Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets

Darko Butina

Science Development Group - Biomet, Glaxo Wellcome Research and Development, Medicines Research Centre, Gunnels Wood Road, Stevenage, Hertfordshire SG1 2NY, U.K.

Received November 11, 1998

One of the most commonly used clustering algorithms within the worldwide pharmaceutical industry is Jarvis−Patrick's (J−P) (Jarvis, R. A. IEEE Trans. Comput. **1973**, *C-22*, 1025−1034). The implementation of J−P under Daylight software, using Daylight's fingerprints and the Tanimoto similarity index, can deal with sets of 100 k molecules in a matter of a few hours. However, the J−P clustering algorithm has several associated problems which make it difficult to cluster large data sets in a consistent and timely manner. The clusters produced are greatly dependent on the choice of the two parameters needed to run J−P clustering, such that this method tends to produce clusters which are either very large and heterogeneous or homogeneous but too small. In any case, J−P always requires time-consuming manual tuning. This paper describes an algorithm which will identify dense clusters where similarity within each cluster reflects the Tanimoto value used for the clustering, and, more importantly, where the cluster centroid will be *at least* similar, at the given Tanimoto value, to every other molecule within the cluster in a consistent and automated manner. The similarity term used throughout this paper reflects the *overall* similarity between two given molecules, as defined by Daylight's fingerprints and the Tanimoto similarity index.

INTRODUCTION

Clustering[2,3] has been described as 'the art of finding groups in data'[4] and is widely used within the pharmaceutical industry to design different representative sets. Most common uses of representative sets could be as training sets in the development of different structure−activity models and for screening in different biological screens. In both cases, one would assume that the cluster centroid is a good representative member of the corresponding cluster. It is therefore of great importance to be able to create homogeneous clusters in a consistent way and to deal with either small or very large sets equally well. Our approach uses desired similarity within the cluster, as defined by Tanimoto index, as the only input to the clustering program.

METHODOLOGY

There are three key steps in this clustering approach:

1. generation of standard Daylight's fingerprints (ASCII);
2. identification of potential cluster centroids;
3. clustering based on the exclusion spheres.

1. Generation of Fingerprints. Fingerprints for each molecule are generated, using Daylight software, as an ASCII string of 1's and 0's (fixed width at 1024). See Appendix 1 for more details on the concept of Daylight's fingerprints.

2. Identifying Potential Cluster Centroids. It is reasonable to postulate that a molecule within a given cluster which has the largest number of neighbors and is therefore 'most like' the rest of the cluster is a good choice to become a cluster centroid. To identify such molecules, we calculate the number of neighbors for each molecule in the set, at the Tanimoto level chosen for the clustering. The set is then sorted in descending order, so that the potential cluster centroids, i.e., the compounds with the largest number of neighbors, are placed at the top of the file. This is the key step needed to ensure the *order-independent features* of this clustering algorithm (for more details on the Tanimoto similarity index, see Appendix 2), but it is also the computationally most expensive step.

3. Cluster Algorithm Based on Exclusion Spheres at a Given Tanimoto Level. The main principle of this algorithm is to start with the first compound in the sorted list from step 2 and calculate its pairwise Tanimoto similarity index to all other compounds. All those molecules with a Tanimoto index above, or equal to, the value used for clustering become members of that cluster. Each molecule that has been identified as a member of the given cluster is flagged and removed from any further comparisons, i.e., the flagged molecule cannot become either another cluster centroid or a member of another cluster. One could envisage this process as putting an exclusion sphere around the newly formed cluster.

Once the first compound in the list has found all its neighbors, the first available (i.e., not flagged) compound at the top of the list becomes the new cluster centroid, and the same process is repeated for all other unflagged molecules down the list. As more compounds are assigned to different clusters through each iteration (i.e., flagged), fewer comparisons are needed to be calculated and the clustering proceeds faster to completion. A schematic presentation of this algorithm can be seen in Scheme 1.

The molecules that have not been flagged by the end of the clustering process, either as a cluster centroid or as a cluster member, become singletons. It is important to emphasize at this stage that one of the consequences of this approach is that some molecules defined as singletons may

10.1021/ci9803381 CCC: $18.00
Published on Web 06/29/1999

Scheme 1

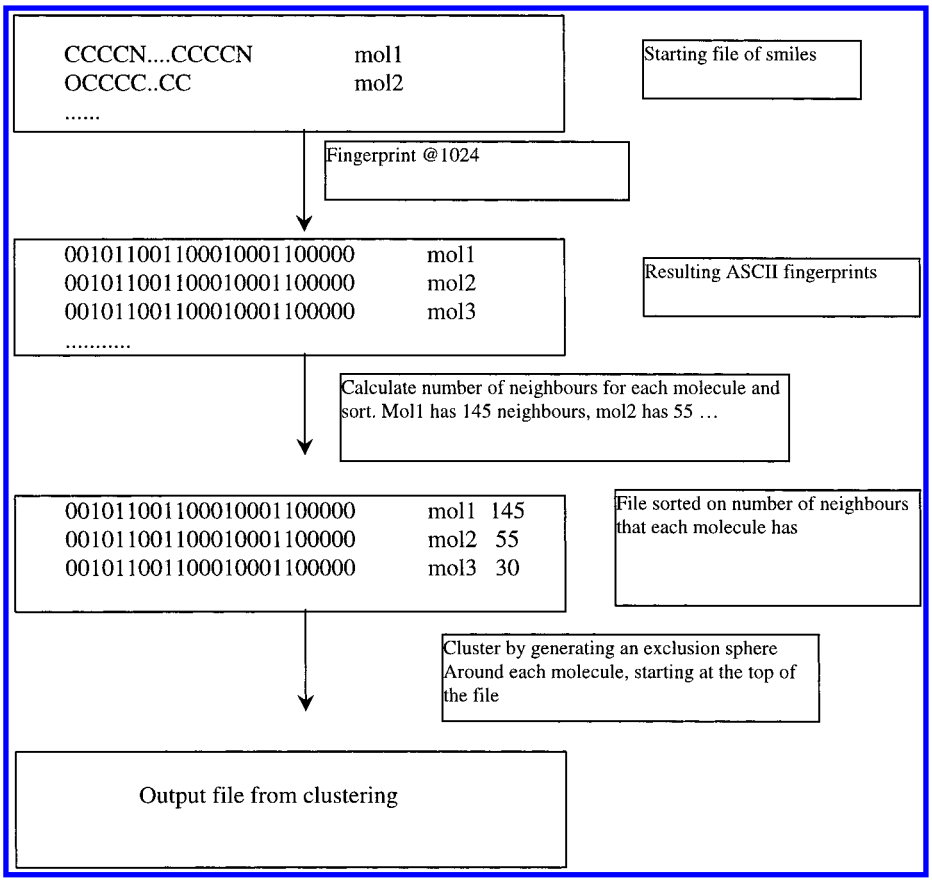
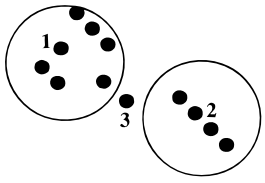


Chart 1



have neighbors at the given Tanimoto similarity index, but those neighbors have been excluded by a 'stronger' cluster centroid, i.e., one with more neighbors in its list. This situation is depicted in Chart 1. Thus, molecules 1 and 2, both having more neighbors than molecule 3, have 'removed' all the molecules within the spheres defined by a given Tanimoto value, creating clusters 1 and 2. Molecule 3 becomes a singleton by the fact that the 'stronger' cluster centroids have removed its own neighbors.

However, the benefit of this approach is in avoiding the formation of highly heterogeneous clusters, a common occurrence in algorithms such as Jarvis−Patrick (J−P). The problem of 'chaining' or 'long' clusters is a notorious one and choosing molecule 3 as the cluster centroid of all the molecules covered by Chart 1 would lead to a highly heterogeneous cluster. Since one of the main objectives for designing this algorithm was to identify dense clusters in a consistent and automated manner, the problem with the creation of a number of false singletons that do in fact have similar compounds within the set is easily offset by the final quality of the clusters that this approach generates.

RESULTS AND DISCUSSION

The main objective for developing this approach was to enable the creation of clusters where homogeneity within the cluster reflects the Tanimoto index used for the clustering in a consistent manner. To demonstrate this point, and also to highlight the problems with the J−P algorithm discussed at the beginning of the paper, a set of compounds from the public domain Medchem database, containing 5424 molecules, was clustered at a Tanimoto level of 0.8 using the method described in this paper (termed DB) and J−P clustering with 0.8 cutoff (see Appendix 3 for more details on Daylight implementation of J−P clustering).

DB clustering gave 764 clusters and 2644 singletons (52% of the set was clustered and 48% assigned as singletons), J−P clustering (−p 6/16) gave 238 clusters and 2533 singletons (54% in clusters and 46% in singletons), and J−P (−p 5/8) gave 575 clusters (51%) and 2663 singletons.

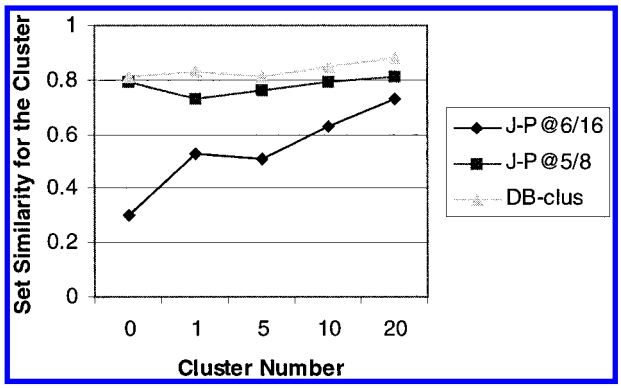
method	no. of clusters	% set clustered	no. of singlets	SS[a] Cl 0	Cl 1	Cl 4	Cl 5
J−P 6/16	238	54	2533	0.39	0.53	0.55	0.57
J−P 5/8	575	51	2663	0.79	0.73	0.76	0.77
DB@0.8	764	52	2644	0.82	0.81	0.79	0.82

[a] SS, set similarity within the cluster.

While both methods have placed a similar number of molecules into clusters, J−P clustering has done it in one-third the number of clusters produced by DB clustering. The price paid is revealed by the heterogeneity of the largest clusters generated by J−P (Chart 2). Sampling of clusters generated by each method and calculating the set similarity within each cluster (see Appendix 4 for details) reveal significant differences between the two algorithms.

Any cluster generated by the DB clustering algorithm in the example used, regardless of its size, has a set similarity

Chart 2



of about 0.8 and the cluster centroid *at least* 0.8 similar to all other cluster members. This fact reflects the relationship between the Tanimoto value used for the clustering and also validates the algorithm used and its coding.

Conversely, the similarity within the clusters generated by the J−P algorithm can vary enormously, from 0.3 for the largest cluster when the 6/16 setup was used to 0.79 for the same cluster when the 5/18 combination was used. This uncertainty about the quality of the clusters generated by the J−P method, necessitating manual inspection whenever a new set is clustered, can be very frustrating and, when clustering large sets (100 k's or more), becomes a very time-consuming process. Many research groups in the field have expressed similar views on the failings of the J−P algorithm, typified recently by the group from Searle.[5]

The table below gives some indication of the times it takes to cluster different sized sets, starting with an existing, fingerprinted set and using a standard Indigo SG workstation with 32 MB RAM and R4000 processor:

set size	time
1 k	15 s
10 k	20 min
100 k	24 h

SUMMARY

A new clustering algorithm has been developed, which has the following features: (a) generation of good quality clusters where set similarity within a cluster reflects the Tanimoto level used for clustering and is the **only** input needed by the clustering algorithm; (b) no requirement for visual inspection of resulting clusters as a quality control, since the combination of standard Daylight fingerprints and similarity at high Tanimoto index produces a very reliable way of grouping together similar molecules; (c) order-independent algorithm which identifies and sorts by potential cluster centroids; (d) can be fully automated and presented in a user-friendly manner; (e) main clustering program is written in ANSI C and as such is independent of Daylight software and can be used to cluster any type of binary fingerprints on the fastest Unix workstation available.

ACKNOWLEDGMENT

I greatly appreciate the help of the following people. Liqun Wang for assistance in optimizing and error trapping my original code. The design and application of a new version of this program, which can deal with 100 000 molecules within 24 h of computing time, will be part of a subsequent publication. Peter McMeekin for supporting this work. John Bradshaw for use of his sim.c program for calculating set similarity. Peter Eddershaw for proofreading the manuscript.

APPENDIX 1. DAYLIGHT FINGERPRINTS

Visit http://www.daylight.com/dayhtml/doc/ for more details on Daylight software. Similarity metrics, calculations that quantify the similarity of two molecules, and screening, a way of rapidly eliminating molecules as candidates in a substructure search, are both processes that use fingerprints. Fingerprints are an abstract representation of certain structural features of a molecule.

Unlike a structural key with its predefined patterns, the patterns for a molecule's fingerprint are generated from the molecule itself. The fingerprinting algorithm examines the molecule and generates the following: a pattern for each atom, a pattern representing each atom and its nearest neighbors (plus the bonds that join them), a pattern representing each group of atoms and bonds connected by paths up to 2 bonds long, ...atoms and bonds connected by paths up to 3 bonds long, ...continuing, with paths up to 4, 5, 6, and 7 bonds long. For example, the molecule OC═CN would generate the following patterns:

0-bond paths: C O N

1-bond paths: OC C═C CN

2-bond paths: OC═C C═CN

3-bond paths: OC═CN

Because there is no predefined set of patterns, and because the number of possible patterns is so huge, it is not possible to assign a particular bit to each pattern as we did with structural keys. Instead, each pattern serves as a seed to a pseudo-random number generator (it is "hashed"), the output of which is a set of bits (typically 4 or 5 bits/pattern); the set of bits thus produced is added (with a logical OR) to the fingerprint.

Although a fingerprint does not indicate with 100% certainty that a particular pattern is present, it contains far more patterns total than a structural key, the net result being that a fingerprint is a far better screen than a structural key in almost all situations.

APPENDIX 2. TANIMOTO SIMILARITY INDEX

There are many different ways one might compute the similarity of two bitmaps. The Daylight system provides three such measures: the Tanimoto coefficient, the Euclidian distance, and the Tversky similarity. For the descriptions of Tanimoto coefficient, we will use the following symbols: B1 = bits(F1), the number of 1's in F1; B2 = bits(F2), the number of 1's in F2; BC = bits(F1 and F2), the number of 1's in common between F1 and F2.

The Tanimoto coefficient is computed as the number of bits in common divided by the total number of bits. The Tanimoto coefficient can be expressed as:

$$\text{Tanimoto} = BC/(B1 + B2 - BC)$$

The Tanimoto coefficient is a particularly intuitive similarity measure, as it is "normalized" to account for the number of bits that might be in common relative to the number that

are in common. That is, it is a measure of the number of common substructures shared by two molecules. As can be seen from the expression above, Tanimoto of 1 indicates identical molecule, while Tanimoto of 0 will indicate that two molecules have nothing in common.

APPENDIX 3. PARAMETERS USED IN DAYLIGHT'S IMPLEMENTATION OF JARVIS-PATRICK CLUSTERING

Visit http://www.daylight.com for more details. jpscan and jarpat both perform J−P clustering based on nearest-neighbors (NN) data. Both programs use two J−P clustering parameters: the number of neighbors to examine and the number required to be in common. jpscan repeatedly clusters data using all possible parameter combinations up to a given limit (typically set to the list length, default is 16) and outputs tables of statistics intended to help in selecting a pair of parameters appropriate to the problem at hand. jarpat requires that the parameters be specified and outputs the clustering results. It is advisable to run jpscan and examine its output before running jarpat.

jarpat provides two (nonexclusive) methods for dealing with singletons: rescuing singletons and writing them out to a separate file. If singleton rescue is used (option -RESCUE_SIZE), rescued singletons will appear in clusters to which they are rescued. If a singleton file is generated (option -SINGLETON_FILE), it may be fed back to near neighbors and then reclustered.

jarpat provides an additional processing option which is not part of the original J−P algorithm. This option (-NN_BEST_THRESHOLD) allows the preprocessing of the neighbors lists as follows: the best neighbor (excluding itself) for each structure is compared with the threshold value. If the best neighbor has a similarity lower than the specified threshold, then the structure is marked as a singleton and is excluded from the clustering. This is a useful way to discover very tight clusters within a data set.

APPENDIX 4. SET SIMILARITY

The program sim.c (written by John Bradshaw) will calculate pairwise similarity for each molecule in the set, and the overall set similarity will be the ratio of the sum of the individual similarities for each molecule to all other molecules in the set, over number of the comparisons made. In other words, the more similar molecules are in the given set, the higher set similarity will be.[6,7] In our case, the tighter the clusters are (the more similar molecules are within the cluster), the higher set similarities will be obtained. For example:

Mol1 0.9204

Mol2 0.9217

Mol3 0.8974

Mol4 0.9009

Mol5 0.8781

Mol6 0.8819

Mol7 0.8781

Mol8 0.8443

Mol9 0.8768

Mol10 0.8570

set resemblance = 0.88

It is quite obvious that the first 10 molecules in the set above are very similar to each other and that a set similarity of 0.88 identifies that set as a very homogeneous one. This approach allows very fast evaluation of tightness of the clusters produced.

REFERENCES AND NOTES

(1) Jarvis, R. A. Clustering using a similarity measure based on shared nearest neighbors. *IEEE Trans. Comput.* **1973**, *C-22*, 1025−1034.
(2) Willett, P. *Similarity and Clustering in Chemical Information Systems*; Research Studies Press: Letchworth, 1987.
(3) 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*, 644−649.
(4) Arbie, P.; Hubert, L. J.; De Soete, G. *Clustering and Classification;* World Scientific: Singapore, 1996; p 2.
(5) Doman, T. M.; et al. Algorithm5: A Technique for Fuzzy Similarity Clustering of Chemical Inventories. *J. Chem. Inf. Comput. Sci*. **1996**, *36*, 1195−1204.
(6) Tversky, A. Perception of Similarity. *Psychol. Rev.* **1977**, *84*, 317−352.
(7) Turner, D. B.; Tyrrell, S. M.; Willett, P. Rapid Quantification of Molecular Diversity for Selective Database Acquisition. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 18−22.

CI9803381