in V_s\} \subseteq E_1 \cap (V_s \times V_s)$ [10].

Common Induced Subgraph If G_{s1} is an induced subgraph of G_1, and G_{s2} is an induced subgraph of G_2, such that G_{s1} and G_{s2} are isomorphic, then G_{s1} and G_{s2} are common induced subgraphs of G_1 and G_2 [10].

MCS The MCS between two graphs G_1 and G_2 is the common induced subgraph $G_m = (V_m, E_m)$, with $|V_m|$ maximal [10]. However, one can decide between connected and non-connected MCSs. Since vertices and edges are labeled in the context of molecules to represent the different elements and bond types, it is reasonable to determine whether the labels need to be exactly the same or not. An additional parameter for the MCS of molecules is the stereochemistry of molecules, where one needs to decide whether to take it into account or not [12]. Further parameters for the MCS calculation of molecules can be found in RDKit. Figure 2.1 presents an example of the MCS of Cocaine and Fentanyl, where the aromatic bonds are detected as similar.

2.4.2. Murcko Scaffolds

Introduced by Bemis and Murcko in 1996, the Bemis-Murcko scaffold has emerged as a pivotal concept in cheminformatics, playing a crucial role in drug discovery and medicinal chemistry. This concept, also referred to as the

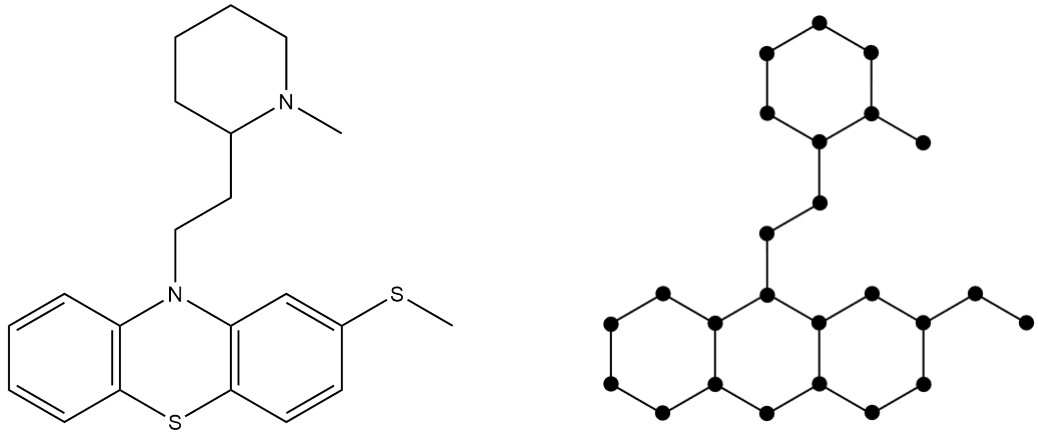


Figure 2.2.: The Thioridazine on the left in comparison to its framework on the right. Each molecule has an underlying framework that can be used for our core calculation.

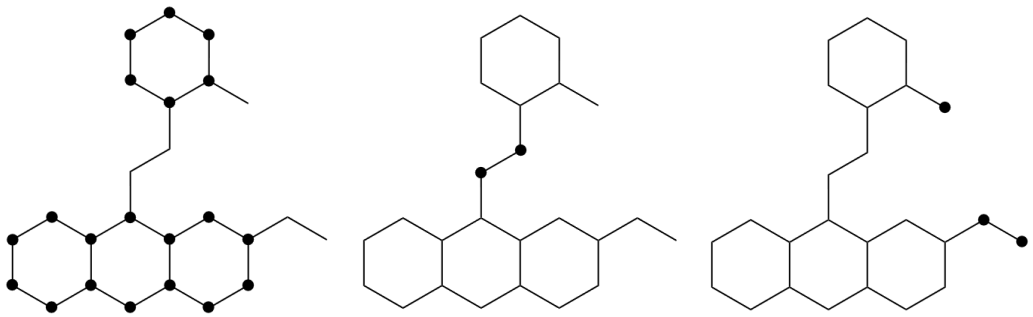
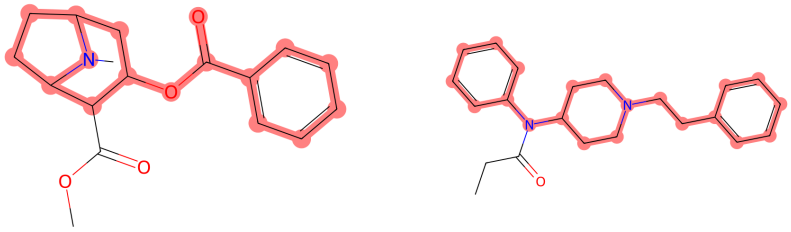


Figure 2.3.: The framework of Thioridazine can be devided into its ring system, side chains and linkers. The ring system, shown on the left, contains all cyclic parts of the framework. The linkers, shown in the middle contain all parts of the framework, that connect the cyclic parts directly or are attached to them. Side chains are all parts that are none of the two and are shown on the right

framework, is shown in figure 2.2 and involves isolating the core structure of a molecule by stripping away all terminal groups, preserving only the interconnected ring systems and critical linkers that join these rings. This partitioning is depicted in figure 2.3. In our tool, the user can choose between two types of core structures, the maximum common substructure and the Bemis-Murcko scaffold [9].

Figure 2.4 presents the Murcko Scaffolds of Cocaine and Fentanyl. The signif-



(a) (b)

Figure 2.4.: Murcko Scaffold of Cocaine and Fentanyl using RDKit.

icance of the Murcko scaffold lies in its capacity to illuminate the molecular backbone, offering insights into the effects of scaffold alterations on biological activity and drug properties. As an alternative to the MCS in core detection, the Murcko scaffold simplifies molecules to their essential structure, serving as a foundation for further analysis. Ring systems are defined as cycles within the molecule's graph representation, with shared edges indicating a bond. Linker atoms are those directly connecting two ring systems, while any atoms not part of a ring or linker are considered side chain atoms. The framework encompasses the collective ring systems and linkers within a molecule [13].

Chapter 3.

3 Methods

This section offers an in-depth explanation of the algorithms that power CoAler. Initially, it outlines the overall process flow. Following that, it discusses the selection and creation of methods for each step. These steps encompass determining the shared core, aligning these cores, sampling conformations, and the process of alignment. An alignment—conformations that maintain consistent absolute coordinates for each ligand in the collection—is subsequently termed an (alignment-) assembly.

3.1. General Workflow

The CoAler multialignment algorithm draws partial inspiration from the work presented in MolAlign [14]. It achieves ligand alignment through the rigid alignment of conformation ensembles. Before the conformation sampling, a common core is identified. This core can either be the MCS of the molecules or the shared Murcko-Bemis Scaffold [9]. Once the core is established, molecule conformations are sampled with the core's absolute coordinates kept consistent across all molecules, facilitated by the coordinate restriction feature of the embedding function. Following this, the algorithm computes the overlap of all possible conformation combinations for each pair of molecules. These pairwise alignment scores are utilized to create promising initial assemblies. These assemblies are then refined by either switching the assigned conformation of the ligands to another previously sampled one or by generating new conformations when no suitable replacements are available. Once all initial assemblies are

refined, the highest-scoring assembly is further optimized and presented as the result of the multialignment.

3.2. Core Calculation

For the initial alignment of ligands using a common core structure, this structure must be present in each molecule. The structure itself does not have to be defined as a real molecule, but as a framework where atoms are depicted as vertices and bonds as edges between those vertices. Depending on preference, one vertex can describe just one or different elements or hybridization of an element, just as one edge can depict one or different types of atom bonds. Depending on the user's choice, the CoAler tool can use one of two different core structure types, both are described in section 2.4. In the calculation of the MCS, the multiple MCS calculator of RDKit is employed. The calculated MCS accommodates various stereochemistry and hybridization types present in the input molecules. To generate the Bemis-Murcko scaffold, the result of the MCS calculation serves as the initial basis. The applied algorithm proceeds to identify all side chains, stripping them away from the main structure. Should the MCS framework lack any ring structures, it implies that all vertices belong to side chains, resulting in an empty Bemis-Murcko scaffold.

3.3. Conformation Generation

The conformation generation workflow is depicted in Fig. 3.1. The initial step involves generating conformations for all input molecules using ETKDGv3 [15,16], ensuring they meet energetic feasibility criteria. The cores of the resulting conformations are subsequently aligned to a sampled reference conformation of the core. The alignment consists of a mapping of the core atoms of the conformations to the reference and a subsequent Kabsch algorithm [17] distance minimization. An example of the resulting conformations in 3D-space is depicted in figure 3.2. All molecule embeddings are performed using a randomized embedding procedure. However we opted to use a set seed in order to retain reproducibility.

The potential symmetry of the core structure is taken into consideration. A symmetric core will have several possible atom mappings depending on the symmetry. Those different mappings are recognized during the embedding and core overlaying steps. If a core features many symmetry axes or the core matches

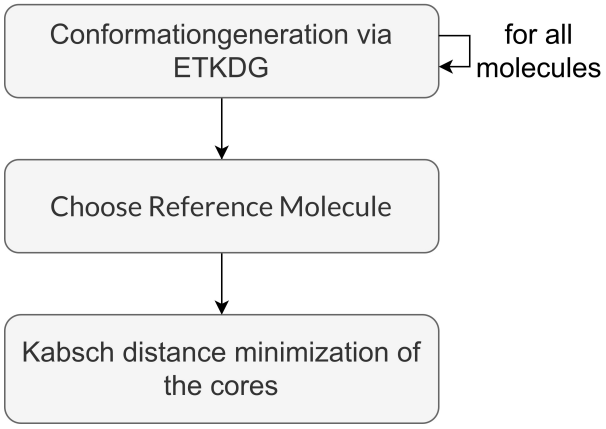


Figure 3.1.: Workflow Conformation Generation.

multiple regions in the molecule, the conformation sampling becomes more complex since many more embeddings are required to sample the conformation space sufficiently. This effect can be observed in Fig. 3.2. In total there are six possible orientations of the molecule due to the symmetry of the benzene core. If a fixed number of conformations are to be embedded for this molecule, the number of conformations has to be divided among the different orientations. For a more extensive conformation sampling, a set number of conformations can be embedded per orientation regardless of the core symmetry. However, this risks a combinatorial explosion in later steps of the multialignment steps (the pairwise alignment, see 3.4.2). This is especially true for molecules, where the core matches multiple substructures. Therefore, both approaches are offered at the user's choice. Either the given number of conformations is embedded for each core match (this includes different positions and rotations) or the embeddings are divided among the matches. In any case, some lower bounds are in place to ensure a feasible CoAler run.

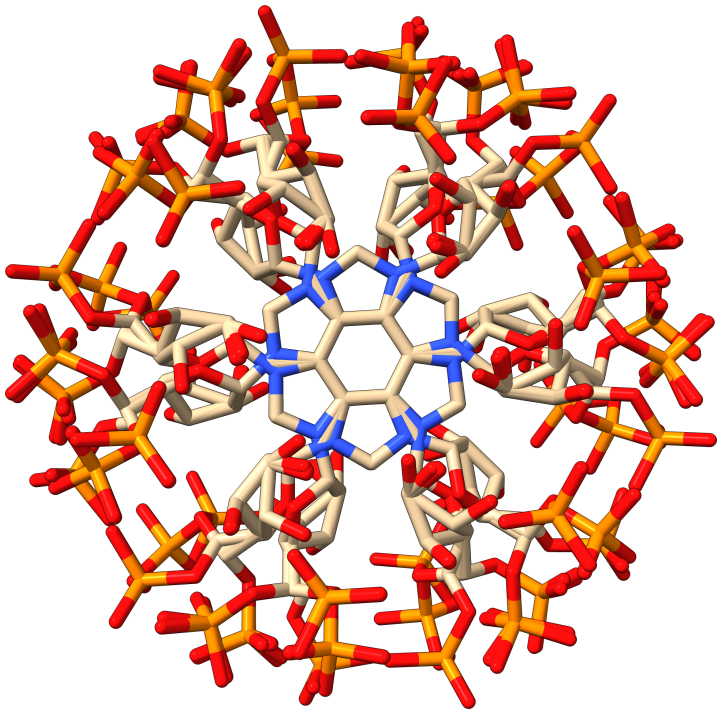


Figure 3.2.: Different conformations of an alpha-ribazole-phosphate derivative. The conformations are generated using the ETKDG version 3 method. All common core structures of the molecules were aligned using distance minimization of mapped atoms between a reference molecule and all conformations. The illustration was generated with Chimera [18].

3.4. Multiple Alignment

The multialignment consists of the generation and optimization of alignment assemblies. An alignment assembly is an assignment of a specific conformation to each molecule (also referred to as ligand here) in the input set. Since the core structures of all generated conformations have (nearly) the same absolute coordinates, every assembly automatically represents a possible, not necessarily optimal, solution to the alignment problem. Fig. 3.3 depicts the general workflow of the multiple alignment process. First, the assemblies are generated. Then, a first optimization routine is applied, here referred to as the "rough" optimization. The assembly with the best score is then "fine-tuned". Occasionally, when certain ligands do not fit well within the assembly post-optimization, we resort to a brute-force approach to find better-fitting conformations for those ligands.

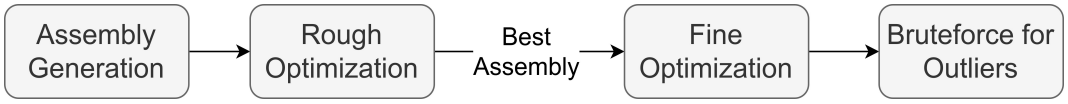


Figure 3.3.: Overall Workflow Assembly Optimization. First, the assemblies are generated. Then, all the generated assemblies are"coarsely"optimized. The best assembly from this step is identified via the assembly score and optimized again with the target to exactly superposition atoms wherever possible.

The number of assemblies to be generated and optimized can be set by the user. It is recommended to choose a multiple of the number of available cores to use the computational power most effectively since each thread is assigned an assembly to optimize.

3.4.1. Scoring

Pairwise-Alignment Scoring In evaluating the quality of a molecular alignment, it is essential to employ a scoring function. The Tanimoto Shape Similarity 2.2 serves as the criterion for this purpose. Essentially, this approach posits that the more extensive the overlap between the shapes of two molecules, the higher the Tanimoto similarity. Consequently, pairwise alignments exhibiting greater shape congruence are awarded higher scores.

$$\text{score}_{pair}(\text{mol}_A, \text{mol}_B) = \text{sim\_tanimoto}(\text{mol}_A, \text{mol}_B) \tag{3.1}$$

Assembly Scoring To evaluate the score of an assembly, which involves the alignment of all molecules within, we compute the mean pairwise score across all included entities using Eq. 3.1. It's important to note that an assembly does not necessarily need to include a conformation for each molecule. The reason for this is explained in section 3.4.2. Because of that, the summed score of each pairwise alignment is always divided by the number of molecules in the input file and not by the number of conformations in the assembly. Otherwise, the scoring would not penalize missing ligands in the assembly. The resulting equation to calculate the score for an assembly is shown in Eq. 3.2.

	Ligand A	**Ligand B**	**Score**
1	Conformation 13	Conformation 5	0.86
2	Conformation 4	Conformation 8	0.75
..	..	..	..
k	Conformation 13	Conformation 24	0.71

Table 3.1.: Example of a pose register for the k best conformation pairs of two ligands A and B. A pose register keeps track of the conformation pairs with the highest pairwise score.

$$\text{score}_{assembly}(\{\text{mol}_1, \ldots, \text{mol}_n\}) = \frac{2}{n(n-1)} \sum_{i<j}^{n} \text{score}_{pair}(i, j) \tag{3.2}$$

3.4.2. Starting Assembly Generation

The generation of the starting assemblies is based on the generation and evaluation of so-called"pose-registers", as described in MolAlign [14]. A pose register holds the k conformation pairs of two ligands with the highest shape congruence. For every pair of ligands, a pose register is generated. Before this, the pairwise shape overlap of all possible conformation pairs has to be calculated. This calculation is highly resource-demanding with a complexity of $\mathcal{O}(n^2 \cdot m^2)$ for n ligands with m conformations each. The result of the pairwise overlap calculation is then used to determine the rankings in the pose registers. Table 3.1 depicts an exemplary pose register.

The starting assembly generation workflow is depicted in Fig. 3.4. Every assembly is built around a seed conformer. This seed conformer determines the approximate shape of the assembly: for all other ligands, the conformation with the highest similarity to the seed conformer is looked up in the corresponding pose register. For example, consider conformation 13 of ligand A as the seed conformation for a starting assembly. The lookup in the pose register in Table 3.1 will return conformation 5 of ligand B as the best match since it has the highest overlap with the seed conformation. Therefore, conformation 5 would be assigned to ligand B in the assembly. This lookup is performed for all other ligands. In some cases, the seed conformation may not be among the best k conformation pairs, i.e., it cannot be found in the pose register. This is referred to as a"missing ligand"and the missing ligand counter of the assembly is increased by one, no conformation is assigned to the missing ligand. The

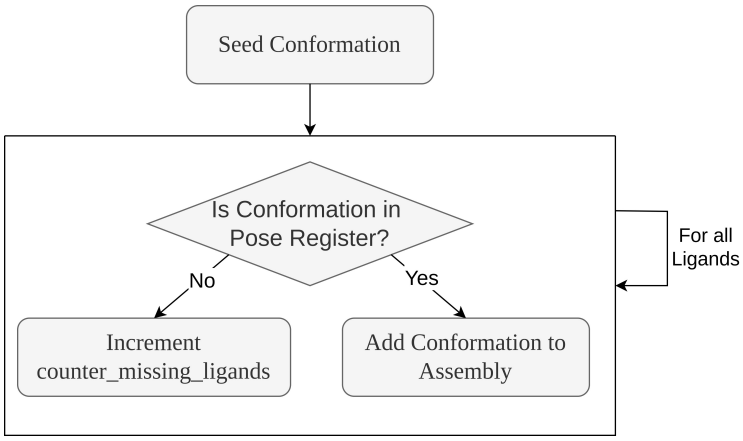


Figure 3.4.: Workflow of the Assembly Generation. Every conformation of every ligand is used as the seed conformation once. The seed conformation determines the geometry, which all other conformations in the assembly will resemble as close as possible. Given a seed conformation, for all ligands, a lookup is made in the corresponding pose register to check whether the conformation is among the best pairwise conformation overlaps of the two molecules. If so, the conformation of the ligand is added to the assembly. If no matching conformation can be found, the ligand is marked as"missing"in the assembly.

effect of this is further elaborated in 3.4.3.

During the generation of the starting assemblies, a priority queue keeps track of the best assemblies. The number of assemblies to be kept and optimized thereafter can be set by the user. The assessment of the assemblies is divided into two layers. The missing ligand count has precedence over the assembly score (see 3.2). Assemblies with fewer missing ligands are favored.

3.4.3. Assembly Optimization

The assemblies are optimized by iteratively changing the assigned conformation for the "worst ligand"in the assembly. The worst ligand is determined using the so-called score deficit, as described in MolAlign [14]. The score deficit is a measure of the quality of a conformation in an assembly compared to the other available (i.e., previously generated) conformations of the molecule.

The assembly optimization consists of an iteration (depicted in Fig. 3.5) where

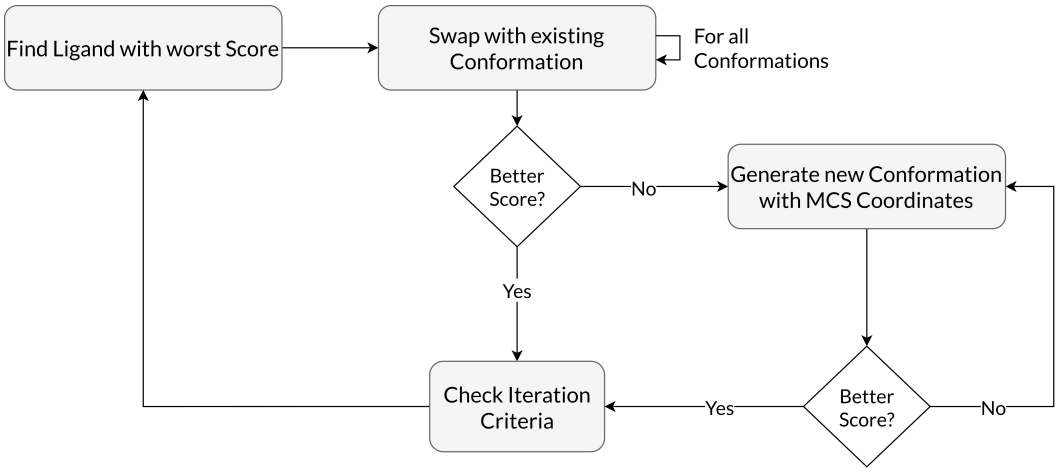


Figure 3.5.: Workflow Rough Assembly Optimization.

in each step an attempt is made to exchange the conformation of the worst ligand with another previously generated conformation of this ligand. If this does not yield a better assembly score (i.e., no better conformation can be found), new conformations are generated for the ligand. The conformation swapping and the generation of new conformations are elaborated in detail in the following.

Conformation Swapping The conformation of the worst ligand is swapped with all other so far generated conformations of this ligand. The score of the assembly resulting from each swap is calculated. The conformation yielding the best new assembly is kept as the new conformation for the ligand. If no improving conformation was found, generating a new conformation is attempted.

Conformation Generation Since generating a random conformation is unlikely to yield a conformation that improves the assembly, reference points are required. Hence, each ligand in the assembly (except for the determined worst ligand) is used as a"target"for the conformation generation. The reference points from the targets are determined by looking up the MCS (which is pre-calculated before the multialignment) between the worst ligand and the target. The MCS is restricted to also contain the calculated common core. The coordinates of the target atoms contained in the MCS are mapped onto the worst ligand and used as fixed constraints for the conformation generation. Thus, each generated conformation will have the same atom coordinates as the

target ligand for all the atoms contained in the MCS. After the conformation generation has been performed for all the targets, the most fitting generated conformation is determined by calculating the corresponding assembly scores. Naturally, the quality of the overlap of the newly generated conformation depends on the size of the calculated MCS: The bigger the MCS, the more atoms will have set coordinates and the fewer atoms will deviate from the target's conformation. Therefore, an MCS search strategy is utilized that attempts to maximize the MCS size. This is achieved by among others disregarding atom and bond types as well as the chiral tags of the atoms. While applying this MCS as a conformation generation restriction turns out to yield valid molecule conformations at a reasonable rate, naturally some molecule conformations generated this way are invalid. This is because of chemically unfeasible bond lengths or impossible molecule geometry: Due to differences in hybridizations and node degrees, along with associated bond angles, the mapping of atoms may potentially be invalid. To tackle cases where this happened, another MCS that considers atom and bond types and chirality is applied. This enables the generation of a valid conformation in most of the attempts. Only a few will still fail due to an issue with the implementation of the MCS-detection used.

Since the generation of new conformations is a costly computation, it is only performed if the conformation of the ligand has a large enough distance from all the other conformations in the assembly. This is expressed by the arithmetic mean of distances from the ligand to all others.

Handling of Missing Ligands Whenever an assembly has at least one ligand without an assigned conformation (indicated by a missing ligand count greater than zero), this condition takes precedence over optimizing the conformation for the least fitting ligand. If a ligand is missing, the search for an improving conformation among the previously sampled is skipped, and instead, the generation of a new conformation matching the targets is performed. This strategy aims to minimize the impact of initial conformation sampling biases, allowing the assembly to better integrate and reflect the geometries of its constituent ligands.

Iteration Control Strategy Sometimes, an optimization step will not yield an improving result, for example when all the newly generated conformations for the worst ligand produce a worse assembly score than the previously assigned conformation. In this case, all new conformations are discarded and no changes to the assembly are made. To avoid the worst ligand being marked as the worst

ligand in the subsequent iteration step (since this would again not improve the assembly), the method described in MolVar [14] is applied, where every ligand is marked to be either "available" or "unavailable". Once an optimization step for a ligand fails to produce an enhancing result, the ligand is marked as unavailable. As a consequence, this ligand cannot be marked as the worst ligand anymore. Once an improvement is made in an iteration step, all ligands are set to available again allowing their conformations to react to the changes made in subsequent steps. The iteration of the optimization stops once all ligands have been marked unavailable, thus no more improvements can be made with the tools at hand. To avoid the "availability reset" occurring too frequently and thus increasing the runtime while potentially only making minor improvements, a threshold is employed. The assembly score's improvement must surpass this threshold to trigger the reset.

Finetuning of the Best Assembly After their initial generation, the assemblies are first optimized with thresholds resulting in a "coarse" optimization. The main goal of the coarse optimization is to identify a good overall geometry for the alignment. Also, if the common core is symmetric and multiple orientations of the ligands are possible, the coarse optimization will determine a good combination of orientations.
The coarse optimization converges faster because the improvement in the assembly score required to reset the availability of the ligands is large. Therefore, only a few iteration steps will be performed. Once no more significant improvements can be made, the iteration stops. Also, the mean distance to the other ligands required to trigger the generation of new conformations is large. Thus, only ligands whose conformation is an "outlier" in the assembly will be changed and adapted to the other ligands in the assembly. After this optimization has been applied to all generated assemblies, the assembly with the best score is chosen as the intermediate result and optimized once more with different threshold parameters to "fine-tune" the assembly. Small changes in a ligand's conformation will set all other ligands available again such that small improvements will be adopted by all ligands. Also, the generation of new conformations has a lower threshold. This is mainly done to enable the superposition of side chains that have previously been aligned closely. In this step (provided they are covered by the pairwise MCS, see 3.4.3), they are superpositioned with exact coordinates.

Bruteforcing of Outliers Using the MCS of the target and the ligand whose conformation is to be improved has one downside. If there exists no

MCS extending the common core by a lot, the conformation generation in the optimizer is unlikely to provide improving conformations. This is especially true for ligands that are much larger than the others or for ligands where a large MCS is"interrupted"by a feature not shared with the others (see Fig. 3.6). Those ligands are identified after the optimization iteration by their large deviation from the alignment (i.e., their mean distance to all the other ligands is much greater than the assembly score). In the case of the presence of such outliers, a brute-forcing of better-fitting conformations is attempted. The conformation generation settings are similar to the initial conformation sampling (see 3.3) but with a much greater number of conformations to be generated.

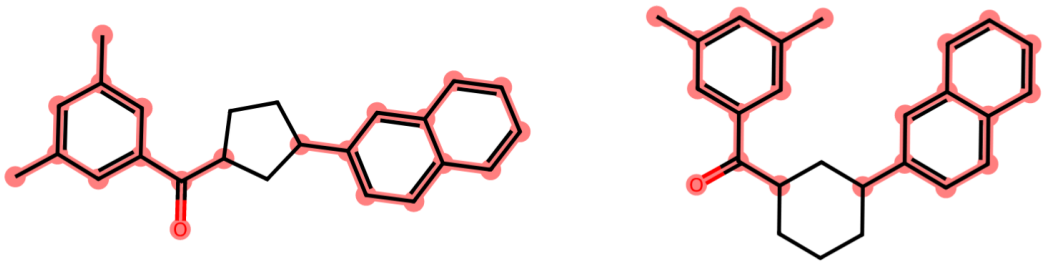


Figure 3.6.: Two very similar compounds that may be aligned suboptimally by CoAler. This is due to the MCS being"interrupted"by the different rings. Therefore, the reference point generation will be restricted to the naphthalene (assuming this is the shared core) and thus not cover the rest of the molecule.

Chapter 4.

4 Validation and Benchmarking

Validating the output of CoAler proved a non-trivial problem. This is due to the complex nature of the task at hand: An alignment of multiple molecules does not have one definite score determining its quality. Multiple metrics such as the average Shape Tanimoto score (Chapter 2) can be calculated to gain insights about the geometric properties of the built alignment. The quality, however, that ideally should be evaluated in the context of this paper is how closely each member of the built molecule alignment resembles its bioactive conformation and positioning in the same environment (i.e. a similar active site of a protein). Although in some cases the shape similarity can be used as a predictor, one can easily think of cases in which chemically distinct molecules can have a high shape overlap or where there exist molecule conformations for an assembly that provide good shape overlap scores but which are far removed from the bioactive conformations. Additionally, the validation had to take into account that the molecules of the assembly should exhibit a common substructure.

4.1. Development and Testing

For the development phase of the project, there was initially the need for molecules sharing a common core structure, without requiring any 3D information or indicators for the bioactivity of the conformations. These structures should mainly be used for unit tests as well as to provide data to try running the software on while it was being built. The evaluation of the alignments

quality during development relied on visual inspection of the alignment using USCF Chimera [18].

For those initial test data sets PubChem BioAssays [1] were used, which are researcher curated molecule assemblies that were usually used in experiments targeting some biological structure, i.e. testing agonistic or antagonistic behavior towards a receptor protein. For identifying candidate bio assays that conform to our requirements, we developed evaluation a utility which downloads the assay's molecule SMILES identifiers and calculates their MCS. The program then produces a report stating metrics of the assembly like the molecule's average number of heavy atoms as well as their maximum and minimum values, the size of the MCS, the MCS size in relation to number of heavy atoms, as well as the distribution of molecule sizes. The program was written in Python and uses the RDKit `Chem.rdFMCS` package to calculate the MCS.

Reports were generated for Bio Assemblies containing up to 100 molecules, as this matches the approximate number of molecules that should be considered according to the assignment. Using the information from the reports, the Bio Assays shown in table 4.1 were chosen. Subsequently, the SMILES descriptions of those assays were provided as part of the CoAler test data sets.

BioAssay ID	**No. Of Molecules**	**Avg. Heavy Atoms**	**Atoms in MCS**
1806504	67	34	28
716614	13	39	23
274396	49	27	16
152374	25	31	14
492217	33	39	28

Table 4.1.: BioAssays chosen as local test data for CoAler from PubChem and their molecular properties.

Although useful for local testing, this approach proved not feasible for evaluating the quality of the CoAler produced alignment at scale. This is due to: (I) While most assays include meta information about the bioactivity values (i.e. IC50 values for experiments using receptor inhibitors), they do not provide any information about the 3D structure of the molecules, against which the alignments calculated by CoAler could be compared and; (II) While PubChem provides a REST based API [19] it does not provide the possibility to search bio assays with a specific common substructure or to parametrize the number

[1] `https://pubchem.ncbi.nlm.nih.gov/bioassay`

of molecules in the assays, which would make calculating the MCS manually difficult, as some BioAssays include thousands of molecules.

4.2. Quality Evaluation Framework

For the actual validation of CoAler we used molecule ensembles from the **N**on-intersecting **B**inding **S**ite **E**nsemble data set (NBSE). The data in the NBSE was originally created together with the binding site ensemble search tool SIENA [1]. SIENA aims to solve the problem of identifying proteins that have structurally similar binding sites by applying an indexing and search strategy on sequentially related structures of a reference protein on entries in the sc-PDB [20].

The NBSE offers nearly 200 protein ensembles with over 9000 ligands that are contained in the proteins active sites. Those were generated by using SIENA to collect structures that were similar to 64 chosen ligands from the PDB [1]. As those ensembles contain the ligands 3D information, which is relative to the aligned binding sites, the NBSE data could be used as a reference point to compare the CoAler alignments. The size of the NBSE ensembles, often containing under 100 ligands, also made it a good fit for using it as a benchmark.

As the first step, we screened all the NBSE data sets to identify the ones in which the ligand molecules contained a sufficiently big MCS. To gain a sizeable collection of data samples, the top 39 data sets were chosen for further evaluation. The results of this first selection process can be found in A.2.

Once we determined the data sets that would be used for the tests, a proper way to compare the 3D information for a given NBSE ensemble and the CoAler output had to be devised. As alignment tools compute their output by changing the coordinates of the molecules as well as their conformations, we used two distance measured to address both of them: (I) The Tanimoto Shape Score was computed for each ligand of the CoAler output to the corresponding ligand in the NBSE ensemble, which can be used to compare the conformations generated by CoAler to the conformations in the binding site; (II) In order to compare the alignment of the found conformations to one another, we used the Kabsch algorithm to compute a point cloud alignment. This uses a mapping from all the ligand atoms in the NBSE ensemble to the atoms in the CoAler output. The resulting Root Mean Square Deviation (RMSD) value was used as the distance metric.

Like we described in Chapter 7 the program is extensible configurable. Therefore, the goal of our validation framework was not only to ensure the general functionality of our program, but also to gain insights about which configuration should be applied for a given set of ligand molecules.

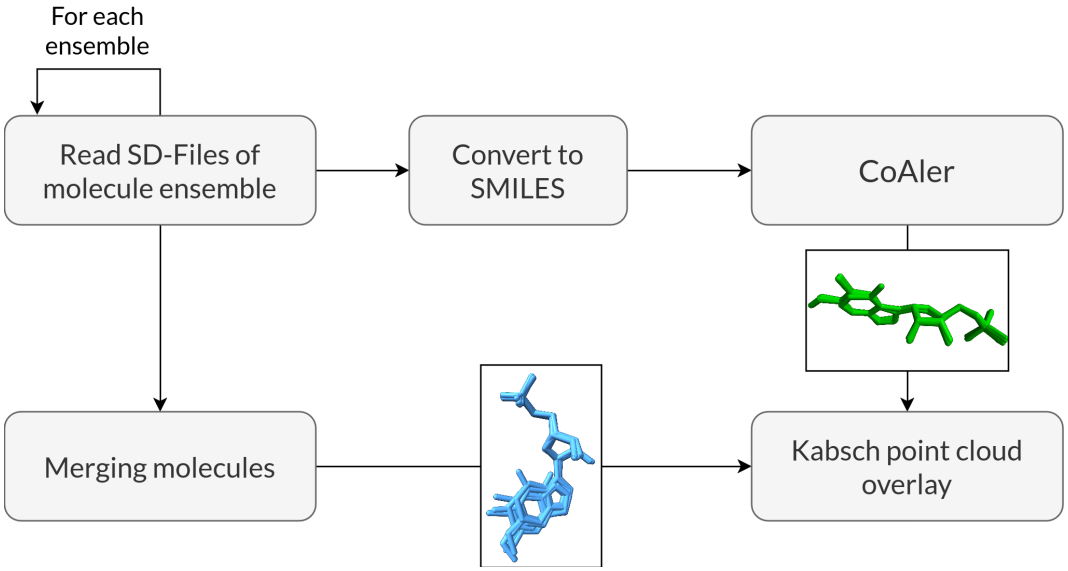


Figure 4.1.: Workflow for calculating the SIENA RMSD metric in the course of the CoAler validation and quality evaluation framework. For all the inspected NBSE ensembles, we compute a point cloud overlay of the ensemble ligands with the alignment produced by CoAler.

To achieve both of these goals, we developed a Python program[2], the basic steps of which are depicted in fig. 4.1. First, the SD-files of the NBSE ensembles are read into an RDKit molecule representation. From these we determine the SMILES strings, in order not to give CoAler any 3D reference points. Subsequently, we call CoAler on the converted molecules. After the alignment calculated, we merge both the CoAler alignment molecules and the NBSE ensemble molecules into two RDKit "macro" molecules from which we can compute the Kabsch point cloud overlay. This provides us with a RMSD value, that denominates the distance of the average atom pair from our macro molecules in Å. This value will be subsequently referred to as the "SIENA RMSD". Additionally, the Tanimoto Shape Score of each of the conformations from the input set to their counterpart in the CoAler output was computed and will be referred to as the "Conformation Score".

[2] `github.com/ciw-seminar-2023/nbse-analysis`

4.3. Data Collection

To achieve manageable run times while offering some variance in the configurations values, we chose a set of configurations, which does not differ too much from the default values of the configuration parameters. Those were chosen during development because they provided stable results in reasonable runtimes. The exact configuration values that were chosen, can be found in table 4.2. An in-depth description of the configuration parameters and their impact on the execution of CoAler can be found in Chapter 7. We ran all tests on an Intel 13th generation i5-13600K processor, offering 20 threads. Every single alignment was given a 15-minute time limit to complete. If not completed, it was marked as failed.

No.	**No. Confs.**	**Assemblies**	**Opt. Coarse**	**Opt. Fine**	**Opt. Steps**	**Divide**	**Core Alg.**
1	10	1	0.5	0.01	100	true	mcs
2	10	10	0.3	0.01	100	true	mcs
3	20	10	0.3	0.01	100	true	mcs
4	20	10	0.3	0.01	100	true	murcko
5	20	10	0.5	0.05	100	true	mcs
6	20	10	0.5	0.1	100	true	mcs
7	20	10	0.5	0.2	100	true	mcs
8	30	1	0.5	0.01	100	true	mcs
9	40	10	0.3	0.01	100	true	mcs
10	60	1	0.5	0.01	100	true	mcs

Table 4.2.: Values for the configuration parameters used in the validation process.

Using those configurations and our 39 ensembles meant computing about 350 alignments. This demonstrates that empirically finding optimal values (if they exit at all) for certain program parameters is very complicated, as CoAler provides many scalar parameters and even with a coarse sampling of those, the search space would easily get too big if one was to compute a significant set of alignments for each parametrization.

Therefore, we had to use a narrow, discrete sampling of those parameters and leave some entirely unchanged. As can be seen in table 4.2 we left the `Divide` and `Optimizer Steps` parameters at their default settings for all the tests, as they were deemed not important for the final outcome. Altering the `Optimizer Steps` parameter was not necessary in any case during testing, as all the optimizations finished before the step limit was reached. The `Divide` Parameter was also kept activated, as it would prevent the runtime from exceeding the time limit by limiting the number of conformations that would be considered in cases that were either highly symmetric or whose MCS would match multiple times in its ligands. Those parameters can be useful in scenarios

involving bigger molecule ensembles or for explicitly addressing highly symmetric molecules.

We also chose to use the MCS as the core calculation method for most assemblies, as computing the Murcko Scaffold does not work for molecule ensembles containing no ring systems, limiting the amount of ensembles we could have evaluated. In order to evaluate the Murcko Scaffold based core calculation, we first computed all the MCS based alignments, then chose configuration 3 as it produced the alignments with the highest average Tanimoto Shape Score and reevaluated our test sets with the Murcko Scaffold core calculation.

4.4. Data Analysis

For each of the calculated alignments we collected the "SIENA RMSD", the "Conformation Score" as well as the average Tanimoto Score of the winning assembly (the value CoAler optimizes) which will be referred to as "Local Similarity". The results were analyzed using DuckDB [21] as well as Python Pandas [22]. We created all visualizations using plotnine [23] which reimplements the R ggplot data visualization framework [24] for use with Python. All the chemical visualizations were created by RDKit using `rdkit.Chem.Draw` and by UCSF Chimera [18].

4.4.1. General Observations

To inspect how close the alignments generated by CoAler come to the position and conformation in the NBSE, we calculated all pairwise Pearson correlation coefficients for the metrics we collected: "SIENA RMSD", "Conformation Score" and "Local Similarity". Those will be subsequently just referred to as the "metrics" while all other fields (i.e. the program parameters and the chemical properties of the ensembles) will be referred to as the "dimensions". The correlation coefficients of the metrics are shown in fig. 4.2. A complete list of the average performance for each of the ensembles can be found in table A.3.

As can be seen from figure 4.2 there exists a strong correlation ($r = -0.781$) between the SIENA RMSD and the Local Similarity. This was expected and shows that at least under the assumption that the alignment of ligands present in the NBSE, CoAler tries to optimize the correct value: The program tries to build ligand assemblies with higher average Shape Tanimoto Scores among them. The data suggest that the more shape overlap is achieved, the closer the

Figure 4.2.: Pearson correlation coefficients between SIENA RMSD, Runtime, Local Similarity and Conformation Score. The highest negative correlation exists between SIENA RMSD and Local Similarity.

configuration is to the spacial configuration and conformation in the binding pockets the ligands were measured in.

As shown in figure 4.3, the best Local Similarity values that CoAler achieved were at around 72% shape similarity. A perfect shape overlap of 100% was never achieved, but was also not expected, as calculating the average shape overlap over a heterogeneous set of molecules will always yield smaller values, as differences in molecule size will lead to not overlapping regions, subsequently lowering the score. The lowest scores achieved were around 30%. The SIENA RMSD values ranged from around 1.6Å to about 8.5Å.

The ensemble for which the lowest SIENA RMSD values could be achieved are `3PCI`, `3KE8`, `2VKE` and `1D0S`. `2HCT` and `2W0V` produced the highest values. For all the examples that are explicitly discussed in the following text, we provide an overview containing a sample of ligand molecules (usually there was not

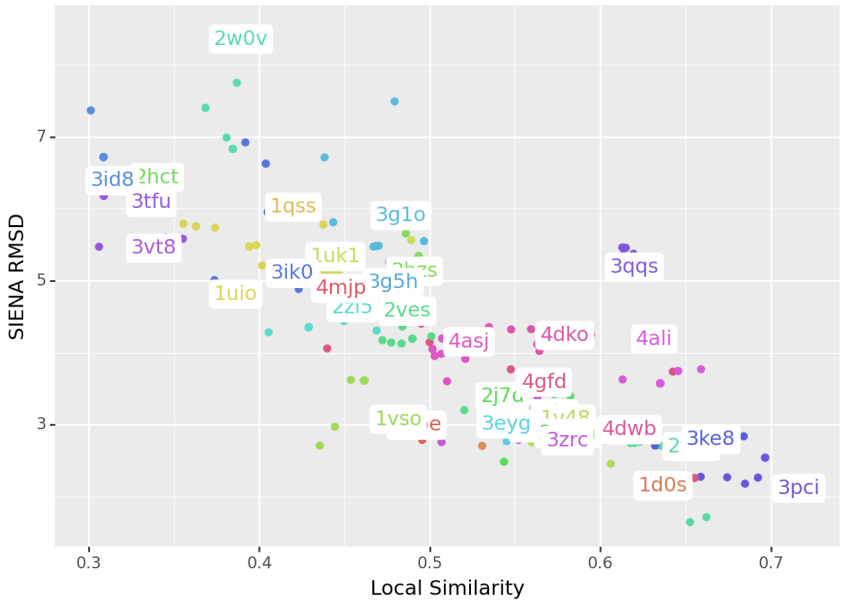


Figure 4.3.: Local Similarity plotted against the SIENA RMSD of the considered ensembles. The best Local Similarity scores were achieved at approximately 72 %. The SIENA RMSD ranged from 1.6Å to 8.5Å. The chart shows a strong linear correlation between the Local Similarity values and the SIENA RMSD.

enough space to display all molecules contained in the ensemble) as well as the best CoAler alignment and the alignment of the result with the original ligand contained in the NBSE.

From the provided visualization in appendix A we can attribute the good alignment achieved for `3PCI` to the small size of the molecules contained in the ensemble (fig. A.1). Additionally, as all but one molecule contain a benzene ring, the pairwise MCS calculation in the optimization will compensate for the small initial MCS. Similarly, the ligands of `3KE8` are quite small (fig. A.4). They could also be aligned with high Local Similarity values, although the resulting conformation of the phosphate groups differs from the more twisted conformation in the NBSE data. As CoAler has no access to the binding pocket context and only optimized the Local Similarity measurement, results like this ought to be expected.

Some more challenging alignments can be found in `1D0S` (fig. A.10) and `2VKE` (fig. A.7). For `1D0S` CoAler managed to achieve almost perfect Local Similarity

as the molecules almost overlap completely. Also, here the conformational space is not big, but contains at least some rotatable bonds in the vicinity of the phosphate group. Although achieving a high Local Similarity score, the SIENA RMSD was only 2.15Å as the calculated alignment differs from the NBSE ensemble through the overall conformation. This can also be attributed to CoAler not being provided with data about the binding pocket, which presumably forces the conformations of the NBSE ligands. `2VKE` contains bigger molecules than `1DOS` and did achieve the best overall SIENA RMSD with 1.6Å. This close alignment can on one hand be explained by the big MCS the molecules exhibit. Although the initial MCS marked in fig. A.7 does not contain all the four connected ring structures, most of the ligands included in the ensemble will have them contained in their pairwise MCS results. This rigid ring structures in the center of the molecules should also make the overall conformational space smaller, which is beneficial for the result. To give an example for an ensemble for which the computed alignment exhibits a high Local Similarity but is far removed from the data seen in the NBSE, we included `3QQS` (fig. A.19). Here too, the NBSE ensemble exhibits less ligand overlap, which is most probably due to the geometry of the binding pockets the ligand structures were measured in.

As examples where finding a good alignment was consistently difficult, we included `2WOV` and `2HCT`. For, `2WOV` the molecules generally contain ring structures that are connected in a chain-like fashion. As depicted in fig. A.25 the ligands in the NBSE ensemble are aligned along those "chains", but the chains are quite heterogeneous in their composition (fig. A.22). This poses a problem for CoAler as the computed initial MCS only consist of a single benzene ring. Subsequently, we can observe CoAler trying to align those long chains with only one of their rings aligned to one of the possible other ring structures of the other molecules. This alignment has high degrees of conformational and rotational freedom, and therefore fails to compute a satisfying result.

For `2HCT`, fig. A.26 shows molecules that sometimes contain macro cycles. The NBSE ligands (fig. A.29) show conformations in which the non-macro-cycle molecules are bent along the existing macro-cycles. CoAler uses the MCS (a hetero five-ring) as an initial reference point. From this the neighboring groups were aligned, but the further we go from the initial MCS, the more degrees of freedom are added, which subsequently makes finding fitting conformations for the optimization step much harder. Additionally, the heterogeneous nature of the ensembles molecules limits the usefulness of the pairwise MCS calculations for the optimization.

As visually inspecting and discussing all the computed alignments does not scale for the over 300 alignments at hand, subsequently we will discuss the alignments based on the gathered metrics.

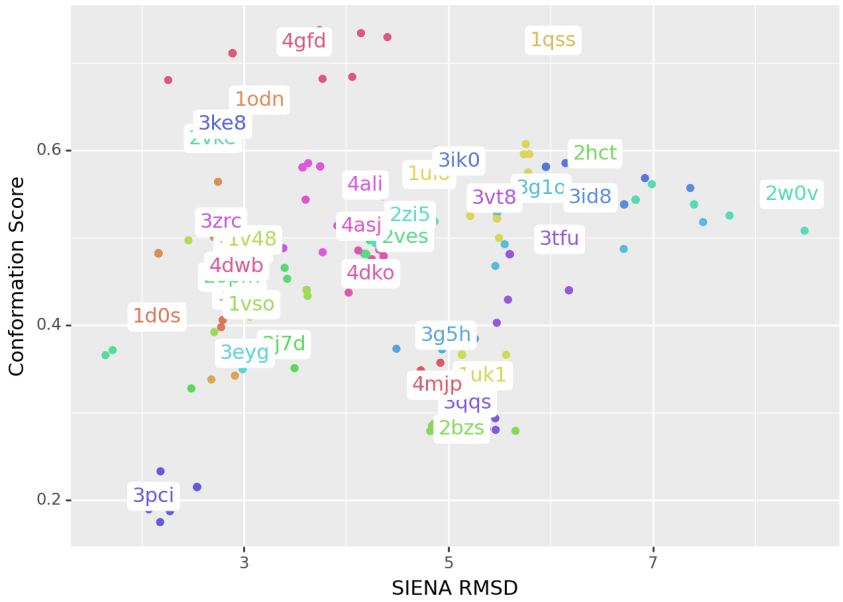


Figure 4.4.: Conformation Score and SIENA RMSD of considered ensembles. Ensembles with low Conformation Score can be aligned with a low SIENA RMSD (i.e. ensemble 3PCI), ensembles like cluster around 2BZS had a high Conformation Score, but also poor RMSD values.

Regarding the correlation of the Conformation Score with the other metrics, the implications are not as clear as with the SIENA RMSD. As can be seen in fig. 4.2 the Conformation Score has no strong correlation with any of the other metrics. Our expectation initially was that we would need to not only compute the right spacial translations and rotations of the molecules, but also to match the conformations of the NBSE ensembles to gain good RMSD values. As shown in fig. 4.4 this is not generally the case: We can see that ensembles with alignments that had a low Conformation Score could be aligned with a low SIENA RMSD value (i.e. ensemble `3PCI`) while ensembles like the cluster around `2BZS` that had a high Conformation Score produced rather poor RMSD values when comparing the full alignment.

One explanatory model could be that producing a good alignment of molecules is largely disconnected from computing the actual conformations but optimizing the spacial positioning of the molecules to one another is more important in

order to get a lower RMSD value. Another explanation could be that the computed values for the Conformation Score are not reliable in all cases: the current implementation uses the RDKit `rdMolAlign.AlignMol` function to align the molecule pair from the NBSE and the CoAler result onto one another. This function computes the MCS of both the molecules and then uses the Kabsch algorithm to align the atoms, matching the MCS in both molecules to one another. In the development phase, this function has sometimes shown problematic behavior when used on the same molecules, as the MCS algorithm does not properly handle molecule chirality in some cases.

4.4.2. Analyzing ensemble properties

To gain an understanding of how the qualities of the molecule ensemble influence the behavior of CoAler, we analyzed the correlation coefficients of the metric dimensions that are summarized in table A.2. Most of the correlations observed here we did expect before running the experiment. So do all metrics (except for the Runtime) show strong correlations with the size of the MCS in the input set as well as with the average size of the molecules. The SIENA RMSD usually shows the same tendencies as the Local Similarity, which is unsurprising as they are highly correlated themselves, as shown in Section 4.4.1.

Most interesting is the connection of the Local Similarity value with the size of the ensemble's MCS. As can be seen fig. 4.8 ensembles with a bigger MCS usually produce higher Local Similarity scores and therefore better alignments with the NBSE data. There are few outliers: `3PCI` is an ensemble of molecules with only an average size of 11 heavy atoms with all the molecules containing a benzene ring. Therefore, the conformational space is limited which makes it easy to compute an alignment with high Local Similarity. On the other hand the ensemble `1ODN` contains a big MCS but sometimes computes poor results. In this case this behavior can be attributed to the MCS being a long chain-like structure (fig. A.14), which showed challenging behavior due to the chirality of member atoms of the chain not being handled well by the RDKit MCS algorithm.

The size of the molecules also were determining for the results that our program could produce: As shown in fig. 4.6 and as stated before, the smaller the molecules were, the higher the produced Local Similarity values were. There were some outliers. The ensemble `2VKE` produced a high Local Similarity while containing on average large molecules. Similarly, the ensemble `3TFU` produced poor Local Similarity values while not containing as molecules as big as other

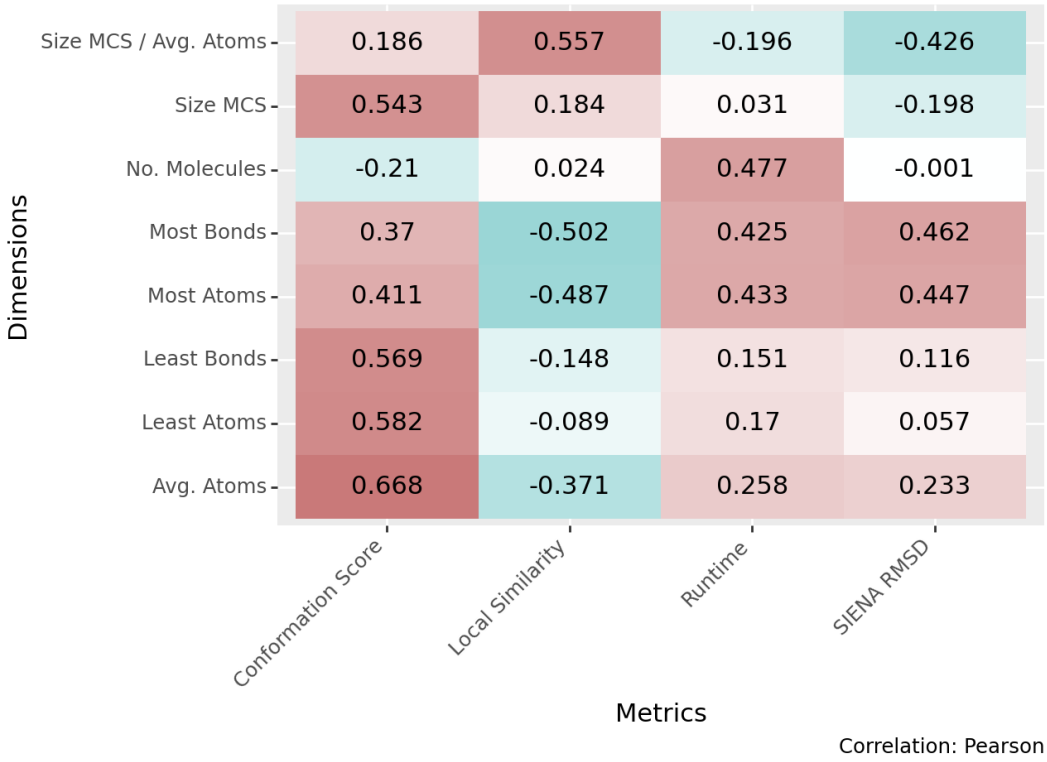


Figure 4.5.: Pearson correlation coefficients of dimensions and metrics. The Conformation Score is highly correlated with the size of the MCS and the number of atoms and bonds. The Local Similarity has the highest correlation with the proportion of the MCS in the molecule. The runtime correlates with the number of molecules and the most atoms and bonds in the ensemble. The SIENA RMSD has a positive correlation with the highest number of bonds and atoms and a negative correlation with the proportion of the MCS in the molecule.

ensembles that produced better values. This can be understood more easily by examining the size of the MCS that those outliers offer (fig. 4.7). While `2VKE` contains a large MCS with multiple ring structures, `3TFU` only contains a single Benzene ring as its MCS, which matches multiple times in some of its molecules and though its symmetry limits the amount of conformations CoAler can use to compute the alignment (at least with the `Divide` option enabled).

The size of the molecules also were determining for the results that our program could produce: As shown in fig. 4.7 and as stated before, the smaller the molecules were the higher the produced Local Similarity values. There were

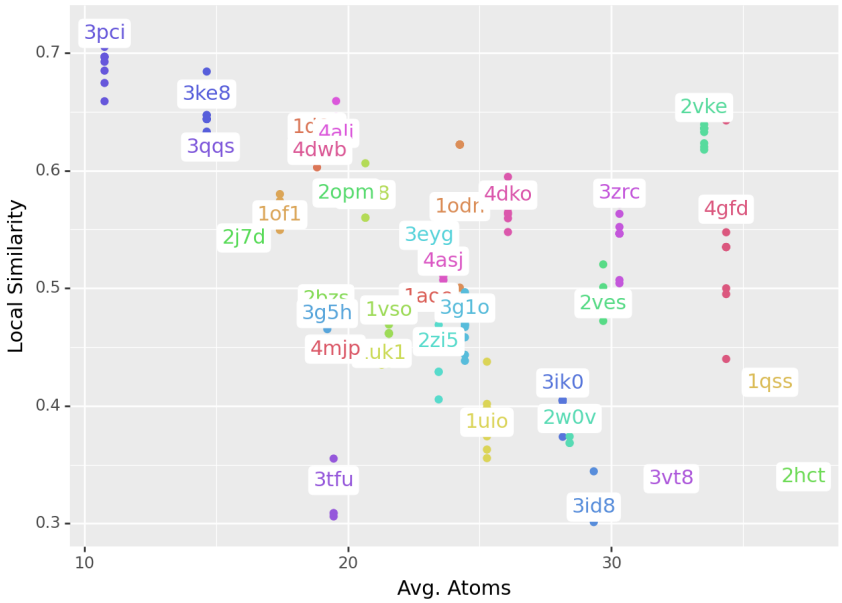


Figure 4.6.: Local Similarity and average atoms of considered ensembles. As the number of average atoms increases, the Local Similarity Score decreases.

some outliers though. The ensemble `2VKE` produced a high Local Similarity while containing on average large molecules. Similarly, the ensemble `3TFU` produced poor Local Similarity values while not containing molecules as big as other ensembles that produced better values. This can be better understood by examining the size of the MCS from those outlier ensembles. While `2VKE` contains a large MCS with multiple ring structures (fig. A.7), `3TFU` only contains a single benzene ring as its MCS, which matches multiple times in some of its molecules and through its symmetry limits the amount of conformations CoAler can use to compute the alignment (at least with the `Divide` option enabled). The Local Similarity was also sensitive to the biggest atom dimension as well as the maximum amount of bonds in an ensemble molecule. This was also expected as those dimensions should be correlated with the average amount of atoms of the ensemble.

The properties of the ensemble in question were also most determining for the time that was needed to compute the alignment. Hereby the amount of molecules was the dimension with the highest correlation, as can be seen in fig. 4.5. This also was expected as the amount of pairwise conformation comparisons that need to be computed scales exponentially with the number of molecules an ensemble contains. More molecules also make the optimization

Figure 4.7.: Local Similarity plotted against average atoms of considered ensembles, sized by the amount of heavy atoms in the ensembles MCS. As the number of average atoms increases, the Local Similarity Score decreases. Outliers like 2VKE and 3TFU can be explained by the size of their MCS. 2VKE has a large MCS with multiple ring structures, 3TFU has a single benzene ring as MCS.

slower since for computing new conformations, more ligands in the assembly will have to be checked for better scores. Hereby, the exact amount of added complexity depends on the chosen optimizer settings. The amount of time it takes to compute an alignment is also dependent on the size of the molecules in the set mostly because the RDKit conformation generator takes up more time to generate chemically valid conformations for bigger molecules.

4.4.3. Analyzing Program Parameters

An additional goal of the validation was to evaluate different program parameters to provide a starting point for potential users. We therefore correlated the measured program metrics with the configuration parameters. The resulting correlation coefficients can be found in fig. 4.9. Unfortunately, none of the program parameters seemed to yield strong correlations with any of the measured metrics except for runtime. We attribute this behavior to (I)

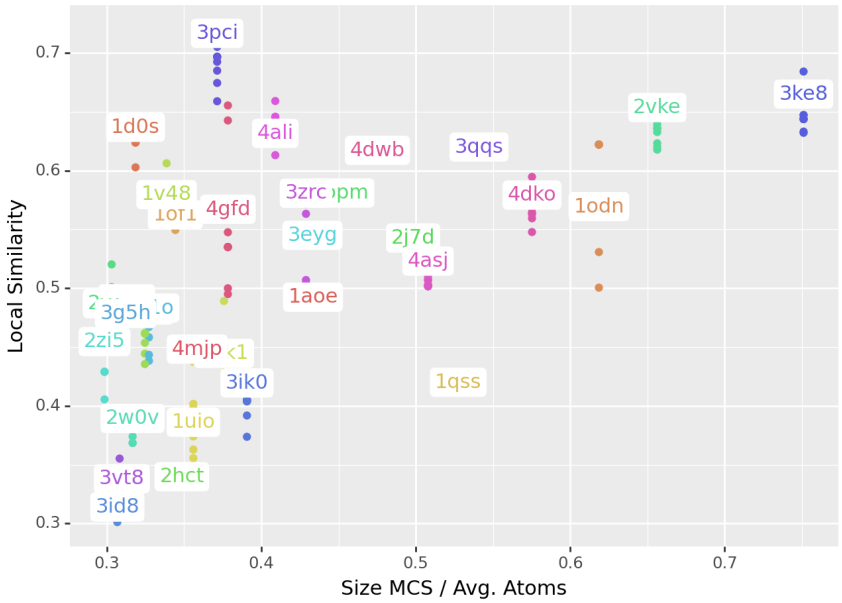


Figure 4.8.: Local Similarity and proportion of the MCS to the average number of atoms of considered ensembles. The larger the MCS, the higher the Local Similarity score.

There is only little variance in most of the program parameters we tested. As described before, exploring the full parameter space on a representative number of ensembles would have taken more time than was available in the context of this research project. (II) In the test setup we varied more than one parameter at one time in order to provide at least some variance of parametrization for the calculation of our alignments and in order to avoid systemic errors by neglecting essential parameters. Due to this, the changes we observed going from one configuration setting to another might not only be caused by this change but by other changes we made in the configuration. Therefore, the benefit of one configuration change could be counteracted by a change in another option. To address those problems, a further validation process should be devised for which all the single parameters are sampled and analyzed programmatically. This might be done on a smaller but representative test set for which the one from this project could be the basis.

Although the other program options did not yield meaningful correlations, we could gain insights about the runtime behavior, especially for the core calculation method. As stated in Chapter 3 CoAler can utilize either the MCS or the Murcko Scaffold of the MCS as the method for calculating the common

Figure 4.9.: Pearson correlation coefficients of dimensions and metrics. No significant correlations were detected.

core structure. When testing using the Murcko Scaffold method, the calculation was limited to molecules containing at least one ring structure in their MCS. Also, we observed a steep incline in the time it took for some of the alignments to be computed. While the runtime increased (fig. 4.10), the results of the Murcko Scaffold based approach could also only surpass the best MCS based results in the case of `4GFD` (fig. A.16). As CoAler uses the common substructures as fixed reference points to align against initially and to search for better conformations in the optimization step, a smaller substructure as provided by the Murcko scaffold was expected to produce worse local similarities. Additionally, for molecules that contain only one ring structure, the Murcko Scaffold will evaluate to that single ring, which is highly symmetric and will therefore lead to more valid conformations than CoAler has to use for its computation.

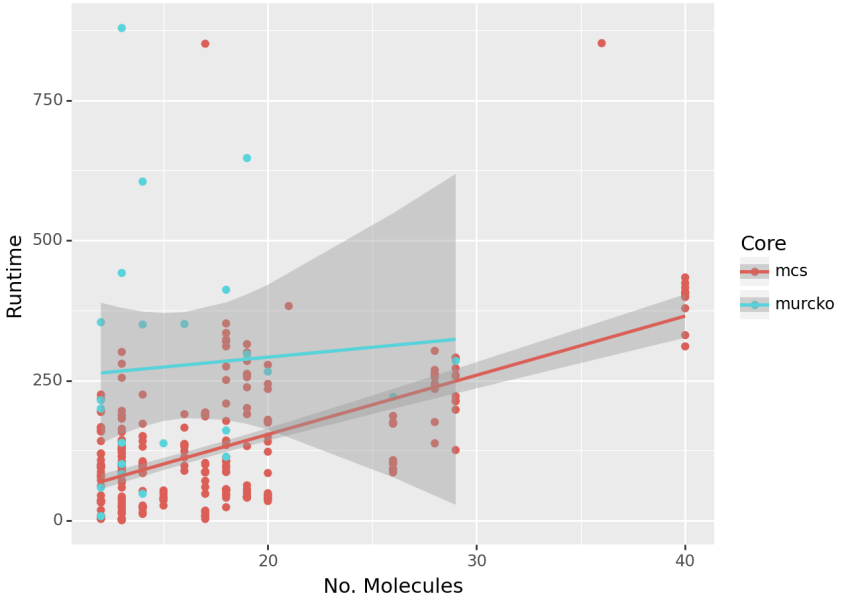


Figure 4.10.: Runtime plotted against the number of molecules colored by the used core calculation algorithm. The runtime increases with the growing number of molecules in the analyzed ensembles. A drastic increase in computation time can be observed for some ensemble when using the Murcko Scaffold to compute the common core structure.

Chapter 5.

5 Discussion

5.1. Validation

Although the metrics chosen for the validation provided certain insights about the geometric properties of the alignment, they could be suboptimal for analyzing all the outputs we produce using CoAler.

For instance, the RMSD is susceptible to differences in the size of molecules. In case of small molecules, all applied transformations only apply changes on very small lengths, compromising comparability. Therefore, the RMSD should be approached with caution when comparing molecule ensembles which differ in molecule size. Since the ensembles used for the validation varied slightly in the size of the molecules, the RMSD might not have been ideal for the analysis.

To obtain a clearer picture of the correlations between the metrics, it may be beneficial to utilize average values for the calculations. This approach could help mitigate the impact of outliers and provide a more representative summary of the data. However, it is important to acknowledge that using average values would have resulted in a reduction of available data points, potentially affecting the statistical robustness of the analysis.

To enable users to utilize the adjustable parameters most effectively for their analysis, further consideration of these parameters is necessary. When precise instructions can be formulated on how the parameters affect the output, targeted application becomes possible without the requirement for trial-and-error sessions by the user.

Conducting additional tests in future stages of the project and comparing CoAler with other alignment tools could offer valuable insights into its performance and might help to optimize the quality of the produced alignments. Also, the establishment of a unified framework for alignment validation could prove beneficial.

5.2. Future Improvements

To further enhance the efficacy of CoAler, several enhancements to the algorithm could be integrated into forthcoming projects.

One prospective improvement involves the recognition that molecules may possess multiple shared core structures. Hence, instead of solely focusing on identifying the maximum common substructure, all common substructures of a specified size could be systematically extracted and evaluated for alignment.

In addition to the exploration of multiple shared core structures, it would be beneficial to incorporate considerations of chirality within these core structures. By accounting for chirality, the algorithm could better capture the stereo chemical aspects of molecular similarity, leading to more refined alignment outcomes.

Moreover, a logical progression in algorithmic refinement involves geometric optimization. After the alignment process, fine-tuning bond angles and lengths through geometric optimization can optimize the spatial arrangement of molecules. This optimization aims to maximize volume overlap between aligned structures, thereby enhancing the accuracy and reliability of the alignment procedure. By systematically adjusting molecular geometry post-alignment, the algorithm can more effectively identify and align structurally similar regions, ultimately contributing to improved alignment results. Section 5.2.1 will discuss a geometric optimization approach, which can be integrated in the future and was already considered in earlier phases of this project.

Additionally, the generation of new conformations during the assembly optimization is another aspect worth improving. CoAler has to address outliers in the alignment through a brute-force mechanism, as discussed in 3.4.3. These outliers only occur due to the disadvantages of the MCS-based reference generation approach. An enhancement of the conformation generation, e.g., calculating more appropriate reference points for the embedding instead of relying solely on the MCS, will remove those outliers. These improvements will lead to more

accurate alignment results, and a brute-force approach for outliers would not be required anymore.

5.2.1. Geometric Optimization

Iterative Closest Point Algorithm The Iterative Closest Point (ICP) algorithm finds extensive application in the rigid body geometric alignment of three-dimensional models, particularly when an initial estimation of the relative pose is available [25].

Typically, the ICP algorithm follows two main steps [26]:

- Identify correspondence set $K = (p, q)$ between the target point cloud P and the source point cloud Q transformed using the current transformation matrix T [26].
- Refine the transformation matrix T by minimizing an objective function $E(T)$ defined over the correspondence set K [26].

In our tested approach, illustrated in Fig. 5.1, we utilized the implementation of the point-to-point ICP algorithm defined in [27] by Open3D. The objective function $E(T)$ is specified in Eq. 5.1. This approach involves obtaining results from the Multi-Aligner, decomposing molecules into R-groups, applying ICP on each R-group except the core, and finally, reassembling R-groups into molecules

$$E(T) = \sum_{(p,q)\in K} ||p - Tq||^2 \tag{5.1}$$

Challenges The challenge when employing a geometric approach is to preserve all molecule properties while making only minor adjustments to atom positions. This underscores the importance of having a well-established initial alignment. At the time when we sought to introduce geometric optimization through the implementation of the ICP algorithm from Open3D, the pairwise distances among the atoms within the generated alignment were found to be too large. That's why we attempted various values for the maximum pairwise distance within the ICP algorithm. Permitting a small maximum pairwise distance (e.g., 0.1) resulted in the failure to find a suitable transformation. Conversely, allowing a large maximum pairwise distance (e.g., 0.6) led to severely flawed transformations, thus distorting molecule properties such as excessively long bonds or inaccurate bond angles.

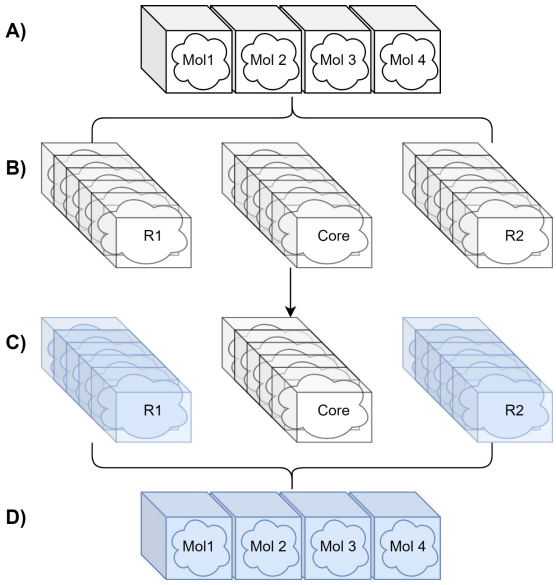


Figure 5.1.: Theoretical Approach for Geometric Optimization using ICP. (A) Results from the Multi-Aligner, (B) Decomposition of the molecules into R-groups, (C) Application of ICP on every R-group except the core, and (D) Reassembly of R-groups into molecules.

5.2.2. Colored Tanimoto

To further refine the alignment process, future projects may consider incorporating the alignment of functional groups. Mapping chemical groups of the same function onto each other increases the likelihood of an accurate overlay, as these groups are typically found in similar regions within the binding pocket. Thus, not only geometric aspects but also chemical characteristics are taken into account.

For this purpose, the color Tanimoto (CT) [28, 29] might be suitable. This similarity metric assesses the overlap of functional groups, each represented by a distinct color. The CT considers six types of functional groups: hydrogen-bond donors, hydrogen-bond acceptors, cations, anions, hydrophobes, and rings. Additionally, merging the volumetric method with the CT could enhance the alignment's accuracy and quality.

As shown in 5.2 [30] the three-dimensional arrangement of the different functional groups can be depicted in colored spheres, whose volume overlaps in an alignment can be calculated as color Tanimoto. Hydrogen-bond acceptors are depicted in red, hydrogen-bond donors in blue, cations in Violet, anions in

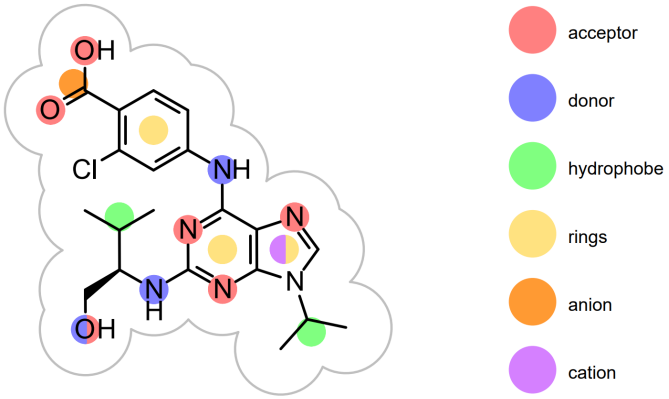


Figure 5.2.: Example of the 2D visualization of the query structure with color atoms on the front page of a ROCS report [30].

orange, hydrophobe groups in green and rings in yellow.

While integrating CT may not significantly improve the runtime performance of CoAler, it is nevertheless likely that the quality of the produced alignments would be improved. If the consideration of chemical properties, for example in the form of color Tanimoto, were to be implemented in the software in the future, it would represent a significant advancement.

Chapter 6.

6 Conclusion

In this project report we presented the novel software tool CoAler.

The development of CoAler has opened up a new facet of the multiple alignment of molecules. By detecting and considering a common core structure, the tool has the potential to be utilized in early-stage drug discovery inquiries, enabling quick, easy, and precise alignments of molecules sharing a core structure. It is particularly applicable to molecules with one large rigid core.

Through further refinement of specific aspects of the algorithm, such as the consideration of chemical properties, like functional groups, the generated alignments can be optimized. Additionally, CoAler could be expanded to handle multiple core structures and several occurrences of a single MCS in the same molecule. Within the optimization step, substituting the MCS with a more appropriate substructure for determining references could enhance alignment quality. Lastly, implementing geometric optimization could optimize the alignment quality even further.

In conclusion, CoAler is a promising tool for molecular alignment of molecules sharing a common core structure and ongoing development efforts have the potential to improve alignment accuracy and efficiency. Its application could offer significant value in addressing specific problem scenarios in drug design.

Chapter 7.

7 User Manual

CoAler is implemented as a C++ 17 program. We used Conan 2.0[1] as the dependency management program. In order to use the RDKit, we created a build configuration and bundled the library using Conan. Our fork, which contains minor changes to the Q3 2023 RDKit release can be found under `https://github.com/ciw-project-2023/rdkit`. In order to efficiently use a GitHub Actions based Continuous Integration Pipeline, the programs dependencies were prebuilt and hosted on a DigitalOcean[2] virtual machine running the official Conan Artifact Mirror[3]. The mirror is available as of March 2024 under `http://server.conan.corealigner.de`.

For building the program we provide a `configure` script as well as a make file with an install target. Building CoAler this way only requires the GCC tool chain, `make` as well as a recent Python version supporting `venv`. For containerized installations we provide a `Dockerfile` which can be used to build a container image with a ready-to-run version of CoAler set as its `ENTRYPOINT`.

The parameters that can be used to configure CoAler for the specific molecule ensemble can be found in table 7.1.

[1] `https://conan.io/`
[2] `https://www.digitalocean.com/`
[3] `https://docs.conan.io/2/reference/conan_server.html`

Short Name	Name	Data Type	Description	Default
h	help		Print the help message	
i	input-file	String	Path to input files	
o	out	String	Path to output files	./out.sdf
j	threads	Integer	Number of threads to use	1
v	verbose	Bool	Activate verbose logging	
	conformers	Integer	Number of conformers per core match to generate for each input molecule	10
	divide	Bool	Divide the number of conformers by the number of times the core is matched in the input molecule. Helps against combinatorial explosion if the core is small or has high symmetry	false
	assemblies	Integer	Number of starting assemblies	10
	core	String	Algorithm to detect core structure (allowed: mcs, murcko)	mcs
	confs-log	String	Optional path to folder to store the generated conformers	
	optimizer-coarse-threshold	Float	Threshold for the optimization step	0.4
	optimizer-fine-threshold	Float	Threshold for the fine optimization step	0.05
	optimizer-step-limit	Integer	Maximum number of steps for the optimizer	100

Table 7.1.: Program parameters available for CoAler. Only the `input-file` parameter is required.

Bibliography

[1] Stefan Bietz and Matthias Rarey. SIENA: Efficient Compilation of Selective Protein Binding Site Ensembles. 56(1):248–259.

[2] Gerald Maggiora, Martin Vogt, Dagmar Stumpfe, and Jurgen Bajorath. Molecular similarity in medicinal chemistry: miniperspective. *Journal of medicinal chemistry*, 57(8):3186–3204, 2014.

[3] Peter Willett, John M Barnard, and Geoffrey M Downs. Chemical similarity searching. *Journal of chemical information and computer sciences*, 38(6):983–996, 1998.

[4] Greg Landrum et al. Rdkit: Open-source cheminformatics. `http://www.rdkit.org`, 2024. Accessed: 2024-03-06.

[5] Nikola Milosavljevic. Rigid body transformations. `https://web.stanford.edu/class/cs273/scribing/2004/class1/lect1.pdf`, 2004. Accessed: 2024-03-06.

[6] Andrew J. W. Orry and Ruben Abagyan, editors. *Homology modeling: methods and protocols*. Number 857 in Methods in molecular biology. Humana Press : Springer. OCLC: ocn758395396.

[7] Fatima Sapundzhi, Metodi Popstoilov, and Meglena Lazarova. Rmsd calculations for comparing protein three-dimensional structures. In *International Conference on Numerical Methods and Applications*, pages 279–288. Springer, 2022.

[8] Jim Lawrence, Javier Bernal, and Christoph Witzgall. A purely algebraic justification of the kabsch-umeyama algorithm. *Journal of Research of the National Institute of Standards and Technology*, 124, October 2019.

[9] Guy W. Bemis and Mark A. Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, 1996. PMID: 8709122.

[10] Yiqun Cao, Tao Jiang, and Thomas Girke. A maximum common substructure-based algorithm for searching and predicting drug-like compounds. *Bioinformatics*, 24(13):i366–i374, 2008.

[11] Scott Fortin. The graph isomorphism problem. `https://era.library.ualberta.ca/items/f8153faa-71bf-4b64-9eb4-f0c6d3b529dd`, 1996. Accessed: 2024-03-06.

[12] Hans-Christian Ehrlich and Matthias Rarey. Maximum common subgraph isomorphism algorithms and their applications in molecular science: a review. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(1):68–79, 2011.

[13] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

[14] Shek Ling Chan. MolAlign: an algorithm for aligning multiple small molecules. *J Comput Aided Mol Des*, 31(6):523–546, June 2017.

[15] Sereina Riniker and Gregory A. Landrum. Better informed distance geometry: Using what we know to improve conformation generation. *Journal of Chemical Information and Modeling*, 55(12):2562–2574, 2015. PMID: 26575315.

[16] Shuzhe Wang, Jagna Witek, Gregory A. Landrum, and Sereina Riniker. Improving conformer generation for small rings and macrocycles based on distance geometry and experimental torsional-angle preferences. *Journal of Chemical Information and Modeling*, 60(4):2044–2058, 2020. PMID: 32155061.

[17] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, Sep 1976.

[18] Eric F. Pettersen, Thomas D. Goddard, Conrad C. Huang, Gregory S. Couch, Daniel M. Greenblatt, Elaine C. Meng, and Thomas E. Ferrin. UCSF Chimera–a visualization system for exploratory research and analysis. 25(13):1605–1612.

[19] PubChem. PUG REST. `https://pubchem.ncbi.nlm.nih.gov/docs/pug-rest`. Accessed: 2024-03-06.

[20] Esther Kellenberger, Pascal Muller, Claire Schalon, Guillaume Bret, Nicolas Foata, and Didier Rognan. Sc-PDB: an Annotated Database of Druggable Binding Sites from the Protein Data Bank. 46(2):717–727.

[21] Mark Raasveldt and Hannes Mühleisen. DuckDB: An Embeddable Analytical Database. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, pages 1981–1984. Association for Computing Machinery.

[22] The pandas development team. pandas-dev/pandas: Pandas. `https://doi.org/10.5281/zenodo.3509134`, February 2020. Accessed: 2024-03-06.

[23] Hassan Kibirige, Greg Lamp, Jan Katins, Gdowding, , Austin, Matthias-K, Tyler Funnell, Florian Finkernagel, Jonas Arnfred, Dan Blanchard, Sergey Astanin, Eric Chiang, Paul Natsuo Kishimoto, Evan Sheehan, Stonebig, Bernard Willers,

Robert Gibboni, Smutch, Yaroslav Halchenko, , Pavel, Brian King, Min RK, John Collins, Zachcp, , Anthony, Bevan Koopman, Carlos H. Grohmann, Dan Becker, Dan Brown, and Daniel Saiz. has2k1/plotnine: v0.8.0. `https://zenodo.org/record/4636791`, 2021. Accessed: 2024-03-06.

[24] Hadley Wickham. ggplot2: Elegant graphics for data analysis. `https://ggplot2.tidyverse.org`, 2016. Accessed: 2024-03-06.

[25] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, 2001.

[26] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[27] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[28] Steven Kearnes and Vijay Pande. Rocs-derived features for virtual screening. *J Comput Aided Mol Des.*, 30(8):609–17, Aug 2016. PMID: 27624668.

[29] Sunghwan Kim, Evan Bolton, and Stephen Bryant. Effects of multiple conformers per compound upon 3-d similarity search and bioassay data analysis. *Journal of cheminformatics*, 4:28, 11 2012.

[30] ROCSReport — Applications. `https://docs.eyesopen.com/applications/rocs/rocsreport.html`. Accessed: 2024-03-06.

Appendix A.

A Experimental Data

No. Confs.	Assemblies	Opt. Coarse	Opt. Fine	Core	Avg. Local Similarity	Avg. SIENA RMSD (Å)	Avg. Conformer Score	Runtime (s)
20	10	0.3	0.01	mcs	0.521	4.032	0.471	136.226
40	10	0.3	0.01	mcs	0.504	4.061	0.466	187.97
20	10	0.5	0.1	mcs	0.519	4.063	0.47	117.323
20	10	0.5	0.05	mcs	0.519	4.063	0.47	121.548
20	10	0.5	0.2	mcs	0.519	4.063	0.47	118.742
30	1	0.5	0.01	mcs	0.513	4.083	0.475	110.871
60	1	0.5	0.01	mcs	0.525	4.093	0.459	113.875
10	10	0.3	0.01	mcs	0.519	4.131	0.462	146.419
10	1	0.5	0.01	mcs	0.508	4.146	0.468	103.871
10	10	0.9	0.4	mcs	0.494	4.202	0.465	105.031

Table A.1.: Average scores of the CoAler computed alignments for different configuration that were used.

PDB ID	MCS Atoms	No. Mols.	Avg. Atoms	MCS Atoms / Avg. Atoms
3KE8	11	17	14.647	75.10%
2VKE	22	12	33.522	65.63%
1ODN	15	28	24.251	61.85%
4DKO	15	13	26.077	57.52%
3QQS	8	19	14.789	54.09%
1QSS	19	21	36.019	52.75%
4ASJ	12	16	23.625	50.79%
2J7D	8	18	16.056	49.83%
4DWB	9	17	18.941	47.52%
3W1T	10	50	21.790	45.89%
2OPM	9	17	19.941	45.13%
1AOE	10	14	23.071	43.34%
3EYG	10	13	23.090	43.31%
3ZRC	13	13	30.308	42.89%
4ALI	8	20	19.557	40.91%
3IKO	11	13	28.154	39.07%
4GFD	13	14	34.357	37.84%
1UK1	8	14	21.286	37.58%
3PCI	4	13	10.769	37.14%
4MJP	7	19	19.526	35.85%
1UIO	9	15	25.280	35.60%
1IE8	12	31	33.710	35.60%
2HCT	13	17	37.294	34.86%
1OF1	6	26	17.423	34.44%
1V48	7	12	20.667	33.87%
3G1O	8	18	24.444	32.73%
1M8D	6	17	18.471	32.48%
1VSO	7	20	21.560	32.47%
1QKN	7	24	21.583	32.43%
1DOS	6	12	18.833	31.86%
2WOV	9	12	28.417	31.67%
2BZS	6	40	19.176	31.29%
3G5H	6	18	19.222	31.21%
3VT8	10	36	32.278	30.98%
3TFU	6	13	19.462	30.83%
3ID8	9	12	29.333	30.68%
2VES	9	13	29.692	30.31%
3SOR	10	23	33.391	29.95%
2ZI5	7	29	23.448	29.85%

Table A.2.: Top 39 SIENA binding size ensembles that contain ligands with a sizeable MCS.

PDB ID	Avg. Local Similarity	Avg. SIENA RMSD (Å)	Avg. Conformation Score
3PCI	0.694	2.299	0.204
3KE8	0.648	2.703	0.617
2VKE	0.637	2.542	0.575
4ALI	0.636	3.737	0.559
1DOS	0.627	2.527	0.411
3QQS	0.616	5.329	0.297
4DWB	0.615	2.861	0.467
1ODN	0.59	2.459	0.524
1V48	0.577	2.905	0.497
2OPM	0.575	3.052	0.451
4DKO	0.57	4.17	0.477
1OF1	0.561	2.92	0.429
3ZRC	0.549	2.899	0.515
3EYG	0.545	2.98	0.364
4GFD	0.544	3.411	0.711
2J7D	0.542	3.237	0.366
4ASJ	0.51	4.025	0.52
1AOE	0.491	2.987	0.472
2BZS	0.491	5.152	0.288
2VES	0.489	4.145	0.481
3G5H	0.477	4.933	0.381
3G1O	0.467	5.85	0.521
1VSO	0.462	3.295	0.428
2ZI5	0.446	4.563	0.526
4MJP	0.446	4.937	0.342
1UK1	0.445	5.204	0.352
1QSS	0.42	6.025	0.726
3IKO	0.404	5.921	0.585
1UIO	0.39	5.499	0.551
2WOV	0.382	7.461	0.54
2HCT	0.34	6.434	0.596
3VT8	0.338	5.451	0.545
3TFU	0.333	5.813	0.463
3ID8	0.313	6.577	0.543

Table A.3.: Average metrics of the CoAler computed alignments for all the NBSE ensembles sorted by their Local Similarity value. Ensembles for which we could not compute alignments in the set time limit of 15 minutes were omitted.

A.1. Ensemble 3PCI

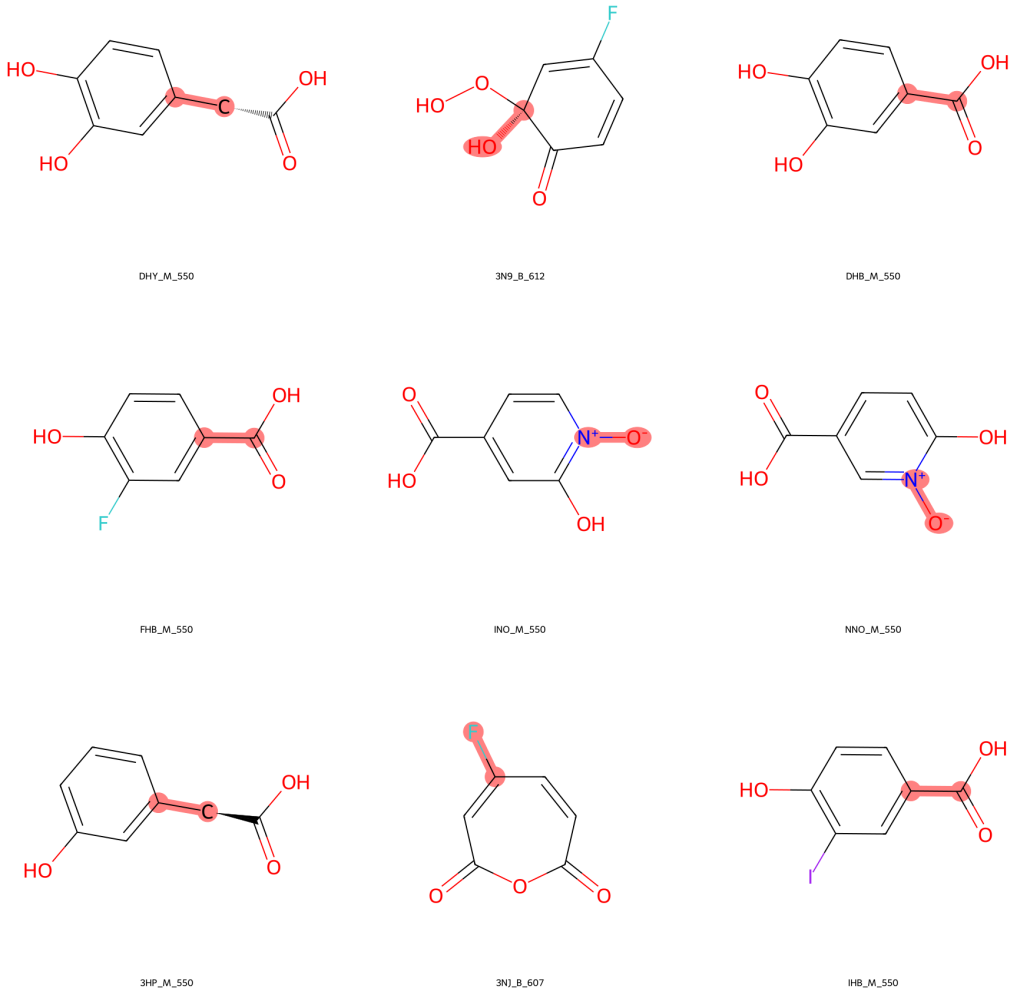


Figure A.1.: Sample of the ligand molecules contained in the ensemble 3PCI.

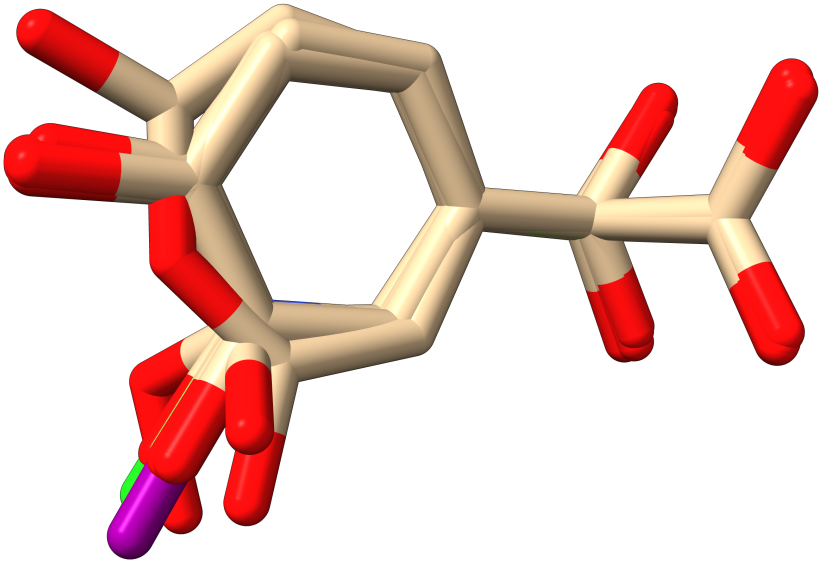


Figure A.2.: Best achieved alignment of the ensemble `3PCI` using CoAler.

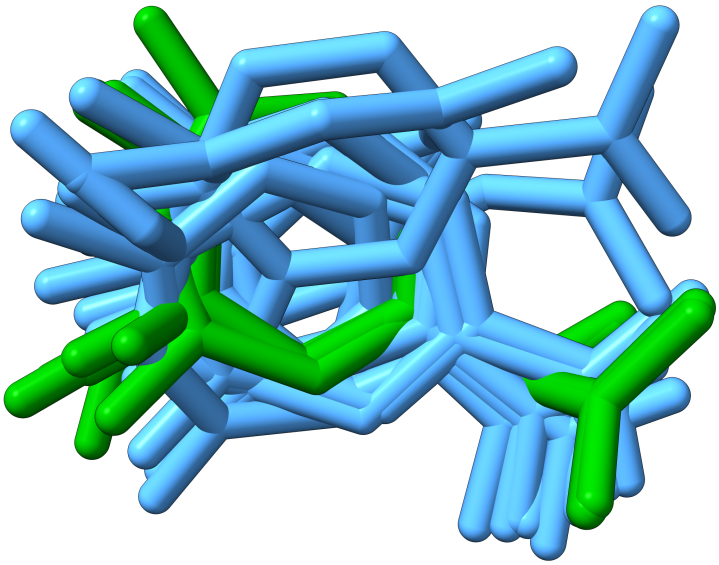


Figure A.3.: Comparison of the best achieved alignment of the ensemble molecules from `3PCI` with the ligand coordinates from the NBSE. Green: CoAler result; Blue: NBSE ensemble

A.2. Ensemble 3KE8

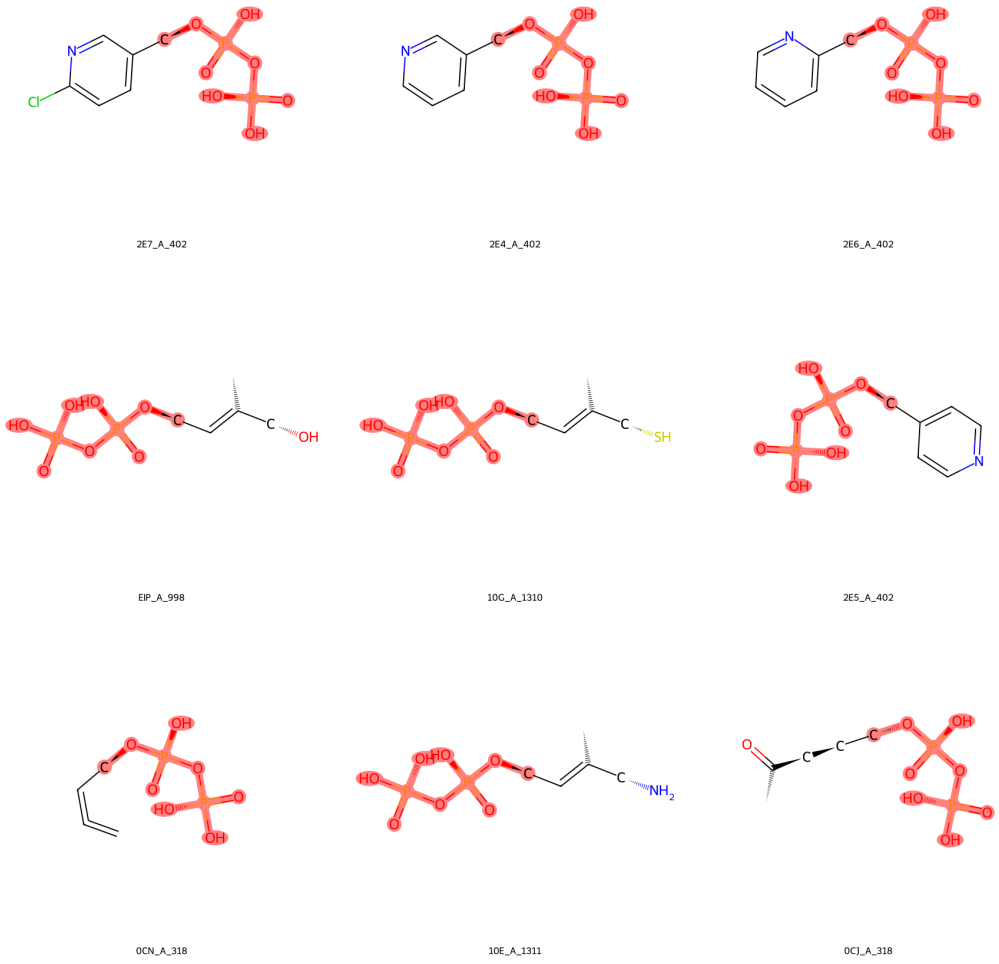


Figure A.4.: Sample of the ligand molecules contained in the ensemble 3KE8.

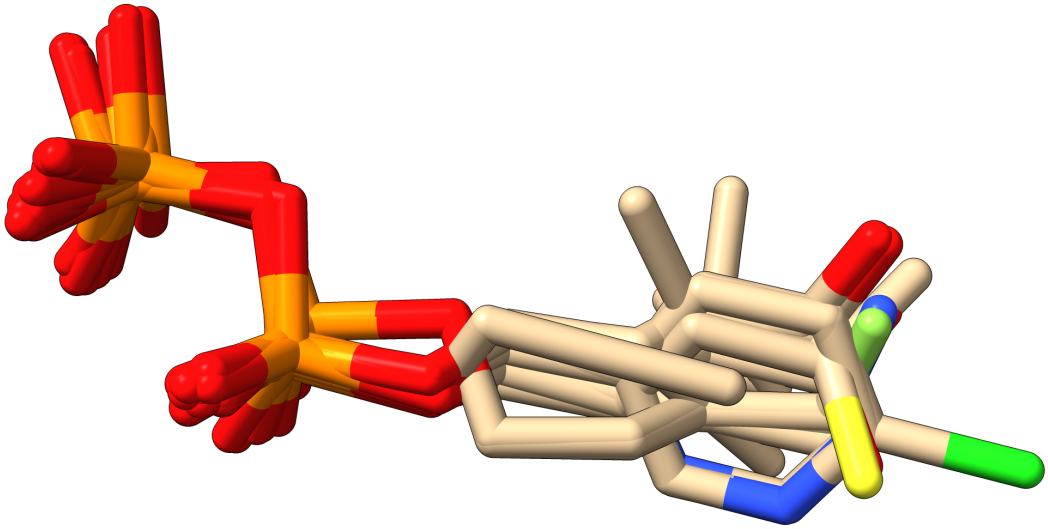


Figure A.5.: Best achieved alignment of the ensemble `3KE8` using CoAler.

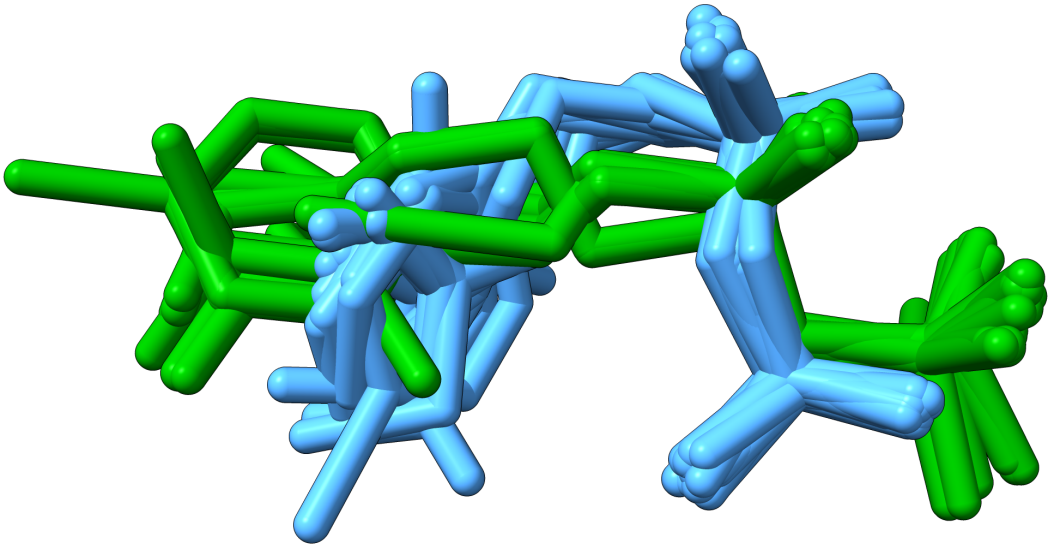


Figure A.6.: Comparison of the best achieved alignment of the ensemble molecules from `3KE8` with the ligand coordinates from the NBSE. Green: CoAler result; Blue: NBSE ensemble

A.3. Ensemble 2VKE

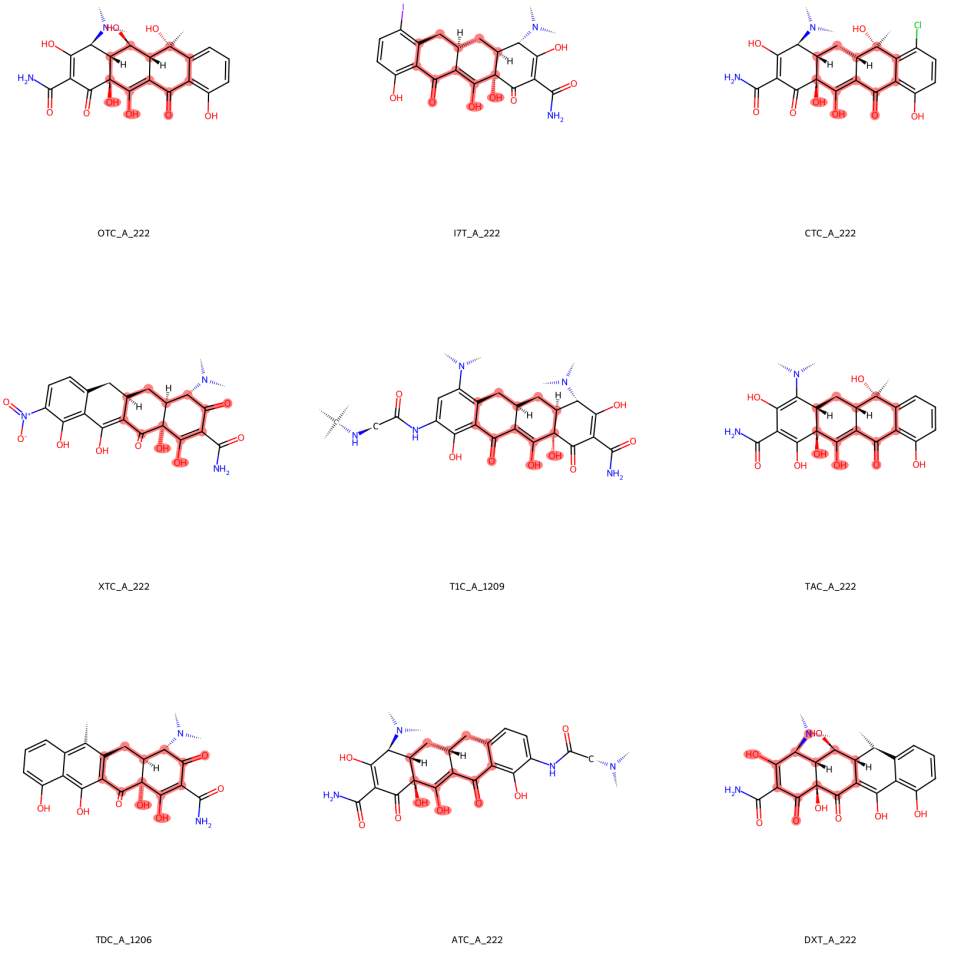


Figure A.7.: Sample of the ligand molecules contained in the ensemble 2VKE.

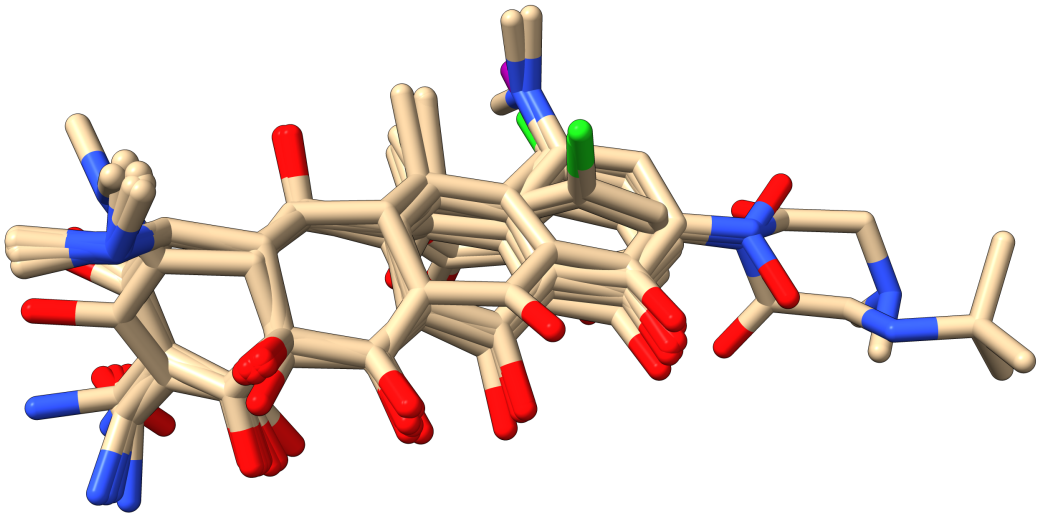


Figure A.8.: Best achieved alignment of the ensemble `2VKE` using CoAler.

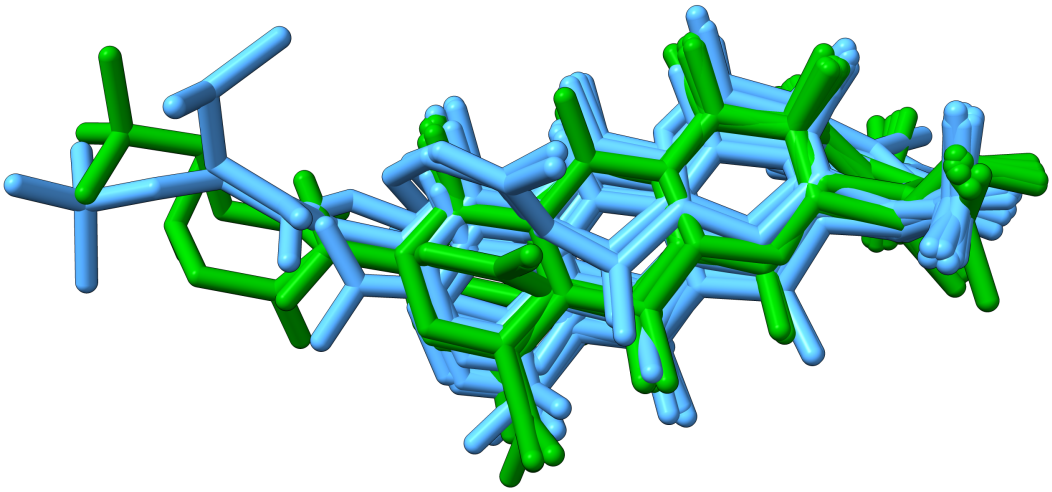


Figure A.9.: Comparison of the best achieved alignment of the ensemble molecules from `2VKE` with the ligand coordinates from the NBSE. Green: CoAler result; Blue: NBSE ensemble

A.4. Ensemble 1D0S

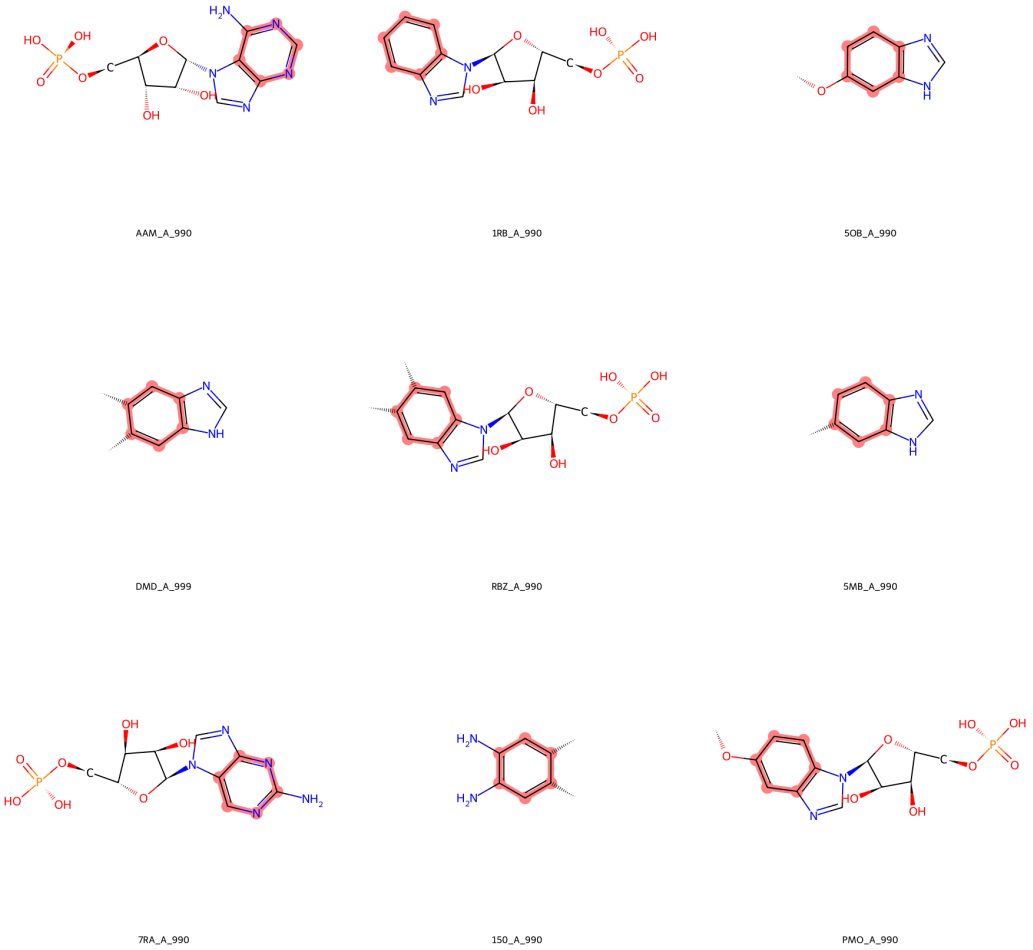


Figure A.10.: Sample of the ligand molecules contained in the ensemble 1D0S.

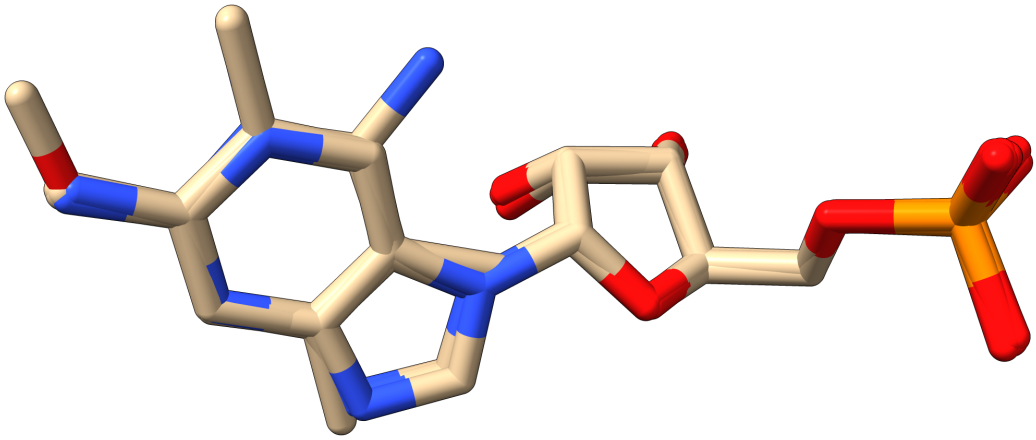


Figure A.11.: Best achieved alignment of the ensemble `1D0S` using CoAler.

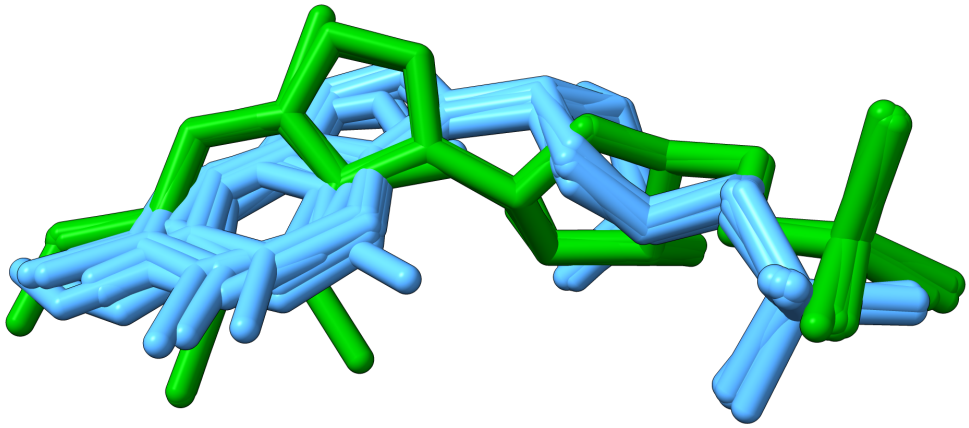


Figure A.12.: Comparison of the best achieved alignment of the ensemble molecules from `1D0S` with the ligand coordinates from the NBSE. Green: CoAler result; Blue: NBSE ensemble

A.5. Ensemble 1ODN

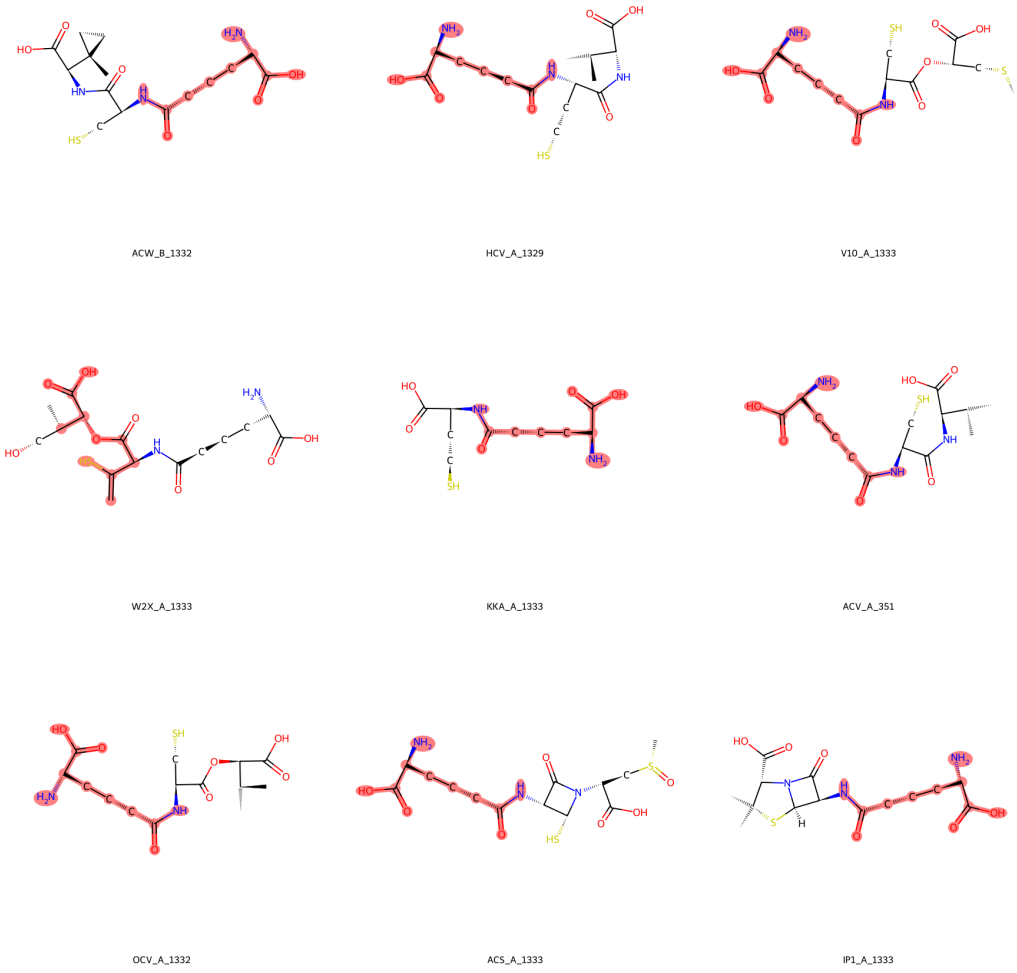


Figure A.13.: Sample of the ligand molecules contained in the ensemble 1ODN.

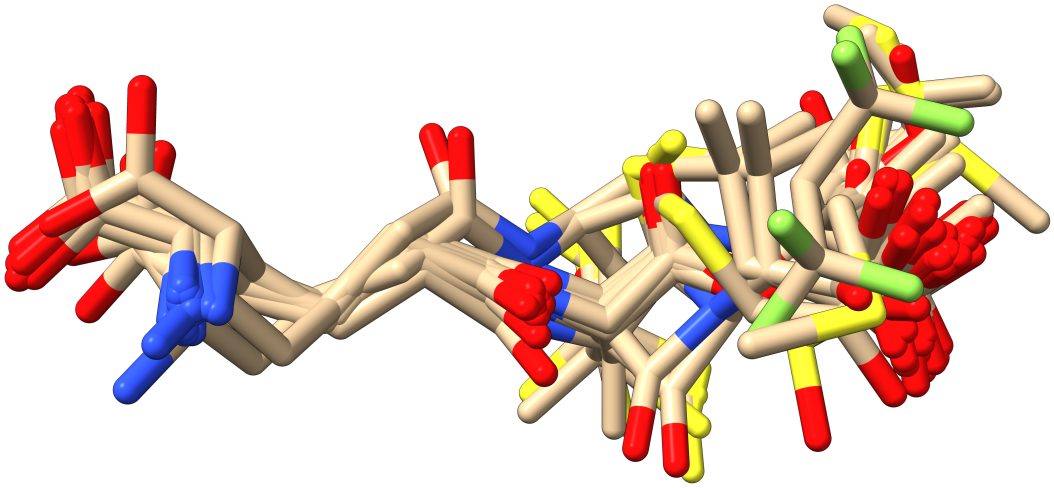


Figure A.14.: Best achieved alignment of the ensemble 1ODN using CoAler.

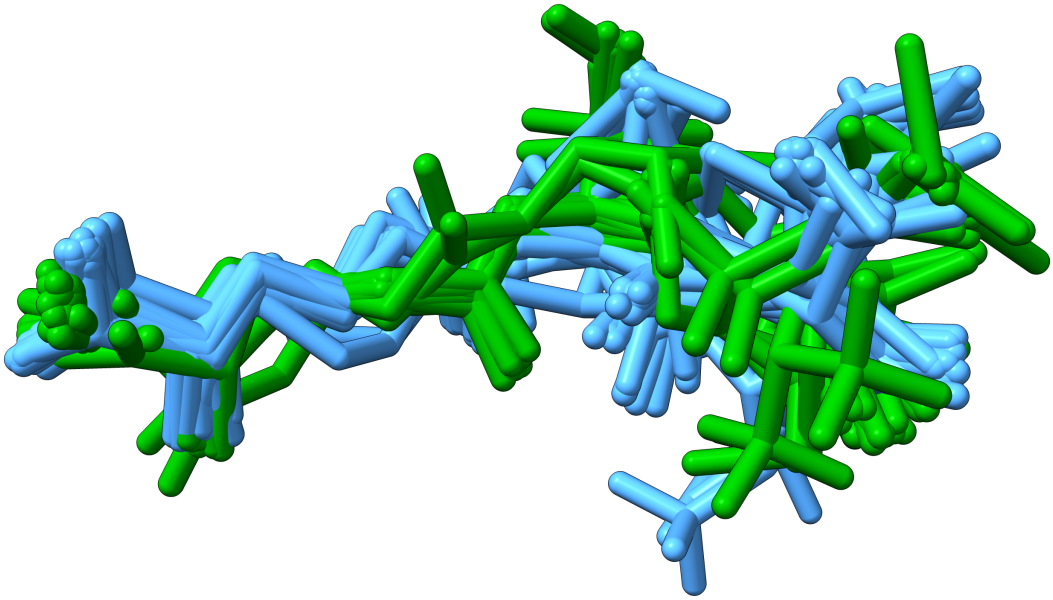


Figure A.15.: Comparison of the best achieved alignment of the ensemble molecules from 1ODN with the ligand coordinates from the NBSE. Green: CoAler result; Blue: NBSE ensemble

A.6. Ensemble 4GFD

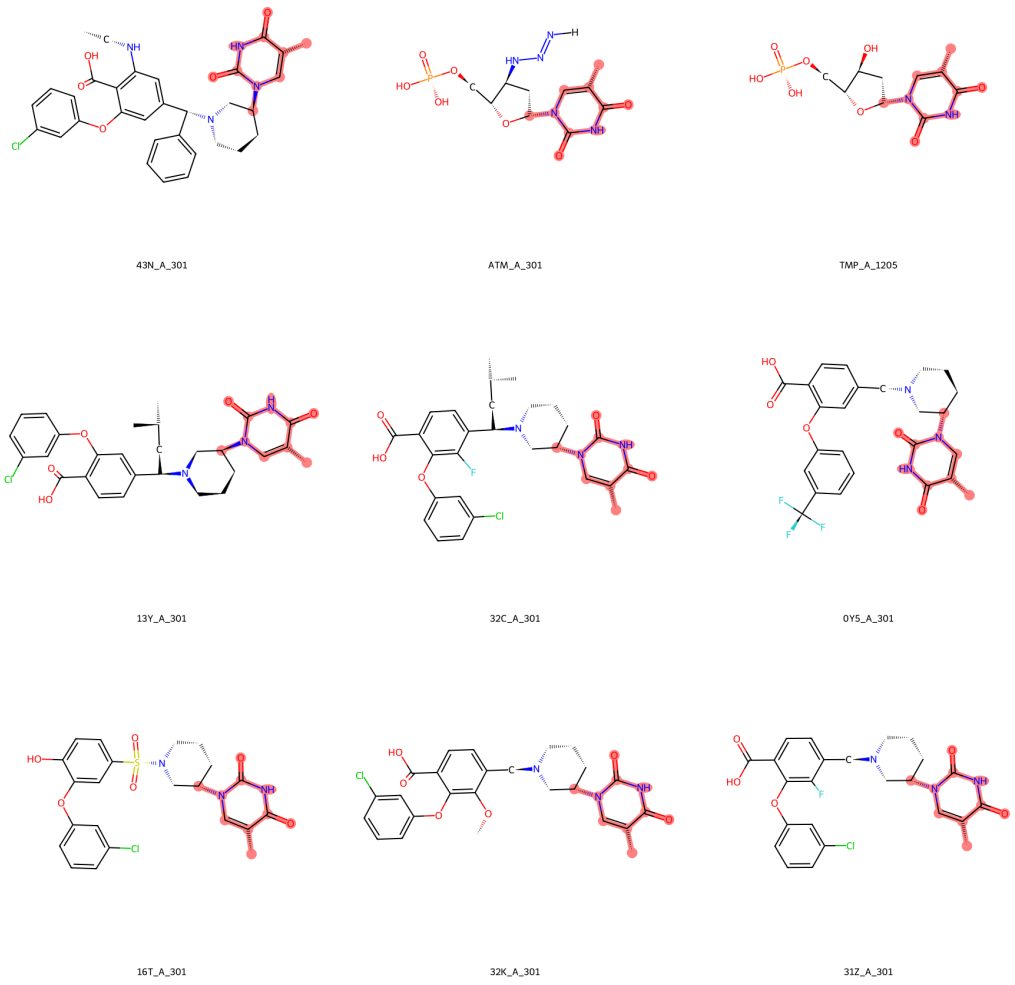


Figure A.16.: Sample of the ligand molecules contained in the ensemble 4GFD.

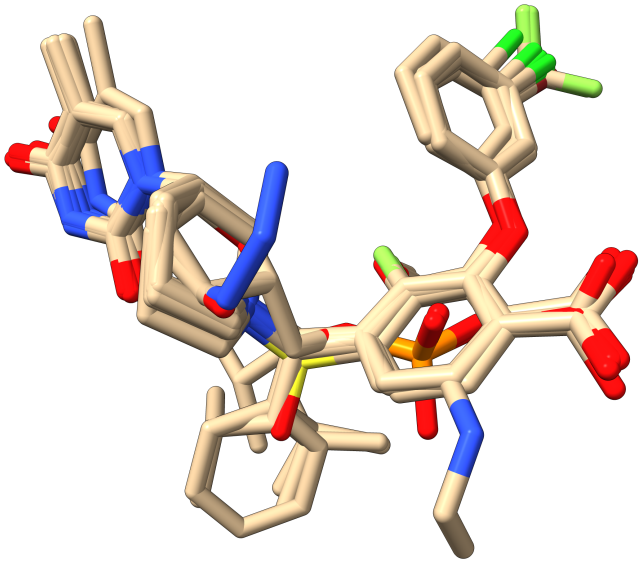


Figure A.17.: Best achieved alignment of the ensemble `4GFD` using CoAler.

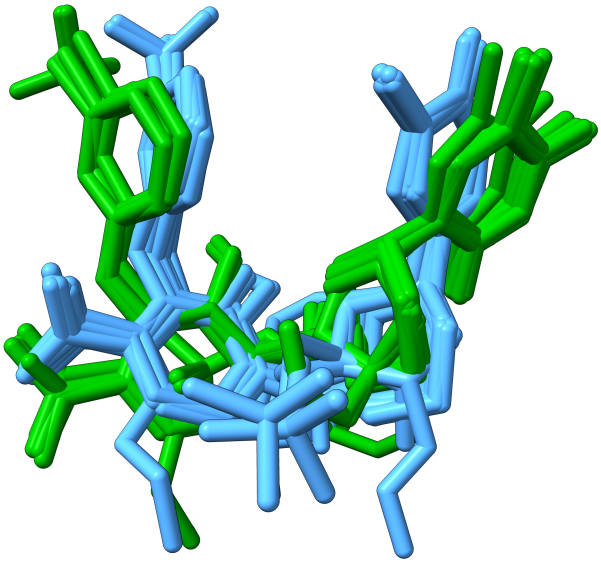


Figure A.18.: Comparison of the best achieved alignment of the ensemble molecules from `4GFD` with the ligand coordinates from the NBSE. Green: CoAler result; Blue: NBSE ensemble

A.7. Ensemble 3QQS

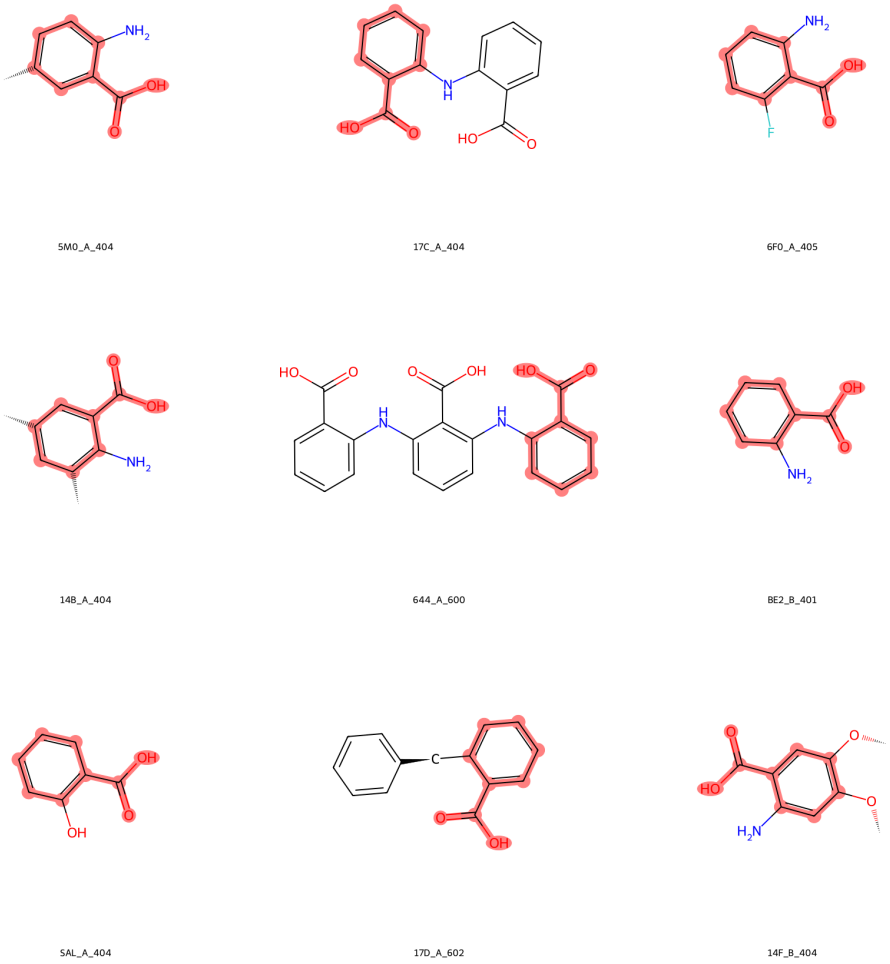


Figure A.19.: Sample of the ligand molecules contained in the ensemble 3QQS.

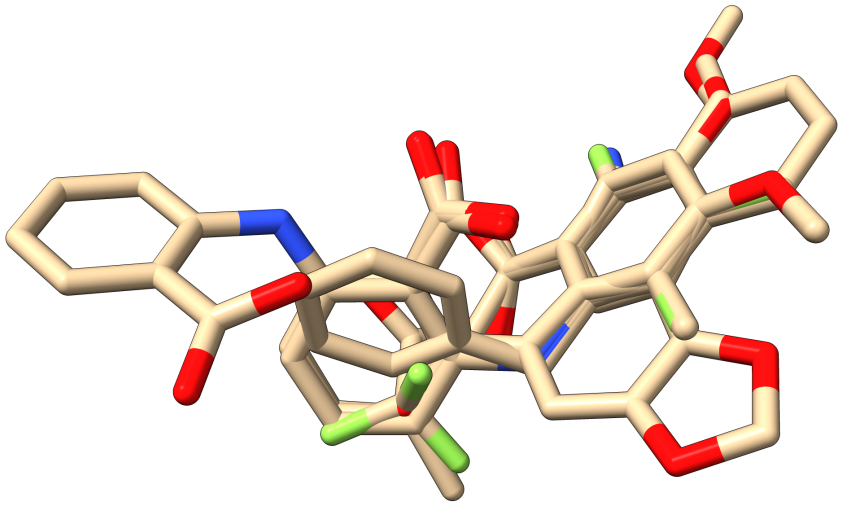


Figure A.20.: Best achieved alignment of the ensemble `3QQS` using CoAler.

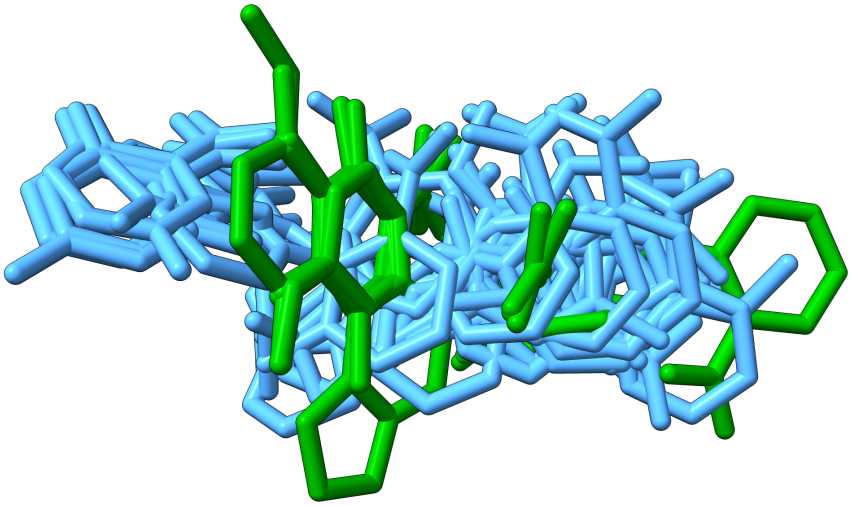


Figure A.21.: Comparison of the best achieved alignment of the ensemble molecules from `3QQS` with the ligand coordinates from the NBSE. Green: CoAler result; Blue: NBSE ensemble

A.8. Ensemble 2WOV

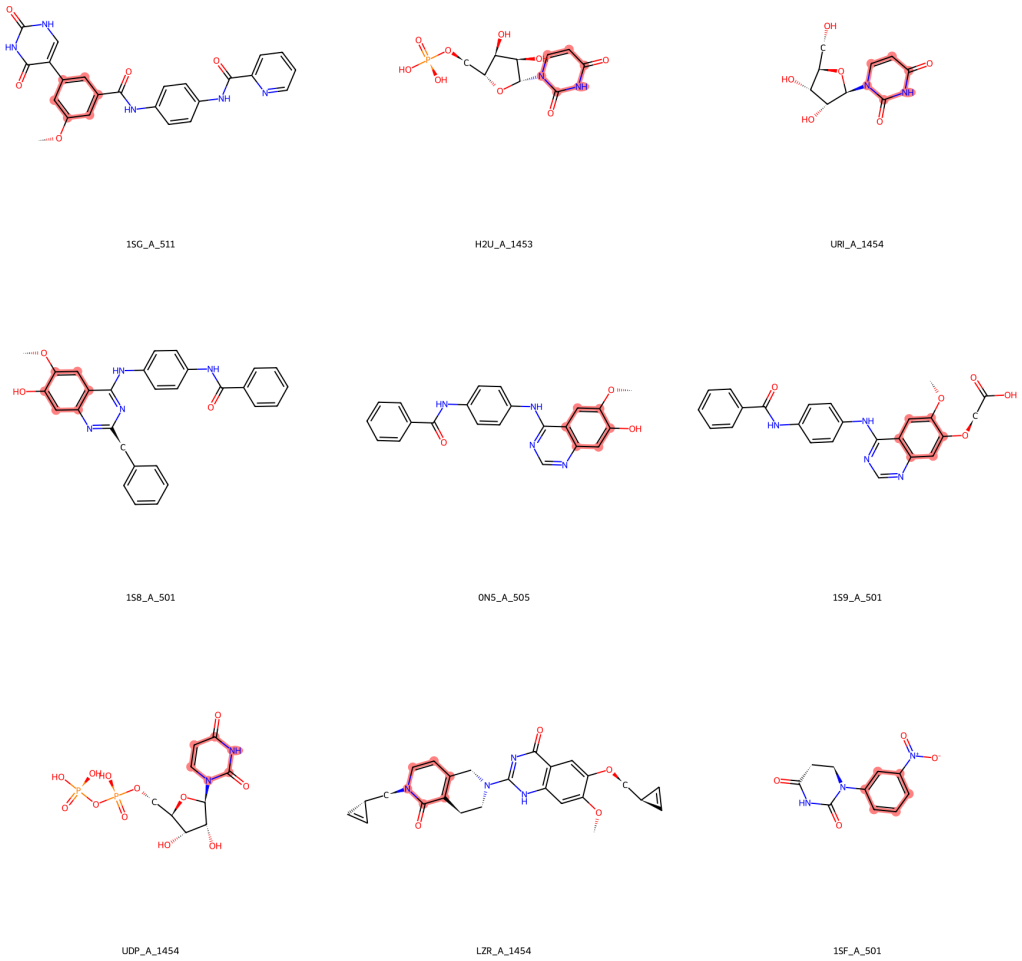


Figure A.22.: Sample of the ligand molecules contained in the ensemble 2WOV.

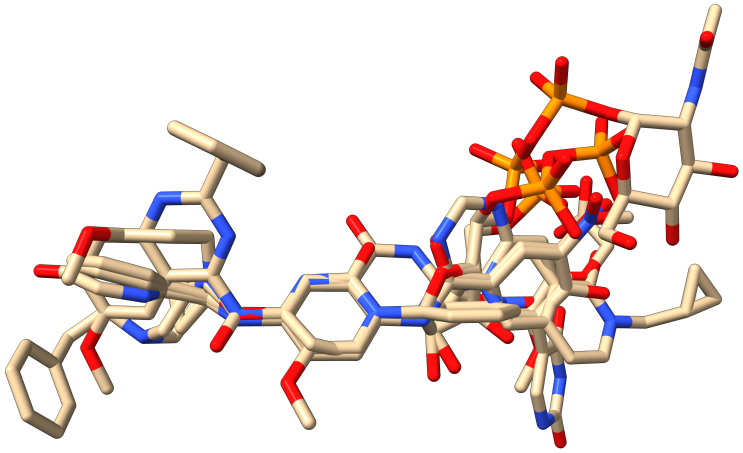


Figure A.23.: Best achieved alignment of the ensemble `2WOV` using CoAler.

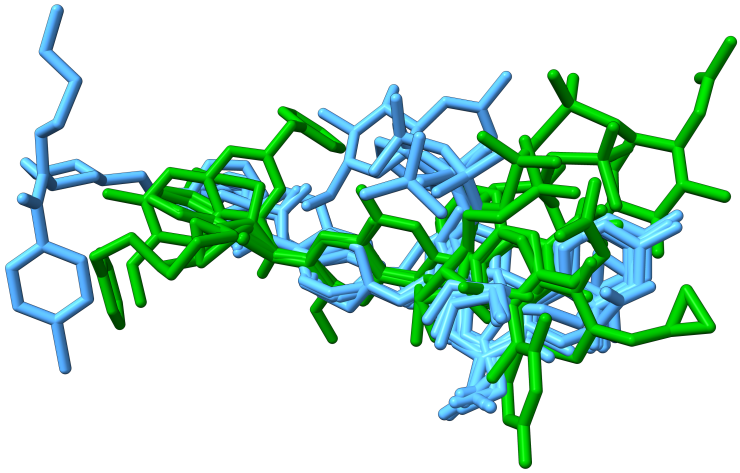


Figure A.24.: Comparison of the best achieved alignment of the ensemble molecules from `2WOV` with the ligand coordinates from the NBSE. Green: CoAler result; Blue: NBSE ensemble

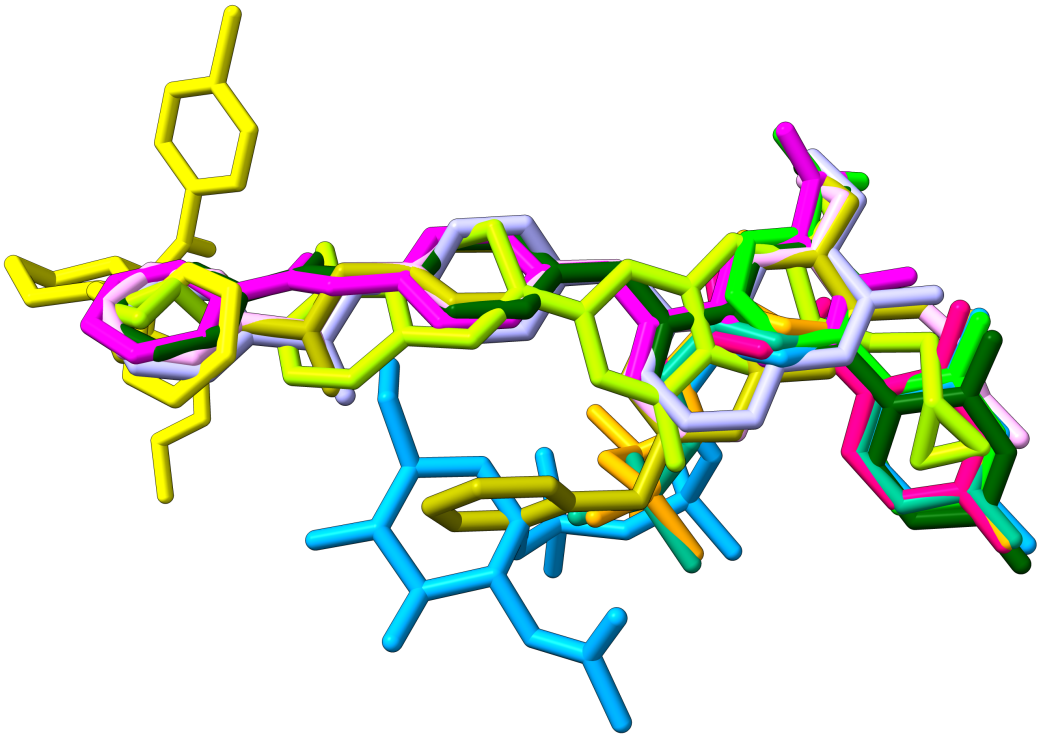


Figure A.25.: Ligand molecules of the NBSE ensemble `2WOV`.

A.9. Ensemble 2HCT

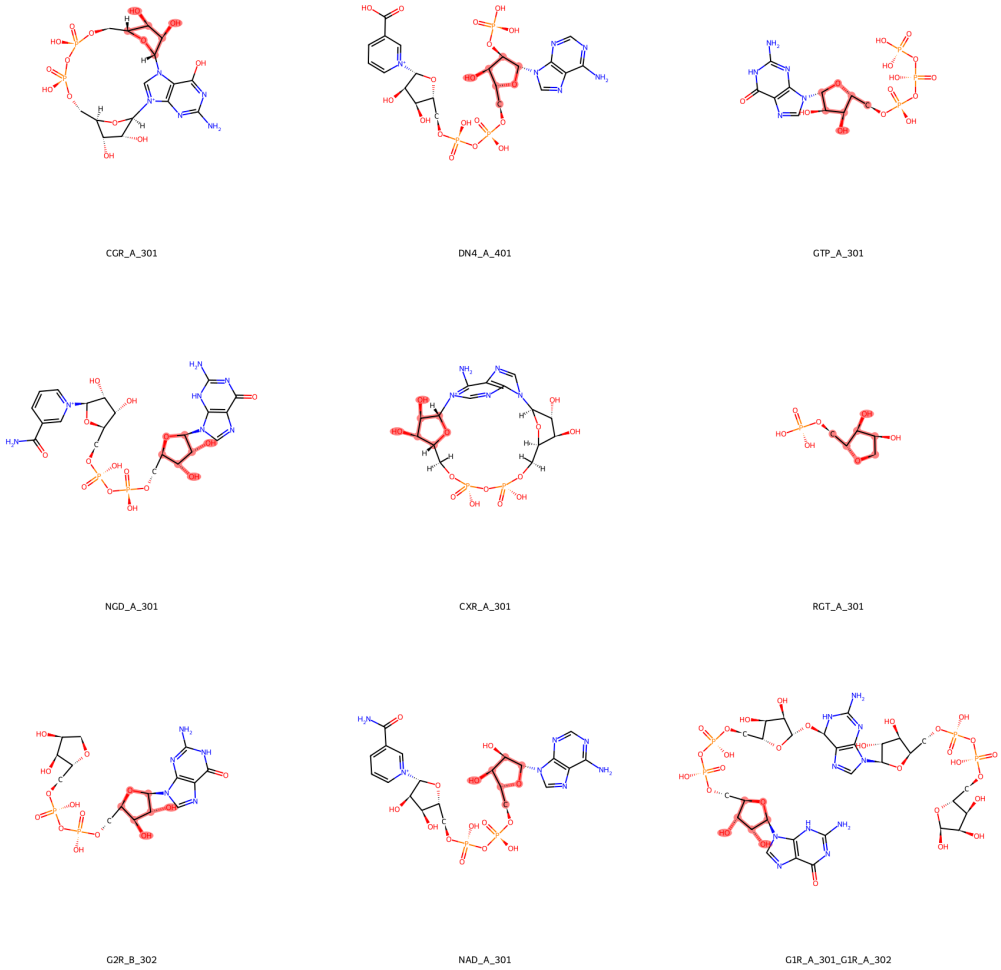


Figure A.26.: Sample of the ligand molecules contained in the ensemble 2HCT.

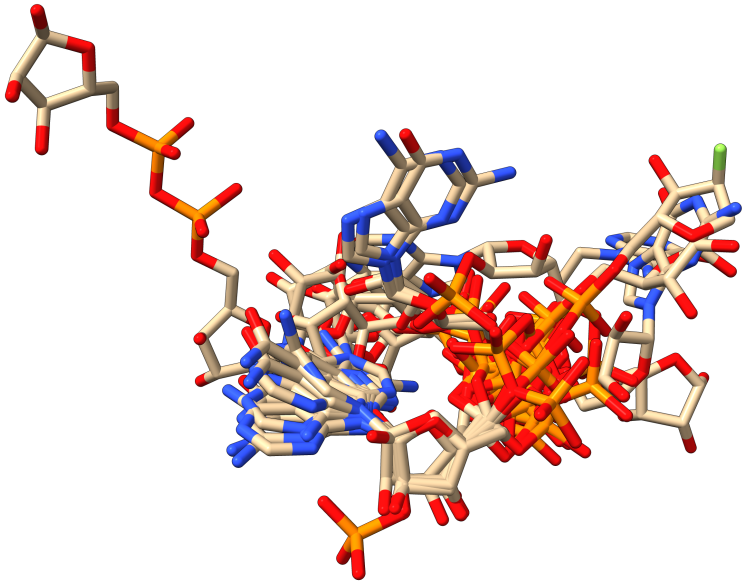


Figure A.27.: Best achieved alignment of the ensemble `2HCT` using CoAler.

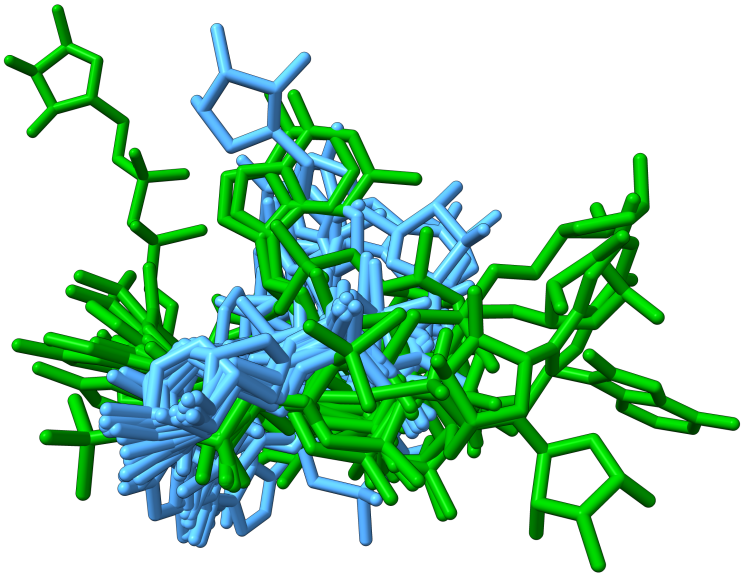


Figure A.28.: Comparison of the best achieved alignment of the ensemble molecules from `2HCT` with the ligand coordinates from the NBSE. Green: CoAler result; Blue: NBSE ensemble

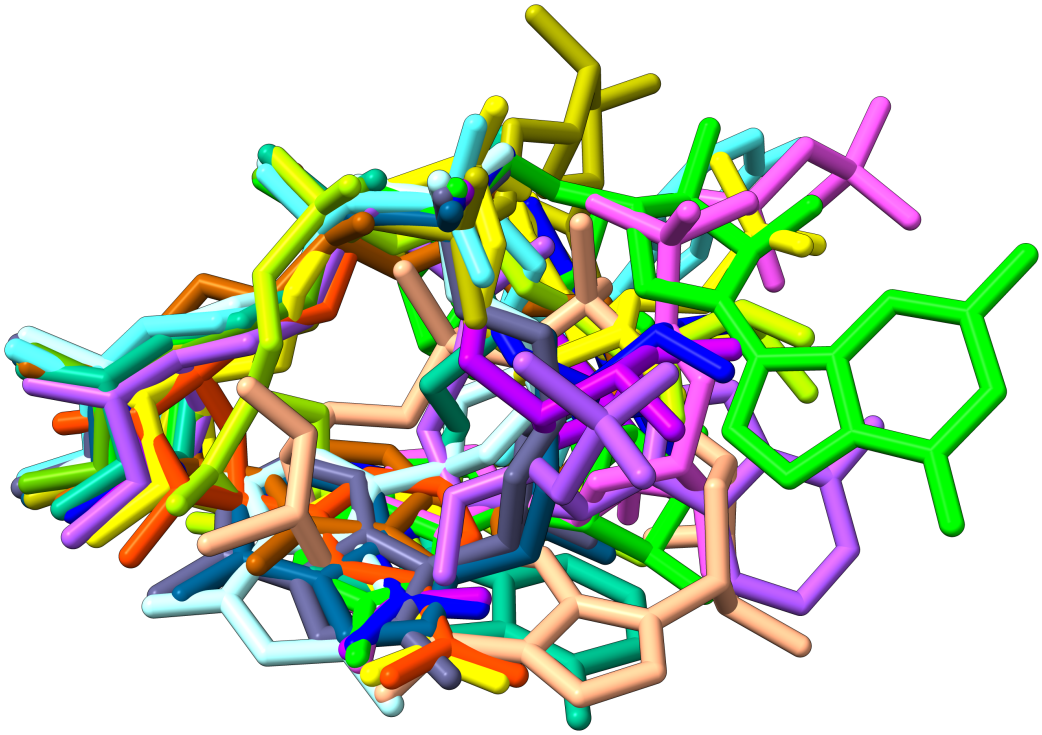


Figure A.29.: Ligand molecules of the NBSE ensemble `2HCT`.