BIOINFORMATICS

Vol. 14 no. 2 1998
Pages 144–150

DIVCLUS: an automatic method in the GEANFAMMER package that finds homologous domains in single- and multi-domain proteins

Jong Park[1,2] and Sarah A. Teichmann[1]

[1]MRC Laboratory of Molecular Biology and [2]Centre for Protein Engineering, Hills Road, Cambridge CB2 2QH, UK

Received on September 29, 1997; revised on November 13, 1997; accepted on November 17, 1997

Abstract

Motivation: *Large-scale determination of relationships between the proteins produced by genome sequences is now common. All protein sequences are matched and those that have high match scores are clustered into families. In cases where the proteins are built of several domains or duplication modules, this can lead to misleading results. Consider the very simple example of three proteins: 1, formed by duplication modules A and B; 2, formed by duplication modules B′ and C; and 3, formed by duplication modules C′ and D. Duplication modules B and B′ are homologous, as are C and C′. Matching the sequences of 1, 2 and 3 followed by simple single-linkage clustering would put all three in the same family, even though proteins 1 and 3 are not related. This is because the different parts of 2 match 1 and 3. This paper describes a procedure, DIVCLUS, that divides such complex clusters of partially related sequences into simple clusters that contain only related duplication modules. In the example just given, it would produce two groups of sequences: the first with domains B of sequence 1 and B of sequence 2, and the second with domain C of sequence 2 and C of sequence 3. DIVCLUS is part of a package called GEANFAMMER, for GEnome ANalysis and protein FAMily MakER. The package automates the detection of families of duplication modules from a protein sequence database.*

Results: *DIVCLUS has been applied to the division of single-linkage clusters generated from the protein sequences of six completely sequenced bacterial genomes. Out of 12 013 genes in these six genomes, 4563 single- and multi-domain sequences formed 1071 complex clusters. Application of the DIVCLUS program resolved these clusters into 2113 clusters corresponding to single duplication modules.*

Availability: *The perl5 program and its documentation are available at the following address: http://www.mrc-lmb.cam.ac.uk/genomes/ and by anonymous ftp at ftp.mrc-lmb.cam.ac.uk in the directory /pub/genomes/Software/.*

Contact: *sat@mrc-lmb.cam.ac.uk; jong@mrc-lmb.cam.ac.uk*

Introduction

The genome sequencing projects and the worldwide determination of individual sequences are producing a vast number of protein sequences. To introduce some order into these data, it is often desirable to group the sequences in a database, in a genome or in a set of genomes, into protein families. This provides evolutionary, functional and structural information on the sequences. The first step in creating the protein families is to match all the sequences in the database to each other. The most sensitive single-sequence matching procedure is the Smith–Waterman algorithm (Pearson, 1995; Brenner, 1996). Using the match scores produced by this algorithm, protein sequences can be placed in families by single-linkage clustering: all sequences connected by scores that are believed to indicate a significant relationship are clustered together. Both sequence matching and the formation of single-linkage clusters are simple to automate and can be carried out using the GEANFAMMER (GEnome ANalysis and protein FAMily MakER) package (see below). However, in bacterial genomes, and to a larger extent in eukaryotic genomes, the matches made by different parts of multi-domain proteins can bring unrelated proteins into one cluster upon single linkage (see Figure 1). To solve this problem, the DIVCLUS procedure was written to sort out such complex clusters.

The protein families resulting from an application of the GEANFAMMER package with the DIVCLUS program to a dataset correspond to a ‘duplication module’ (Riley and Labedan, 1997; Teichmann *et al.*, 1998). A biological duplication module is a whole gene or part of a gene that can be duplicated separately and which may also undergo recombination with other duplication modules. It may correspond to a protein structural domain, but often corresponds to several of these domains in series. However, without structural or other experimental evidence, ‘duplication module’ means a probable duplication module which can be detected by a sequence search method. Therefore, in the context of this

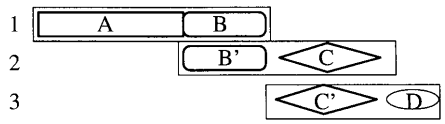

Fig. 1. Single-linkage clustering of multidomain proteins can lead to unrelated proteins being in the same cluster.

work, 'duplication module' means a duplication module as detected by a sequence search method.

System and methods

Language

GEANFAMMER is a suite of programs that finds the protein families of a set of sequences. It has three main parts: a sequence-matching procedure (which uses the FASTA or SSEARCH implementation of the Smith–Waterman algorithm); a single-linkage cluster algorithm; and DIVCLUS. The constituent parts can be used in one step or separately. All programs are written in perl5 in modular form to ensure flexibility. The collection of subroutines used in the GEANFAMMER package is available as a subroutine library and as a perl5 module, Geanfammer.pm. GEANFAMMER and DIVCLUS run on all LINUX, UNIX, WinNT, Windows95 and MAC OS systems that have a perl5 compiler/interpreter installed.

Use

The rate-determining steps in GEANFAMMER are the sequence search and the DIVCLUS division of clusters. The speed of the sequence search depends entirely on the algorithm used (e.g. SSEARCH or FASTA with ktup parameter 1 or 2) and the database size. The DIVCLUS program can handle any size of single-linkage cluster provided enough memory is available. The time it takes depends on the size of the single-linkage clusters: one containing 828 sequences took on the order of 2 h on a DEC alpha 500 MHz processor.

Algorithm

To generate protein families from a set of protein sequences, several steps are involved, as shown in Figure 2. For clarity, the stages will be described here in terms of the separate programs in the GEANFAMMER program suite, although it can perform all steps automatically. Initially, the protein sequences are compared to each other using an implementation of the Smith–Waterman algorithm such as SSEARCH or FASTA (Pearson and Lipman, 1991) with the '-m 10' option to produce output files in a machine-readable format. This comparison is carried out by the GEANFAMMER programs DO_SEQUENCE_SEARCH or using one's own scripts.

The SSEARCH/FASTA output format (sso) files are then converted to a form called msp ('matching sequence pair') format by the program SSO_TO_MSP. An example of msp format is given in Figure 3. Each line of the msp file contains the information on the relationship of the query sequence and its target sequences. It has the SSEARCH/FASTA score, expectation value (E-value) and percentage sequence identity of the two sequences, as well as the regions of the two sequences that match.

Next, the sequences have to be connected to each other by single-linkage clustering. Single linkage entails connection of all sequences to one another if they are linked by a significant score in any region of the sequence. This can result in large clusters, as one can connect very distantly related sequences to one another via any number of intermediate sequences. It also leads to 'complex' or incorrect clusters, since the sequence region is not taken into consideration. This is carried out by the program MSP_SINGLE_LINK, the only argument of which is the expectation value threshold. After this, the single-linkage clusters are converted to a form that DIVCLUS can use: an msp file for each single-linkage cluster. These secondary msp files contain the information described above for all pairwise relationships within a single-linkage cluster that fulfil the upper E-value limit parameter.

It is essential for the user to find an E-value for his target sequence database which generates acceptable families. If no E-value is specified, the default E-value is set by the program on a rough sliding scale depending on the target sequence database size. As a rough guide, an E-value of 0.081 produces a 1% error rate in a database of 180 000 sequences (Park *et al.*, 1997). Also, the E-value threshhold of 0.001 was used for a database of 12 000 sequences, and was derived empirically from the visual inspection of the sequence alignment of clusters to minimize the erroneous clustering of two unrelated sequences.

DIVCLUS then inspects each msp file in turn and checks the SSEARCH/FASTA score, E-value and overlap region between each pair of sequences (i.e. each line of the msp file) as to whether they fulfil specific parameters. The default for the SSEARCH/FASTA score is 75, which is very unrestrictive. The purpose is simply to remove insignificant small fragments that match with good E-values. The default E-value is the same as the default single-linkage E-value described above. Thirty amino acids is the default minimum threshold for overlap empirically chosen from our experience with bacterial protein sequences. It is comparable with the sequence length minimum set for a domain in the protein structure comparison program DALI (Holm and Sander, 1994).

The algorithm works in an iterative manner, with successive tests of pairs of sequences. This test is described in Figure 4a and functions as follows. The match regions of two pairs

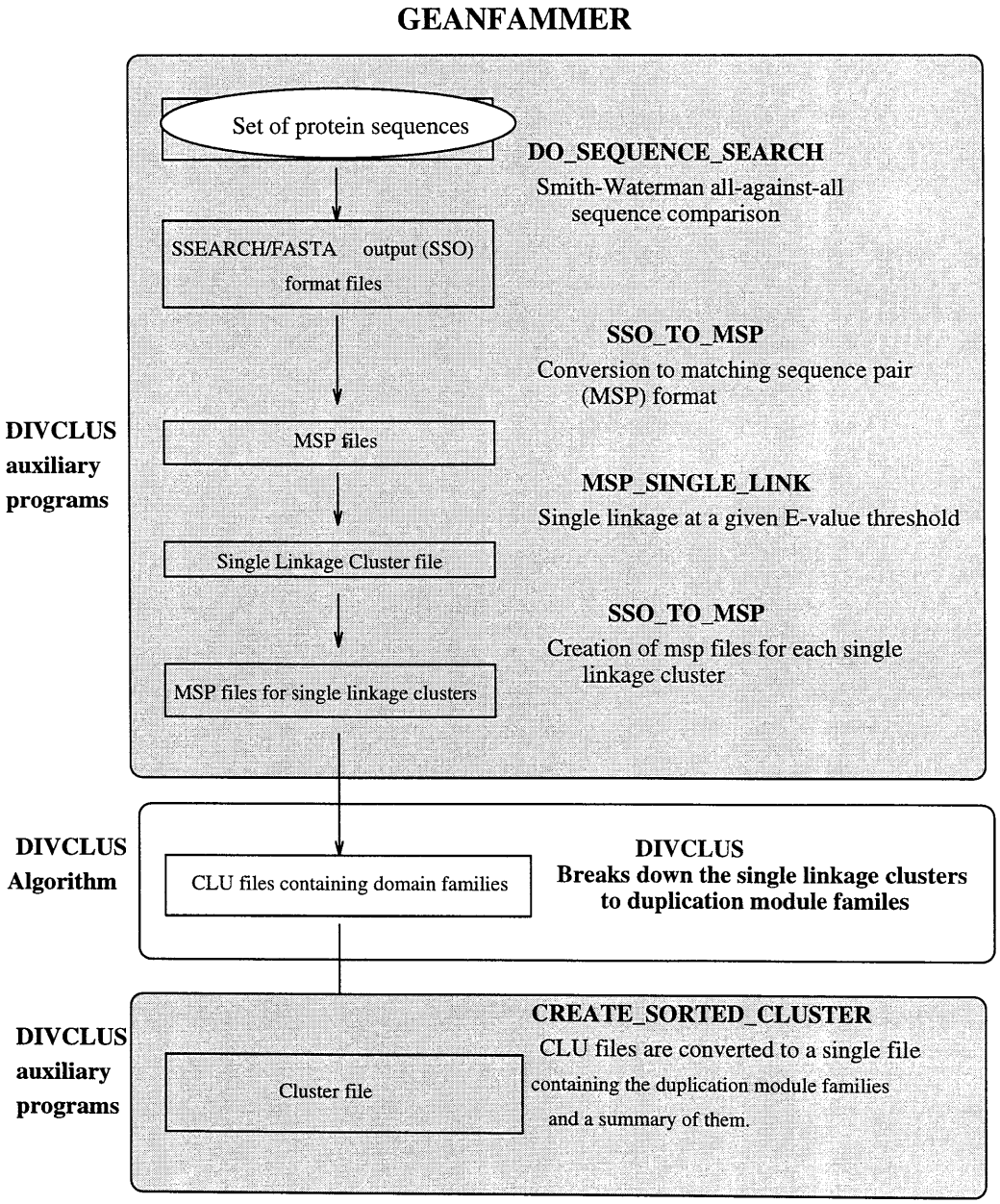

Fig. 2. A flow chart of the process of creating families of duplication modules.

of sequences are compared. Three sequences are accepted as belonging to the same family if the match regions overlap by at least 30 residues and the overlapping region represents at least 70% (optional parameter) of the overlap of the shorter of the two pairs. The denominator in calculating the percentage can be chosen as overlap2, overlap4, the shorter one (default) or the average of the two. The accuracy of the duplication module boundaries depends on both the overlap threshhold and the percentage overlap parameter. An overlap threshhold of 30 residues was considered to be the minimum size for a domain. To decrease the discrepancies in the duplication module boundaries further, the percentage overlap parameter can be increased, although this may prevent the inclusion in a family of distantly related members.

If the three sequences are accepted as having a common homologous region, the smaller (overlap1), larger (overlap3) or average (default) of the two regions of the common sequence (B) becomes the new overlap sequence, which is tested against the succeeding pairs of sequences. The average region chosen would be sequence B with the ends taken as

score	Evalue	Sequence Identity	region	enquiry seq	region	target seq
233	0.00001	0.568	1-300	Enq_seq	2-301	Target_seq1
244	0.00002	0.489	2-299	Enq_seq	3-301	Target_seq2
444	0.000001	0.478	3-355	Enq_seq	4-411	Target_seq3
111	0.00003	0.508	4-355	Enq_seq2	2-111	Target_seq4
222	0.00002	0.431	5-366	Enq_seq2	3-123	Target_seq5
112	0.00014	0.399	6-321	Enq_seq2	4-119	Target_seq6

Fig. 3. An example of an msp format file.

halfway between the ends of overlap1 and overlap3 at either end of the sequence. Choosing the smaller region will not produce wrong subclusters, but may fail to detect the true connection between pairs of sequences. The larger overlap region option could cause wrong clusters, but would be less likely to fail to cluster sequences when they are truly connected. The average region was chosen empirically as the default.

A case where three sequences are rejected as being the same duplication module is shown in Figure 4b. The overlap1 region is <70% of the overlap2 region, so the sequences are not merged. The checking process consists of a merging of accepted sequences (with their overlap regions attached) into an array. After this process is iterated over the whole msp file, another subroutine compares the contents of all the resulting arrays to check whether further merging should occur. This iteration continues until there is no change in the arrays, as shown in Figure 5.

From this description of the steps involved in the clustering process, it should be clear what is meant when the term 'duplication module' is assigned to the resulting sequence families. The sequence families are simply regions for which equivalents have been found in other proteins. Therefore, if a protein A has a long, potentially multi-domain region which is found in other proteins, this is a duplication module for which a sequence family will exist. If a short region inside the long region of A is found to match other proteins, the shorter region will also be a duplication module in a distinct sequence family. In this way, one sequence region can be present in more than one sequence family. This is biologically realistic, though, because a domain can be duplicated on its own, or in tandem with other domains, thereby being a member of two or more units of duplication, or duplication modules. Hence, both duplication of individual domains and sets of domains is detected, retaining both the information about domain boundaries and about duplication patterns. This is a feature which is distinct from previous domain-clustering programs.

It should be noted that internal repeats can be reported using this method if a circular relationship within the family exists. For instance, consider the situation where a protein A contains two repeats, r_1 and r_2, of the same domain. Protein B contains one copy, r_B, of the same domain and protein C contains one copy, r_C, of this domain. If r_B matches r_1, r_C matches r_2 and r_B matches r_C, then both repeats r_1 and r_2 will be part of the family separately. If only one repeat is matched, then the other repeat will not be in the family. If only both repeats are matched to similar tandem repeats in other proteins, then the region containing both copies will be part of a family consisting of the entire repeated region.

The files produced for each single-linkage cluster are called _cluster.clu files, and contain the sequences and their regions for all the subclusters in a single-linkage cluster. The format is shown in Figure 6. For each subcluster, the first column contains the cluster size, the second the unique identifier

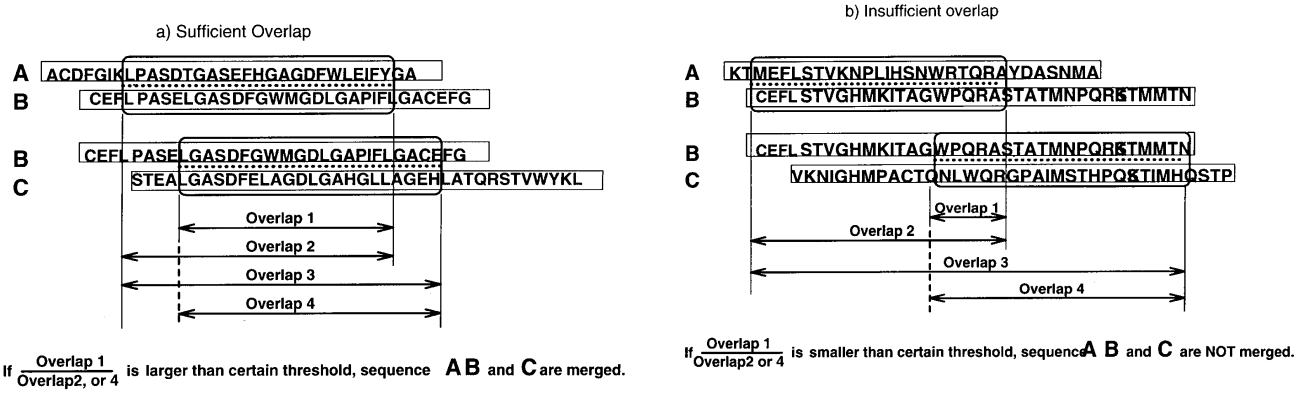

Fig. 4. The test for merging sequences according to their overlap size: (**a**) an example of three sequences that are related; (**b**) an example of three sequences that are not related.

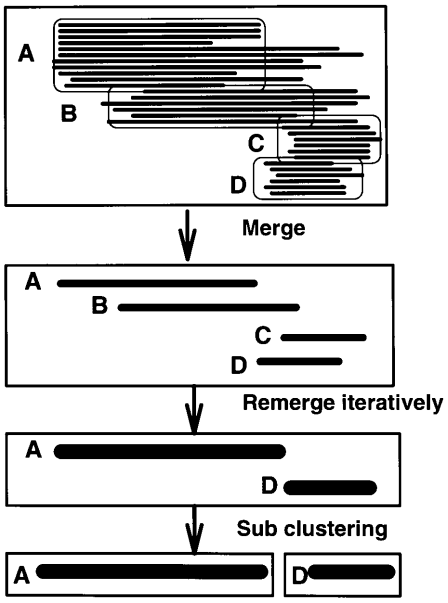

Fig. 5. The merging process is iterative.

number of the original single-linkage cluster, the third the sequence name, the fourth the sequence region, and the fifth the number of times the region matched other sequences in the cluster. The header line for each cluster also contains the following information: the cluster size (Cluster size), a unique cluster identifier (Cluster number), the expectation value threshhold used in DIVCLUS (E), the overlap factor (Factor), the resulting percentage overlap (P), the size of the original single-linkage cluster (Ori size), the size of the subcluster (Sub) and the unique identifier number of the original single-linkage cluster (From). The unique cluster identifier given as 'Cluster number' is formed from the following three numbers separated by 0s: the number of the single-linkage cluster, the subcluster number and the size of the subcluster.

The iterative procedure, though fast, is not perfect; for some large clusters with >200 sequences, it has been unsuccessful with default values for the parameters in the program. In these cases, it is necessary to use a higher percentage overlap, such as 80%. However, such a stringent factor is not necessary for smaller clusters to be broken down correctly. Therefore, there is an option for a dynamic percentage overlap to be implemented. Higher percentages are then used for larger clusters (e.g. 90% for a 500 sequence cluster). The percentage overlap can be varied by choosing a different overlap factor, *f*. This factor determines the percentage overlap as follows: percentage overlap = 100 – (100/*f*).

Once DIVCLUS has divided all the msp files and created cluster (clu) files, the program CREATE_SORTED_CLUSTER assembles the duplication module families into a single file. The families are placed in order of the number

```
Cluster size 4
Cluster number 1110104                  # E: 0.02   Factor: 7 P:85, Ori size: 12   Sub:4  From: 111

4   111   MG312      359-849      2
4   111   MG386      946-1395     2
4    111 MG200       10-436        1
4    111 MG317       70-511        3

Cluster size 4
Cluster number 1110204                  # E:0.02   Factor: 7   P: 85, Ori size: 12   Sub: 4 From: 111

4   111  MG318       99-279       3
4   111  MG386       1155-1333    1
4   111  MG217       2-176         1
4   111  MG317       259-401       1
```

Fig. 6. An example of a clu format file.

of members that they contain, with a summary with the distribution of families for each family size in the header of the file.

Implementation

The program suite GEANFAMMER, consisting of DO_SEQUENCE_SEARCH, MSP_SINGLE_LINK, SSO_TO_MSP, DIVCLUS and CREATE_SORTED_CLUSTER, was applied to the complexity-masked (Wootton, 1994) protein sequences of six completely sequenced bacterial genomes: *Escherichia coli* (EC) (Blattner *et al.*, 1997), *Haemophilus influenzae* (HI) (Fleischmann *et al.*, 1995), *Synechocystis* sp. (SS) (Kaneko *et al.*, 1996), *Mycoplasma genitalium* (MG) (Fraser *et al.*, 1995), *Mycoplasma pneumoniae* (MP) (Himmelreich *et al.*, 1996) and *Methanococcus jannaschii* (MJ) (Bult *et al.*, 1996). Application of DIVCLUS to the 12 013 proteins resulted in 13 076 duplication modules. The resulting protein families are between 2 and 235 sequences in size, and are available at http://www.mrc-lmb.cam.ac.uk/genomes/. The duplication modules are between 50 and 2600 residues in size. However, more than two-thirds of the duplication modules are between 50 and 250 residues in size. This size range is characteristic of the structural domains that are well known as units of duplication in evolution.

The use of breaking down of multi-domain proteins can be seen in the example of the bacterial two-component signal transduction system. A family of 80 sequences represents the response regulator receiver domain. Several smaller families contain homologues of the domains C-terminal to the receiver domain (which is usually found as the N-terminal domain of proteins). One of these is a family of 14 sequences (family 102096014 in our database) representing the luxR helix–turn–helix DNA binding domain. A further family of 14 sequences also represents a domain C-terminal to the receiver domain, which could also potentially be a DNA bind-

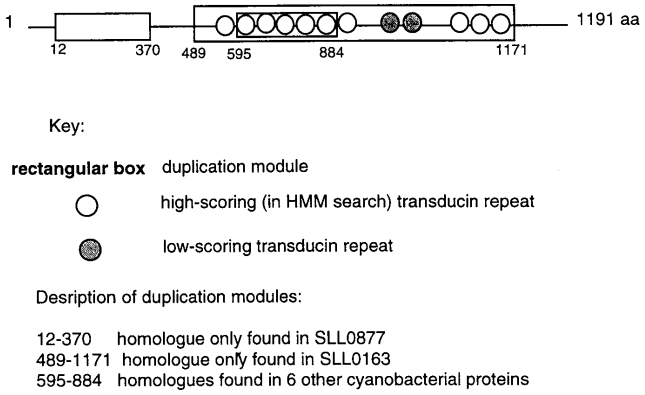

Fig. 7. The duplication modules in the cyanobacterial protein SLR0143 as identified using DIVCLUS.

ing domain, as the PROSITE (Bairoch *et al.*, 1995) leucine zipper motif matches two of its members.

The bacterial duplication module families can provide additional information about a given sequence from the perspective of the duplication pattern in that family. An example of this is the cyanobacterial protein SLR0143. The region between residues 595 and 884 is a member of a family of seven cyanobacterial proteins (family 16880207). This region contains five copies of the transducin 40-residue repeat (also called the WD40 repeat), as found by a different sequence search method, the readily available G-beta Hidden Markov Model of the Pfam database (Sonnhammer *et al.*, 1997). Five proteins containing this domain, which had only been noticed in eukaryotes before, were previously identified by Kaneko *et al.* (1996). Therefore, this region of five units may form an evolutionary 'duplication module'. In addition, the protein SLR0163 (which also contains WD40 repeats) matches SLR0143 in the region 489–1171. Hence, this larger unit of the entire C-terminal half of the protein may be a duplication module too. SLR0143 also has an N-terminal duplication module from residues 12 to 370 that is uncharacterized to our knowledge, which it shares with another cyanobacterial protein (SLL0877) that does not contain WD40 repeats. The whole duplication module layout of SLR0143 is shown in Figure 7.

Comparison of GEANFAMMER to other clustering methods

The two other clustering methods known to the authors are CLUSDOM (Koonin *et al.*, 1996) and DOMAINER (Sonnhammer and Kahn, 1994). Little information is available on the details of CLUSDOM, and the program has not been available at the location given in the reference since DIVCLUS started to be developed. However, the DOMAINER algorithm is described extensively in the above reference. They are both programs to deal with the problem of complex protein families produced from sequences that have been matched using the BLAST algorithm (Altschul *et al.*, 1990). This produces different results, because BLAST does not allow gaps, but the Smith–Waterman algorithm produces gapped alignments (Smith and Waterman, 1981).

The fact that BLAST is used as the search algorithm means that, in DOMAINER, internal repeats can be found by the circular relationship principle as in GEANFAMMER, but can also be detected directly from single pairwise comparisons. For the single-linkage phase, a default value of a minimum of 10 residues for the threshhold of overlap between two sequences is used in DOMAINER, as opposed to no overlap threshhold in GEANFAMMER. (The overlap threshhold is introduced in the later stage of dividing the clusters with DIVCLUS in GEANFAMMER.)

The splitting of single-linkage clusters is carried out using the principle of mutual exclusion in DOMAINER. This means that regions in which different members overlap are treated as separate domains. Hence, no one region can be a member of two families, in contrast to DIVCLUS, which allows regions to be present both by themselves and in combination with other regions when they have been duplicated together. The idea behind this is that GEANFAMMER produces families of duplication modules, preserving the information about duplication patterns as well as yielding the domain information. Finally, it should be noted that the purpose of DOMAINER is quite different to that of GEANFAMMER, in that it aims to produce multiple alignments and a consensus sequence for each family.

On a more general level, the results obtained using the BLAST algorithm and DOMAINER lead to smaller families with shorter sequences. In at least some cases, these families are not whole domains in the structural sense.

Discussion and conclusion

It is well known that protein domains are the unit of duplication and, in some cases, recombination. Therefore, classifying large sets of protein sequences, for instance whole genomes, into their constituent duplication modules provides not only a useful database for functional investigations, but also a framework for understanding protein evolution. For a large number of protein sequences, the process of clustering duplication modules into families cannot be done by hand.

DIVCLUS, together with the other programs described here, provides an automatic method of processing the output of gapped sequence comparisons to create families of duplication modules. Although the sequence comparison method used initially was the FASTA/SSEARCH implementation of the Smith–Waterman algorithm, GEANFAMMER can easily be extended to accept the output of

other gapped algorithms, such as gapped BLAST and PSI-BLAST (Altschul *et al.*, 1997).

Acknowledgements

We thank Alex Bateman for discussions and helpful suggestions. We are especially grateful to Cyrus Chothia for encouragement and suggestions. We also appreciate the comments from the three anonymous referees. S.A.T. has a pre-doctoral fellowship from the Boehringer Ingelheim Fonds.

References

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Altschul,S.F., Madden,T.L., Schäfer,A.A., Zhan,J., Zhan,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Bairoch,A., Bucher,P. and Hofmann,K. (1995) The PROSITE database: its status in 1995. *Nucleic Acids Res.*, **24**, 189–196.

Blattner,F.R. *et al*. (1997) The complete genome sequence of *Escherichia coli* K-12. *Science*, **277**, 1453–1462.

Brenner,S.E. (1996) Molecular propinquity. PhD Thesis, University of Cambridge.

Bult,C.J. *et al*. (1996) Complete genome sequence of the methanogenic archaeon, *Methanoccocus jannaschii*. *Science*, **273**, 1058–1073.

Fleischmann,R.D. *et al*. *(*1995) Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science*, **269**, 496–512.

Fraser,C.M. *et al*. (1995) The minimal gene complement of *Mycoplasma genitalium*. *Science*, **270**, 349–548.

Himmelreich,R., Hilbert,H., Plagens,H., Pirkl,E., Li,B.-C. and Herrmann,R. (1996) Complete sequence analysis of the genome of the bacterium *Mycoplasma pneumoniae*. *Nucleic Acids Res.*, **24**, 4420–4449.

Holm,L. and Sander,C. (1994) Parser for protein folding units. *Proteins*, **19**, 256–268.

Kaneko,T. *et al*. (1996) Sequence analysis of the genome of the unicellular cyanobacterium *Synechocystis sp.* strain PCC6803. II. Sequence determination of the entire genome and assignment of potential protein-coding regions. *DNA Res.*, **3**, 109–136.

Koonin,E.V., Tatusov,R.L. and Rudd,K.E. (1996) Protein sequence comparison at genome scale. *Methods Enzymol.*, **266**, 295–322.

Park,J., Teichmann,S.A., Hubbard,T. and Chothia,C. (1997) Intermediate sequences increase the detection of distant sequence homologies. *J. Mol. Biol.*, **273**, 349–354.

Pearson,W.R. (1991) Searching protein-sequence libraries—comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms.*Genomics*, **11**, 635–650.

Pearson,W.R. (1995) Comparison of the methods for searching protein-sequence databases. *Protein Sci.*, **4**, 1145–1160.

Riley,M. and Labedan,B. (1997) Protein evolution viewed through *Escherichia coli* protein sequences: introducing the notion of a structural segment of homology, the module. *J. Mol. Biol.*, **268**, 857–868.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Sonnhammer,E.L.L. and Kahn,D. (1994) Modular arrangement of proteins as inferred from analysis of homology. *Protein Sci.*, **3**, 482–492.

Sonnhammer,E.L.L., Eddy,S.R. and Durbin,R. (1997) Pfam: A comprehensive database of protein domain families based on seed alignments. *Proteins*, **28**, 405–420.

Teichmann,S.A., Park,J., Al-Lazikani,B., Hubbard,T. and Chothia,C. (1998) The major protein families in six bacteria. Submitted.

Wootton,J.C. (1994) Sequences with unusual amino-acid compositions. *Curr. Opin. Struct. Biol.*, **4**, 413–421.