Clustering Molecular Dynamics Trajectories with a Univariate Estimation of Distribution Algorithm

Rodrigo C. Barros*, Christian V. Quevedo*, Renata De Paris* and Márcio P. Basgalupp†
*Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática, Porto Alegre, RS, Brazil
Email: rodrigo.barros@pucrs.br
†Universidade Federal de São Paulo
Instituto de Ciência e Tecnologia, São José dos Campos, SP, Brazil
Email: basgalupp@unifesp.br

***Abstract*—Molecular Dynamics simulations of protein receptors are an emergent tool in rational drug discovery. Nevertheless, employing Molecular Dynamics trajectories in virtual screening of large repositories is a very costly procedure, which ultimately may become unfeasible. Data clustering have been applied in this context with the goal of reducing the overall computational cost in order to make this task feasible. In this paper, we develop a novel estimation of distribution algorithm called Clus-EDA for clustering entire trajectories using structural features from the substrate-binding cavity of the protein receptor. This novel approach is capable of reducing the original trajectory to about 4% of its original size whilst keeping all relevant information for the analysis of receptor-ligand binding. The resulting partition generated by the estimation of distribution algorithm is further validated by analyzing the interactions between 20 ligands and a Fully-Flexible Receptor model containing a 20 *ns* Molecular Dynamics simulation trajectory. Results show that Clus-EDA is capable of outperforming traditional clustering algorithms such as k-means and hierarchical clustering by providing the smallest variance of the free energy of binding within the conformations in each cluster.**

I. Introduction

Proteins are flexible systems, and such a flexibility is key in determining their corresponding functions [1]. Rational drug design is an emerging technology which attempts to map the behavior of a protein and its potential binding to a given drug candidate. It is the interaction between drug candidates (hereby also referred to as *ligands*) and target proteins (*receptors*) in computational simulations that form the basis of rational drug design. The rationale is the following: given a receptor, molecular docking experiments sample a large number of conformations and orientations of a candidate ligand within the protein binding site. The energy provided by the potential binding of ligand and receptor roughly indicates whether the ligand is a potential drug for the given disease [2].

The major problem with molecular docking software nowadays is that they only consider the flexibility of the ligand, whereas the receptor is assumed to be a rigid structure. This is explained due to the large computational cost that is required for also considering the receptor's flexibility. Among the available methodologies to explicitly consider the receptor's flexibility in docking simulations, a possible alternative is to select a series of different conformations derived from a Molecular Dynamics (MD) simulation of the target receptor. MD simulation is a well-known strategy to investigate in detail the atomic dynamic behavior of proteins in aqueous solution. It recognizes subtle internal motions and slow conformational deviations, including bond vibration, chain reorientation, and backbone rearrangements at different timescales [3], [4].

978-1-4799-7492-4/15/$31.00

MD simulations are intrinsically a time-consuming process. The computational cost largely increases when docking experiments are used for the fast screening of virtual libraries against an entire MD ensemble in order to exploit all conformations of the protein receptor [3]. These MD ensembles, hereby called a Fully-Flexible Receptor (FFR) model, typically hold over 10^4 MD structures. For this reason, recent studies on combining docking and MD simulations have developed novel techniques to systematically reduce the number of MD structures without losing essential information, usually employing clustering algorithms for achieving the desired reduction [5]–[7]. By clustering highly-similar MD conformations regarding their substrate-binding cavities, one can extract the most relevant information during the molecular docking experiments, reducing its overall computational cost. Even though clustering is the approach employed in this work, we note that several papers employ learning approaches for the domain of molecular dynamics, with goals as diverse as predicting bioactivities of ligands to target proteins [8], drug classification [9], [10], and free energy of binding prediction [11], [12].

In this paper, we propose a novel clustering algorithm for reducing the number of MD conformations and making it feasible to combine docking and MD simulations. Since data clustering is basically a combinatorial optimization problem, we propose a novel method based on the Estimation of Distribution Algorithms (EDAs), namely Clus-EDA. It is designed to considerably reduce the number of conformations (roughly keeping $\approx 5\%$ of the original MD ensemble) while optimizing a measure of clustering validity. Our research hypothesis is that Clus-EDA is capable of outperforming traditional clustering algorithms in the task of reducing MD conformations. For verifying such a hypothesis, we analyze the partitions generated by Clus-EDA based on features from the binding cavity of an MD simulation regarding the InhA-NADH complex [13], and we compare the provided results with those achieved by k-means [14] and hierarchical agglomerative clustering [15].

This paper is organized as follows. Section II details our novel method for clustering MD simulations based on Estimation of Distribution Algorithms, namely Clus-EDA. Section III describes the methodology adopted for performing the empirical analysis, and the results of the experiments are discussed in Section IV. Section V presents work related to the proposed approach, and we end this paper with our conclusions and future work directions in Section VI.

II. CLUS-EDA

Clus-EDA is an Estimation of Distribution Algorithm (EDA) for data clustering. EDAs are a particular class of evolutionary algorithms that explore the space of candidate solutions by building and sampling explicit probabilistic models of promising solutions [16]. The main characteristic of EDAs is the absence of random operators during evolution. Instead of employing these nature-inspired genetic operators, the future populations are generated by learning and simulating a probability distribution based on the individuals that are selected from the current population [17].

Clus-EDA samples solutions encoded by a probabilistic model, which is responsible for determining whether each object in a dataset is a *medoid* or not. A medoid is a cluster representative, and the number of medoids indicate the number of clusters found by Clus-EDA. Each individual in Clus-EDA is a binary vector of size n, where n is the number of objects in the dataset – in our case, the number of conformations of an MD simulation. Since we are working with a FFR model that contains a 20 ns MD simulation trajectory, the number of objects in the trajectory dataset is 20,000. Figure 1 depicts the encoding scheme in Clus-EDA.

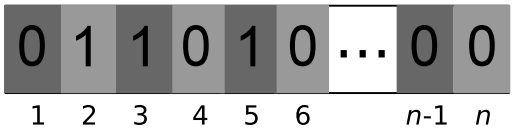


Fig. 1. Individual representation in Clus-EDA. Each gene corresponds to an object, and the encoding indicates whether the object is a medoid (1) or not (0).

Clus-EDA is a univariate EDA also regarded as a univariate marginal distribution algorithm (UMDA) [18]. It employs a binary probability vector $p = (p_1, p_2, ..., p_n)$ as its probabilistic model, where p_i denotes the probability of object $\mathbf{x}_i$ to be a medoid. To learn the probability vector, each p_i is set to the proportion of 1s in the population of selected individuals. Clus-EDA follows the pseudocode described in Algorithm 1.

Algorithm 1 Clus-EDA high-level pseudocode.

```
initialize probability vector p
sample p to generate initial population P
while (not done) do
    select population of promising solutions S from P
    update probability vector p with S
    sample p to generate new candidate solutions N
    erase P and incorporate N into P
end while
```

Since our goal is to considerably reduce the size of an MD simulation, we initialize the probability vector p with probability of 0.05 for every single position. In other words, the initial probability for each object to be a medoid is 5%. In each generation of Clus-EDA, we employ the truncation method for selection, which chooses 50% of the fittest individuals of that particular generation to update the probabilistic model. Once the model is updated, Clus-EDA samples the probability vector p to generate an entire novel population of individuals that fully replace the previous generation. The iteration continues until a maximum number of generations is achieved.

A. Decoding Individuals into Partitions

Clus-EDA is an evolutionary hard partitional clustering algorithm. Given a set of n objects to be clustered, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, a hard partition is a collection of k non-overlapping clusters $\mathbf{P} = \{\mathbf{C}_1, \mathbf{C}_2 ..., \mathbf{C}_k\}$ such that:

- $\mathbf{C}_1 \cup \mathbf{C}_2 \cup ... \cup \mathbf{C}_k = \mathbf{X}$
- $\forall i,\ \mathbf{C}_i \neq \emptyset$
- $\mathbf{C}_i \cap \mathbf{C}_j = \emptyset$ for $i \neq j$

For decoding the individuals into partitions, the first step is to identify which objects are defined as medoids. Note that the number of clusters is variable since it is constantly updated according to the EDA's probabilistic model. For each non-medoid object $\mathbf{x}_i$, Clus-EDA computes the Euclidean distance between $\mathbf{x}_i$ and every single medoid, and finally assigns $\mathbf{x}_i$ to the cluster represented by its closest medoid.

The binary encoding adopted by Clus-EDA has several advantages over other typical encodings in evolutionary clustering problems. For instance, let us consider the case of the integer encoding in which each gene (object) has a value over the alphabet $1, 2, ..., k$. Such an encoding is naturally redundant (1-to-many), since there are $k!$ different genotypes that represent the same solution [19]. Furthermore, it assumes the number of clusters k is previously known, which is often not the case in real world applications, such as the one that is being approached in this paper.

B. Fitness Function

For evaluating how fit an individual is in Clus-EDA, we make use of an efficient clustering validity criterion, namely the simplified silhouette width criterion ($SSWC$) [20]. It is an efficient implementation of the well-known silhouette width criterion (SWC) [21], which is given by:

$$SWC = \frac{1}{n}\sum_{i=1}^{n} s(i) \tag{1}$$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2}$$

where $a(i)$ is the average dissimilarity between the i^{th} object and its cluster, and $b(i)$ is the average dissimilarity between the i^{th} object and the nearest neighbor cluster. For singletons (clusters with a single object $\mathbf{x}_j$), it is assumed by convention that $s(j) = 0$.

The difference between $SSWC$ and the original SWC is in how $a(i)$ and $b(i)$ are computed. Whereas SWC computes

the average dissimilarity by employing all objects belonging to the corresponding cluster (complexity of $O(n^2)$), $SSWC$ computes the average dissimilarity by using the cluster prototypes instead (complexity of $O(n)$). Clus-EDA employs the $SSWC$ as its fitness function, computing the average dissimilarity based on distances to the corresponding medoids.

Besides the possibility of employing $SSWC$ as a single-objective fitness function, Clus-EDA also allows the search for a trade-off between performance (as given by $SSWC$) and complexity (given by the number of clusters). The idea is to penalize solutions with too many clusters. Since $SSWC \in [-1, +1]$ is a measure that should be maximized, we decrease it by a factor of $w \times k$, where w is a user-defined parameter and k is the number of clusters.

III. Methodology

In this section, we present the Molecular Dynamics dataset in detail, and the experimental methodology that was employed for comparing Clus-EDA with k-means [14] and hierarchical agglomerative clustering [15] in the task of clustering MD trajectories.

A. MD Trajectory Data

The MD trajectory data comprises conformational features from the substrate-binding cavity of an MD simulation considering the InhA-NADH complex from *Mycobacterium tuberculosis* (PDB ID: 1ENY) as described in [13]. Data for the MD ensemble were collected at every 1 ps, resulting in a set of 20,000 instantaneous receptor conformations. The structural properties that were extracted from the substrate-binding cavity of every MD conformation are:

1) the pairwise RMSD distances between binding cavity atoms (in Å);
2) the volume (in Å^3); and
3) the number of heavy atoms of each residue that belongs to the substrate-binding cavity of the enzyme (PDB ID: 1BVR) [22].

The first property (pairwise RMSD distances) was evaluated by monitoring the differences between the backbone atoms (N, Cα, C, and O) within the substrate-binding cavity from the first structure against the conformation under comparison. The RMSD values were calculated using the *ptraj* module from AmberTools12 [23]. The second and third properties were collected using the CASTp software (Computed Atlas of Surface Topography of proteins) [24]. CASTp provides an online resource for locating, delineating, and measuring concave surface regions on three-dimensional structures of proteins based on the solvent-accessible surface area model [25] and the molecular surface model [26]. The measurement of volume for every MD conformation was obtained by considering the residues that encloses the cavity of the InhA substrate analog from the 1BVR structure [22], which contains the largest number of atoms. We collected the number of heavy atoms from the 10 main residues that lie in the substrate-binding cavity of the enzyme (Figure 2).

Our final objective is to cluster different behaviors found within the substrate-binding cavity along an MD simulation, drastically reducing the number of snapshots in the Fully-Flexible receptor model while keeping all relevant information from the original number of snapshots.

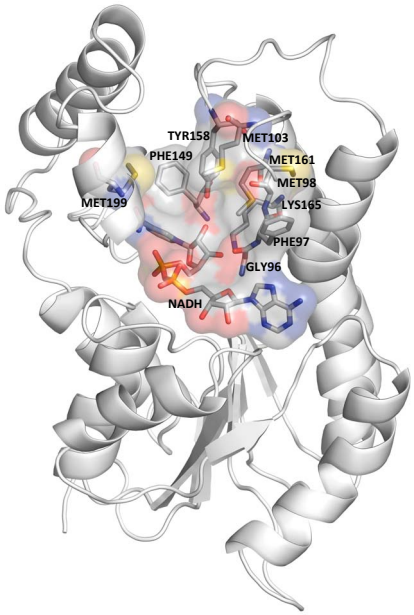


Fig. 2. Substrate-binding cavity of the InhA enzyme from *Mycobacterium tuberculosis* (PDB ID: 1BVR) identified by the CASTp software. The protein receptor, with secondary structures represented by ribbons, is colored grey. The residues are represented in sticks within the molecular surface, which represents the substrate-binding cavity. Image generated by PyMol [27].

B. Baseline Algorithms

In order to verify the effectiveness of Clus-EDA in reducing the number of MD conformations and making it feasible to combine docking and MD simulations, we analyze the obtained results with those provided by well-known clustering algorithms, namely k-means [14] and UPGMA [15], the latter being a hierarchical agglomerative clustering algorithm.

k-means is a well-known partitional method that locally optimizes the average squared distance between objects and their nearest cluster center (centroid). It randomly generates k centroids, and refines them throughout several iterations, where it computes the distance of every object to the k centroids in order to determine the cluster memberships [14]. To make the resulting k clusters as compact and separate as possible, the k-means algorithm minimizes the sum of squared errors (J) between every object $\mathbf{x}_i$ that belongs to a given cluster C_j and its centroid c_j, for all k clusters, as follows:

$$J = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathbf{C}_j} dist(\mathbf{x}_i, c_j)^2 \tag{3}$$

where $dist(a, b)$ is the Euclidean distance between a and b.

Unlike partitional clustering methods such as k-means, hierarchical clustering algorithms seek to cluster data into levels of a hierarchical structure, such as a "tree" of clusters [28]. They can be divided into two basic approaches, namely agglomerative and divisive clustering. Agglomerative hierarchical clustering employs the bottom-up strategy, starting with

each object as a singleton (cluster with a single object) and iteratively merging the closest pair of clusters until all the objects lie within a single cluster or the maximum number of clusters is reached. Divisive hierarchical clustering, on the other hand, employs the top-down strategy, starting with all objects within the same cluster and splitting a cluster into smaller clusters until each object becomes a singleton cluster or a termination condition holds. We decided to employ an agglomerative algorithm since the divisive methods cannot efficiently handle large datasets. To measure the proximity between two objects in different clusters, the agglomerative algorithms employ distinct strategies, each one defining its method's name: single linkage, complete linkage, median, centroid, group average, and Ward's.

In UPGMA [15], which is a group average based agglomerative algorithm, the distance between two clusters is defined as the average pairwise proximity among all pairs of objects in different clusters. UPGMA takes into account the number of objects in each cluster, as follows:

$$UPGMA(\mathbf{C}_i, \mathbf{C}_j) = \frac{1}{|\mathbf{C}_i||\mathbf{C}_j|} \sum_{\mathbf{x}\in\mathbf{C}_i} \sum_{\mathbf{y}\in\mathbf{C}_j} dist(\mathbf{x}, \mathbf{y}) \quad (4)$$

Both k-means and UPGMA require the number of clusters, k, to be set by the end-user.

C. Validation of Data Partitions

After Clus-EDA generates its final near-optimal partition, we perform exhaustive docking experiments on AutoDock4.2 [29] in order to search for evidence that validates the quality of such a partition. These experiments are conducted between 20,000 snapshots (FFR model) and 20 different compounds, which are extracted from 20 InhA structures deposited at PDB [30]. Figure 3 shows the 3D structures of the 20 compounds and the rotatable bounds defined in the docking experiments.

In order to preserve the reaction mechanism between ligands and the target protein, NADH should be treated as a coenzyme. Hence, for experiments with ligands, the coenzyme was considered as part of the protein receptor structure. Conversely, we removed the NADH coenzyme from each snapshot of the FFR model when we performed the experiments with adducts (INH-NAD and PTH-NAD), since they already have the coenzyme as part of their structures.

Once the exhaustive molecular docking experiments are performed, we analyze the agreement among snapshots that lie within the same cluster. For each snapshot, the docking experiments produce a value of the free energy of binding (FEB) of that particular receptor conformation with the ligand at hand. Hence, the rationale for validating a clustering partition is the following: a good partition should have snapshot clusters with low FEB variance regarding the FEB achieved by the cluster representative (medoid). Such a measure is hereby referred to as FEB_{var}, and is given by:

$$FEB_{var} = \frac{\sum_{j=1}^{k} \sum_{\mathbf{x}_i\in\mathbf{C}_j} \left(FEB_{\mathbf{x}_i} - FEB_{\mu_j}\right)^2}{k - s} \quad (5)$$

where $FEB_{\mathbf{x}_i}$ is the free energy of binding between the snapshot represented by object $\mathbf{x}_i$ and the corresponding ligand, and FEB_{μ_j} is the free energy of binding between the representative snapshot (cluster medoid) and the corresponding ligand. Finally, s is the number of singletons, which should be excluded from the computation since singletons have a single object (the cluster medoid) and thus a null deviation. If singletons were taken into account, partitions with many singletons would artificially reduce the final FEB_{var} value.

The lower the FEB_{var}, the greater the agreement among snapshots that were clustered together, which means that all snapshots that lie within the same cluster may be discarded except for their representative, which is now the only required snapshot to be used during the molecular docking experiments.

Besides the biological validation through the analysis of FEB_{var} values, we also analyze the $SSWC$ values of the resulting partitions generated by Clus-EDA and the baseline algorithms k-means and UPGMA.

D. Parameters

Clus-EDA is an evolutionary algorithm, and as such it requires two main parameters to be set *a priori*: number of individuals and number of generations. In the experiments, we set those values to 500 individuals and 500 generations, and no effort towards parameter tuning was performed whatsoever. The selection mechanism in Clus-EDA is by truncation, which defines that 50% of the individuals in the current generation are used to update the probabilistic model.

For the fitness function that penalizes $SSWC$'s value according to the number of clusters in the partition, we set the penalizing factor w to 0.00005, which means that a partition with 5,000 clusters penalizes $SSWC$ in 0.25, whereas the trivial solution (a partition with 20,000 clusters) decreases the value of $SSWC$ in 1 unity.

Finally, we set the probability of generating medoids in the initial population to 5%, considering that we could achieve a considerable reduction in computational effort by reducing the MD simulation to around 1,000 receptor snapshots (5% of the total size of the trajectory, which is 20,000 snapshots). Nevertheless, we are aware that the initial probability of 5% of generating medoids will be iteratively updated by Clus-EDA, and no further constraint on k is imposed here.

E. Statistical Analysis

In order to provide some reassurance about the validity and non-randomness of the results, we validate our novel algorithm by presenting results of statistical tests that follow the approach proposed by Demšar [32].

In a nutshell, this approach seeks to compare multiple algorithms on multiple datasets, and it is based on the use of the Friedman test with a corresponding post-hoc test. The Friedman test is a non-parametric counterpart of the ANOVA test, as follows. Let R_i^j be the rank of the j^{th} of k algorithms on the i^{th} of N data sets. The Friedman test compares the average ranks of algorithms, $R_j = \frac{1}{N}\sum_i R_i^j$. The Friedman statistic is given by:

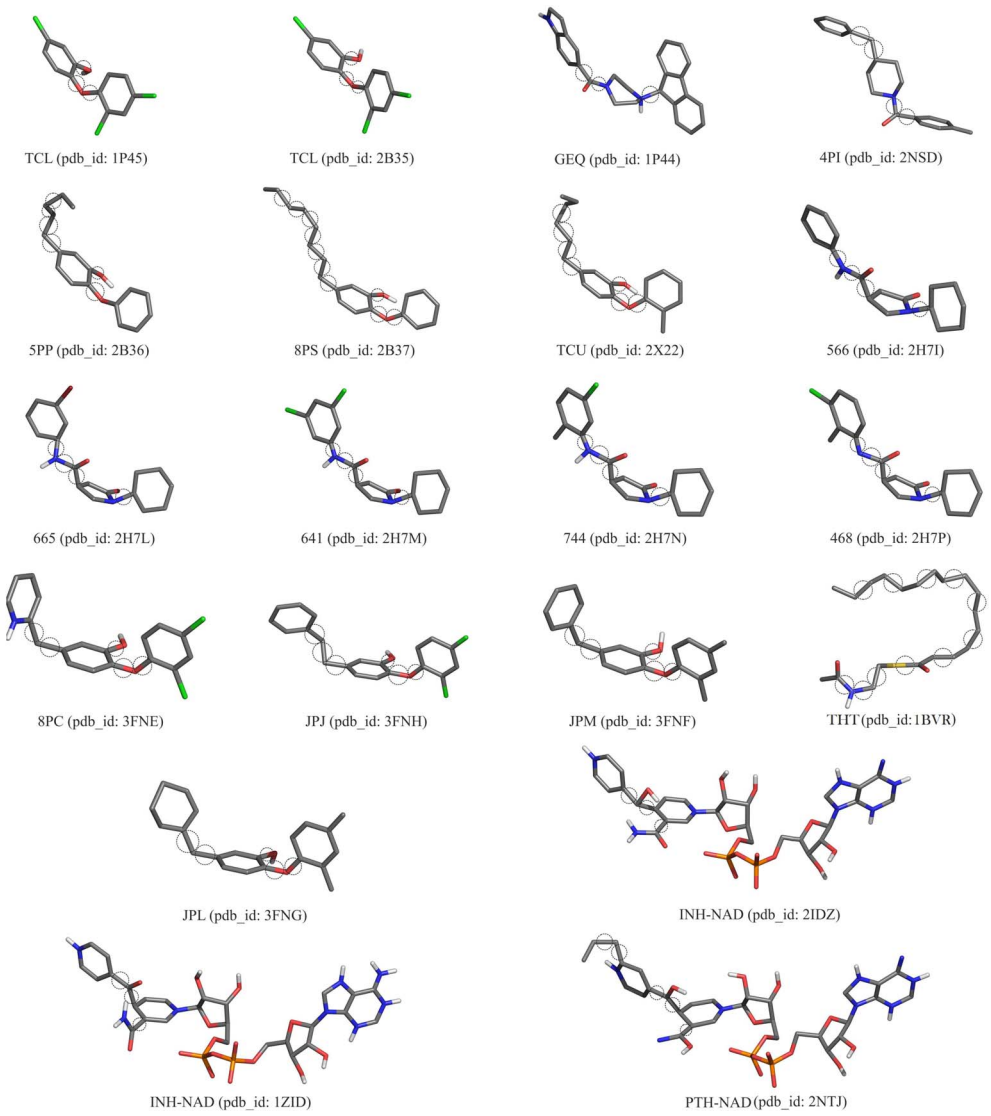


Fig. 3. Stick representation of the 3D structures of the 20 ligands used in docking experiments. Each ligand, with its structures colored by atom type, is identified by their name and its corresponding PDB ID. The dashed circle represents the rotatable bounds selected by AutoDockTools 1.5.6 [31].

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{6}$$

is distributed according to χ_F^2 with $k-1$ degrees of freedom, when N and k are big enough.

Iman and Davenport [33] prove that Friedman's χ_F^2 is undesirably conservative, and thus they derive an adjusted statistic:

$$F_f = \frac{(N-1) \times \chi_F^2}{N \times (k-1) - \chi_F^2} \tag{7}$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom.

If the null hypothesis of similar performances is rejected, then we proceed with the Nemenyi post-hoc test for pairwise comparisons. The performance of two classifiers is significantly different if their corresponding average ranks differ by at least the critical difference:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{8}$$

where critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$.

IV. Results

We executed Clus-EDA, k-means, and UPGMA in the MD trajectory dataset described in Section III-A. Since Clus-EDA is a stochastic approach, we execute it 10 times by varying the seed of the random number's generator. Both k-means and UPGMA have a single parameter, which is the number of clusters k, and since they are deterministic approaches, they are executed a single time. Instead of averaging the results from the multiple executions of Clus-EDA, we present the results

from the best and the worst executions according to the fitness function's values.

The results reported in this section for k-means and UPGMA are regarding the FEB_{var} and $SSWC$ values collected from the partitions with the same k value than the one generated by Clus-EDA, since FEB_{var} tends to monotonically decrease as the number of clusters increase. Moreover, we executed two versions of Clus-EDA. The first version, hereinafter simply referred to as "Clus-EDA", employs the $SSWC$ value as its fitness function. The second version, namely Clus-EDA$_{wf}$, employs the weighted formula that penalizes according to the number of clusters – $SSWC$ is decreased by a factor of $w \times k$.

Table I shows the FEB_{var} results provided by the best executions of Clus-EDA and Clus-EDA$_{wf}$. To be precise, the partitions found by Clus-EDA and Clus-EDA$_{wf}$ were further evaluated in exhaustive molecular docking experiments using all the 20,000 snapshots and one ligand at a time, and for each ligand it was computed the FEB_{var} for the corresponding partition. Recall that FEB_{var} is an average of the within-cluster FEB dispersion. The best execution of Clus-EDA provided a partition with 5060 clusters, whereas the best execution of Clus-EDA$_{wf}$ generated a partition with 779 clusters. The best (lowest) results are highlighted in boldface. Note that Clus-EDA outperformed k-means and UPGMA in 16 out of the 20 ligands that were used during the experiments. Clus-EDA$_{wf}$, in turn, also outperformed k-means and UPGMA in 16 out of the 20 ligands, though generating a much smaller number of clusters (779), considering it penalizes for large values of k.

TABLE I. FEB_{var} VALUES OF THE PARTITIONS PROVIDED BY THE BEST EXECUTIONS OF CLUS-EDA AND CLUS-EDA$_{wf}$.

		Clus-EDA ($k = 5060$)			Clus-EDA$_{wf}$ ($k = 779$)		
Ligand	Protein	Clus-EDA	UPGMA	k-means	Clus-EDA$_{wf}$	UPGMA	k-means
TCL	1P45	0.23	0.24	0.23	**0.30**	0.34	0.33
TCL	2B35	**0.33**	0.35	0.36	**0.46**	0.50	0.48
665	2H7L	**0.31**	0.32	0.32	**0.42**	0.43	0.47
566	2H7I	0.25	0.25	0.26	**0.33**	0.35	0.39
8PC	3FNE	**0.37**	0.41	0.40	**0.48**	0.53	0.55
JPJ	3FNH	**0.41**	0.42	0.42	**0.52**	0.56	0.59
JPL	3FNG	**0.42**	0.43	0.43	**0.54**	0.60	0.63
JPM	3FNF	**0.49**	0.51	0.51	**0.63**	0.67	0.73
468	2H7P	**0.36**	0.38	0.38	**0.46**	0.52	0.53
641	2H7M	**0.33**	0.35	0.34	**0.42**	0.45	0.50
744	2H7N	**0.33**	0.35	0.35	**0.42**	0.46	0.51
INH-NAD	1ZID	0.84	**0.80**	0.86	1.16	**0.91**	1.19
5PP	2B36	**0.28**	0.29	0.29	**0.37**	0.44	0.39
8PS	2B37	**0.45**	0.46	0.47	**0.58**	0.61	0.63
TCU	2X22	**0.31**	0.33	0.32	**0.41**	0.47	0.45
PTH-NAD	2NTJ	**1.06**	1.07	1.12	1.55	**1.24**	1.49
THT	1BVR	**0.38**	0.39	0.39	**0.50**	0.530	0.54
4PI	2NSD	**0.40**	0.43	0.44	**0.53**	0.61	0.61
GEQ	1P44	1.98	1.98	2.08	2.47	**2.04**	2.97
INH-NAD	2IDZ	**0.85**	0.86	0.87	1.20	**0.97**	1.21
Number of Wins:		**16**	1	0	**16**	4	0
Average Rank:		**1.13**	2.30	2.58	**1.25**	2.03	2.73

Table II shows a similar picture than Table I, but now presenting the results of the worst executions of Clus-EDA and Clus-EDA$_{wf}$. In this scenario, Clus-EDA once again outperformed k-means and UPGMA in 16 out of the 20 ligands, which was also the case of Clus-EDA$_{wf}$. Note that no significant difference was noticed regarding the worst and best executions of both versions of Clus-EDA, indicating a stable performance throughout executions.

TABLE II. FEB_{var} VALUES OF THE PARTITIONS PROVIDED BY THE WORST EXECUTIONS OF CLUS-EDA AND CLUS-EDA$_{wf}$.

		Clus-EDA ($k = 5174$)			Clus-EDA$_{wf}$ ($k = 817$)		
Ligand	Protein	Clus-EDA	UPGMA	k-means	Clus-EDA$_{wf}$	UPGMA	k-means
TCL	1P45	**0.22**	0.24	0.24	**0.28**	0.34	0.33
TCL	2B35	**0.33**	0.34	0.35	**0.44**	0.51	0.49
665	2H7L	**0.31**	0.32	0.33	**0.39**	0.44	0.48
566	2H7I	0.25	0.25	0.26	**0.33**	0.35	0.37
8PC	3FNE	**0.38**	0.41	0.39	**0.49**	0.53	0.55
JPJ	3FNH	**0.40**	0.41	0.43	**0.52**	0.58	0.58
JPL	3FNG	**0.41**	0.43	0.43	**0.55**	0.59	0.60
JPM	3FNF	**0.49**	0.51	0.50	**0.63**	0.67	0.69
468	2H7P	**0.35**	0.38	0.38	**0.44**	0.52	0.54
641	2H7M	**0.33**	0.35	0.35	**0.43**	0.46	0.48
744	2H7N	**0.33**	0.35	0.35	**0.42**	0.46	0.48
INH-NAD	1ZID	0.82	**0.80**	0.85	1.16	**0.90**	1.17
5PP	2B36	**0.28**	0.29	0.30	**0.36**	0.44	0.41
8PS	2B37	**0.45**	0.46	0.46	**0.57**	0.61	0.65
TCU	2X22	**0.31**	0.33	0.33	**0.37**	0.48	0.45
PTH-NAD	2NTJ	1.07	1.07	1.10	1.55	**1.22**	1.49
THT	1BVR	**0.36**	0.38	0.40	**0.47**	0.54	0.54
4PI	2NSD	**0.42**	0.43	0.44	**0.52**	0.61	0.60
GEQ	1P44	2.03	**1.96**	2.08	2.52	**2.04**	2.72
INH-NAD	2IDZ	**0.83**	0.86	0.87	1.16	**0.96**	1.15
Number of Wins:		**16**	2	0	**16**	4	0
Average Rank:		**1.15**	2.13	2.73	**1.30**	2.10	2.60

Finally, Table III presents the $SSWC$ values provided by both versions of Clus-EDA in the 4 experimental scenarios (varying the versions of Clus-EDA and whether picking the best or worst executions). Note that the partitions generated by Clus-EDA consistently present much better values of $SSWC$ than UPGMA and k-means, indicating that the resulting clusters are compact and reasonably well-separated.

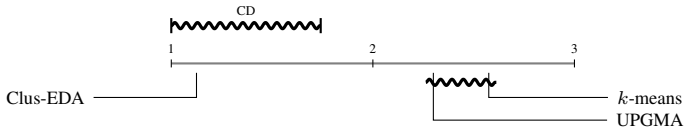
TABLE III. $SSWC$ VALUES OF THE PARTITIONS PROVIDED BY CLUS-EDA VERSIONS AND BY THE BASELINE ALGORITHMS.

Number of clusters	Clus-EDA	UPGMA	k-means
779	**0.26**	0.06	0.12
817	**0.26**	0.07	0.12
5060	**0.41**	0.11	0.18
5174	**0.41**	0.11	0.18
Number of Wins	**4**	0	0

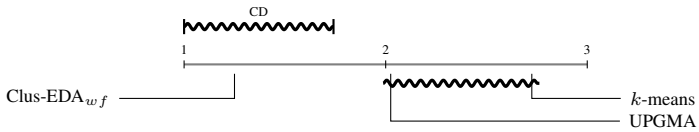
The next step of the experiments is to verify whether the differences in rank values are statistically significant. For that, we employ the graphical representation suggested by Demšar [32], the *critical diagrams*. In this diagram, a horizontal line represents the axis on which we plot the average rank values of the methods. The lowest (best) ranks are to the left of the diagram. When comparing all the algorithms against each other, we connect the groups of algorithms that are not significantly different through a bold line. We also show the critical difference given by the Nemenyi test above the graph.

Figure 4 shows the critical diagrams for all experimental configurations. Note that both versions of Clus-EDA out-

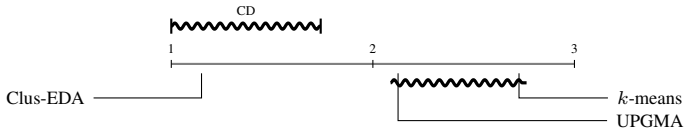
perform k-means and UPGMA with statistical significance, regardless of the fitness function and of the execution. Hence, we are confident to affirm that Clus-EDA is an effective approach for clustering MD simulations. Furthermore, by adjusting parameter w in Clus-EDA$_{wf}$, one can guide the search for solutions with different number of clusters, which ultimately indicates whether one needs a greater or smaller reduction in the computational cost of the task. Our results with Clus-EDA$_{wf}$ reduce the MD simulation to 779 (best execution) and 817 (worst execution) snapshots, which accounts for a reduction of the trajectory size to $\approx 4\%$ of its original size.



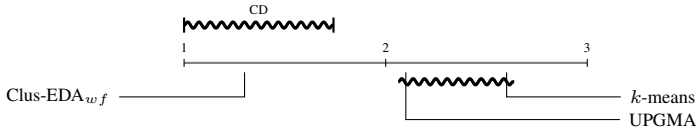
(a) Best execution of Clus-EDA.



(b) Best execution of Clus-EDA$_{wf}$.



(c) Worst execution of Clus-EDA.



(d) Worst execution of Clus-EDA$_{wf}$.

Fig. 4. Critical diagrams showing average ranks and Nemenyi's critical difference.

V. Related Work

Since data clustering can be seen as a combinatorial optimization problem, several efforts towards building evolutionary algorithms for unsupervised problems have been proposed in the literature. We refer the interested reader to a thorough survey on the subject by Hruschka et al. [19], which describes several evolutionary algorithms (mainly genetic algorithms and genetic programming) for generating data partitions and data hierarchies. Notwithstanding, very few studies propose novel Estimation of Distribution Algorithms for data clustering.

Roure et al. [34] propose an EDA that performs partitional clustering with an integer encoding in which each gene has a value over the alphabet $1, 2, ..., k$. As previously discussed, this encoding presents several disadvantages regarding the binary medoid-based encoding of Clus-EDA, such as the permutation problem and the need of defining the number of clusters k in advance. Santana et al. [35] propose an EDA to select parameters for the Affinity Propagation clustering algorithm [36], with the final goal of performing gene expression classification. Meiguins et al. [37] propose the use of EDAs for the automatic generation of density-based clustering algorithms, in an approach that makes use of an EDA as a hyper-heuristic to optimize macro-parameters of a density-based clustering strategy. Finally, note that several papers propose clustering strategies to enhance EDAs [17], [38]–[40], though not EDAs to generate clustering partitions, which is the case of Clus-EDA. To the best of our knowledge, this paper presents the first EDA that generates data partitions following the binary medoid-based approach. Moreover, this is the first attempt in making use of evolutionary computation for clustering MD trajectories in order to reduce the computational cost of molecular docking experiments with a Fully-Flexible Receptor model.

VI. Conclusions and Future Work

In this paper, we presented Clus-EDA, which is a novel clustering algorithm based on a univariate estimation of distribution algorithm. We employed Clus-EDA for clustering a Molecular Dynamics trajectory that makes use of structural features from the substrate-binding cavity of the protein receptor. Our hypothesis was that Clus-EDA could considerably reduce the number of conformations of the molecular dynamics ensemble in order to allow the further execution of molecular docking experiments with ligands of interest.

Clus-EDA works by iteratively updating a univariate probabilistic model in order to select cluster representatives (medoids) that ultimately generate the final data partition. For that, Clus-EDA optimizes an internal clustering validity criterion, which is an efficient implementation of the silhouette width criterion [21].

For validating the proposed approach, we compared Clus-EDA with traditional clustering algorithms such as k-means and an implementation of hierarchical agglomerative clustering over the Molecular Dynamics trajectory data regarding the InhA-NADH complex from *Mycobacterium tuberculosis*. Results show that the data partitions generated by Clus-EDA provide a reduction in variance of the free energy of binding for most of the tested ligands. Statistical non-parametric tests indicated that Clus-EDA outperforms the traditional clustering algorithms with statistical significance considering this bioinformatics task.

As future work, we intend to test different clustering validity criteria as fitness function, such as the Davies-Bouldin index [41], and also employ multi-objective optimization techniques such as the Pareto approach. Furthermore, we believe we can improve Clus-EDA with more sophisticated multivariate probabilistic models that do not assume independence among the variables.

Acknowledgements

The authors would like to thank Brazilian research agencies FAPERGS (grant TO2054-2551/13-0), CNPq (grant 305984/2012-8), FAPESP (grant 2010/20255-5), and CAPES, for funding this research. Renata De Paris is supported by the HP-PROFACC grant. Christian Quevedo is supported by a CAPES/FAPERGS PhD scholarship. Moreover, the authors would like to thank Dr. Osmar Norberto de Souza and Dr. Duncan Ruiz for the MD simulation dataset and their expertise in the application domain.

REFERENCES

[1] P. Cozzini, G. E. Kellogg, F. Spyrakis, D. J. Abraham, G. Costantino, A. Emerson, F. Fanelli, H. Gohlke, L. A. Kuhn, G. M. Morris *et al.*, *Journal of medicinal chemistry*, vol. 51, no. 20, pp. 6237–6255, 2008.

[2] C. N. Cavasotto, J. A. Kovacs, and R. A. Abagyan, "Representing receptor flexibility in ligand docking through relevant normal modes," *Journal of the American Chemical Society*, vol. 127, no. 26, pp. 9632–9640, 2005.

[3] H. Alonso, A. A. Bliznyuk, and J. E. Gready, "Combining docking and molecular dynamic simulations in drug design," *Med. Res. Rev.*, vol. 26, no. 5, pp. 531–568, 2006.

[4] S. E. Nichols, R. V. Swift, and R. E. Amaro, "Rational prediction with molecular dynamics for hit identification," *Current topics in medicinal chemistry*, vol. 12, no. 18, p. 2002, 2012.

[5] C. V. Quevedo, R. De Paris, D. D. Ruiz, and O. Norberto de Souza, "A strategic solution to optimize molecular docking simulations using fully-flexible receptor models," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7608–7620, 2014.

[6] L. S. Cheng, R. E. Amaro, D. Xu, W. W. Li, P. W. Arzberger, and J. A. McCammon, "Ensemble-based virtual screening reveals potential novel antiviral compounds for avian influenza neuraminidase," *Journal of medicinal chemistry*, vol. 51, no. 13, pp. 3878–3894, 2008.

[7] R. De Paris, F. A. Frantz, O. Norberto de Souza, and D. D. Ruiz, "wFReDoW: a cloud-based web environment to handle molecular docking simulations of a fully flexible receptor model," *BioMed Res.*, vol. 2013, pp. 1–12, 2013.

[8] N. Sugaya, "Ligand efficiency-based support vector regression models for predicting bioactivities of ligands to drug target proteins," *J. Chem. Inf. Model.*, vol. 54, no. 10, pp. 2751–2763, 2014.

[9] S. Korkmaz, G. Zararsiz, and D. Goksuluk, "Drug/nondrug classification using support vector machines with various feature selection strategies," *Comput. Meth. Prog. Bio.*, vol. 117, no. 2, pp. 51–60, 2014.

[10] Q. Zang, D. M. Rotroff, and R. S. Judson, "Binary classification of a large collection of environmental chemicals from estrogen receptor assays by quantitative structure–activity relationship and machine learning methods," *J. Chem. Inf. Model.*, vol. 53, no. 12, pp. 3244–3261, 2013.

[11] K. S. Machado, A. T. Winck, D. D. Ruiz, and O. Norberto de Souza, "Mining flexible-receptor docking experiments to select promising protein receptor snapshots," *BMC Genomics*, vol. 11, no. Suppl 5, pp. 1–10, 2010.

[12] R. C. Barros, A. T. Winck, K. S. Machado, M. P. Basgalupp, A. de Carvalho, D. D. Ruiz, and O. Norberto de Souza, "Automatic design of decision-tree induction algorithms tailored to flexible-receptor docking data," *BMC Bioinformatics*, vol. 13, p. 310, 2012.

[13] F. Gargano, A. L. Costa, and O. Norberto de Souza, "Effect of temperature on enzyme structure and function: a molecular dynamics simulation study," *Annals of the 3rd International Conference of the Brazilian Association for Bioinformatics and Computational Biology, São Paulo, Brazil*, 2007.

[14] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. California, USA, 1967, pp. 281–297.

[15] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Willey & Sons, 1990.

[16] M. Hauschild and M. Pelikan, "Swarm and Evolutionary Computation," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.

[17] J. M. Peña, J. A. Lozano, and P. Larrañaga, "Unsupervised Learning Of Bayesian Networks Via Estimation Of Distribution Algorithms: An Application To Gene Expression Data Clustering." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems ()*, vol. 12, pp. 63–82, 2004.

[18] H. Mühlenbein and G. Paaβ, "From recombination of genes to the estimation of distributions i. binary parameters," ser. Lecture Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 1996, vol. 1141, pp. 178–187.

[19] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho, "A Survey of Evolutionary Algorithms for Clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 39, no. 2, pp. 133–155, 2009.

[20] E. R. Hruschka, R. J. G. B. Campello, and L. N. de Castro, "Evolving clusters in gene-expression data," *Information Sciences*, vol. 176, no. 13, pp. 1898–1927, 2006.

[21] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 0, pp. 53 – 65, 1987.

[22] D. A. Rozwarski, C. Vilchèze, M. Sugantino, R. Bittman, and J. C. Sacchettini, "Crystal structure of the Mycobacterium tuberculosis enoyl-ACP reductase, InhA, in complex with NAD+ and a C16 fatty acyl substrate," *J. Biol. Chem.*, vol. 274, no. 22, pp. 15 582–15 589, 1999.

[23] D. Case, T. Darden, T. Cheatham III, C. Simmerling, J. Wang, R. Duke, R. Luo, R. Walker, W. Zhang, K. Merz *et al.*, "AMBER 12; University of California: San Francisco, CA." 2012.

[24] T. A. Binkowski, S. Naghibzadeh, and J. Liang, "CASTp: computed atlas of surface topography of proteins," *Nucleic Acids Res.*, vol. 31, no. 13, pp. 3352–3355, 2003.

[25] F. M. Richards, "Areas, volumes, packing and protein structure," *Annu. Rev. Biophys. Bio.*, vol. 6, pp. 151–176, 1977.

[26] M. L. Connolly, "Analytical molecular surface calculation," *J. Appl. Crystallogr.*, vol. 16, no. 5, pp. 548–558, 1983.

[27] W. L. DeLano, "Pymol," *DeLano Scientific, San Carlos, CA*, vol. 700, 2002.

[28] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[29] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, and A. J. Olson, "AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility," *J. Comput. Chem.*, vol. 30, no. 16, pp. 2785–2791, 2009.

[30] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–242, 2000.

[31] R. D. Paris, C. V. Quevedo, D. D. A. R. andO. Norberto de Souza, and R. C. Barros, "Clustering molecular dynamics trajectories for optimizing docking experiments," *Comput. Intell. Neurosci.*, vol. in press, 2015.

[32] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[33] R. Iman and J. Davenport, "Approximations of the critical region of the friedman statistic," *Communications in Statistics*, pp. 571–595, 1980.

[34] J. Roure, P. Larranaga, and R. Sangüesa, "An empirical comparison between k-means, GAs and EDAs in partitional clustering," in *Estimation of distribution algorithms*. Springer, 2002, pp. 343–360.

[35] R. Santana, C. Bielza, and P. Larrañaga, "Affinity propagation enhanced by estimation of distribution algorithms," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 331–338.

[36] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, p. 2007, 2007.

[37] A. Meiguins, R. Limao, B. Meiguins, S. Junior, and A. Freitas, "AutoClustering: An estimation of distribution algorithm for the automatic generation of clustering algorithms," in *IEEE Congress on Evolutionary Computation (CEC)*, 2012, pp. 1–7.

[38] J. Peña, J. Lozano, and P. Larrañaga, "Benefits of data clustering in multimodal function optimization via EDAs," in *Estimation of Distribution Algorithms*, ser. Genetic Algorithms and Evolutionary Computation, P. Larrañaga and J. Lozano, Eds. Springer US, 2002, vol. 2, pp. 101–127.

[39] Q. Lu and X. Yao, "Clustering and learning gaussian distribution for continuous optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 2, pp. 195–204, 2005.

[40] C. W. Ahn and R. S. Ramakrishna, "Clustering-based probabilistic model fitting in estimation of distribution algorithms," *IEICE transactions on information and systems*, vol. 89, no. 1, pp. 381–383, 2006.

[41] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.