A set-theoretic approach to database searching and clustering

Antje Krause and Martin Vingron
Deutsches Krebsforschungszentrum (DKFZ), Theoretische Bioinformatik, Im Neuenheimer Feld 280, D-69120 Heidelberg, Germany

Received on October 9, 1997; revised on December 9, 1997; accepted on December 17, 1997

Abstract

**Motivation:** In this paper, we introduce an iterative method of database searching and apply it to design a database clustering algorithm applicable to an entire protein database. The clustering procedure relies on the quality of the database searching routine and further improves its results based on a set-theoretic analysis of a highly redundant yet efficient to generate cluster system.

**Results:** Overall, we achieve unambiguous assignment of 80% of SWISS-PROT sequences to non-overlapping sequence clusters in an entirely automatic fashion. Our results are compared to an expert-generated clustering for validation. The database searching method is fast and the clustering technique does not require time-consuming all-against-all comparison. This allows for fast clustering of large amounts of sequences.

**Availability:** The resulting clustering for the PIR1 (Release 51) and SWISS-PROT (Release 34) databases is available over the Internet from http://www.dkfz-heidelberg.de/tbi/services/modest/browsysters.pl.

**Contact:** a.krause@dkfz-heidelberg.de; m.vingron@dkfz-heidelberg.de

Introduction

Searching a sequence database with a query sequence looking for homologues has become a routine operation. This is largely due to the availability of fast heuristic search routines like the BLAST (Altschul et al., 1990) and FASTA (Pearson and Lipman, 1988) families of programs. High-quality searches with the Smith–Waterman (Smith and Waterman, 1981) algorithm can also be performed on a database, although they take much longer and are therefore done very rarely or on specialized hardware (Brutlag et al., 1993; Starrock and Collins, 1993; Compugen Ltd, 1996; Hughey, 1996).

In spite of the high degree of sophistication of existing approaches, it is still virtually impossible to identify quickly and clearly a cluster of sequences that a given query sequence belongs to. Generally, the cluster will either be trivial, e.g. all similar sequences scoring higher than some safe cut-off, or there will be a twilight zone and unrelated sequences may score high enough to be mistakenly included in the cluster. Here we want to introduce a procedure we call ‘SYSTEmatic Re-Searching’ (SYSTERS). Given a query sequence, SYSTERS delineates a cluster of sequences that the query is a member of. Membership in such a cluster is not given with some probability or other quantitative measure. A database sequence is either in the SYSTERS cluster or it is not.

SYSTERS systematically iterates database searching with a sequence identified in the current search. In contrast to PSI-BLAST (Altschul et al., 1997) which generates a position-specific score matrix as input for the next search, or the exploratory approach of Neuwald et al. (1997), we take a strictly conservative approach where only sequences of high similarity to the last query are used for re-searching. This approach allows us to define a set of sequences related to the query where false positives are extremely unlikely.

In this paper, considerable effort will be invested into proving that the clusters delineated by SYSTERS are correct and informative. At the heart of the validation of SYSTERS is the detailed analysis of the consistency of SYSTERS in identifying the same cluster of sequences independent of which of the sequences from the cluster was used to start the search. This validation relies on a set of SYSTERS searches where each sequence in a database is used to start a search. This results in a highly redundant family of clusters where each sequence may have been identified repeatedly by searches seeded by several related sequences.

This set of clusters turns out to be extremely valuable for database clustering. We designed a set-theoretic clustering method that uses this information to partition a large fraction of a sequence database into disjoint clusters. This SYSTERS-based clustering procedure is fully automatic, does not require pairwise comparisons between sequences, and is extremely fast. In contrast to distance-based clustering algorithms, this set-theoretic clustering method does not enforce any further data-dependent decisions from the user: a comparatively small part of the database will not be assigned to disjoint clusters but to clusters which overlap each other. Thus, the method is self-validating in the sense that errors would manifest themselves in further overlaps...
Set-theoretic approach to database searching

**Fig. 1.** Excerpt of the BLASTP output searching the SWISS-PROT database Release 34 with the SYMC_YEAST sequence. Sequences identified by the SYSTERS procedure are marked with an ‘X’.


SYSTERS-based clustering distinguishes by itself between cases where it can make a decision and cases where it cannot.

This paper is organized as follows. First, we introduce the database searching method SYSTERS. To give an intuition of how it performs, we will present a few examples before proceeding to a systematic evaluation. This systematic evaluation is based on the newly introduced notion of the consistency of a database searching method. The following section will then turn to database clustering by first analyzing the set-theoretic features of the complete set of SYSTERS clusters. This motivates the definition of the set-theoretic, SYSTERS-based clustering algorithm. The biological validity of the computed clustering will be established by a detailed comparison to an expert-made biological grouping available for the PIR1 database (George *et al.*, 1997). As a challenging large-scale test, we finally describe the results of automatically clustering the SWISS-PROT database (Bairoch and Apweiler, 1997).

**Database searching by SYSTEmatic Re-Searching**

**Searching algorithm**

SYSTERS is an algorithm that iterates traditional protein sequence database searches in a specific way in order to delineate for a given protein sequence a set of related ones. We call the sequence for which we want to extract its related sequences from a database the ‘seed’. SYSTERS starts with a database search using the seed sequence as a query. In our implementation, we used the BLASTP program (Altschul *et al.*, 1990) as the single-search subroutine. All hits in this search that are highly significant—we chose a P-value of $10^{-30}$ as cut-off—are retained. The lowest scoring sequence from this set is used as query for a next database search to explore the sequence space below the cut-off for sequences possibly related to the seed. The procedure is iterated either until the current search provides no not yet accepted sequence above the cut-off or until the new search result has no sequence in common with the set of accepted hits from the first search. Note that this procedure does not rank hits. A set of sequences is returned and we call this set the cluster obtained with the seed query. Thus, the main advantage of SYSTERS lies in the fact that homologies are not all scored in relation to one sequence, but the set of sequences already identified supplies further queries that allow the addition of other sequences to a cluster.

**Description of clusters**

Before embarking on a systematic evaluation, we briefly summarize some typical results of SYSTERS in an anecdotal way. Sequences that do not share domains with other families generally pose no difficulty to SYSTERS. For example, searching the SWISS-PROT database (Bairoch and Apweiler, 1997, Release 34) with the methionyl tRNA synthetase sequence from yeast (SYMC_YEAST) identifies exactly all other methionyl tRNA synthetase sequences from the database, as shown in Figure 1. Similar examples, where exactly the sequences with the same SWISS-PROT description line are found, are abundant.

One would expect common domains between proteins to create more of a problem. The homeobox is a DNA binding domain common to many different gene families. The homeobox itself is generally highly conserved even across
families, while the genes may show no other similarity. Prominent gene families are, among others, the engrailed family, the ultrabithorax (убx) family and the hox family, all of which are involved in pattern formation (Gehring et al., 1994). In one test, we used the human engrailed homeobox gene sequence (HME1_HUMAN) as seed for SYSTERS. The resulting cluster identified all homeobox protein sequences from SWISS-PROT that contained the word ‘engrailed’ in their annotation with the exception of two entries. These two were annotated ‘engrailed-like’ and one of them was a fragment of only 60 amino acids length.

By comparing SYSTERS results to their corresponding search outputs generated by a rigorous Smith–Waterman (Smith and Waterman, 1981; Pearson, 1991; SSEARCH program) alignment, we studied down to which significance level in the Smith–Waterman search cluster members were identified and also which higher scoring sequences were not included into the SYSTERS clusters. The statistical significances assigned by the SSEARCH program give an impression of how easy or difficult identification of these homologues is. In the case of the engrailed homeobox genes, the lowest member sequence showed a significance of only 0.00014, while several sequences of higher significance (up to 0.0000011) were rejected. In other instances, SYSTERS correctly retrieved sequences only above a very stringent significance level. In particular, this shows that SYSTERS is capable of circumventing the problems associated with fixing a significance cut-off in traditional database searching.

**Consistency of cluster identification**

Database searching routines are usually judged based on speed and sensitivity. In terms of speed, most SYSTERS runs require 2–3 calls to a fast searching routine like BLAST on average, depending on the size of the cluster that a query is a member of (in some rare cases there were up to 14 calls). SYSTERS is thus still very fast. Sensitivity is usually judged by evaluating the number of false positives and false negatives (Pearson, 1995). False positives are sequences that are not related to the query and yet are picked up by the search. False negatives are sequences that are not identified by the search although they are in fact related to the query. In traditional search routines, these notions need to be defined relative to some cut-off, whereas SYSTERS, returning a cluster and not a ranking, implies clear definitions of false positives and false negatives.

While the SYSTERS clusters in the above examples all make sense, establishing their biological validity is a process which is difficult to automate. Automatic schemes for checking database search sensitivity are usually based, for example, on PROSITE (Bairoch et al., 1997) assignments of certain motifs. These allow the user to decide automatically whether a sequence identified in the search contains the same motif as the query or not. SYSTERS clusters tend to agree with the annotation in SWISS-PROT, which is not standardized and thus difficult to use for automatic validation. In particular, there may be description lines stating that a sequence is a ‘hypothetical protein’ or that the information was derived by similarity.

Instead of using database annotations for validation of the clustering, we introduce a new, formal criterion for the quality of SYSTERS searches. The focus of the new quality criterion is on internal consistency of a search. If one SYSTERS seed identifies a certain cluster, then every other cluster member, when used as a seed, should identify the same or at least a very similar cluster. If a set of sequences indeed forms a cluster, this implies that the same cluster should be identified independently of which cluster member is used as seed for the search. To check this criterion, we generated one cluster per sequence in the database using SYSTERS. The resulting, very large set of SYSTERS clusters provides the information to check internal consistency.

Denote the set of SYSTERS clusters for all queries from a database as the complete cluster set (CCS). For every sequence in the database, we computed the following quantities. First, the set theoretic union and set theoretic intersection of all clusters in the CCS that contain the sequence are identified. We denote the cardinality of the union of clusters containing sequence s by $U(s)$ and the cardinality of the intersection of clusters containing sequence s by $I(s)$. Ideally, all members of a cluster would identify the cluster in exactly the same way. Then for each of the sequences in the cluster, the union and the intersection of the clusters containing this sequence would coincide, and thus their cardinalities would agree. However, if a sequence constitutes a false positive for a specific search, then it is not only contained in the CCS cluster of its own related sequences, but also in one or more other clusters where it appeared erroneously. Thus, for a false positive, $U(s)$ will be significantly greater than $I(s)$. On the other hand, suppose a sequence is a false negative in some search. Then of the CCS clusters that try to describe these related sequences, one or more will lack this one sequence. This, in turn, results in a difference between union and intersection of those clusters where other cluster members are contained. For such a sequence, the union will exceed the intersection by at least the false negative.

This criterion of internal consistency was systematically tested by performing SYSTERS searches for the PIR1 database (George et al., 1997, Release 51). This release of PIR1 contains 13 489 sequences. Figure 2 shows a three-dimensional histogram of the number of sequences at their respective values of $U(s)$ and $I(s)$ for all database entries. The sequences for which SYSTERS is perfectly consistent [$U(s) = I(s)$] are on the main diagonal. The sum of the bars on the main diagonal is 9659, corresponding to 72% of sequences. The highest peak on the main diagonal refers to 2694 single-sequence clusters. Then there is a bar with 11 48 sequences contained in clusters of two sequences, and so on. Off diag-
Fig. 2. Three-dimensional representation of $U(s)$, $I(s)$ and the corresponding number of sequences for all PIR1 database entries. Perfectly clustered sequences are on the main diagonal [$U(s) = I(s)$]. The insert is a zoom into the left part of the histogram [$U(s) \leq 50$]. The highest peak is indicated only by an arrow and in fact has a height of 2694.

SYSTERS-based database clustering

Dissecting the complete cluster set

The above analysis led us to sorting database entries into their respective clusters based on the SYSTERS-generated cluster system. Here, we do not aim at a hierarchical clustering. We only wish to compute disjoint clusters, i.e. a meaningful partitioning of the data set.

The first observation is that a CCS cluster all of whose members have identical $U(s)$ and $I(s)$ is already perfectly defined. It cannot have any overlap with other clusters and each of its member sequences identifies the cluster in exactly the same way. These perfect clusters may of course be trivial in the sense that they contain only a single sequence. Even in this case, though, one knows in particular that the union of clusters containing it is a one-element set, which implies that there are no other CCS clusters that contain this one sequence. In the case of the PIR1 database, there are 2694 such perfect, single-sequence clusters. Further, 1421 perfect clusters contain 6688 sequences altogether.
Fig. 3. Set-membership matrix for all tryptophan synthase proteins contained in the PIR1 database and clustered together by SYSTERS. Columns represent the clusters found with the seed mentioned on top. Rows show all clusters which the sequence on the left is a member of.

Since a perfect cluster is disjoint from any other CCS cluster, one may consider the sequences contained in the set of perfect clusters as perfectly sorted. Among the remaining clusters, accounting for 4107 of 13,489 sequences, there exist inclusions and overlaps. One cluster being included by another one typically is the consequence of false negatives in a search. When the same cluster is identified using another seed, the (formerly) false negative may be found and the resulting cluster contains the other cluster. Consequently, one wishes to use the larger cluster for the partitioning. However, there may be another cluster containing this one, and so on. Therefore, one needs to check for chains of inclusions among clusters. Only the final, largest cluster in such a chain is a candidate for our database clustering. However, this set of maximal clusters falls into two groups again. One group, we call them separate maximal clusters, are those maximal clusters that are disjoint from any other maximal cluster. The final, residual group of clusters are maximal clusters that overlap with other maximal clusters. These we call the overlapping maximal clusters.

Obviously, neither the separate maximal clusters overlap each other nor can a separate maximal cluster overlap with a perfect cluster. For the PIR1 database, there are 152 separate maximal clusters comprising 2898 sequences. As an example of the inner structure of a separate maximal cluster, Figure 3 shows such a cluster and all the clusters it contains. In this case we are looking at tryptophan synthase proteins. Ten of 16 sequences, when used as seed, identified the cluster that is also the separate maximal cluster. Six sequences in the SYSTERS search identified only a smaller subset. Perfect and separate maximal clusters together comprise 91% of the database sequences. There remain 112 overlapping maximal clusters accounting for the missing 1209 sequences (9% of the sequences, Table 1).

Since overlapping maximal clusters do not constitute a partitioning of the data, we sort them into connected components where by a connected component we mean a set of clusters which are all linked by overlaps. Thus, a first cluster in this connected component might overlap another one, which in turn overlaps a third one, and so on.

The overlapping maximal clusters for the PIR1 database fall into 23 connected components. The connected components are precisely those cases where SYSTERS cannot delineate separate clusters. Typical members of this group were kinases and proteins that contain a kinase domain. Another problematic case comprised coiled-coil-containing proteins like myosin which are distributed over 26 clusters forming one connected component. Several other connected components were trivial in the sense that the overlap between the clusters contained most of the sequences in the connected component. Figure 4 shows the clusters for *Escherichia coli* ribosomal protein L12 as an example. For such cases of connected components, one may safely choose the union of the clusters in the connected component as a new cluster to use instead. However, such a decision should be taken by an expert and is not part of our algorithm.

<table>
<thead>
<tr>
<th>No. of sequences (%)</th>
<th>No. of clusters (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-sequence clusters</td>
<td>2694 (20)</td>
</tr>
<tr>
<td>Perfect clusters</td>
<td>6688 (50)</td>
</tr>
<tr>
<td>Separate maximal clusters</td>
<td>2898 (21)</td>
</tr>
<tr>
<td>Overlapping maximal clusters</td>
<td>1209 (9)</td>
</tr>
<tr>
<td>Total</td>
<td>13489 (100)</td>
</tr>
</tbody>
</table>

**Clustering algorithm, implementation, performance**

From the above description of the set-theoretic features of the CCS, we extract the following SYSTERS-based clustering algorithm for a protein sequence database. Input to the algorithm is a set of sequences. Its output is a collection of sets of sequences, the clusters.

1. Compute SYSTERS clusters for all sequences in a database.
2. Extract a subset of clusters that partition the database:
   (a) For all identical clusters, eliminate all but one.
   (b) For any two pairs of clusters where one includes the other, eliminate the smaller one. Repeat this until no inclusions are left.
   (c) From this set, accept the clusters which do not overlap any other cluster. These are the perfect and the separate maximal clusters.
Fig. 4. Set-membership matrix for all *Escherichia coli* ribosomal proteins L12 contained in the PIR1 database. The columns on the right show the resulting overlapping maximal clusters after resolving all inclusions.

3. Compute the connected components of the remaining overlapping maximal clusters.

In this description of the procedure, $U(s)$ and $I(s)$ are not considered any more because the perfect clusters can be generated in one step together with the separate maximal clusters. Overlapping maximal clusters are grouped into connected components and possible decisions about merging the clusters from a connected component into one cluster are left to an expert.

The run time for the database clustering is dominated by the BLAST runs performed for the SYSTERS searches. In our implementation, all BLASTP searches are first done in parallel on a workstation cluster consisting of eight SUN Ultra workstations. Approximately 60 000 searches (performed for the clustering of the SWISS-PROT database) took ∼5 days. Then, based on the BLASTP output, the CCS of SYSTERS clusters is derived by a script written in Perl. A program written in C++ using the LEDA library (Mehlhorn and Näher, 1995) then executes the above procedure extracting the database clustering. The overall execution time for processing the CCS using those two programs is of the order of a few hours.

**Comparison of SYSTERS clusters to PIR superfamilies**

Ultimately, the criterion to judge a database clustering is its biological validity. The PIR1 database (George et al., 1997) is very carefully annotated with superfamilies (Barker et al., 1996) that the sequences are grouped into. This assignment is done by running a FASTA search for every new sequence. Candidate sequences that meet rather stringent requirements are routinely classified and aligned automatically to the corresponding protein family; others are examined and classified by an expert. Therefore, we decided to compare the SYSTERS-based clustering of PIR1 described in the prior section to the PIR1 superfamily annotation.

We exclude the 1209 sequences contained in overlapping maximal clusters from the comparison since we have not unambiguously assigned them to a cluster. Our comparison is thus based on 4267 pairwise disjoint SYSTERS clusters and 3379 PIR1 superfamilies—each set containing 12 280 sequences.

As shown in Table 2, the SYSTERS-based clustering sorts 51% of the sequences into clusters that are identical to a PIR1 superfamily. The largest of these identical clusters (ribulose-bisphosphate carboxylase large chain) contains 59 sequences. A total of 46% of the sequences are involved in inclusions of PIR1 superfamilies into SYSTERS clusters or vice versa. Analysis of these inclusions shows that the SYSTERS-based clustering tends to be finer than the superfamily classification: 80% of these sequences are members of a SYSTERS cluster included in a PIR1 superfamily, while only 20% fall into a SYSTERS-generated cluster containing two or more superfamilies. In only 22 cases did we observe that a PIR1 superfamily and a SYSTERS cluster overlapped without one containing the other (21 cases), or that a SYSTERS cluster contained elements from different PIR1 superfamilies (one case, cytochrome c/cytochrome c₆).

<table>
<thead>
<tr>
<th>No. of sequences</th>
<th>(%)</th>
<th>No. of PIR1 superfamilies</th>
<th>No. of SYSTERS clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIR1 = SYSTERS</td>
<td>6209 (51)</td>
<td>2553</td>
<td>2553</td>
</tr>
<tr>
<td>PIR1 &gt; SYSTERS</td>
<td>4598 (37)</td>
<td>464</td>
<td>1526</td>
</tr>
<tr>
<td>PIR1 &lt; SYSTERS</td>
<td>1137 (9)</td>
<td>315</td>
<td>112</td>
</tr>
<tr>
<td>PIR1 ≠ SYSTERS</td>
<td>336 (3)</td>
<td>47</td>
<td>76</td>
</tr>
<tr>
<td>Total</td>
<td>12289 (100)</td>
<td>3379</td>
<td>4267</td>
</tr>
</tbody>
</table>

PIR1 = SYSTERS: the PIR1 superfamily classification and the corresponding SYSTERS cluster are identical; PIR1 > SYSTERS: several SYSTERS clusters are proper subsets of one PIR1 superfamily; PIR1 < SYSTERS: several PIR1 superfamilies are proper subsets of one SYSTERS cluster; PIR1 ≠ SYSTERS: PIR1 superfamilies and SYSTERS clusters are involved in an overlap.
Comparing the SYSTERS clustering of the SWISS-PROT database to single BLAST clustering

The same algorithm for database clustering was applied to the SWISS-PROT database (Release 34) containing 59,021 sequences. The statistics for this clustering are shown in Table 3a. Twenty-four percent of the sequences ended up in single-sequence clusters and a further 33% are contained in perfect clusters. The remaining 43% fall into separate and overlapping maximal clusters as follows. After removing all inclusions from the CCS, there are 735 separate maximal clusters comprising 23% of the sequences and 147 connected components with the remaining 20%. The largest of these connected components comprises 4000 sequences contained in 683 clusters and is a union of all problematic cases already mentioned for PIR1. All other connected components are again trivial.

The differences between the SYSTERS clustering of the PIR1 and the one of the SWISS-PROT arise from the sequence composition of these databases. While the PIR1 sub-section of the PIR database only consists of well-known and characterized protein sequences, the SWISS-PROT database also contains fragments, tentative and hypothetical proteins, which mostly ended up in single sequence and overlapping maximal clusters.

As an example, consider the family of engrailed homeobox genes discussed earlier. In the final clustering, all 27 engrailed or engrailed-like genes form one separate maximal cluster.

Calculating $U(s)$ and $I(s)$ as defined earlier for the CCS of SWISS-PROT, the two numbers agree for 59% (34,828 sequences) of the sequences. The corresponding sequences are therefore already assigned to exactly one cluster.

Comparing this clustering to a simple BLAST clustering without iteration where we accepted all sequences above the same conservative cut-off of $10^{-30}$ as the cluster of sequences found by this BLAST query, With this clustering, we obtained the following results (Table 3b). As expected, there is exactly the same number of single-sequence clusters (24% of all sequences), contained in this cluster set. The total number of perfect clusters decreased, now containing 26% of the sequences. After removing all inclusions from this CCS, there are 894 separate maximal clusters comprising 17% of the sequences, so the average size of a maximal cluster is much smaller than in the SYSTERS clustering. The number of connected components increased significantly to 518 with the remaining 33% of all sequences contained in 3869 clusters. The largest of these connected components again comprises 4000 sequences, but this time contained in 980 clusters.

Calculating $U(s)$ and $I(s)$ for this CCS, the two numbers agree for 51% (29,829 sequences) of the sequences. It is obvious that the iterative step of the SYSTERS-based clustering in comparison to the single BLAST clustering is capable of including more sequences in the set of disjoint clusters.

The SYSTERS clustering of the PIR1 and the SWISS-PROT databases is being made available on the World Wide Web. The clusters are searchable via the database annotations of the sequences in the clusters. Every cluster is linked to its set-membership matrix (cf. Figure 3), an automatically generated multiple alignment of the sequences in the cluster.
and the corresponding unrooted tree calculated by ClustalW 1.7 (Thompson et al., 1994).

Discussion

We introduced a novel database search method, SYSTERS, that iterates a traditional search procedure like BLAST and produces clusters of sequences related to the query. We showed that this procedure is to a large degree internally consistent in the sense that the resulting clusters in most cases show little dependence on the specific query. This has provided the foundation for a database clustering method that sorts sequences into clusters, a large fraction of which is pairwise disjoint.

In comparison to distance-based (Sneath and Sokal, 1973) and graph-theoretical (Matsuda et al., 1996) clustering methods, the clusters produced by the SYSTERS clustering method are determined by their set-theoretical relationship.

The resulting set of clusters is self-validating since problems become manifest in overlaps between clusters. The real success of the method lies in automatically delineating the subset of sequences that can be sorted into non-overlapping clusters. Since no special procedure is applied to enforce the disjointness, it may be taken as an indicator that this information is in fact correct. Problematic cases reveal themselves through overlaps and are collected in the connected components. Thus, the limit of automation is pushed as far as possible without enforcing arbitrary decisions.

The granularity of this clustering is determined by the data and not by a user-supplied data-dependent cut-off. Mostly, the clustering is conservative and clusters do not comprise evolutionarily diverse families. Detailed inspection of the clusters, however, provides exceptions to the rule and diverse members can be found in some clusters. Since the clusters are non-overlapping, they can by definition not represent common domains between different protein families. Rather, knowledge about common protein folds or common domains needs to be referenced on top of this clustering. In contrast to algorithms like DOMAINER (Sonnhammer and Kahn, 1994) or the BLOCKS database (Henikoff and Henikoff, 1991), our method attempts only to cluster together sequences that share global, or at least very strong, similarity. Having established an automatic and efficient method for this conservative clustering should provide a sound basis for the automation of the second level of analysis.

At the heart of the clustering is the SYSTERS database searching routine. The specific difference between SYSTERS and traditional methods is that, instead of simply accepting sequences down to some threshold, the new method dynamically changes its view on the cluster under study. This is achieved by using another sequence as query and choosing this one to be as different from the prior query as is still safe, i.e. above a cut-off that by itself and for a single search would be very conservative. The fact that this change of perspective allows the delineation of clusters with the described degree of consistency suggests that these clusters are in fact well separated in the space of sequences.

The possibilities opened up by this new computational tool are manifold. As mentioned, further correspondences between sequences can be generated and annotated on top of the SYSTERS-based clustering. Furthermore, the inner, evolutionary structure of the clusters can be studied. This is expected to be of special interest for the case of separate maximal clusters because these tend to have subgroups. The overlapping maximal clusters are likely to contain interesting cases of common domains. New options also arise in the field of automatic annotation and the analysis of new sequence sets. Any set of unknown sequences can simply be merged into an existing, clustered database using SYSTERS. Here, the new sequence need not be compared to the set of clusters using computationally expensive tools like profile matching or hidden Markov models. Instead, new sequences can be searched versus the database itself using SYSTERS. The resulting clusters can be compared to the existing ones and due to the high reliability of the assignment to clusters, functional inference can be made based on this information. Traditional problems with cut-offs and decisions about where a new sequence belongs are considerably alleviated by such a procedure.

Acknowledgements

We thank Marc Rehmsmeier for the creation of the World Wide Web interface, including the generation of the multiple alignments and the corresponding trees. We would like to acknowledge financial support from the BMBF (Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie).

References


Compugen Ltd (1996) BIOCCELERATOR. http://www.compu-gen.co.il/Bic/


