# NIPALSTREE: A New Hierarchical Clustering Approach for Large Compound Libraries and Its Application to Virtual Screening†

Alexander Böcker,‡,§ Gisbert Schneider,‡ and Andreas Teckentrup*,§

Institut für Organische Chemie und Chemische Biologie, Johann Wolfgang Goethe-Universität, Marie-Curie-Strasse 11, D-60439 Frankfurt, Germany, and Department of Lead Discovery, Boehringer Ingelheim Pharma GmbH & Co. KG, Birkendorfer Strasse 65, D-88397 Biberach a.d. Riss, Germany

Received December 21, 2005

A hierarchical clustering algorithm—NIPALSTREE—was developed that is able to analyze large data sets in high-dimensional space. The result can be displayed as a dendrogram. At each tree level the algorithm projects a data set via principle component analysis onto one dimension. The data set is sorted according to this one dimension and split at the median position. To avoid distortion of clusters at the median position, the algorithm identifies a potentially more suited split point left or right of the median. The procedure is recursively applied on the resulting subsets until the maximal distance between cluster members exceeds a user-defined threshold. The approach was validated in a retrospective screening study for angiotensin converting enzyme (ACE) inhibitors. The resulting clusters were assessed for their purity and enrichment in actives belonging to this ligand class. Enrichment was observed in individual branches of the dendrogram. In further retrospective virtual screening studies employing the MDL Drug Data Report (MDDR), COBRA, and the SPECS catalog, NIPALSTREE was compared with the hierarchical *k*-means clustering approach. Results show that both algorithms can be used in the context of virtual screening. Intersecting the result lists obtained with both algorithms improved enrichment factors while losing only few chemotypes.

## INTRODUCTION

Virtual screening has become one cornerstone technique in early drug discovery. In this context a plethora of methods has been proposed, and they have proven their usefulness in discovering new lead structures.[1] These methods can generally be classified into ligand-based and structure-based virtual screening approaches.[2−5] One successfully applied type of ligand-based virtual screening is similarity searching.[6−8] It requires a suitable description of molecules and a searching technique that employs different similarity indices. Much effort has been put into the description of molecules with the aim to allow bioisosteric replacements to discover new scaffolds which are not protected by patents.[6−8] When several ligands are available as reference for similarity searching, ensemble techniques have produced activity-enriched hit lists.[9−11] Another way to combine more than one search is to apply clustering techniques on the entire data set, which contains both the reference and the screening substances, and analyze the clusters containing the known actives.

Numerous clustering techniques have been proposed and compared to each other.[12−16] They can be separated into hierarchical clustering methods—like Ward's Clustering as a prominent member[15]—and nonhierarchical clustering techniques such as Jarvis Patrick clustering,[16] *k*-means clustering,[12] or Bayes' unsupervised clustering.[12] Hierarchical clustering techniques establish a relation between molecules, allowing to zoom into or out of data clusters. Compared to nonhierarchical clustering techniques such a relation is usually obtained by calculating a distance matrix, leading to quadratic complexity of calculation time and space.[12] In contrast, nonhierarchical methods are computationally more efficient and scale with $O(n \cdot c)$ for the calculation time and with $O(n)$ for the space requirement, with $n$ being the number of molecules and $c$ being the predefined number of clusters. When choosing the final amount of clusters, a tradeoff must be found to avoid either large and heterogeneous or small and exclusive clusters.[12] To circumvent these shortcomings of nonhierarchical clustering methods several improvements have been proposed: Singletons can be added to the clusters in a fuzzy way,[17] or reclustered using milder boundary conditions,[18] or added to clusters based on maximum common substructure comparisons.[19] Constantly new methods and ideas are introduced, for example building phylogeny-like trees by employing maximum common substructures[20] or clustering data sets according to the frequency of substructure elements.[21]

We recently reimplemented a hierarchical clustering technique—hierarchical *k*-means—that was shown to be able to deal with more than 800 000 data points in high-dimensional descriptor space in less than an hour computation time.[22−24] In the present article we present a second hierarchical clustering technique with the same capability. The algorithm is based on principle component analysis (PCA) using the NIPALS algorithm.[25] Since a hierarchical dendrogram-like structure is constructed for the clusters, the algorithm is termed NIPALSTREE. The schematic outcome of a binary treelike structure is shown in Figure 1 (right side). The left side of Figure 1 represents the corresponding data objects in a 2D illustration. The algorithm was validated using the Collection of Bioactive Reference Analogues (COBRA)[26] as test data set. In retrospective virtual screening

† Dedicated to Professor Johann Gasteiger.

* Whom correspondence should be addressed. E-mail: andreas.teckentrup@bc.boehringer-ingelheim.com.

‡ Johann Wolfgang Goethe-Universität.

§ Boehringer Ingelheim Pharma GmbH & Co. KG.

10.1021/ci050541d CCC: $33.50 
Published on Web 03/24/2006

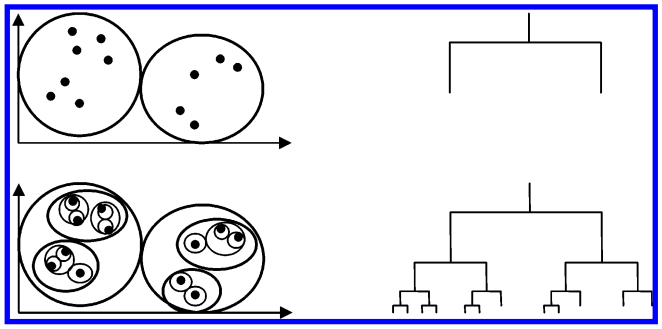

**Figure 1.** Example of binary hierarchical clustering. Nested data objects (left) are represented by a tree structure (right).

studies employing the MDL Drug Data Report (MDDR),[27] COBRA, and the SPECS catalog,[28] NIPALSTREE was compared to the hierarchical *k*-means algorithm. Results show that both algorithms are appropriate for the use in virtual screening applications. In these studies a combination of both techniques even drastically improved the results.

## DATA AND METHODS

**Compound Selection and Preparation.** Three data sets were applied in this study: COBRA,[26] the SPECS catalog,[28] and MDDR.[27] COBRA (Collection of Bioactive Reference Analogues) is a constantly updated small high-quality data set containing 5375 pharmacologically active molecules (version 3.1) taken from the literature. The data set additionally contains information about receptor class, name, and subtype of the target and indication field for each entry. The used version of the COBRA data set contains only druglike molecules, with desalted and formally neutral structures. Consequently, no further filtering steps were applied.

The SPECS catalog (version June 2003) is a vendor data set consisting of 229 658 small organic molecules which can be purchased to build up diverse screening libraries for high-throughput screening (HTS) and lead discovery programs. The used version contains only desalted and neutralized compounds.

The MDDR data set (status August 2003) contains 141 692 biologically relevant structures, reported in the literature.[27] Each entry contains a 2D molecular structure field, of which 2655 are empty, an activity class field, and a corresponding activity class index. The MDDR was prepared as described in ref 22.

**Descriptor Calculation and Pruning.** All compounds of the MDDR, COBRA, and the SPECS catalog were characterized by 2D descriptors provided by the program package MOE.[29] Altogether 146 descriptors are available describing physical properties, subdivided surface areas, atom counts and bond counts, Kier&Hall connectivity and kappa shape indices, adjacency and distance matrices, and pharmacophore features.

When calculating large descriptor sets for a compound collection, two aspects have to be considered. One is redundancy, the other is relevance.[30] Redundancy emerges from the computation of highly correlated descriptors which makes the dimensionality of a data set unnecessarily high. The consequence is an over-representation of the associated properties. Relevance describes whether a descriptor contributes information about the analyzed molecules. One way to ensure relevance is to reject descriptors with low standard deviation since these descriptors contain little information. However, the statistical variance strongly depends on the presence of outliers. To avoid this problem, another way to deal with relevance is to quantify the information content of descriptors by their Shannon entropy (SE).[31−33] This concept was adopted to reject those dimensions having a low information content. The Shannon entropy is defined by eq 1[31]

$$\mathrm{SE} = -\sum_i p_i \log_2 p_i \qquad (1)$$

with $p_i$ giving the probability of the number of data points $c_i$ within a data range $i$ (eq 2):

$$p_i = \frac{c_i}{\sum c_i} \qquad (2)$$

As proposed by Bajorath and co-workers each descriptor value range was subdivided into $N_i = 100$ equidistant data ranges $i$ ("bins").[32,33] Descriptors having only a constant value over the whole data set have no statistical variance and were therefore discarded prior to the calculation of SE. To make the SE independent of the number of bins, the obtained values were normalized by the logarithm to base two of the amount of $N_i$ (eq 3):

$$\mathrm{sSE} = \frac{\mathrm{SE}}{\log_2(N_i)} \qquad (3)$$

Bajorath and co-workers further considered descriptors having a scaled SE (sSE) equal to or less than 0.3 as information-poor.[32,33] In our approach these descriptors were discarded.

To additionally remove redundant dimensions from the remaining descriptor set, unsupervised forward selection (UFS) was performed by Whitely and co-workers.[30] Starting with the two least correlated descriptors, this method builds up a descriptor space by choosing the next descriptor having the lowest multiple correlation coefficient $R^2$ to the current descriptor set. This is repeated until a predefined threshold for $R^2$ is reached. For the COBRA compound collection we applied two thresholds, a conservative value $R^2 = 0.99$ and $R^2 = 0.8$ as a more stringent value. This resulted in two sets of 22 (COBRA08) and 53 (COBRA099) descriptors (Table 2 a). In the following the short names of the data sets will be used. For the comparison of both clustering algorithms

**Table 1:** Reduction of the MDDR Data Set during the Data Preparation Process

| filter | MDDR entries |
|---|---|
| none | 141 692 |
| remove entries lacking structure information | 139 037 |
| remove counterions[a] | 138 584 |
| neutralize and remove non-organic compounds[b] | 136 702 |
| remove reactive and unsuited compounds[c] and compounds with nondrug activity record | 109 528 |

[a] Entries where counterions could not be identified uniquely were discarded. Kensington discovery edition was used to discover and remove counterions.[38] [b] Scitegig Pipeline pilot was used to neutralize the structures and discard nonorganic compounds.[39] [c] The program Filter from OpenEye[40] was adopted according to a publication from Hann and co-workers.[34]

**Table 2.** (a) Data Sets and (b) Descriptor List After Descriptor Pruning

(a) Data Sets

| | COBRA | | SPECS + COBRA + MDDR |
|---|---|---|---|
| descriptor set | MOE2D | | MOE2D |
| original number of descriptors | 146 | | 146 |
| entropy-based pruning[a] | 111 | | 111 |
| UFS $R^2$ threshold | 0.8 | 0.99 | 0.99 |
| UFS pruning | 22 | 53 | 60 |
| final data set name | COBRA08 | COBRA099 | SPECS_COBRA_MDDR |
| threshold $\Theta$[b] | 4.0 | 5.6 | 5.2 |

(b) Descriptor List

| data set name | number of descriptors | descriptor list |
|---|---|---|
| COBRA099 | 53 | Petitjean, petitjeanSC, radius, weinerPath, a_ICM, b_1rotN, b_ar, b_double, b_rotR, a_nN, a_nO, b_heavy, VAdjEq, balabanJ, PEOE_RPC+, PEOE_RPC-, PEOE_VSA+0, PEOE_VSA+1, PEOE_VSA+2, PEOE_VSA+3, PEOE_VSA+4, PEOE_VSA+5, PEOE_VSA-0, PEOE_VSA-1, PEOE_VSA-4, PEOE_VSA-5, PEOE_VSA-6, PEOE_VSA_FNEG, PEOE_VSA_FPNEG, PEOE_VSA_FPPOS, KierA3, a_acc, a_don, vsa_acc, vsa_don, vsa_other, vsa_pol, SlogP_VSA0, SlogP_VSA1, SlogP_VSA2, SlogP_VSA3, SlogP_VSA4, SlogP_VSA5, SlogP_VSA8, SlogP_VSA9, SMR_VSA1, SMR_VSA2, SMR_VSA3, SMR_VSA4, SMR_VSA6, SMR_VSA7, density, logP(o/w) |
| COBRA08 | 22 | petitjeanSC, b_rotR, b_heavy, PEOE_RPC+, PEOE_VSA+2, PEOE_VSA+3, PEOE_VSA+5, PEOE_VSA-0, PEOE_VSA-4, PEOE_VSA-6, PEOE_VSA_FNEG, PEOE_VSA_FPPOS, SlogP_VSA3, SlogP_VSA4, SlogP_VSA5, SlogP_VSA8, SMR_VSA1, SMR_VSA2, SMR_VSA3, SMR_VSA4, SMR_VSA7, density |
| SPECS_COBRA_MDDR | 60 | radius, petitjeanSC, weinerPath, weinerPol, a_ICM, b_1rotN, b_ar, b_double, b_rotR, chi1v, a_nN, a_nO, VAdjEq, balabanJ, PEOE_PC+, PEOE_RPC+, PEOE_RPC-, PEOE_VSA+0, PEOE_VSA+1, PEOE_VSA+2, PEOE_VSA+3, PEOE_VSA+4, PEOE_VSA+5, PEOE_VSA+6, PEOE_VSA-0, PEOE_VSA-1, PEOE_VSA-3, PEOE_VSA-4, PEOE_VSA-5, PEOE_VSA-6, PEOE_VSA_FNEG, PEOE_VSA_FPNEG, PEOE_VSA_FPPOS, Q_VSA_HYD, KierA3, KierFlex, a_acc, a_don, vsa_acc, vsa_don, vsa_other, SlogP, SlogP_VSA0, SlogP_VSA1, SlogP_VSA2, SlogP_VSA3, SlogP_VSA4, SlogP_VSA5, SlogP_VSA7, SlogP_VSA8, SlogP_VSA9, SMR_VSA0, SMR_VSA1, SMR_VSA2, SMR_VSA3, SMR_VSA4, SMR_VSA6, SMR_VSA7, density, logP(o/w) |

[a] Descriptors having a scaled Shannon entropy equal or less 0.3 were rejected. [b] Calculated similarity threshold for clustering.

an additional data set was constructed, combining the descriptor sets from SPECS, COBRA, and MDDR, which were pruned employing the SE and UFS approach ($R^2$ = 0.99). This led to 60 remaining descriptors (Table 2a). The reduced descriptor lists of the three data sets are present in Table 2b.

**NIPALSTREE Clustering.** Many hierarchical clustering techniques exhibit quadratic complexity.[12,13,15] To reduce the complexity of hierarchical clustering we project the descriptor matrix onto one dimension and convert the clustering problem into a sorting problem which scales in $O(n \cdot \log \cdot n)$, with $n$ being the number of data points (e.g. molecules). One such appropriate projection technique is PCA.[25] It is often used to reduce the dimensionality of a data set to a few principle components (PC),[35] where the "axis" direction and the coefficients correspond to the so-called loading vector **L** and the scoring vector **S**, respectively. We employed PCA for our hierarchical clustering algorithm as follows: A $d$-dimensional descriptor matrix is projected onto the first PC. Based on the scoring vector **S**, the given descriptor matrix is sorted in ascending order and split at the median position, i.e., two equally large descriptor sets—from now on termed "left" and "right" submatrix—are created. This is repeated for the new subsets until the maximum distance between the entries in a submatrix underscores a predefined similarity threshold $\Theta$ (for an estimation of $\Theta$ see the chapter after the next chapter).

Two principal shortcomings of this method should be mentioned: One is that topological errors can occur performing the projection. Thus clusters which exist in $d$-dimensional space may be distributed over a broad data range in the first PC. These clusters would be torn apart. A second shortcoming is that splitting at the median may lead to separation of similar entries lying around the median position. The relevance of the first problem can be assessed for each individual case by performing similarity searches in the $d$-dimensional descriptor space after clustering to detect related submatrices. The second led to the current version of the algorithm:

To build up a hierarchical dendrogram, the following steps are recursively performed on the data set:

Step 0: Define a distance threshold $\Theta$. In the present study, the Euclidian metric was employed.

Step 1: Create a copy of the current descriptor matrix. The PCA projection is performed on this data matrix. The original data matrix is sorted according to the values in the scoring vector, and the splitting procedure is initiated with the original data matrix (Figure 2).

Step 2: Generate clusters by splitting. The splitting procedure is illustrated schematically in Figure 2B: Three clusters (black, white, and gray circles) are represented by one descriptor. Splitting at the median position pulls the white cluster apart. Starting from the median position the left and right neighboring entries are examined stepwisely to find a

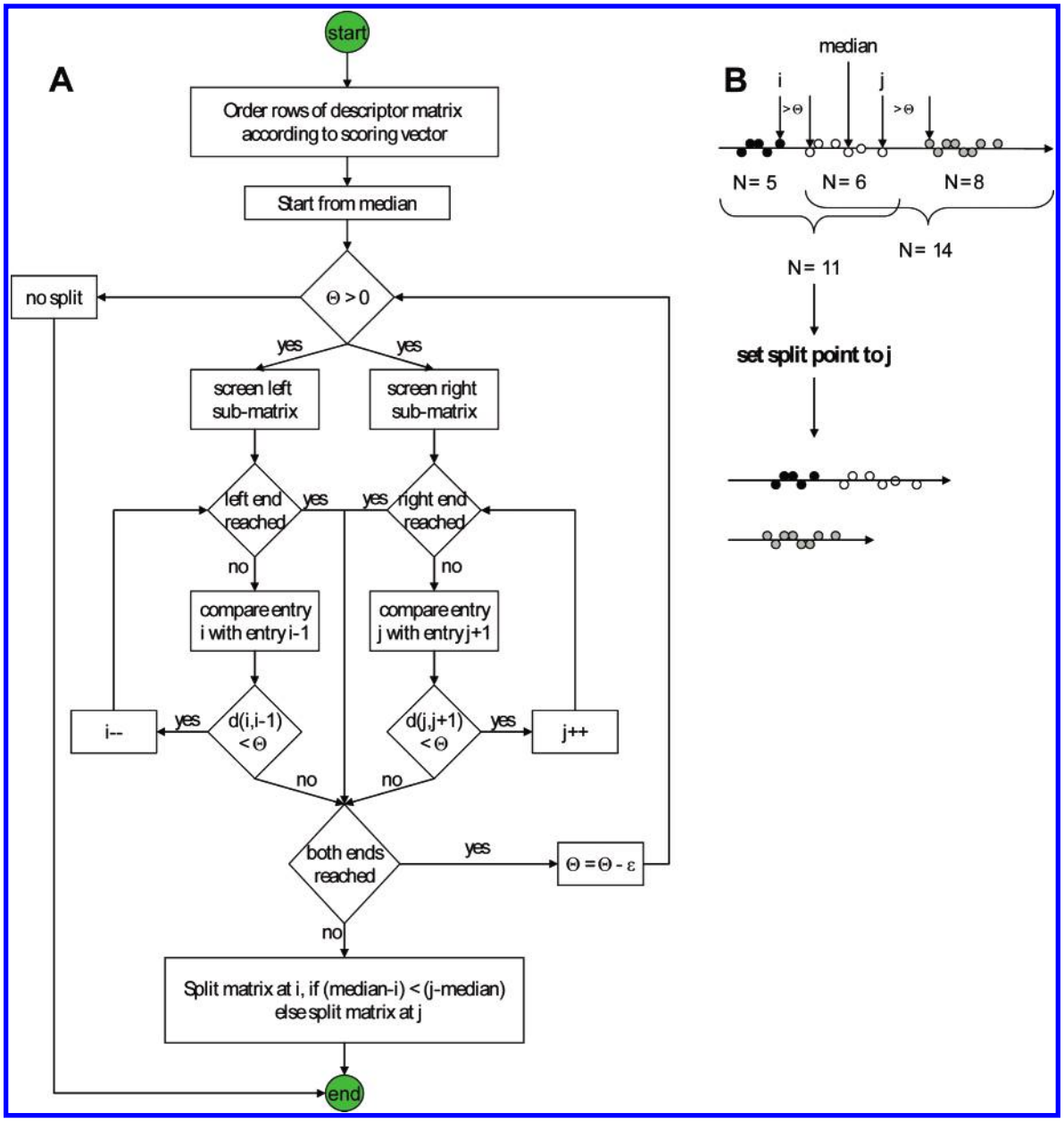


**Figure 2.** Concept of wandering neighbor. A. Algorithmic flowchart. Abbreviations: $d(i,i\text{-}1)$ = Euclidian distance between molecule $i$ and $i$-1, $d(j,j+1)$ = Euclidian distance between molecule $j$ and $j$+1. $\Theta$ = stop threshold, $\epsilon$ = parameter used for systematically lowering the stop threshold. B. Schematic illustration of the concept: Three clusters are present in a one-dimensional data set (black, green, and gray). Splitting at the median position pulls the white cluster apart. Starting from the median position the left and right neighboring entries are examined stepwisely to deduce a better split point $i$ (left side) or $j$ (right side). The point $j$ is used as new split point since no clusters are separated, and the resulting left and right data sets are of mostly equal size ($N$ = 11 left and $N$ = 8 right) compared to splitting at $i$ ($N$ = 14 left and $N$ = 5 right).

better split point $i$ (left side) or $j$ (right side). The point $j$ is used as new split point since no cluster is separated, and the resulting left and right data sets are of comparable size. In detail the procedure is shown in Figure 2A as an algorithmic flowchart: The splitting procedure starts by setting a pointer to the median entry, where the split point is by definition assigned to the right-hand side. Starting from the median position in the original data matrix, the Euclidian distance to the left neighboring entry is calculated. If the distance falls below the threshold $\Theta$, the neighboring entry is defined as the new temporary split point. This is done iteratively until the left end of the matrix is reached or the threshold $\Theta$ is exceeded. The procedure is initiated analogously for the right side starting from the split point lying directly left to the median. For both processes the number of comparisons is counted. If both stepping procedures have reached the end of the data matrix, $\Theta$ is decreased, and the procedure is reinitiated. If $\Theta$ reaches zero no splitting is performed. If the number of comparisons for the left side is larger than for the right side, the splitting position is set to the right temporary split point and vice versa. $\Theta$ is set back to the original value, and the left and right submatrices are created.

Step 3: Check the maximum distance within the new subsets. If the maximum Euclidian distance between the entries in one subset does not exceed the predefined threshold $\Theta$, no further splitting of this matrix is performed. Otherwise the algorithm restarts with the created subsets at step 1.

The herein described algorithms separates the data according to the first PC. This makes it monthetic in nature like recursive partitioning.[36] However in contrast to recursive

partitioning NIPALSTREE is an unsupervised classification technique separating a data set according to its inherent properties present in the loading vector.

NIPLASTREE is able to cluster large data sets with feasible run time behavior and space requirements. The reading, clustering, and displaying of 404 148 molecules with 60 descriptors took 39 min on a Linux workstation employing a 3.2 GHz Intel Xeon Processor and required less than 2 GB of memory. For comparison, the same clustering using the hierarchical *k*-means algorithm (see next chapter), which as well is able to cluster large data sets with minor memory occupation, was 6 times faster and employed 20% less memory.

**Hierarchical *k*-Means Clustering.** The *k*-means algorithm is a nonhierarchical clustering technique.[12] It requires O($k \cdot n$) computation time and internal memory, with $n$ being the number of data points and $k$ being the number of clusters. The *k*-means algorithm randomly selects $k$ data points as initial cluster centroids. The $k$ clusters are formed by assigning each data point to its nearest centroid. New virtual centroids are then calculated for each cluster. The calculations of centroid vectors and the cluster generation are repeated until a predefined number of iterations is reached or the clusters do not change anymore.

To convert the *k*-means algorithm into a hierarchical clustering algorithm, we implemented a modified form of the *k*-means algorithm, i.e., forming $k$ clusters at each level of a hierarchical dendrogram.[22] The basic steps of the modified algorithm are as follows:

Step 0: Define $k$ (For a binary dendrogram, $k = 2$.) and select a distance threshold $\Theta$. The Euclidian metric is employed to define "distance".

Step 1: Perform data clustering for the actual dendrogram level. Starting with the root cluster, $k$ child clusters are created, and the data set is partitioned according to the *k*-means algorithm.

Step 2: Check for each cluster if the maximum distance between the data points exceeds the threshold $\Theta$ and continue with step 1 for this cluster. Otherwise terminate.

It should be noticed that the employed variant of hierarchical *k*-means is a technique using a randomization step during the initialization of the centroid vectors. Thus, in contrast to the NIPALSTREE algorithm, multiple runs on the same data set will not necessarily result in identical dendrograms.

**Defining the Threshold Value $\Theta$ for NIPALSTREE and Hierarchical *k*-Means.** Current hierarchical clustering algorithms use criteria like the homogeneity or the heterogeneity of the resulting clusters to assess the quality of the algorithm.[12] These criteria are used to determine whether clusters containing similar data should be merged. The difficulty is that these techniques require a definition of "similarity". Both the NIPALSTREE and the hierarchical *k*-means algorithm face the same problem since the threshold $\Theta$ has to be defined optionally. The threshold $\Theta$ is defined as the maximum allowed distance between two entries in a terminal cluster. For the hierarchical *k*-means algorithm we have proposed a method helping to find a useful threshold value:[22]

For a specified distance range the number of singletons, the number of clusters, and the sum of the maximum distances in each terminal cluster, $D_{max}$ (eq 4) are calculated, scaled to [0,1], and plotted in one graph (Figure 3). $D_{max}$

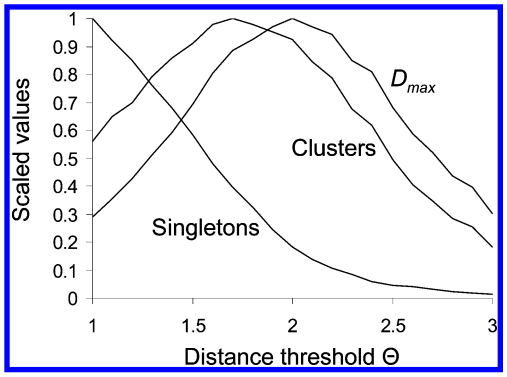


**Figure 3.** Identification of a reasonable value for the distance threshold $\Theta$. Clustering was done for the COBRA08 data set using a subset of all MOE-2D descriptors ($N$=22).

can be interpreted as the sum of the cluster diameters of the $n$ terminal clusters

$$D_{\max} = \sum_i \max_j \left(D(x_{i,j}, c_i)\right) \text{ with } 1 \le j \le N_i \qquad (4)$$

where $D$ is the Euclidian distance, $x_{i,j}$ represents data points that are members of the same terminal cluster $i$, $c_i$ represents the centroid of cluster $i$, and $N_i$ is the number of members of cluster $i$. Finally, the $\Theta$ value that leads to the maximal $D_{max}$ value is used for the clustering (Figure 3). It is a compromise between fine- and coarse-grained partitioning of the data space. The last row of Table 2a summarizes the results obtained for the different data sets used in the present work.

**Statistical Assessment of the Clustering Algorithms.** To judge the quality of a clustering result an index is introduced to assess whether molecules interacting with the same target (receptor or receptor family) lie in the same subtree. We calculated an enrichment factor (EF) for each cluster,[37] which gives an estimate of how well compounds that bind to the same target (or target class) are clustered in a dendrogram node $i$ (eq 5)

$$\mathrm{EF}_{i,c} = \frac{\dfrac{N_{i,c}}{N_c}}{\dfrac{N_i}{N}} \qquad (5)$$

with $N_{i,c}$ being the number of entries in node $i$ belonging to class c, $N_i$ being the total number of entries in node $i$, $N_c$ being the total number of entries of class c in the data set, and and $N$ being the overall number of entries. EF > 1 indicates that more compounds belonging to the activity class c are clustered in a tree node than expected from an equal distribution. The EF value depends on the size of the dendrogram section under consideration: On upper dendrogram levels, where clusters are large, EF values are usually small, whereas EF values on the lower dendrogram levels can get large without statistical relevance. One possible way to overcome the cluster size dependency of the EF is to additionally divide it by the logarithm of the dendrogram level, assuming that at each cluster the data set is separated into equally large partitions. By this an adoption of the EF to the dendrogram level can be achieved.

To obtain a generalized view on the distribution of molecules interacting with the same receptor target in the

**Table 3:** SPECS_COBRA_MDDR: Sizes of Inhibitor/Ligand Classes Used

| label | *N* (COBRA) | *N* (MDDR) |
|---|---|---|
| ACE inhibitor | 48 | 494 |
| COX inhibitor | 149 | 1556 |
| adrenoceptor ligand | 200 | 542 |
| GABA receptor ligand | 85 | 478 |
| glucocorticoid receptor ligand | 18 | 91 |

overall cluster dendrogram, we suggest the following analysis: For a dendrogram level we calculate the average EF of all $n$ ($n$ = number of clusters) EFs of class c, which are larger or equal to one (eq 6).

$$\text{average EF} = \frac{1}{n}\sum_{i} \text{EF}_{i,c};\ \text{EF} \geq 1 \qquad (6)$$

Average enrichment factors are calculated for all dendrogram levels, where the number of clusters is less or equal to the number of molecules interacting with receptors of class c. On higher dendrogram levels, artificially large enrichment factors bias the average. For these levels no average enrichment factor is calculated.

**Software.** Removing structure deficient molecules, adding explicit hydrogens, and setting the atom ionization to the formal charge was done using MOE.[29] Counterions were removed using an approach implemented in Kensington Discovery Edition (InforSense)[38] at Boehringer Ingelheim. Neutralization of structures was done using Pipeline Pilot (SciTegic)[39] or MOE-SVL scripts for special cases.

The filtering of nondrug MDDR entries (e.g. radio sensitizers) was performed using standard data manipulating functions of the Kensington Discovery Edition. Subsequent filtering steps were performed with the program FILTER (OpenEye)[40] or in-house Perl scripts implemented at Boehringer Ingelheim. UFS v.1.8[30] was employed for descriptor pruning of correlated descriptors. The pruning of information-poor descriptors was implemented in C and Java.[41]

The clustering algorithms and their statistical assessment were both implemented in Java. The hierarchical *k*-means program is freely available.[42]

For validation purposes we used the program ClassPharmer (Bioreason)[43] and Daylight fingerprints.[44]

## RESULTS AND DISCUSSION

The NIPALSTREE algorithm was applied to COBRA08 and COBRA099 data sets. Within COBRA we have focused on the inhibitor types listed in Table 3. To discuss the results we have picked one example, i.e., angiotensin converting enzyme (ACE) inhibitors.[45] ACE, a zinc dependent metalloprotease, plays a central role in the angiotensin-renin system. ACE cleaves the decapeptide angiotensin I into the vasopressor angiotensin II. ACE inhibitors are used for treatment of cardiovascular diseases, including high blood pressure, heart failure, and kidney failure.[45]

The compound sets contained 48 molecules categorized as ACE inhibitors. These can be grouped into four structural classes (class representatives are shown in Figure 4, 1−4) and a few "outliers", which cannot easily be assigned to any of the classes (four example structures are shown in Figure 4, 5a−5d). Seven molecules of class 1, 11 molecules of class 2, 12 molecules of class 3, and 6 molecules of class 4 are present in COBRA.

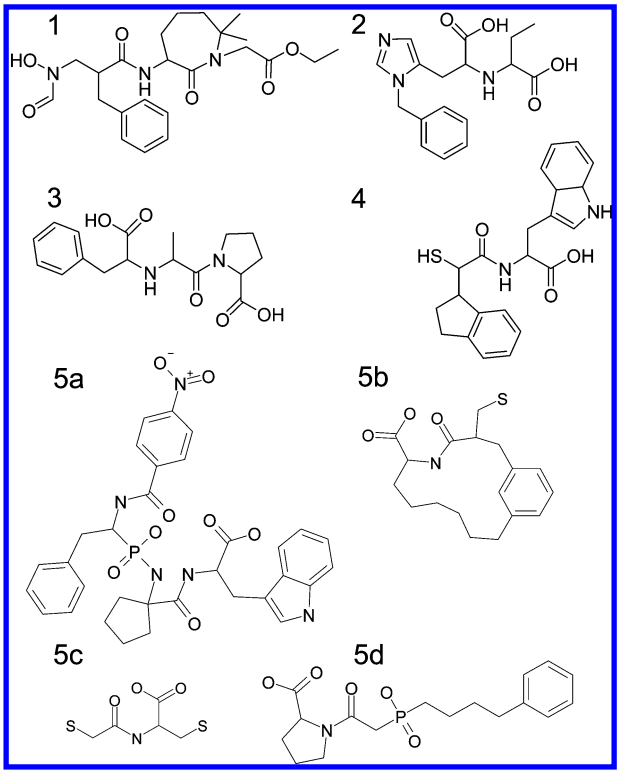


**Figure 4.** Four class representatives (1−4) and four examples of outliers (5a−5d) of ACE inhibitors in the COBRA data set.

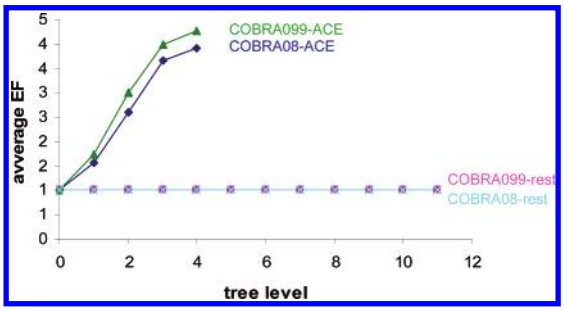


**Figure 5.** Average enrichment factors for each dendrogram level obtained for ACE inhibitors and non-ACE inhibitors (rest) of the COBRA08 and the COBRA099 descriptor sets. Green: COBRA099 ACE inhibitors; dark blue: COBRA08 ACE inhibitors; magenta: COBRA099 non-ACE inhibitors; light blue: COBRA08 non-ACE inhibitors.

We clustered both descriptor versions of COBRA using the NIPALSTREE algorithm. Figure 5 shows the average enrichment factors (eq 6) for each dendrogram level and both COBRA versions, separated in ACE inhibitors and non-ACE inhibitors. The non-ACE inhibitors show an average enrichment factor of 1 (i.e. no enrichment) in the dendrogram. In contrast ACE inhibitors show constantly increasing average enrichment factors for both COBRA versions, when stepping down the dendrogram hierarchy. Therefore going from the uppermost cluster to clusters located on deeper levels, ACE inhibitor structure classes are enriched employing the clustering algorithm and both descriptor sets.

We analyzed the separation of the four ACE inhibitor classes in the resulting dendrograms. On the root level most "outliers" were assigned to the right subdendrogram, whereas the four structural classes were completely assigned to the left subtree. On dendrogram level four employing CO-

**Table 4:** Mean Descriptor Value and Standard Deviation for ACE Inhibitors of Class 2 and Classes 1, 3, and 4

| | PEOE_VSA-0 | PEOE_VSO-4 | SlogP_VSA1 | SMR_VSA1 |
|---|---|---|---|---|
| class 2 | 32.9 ± 15.1 | 5.68 | 35.7 ± 3.3 | 18.7 |
| classes 1, 3, 4 | 51.8 ± 15.9 | 0 | 21.6 ± 4.8 | 3.49 ± 2.9 |

**Table 5:** Occurring Terminal Clusters Enriched with ACE Inhibitors of COBRA099 or COBRA08

| data set | ACE class | *N* (ACE class) | *N* (total) | EF (ACE class) |
|---|---|---|---|---|
| COBRA099 | 2 | 4 | 5 | 391 |
| COBRA099 | 2 | 4 | 9 | 217 |
| COBRA099 | 3 | 4 | 6 | 299 |
| COBRA099 | 3 | 4 | 5 | 358 |
| COBRA099 | 1 | 3 | 5 | 461 |
| COBRA08 | 3 | 5 | 7 | 320 |
| COBRA08 | 2 | 4 | 7 | 279 |
| COBRA08 | 3 | 3 | 8 | 168 |

BRA099 and level five employing COBRA08 class 2 was the first one that was separated from the other classes. Class 2 substances differ from the other classes in having an imidazole substructure element, whereas an amide moiety is completely missing. We compared the loading vectors of the class 2 containing clusters and their brother clusters. For COBRA08 major differences in descriptor weighting were present for PEOE_VSA-0, PEOE_VSA-4 and SMR_VSA1 and for COBRA099 for PEOE_VSA-4 and SlogP_VSA1, respectively. We analyzed the corresponding original descriptor values of class 2 and classes 1, 3, and 4. Resulting means and standard deviations are presented in Table 4. Marked differences were obtained for class 2 compared to the other ACE classes. This shows that by analyzing loading vectors in class separating dendrogram clusters important descriptors can be identified.

We further analyzed the resulting terminal clusters obtained with both descriptor sets and the employed termination thresholds. Since here the interest lies on separating structural classes, clusters were not judged according to identified ACE inhibitors but according to the number of identified class members and the enrichment factors of the corresponding classes. The results are summarized in Table 5.

The terminal clusters obtained for the COBRA099 descriptor set show two clusters enriched with class 3, one cluster enriched with class 1, and two clusters enriched with class 2 ACE inhibitors. Class 4 ACE inhibitors occur mainly as singletons. The COBRA08 terminal clusters show two clusters enriched with class 3 and one cluster enriched with class 2 ACE inhibitors, whereas class 1 and class 4 inhibitors occur mainly as singletons. This reflects a characteristic of the NIPALSTREE algorithm, which is a consequence of keeping the use of internal memory as low as possible: via the projection of a *d*-dimensional space onto one dimension mapping errors occur. As a consequence, closely related molecules may appear as singletons in different parts of the resulting dendrograms. This problem can be fixed e.g. by performing additional similarity searches around the resulting terminal cluster centroids.

In general the results show that a suitable clustering of ACE inhibitors can be obtained using the NIPALSTREE algorithm. Although differences exist, clustering both descriptor sets led to a comparable separation of the molecules

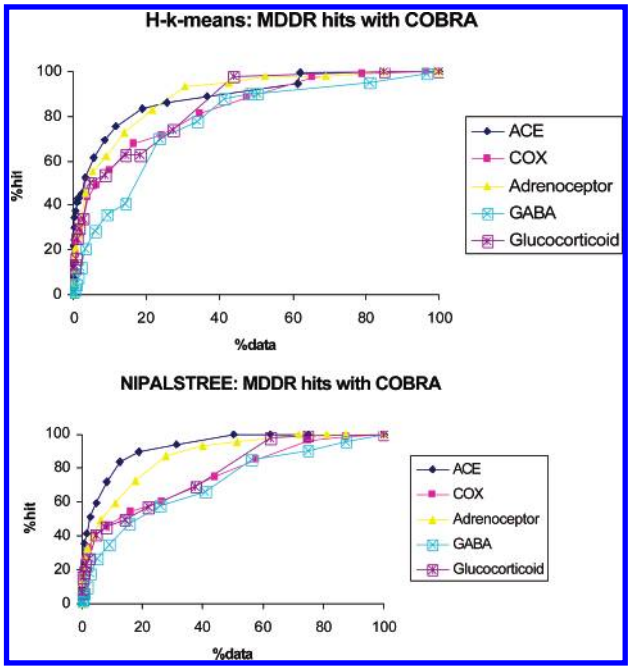


**Figure 6.** Clustering of the SPECS_COBRA_MDDR data set using NIPALSTREE (left) and hierarchical *k*-means (right) and specific labels (see color panels). On a dendrogram level the number of data points in clusters containing COBRA entries having a specific label are translated into the percentage of virtually screened compounds. The number of coclustered MDDR entries having the same label are translated into the percentage of retrieved hits. Points in the diagram correspond to dendrogram levels.

in the dendrogram, and no clear descriptor preference can be given. Screening through the terminal clusters enriched with ACE inhibitors, six additional protease inhibitors and five molecules binding to other receptor classes were found for COBRA099. In the case of COBRA08 seven additional protease inhibitors were found and five molecules binding to other receptors.

**Virtual Screening Using SPECS, COBRA, and MDDR.** The results encouraged us to compare NIPALSTREE with the previously published hierarchical *k*-means algorithm,[22] which -- like NIPALSTREE -- has the capability to hierarchically cluster data sets with more than 1 million data points. To examine the usefulness of both algorithms in the context of virtual screening we decided to construct a combined data set of the SPECS catalog, COBRA and MDDR. The SPECS catalog was employed to increase the number of molecules for which the activity is unknown. These molecules are treated as "inactive". With the combined data the hierarchical clustering was performed using both algorithms. Additionally the inhibitor classes listed in Table 3 were analyzed and used as compound labels.

We compared the quality of both clustering algorithms by examining for each dendrogram level what percentage of MDDR entries with a certain label have been coclustered with COBRA entries bearing the same label. By summing up the cluster sizes of the examined clusters, the screened percentage of the original data set was additionally derived. Enrichment curves were created by plotting the percentage of retrieved MDDR entries having a certain label against the percentage of the screened data set for each dendrogram level. Figure 6 shows the enrichment curves for both algorithms and all label classes. All curves show a steep rising in the lower percentage range (deeper dendrogram

**Table 6:** Enrichment Factors Obtained with the Clustering Algorithms on Dendrogram Level 11

| | ACE ($N = 494$)[a] | COX ($N = 1556$)[a] | adrenoceptor ($N = 542$)[a] | glucocorticoid receptor ($N = 91$)[a] | GABA-receptor ($N = 478$)[a] |
|---|---|---|---|---|---|
| hierarchical *k*-means | 31.2 ($N = 246$)[b] | 11 ($N = 761$)[b] | 8.99 ($N = 298$)[b] | 27.3 ($N = 27$)[b] | 7.97 ($N = 110$)[b] |
| NIPALSTREE | 16.2 ($N = 188$)[b] | 6.16 ($N = 625$)[b] | 6.14 ($N = 270$)[b] | 18.7 ($N = 17$)[b] | 3.51 ($N = 84$)[b] |
| hierarchical *k*-means + NIPALSTREE disjunction | 17.6 ($N = 306$)[b] | 6.42 ($N = 980$)[b] | 5.93 ($N = 394$)[b] | 15.6 ($N = 30$)[b] | 4.86 ($N = 165$)[b] |
| hierarchical *k*-means + NIPALSTREE conjunction | 54 ($N = 128$)[b] | 22.6 ($N = 406$)[b] | 16.3 ($N = 174$)[b] | 98 ($N = 14$)[b] | 7.77 ($N = 29$)[b] |

[a] In parentheses the total number of MDDR ligands in the data set is shown. [b] In parentheses the number of MDDR ligands is shown retrieved with the corresponding method.

**Table 7:** MDDR ACE Inhibitor Sets

| data set[a] | final data set name | no. of MDDR ACE inhibitors | ClassPharmer classes |
|---|---|---|---|
| hierarchical *k*-means | set 1 | 246 | 39 |
| NIPALSTREE | set 2 | 188 | 29 |
| hierarchical *k*-means + NIPALSTREE conjunction | set 3 | 128 | 24 |

[a] Data sets contain only MDDR ACE inhibitors found with both algorithms on tree level 11.

levels) for all labels. The markers in the curves correspond to dendrogram levels, with the right-most point being the root of the dendrogram.

To compare both clustering algorithms we selected dendrogram level 11. On this level the screened data size of each selected screening application was above 3500 compounds which is a suitable size for further filtering steps, ordering, and experimentally testing of the molecules. Enrichment factors were calculated for both algorithms and the disjunctions and conjunctions of the result lists of both algorithms. Table 6 shows the obtained enrichment factors. In parentheses the number of retrieved MDDR entries is shown, interacting with the examined receptor class.

The results show that for all listed examples the hierarchical *k*-means algorithms seems to perform better than the NIPALSTREE algorithm, since all calculated enrichment factors are higher. However it can be seen that with the exception of GABA receptor ligands, for the conjunctive combination of both algorithms the enrichment factor rises at least 2-fold. Reducing the data space by intersecting the results of the two algorithms can have the consequence of losing structural classes. To test how many structural classes get lost in combining both algorithms we carefully reanalyzed the distribution of ACE inhibitors on level 11. The employed ACE inhibitor sets are presented in Table 7.

To identify the structural classes present in the data sets, a Bioreason ClassPharmer[43] analysis was performed. This program is a hierarchical cluster analysis tool extracting maximum common substructures (MCS) of a given data set. The resulting classes are assumed to correspond closest to lead structure classes. Table 7 shows, that after combining results of both algorithms (set 3), 15 ClassPharmer classes do no longer occur with respect to set 1 (hierarchical *k*-means) or five classes with respect to set 2 (NIPALSTREE).

To estimate whether or not occurring classes represent a loss of real "lead" classes, we calculated Daylight Fingerprints for all MCS of the classes of sets 1−3. We then identified for each rejected MCS of set 1 or 2 the most similar MCS in set 3 using Tanimoto similarity calculations.

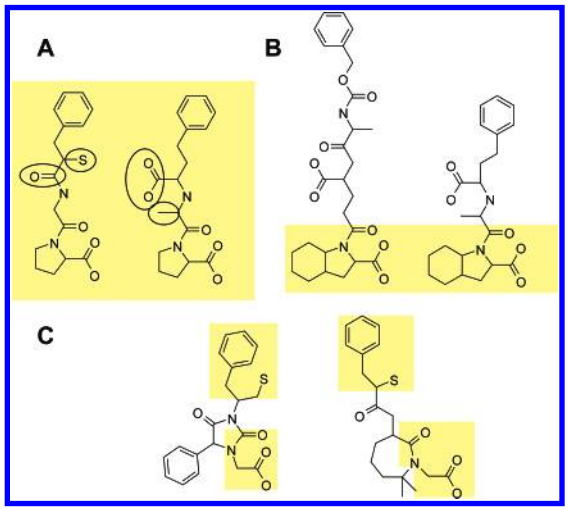


**Figure 7.** Example of three most similar MCS pairs obtained by ClassPharmer analysis for set 3 (left side of each structure pair) and set 1 (right side of each structure pair). Similarity was determined by calculating Tanimoto coefficients employing Daylight Fingerprints. Similar regions in the structure are highlighted in yellow.

Three representative examples of the emerging nearest neighbor pairs are highlighted in Figure 7. All structures contain the common theme of a carboxyl and an amide group separated by an aliphatic carbon atom. However differences exist in the adjacent moieties. The left side of the structure pairs represents a nonmatching class of set 1 (hierarchical *k*-means) and the right side the most similar class of set 3. Figure 7A shows the case where only minor differences exist between the structural pairs. This occurred for nine of the nonmatching classes from data set 1 and three from set 2. Figure 7B exemplifies the case where the core "lead" structure with the common theme is still equal in both structure pairs, but the rest of the groups show differences. This was observed for five of the not matching classes of set 1 and for two from set 2. Figure 7C shows the only example where we think a structural class is lost, since on the left side the hydantoine core structure is replaced by a tetrahydroazepinone on the right side. This structural arrangement is present in four structures of set 2.

In summary the results demonstrate that combining both algorithms reduces the number of compounds to be screened, while the number of actual hits is diminished to a lower extent. For example, on level 11 of the dendrogram for ACE

inhibitors, only one "lead" class gets lost. This effect may be different for other ligand classes. However, current results make us confident that both algorithms are likely to produce overlapping but to some extend different clusters.

## CONCLUSION

We have presented a new hierarchical clustering algorithm, NIPALSTREE, which can deal with large data sets in high-dimensional space. The algorithm projects a data set onto one dimension by PCA. The data set is sorted according to this one dimension (first PC) and split at the median position. To avoid distortion of clusters lying around the median position, the algorithm searches additionally for a better split point left and right of the median. The algorithm was validated using ACE inhibitors in the COBRA data set and was shown to produce meaningful results.

In retrospective virtual screening applications NIPALSTREE was compared with the hierarchical *k*-means algorithm. Both algorithms produced useful results for the selected ligand classes. The hierarchical *k*-means algorithm seems to produce higher enrichment factors on deeper dendrogram levels in the chosen examples. One clear advantage of the hierarchical *k*-means algorithm is that the data need not be split into equally large partitions. Thus extreme outlying data points can be separated relatively early from the rest of the data, which gives valuable information about the data set. This observation cannot be made for the NIPALSTREE algorithm, since here the data set is split in fractions of almost the same size. An advantage of the NIPALSTREE algorithm is that it is deterministic. This is not the case for the applied variant of the hierarchical *k*-means algorithm, since it starts with a random selection of the initial cluster centroids and the separation is performed according to the resulting centroids. A possibility to circumvent this random step is to initially select the most dissimilar entries. However this requires additional calculation steps increasing calculation time. A second advantage of the NIPALSTREE algorithm is that for each cluster the first loading vector is calculated. This gives a hint which descriptors play an important role in forming a cluster. Moreover if one compares the descriptor influences of a cluster with the corresponding descriptor influences in its brother cluster, descriptors can be identified that have a higher impact on one side. By that, rough structure−activity relationships (SAR) for directly related clusters can be obtained.

Intersecting the result lists obtained by both algorithms improved enrichment factors while losing only few chemotypes. This is promising in the context of virtual screening where result lists can easily exceed the required maximum number of compounds for pharmacological testing.

**Abbreviations.** 2D: two-dimensional; ACE: angiotensin converting enzyme; COBRA: collection of bioactive reference analogues; COX: cyclooxygenase; EF: enrichment factor; GABA: gamma amino butyric acid; GB: gigabyte; GHz: gigahertz; GUI: graphical user interface; HTS: high-throughput screening; MCS: maximum common substructures; MDDR: MDL Drug Data Report; MOE: molecular operating environment; PC: principle component; PCA: principle component analysis; RAM: random access memory; SAR: structure−activity relationship; SE: Shannon entropy; sSE: standardized Shannon entropy; SVL: support vector language; UFS: unsupervised forward selection.

## ACKNOWLEDGMENT

This research was supported by the Beilstein Institut zur Förderung der Chemischen Wissenschaften and the Fonds der Chemischen Industrie (FCI).

## REFERENCES AND NOTES

(1) Böhm, H.-J.; Schneider, G. *Virtual Screening for Bioactive Molecules*; Wiley-VCH: Weinheim, 2000.
(2) Bajorath, J. Integration of virtual and high-throughput screening. *Nat. Rev. Drug Discovery* **2002**, *1*, 882−894.
(3) Kitchen, D. B.; Decornez, H.; Furr, J. R.; Bajorath, J. Docking and Scoring in Virtual Screening for Drug Discovery: Methods and Applications. *Nat. Rev. Drug Discovery* **2004**, *3*, 935−949.
(4) Gohlke, H.; Klebe, G. Approaches to the Description and Prediction of the Binding Affinity of Small-Molecule Ligands to Macromolecular Receptors. *Angew. Chem. Int. Ed.* **2002**, *41*, 2644−2676.
(5) Wilton, D.; Willett, P.; Lawson, K.; Mullier, G. Comparison of Ranking Methods for Virtual Screening in Lead-Discovery Programs. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 469−474.
(6) Renner, S.; Schneider, G. Fuzzy Pharmacophore Models from Alignments for Correlation-Vector-Based Virtual Screening. *J. Med. Chem.* **2004**, *47*, 4653−4664.
(7) Lloyd, D. G.; Buenemann, C. L.; Todorov, N. P.; Manallack, D. T.; Dean, P. M. Scaffold Hopping in De Novo Design. Ligand Generation in the Absence of Receptor Information. *J. Med. Chem.* **2004**, *47*, 493−496.
(8) Harper, G.; Bravi, G. S.; Pickett, S. D.; Hussain, J.; Green, D. V. S. The Reduced Graph Descriptor in Virtual Screening and Data-Driven Clustering of High-Throughput Screening Data. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 2145−2156.
(9) Hert, J.; Willett, P.; Wilton, D. J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. Comparison of Fingerprint-Based Methods for Virtual Screening Using Multiple Bioactive Reference Structures. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1177−1185.
(10) Whittle, M.; Gillet, V. J.; Willett, P.; Alex, A.; Loessl, J. Enhancing the Effectiveness of Virtual Screening by Fusing Nearest Neighbour List: A Comparison of Similarity Coefficients. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1840−1848.
(11) Merkwirth, C.; Mauser, H.; Schulz-Gasch, T.; Roche, O.; Stahl, M.; Lengauer, T. Ensemble Methods for Classification in Cheminformatics. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1971−1978.
(12) Jain, A. K.; Murty, M. N.; Flynn, P. J. Data Clustering: A Review. *ACM Comput. Surv.* **1999**, *31*, 265−323
(13) Brown, R. D.; Martin, Y. C. Use of Structure−Activity Data To Compare Structure-Based Clustering Methods and Descriptors for Use in Compound Selection. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 572−584.
(14) Willett, P.; Winterman, V.; Bawden, D. Implementation of Nonhierarchic Cluster Analysis Methods in Chemical Information Systems: Selection of Compounds for Biological Testing and Clustering of Substructure Search Output. *J. Chem. Inf. Comput. Sci.* **1986**, *26*, 109−118.
(15) Ward, J. H. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **1963**, *58*, 236−244.
(16) Jarvis, R. A.; Patrick, E. A. Clustering using a similarity measure based on shared nearest neighbors. *IEEE Trans. Comput.* **1973**, *22*, 1025−1034.
(17) Doman, T. N.; Cibulskis, J. M.; Cibulskis, M. J.; McCray, P. D.; Spangler, D. P. Algorithm5: A Technique for Fuzzy Similarity Clustering of Chemical Inventories. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 1195−1204.
(18) Menard, P. R.; Lewis, R. A.; Mason, J. S. Rational Screening Set Design and Compound Selection: Cascaded Clustering. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 497−505.
(19) Stahl, M.; Mauser, H. Database Clustering with a Combination of Fingerprint and Maximum Common Substructure Methods. *J. Chem. Inf. Model.* **2005**, *45*, 542−548.
(20) Nicolaou, C. A.; Tamura, S. Y.; Kelley, B. P.; Bassett, S. I.; Nutt, R. F. Analysis of large screening data sets via adaptively grown phylogenetic-like trees. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1069−1079.
(21) Roberts, G.; Myatt, G. J.; Johnson, W. P.; Cross, K. P.; Blower, P. E., Jr. LeadScope: software for exploring large sets of screening data. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1302−1314.

(22) Böcker, A.; Derksen, S.; Schmidt, E.; Teckentrup, A.; Schneider, G. A Hierarchical Clustering Approach for Large Compound Libraries. *J. Chem. Inf. Model.* **2005**, *45*, 807−815.
(23) Barnard, J. M.; Downs, G. M.; Wild, D. J.; Wright, P. M. Better Clusters Faster. *Third Joint Sheffield Conference on Chemoinformatics* **2004**.
(24) Sultan, M.; Wigle, D. A.; Cumbaa, C. A.; Maziarz, M.; Glasgow, J.; Tsao, M. S.; Jurisica, I. Binary tree-structured vector quantization approach to clustering and visualizing microarray data. *Bioinformatics* **2002**, *18*, 111−119.
(25) Otto, M. *Chemometrics. Statistics and Computer Application in Analytical Chemistry;* Wiley-VCH: Weinheim, 1998**.**
(26) Schneider, P.; Schneider, G. Collection of Bioactive Reference Compounds for Focused Library Design. *QSAR Comb. Sci.* **2003**, *22*, 713−718.
(27) MDL Drug Data Report; Elsevier MDL: San Leandro, CA. http://www.mdl.com/.
(28) SPECS, Delft, The Netherlands. http://ww.specs.net/.
(29) Chemical Computing Group (CCG). http://www.chemcomp.com/.
(30) Whitley, D. C.; Ford, M. G.; Livingstone, D. J. Unsupervised forward selection: a method for eliminating redundant variables. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1160−1168.
(31) Shannon, C. E. A mathematical theory of communication. *Bell System Tech. J.* **1948**, *27*, 379−423.
(32) Godden, J. W.; Bajorath, J. Shannon entropy − a novel concept in molecular descriptor and diversity analysis. *J. Mol. Graph. Model.* **2000**, *18*, 73−76.
(33) Godden, J. W.; Bajorath, J. Differential Shannon Entropy as a Sensitive Measure of Differences in Database Variability of Molecular Descriptors. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1060−1066.
(34) Hann, M.; Hudson, B.; Lewell, X.; Lifely, R.; Miller, L.; Ramsden, N. Strategic pooling of compounds for high-throughput screening. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 897−902.
(35) Leach, A. R.; Gillet, V. J. *An Introduction to Chemoinformatics*; Kluwer Academic Publisher: Dordrecht, 2003.
(36) Rusinko, A., III.; Farmen, M. W.; Lambert, C. G.; Brown, P. L.; Young, S. S. Analysis of a Large Structure/Biological Activity Data Set Using Recursive Partitioning. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1017−1026.
(37) Duda, R. O.; Hart, P. E.; Stork, D. G. *Pattern Classification*; John Wiley & Sons: 2000.
(38) InforSense Ltd., London, U.K. http://www.inforsense.com/.
(39) SciTegic, San Diego, CA. http://www.scitegic.com/.
(40) OpenEye Scientific Software, Santa Fe, NM. http://www.eyesopen-.com/.
(41) Sun Microsystems, Inc. http://www.sun.com/.
(42) The Molecular Design Laboratory. http://gecco.org.chemie.uni-frankfurt.de/gecco.html.
(43) Bioreason, Inc. Santa Fe, NM. http://www.bioreason.com/.
(44) Daylight Chemical Information Systems, Inc. Los Altos, CA. http://www.daylight.com/.
(45) Acharya, K. R.; Sturrock, E. D.; Riordan, J. F.; Ehlers, M. R. Ace revisited: a new target for structure-based drug design. *Nat. Rev. Drug Discovery* **2003**, *2*, 891−902.

CI050541D