2021 IEEE 17th International Conference on eScience (eScience) | 978-1-6654-0361-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/eScience51609.2021.00016

Compound Segmentation via Clustering on Mol2Vec-based Embeddings

Daniyal Kazempour
Ludwig-Maximilians-University
Munich, Germany
kazempour@dbs.ifi.lmu.de

Anna Beer
Ludwig-Maximilians-University
Munich, Germany
beer@dbs.ifi.lmu.de

Melanie Oelker
Umeå University
Umeå, Sweden
melanie.oelker@umu.se

Peer Kröger
Christian-Albrechts-University
Kiel, Germany
pkr@informatik.uni-kiel.de

Thomas Seidl
Ludwig-Maximilians-University
Munich, Germany
seidl@dbs.ifi.lmu.de

***Abstract*—During different steps in the process of discovering drug candidates for diseases, it can be supportive to identify groups of molecules that share similar properties, i.e. common overall structural similarity. The existing methods for computing (dis)similarities between chemical structures rely on a priori domain knowledge. Here we investigate the clustering of compounds that are applied on embeddings generated from a recently published Mol2Vec technique which enables an entirely unsupervised vector representation of compounds. A research question we address in this work is: do existent well-known clustering algorithms such as k-means or hierarchical clustering methods yield meaningful clusters on the Mol2Vec embeddings? Further, we investigate how far subspace clustering can be utilized to compress the data by reducing the dimensionality of the compounds vector representation. Our first conducted experiments on a set of COVID-19 drug candidates reveal that well-established methods yield meaningful clusters. Preliminary results from subspace clusterings indicate that a compression of the vector representations seems viable.**

***Index Terms*—Clustering, Subspaces, Embedding, Compounds**

I. Introduction

Within the drug discovery process, there are certain aspects that can benefit from data mining techniques. During a virtual screening, it is desired to achieve high coverage of the chemical space while keeping the number of compounds to be screened to a minimum. Screening only the representative compounds from a cluster can reduce the number of compounds, especially for computationally high-demanding screening methods. During the compound optimization step often similar compounds are tested. While designing very similar compounds is not so difficult for medicinal chemists, finding non-obvious but still similar compounds is more challenging. Picking larger clusters of neighbouring clusters could assist with finding these non-obvious, but yet similar compounds. The existent approaches to obtain the next (k) similar compounds rely on domain expert knowledge and feature engineering. With the introduction of unsupervised neural-based embedding techniques such as among their most prominent representative Word2Vec [1] for word embeddings from text data, promising alternatives to the 'hand-crafted' embedding methods entered the stage. Recently, Mol2Vec [2] has been proposed to embed the complex data of molecules so that their vector representation can be utilized for computing (dis)similarities. This in turn paves the path for computing segmentations (clusterings) of compound sets and further for identifying representative compounds per each cluster (prototypes such as centroids, medoids etc.). One essential question that emerges at this stage is: how meaningful are the resulting clusterings from well-established algorithms such as k-means [3] or hierarchical clustering [4] on the Mol2Vec-based embeddings? Another aspect investigated in this work is the data compression by applying dimensionality reduction techniques. The resulting embeddings from Mol2Vec encompass several hundred dimensions ($d = 300$). If the dimensionality reduction to e.g. $d = 10$ can maintain a meaningful (dis)similarity computation it would make a (substantial) difference for the computational costs. To provide a first approach to this we apply the ORCLUS [5] algorithm, an arbitrarily oriented subspace clustering method.

In summary, we provide in this work the following contributions:

- A novel, publicly available dataset of potential compounds against SARS, MERS and COVID-19 based on recent publications, compiled by the authors of this paper
- A publicly available pipeline for automatically obtaining SDF files, embedding them in vector representations through Mol2Vec and clustering them
- An investigation on the applicability and meaningfulness of k-means and hierarchical clustering algorithms on Mol2Vec embeddings
- An investigation of compressibility w.r.t. the dimensionality of the embeddings through subspace clustering approaches
- A prototype compound for (subspace)clusters in form of a medoid

DOI 10.1109/eScience51609.2021.00016

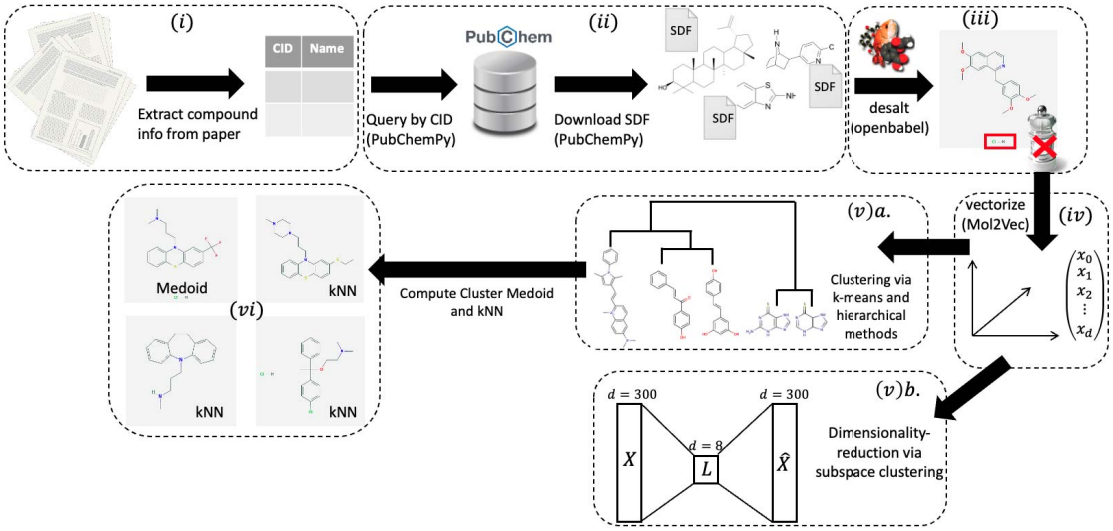

Fig. 1. Pipeline for the segmentation of compounds w.r.t subspaces.

A. Data Description

Due to its topicality, we collected over 100 compound names of potentially effective compounds against SARS, MERS and COVID-19 from the recent literature [6], [7], [8]. From the compound names, CIDs (compound ids) and CAS (chemical abstract service) we obtained automatically all corresponding SDF (structural data file) [2D representation of compounds]. Openbabel [9] was used for desalting the compounds, which is the removal of counterions from the SDF file. This is necessary to prevent biased effects in the clustering process. Finally, these SDFs are vectorized by the embedding method Mol2Vec [2].

B. Framework Overview

Figure 1 provides an overview over the (following) steps in our framework: (i) Obtaining an array of compound ids (CID) and names from the literature, (ii) automatically fetching SDF via a PubChem API using PubChemPy[1], (iii) desalting the obtained SDFs using functionalities from Openbabel[2], (iv) vectorizing (via embedding) the desalted SDFs using Mol2Vec, (v) a. investigating the meaningfulness of segmentations from prominent clustering techniques (k-means, hierarchical clustering) (v) b. investigating the compressibility of the data through dimensionality reduction via subspace clustering (vi) obtaining for each cluster their respective medoid and their corresponding kNN.

II. Related Work

The vector representation of compounds unlocks the possibility to use a wider range of clustering algorithms. To combine these two techniques the literature was reviewed regarding vector embedding and clustering methods for molecules. Objects sharing a common context can be identified via their vector representation as their vectors will be located closely in vector space. In the case of the $*$-2Vec embeddings neural networks are trained to learn different contexts, which allows the vectorization of objects. Since the publication of Word2Vec [1] in 2013 the popularity of the $*$-2Vec concept increasing in different scientific domains as for example Node2Vec [10] for graph embeddings or Gene2Vec [11] for learning embeddings of genes (in the bioinformatics domain). With the introduction of Mol2Vec molecular structures can be embedded as follows: First, a compound is fragmented into substructures of a fixed radius. This radius influences the magnitude at which substructures of compounds are encoded. The encoding of the substructures is performed by mapping substructures to so-called Morgan fingerprints [12]. These encoded substructures serve as an input for Mol2Vec which returns a vector representation of substructures in form of several dimensions. Subsuming the substructure vectors (finally) results in a single vector representation for a molecule. The clustering algorithms for chemical compounds can be divided into two categories: (1) hierarchical and (2) non-hierarchical. For the hierarchical methods, one prominent technique relies on the maximum common substructure (MCS) search [13], which is used commercially as LibMCS. Internally it computes the maximum common substructures between molecules and computes the distances between the MCS. Based on the distances a Ward hierarchical clustering [4] is applied. For the non-hierarchical approaches, one algorithm is indeed k-means [3] which is applied on fingerprints such as i.e. Morgan fingerprints [12]. Another clustering method is the Jarvis-Patrick algorithm [14], which is a kNN based clustering approach. The similarity computation for the kNN is based on the Tanimoto coefficient [15]. Lastly, the so-called Bemis-Murcko algorithm [16] is effective in deriving scaffolds from compounds by side-chain atom removal. Through this, a hierarchy is obtained of

[1]https://pubchempy.readthedocs.io/en/latest/
[2]http://openbabel.org/wiki/Main_Page

reduced graphs representing the scaffolds of the compounds. In contrast to the named approaches, our framework first performs an embedding into a high-dimensional vector space which preserves the characteristics of the compounds and the similarities. After this embedding step, we can potentially apply any existent clustering method which is capable of handling vector representations of objects. This is of interest, since different clustering algorithms reveal different properties within a given dataset.

III. The Semantics Behind Clustering of Molecules

There exists a rich body of literature revealing a broad spectre of clustering algorithms within the past decades. At that point, there may be the temptation to just randomly select one and apply it to the compound data. Yet there are certain questions that emerge when using clustering algorithms:

(1) What are the semantics of the result and the underlying models of the clustering, i.e. what is the information which a *mean* of a cluster of compounds conveys? and (2) How does one set the parameters, which is ultimately connected to the semantics of the parameters themselves, i.e. what does increasing or decreasing the number of partitions k mean for the resulting clusters of compounds.

To approach these questions we dedicate this section to elaborate on three archetypes of clustering algorithms with one aspect in mind, namely to establish a connection between the theory behind the algorithms and their meaning in the context of compound segmentation.

A. k-means

We begin our elaborations with one of the oldest, yet among the most used clustering algorithms so far: k-means [3]. This popularity is inter alia owed to the simplicity of the method. From an operational level the algorithm works as follows: First randomly k centroids $\mathcal{M} = \{\mu_1, ..., \mu_k\}$ are distributed to a given dataset $\mathcal{X}$ (random seeds), then for each object $x_i \in \mathcal{X}$ the distance to each of the centroids $d(x_i, \mu_j) \forall \mu_j \in \mathcal{M}$ is computed. An object x_i is then assigned to its closest centroid μ_j. Afterwards, from all objects x_i that are assigned to their respective centroid μ_j and thus their cluster C_j the mean is re-computed $\mu_{j_{new}} = \frac{\sum_{i=1}^{|C_j|}}{|C_j|} x_i$. These two steps of assigning objects to their closest centroid and re-computing the centroids are repeated until convergence is reached, which means in other terms that the centroids barely change their positions after the next iteration. While the previous elaborations were on an operational level (aka 'how does it work'), k-means can be expressed in terms of an optimization problem:

Definition 1 (k-means objective)**.** *Given a dataset $\mathcal{X}$ and a number k of wanted partitions $\mathcal{C} = \{C_1, ..., C_k\}$, the objective of the k-means clustering algorithm is:*

$$\underset{\mathcal{C}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{j=1}^{|C_i|} ||x_j - \mu_i||^2 = \underset{\mathcal{C}}{\operatorname{argmin}} \sum_{i=1}^{k} |C_i| \sigma(C_i), \forall x_j \in C_i$$

where σ denotes the variance.

This objective formulation translates to minimizing the within-cluster sum of squares distance between the objects x_j to their respective centroid μ_i (left side of the equation) which is synonymous with minimizing the variance within each of the cluster (right side).

Following the formal aspects of k-means, we approach now the question which we previously announced: What are the semantics of a k-means result and the centroids in the context of partitioning a set of compounds? The k-means clustering of a dataset of compounds has as one property, namely that the k partitions are maximized with respect to their compactness, or in other terms: the clusters exhibit a minimized variance. This implies that the clusters exhibit a convex shape. This convex character of the resulting clusters comes with the implication that potentially densely connected, arbitrarily-shaped clusters would be separated. This effect however can be mitigated up to a certain scale, by decreasing in such cases the number of expected partitions k, which can enforce a merging of previously separate adjacent and density connected clusters. The centroid of a cluster represents a non-existent mean prototype/representative for an entire cluster. If the single attributes or features of a vector can be assigned to a specific property (i.e. length, volume, etc.) then the mean vector would embody the mean for each of the single features. However, we shall elaborate more in detail in the upcoming subsections that the Mol2Vec embeddings do not provide explicit semantics for each of the features. Moving away from the semantics of the results, we shift our attention to the semantics of the parameter k. In the context of compound segmentation, k is an immediate control for the number of distinctive partitions of molecules that a domain expert would expect. In cases where the domain experts are uncertain about the chosen number of clusters, one heuristic is to increase or decrease k and to observe the resulting clusters with respect to their meaningfulness. One issue of k-means and its variants is the lacking handling of outliers. An outlier is in this term a compound which is $n \times \sigma$ the variance distant. However to make these outliers *visible*, one can plot the distances of compounds to their centroid with the vertical axis denoting the distances to the centroid and the horizontal axis being the compounds sorted in ascending order of distances. As a teaser, such visualization can be seen in Figure 8 where the last compound has a distance to its prototype which is several times higher compared to the other compounds. Lastly, there is however a word of caution regarding the outliers in k-means. Since they are explicitly not detected by the algorithm, they may cause a phenomenon that we coin with the term "centroid poisoning". Imagine a sequence of numbers $S = [2, 2, 8, 8, 100]$. The mean of S is 60, while without the last element (100), the mean would be 10. An outlier therefore can distort the centroid, since centroids are a mean computation and the mean is susceptible to outliers. A remedy for future work may be the use of medoid based methods that rely on the concept of the median, which is known to be more robust towards outliers.

B. Hierarchical Clustering

In contrast to k-means, hierarchical clustering approaches are not limited to yield clusters of convex structures. Further one of their properties is that they provide a hierarchy of distances between compounds which comes with certain potential benefits as we shall see below. On an operational level, we have here two common approaches in our taxonomy, namely *divisive* hierarchical clustering, which is a top-down approach and *agglomerative* which is a bottom-up approach. In this work we decided to investigate the agglomerative variant, yet we encourage in future work to also study the divisive technique on compound data. For the agglomerative approach, the steps are as follows: First, each compound represents its own cluster, then all pairwise distances between initial compounds are computed. Afterwards, those pairs of compounds A and B are merged which are most close to each other, forming a new cluster $C = A \cup B$. A and B are removed from the set of current clusters, while C is included in the current set. These steps are repeated until the set of current clusters contains only a single cluster in the end. Following the operational description of the agglomerative hierarchical clustering, we now elaborate below on the semantics with respect to compound segmentation. The dendrogram that a hierarchical clustering yields can be interpreted as follows: The root represents the entire compound dataset. In contrast, each leaf represents a single compound in the data. Intermediate nodes have the semantics of the union of all compounds in their sub-trees. The vertical axis or height of a dendrogram represents the distances between two child nodes. In general, the intuition behind the distances within a dendrogram are as follows: the more upwards we go, the fewer cluster we obtain where at the same time within the clusters the distances increase. In contrast, the more downwards we move in the dendrogram towards the single compounds (leaf nodes), the more, smaller clusters (or sub-groups, to be more precise) we obtain. Furthermore, the hierarchical character of the dendrogram provides information regarding a potential question that may arise, namely: "Are within a given group of compounds, subgroups in which the compounds are more similar to each other?"

Another property that is inherent to hierarchical clustering is the choice of the *linkage criterion*. In this criterion, it is manifested when to merge clusters. There are four prominent linkage criteria, namely:

(1) Single-link (Nearest Point algorithm)

$$d(C_i, C_j) = \min(dist(x_i, x_j)), \forall x_i \in C_i, x_j \in C_j$$

(2) Complete-link (Farthest Point or Voor Hees algorithm)

$$d(C_i, C_j) = \max(dist(x_i, x_j)), \forall x_i \in C_i, x_j \in C_j$$

(3) Average-link (Unweighted Pair Group Method with Arithmeic mean [UPGMA])

$$d(C_i, C_j) = \sum_{i,j} \frac{dist(x_i, x_j)}{|C_i| \cdot |C_j|}, \forall x_i \in C_i, x_j \in C_j$$

(4) Ward (Incremental algorithm)

$$d(C_{C_i \cup C_j}, C_u) = \sqrt{\frac{|C_u| + |C_i|}{T} d(C_u, C_i)^2 + \frac{|C_u| + |C_j|}{T} d(C_u, C_j)^2 - \frac{|C_u|}{T} d(C_i, C_j)^2}$$

where $T = |C_u| + |C_i| + |C_j|$ Here C_u denotes an unused cluster in the forest.

Those four different linkage types preserve different semantics. With single-link, the distance between two compound clusters is computed based on the two closest compounds from both clusters. In contrast with complete-link, the distance is computed based on the two most far/dissimilar compounds from both clusters. In average-link, the average distance between all pairs of compounds from both clusters is computed. Lastly in the Ward linkage, the distance between two compound clusters is computed by minimizing the variance of the clusters that are supposed to be merged. Besides the semantics of the distance functions, some of the methods come with their own properties. As an example using the single-link distance, yields long chain-like clusters with large variance, known in the literature as the *single-link effect*. This also implies, that unlike with k-means, here arbitrarily oriented clusters can be found. It remains to investigations in future work which semantics long arbitrarily-shaped chains of compounds bare in context of Mol2Vec embeddings. In contrast, complete-link yields smaller, more separated, equi-sized convex clusters. As such, complete-link fosters a better separability of clusters, however, by relying on the maximum distances of objects between two clusters, it is also highly susceptible to outliers. Average-link aims to provide a compromise between single-link and complete-link. Ward-based clusterings are more in the direction of a k-means clustering since it is tailored at minimizing the intra-cluster variance. In this work, we chose the ward method for our experiments, since it may facilitate the comparisons to k-means clusterings. Lastly, it is noteworthy to mention that hierarchical approaches are parameter-free, since they do not actually provide a clustering in the first place, but a dendrogram. The clustering can emerge based on two approaches: (1) the domain-experts know how many clusters they expect, in such a case, a horizontal cut can be made in the dendrogram where the one tree splits into k sub-trees yielding the compound clusters, or (2) domain-experts perform a cut in the dendrogram at a specific distance where they deem the clusterings to be meaningful.

C. Subspace Clustering

With increasing dimensionality, the computation of distances becomes less expressive and therefore meaningful to provide statements like "compound X is similar to compound Y" which can be traced back to the so-called "curse of dimensionality" as elaborated on in [17]. Further, from all the number of dimensions ($d = 300$, in our case of the Mol2Vec embeddings) one is tempted to question if all dimensions are needed. To approach both aspects (the curse and the wish to get rid of unnecessary dimensions) dimensionality reduction

seems to be the method of choice. For this purpose, one may argue that applying a simple Principal Component Analysis (PCA) would be sufficient. PCA is a linear dimensionality reduction technique that yields a single arbitrarily oriented subspace. It may however occur that clusters do not reside in single but multiple linear subspaces, or that they are located in arbitrarily shaped, inherently non-linear subspaces. Projecting in such cases the compounds down to a single linear subspace from PCA would mean that non-linearities of the underlying distributions are neglected. One approach that performs a piece-wise linear approximation of the data (PLA) is local PCA [18] by combining vector quantization like in k-means with PCA. As such, local PCA can be considered as the ancestor of the field of arbitrarily oriented subspace clustering (AOSC) algorithms, which are in the literature also coined with the term of *subspace clustering*. One formal definition of (arbitrarily oriented) subspace clustering is as follows:

Definition 2 (Subspace Clustering)**.** *Given a dataset $\mathcal{X}$ of dimensionality d. The task of subspace clustering is to detect k clusters $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ which reside in their respective individual l-dimensional ($l < d$) subspaces $\mathcal{S} = \{S_1, S_2, ..., S_k\}$. The variable l denotes the l-dimensions of the arbitrarily oriented subspace which exhibit the l-highest variance σ (eigenvalues, in terms of PCA). This l-dimensional subspace is denoted as correlation subspace. The complementary $c = d - l$-dimensional subspace exhibits the lowest variance, meaning the compounds projected to that subspace are dense and therefore highly similar. This complementary subspace is denoted as cluster subspace. The task of subspace clustering is to find for a given number of clusters k and the correlation subspace dimensionality l or cluster subspace c, the subspaces in a way s.t. it holds for each cluster C_i:*

$$\min \sigma(\mathcal{P}(C_i, S_{i_c})) \equiv \max \sigma(\mathcal{P}(C_i, S_{i_l})), \forall C_i \in \mathcal{C}, S_i \in \mathbb{S}$$

where $\mathcal{P}(C_i, S_i)$ denotes the projection of all objects $x_j \in C_i$ to their respective subspace S_i.

To briefly recap, with subspace clustering we achieve a partitioning of compounds in such a way that the compounds within each cluster are maximized w.r.t. their similarity in their respective cluster subspace and equivalently maximized w.r.t. their variance in their respective correlation subspace. On this equivalence between minimizing the variance in cluster subspace and maximizing the variance in correlation subspace, elaborations have been made in a recently published work [19]. The application of such algorithms approaches the two expected benefits: (1) mitigating the curse of dimensionality by (2) getting rid of unnecessary/irrelevant features. Having both, fewer features and a partitioning of the dataset, based on similarity of objects in their lower-dimensional subspaces is (a) a compression of the data, in a sense that most of the information is expressed with less and (b) that faster queries can be performed. The latter is connected to the following observation: having already partitions where objects are most similar within them accelerates queries by no longer having the need to compare or compute distances between all

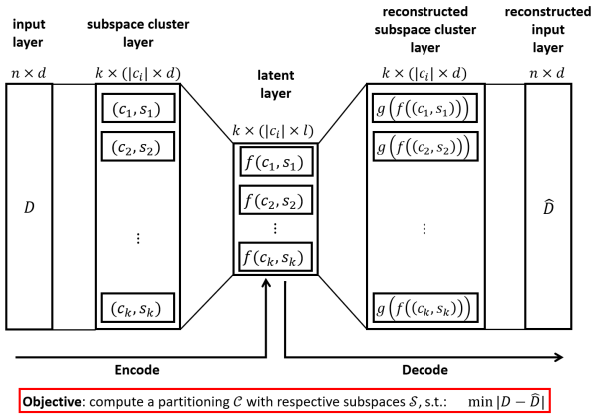

Fig. 2. Architecture of Subspace Clustering as an Autoencoding Task.

objects but only to their prototypes (centroids, medoids etc.), further instead of using $d = 300$ dimensions for similarity computations, a smaller subset i.e. $d = 10$ are needed.

The aspect of compression reveals a trade-off between two criteria, namely: (i) reconstruction error and (ii) model complexity. On the one hand, we wish to find a subspace clustering result in such a way that when the compounds are projected to their subspaces, the 'loss' of information (aka reconstruction error) is kept to a minimum, while on the other hand, we do not wish to have more clusters and dimensionality of subspaces than needed. To approach this delicate balance, we apply an autoencoding perspective on the subspace clustering problem. This perspective has been elaborated on in detail in a recent work [20]. At this point we want, to avoid any confusions or misunderstandings, by explicitly stating that we apply the autoencoding *perspective*, not any neural autoencoders. For subspace clustering, we use the already existent ORCLUS [5] algorithm. It is a method that relies on vector quantization (i.e. k-means [3]). The parameters of this method are the dimensionality of the cluster subspace l and the number of clusters k. With both parameters the so called *model complexity* can be controlled. If we would choose a high dimensionality of the subspaces l and a high number of clusters k, ORCLUS would not learn within the autoencoding architecture (c.f. Fig. 2). However, to enforce the framework to learn a lower-dimensional representation we impose a so-called *bottleneck* by setting k and l to low values. As a consequence, fewer subspaces with lower dimensionality represent the original data in the *latent layer* as seen in Figure 2 in the center of the architecture.

At that point, one may be tempted to down-regulate the number of clusters and the dimensionality of the subspaces to reduce the model complexity. This is however only one side of the coin and comes with the cost of the reconstruction error. Maintaining a good "quality" of the compressed representation in terms of obtaining a representation with low reconstruction error is the other side. This balance between model accuracy and model complexity is the cardinal challenge of this compression task that is approached in the experiment section.

D. The Back-Mapping Problem and Medoids

We wish to introduce a last formal aspect of this work by sketching the following case: Suppose we are given a set of Mol2Vec embedded compounds and compute the centroid of this set by summing up the scalars of all compounds among all 300 features individually and dividing it by the number of compounds in this set. What we obtain is a mean vector from all the compounds. While for each compound vector, we have its corresponding original compound (2D) structure, we do not have a 2D structure for the computed mean vector (centroid) and have, as the current state of the Mol2Vec framework, no "mean compound". This lacking of a "pre-image" we coin here with the term *back-mapping problem*.

Definition 3 (Back-Mapping Problem)**.** *Given a dataset of compounds $\mathcal{M}$ and their Mol2Vec embedding $\varepsilon(\mathcal{M})$. For each compound $m_i \in \mathcal{M}$ it holds:*

$$\exists : m_i \mapsto \varepsilon(m_i) \wedge \varepsilon(m_i) \mapsto m_i$$

For a computed centroid $\varepsilon(\mu) = \frac{\sum_{i=1}^{|N|} m_i}{|N|}$ from a subset of compounds $N \subseteq \mathcal{M}$ in embedded space it holds:

$$\neg\exists\varepsilon(\mu) \mapsto \mu$$

meaning that there does not exist a back-mapping of the calculated mean compound in embedded space to a compound.

A question that arises at this point is now: how can we obtain a prototype or a representative compound from a detected cluster which actually also exists in the original data and therefore does not succumb to the back-mapping problem? To approach this issue, we refer to the computation of the median, where the values are first sorted in ascending order and then the "middle" of this series is chosen. This "middle" element is existent and unlike the mean not computed. Adhering to this idea, we determine from a cluster of compounds its medoid, which can be considered as the "median" compound in a given cluster. In more formal terms, a medoid η is a representative compound of a cluster of molecules for which holds that the average dissimilarity to all other compounds within the cluster is minimal, which can be expressed as:

Definition 4 (Medoid of a Cluster)**.** *Given a cluster of compounds $C_i \in \mathcal{C} = \{C_1, ..., C_k\}$ with x_j being compounds of $C_i = \{x_1, ..., x_n\}$. Further given a distance function d. The medoid of a compound of clusters is defined as:*

$$\eta := \underset{y \in C_i}{\operatorname{argmin}} \sum_{i=1}^{n} d(y, x_i)$$

IV. Experimental Results

In this section, we elaborate on the conducted experiments and discuss the results and insights. All experiments were conducted on the COVID-19 dataset as described in Section I A. The full dataset, the entire code, and all intermediate, as well as final results, are publicly available on https://github.com/hamilton-function/MolClust. All experiments were conducted on a machine with Intel Core i7-6700 with 3.4GHz, 32 GB available RAM. In order to avoid any misunderstandings, we'd like to finally elaborate on the term *outlier*. In the context of clustering, we consider an object as an outlier if it is "too distant" to a cluster or a cluster model (i.e. centroid or hyperplane). In a chemical context, an outlier refers to a single molecule, which is significantly different to others, yet not in the sense of "not-belonging" to the dataset i.e. being noise, or not being relevant for chemical questions.

A. Model Comparison

In this subsection, we compare in a qualitative approach the results of k-means, hierarchical clustering and subspace clustering against a manual segmentation performed by our domain expert which we will refer to in the remainder of the experiment section as "ground truth". For k-means and hierarchical clustering we utilized the implementation provided by the sklearn framework[3]. The k-means implementation by default applies a k-means++ [21] strategy for obtaining more stable initial centroids and for faster convergence. However, the sklearn implementation performs 10 runs and selects the best result from all 10 clusterings based on the inertia criterion (Sum of squared distances of samples to their closest cluster center). The hierarchical clustering from sklearn applies by default the ward agglomerative strategy. For subspace clustering, we used the ORCLUS algorithm, which is implemented in the ELKI framework [22]. What may appear striking is the choice of a large $k = 16$ for comparing the results from different clustering models. The decision is justified by the clustering of the "ground truth". In its entirety (all methods and all tested k) we could also observe that the results performed better for larger k, with the exception of ORCLUS on which we will elaborate more in detail in the upcoming subsections.

The hierarchical clustering on the Mol2Vec embedding yields for $k = 16$, as seen in Figure 3, mostly meaningful partitions. As one example the pink-colored cluster contains mostly steroid analogues as shown by the medoid representative compound (dashed line). Another example is the green-colored cluster (second from left) which contains nucleoside analogues. On the contrary, there are clusters such as the red one (third from left) where it is not obvious what is actually the common theme between the members of that subset. It may be that this cluster needs further refinement by splitting into more clusters. It is the subject of future work to further investigate if there exists a semantically common theme of the compounds residing in this cluster. Furthermore, we have four singleton clusters, characterized by consisting of only one compound. These "clusters" can actually be considered as outliers, which nevertheless does not mean that they are less important or of lower interest. To see the full clustering for different $k = 2, 4, 6, 10, 16$ we refer to our repository [https://github.com/hamilton-function/MolClust/tree/master/expres/slink] where for each clustering each cluster contains png images of the compounds with their compound id (CID) as their filenames.

[3]https://scikit-learn.org/stable/modules/clustering.html

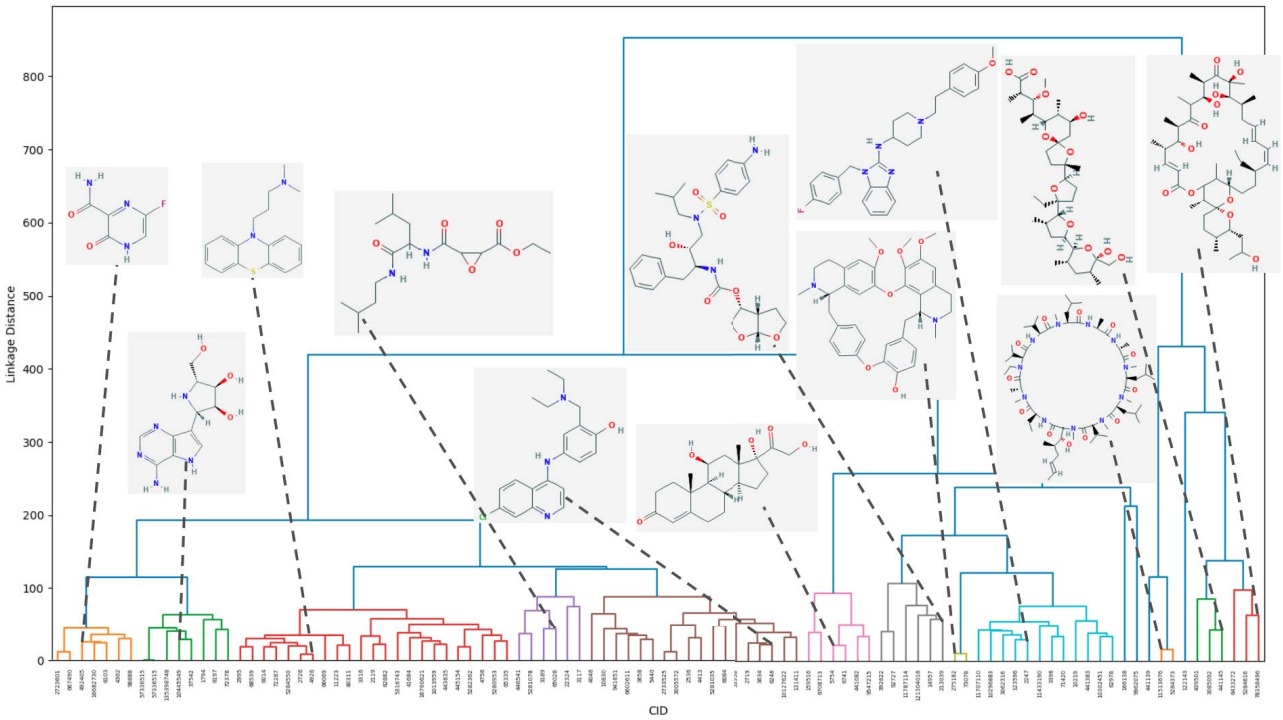

Fig. 3. Resulting hierarchy from an agglomerative hierarchical clustering on Mol2Vec embedded compounds. For each cluster (except four outliers/'singleton clusters') the respective medoids are shown. Outlier compounds are colored in blue like the rest of the dendrogram.

For the k-means clustering, we can see in Figure 5 many similarities like e.g. the nucleoside analogues (top left corner). We also observe a larger cluster where no clear semantics can be derived from it, namely the purple segment (bottom center). There are however also cases like the afore mentioned steroid cluster which is split here into two clusters (top center), one bigger group and a smaller cluster consisting of two steroidal compounds. The split into two clusters seems to be made based on differences in the substructures. Since k-means has no explicit noise handling we obtain here also four singleton clusters. The single clusterings and clusters can be inspected, like for the hierarchical clustering, in our repository [https://github.com/hamilton-function/MolClust/tree/master/expres/kmeans] for $k = 2, 4, 6, 10, 16, 24$.

In Figure 4 one can observe at the example of one cluster that all the different methods are capable to detect meaningful groupings. In this particular case, all approaches are capable to form a cluster of steroidal compounds with certain deviations. On the contrary, ORCLUS fails to detect meaningful cluster (except this steroid group) while k-means and the hierarchical approach are capable of detecting results similar to the "ground truth". At this point, it is also vital to note that despite yielding better clusterings compared to ORCLUS, k-means and hierarchical clustering are still struggling with assigning certain compounds to appropriate clusters.

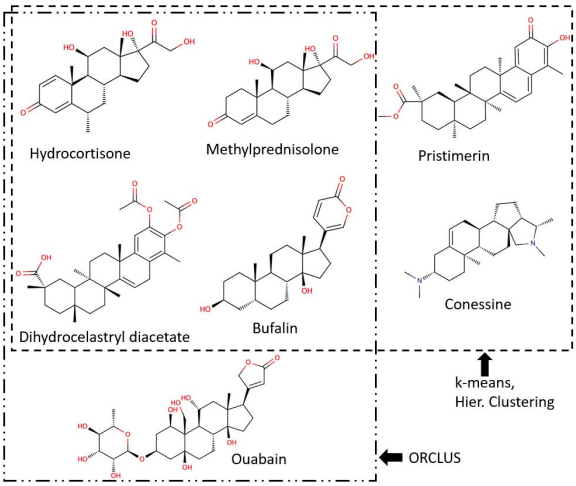

Fig. 4. Cluster of steroidal compounds detected at $k = 16$ ($l = 4$ for ORCLUS), with k-means, hierarchical clustering and ORCLUS. All seven compounds were identified in the "ground truth" as one cluster.

B. In-depth Analysis of Subspace Clustering Results

From the rather disappointing results of ORCLUS we wanted to further investigate why this approach failed, by discovering possible reasons why the ORCLUS results are not as meaningful as those from k-means and hierarchical clustering. To approach this task, we identified potential targets for facilitating the understanding of this performance. One

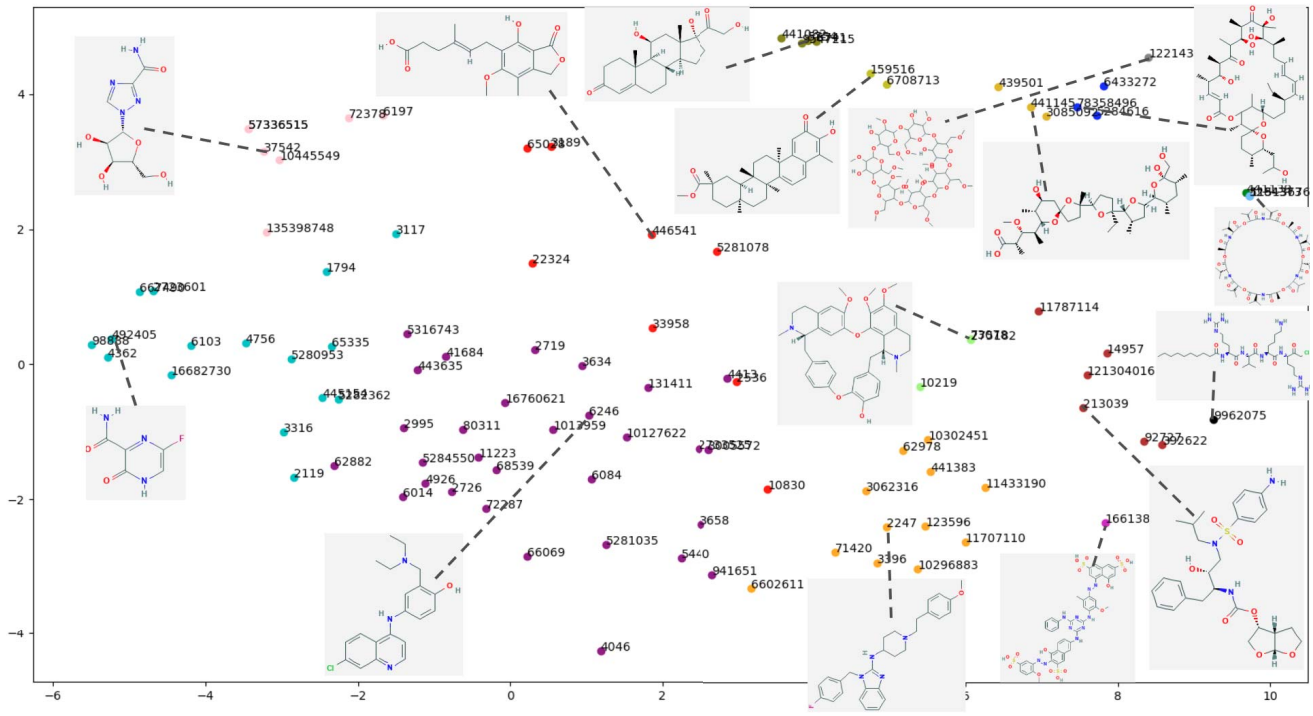

Fig. 5. k-means clustering on Mol2Vec embeddings. 2D visualization enabled through a t-SNE embedding of the k-means results. The illustrated molecules are the medoid prototypes of each cluster. The clustering was performed prior the t-SNE embedding on the Mol2Vec representations.

initial thought was that for $k = 16, l = 4$ we picked up a bad combination of cluster and dimension cardinality. Therefore, one of the more obvious targets was to adhere to the parameters that our model takes for the computation of the clusterings. Does this bad performance also occur at different (k, l) pairs? For this we looked closely at the clustering results of ORCLUS for $l = 4$ and $k = \{2, 4, 8, 10, 16\}$. Further we kept in another trial $k = 4$ constant and analyzed the results for different $l = \{4, 5, 10, 15, 20, 25\}$. We deliberately chose here smaller k since we already observed at $k = 16$ that ORCLUS fails to form meaningful clusters. While the choice of the fixed k and l seems arbitrary, we chose on purpose $k = 4$ and $l = 4$ since based on the autoencoder principle in III C, we observed there a striking minimum of the reconstruction loss as it can be seen in Figure 6 (circled region).

The main purpose is to obtain with few dimensions and few clusters a clustering which comes at the same time with a low reconstruction error. Exploiting the boundaries to compress the compound data, in this case from 300 dimensions down to 4 is ultimately opposed to the question of how meaningful the resulting clusters are. Another potential reason besides 'bad' parameter settings is the fact that it is not possible or at best very challenging to identify the common (sub)structures by manual inspection for investigating the meaningfulness of the clustering. In order to facilitate the detection and thus increasing the chance of observing potential common (sub)structures within the compounds, we added further methods with the hope of being capable to recognize commonalities within a

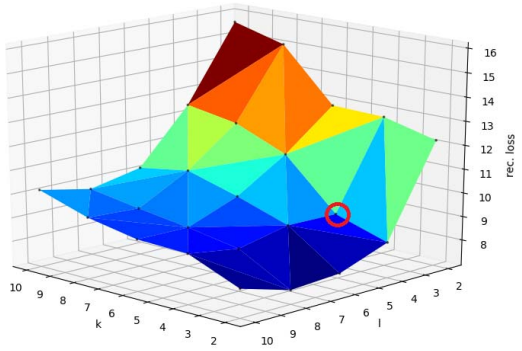

Fig. 6. Landscape of the reconstruction loss, w.r.t. different number of clusters k and different number of subspace dimensionalities l. An exceptional minimum is visible at $k = 4, l = 4$ (red circle), which means that the clustering with these parameters enables a good reconstruction of the data with comparably low number of dimensions and low number of clusters. $k = 4, l = 4$ is an exceptional minimum in the sense that it leads to a lower reconstruction error despite having lower number of clusters k and at the same lower dimensionality l.

cluster. First, we constructed distance plots where for each entry in the plot the distance of a compound to the medoid of its cluster is computed. The entries in the plot are sorted in ascending order by their distance to the medoid. These plots reveal potential outliers within a cluster. Here we have the hypothesis that single highly diverse compounds end up in the clusters, since ORCLUS does not provide a native

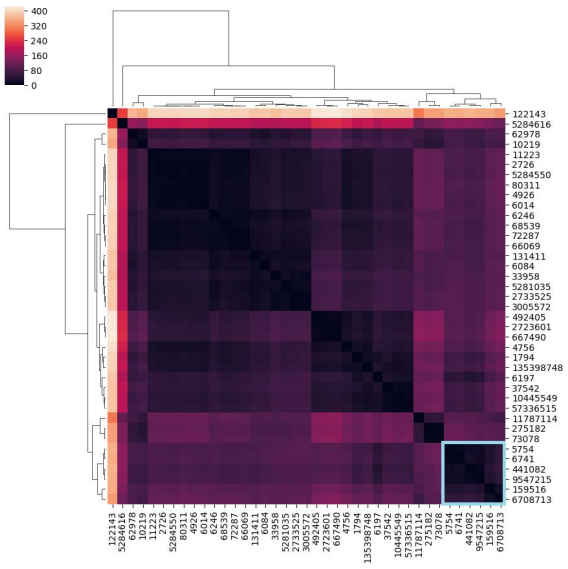

Fig. 7. Clustermap of the result from ORCLUS fir $k = 2, l = 4$ in c_1. The clustermap was created with ward hierarchical clustering on the $l = 4$ dimensional projected compounds.

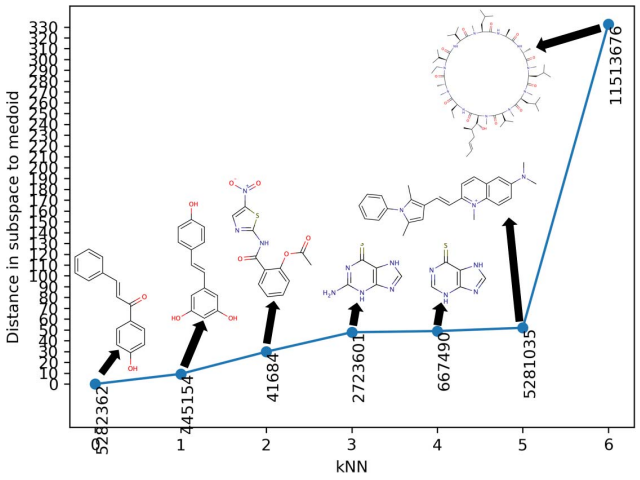

Fig. 8. Projected distance of each compound from cluster c_9 of $k = 16, l = 4$ within the 4-dimensional subspace to the medoid of the cluster (5282362). The outlier (11513676) is by a factor of six further distant to the medoid compared to the second most far compound.

outlier handling. These outliers make the detection of common (sub)structures within a cluster significantly difficult since it is not obvious which compounds should be less considered for the common (sub)structure detection. Further we had the hypothesis that smaller groups within each cluster may exhibit high similarities within their small local groups but not w.r.t. the entire cluster. In order to identify these smaller subgroups, we computed cluster maps which reveal smaller subgroups within a cluster, with the hope of facilitating the common (sub)structure detection.

C. Insights of the in-depth Analysis

From the in-depth analysis, we obtained results which we discuss in detail in this subsection. By analyzing different k we made an interesting observation in c_1 in the clustering of $k = 2, l = 4$. There are good efforts visible if we look closer at the subgroups (which are actually clusters *within* the subspace of c_1). The results of ORCLUS are comparable to clusters of k-means and hierarchical clustering and the "ground truth" observed for higher k. As an explicit example, one can observe the steroid subgroup in the cluster map of Figure 7 (blue rectangle) which is also detected by k-means and hierarchical clustering. These are the first indications that ORCLUS is potentially capable of detecting meaningful clusters with lower-dimensional representations of the vector embeddings.

While the results are improving with increasing k for k-means and hierarchical clustering, the results are getting worse for ORCLUS. For increasing k we observe that highly similar molecules end up in different clusters. As an example we have identified two compounds, namely Toremifene (3005572) and Tamoxifen (2733525). Both compounds differ by just one atom. From their global structure, both compounds are highly similar to each other. In the ORCLUS clustering of $k = 10, l = 8$ we observe them in different clusters (c_1 and c_9). As a small experiment, we first computed the $l = 8$ dimensional subspace of c_1 and projected Toremifene and Tamoxifen to this respective subspace. The distance of these two compounds to each other is 6.16. Computing the subspace of c_9, projecting both compounds to the subspace of c_9 and computing in this subspace their distances to each other yields 4.37. From this observation, we can draw the conclusion that ORCLUS tears subsets of compounds apart that are *even within both projected subspaces* highly similar to each other. Possible reasons for this behaviour can be the following: (1) different initialization of ORCLUS may lead to cases where similar objects are assigned to different subspaces and (2) outliers can, depending on their distance to the hyperplane on which most of the other compounds are located around, heavily skew the orientation and translation of the subspace, which leads to an effect that can be described as "subspace poisoning". In [23] the authors already made the observation that the covariance for a PCA is potentially susceptible to outliers. As an example we can observe in Figure 8 for $k = 16, l = 4$ a cluster where one compound has an obviously high distance to the clusters medoid in contrast to the rest of the compounds within the cluster. Here a second factor plays also an important role: it is not only the distance of the outlier to the subspace but also the number of objects a cluster contains. A distance of 300 to the medoid is less severe for the orientation of the subspace for a cluster that contains 100 compounds, compared to a cluster with only 5 compounds. To summarize, for smaller k we can observe that ORCLUS enables the detection of subgroups, however, it fails to maintain the subgroups for higher k, which is traceable to the compromising effects of outliers.

Regarding the analysis of the results for different subspace dimensionalities l it leaves room for the hypothesis that with $l = 4$ this may be a sufficient number of dimensions to cluster the compounds somewhat meaningful at $k = 2$. This in turn implies that $\frac{4}{300} \approx 1\%$ of the feature information capture a certain amount of the molecular structure which would speak for efforts to compress the data to lower dimensions using subspace clustering. However the dataset contains many different unique molecules or at best small groups of molecules which are different to each other, and as such should form their own small clusters. Due to this observation, one is tempted to choose a high k, which is not an option since it is too sensitive towards outliers. At the same time, one can not simply increase l since for testing larger l it has to be ensured that there are sufficiently ($\geq l$) similar objects within a dataset for the majority of clusters. Conclusively the dataset limits the choice of k and l significantly and for the conducted k, l-settings the results are not further usable in this current state for an application on this dataset.

V. Conclusion and Future Work

In this work, we constructed an entire pipeline and curated a new dataset on potentially effective SARS, MERS, COVID-19 compounds. This allowed us to investigate the meaningfulness of clusterings on Mol2Vec embeddings and the applicability of subspace clustering for learning lower-dimensional representations of the embeddings. The full-dimensional clustering methods, k-means and hierarchical clustering yielded more meaningful results than subspace clustering. Our analysis identified several reasons why utilizing subspace clustering to compress the vectors to fewer dimensions could lead to this insufficient performance. The small size of the dataset and the resulting small clusters limit the investigation of the full k, l-parameter landscape for the subspace clustering approach and it increases the cluster's susceptibility towards outliers. Using a larger dataset can result in larger clusters that allow exploration of higher subspace dimensionalities l and a higher number of clusters k. Other methods addressing the outlier problem are more robust subspace clustering methods or a denoising step up-stream to ORCLUS, which can also be applied in case a large dataset is not at disposal. Adding the here discussed algorithms to the plethora of clustering methods for chemical compounds has the potential to expand the field of molecule clustering. Furthermore, the future exploration on the possibility of embedding compression and the usage of cluster representatives have the capability for a (substantial) acceleration of similarity searches on databases with vast amounts of compounds. Altogether, this can give the chance to make contributions to research fields associated to chemical structures, for example the field of drug discovery.

Acknowledgements

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibility for its content.

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[2] S. Jaeger, S. Fulle, and S. Turk, "Mol2vec: unsupervised machine learning approach with chemical intuition," *Journal of chemical information and modeling*, vol. 58, no. 1, pp. 27–35, 2018.

[3] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[4] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[5] C. C. Aggarwal and P. S. Yu, "Finding generalized projected clusters in high dimensional spaces," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 70–81.

[6] T. Pillaiyar, S. Meenakshisundaram, and M. Manickam, "Recent discovery and development of inhibitors targeting coronaviruses," *Drug discovery today*, vol. 25, no. 4, pp. 668–688, 2020.

[7] A. Fischer, M. Sellner, S. Neranjan, M. A. Lill, and M. Smieško, "Inhibitors for novel coronavirus protease identified by virtual screening of 687 million compounds," *ChemRxiv Prepr*, 2020.

[8] G. Li and E. De Clercq, "Therapeutic options for the 2019 novel coronavirus (2019-ncov)," 2020.

[9] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, "Open babel: An open chemical toolbox," *Journal of cheminformatics*, vol. 3, no. 1, p. 33, 2011.

[10] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[11] Q. Zou, P. Xing, L. Wei, and B. Liu, "Gene2vec: gene subsequence embedding for prediction of mammalian n6-methyladenosine sites from mrna," *Rna*, vol. 25, no. 2, pp. 205–218, 2019.

[12] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, 2010.

[13] J. W. Raymond and P. Willett, "Maximum common subgraph isomorphism algorithms for the matching of chemical structures," *Journal of computer-aided molecular design*, vol. 16, no. 7, pp. 521–533, 2002.

[14] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Transactions on computers*, vol. 100, no. 11, pp. 1025–1034, 1973.

[15] T. T. Tanimoto, "Elementary mathematical theory of classification and prediction," 1958.

[16] G. W. Bemis and M. A. Murcko, "The properties of known drugs. 1. molecular frameworks," *Journal of medicinal chemistry*, vol. 39, no. 15, pp. 2887–2893, 1996.

[17] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *Acm transactions on knowledge discovery from data (tkdd)*, vol. 3, no. 1, pp. 1–58, 2009.

[18] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural computation*, vol. 9, no. 7, pp. 1493–1516, 1997.

[19] D. Kazempour, P. Kröger, and T. Seidl, "Towards an internal evaluation measure for arbitrarily oriented subspace clustering," in *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2020, pp. 300–307.

[20] D. Kazempour, A. Beer, P. Kröger, and T. Seidl, "I fold you so! an internal evaluation measure for arbitrary oriented subspace clustering," in *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2020, pp. 316–323.

[21] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[22] E. Schubert and A. Zimek, "ELKI: A large open-source library for data analysis - ELKI release 0.7.5 "heidelberg"," *CoRR*, vol. abs/1902.03616, 2019. [Online]. Available: http://arxiv.org/abs/1902.03616

[23] D. Kazempour, M. A. X. Hünemörder, and T. Seidl, "On comads and principal component analysis," in *Similarity Search and Applications*. Cham: Springer International Publishing, 2019, pp. 273–280.