A Fast Clustering Algorithm for Analyzing Highly Similar Compounds of Very Large Libraries

Weizhong Li*,†

Burnham Institute for Medical Research, 10901 N. Torrey Pines Rd., La Jolla, California 92037

Received March 13, 2006

As a result of the recent developments of high-throughput screening in drug discovery, the number of available screening compounds has been growing rapidly. Chemical vendors provide millions of compounds; however, these compounds are highly redundant. Clustering analysis, a technique that groups similar compounds into families, can be used to analyze such redundancy. Many available clustering methods focus on accurate classification of compounds; they are slow and are not suitable for very large compound libraries. Here is described a fast clustering method based on an incremental clustering algorithm and the 2D fingerprints of compounds. This method can cluster a very large data set with millions of compounds in hours on a single computer. A program implemented with this method, called cd-hit-fp, is available from http://chemspace.org.

INTRODUCTION

The recent advances in high-throughput screening (HTS) in drug development have promoted the large-scale productions of screening compounds. New technologies, such as combinatorial chemistry, provide an economic way to produce large compound libraries. Millions of compounds are available from chemical vendors, and many pharmaceutical companies also have significant numbers of compounds within their own inventories.

The growth of compound libraries increases the coverage of chemical space and gives more opportunities in identifying leads in HTS; on the other hand, the available compound libraries contain a significant amount of redundancy, and the redundancy is very unevenly distributed. For example, some compounds may have hundreds of highly similar or nearly identical neighbors, but many other compounds exist as singletons. Therefore, it is very important to analyze the redundancy of these compounds and to know the number of distinct compound families. A key tool for such analyses is clustering, a technique that groups similar compounds into families.

Clustering chemical structures is an old topic. The pioneer works started in the early 1980s.[1−4] Since then, this field have been studied extensively.[5−28] For a comprehensive review, please see the paper by Downs and Barnard.[29] Most existing clustering methods can be classified as hierarchical or nonhierarchical. Hierarchical methods, such as Ward's method,[30] build clusters by iteratively joining most similar compounds or existing clusters. Nonhierarchical methods include nearest-neighbor, relocation, mixture model, density-based, single-pass, and so on. For example, Jarvis and Patrick's method[31] is based on the nearest-neighbor algorithm.

Most existing clustering methods focus on the accurate classification of compounds; they are slow or not suitable for a very large compound set. For example, to cluster a library with 10^6 compounds, many standard clustering methods need to calculate the "all-against-all" similarities on the order of 10^{12} and store them in a matrix, which is extremely difficult with current computer power.

Two of the fastest existing algorithms are leader and *k*-means. The leader algorithm,[23,26,27] a single-pass method, builds clusters with the following steps: (1) Start the first cluster with the first compound. (2) Calculate the similarities between the next remaining compound and all existing clusters. If the similarity to its most similar cluster meets a predefined threshold, assign it to that cluster; otherwise, start a new cluster with it. (3) Repeat step 2 until done. The main drawback of this algorithm is order dependency; that is, the clusters may change if the order of input compounds changes. *k*-means is a relocation method, and there are many implementations and derivatives.[6,7,28] It first selects an initial set of *k* compounds as cluster centers and collects neighboring compounds to these clusters. It then iteratively recalculates the centroids of all clusters and reassigns each compound to its nearest centroids until converge or a predefined maximum number of iterations is reached. There are two problems with the *k*-means method. First, the number of clusters *k* has to be predefined. An inappropriate *k* may lead to under- or overclustering. Second, clusters may vary depending on the initial set of clusters.

The motivation of this study is to develop a very fast algorithm to analyze the redundancy of a very large HTS library. The goal here is not to make an accurate classification of compounds or to explain structure−activity relationships (SARs); speed is the key concern. In this paper, the leader algorithm is modified to overcome its drawback and increase the efficiency. A computer program implemented with the improved algorithm, called cd-hit-fp, can cluster millions of compounds in hours on a single computer. In a large-scale HTS experiment, this program can be used to evaluate the diversity and redundancy of a screening library, make a nonredundant library, and identify overlaps between several screening libraries. For example, a typical screening library can be reduced to half of its size after removing nearly

* Author e-mail: liwz@sdsc.edu; URL: http://chemspace.org.

† Current address: University of California San Diego, Calit2, 9500 Gilman Drive, La Jolla, California 92093.

10.1021/ci0600859 CCC: $33.50 © 2006 American Chemical Society
Published on Web 08/08/2006

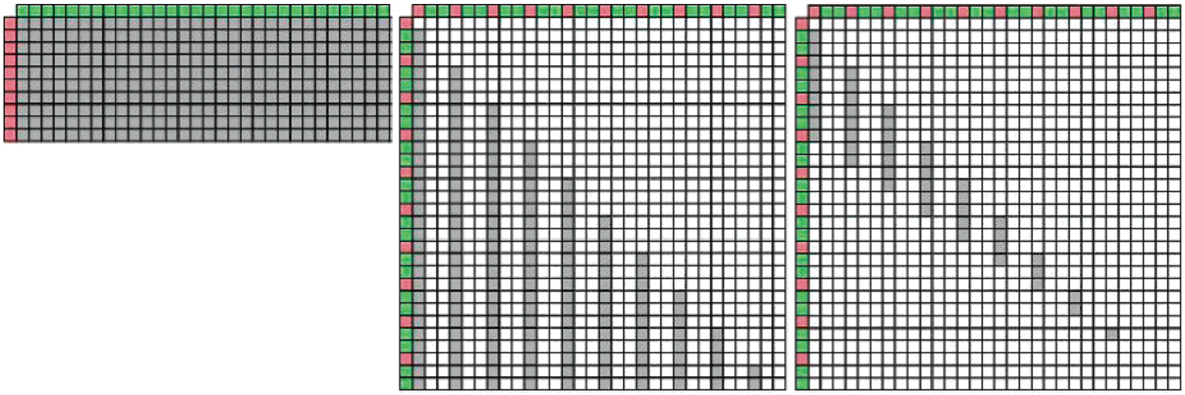

Figure 1. Calculation of compound comparisons for *k*-means, leader, and cd-hit-fp algorithms. This is a simplified example of 30 compounds being clustered into 10 clusters by the *k*-means (left), standard leader (middle), and cd-hit-fp (right) algorithms. For *k*-means, the red and green blocks stand for centroids and compounds, respectively. For leader and cd-hit-fp, the red and green blocks stand for representative and nonrepresentative compounds, respectively. The fingerprints are sorted already for cd-hit-fp. Required comparisons of compounds or clusters are in gray. In the right-hand figure, gray blocks are those that meet the condition "$N_b \geq N_a \times$ Tc" (see text). Please note that the figure just shows a single iteration for a *k*-means method, which actually needs more than one iteration.

identical compounds even at a very high similarity level of 0.95 (Tanimoto coefficient, see the Algorithm section). It means that the cost of a large-scale HTS job can be cut by half without losing hardly any chemical diversity information.

In the rest of the paper, the algorithm will first be presented; then, the performance of cd-hit-fp will be shown on several large compound libraries, and the redundancy of these libraries will be analyzed.

ALGORITHM

In this study, compounds are represented by their fingerprints, long binary strings composed of bit "1" or "0". Each bit in a fingerprint represents the presence or absence of certain two-dimensional structural features of the compound. There are several widely used fingerprints. In this study, we use the Unity fingerprint from Tripos (http://tripos.com), which is a 988-bit string. The similarity between two compounds is defined by the Tanimoto coefficient (eq 1) between their fingerprints.

$$T_{ab} = N_{ab}/(N_a + N_b - N_{ab}) \qquad (1)$$

where T_{ab} is the Tanimoto coefficient between fingerprint a and fingerprint b, N_{ab} is the number of "1" bits that occur in both fingerprint a and fingerprint b, N_a is the number of "1" bits in fingerprint a, and N_b is the number of "1" bits in fingerprint b.

For clustering, we use an algorithm similar to the incremental clustering algorithm that has been applied in selecting representative protein sequences.[32,33] It performs according to the following steps: (1) Fingerprints are sorted in order of decreasing number of bits "1". For the compounds having the same number of bits "1", their fingerprints are sorted by the positions of bits "1". For example, "0101010" is in front of "0011010". (2) The first compound, that is, the one with the most bits "1", becomes the representative of the first cluster. (3) Each remaining fingerprint is compared to the representatives of existing clusters. If the similarity to any representative is above a given threshold, it is grouped into that cluster. Otherwise, a new cluster starts with it as the representative. (4) Step 3 is repeated until done. This algorithm belongs to the "single pass" method in the paper by Downs and Barnard.[29]

With this method, not all of the pairwise similarities need to be calculated; for example, it never calculates similarities between nonrepresentatives. If calculated, the similarity between any two compounds is used just once. Therefore, any pairwise similarity can be calculated on the fly, so this method does not require the huge $N \times N$ similarity matrix.

The key novelty of the improvement of the leader algorithm in this paper is the sorting of fingerprints. The sorting of fingerprints by bit "1" has two purposes. First, it helps keep a stable clustering structure if the order of input compounds changes. With this method, every cluster has only one representative, the one with the most bits "1". If, in the first step, the fingerprints are sorted in reverse order, then the representative of each cluster is the one with most bits "0". Both decreasing and increasing sortings are implemented in the cd-hit-fp program. The sorting of fingerprints also helps improve clustering efficiency by skipping the unnecessary calculation of Tanimoto coefficients. From eq 1, it can be proved that two fingerprints a and b have the maximum Tanimoto coefficient when the "1" bits in b are a subset of the "1" bits in a, providing that $N_a \geq N_b$. So, $T_{ab\_max} = N_b/N_a$. Given a clustering threshold Tc, condition "$N_b \geq N_a \times$ Tc" is required for $T_{ab} \geq$ Tc. Therefore, it is only necessary to calculate Tanimoto coefficients within a band, that is, the space of the fingerprints and their close neighbors with a similar number of bits "1" that meet the condition. A higher Tc gives a narrower band. In the actual program, this condition is used to break the calculation loop that compares a new fingerprint to fingerprints of existing representatives.

The comparison of the cd-hit-fp algorithm to *k*-means and standard leader, the two fastest existing algorithms, is illustrated in Figure 1. Assuming that the number of final clusters is identical, a standard leader method needs half of the comparison that a single iteration of the *k*-means method needs, and the calculation of cd-hit-fp is a fraction of that in the standard leader method, see diagonal area in Figure 1. Also, the *k*-means method needs multiple iterations, and the number of iterations may vary remarkably.

RESULTS

A computer program, called cd-hit-fp, was implemented with the above algorithm. It was written in C++ and was

Table 1. Clustering of Three Compound Libraries

library	clustering threshold (Tanimoto coefficient)	number of clusters	reduced to	CPU time (hour minute)
Pubchem 5 134 871 structures	0.99	3 047 583	59%	24 h 30 m
	0.95	2 019 044	39%	57 h 11 m
	0.90	1 413 445	27%	75 h 20 m
	0.80	625 309	12%	53 h 20 m
Vendor 1 749 328 structures	0.99	1 456 787	83%	8h 3m
	0.95	1 000 817	57%	13 h 6 m
	0.90	716 788	41%	12 h 51 m
	0.80	335 766	19%	7 h 34 m
NCI 250 251 structures	0.99	203 324	81%	6 m
	0.95	174 807	70%	16 m
	0.90	148 584	59%	21 m
	0.80	95 849	38%	22 m

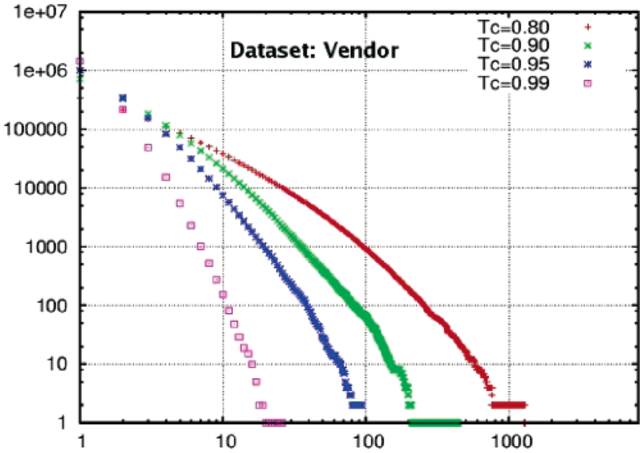

Figure 2. Cluster size distribution for the Vendor data set. The *x* axis is the cluster size *X*, and the *y* axis is the number of clusters of size at least size *X*.

tested on computers running several versions of Linux systems. This program was first run on three compound libraries: Pubchem, Vendor, and NCI.

Pubchem (http://pubchem.ncbi.nlm.nih.gov/) was downloaded in October of 2005 with 5 134 871 structures. Vendor was my collection of structures from major screening library vendors. The vendors include ASDI (asdibiosciences.com), Asinex (asinex.com), Aurora (aurora-feinchemie.com), Key Organics (keyorganics.ltd.uk), Chemdiv (chemdiv.com), Chemstar (chemstar.ru), Chembridge (chembridge.com), Comgenex (comgenex.hu), Emc (microcollections.de), Enamine (enamine.net), Life Chemicals (iflab.kiev.ua), Ibs (ibscreen.com), Maybridge (maybridge.com), Moscow Medchemlabs (mosmedchemlabs.com), Nanosyn (nanosyn.com), Peakdale (peakdale.co.uk), Princeton Biomolecular Research (princetonbio.com), Ryan (ryansci.com), Sigma-Aldrich (sigmaaldrich.com), Spectrum Info (spectrum.kiev.ua), Specs (specs.net), Timtec (timtec.net), Tripos (leadquest.tripos.com), and Vitas-m Laboratory (vitasmlab.com). I collected ∼3.85 million structures, but there were significant overlaps between the libraries of these vendors. The redundant compounds with identical fingerprints were removed, and 1 749 328 unique compounds were left in the final Vendor data set. NCI database release 2 (http://cactus.nci.nih.gov/ncidb2/download.html) contained 250 251 structures.

Table 1 lists the clustering results of these three libraries at several clustering thresholds: 0.99, 0.95, 0.90, and 0.80. The calculation was performed on a computer with a 3.0 GHz Xeon processor and 2 GB of RAM running Linux. It took the program hours to a few days to cluster the huge Pubchem library. For the NCI library, it only took minutes to cluster it. The clustering threshold is an important factor affecting the calculating time. A higher clustering threshold reduces the calculation of Tanimoto coefficients by narrowing down the band of the compound comparison (see the algorithm). In the meantime, the program produces more clusters with a higher threshold so that a new compound needs to be compared to more existing representatives. The tests listed in Table 1 and many other tests that were performed showed that the thresholds of the most time-consuming clustering ranged from 0.80∼0.95. Table 1 also shows the redundancy of these libraries. For example, after highly similar structures are removed at Tc = 0.99, the size of the Pubchem library can be reduced to 59%. At all similarity levels, NCI is the most nonredundant library, while Pubchem contains the most redundant structures.

The distribution of cluster size for Vendor is shown in Figure 2. The plot shows the number of clusters of at least a certain size. For example, Vendor has 155, 7585, 21 336, and 38 001 clusters of ≥10 compounds at Tc = 0.99, 0.95, 0.90, and 0.80, respectively.

Compounds from three individual vendors were clustered separately. These three libraries have 348 000, 439 000, and

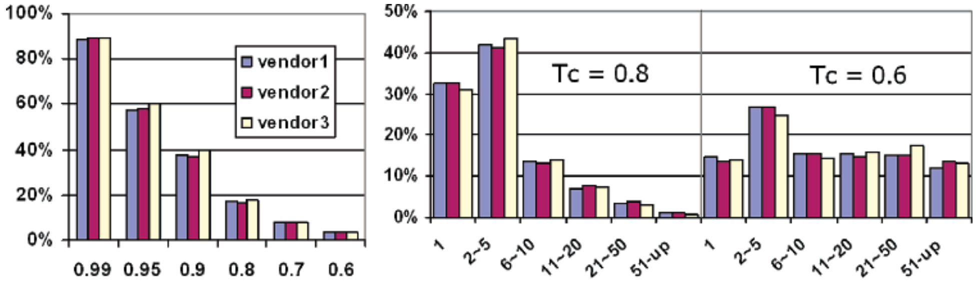

Figure 3. Clustering of libraries of three vendors. Left is the relative size of clustered libraries, where the *x* axis is the clustering threshold Tc. Right is percentage distribution of clusters of a certain size, where the *x* axis is the size of a cluster.

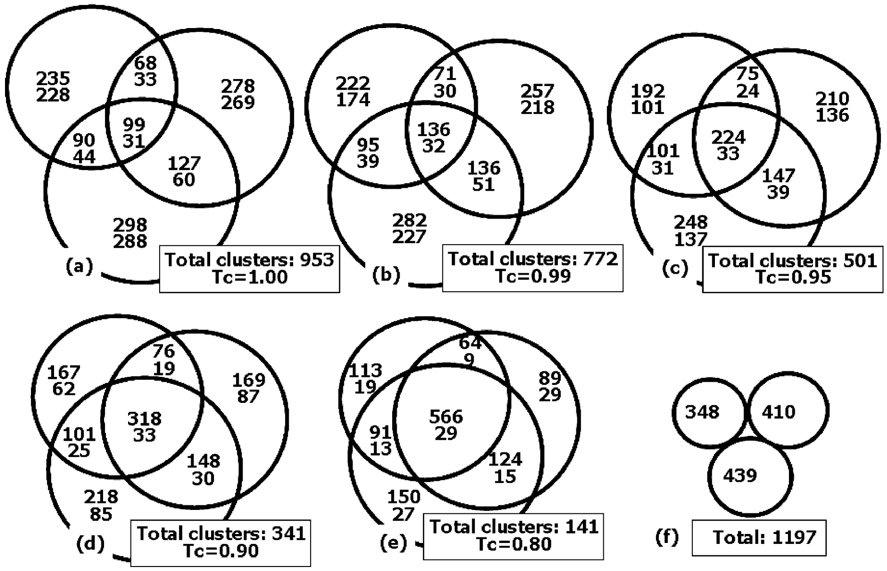

Figure 4. Overlaps between libraries of three vendors at different similarity levels. Compounds of three vendors are clustered together at several similarity thresholds (Tc), from a−e, 1.00, 0.99, 0.95 0.90, and 0.80. The upper and lower numbers in each area are the number of compounds and the number of clusters of these compounds at the corresponding clustering threshold. Part f shows the sizes of these libraries. Please note that the numbers of compounds and clusters are in thousand.

410 000 compounds. The left-hand graph in Figure 3 shows the relative size of the clustered libraries. Interestingly, these three libraries are very similar. On average, these libraries can be reduced to <20% at Tc = 0.8, which is in accordance with the all-vendor set as shown in Table 1. At Tc = 0.7 and 0.6, these libraries can be reduced to <10% and <4%, respectively. The right-hand graph in Figure 3 shows the percentage distribution of clusters of a certain size, and the three libraries are also very similar. At Tc = 0.8, over 70% of the clusters are singletons and small clusters (two to five compounds), and at Tc = 0.6, over 60% of the clusters are large clusters (six and up compounds).

Clustering several libraries together can deduce the overlaps between them at a certain similarity level. Figure 4 shows such an analysis. These three libraries in the above analysis were clustered together at several thresholds: 1.00, 0.99, 0.95, 0.90, and 0.80. Significant overlaps were identified between these three sets. For example, the second vendor only had 298 000 compounds with unique fingerprints. Although, in principle, one fingerprint may match more than one structure, it happens very rarely, <1%, on the basis of my large-scale statistical analysis on Unity fingerprints. When the similarity threshold, Tc, was decreased to 0.8, about half of the compounds were common to all three vendors.

DISCUSSION

Cost is an important issue in HTS experiments. When the available screening library is much greater than what the experimental capacity and cost allow, knowing and handling the redundancy within the library improves the HTS efficiency. In my opinion, in a general-purpose HTS experiment, screening a large amount of near-identical compounds is a waste. Screening compounds that have no neighbors at all may also cause problems such as validation. Small and medium clusters are more ideal in HTS experiments in terms of cost and validation. A case where a cluster is hit several times is not only a validation by itself but also provides immediate data for initial SAR analysis.

My goal of developing cd-hit-fp is to try and help analyze and handle redundancy in large screening libraries and select compounds from these libraries to make more cost-effective HTS experiments. Because this program is very fast, even with millions of compounds, it is very easy to use in practical experiments.

REFERENCES AND NOTES

(1) Willett, P. Evaluation of relocation clustering algorithms for the automatic classification of chemical structures. *J. Chem. Inf. Comput. Sci.* **1984**, *24* (1), 29−33.

(2) Rubin, V.; Willett, P. A comparison of some hierarchal monothetic divisive clustering algorithms for structure−property correlation. *Anal. Chim. Acta* **1983**, *151* (1), 161−6.

(3) Willett, P. A comparison of some hierarchal agglomerative clustering algorithms for structure−property correlation. *Anal. Chim. Acta* **1982**, *136*, 29−37.

(4) Adamson, G. W.; Bawden, D. Comparison of hierarchical cluster analysis techniques for automatic classification of chemical structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21* (4), 204−9.

(5) Luque Ruiz, I.; Cerruela Garcia, G.; Gomez-Nieto, M. A. Clustering chemical databases using adaptable projection cells and MCS similarity values. *J. Chem. Inf. Model.* **2005**, *45* (5), 1178−94.

(6) Smellie, A. Accelerated K-means clustering in metric spaces. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (6), 1929−35.

(7) Holliday, J. D.; Rodgers, S. L.; Willett, P.; Chen, M. Y.; Mahfouf, M.; Lawson, K.; Mullier, G. Clustering files of chemical structures using the fuzzy k-means clustering method. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (3), 894−902.

(8) Raymond, J. W.; Willett, P. A line graph algorithm for clustering chemical structures based on common substructural cores. *MATCH* **2003**, *48*, 197−207.

(9) Raymond, J. W.; Blankley, C. J.; Willett, P. Comparison of chemical clustering methods using graph- and fingerprint-based similarity measures. *J. Mol. Graphics Modell.* **2003**, *21* (5), 421−33.

(10) Feher, M.; Schmidt, J. M. Fuzzy clustering as a means of selecting representative conformers and molecular alignments. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (3), 810−8.

(11) Downs, G. M.; Barnard, J. M. Clustering of very large datasets. *Abstracts of Papers*, 222nd ACS National Meeting, Chicago, IL, August 26−30, 2001; American Chemical Society: Washington, DC, 2001; COMP-092.

(12) Wild, D. J.; Blankley, C. J. Comparison of 2D fingerprint types and hierarchy level selection methods for structural grouping using Ward's clustering. *J. Chem. Inf. Comput. Sci.* **2000**, *40* (1), 155−62.

(13) Butina, D. Unsupervised database clustering based on Daylight's fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (4), 747−750.

(14) Barnard, J. M.; Downs, G. M.; Brown, R. D. Use of Markush structure-analysis techniques for rapid processing of large combinatorial libraries. *Book of Abstracts*, 218th ACS National Meeting, New Orleans, LA, Aug. 22−26, 1999; American Chemical Society: Washington, DC, 1999; CINF-005.

(15) Reynolds, C. H.; Druker, R.; Pfahler, L. B. Lead discovery using stochastic cluster analysis (SCA): A new method for clustering structurally similar compounds. *J. Chem. Inf. Comput. Sci.* **1998**, *38* (2), 305−312.

(16) 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* (3), 497−505.

(17) McGregor, M. J.; Pallai, P. V. Clustering of Large Databases of Compounds: Using the MDL "Keys" as Structural Descriptors. *J. Chem. Inf. Comput. Sci.* **1997**, *37* (3), 443−448.

(18) Mansfield, J. R.; Sowa, M. G.; Scarth, G. B.; Somorjai, R. L.; Mantsch, H. H. Fuzzy C-means clustering and principal component analysis of time series from near-infrared imaging of forearm ischemia. *Comput. Med. Imaging Graph.* **1997**, *21* (5), 299−308.

(19) Barnard, J. M.; Downs, G. M. Chemical fragment generation and clustering software. *J. Chem. Inf. Comput. Sci.* **1997**, *37* (1), 141−142.

(20) 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* (6), 1195−1204.

(21) Downs, G. M.; Willett, P.; Fisanick, W. Similarity searching and clustering of chemical-structure databases using molecular property data. *J. Chem. Inf. Comput. Sci.* **1994**, *34* (5), 1094−102.

(22) Barnard, J. M.; Downs, G. M. Clustering of chemical structures on the basis of two-dimensional similarity measures. *J. Chem. Inf. Comput. Sci.* **1992**, *32* (6), 644−9.

(23) Hodes, L. Clustering a large number of compounds. 1. Establishing the method on an initial sample. *J. Chem. Inf. Comput. Sci.* **1989**, *29* (2), 66−71.

(24) Willett, P. *Similarity and clustering in chemical information systems*; Publisher: Place of Publication, 1987; p 254.

(25) 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* (3), 109−18.

(26) Hodes, L.; Feldman, A. Clustering a large number of compounds. 3. The limits of classification. *J. Chem. Inf. Comput. Sci.* **1991**, *31* (2), 347−50.

(27) Whaley, R.; Hodes, L. Clustering a large number of compounds. 2. Using the Connection Machine. *J. Chem. Inf. Comput. Sci.* **1991**, *31* (2), 345−7.

(28) Bocker, A.; Derksen, S.; Schmidt, E.; Teckentrup, A.; Schneider, G. A hierarchical clustering approach for large compound libraries. *J. Chem. Inf. Model.* **2005**, *45* (4), 807−15.

(29) Downs, G. M.; Barnard, J. M. Clustering methods and their uses in computational chemistry. *Rev. Comput. Chem.* **2002**, *18*, 1−40.

(30) Ward, J. H. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **1963**, *58*, 236−244.

(31) Jarvis, R. A.; Patrick, E. A. Clustering using a similarity based on shared near neighbors. *IEEE Trans. Comput.* **1973**, *C22*, 1025−1034.

(32) Li, W.; Jaroszewski, L.; Godzik, A. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics* **2001**, *17* (3), 282−3.

(33) Hobohm, U.; Scharf, M.; Schneider, R.; Sander, C. Selection of representative protein data sets. *Protein Sci.* **1992**, *1* (3), 409−17.

CI0600859