A Fast Algorithm For Selecting Sets Of Dissimilar Molecules From Large Chemical Databases

John D. Holliday, Sonia S. Ranade and Peter Willett*

Krebs Institute for Biomedical Research and Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TN, United Kingdom

Abstract

Current algorithms for the selection of a set of *n* dissimilar molecules from a dataset of *N* molecules have an expected time complexity of $O(n^2N)$. This paper describes an improved algorithm that has an expected time complexity of $O(nN)$ and that will identify exactly the same set of molecules as the normal algorithm if the cosine coefficient is used for the calculation of the inter-molecular (dis)similarities. The algorithm is applicable to any type of representation that characterises a molecule by a set of attribute values and to any procedure that involves calculating a sum of inter-molecular similarities. It is also both more effective and more efficient than our implementation of a genetic algorithm for the selection of maximally-dissimilar sets of molecules.

Key words: Algorithmic complexity; Compound selection; Dissimilarity selection; Random screening; Similarity coefficient

1 Introduction

Random screening is widely used for the identification of lead compounds in drug- and pesticide-discovery programmes, where compounds are selected from a database, such as a company's corporate structure file, for testing in one or more primary biological screens. The need to ensure that the full range of structural diversity present in the database is tested in the biological screen(s) has led to the development of two systematic approaches to the selection of compounds [1, 2].

Cluster-based selection involves the identification of groups, or clusters, of structurally-related molecules on the basis of inter-molecular similarity calculations. Once the clusters have been identified, one molecule is selected from each cluster in turn for inclusion in the biological screening programme [3–6]. An effective clustering method will seek to identify groupings that maximise the intra-cluster similarity and minimise the inter-cluster similarity, so that the selection of molecules from individual clusters should result in a structurally-diverse set of molecules for screening. *Maximum dissimilarity selection* seeks to identify a maximally-dissimilar subset of the molecules in the database without the intermediate clustering step [7–10]. In both approaches to compound selection, the inter-molecular (dis)similarities are generally obtained by comparing the fragment bit-strings, which are commonly used for the screening stage of a substructure search, to identify the bits (and hence fragments) common to a pair of compounds and then calculating a similarity coefficient such as the Tanimoto coefficient [1–4].

Analogous procedures may be used to process the outputs of broadly-defined chemical substructure searches. The execution of such a search on a large database can often result in the retrieval of many thousands of molecules, especially in the case of pharmacophoric-pattern searches of databases of three-dimensional structures. The user's assimilation of the search-output can be eased by tools that provide an overview of the extent of the structural heterogeneity in the search output [4, 5]. Another application where compound selection is of importance is in the generation of diverse sets of molecules for use in combinatorial approaches to drug design, where there is a need to maximise the range of structural types that are included in a combinatorial library [11].

The identification of the maximally-diverse subset is computationally infeasible since it requires the use of a combinatorial algorithm that considers all possible subsets of a given dataset. For example, the identification of the maximally-diverse subset of size *n* from a dataset of size *N* (where, typically, $n<<N$) would involve consideration of up to

$$\frac{N!}{n!(N-n)!}$$

possible subsets. Practical considerations hence dictate a simpler form of optimisation that focuses on the pair-wise molecular similarities and that ignores all of the higher-order associations. Such an algorithm is illustrated in Figure 1. While this is clearly simple, it involves the comparison of each molecule in a database with every other molecule, with a consequent expected time complexity of order $O(N^2)$. The quadratic nature of the algorithm means that it is extremely time-consuming if a dataset of realistic size, *e.g.*, a corporate database containing 150000 molecules, is to be processed (although the computational requirements are very much less than in the case of the combinatorial procedure that is required for the identification of the maximally-dissimilar subset). This paper describes a fast algorithm that enables dissimilarity selection to be carried out on even the largest databases in an acceptable amount of time.

1. Select a molecule, *J*, in the database and calculate the sum, *SUM(J)*, of its similarities with each of the other molecules using the chosen similarity metric.
2. Repeat Step 1 for each of the other *N*-1 molecules in the database.
3. Select those *n* molecules that have the smallest values for *SUM(J)*.

Figure 1. A simple algorithm for dissimilarity selection from a database containing *N* molecules.

* to receive all correspondence

2 Material and Methods

The algorithm presented here derives from work by Voorhees [12] on the implementation of the group-average method for clustering document databases. This clustering method involves the fusion of pairs of clusters of objects (documents represented by index terms in her application) on the basis of the average of the inter-object similarities, where one object is in one cluster of the pair and the other object is in the other cluster. Voorhees described an efficient algorithm for the group-average method that is appropriate for all similarity measures where the mean similarity between the objects in two sets is equal to the similarity between the mean objects of each of the two sets. However, the basic idea is applicable to any environment where sums of similarities, rather than the individual similarities, are required: it has been used previously in our laboratory in studies of index-term weighting schemes for text retrieval [13, 14] and by a group at Xerox PARC for the implementation of non-hierarchic document clustering [15] but its application to compound selection is, to our knowledge, completely novel.

Let us assume that one wishes to investigate the similarities between each of the molecules in two separate groups, A and B (in Voorhees' work, A and B were two clusters of documents that were to be fused). Next, assume that each molecule, J, in these groups is represented by a vector in which the I-th element, $M(J,I)$, represents the weight of the I-th feature in $M(J)$, *e.g.*, a one or zero denoting the presence or absence of a particular fragment or a numeric value denoting some physical property. Let A_C be the linear combination, or *centroid*, of the individual molecule vectors $M(J)(1 <= J <= N(A)$, where $N(A)$ is the number of molecules in the first group) with $W(J)$ being the weight of the J-th vector in A. The I-th element of A_C, $A_C(I)$, is thus given by

$$A_C(I) = \sum_{J=1}^{N(A)} W(J) \times M(J,I).$$

The vector B_C, which denotes the centroid of the second group, is similarly defined in terms of its $N(B)$ constituent molecules.

The dot product of the two vectors A_C and $B_C, A_C \cdot B_C$, is given by

$$A_C \cdot B_C = \sum_{J=1}^{N(A)} W(J) \times M(J) \sum_{K=1}^{N(B)} W(K) \times M(K),$$

which may be rewritten as

$$A_C \cdot B_C = \sum_{J=1}^{N(A)} \sum_{K=1}^{N(B)} W(J) \times W(K) \times M(J) \times M(K).$$

Now, $M(J) \times M(K)$, the dot product of the two vectors, is given by

$$\sum_{I=1}^{C} M(J,I) \times M(K,I),$$

where C is the number of different characteristics that are used to represent each of the molecules in A and in B. We can hence write

$$A_C \cdot B_C = \sum_{J=1}^{N(A)} \sum_{K=1}^{N(B)} \sum_{I=1}^{C} W(J) \times W(K) \times M(J,I) \times M(K,I).$$

Let us define the weight, $W(J)$, to be the reciprocal square root of the squared elements of the vector $M(J)$, *i.e.*,

$$W(J) = 1 \Bigg/ \sqrt{\sum_{I=1}^{C} M(J,I)^2},$$

and similarly for $W(K)$. Hence,

$$A_C \cdot B_C = \sum_{J=1}^{N(A)} \sum_{K=1}^{N(B)} \left[\sum_{I=1}^{C} M(J,I) \times M(K,I) \Bigg/ \sqrt{\sum_{I=1}^{C} M(J,I)^2 \times \sum_{I=1}^{C} M(K,I)^2} \right].$$

There are many ways of measuring the degree of resemblance between a pair of objects. A common class of similarity coefficients are the *association coefficients* [3, 16]. There are many such coefficients, but they all involve some normalised form of the dot product of the vector representations of the two molecules that are being compared, with the coefficients of this class differing only in the type of normalisation that is involved. The *Tanimoto coefficient* is one example of an association coefficient that is widely used as an effective measure of intermolecular similarity in applications such as similarity searching, database clustering and QSAR [3, 4, 17, 18]. Another association coefficient is the *cosine coefficient* [3, 16], and the bracketed function on the right-hand side of the expression above is simply the cosine coefficient, $COS(J,K)$, for the similarity between the molecules J and K. We can hence write

$$A_C \cdot B_C = \sum_{J=1}^{N(A)} \sum_{K=1}^{N(B)} COS(J,K).$$

This expression demonstrates that if we wish to calculate the sum of all of the $N(A)xN(B)$ intermolecular similarities using the cosine coefficient, then the resulting sum can equally well be obtained by calculating the dot product of the two vector centroids.

It is simple to apply this equivalence to the selection of n compounds from a database, D, using the algorithm shown in Figure 1. In this case A denotes a single molecule, J, from D and B denotes all of D with the exception of J, which we shall refer to as DJ. Then $A_C \cdot B_C$, *i.e.*, $D_C \cdot DJ_C$ in the present context, gives the sum of all of the similarities between the molecule, J, that has been selected from D and all of the molecules in DJ, *i.e.*, the value $SUM(J)$ that has been defined previously. The only difference is that $SUM(J)$ is obtained here by a single similarity calculation (that between J and the centroid of DJ), rather than by N-1 similarity calculations (between J and each separate molecule in DJ). The calculation of all of the N $SUM(J)$ values hence requires only N similarity calculations; once these have been carried out, the n lowest $SUM(J)$ values can be obtained by a simple sorting operation.

This *centroid algorithm* clearly provides an extremely efficient way of calculating the sums of inter-molecular similarities that are required for dissimilarity selection. The algorithm does, however, assume that the cosine coefficient is being used to measure the inter-molecular similarities, and it is hence necessary to consider whether such an assumption is appropriate in the present context. As noted previously, the cosine coefficient is an example of an association coefficient, and the various members of this class of coefficients differ only in the manner in which they normalise the dot product of two vectors [3]. Using the notation given previously, examples of association coefficients include the Tanimoto coefficient

$$\sum_{I=1}^{C} M(J,I) \times M(K,I) \Bigg/ \left[\sum_{I=1}^{C} M(J,I)^2 + \sum_{I=1}^{C} M(K,I)^2 - \sum_{I=1}^{C} M(J,I) \times M(K,I)\right],$$

the Dice coefficient

$$2\sum_{I=1}^{C} M(J,I) \times M(K,I) \Bigg/ \left[\sum_{I=1}^{C} M(J,I)^2 + \sum_{I=1}^{C} M(K,I)^2\right]$$

and the cosine coefficient

$$\sum_{I=1}^{C} M(J,I) \times M(K,I) \Bigg/ \sqrt{\sum_{I=1}^{C} M(J,I)^2 \times \sum_{I=1}^{C} M(K,I)^2},$$

inter alia,

An inspection of these three formulae will show that they are very similar to each other (and this resemblance also applies if we consider binary molecular descriptions in which each element is either one or zero, as happens with fragment bit-strings). It is thus hardly surprising that whilst the *absolute values* of the similarities resulting from their use may differ quite considerably, there is generally little difference in the *rankings* that they produce. Indeed, the rankings produced by the Dice and the Tanimoto coefficients can be shown to be identical to each other. The Tanimoto and cosine rankings are not completely monotonic, but comparative experiments suggest that they do give broadly comparable results with a range of types of data [16, 19, 20]. The Tanimoto coefficient is known to perform well in chemical applications of similarity and clustering [3, 4, 17, 18]. It accordingly seems reasonable to assume, firstly, that the cosine coefficient will yield equally effective results and, secondly, that it is feasible to use the centroid algorithm for large-scale dissimilarity selection. The next section describes the experiments we have carried out to test these assumptions.

3 Results

We have carried out three, distinct sets of experiments to evaluate the use of the centroid algorithm described above. Firstly, we have investigated the relationships that exist between the inter-molecular similarities resulting from the use of the cosine and the Tanimoto coefficients, to determine whether they do, in fact, yield broadly comparable results. To ensure the generality of the algorithm, these experiments have involved three different types of molecular characterisation that have been used previously in studies of molecular similarity [17]. Secondly, we have investigated the run-time behaviour of the centroid algorithm using a large set of structures characterised by fragment bit-strings, which provide the most common form of data for clustering and dissimilarity-selection [1, 4, 8–10]. Finally, we have used the algorithm in a fast implementation of the selection procedure described by Bawden [10] and used it to provide an overview of the outputs of several 2-D chemical substructure searches.

3.1 Comparison of Cosine and Tanimoto Coefficients

The first set of experiments used 200 structurally-heterogeneous dye-stuffs, each of which had been characterised in one of three ways: using calculated physical properties; using fragment bit-strings; and using topological indices. Each molecule in the physical-property dataset was characterised by the following real-valued properties calculated using the MOPAC MNDO program [21]: heat of formation, electronic energy, core-core repulsion, dipole moment, number of filled levels, ionization potential and molecular weight. The fragment bit-string dataset was generated by searching for the presence of each of 1536 pre-defined substructural fragments (augmented atoms, strings of four atoms and monocycle descriptors) in each structure and then setting the *I*-th bit if the *I*-th fragment was present. The topological indices were generated using the Molconn-X program of Hall and Kier [22] and comprised 50 real-valued indices of various sorts. The physical-property and topological-index datasets were both standardised so that each variable had a mean of zero and a standard deviation of unity.

Each molecule in turn was selected and its similarity calculated with each of the other 199 structures using both the cosine coefficient and the Tanimoto coefficient. The product moment correlation coefficient was then calculated between the sets of cosine and Tanimoto coefficients for each molecule, and the mean and the standard deviation of this correlation calculated when averaged over all of the 200 molecules in the dataset. The mean correlations, with the standard deviation in brackets, for the physical-property,

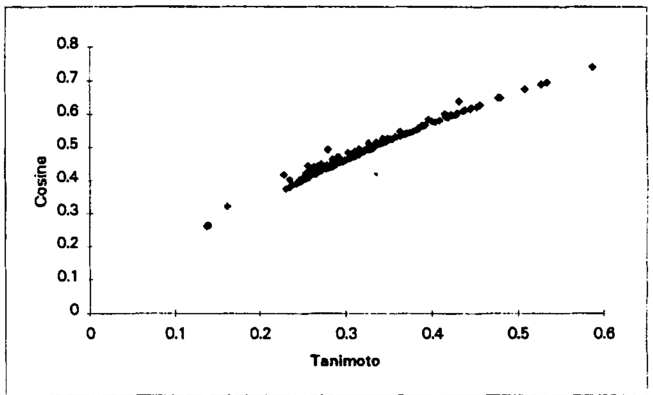

Figure 2. Scatter diagram demonstrating the high degree of correlation between the Tanimoto and cosine coefficients for the comparison of one molecule's bit-string against the bit-strings of 199 other molecules.

Number Of Structures	Centroid Algorithm	Conventional Algorithm
100	0.15	2
300	0.51	24
500	0.94	70
700	1.38	137
900	1.85	227

Figure 3. Run-times, in CPU seconds on an R4000 Silicon Graphics Indigo workstation, for the calculation of the *SUM(J)* values in Figure 1 using the centroid and conventional algorithms.

fragment bit-string and topological-index datasets were 0.955 (0.010), 0.984 (0.007) and 0.947 (0.027), respectively. There is thus an extremely high level of correlation between the similarities resulting from the two types of coefficient, as is evidenced by the scatter diagram of the coefficients for a single molecule that is shown in Figure 2. We hence conclude that the cosine coefficient and the Tanimoto coefficients give similarities that are very highly correlated with each other, and that it is thus appropriate to use the cosine coefficient for clustering and dissimilarity-selection, in just the same way as the Tanimoto coefficient is currently used.

3.2 Calculation of SUM(J) Values

The second set of experiments used a file of 5000 compounds from the Starlist database, each of which was characterised by the fragment bit-strings used previously. Here, we calculated values for *SUM*(J), as defined previously, for each of the structures using both the centroid algorithm and the conventional, $O(N^2)$ algorithm and measured the run-times (in CPU seconds for C programs running on an R4000 Silicon Graphics Indigo workstation) for processing datasets containing between 100 and 900 structures. These CPU times are listed in Figure 3, which demonstrates clearly the differing computational requirements of the two algorithms.

An analysis of the centroid algorithm shows that the calculation of the *SUM*(J) values has a time complexity of order $O(N)$, *i.e.*, it is linear in the size of the dataset that is to be processed. However, Figure 4 shows that the run-times in our timing experiments increased slightly more than linearly when we ran the centroid algorithm with datasets containing up to 5000 compounds, although

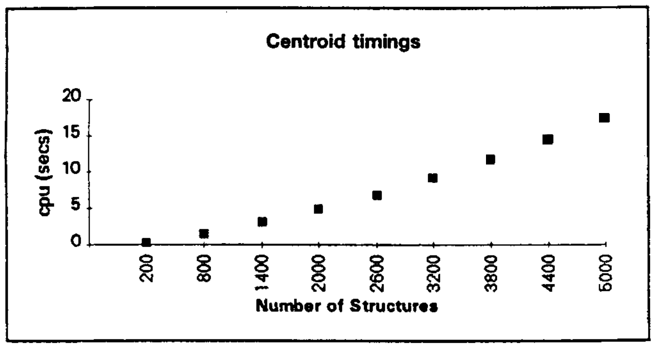

Figure 4. Run-times for processing varying numbers of molecules using the centroid algorithm.

1. Select a molecule at random from *Database* and place it in *Subset*.
2. Identify that molecule in *Database* that is most dissimilar to the molecules already in *Subset* and add that molecule to *Subset*.
3. Repeat Step 2 a total of n-2 times.

Figure 5. Selection of (near-)maximally dissimilar molecules using an algorithm based on that described by Bawden [10]. The algorithm assumes that molecules are selected from *Database* and placed in *Subset*.

the increase is clearly far less than would be the case for an $O(N^2)$ algorithm. Tests with datasets of varying sizes and bit-strings of varying lengths lead us to believe that this is due to memory overheads on the machine that was used in these tests. Even so, the full set of 5000 structures required just 17.39 CPU seconds when the centroid algorithm was used, thus demonstrating its efficiency for processing large datasets. We have recently implemented an alternative version of the algorithm that reads just a single molecule at a time into memory. The resulting increase in input/output means that this version is about two times slower than that described previously (which holds all of the dataset in main memory): it does, however, exhibit the expected linear behaviour and can thus be used even with massive files.

3.3 Dissimilarity Selection

Thus far, we have considered only the calculation of the *SUM*(J) values but these are not sufficient in themselves to ensure the selection of a set of highly dissimilar structures. This is because no attempt has been made to maximise the dissimilarity between the molecules in the chosen sub-set of the database, as against the dissimilarities between molecules in the dataset as a whole. If this is to be achieved, the centroid algorithm needs to be included in a procedure such as that described by Bawden [10], which makes an explicit attempt to identify a (near-) maximally dissimilar sub-set. A simplified version of Bawden's algorithm is shown in Figure 5, where it will be seen that a set of dissimilar structures is grown from an initial molecule by the progressive addition of new molecules that are as dissimilar as possible to those selected thus far. Once the initial compound has been selected, the first invocation of Step 2 of the algorithm shown in Figure 5 will require N-1 similarity calculations, *viz* those between the selected compound and each of the N-1 unselected compounds. Similarly, the i-th iteration, *i.e.*, after i compounds have been selected, will require $i(N-i)$ similarity calculations. Thus, if n compounds are to be selected, the total number of similarity calculations is given by

$$\sum_{i=1}^{n-1} i(N-i)$$

which is equal to

$$\frac{n(n-1)}{2}N - \frac{n(n-1)(2n-1)}{6}.$$

If we assume, as is entirely reasonable for a compound-selection procedure, that $n<<N$, then this corresponds to an expected time complexity of $O(n^2N)$.

1. Run the centroid algorithm to identify that molecule in *Database* that is most dissimilar to all of the other molecules, and place it in *Subset*.
2. For each remaining molecule in *Database*, use the centroid algorithm to calculate the sum of its similarities with the molecules currently in *Subset* and add the molecule with the smallest such sum to *Subset*.
3. Repeat Step 2 a total of n-2 times.

Figure 6. Centroid versions of the Bawden algorithm.

A modification of this procedure that is based on the centroid algorithm is shown in Figure 6. Here, the total number of similarity calculations for Steps 2 and 3 is given by

$$\sum_{i=1}^{n-1} N - i,$$

i.e.,

$$(n-1)N - \frac{n(n-1)}{2}.$$

Assuming again that $n << N$, this corresponds to an expected time complexity of *O(nN)*, thus demonstrating the substantial savings in computation that are to be expected from the use of the centroid algorithm in this way.

We have applied the procedures shown in Figure 6 to the selection of compounds from the outputs of the five 2-D substructure searches shown in Figure 7. The searches were carried out on a file of 117,650 structures from the National Cancer Institute using the UNITY chemical database system. The resulting sets of hit compounds were then processed to identify sets of 20 dissimilar molecules in each case. The effectiveness of the dissimilarity-selection procedure was investigated by calculating the sum of the inter-molecular cosine similarities for all 380 (*i.e.*, 20 × 19) pairs of selected dissimilar compounds. For comparison, we have calculated analogous sums of similarities for randomly-selected sets of 20 compounds from each of the five outputs. In each case, 100 sets of 20 compounds were generated, and the mean and the standard deviation of the sum of similarities calculated. The resulting figures are shown in Figure 8 (where the column headed Centroid Selection relates to the algorithm shown in Figure 6 and where the column headed Genetic Algorithm is discussed below). It will be seen that the centroid-selection procedure yields sets of compounds that are significantly more dissimilar than those obtained by random selection, thus demonstrating the effectiveness of the algorithm.

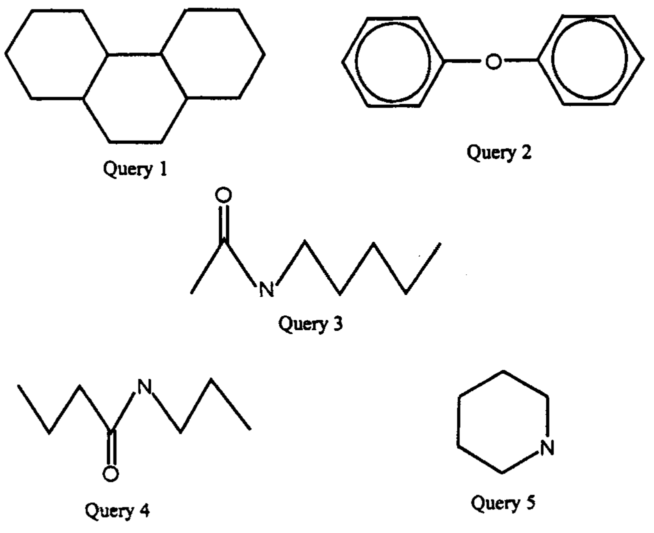

Figure 7. The five substructure search queries that were used to evaluate the effectiveness of the dissimilarity-selection procedure shown in Figure 6.

Query	Hitlist Size	Centroid Selection	Random Selection	Genetic Algorithm
1	826	146.9	218.9 (9.8)	149.9
2	737	126.8	171.8 (6.2)	128.7
3	1735	122.8	172.3 (6.5)	123.4
4	1306	140.3	191.2 (7.5)	142.0
5	3054	134.16	194.0 (7.9)	135.7

Figure 8. Selection of 20 dissimilar compounds from the hitlists resulting from the five substructure searches shown in Figure 7. The last three columns contain the sums of the inter-molecular cosine similarities for the 20 compounds selected in each case. The Random Selection figures are mean values averaged over 100 randomly-generated sets, with the standard deviation in brackets.

3.4 A Genetic Algorithm for Dissimilarity Selection

Genetic algorithms are an increasingly popular means of identifying good, approximate solutions to combinatorial optimisation problems [23]. They have already been used for several chemical applications, such as flexible docking [24] and conformational analysis [25], and we have thus developed such an algorithm, hereafter a GA, for the inherently combinatorial problem of selecting maximally-dissimilar sets of compounds.

A chromosome in our GA contained n integers, these representing a possible subset of n compounds selected from a dataset of size N compounds. The initial members of each chromosome in the population were selected using roulette-wheel selection on the inverse *SUM(J)* values for the N compounds in the dataset, so as to try to ensure the selection of compounds that would be expected not to have a high degree of similarity with each other. Simple one-point crossover was used, with a check to ensure that a child chromosome resulting from crossover did not contain more than a single occurrence of any one compound. The mutation operator randomly replaced an individual compound within a chromosome by another member of the dataset, with the replacement being chosen either by using roulette-wheel selection on the $SUM(J)$ values or by using the centroid algorithm to add the molecule that was most dissimilar to the other n-1 members of the chromosome. The fitness function was the sum of the $n(n-1)$ similarities between the current n members of a chromosome, this being calculated rapidly using a suitably-modified version of the centroid algorithm, with the individual fitness values being linearly normalised to obtain an appropriate selection pressure.

A steady-state-with-no-duplicates GA was implemented, and a large number of runs carried out using a wide range of crossover and mutation rates. The GA was successful, in that the sums of the $n(n-1)$ similarities for each of the chromosomes in a population fell rapidly from their values in the initial generation. However, we were never able to obtain sets of molecules that were as dissimilar as those identified by the algorithm shown in Figure 6, despite running the GA for up to 100 000 generations. The results that were obtained are exemplified by those listed in the final column of Figure 8, which are slightly, but consistently, less than those in the column headed Centroid Selection.

4 Conclusions

This paper describes an algorithm for identifying those molecules in a dataset that have the lowest sum of pair-wise similarities with the other members of the dataset. The algorithm is applicable to any type of representation that characterises a molecule by a set of attribute values, such as the physical properties, chemical fragments or topological indices considered in our experiments. Its *O(nN)* expected time complexity means that it is sufficiently fast in operation to be used for applications such as the selection of compounds from a corporate database for random screening, the processing of large substructure-search outputs and the design of combinatorial libraries.

Acknowledgements

This work was funded by Tripos Inc. and the Engineering and Physical Sciences Research Council. We thank Biobyte Corporation for provision of the Starlist database, Tripos Inc. for provision of the National Cancer Institute database and of the UNITY software, Geoffrey Downs and Richard Horobin for assistance with the preparation of the test data, and Gareth Jones for commenting on an earlier draft of this paper.

5 References

[1] Downs, G.M. and Willett, P., in: van de Waterbeemd, H. (Ed.), *Advanced Computer-Assisted Techniques in Drug Discovery*, VCH, New York, pp. 111-130 (1994).
[2] Taylor, R., *Journal of Chemical Information and Computer Sciences, 35*, 59-67 (1995).
[3] Willett, P., *Similarity and Clustering in Chemical Information Systems*. Research Studies Press, Letchworth (1987).
[4] Barnard, J.M. and Downs, G.M., *Journal of Chemical Informations and Computer Sciences, 32*, 644-649 (1992).
[5] Willett, P., Winterman, V. and Bawden, D., *Journal of Chemical Information and Computer Sciences, 26*, 109-118 (1986).
[6] Hodes, L., *Journal of Chemical Information and Computer Sciences, 29*, 66-71 (1989).
[7] Lajiness, M., in: Rouvray, D.H. (Ed.), *Computational Chemical Graph Theory*, Nova Science Publishers, New York, pp. 299-316 (1990).
[8] Lajiness, M., in: Silipo, C. and Vittoria, A. (Eds.), *QSAR: Rational Approaches to the Design of Bioactive Compounds*, Elsevier Science Publishers, Amsterdam, pp. 201-204 (1991).
[9] Bawden, D., in: Johnson, M.A. and Maggiora, G.M. (Eds.), *Concepts and Applications of Molecular Similarity*, John Wiley, New York, pp 65-76 (1990).
[10] Bawden, D., in: Warr, W.A. (Ed.), *Chemical Structures 2*, Springer Verlag, Heidelberg, pp. 383-388 (1993).
[11] Martin, E.J., Blaney, J.M., Siani, M.A., Spellmeyer, D.C., Wong, A.K. and Moos, W.H., *Journal of Medicinal Chemistry, 38*, 1431-1436 (1995).
[12] Voorhees, E.M., *Information Processing and Management, 22*, 465-476 (1986).
[13] El-Hamdouchi, A. and Willett, P.; *Information Processing and Management, 24*, 17-22 (1988).
[14] Biru, T., El-Hamdouchi, A., Rees, R.S. and Willett, P., *Journal of Documentation, 45*, 85-109 (1989).
[15] Cutting, D.R., Pedersen, J.O., Karger, D. and Tukey, J.W., *Proceedings of the Annual International Conference on Research and Development in Information Retrieval, 15*, 318-329 (1992).
[16] Willett, P. and Winterman, V., *Quantitative Structure-Activity Relationships, 5*, 18-25 (1986).
[17] Downs, G.M. and Willett, P., *Reviews in Computational Chemistry*, in press.
[18] Stanton, D.T., Murray, W.J. and Jurs, P.C., *Quantitative Structure-Activity Relationships, 12*, 239-245 (1993).
[19] Ellis, D., Furner-Hines, J. and Willett, P., *Perspectives in Information Management, 3*, 128-149 (1994).
[20] Hubalek, Z., *Biological Reviews of the Cambridge Philosophical Society, 57*, 669-689 (1982).
[21] Stewart, J.P., *Journal of Computer-Aided Molecular Design, 4*, 1-105 (1990).
[22] Hall, L.H. and Kier, L.B., *Molconn-X User's Guide*, Hall Associates Consulting, Quincy MA (1991).
[23] Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York (1991).
[24] Jones, G., Willett, P. and Glen, R.C., *Journal of Molecular Biology, 245*, 43-53 (1995).
[25] Brodmeir, T. and Pretsch, E., *Journal of Computational Chemistry, 15*, 588-595 (1994).

Received on May 29th, 1995;
accepted on July 30th, 1995.