Protein Engineering vol.15 no.8 pp.643–649, 2002

Sequence clustering strategies improve remote homology recognitions while reducing search times

Weizhong Li[1], Lukasz Jaroszewski[2] and Adam Godzik[3]

The Burnham Institute, La Jolla, CA 92037, USA

[1]Present address: Quorex Pharmaceuticals, 1890 Rutherford Road, Suite 200, Carlsbad, CA 92008, USA

[2]Present address: Bioinformatics Core of the Joint Center for Structural Genomis, San Diego Computer Center, 9500 Gilman Drive, La Jolla, CA 92093, USA

[3]To whom correspondence should be addressed at: Program in Bioinformatics and Biological Complexity, The Burnham Institute, 10901 N. Torrey Pines Road, La Jolla CA 92037, USA
E-mail: adam@burnham.org

Sequence databases are rapidly growing, thereby increasing the coverage of protein sequence space, but this coverage is uneven because most sequencing efforts have concentrated on a small number of organisms. The resulting granularity of sequence space creates many problems for profile-based sequence comparison programs. In this paper, we suggest several strategies that address these problems, and at the same time speed up the searches for homologous proteins and improve the ability of profile methods to recognize distant homologies. One of our strategies combines database clustering, which removes highly redundant sequence, and a two-step PSI-BLAST (PDB-BLAST), which separates sequence spaces of profile composition and space of homology searching. The combination of these strategies improves distant homology recognitions by more than 100%, while using only 10% of the CPU time of the standard PSI-BLAST search. Another method, intermediate profile searches, allows for the exploration of additional search directions that are normally dominated by large protein sub-families within very diverse families. All methods are evaluated with a large fold-recognition benchmark.

Keywords: fold recognition/intermediate profile search/sequence clustering

Introduction

The success of recent genome sequencing projects has resulted in a dramatic increase in the number of known protein sequences. With this increasing density of sequence space, it has become easier to develop generalized descriptions of protein families, using methods such as profiles, position specific scoring matrices (PSSMs) or hidden Markov models (HMMs). Homology detection and structure prediction methods based on exploring information from multiple alignments of homologous families have gained a lot from the databases' growth.

On the other hand, the growth of the databases has slowed down the searches because of the sheer number of sequences that have to be considered; however a more serious problem is caused by the uneven growth of the databases. In the widely used non-redundant protein database (NR) maintained at the National Center for Biotechnology Information, a typical protein family may contain numerous identical or almost identical entries from some species but only a few homologs from other sources. The almost identical sequences are important in many research problems, such as studies of single nucleotide polymorphisms (SNPs) or splice variants, but such biased data creates two serious problems when using PSI-BLAST (Altschul *et al*., 1997), the most popular profile-based homology recognition tool and other profile based tools.

PSI-BLAST is performed in a iterative way. First, an initial sequence–sequence comparison is performed and the hits are ranked according to their alignment scores. Secondly, a profile in the form of a PSSM is calculated from a certain number of sequences from the top of the hit list. Thirdly, with this PSSM, the next search is a profile–sequence comparison, and in most cases some new sequences can be found. Finally, a search loop from steps 2 to 3 is repeated until no more new hits are found or the maximum number of iterations is reached.

The first problem created by the biased data affects large sequence families. Because the number of sequences to be included in the profile calculation (step 2) is limited, proteins that are highly homologous to the query can saturate the profile. The profile does not provide much more information than the query sequence itself, since more diverse homologs are ranked too low in the hit list to come above the threshold number of sequences for the profile calculation. One solution is to increase the number of explicitly considered alignments, but this dramatically slows the algorithm and may be still insufficient for families consisting of tens of thousands of homologs (Park *et al*., 2000). Another aspect of this problem is that if we are interested only in certain homologs with specific features, such as coming from a certain genome or having a known 3D structure, these may not be included or may be difficult to find in the PSI-BLAST output because of output size restrictions.

One of our strategies to address this problem is a two-step PSI-BLAST search approach, where a profile is first built from a search against a large database like the NR, and then this profile is used to search a small database like the PDB. This method called PDB-BLAST was introduced as a reference method for our profile–profile alignment method FFAS (Rychlewski *et al*., 2000) and was shown to perform respectably well in the CASP4 fold-prediction competition, where it was classified in the middle of all competing algorithms and groups (http://predictioncenter.llnl.gov/). For fold-recognition benchmarking purposes only, the profile can be built from a database of PDB sequences and all their homologs in the NR instead of the entire NR. We refer to the expanded database as PDBX. A similar database, PDB-ISL, has been used as an intermediate sequence library to speed up PSI-BLAST searches (Teichmann *et al*., 2000). But our aim here was to narrow the searching space, because only the sequences in PDBX will potentially contribute to the profile being used.

Another approach to this problem is to cluster similar

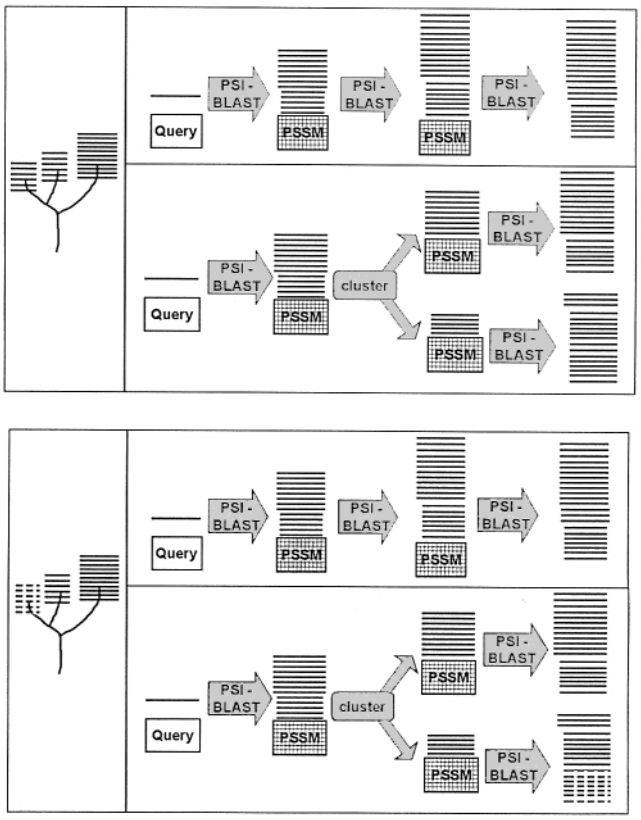

Fig. 1. PSI-BLAST search versus IPS. A large protein family consisting of three distantly related sub-families: Dashed (D), short (S) and long (L), and S is between D and L (left panel). When a sequence from L is used as a query, it is difficult to find D with the profile dominated by L and its closer neighbor S (upper panel). However, if sequences are grouped into sub-families S and L, the most remote sub-family D is likely to be found by a profile made from S.

sequences in the database and to only search against the representative sequence from each cluster. The size of the representative set is decided by a threshold of sequence similarity between the proteins in one cluster. After clustering, the representative set will not have sequences that are more similar than the threshold. When we cluster the NR database at 80%, we refer to the representative set as NR80 and will use this convention to name other clustered databases. In a naive approach, such clustering would require all-against-all sequence comparisons, which would be essentially impossible for large data sets like the NR, so techniques have been developed to speed up clustering and make it possible. For example, the nrdb90 program (Holm and Sander, 1998), implemented with short peptide filters and a lookup table, can cluster the NR to the 90% threshold level. Series of representative sequence databases (RSDB) (Park *et al*., 2000) were generated from the NR at different identities from 20 to 99% by a comprehensive pairwise comparison database made by a large number of BLAST searches. We have developed algorithms that were implemented in the CD-HI (Li *et al*., 2001) and CD-HIT (Li *et al*., 2002) programs, which are able to cluster NR at 65% identity in hours and at 50% identity in 5 days on a midsize Linux workstation. In the RSDB study, it was found that clustering saved a remarkable amount of search time and that even the database clustered at 50% identity (RSDB50) did not compromise homology detection in comparison to the full NR.

The second problem with the profile approach is encountered if a large protein family is composed of several sub-families of uneven sizes (Figure 1). The profile is thus likely to be dominated by proteins from the largest sub-family and its close relatives, and the profile becomes trapped by the dominating sequences in successive iterations, making it difficult to find more distant sub-families. For example, if the search was initiated by a query from a small sub-family, the PSI-BLAST iterations converge on a group that often excludes the initial sequence. This phenomenon is referred to as the profile trap.

We have tried several methods to address this second problem. The profile can be rebalanced using a two-dimensional weighting system, FFAS (Rychlewski *et al*., 2000), resulting in more sensitive recognitions. Another solution is the intermediate sequence search (ISS) (Park *et al*., 1997, 1998; Karplus *et al*., 1998; Salamov *et al*., 1999) where a cascade of BLAST or PSI-BLAST searches are performed using the intermediate sequences from each sub-family as the new search queries. The automated protocol of such a cascade search is implemented in our package: Saturated BLAST (Li *et al*., 2000). We introduce here a new method to handle the profile trap using multiple profiles from each sub-family (see Figure 1), and refer to this method as the intermediate profile search (IPS). IPS is more powerful in distant homology detection than ISS because it uses an intermediate profile instead of a single intermediate sequence to establish remote homology.

In this study, our focus was on how various search strategies improve the recognition of distant homologs, as evaluated using a large fold-recognition benchmark. We introduce and test several methods to solve the profile problems discussed in the Introduction. These methods include database clustering, IPS, PDB-BLAST, and can be used individually or in different combinations.

Our search strategies are more than just technical tricks that improve the performance of PSI-BLAST. Apart from offering significant practical advantages in applying fold prediction and distant homology recognition to large groups of sequences, they offer avenues to include additional information such as functional similarity into sequence searches. Our strategies also allow us to learn more about the underlying structure of sequence space: first by exploring its granularity, and ultimately understanding the constraints that have shaped it. All other profile-based algorithms face the same problems and the solutions presented here would also apply to them.

Materials and methods

Fold-recognition benchmarks

A fold-recognition benchmark contains distant homology pairs that are difficult to recognize with a simple sequence alignment method. The sensitivity of different algorithms is evaluated by listing the number of correctly recognized pairs in the benchmark as a function of a number of false positive hits (sensitivity plot) for a given level of statistical significance. Such plots have become standard in describing fold-recognition algorithms.

The benchmark in this study was prepared from the SCOP Database (Murzin *et al*., 1995) Release 1.53 from http://scop.mrc-lmb.cam.ac.uk/scop/. The ASTRAL compendium (Brenner *et al*., 2000) at http://astral.stanford.edu/ provides a series of SCOP domain sequence databases clustered at different identities. Our benchmark was based on SCOP domain sequences comprising 2417 sequences with lengths

longer than 40 amino acids and with <30% identity with each other (SCOPD30).

We performed all-against-all BLAST searches and collected pairs of domains that could not be recognized with a BLAST expect value better than 0.1 despite having the same SCOP fold type. This benchmark contains 11 853 pairs with the same fold type but very low sequence similarity. It is available at http://bioinformatics.burnham-inst.org/liwz/research/benchmark

Database clustering

The NR protein database was downloaded from NCBI on September 20, 2000 and contains 563 276 sequences. It was clustered with CD-HIT at 90, 80, 65 and 50% sequence identities, and four databases NR90, NR80, NR65 and NR50 containing only the representative sequences were obtained. We did not include the EBI RSDB database, because it was derived from a much earlier NR, and it was too time consuming to prepare it locally using the EBI method.

Apart from NR database and its derivatives, we also created another database called PDBX, which contains all the sequences or sequence fragments that are possibly homologous to known PDB sequences. The PDBX was built by the following steps: (i) all of the entries in NR were marked as blank sequences; (ii) all of the sequences in SCOPD30 were used as queries to search the NR by PSI-BLAST; (iii) all of the sequence segments identified in the searches in (ii) were marked as foldable; (iv) in the same sequence, overlapping foldable fragments were merged; (v) each blank fragment shorter than 40 amino acid between two foldable regions was also marked as foldable; (vi) all of the foldable segments were accumulated. PDBX was then clustered at 80, 65 and 50% to yield PDBX80, PDBX65 and PDBX50.

Fold recognition

We searched these prepared databases using three fold-recognition tools: PSI-BLAST, PDB-BLAST and IPS. The PSI-BLAST parameters were 500 alignments and descriptions, three iterations, 0.001 expect value of sequences to be included in the profile.

PDB-BLAST is a two-step PSI-BLAST search. The parameters for the first search were identical to the PSI-BLAST parameters above except that the binary format sequence profile was saved with the '-R' option. The second search was without iteration and was against the SCOPD30 database.

The PDB-BLAST procedure was introduced in one of our previous papers as a reference method (Rychlewski *et al*., 2000) and gained a significant popularity since then. As we reported before, PDB-BLAST was more sensitive than PSI-BLAST in fold-recognition. The PDB-BLAST fold-recognition strategy is implemented on our fold-recognition server at http://bioinformatics.burnham.org/pdb_blast

The IPS was a PSI-BLAST search using the same parameters as above, and the sequences with expect values lower than 0.001 were clustered into sub-groups by an identity threshold. We then built a multiple alignment for each sub-group and ran a single iteration of a PSI-BLAST using this alignment and the '-B' option. The new hit sequences were added to the output list and a search loop was repeated for another round. The clustering threshold was optimized 20% by tests with the above benchmark and if a sequence was found again during an iteration, it replaced the old one if its expect value was better and if the alignment was longer, otherwise the new hit was rejected.

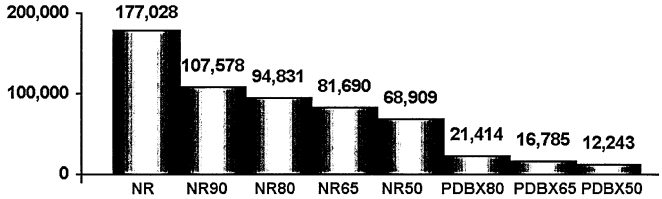

Fig. 2. Sizes of clustered databases as the number of thousands of amino acids.

There is a PDB-BLAST-like IPS-variant of IPS where the object database is set to SCOPD30 in the last rounds of the searches. To test the IPS, please contact the author at liwz@burnham-inst.org for the Perl scripts.

Results and discussion

Clustered NR and PDBX databases

Figure 2 compares the sizes of the original NR and the sizes of the clustered databases: NR90, NR80, NR65, NR50, PDBX80, PDBX65, PDBX50. There is a large decrease going from NR to NR90, with more gradual decreases in sizes down to NR50. There is another large decrease in size when going from the NR to the PDBX databases. PDBX50 is the smallest and is 7% of NR's size. In order to demonstrate the scope of redundancy in the sequence databases, we present a pie chart of the distribution of redundant sequences in NR in Figure 3. At 90% identity, 42% of the sequences in NR do not have redundant neighbors. When the threshold is lowered to 65% identity, only 29% of the sequences of NR can form single-member clusters, but the percentage of highly redundant sequences, as defined as having more than 50 members in a cluster, is raised to 21%. There are several well populated clusters of over 1000 members. For example, the largest cluster represented by a sequence (NCBI-gi identifier 3002851) has 12 990 sequences at a threshold of 65% identity.

In our fold-prediction experiment, the SCOPD30 database was appended to each of the NR and PDBX series of databases when evaluating the benchmark. This is a neglible difference in the databases size because the SCOPD30 is tiny compared to the NR or PDBX databases.

Fold recognition of PSI-BLAST and PDB-BLAST

The clustered NR and PDBX databases were used in PSI-BLAST and PDB-BLAST searches to show how the database clustering strategy improved fold-recognition. The sensitivities of PSI-BLAST and PDB-BLAST with these databases are plotted in Figure 4.

For PSI-BLAST, the fold-recognition accuracy was strongly affected by the database used to generate the profile, which is similar to the results of the EBI group (Park *et al*., 2000). The full-size NR is clearly the worst performer and the results improve with a decreasing clustering threshold. For clarity, the NR90, NR65 and PDBX65 are not plotted in Figure 4. Results for the NR90 are between the results for NR and NR80; NR65 is between NR80 and NR50, and PDBX65 is between PDBX80 and PDBX50. The same trend was found in PDB-BLAST searches, but the differences between each database were much smaller than in PSI-BLAST.

The advantages of clustered databases are even clearer in terms of search time. The search time in PSI-BLAST, which includes scanning the database and calculating the profile, is not linear in relation to the size. The size of NR50 is 39% of

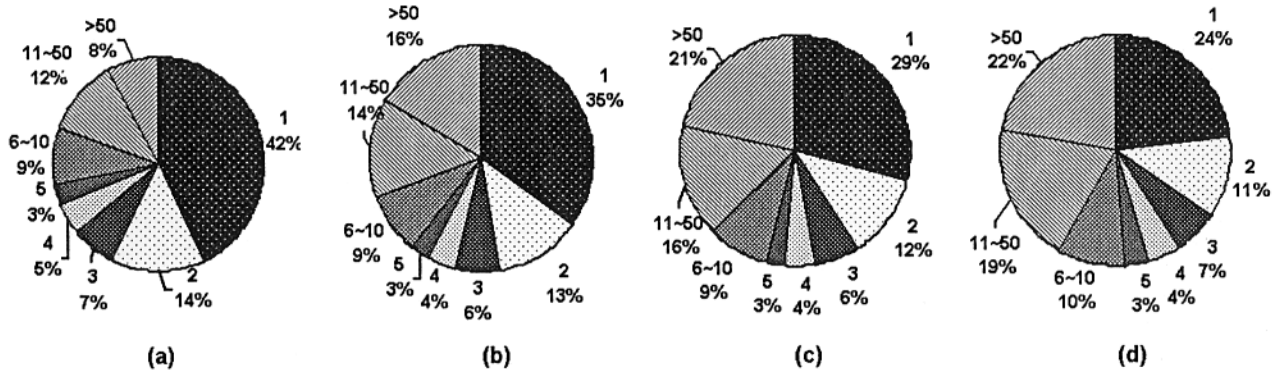

Fig. 3. Redundancy of NR database in terms of the distribution of sequence clusters at different levels of sequence similarity. The thresholds from (**a**) to (**d**) are 90, 80, 65 and 50%, respectively. The two numbers beside each area represent the sizes of clusters and the percentage of such sequences in NR. For example, as seen in (b), at an 80% threshold level, proteins in five-member clusters form 3% of the NR database.

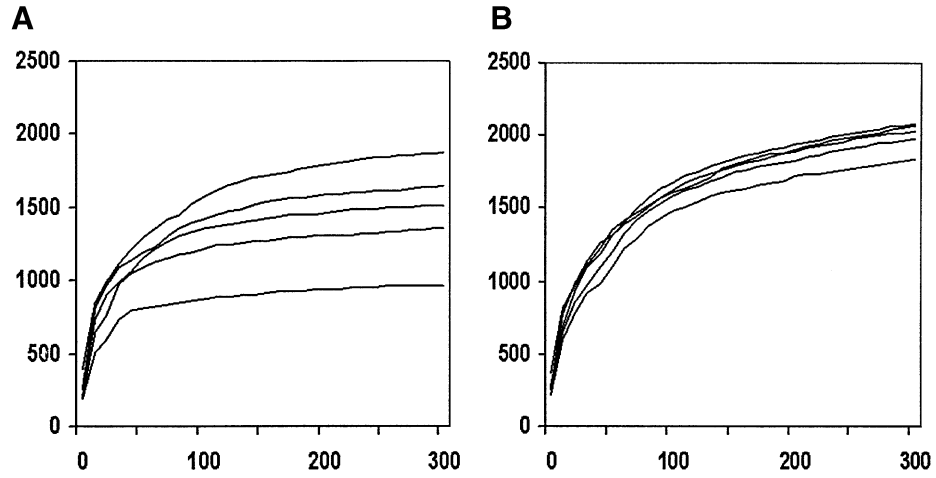

Fig. 4. Sensitivity of fold-recognition in PSI-BLAST (**A**) and PDB-BLAST (**B**) with different databases. The *x*-axis is the number of false positive hits, the *y*-axis is the number of correct hits. (**A**) Lines from top to bottom at $x = 300$ are PDBX50, NR50, PDBX80, NR80 and NR. (**B**) Lines from top to bottom at $x = 300$ are PDBX80, PDBX50, NR50, NR80 and NR.

NR, but as a result the PSI-BLAST searches are three times faster in our benchmark. The searches with PDBX50 were only 10% of the search time for NR.

Figure 4 shows that PDB-BLAST is more sensitive than PSI-BLAST on identical databases. Since the only difference between PDB-BLAST and PSI-BLAST is the second database search against SCOPD30, the improvement must come from this step. There are two reasons for this phenomenon: SCOPD30 is 350 times smaller than the NR so the PDB-BLAST output covered more hits; and the second search is another iteration. Another possible explanation is that the SCOPD30 search identifies the least dissimilar protein, which would not be recognized from the NR because of high-scoring false positive alignments. It is interesting to note that this improvement cost almost no extra CPU time because of the tiny size of SCOPD30.

The PDBX databases performed better in fold-recognition than their corresponding NR databases, but the use of PDBX databases is questionable in real fold-recognitions where the query sequences may have an inadequate number of homologs in the PDBX database.

Profile quality

The main result of the paper, that recognition accuracy is better when using a clustered database, is puzzling: one would not have expected to improve recognition by discarding information. So we examined the sequence profiles from the different databases for all the queries, especially the profiles from NR65 and NR, which we later called Prof-NR65 and Prof-NR. For each query, we extracted the sequences used to calculate Prof-NR65 and Prof-NR and discarded the redundant sequences from NR that were eliminated from NR65 during clustering. We then generated multiple alignments using ClustalX (Jeanmougin *et al*., 1998) and computed phylogenic trees. Typical trees are shown in Figure 5. In all cases, sequences in Prof-NR were also in Prof-NR65, but the opposite was not always true. In viewing these trees, it was easy to state the extent and the percentage to which NR65 brought new sequences; and thus there was more information in the profiles despite being derived from a smaller database.

We obtained four types of phylogenic tree. The first type is in Figure 5a and Prof-NR65 and Prof-NR contain the same sequences. This is the most common type, accounting for 50% of all computed trees. Figure 5b is the second type, where most sequences (>80%, but not all) are present in both Prof-NR65 and Prof-NR; the occurrence of this type is approximately 33%. The third type is in Figure 5c, where Prof-NR65 contained more new sequences (from 20 to 80%); this type is 14% of the trees. Figure 5d is the fourth type where the tree is dominated by Prof-NR65 at over 80%; these types of trees are rare and only 3% of the trees.

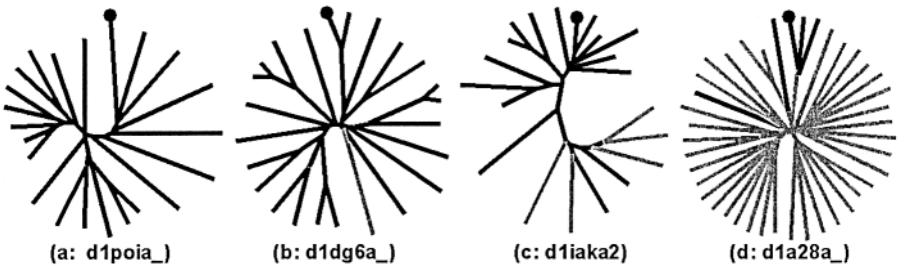

Fig. 5. The sequence diversity of profiles in the PSI-BLAST searches of NR65 and NR. The phylogenic graph shows the sequences included in the profile. In order to reduce the number of branches, homologous sequences (from 30 to 60% identities) were merged into a single branch. Branches in black represent sequences found in both NR65 and NR90, and the gray branches are the sequences found only in NR65. The capped branches represented the query sequences. The queries in (**a**–**d**) are the SCOP domains d1poia_, d1dg6a_, d1iaka2 and d1a28a_. The phylogenetic tree was generated by ClustalX (Jeanmougin *et al.*, 1998) and drawn by the Treeview program (http://taxonomy.zoology.gla.ac.uk/rod/rod.html).

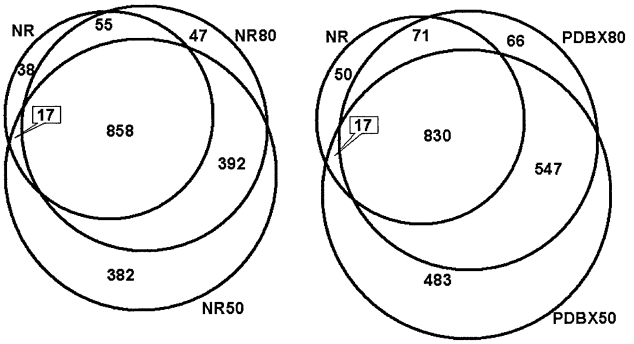

Fig. 6. Overlaps of correct hits of PSI-BLAST against different databases.

As shown in Figure 5, fold prediction using clustered databases is primarily improved in third and fourth types of trees: 17% of all queries.

The overall performances of clustered databases were better than the original NR, but a clustered database may also decrease the quality of the profile in some individual cases because some information is inevitably lost when near-redundancy is eliminated. Figure 6 shows the overlaps of correct hits found by PSI-BLAST searches against NR and the clustered databases at a level of 300 false-positive hits. Profiles derived from every database can distinguish some benchmark pairs that are missed by others. Therefore, the optimal cluster threshold is not fixed for an individual protein family. This phenomenon suggests an intelligent way of preparing a clustered database: use a variable threshold for different proteins. A method based on this idea is now being evaluated.

Intermediate profile search

The IPS was tested with the NR80 and PDBX50 databases and compared with PDB-BLAST and PSI-BLAST (see Figure 7). As introduced in Materials and methods, the IPS was implemented in two ways: IPS and a PDB-BLAST-like IPS. For both the NR80 and PDBX50, the PDB-BLAST-like IPS was the better method. Both IPS and PDB-BLAST are better than PSI-BLAST, and combining them into a PDB-BLAST-like IPS is even better.

With NR80, the PDB-BLAST-like IPS increased sensitivity by 5% compared with PDB-BLAST at a level of 300 false-positives, while IPS gained 38% compared to PSI-BLAST. With PDBX50, these two numbers are 12 and 20%, respectively.

But IPS is more time consuming than other methods, because additional PSI-BLAST searches are run after the first search. Depending on the diversity of a protein family and the clustering threshold of intermediate profile calculations, the number of additional searches can range from zero to several dozen. With additional searches, it is likely that IPS finds more potential homologs in the search, but it can also introduce more false predictions. We used a fixed threshold of 20% to make the intermediate profile for the whole benchmark calculation, but this threshold may not be suitable for every individual case. For smaller number of additional searches and better sensitivity, an optimized threshold can be obtained, so practically, IPS can be more powerful in finding remote homologies.

We selected one example to illustrate the power of IPS in finding remote homologs. The query sequence is SCOP domain d1iray3 with SCOP token '2.1.1.4.10'. The structure has the immunoglobulin-like beta-sandwich fold type, which contains many very diverse sequences that are hard to detect. The PSI-BLAST search against PDBX50 found seven correct hits; the IPS search found 30 correct hits (Figure 8). In addition to detecting more homologs, IPS usually provides better alignments. From the above example, we compared the alignments between SCOP domains d1iray3 and d1tlk as found by PSI-BLAST and IPS, and the IPS alignment is longer than the PSI-BLAST alignment. When evaluated with the structural comparison CE method (Shindyalov and Bourne, 1998), IPS provides 25 more correctly aligned positions than PSI-BLAST (Figure 9). Based on the IPS alignment and PSI-BLAST alignment, two models of the domain d1ltk were built using d1iray as the template and with the comparative modeling algorithm, Modeler (Sali and Blundell, 1993). The RMSD between the IPS model and the crystal structure is 1.9 Å (the result from the CE server), but the RMSD for PSI-BLAST model is 4.4 Å.

Although IPS is expensive in time as compared to PDB-BLAST or PSI-BLAST with the same database, the average time of IPS against the PDBX database is still less than PDB-BLAST or PSI-BLAST on the NR.

Conclusion

We improved the quality of sequence profiles using two clustering strategies: database clustering and IPS. The first clustering procedure is performed before searching the database searching and rebalances the sequence profile by removing highly homologous redundant proteins that compete with intermediate or remote homologs in the profile calculation. The second clustering procedure explored additional search directions by using profiles made with separate sub-families.

Our search strategies, along with the PDB-BLAST strategy, double the sensitivity of recognizing a remote homology while reducing the search time 10-fold as compared to standard PSI-

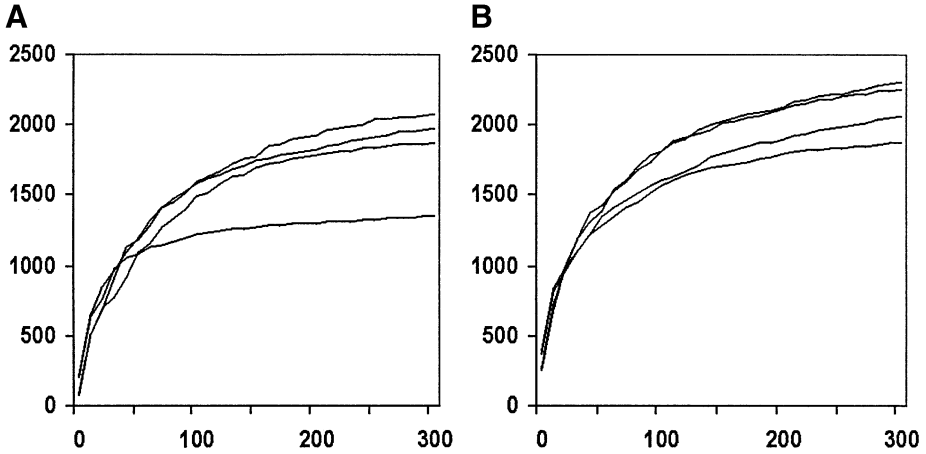

Fig. 7. Sensitivity of fold-recognition of PDB-BLAST-like IPS, IPS, PDB-BLAST and PSI-BLAST on the NR80 (**A**) and PDBX50 (**B**) databases. The *x*-axis is the number of false positive hits, the *y*-axis is the number of correct hits. (A) Lines from top to bottom at $x = 300$ are PDB-BLAST-like IPS, PDB-BLAST, IPS and PSI-BLAST. (B) Lines from top to bottom at $x = 300$ are PDB-BLAST-like IPS, IPS, PDB-BLAST and PSI-BLAST.

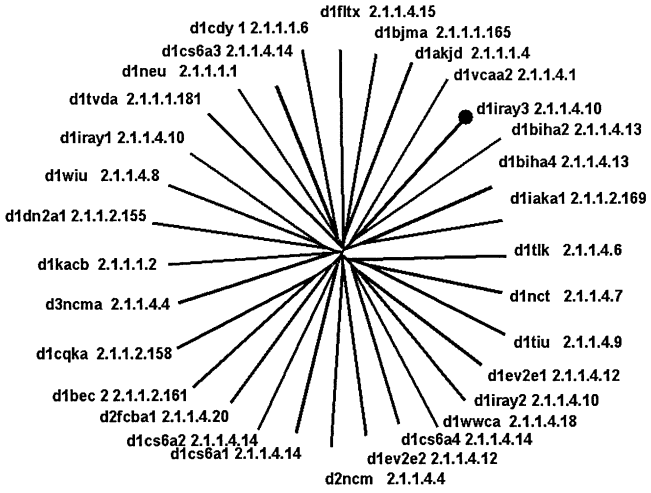

Fig. 8. Fold recognition of IPS with SCOP query d1iray3 against the PDBX50 database. The sequences found by IPS before the first false positive hit. The capped branches represented the query sequences (d1iray3). The branches in black are the sequences found by PSI-BLAST and IPS, and the branches in gray are the sequences found only by IPS. The phylogenetic tree was generated by the Phylip program (http://evolution.genetics.washington.edu/phylip.html) and was based on the multiple alignment provided by IPS. The tree was drawn by the Treeview program (http://taxonomy.zoology.gla.ac.uk/rod/rod.html).

BLAST. One shortcoming of using clustered databases is that the recognition sensitivity can get worse for some protein families. We are currently working on using variable cluster thresholds that can separately balance redundancy and information for every protein family.

The IPS method provides additional possibilities to find remote homology by extending the search in directions other than that of the dominant sub-family. Despite the relatively slow speed as compared to PDB-BLAST and PSI-BLAST, this method is a good choice when PSI-BLAST and PDB-BLAST fail. If the purpose of the search is to identify a template for or to develop a 3D model or to analyze the conservation of specific residues, then IPS is worth a try because it often provides better alignments.

The Materials and methods introduced in this paper are available on-line. The web addresses are http://bioinformatics.burnham-inst.org/liwz/research/benchmark for the fold-recognition benchmark, http://bioinformatics.burnham-inst.org/cd-hi

A

```
IPS
D1IRAY3 ---VIVSPANETMEVDLGSQIQLICNVTGQLSDIAYWKWNGSVIDEDDPVLGE
D1TLK__ ---PYFTKTILDMDVVEGSAARFDCKVEGYPDPEVMWFKDDNPVKESRHFQID
                ^^                                  ^^^^^^^

D1IRAY3 DYYSVENPANKRRSTLITVLNISEIESRFYKHPFTCFAKNTHGIDAAYIQLIY
D1TLK__ YD-----------EEGNCSLTISEVCGDDDA-KYTCKAVNSLGEATCTAELLV
          ^^^^^^^^  ^^^^^          ^                    ^^

PSI-BLAST
D1IRAY3 ------------MEVDLGSQIQLICNVTGQLSDIAYWKWNGSVIDEDDPVLGE
D1TLK__ ------------MDVVEGSAARFDCKVEGYPDPEVMWFK-------DDNPVKE

D1IRAY3 DYYSVENPANKRRSTLITVLNISEIESRFYKHPFTCFAKNTHGIDAAYIQL--
D1TLK__ SRHFQIDYDEEGNCSL----TISEVCGDD-DAKYTCKAVNSLGEATCTAEL--

CE
D1IRAY3 TRPVIVSPANETMEVDLGSQIQLICNVTGQLSDIAYWKWNGSVIDEDDPVLGE
D1TLK__ VKPYFTKTI-LDMDVVEGSAARFDCKVEGYPDPEVMWFKDDNPVKE-SRHFQI

D1IRAY3 DYYSVENPANKRRSTLITVLNISEIESRFYKHPFTCFAKNTHGIDAAYIQLI-
D1TLK__ DYDEE----------GNCSLTISEVCGDDD-AKYTCKAVNSLGEATCTAELL-
```

B

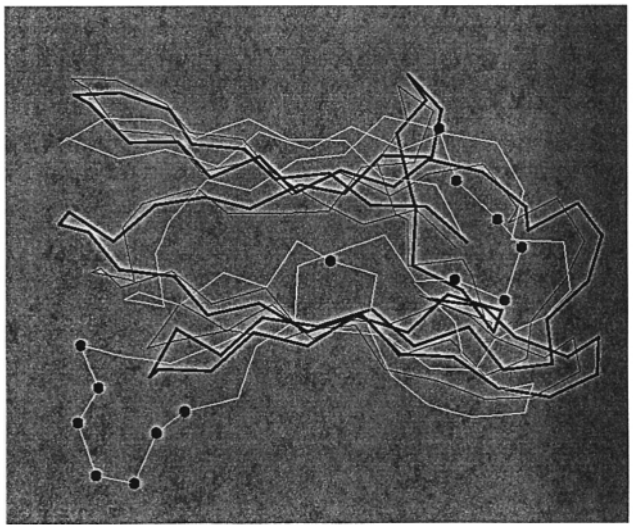

Fig. 9. Comparison of the alignments and the 3D-model made by IPS, PSI-BLAST and CE. (**A**) The alignments between SCOP domains d1iray3 and d1tlk as made by IPS, PSI-BLAST and CE, the structure comparison server at http://cl.sdsc.edu/ce.html. The positions in the IPS alignment marked with '^' are in agreement with the CE alignment but not with PSI-BLAST alignment. (**B**) Based on the alignments in (A), the models of d1tlk were built by the Modeler program (Sali and Blundell, 1993). This is the overlap of the native crystal structure (thick line), the IPS model (thin black line) and the PSI-BLAST model (white line). Positions with large deviations in the PSI-BLAST model are marked at CA atoms.

for the fast database clustering programs CD-HI and CD-HIT and http://bioinformatics.burnham-inst.org/pdb_blast for PDB-BLAST search. The complete PDB_BLAST WEB server is available upon request. The script for IPS can be obtained by contacting the author at liwz@burnham-inst.org

Acknowledgements

We want to thank Jason Hoffman (http://www.sciedit.com) for help with editing the manuscript. The research described in this manuscript was supported by NIH grant GM60049.

References

Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) *Nucleic Acids Res*., **25**, 3389–3402.
Brenner,S.E., Koehl,P. and Levitt,M. (2000) *Nucleic Acids Res*., **28**, 254–256.
Holm,L. and Sander,C. (1998) *Bioinformatics*, **14**, 423–429.
Jeanmougin,F., Thompson,J.D., Gouy,M., Higgins,D.G. and Gibson,T.J. (1998) *Trends Biochem. Sci*., **23**, 403–405.
Karplus,K., Barrett,C. and Hughey,R. (1998) *Bioinformatics*, **14**, 846–856.
Li,W., Pio,F., Pawlowski,K. and Godzik,A. (2000) *Bioinformatics*, **16**, 1105–1110.
Li,W., Jaroszewski,L. and Godzik,A. (2001) *Bioinformatics*, **17**, 282–283.
Li,W., Jaroszewski,L. and Godzik,A. (2002) *Bioinformatics*, **18**, 77–82.
Murzin,A.G., Brenner,S.E., Hubbard,T. and Chothia,C. (1995) *J. Mol. Biol.*, **247**, 536–540.
Park,J., Teichmann,S.A., Hubbard,T. and Chothia,C. (1997) *J. Mol. Biol.*, **273**, 349–354.
Park,J., Karplus,K., Barrett,C., Hughey,R., Haussler,D., Hubbard,T. and Chothia,C. (1998) *J. Mol. Biol.*, **284**, 1201–1210.
Park,J., Holm,L., Heger,A. and Chothia,C. (2000) *Bioinformatics*, **16**, 458–464.
Rychlewski,L., Jaroszewski,L., Li,W. and Godzik,A. (2000) *Protein Sci*., **9**, 232–241.
Salamov,A.A., Suwa,M., Orengo,C.A. and Swindells,M.B. (1999) *Protein Eng*., **12**, 95–100.
Sali,A. and Blundell,T.L. (1993) *J. Mol. Biol.*, **234**, 779–815.
Shindyalov,I.N. and Bourne,P.E. (1998) *Protein Eng*., **11**, 739–747.
Teichmann,S.A., Chothia,C., Church,G.M. and Park,J. (2000) *Bioinformatics*, **16**, 117–124.

Received November 30, 2001; revised April 30, 2002; accepted May 3, 2002