BIOINFORMATICS ORIGINAL PAPER Vol. 25 no. 19 2009, pages 2500–2505
doi:10.1093/bioinformatics/btp446

Sequence analysis

Benchmarking homology detection procedures with low complexity filters

Kristoffer Forslund* and Erik L. L. Sonnhammer

Stockholm Bioinformatics Center, Stockholm University, SE-10691 Stockholm, Sweden

Received on March 16, 2009; revised on July 14, 2009; accepted on July 15, 2009

Advance Access publication July 20, 2009

Associate Editor: Dmitrij Frishman

ABSTRACT

Background: Low-complexity sequence regions present a common problem in finding true homologs to a protein query sequence. Several solutions to this have been suggested, but a detailed comparison between these on challenging data has so far been lacking. A common benchmark for homology detection procedures is to use SCOP/ASTRAL domain sequences belonging to the same or different superfamilies, but these contain almost no low complexity sequences.

Results: We here introduce an alternative benchmarking strategy based around Pfam domains and clans on whole-proteome data sets. This gives a realistic level of low complexity sequences. We used it to evaluate all six built-in BLAST low complexity filter settings as well as a range of settings in the MSPcrunch post-processing filter. The effect on alignment length was also assessed.

Conclusion: Score matrix adjustment methods provide a low false positive rate at a relatively small loss in sensitivity relative to no filtering, across the range of test conditions we apply. MSPcrunch achieved even less loss in sensitivity, but at a higher false positive rate. A drawback of the score matrix adjustment methods is however that the alignments often become truncated.

Availability: Perl scripts for MSPcrunch BLAST filtering and for generating the benchmark dataset are available at http://sonnhammer.sbc.su.se/download/software/MSPcrunch+Blixem/benchmark.tar.gz

Contact: kristoffer.forslund@sbc.su.se

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

A crucial part of many bioinformatics applications is determination of whether or not two sequences should be seen as homologous, that is to say, that they share a common ancestral sequence. While several techniques have emerged for achieving this, pairwise alignment-type methods have emerged as the leaders in the field; by aligning the sequence under a hypothesis of a common origin, the minimal distance between them, under some measure, becomes defined. The popular BLAST approach (Altschul *et al*., 1990) provides statistical support for evaluating such measures, by comparing with the number of equally similar sequences found in a database search in the absence of homology. As a result, it is possible to determine whether two sequences can be seen as homologous with a given degree of confidence, under a given set of assumptions on evolutionary rates and similar properties.

Many protein or nucleic acid sequences are classified as having low complexity, for instance by being composed of repetitive subsequences or being heavily biased towards a subset of characters in the given alphabet. This is a common source of false positive homology assignments: while there is little reason to believe these sequence features stem from a common origin in many cases, proteins or genes that share them will nevertheless appear relatively similar to sequence comparison algorithms. The classical approach towards handling this in the BLAST software is to apply low-complexity masking. By adapting a tool that recognizes regions of low complexity, these may be masked so that the algorithm ignores them (Altschul *et al*., 2005; Wootton and Federhen, 1996; Yu and Altschul, 2005; Yu *et al*., 2003). In the case of nucleotide sequences, the RepeatMasker (http://www.repeatmasker.org) application, has proven to be very effective. It recognises both low-complexity regions, tandem repeats, and interspersed repeats such as transposons, and replaces these sequence regions with strings of mask characters.

Another approach is to adapt the evolutionary model to the sequence composition of query and database protein in a homology search. This has recently been implemented in BLAST by adjusting the substitution matrix according to the sequences if any of a set of criteria are met (Altschul *et al*., 2005; Schäffer *et al*., 2001; Yu and Altschul, 2005; Yu *et al*., 2003). The inverse approach, i.e. leaving the substitution matrix unchanged but evaluating whether a given match score is mainly caused by biased sequence composition, has long been implemented in the MSPcrunch BLAST parser (Sonnhammer and Durbin, 1994).

We here focus specifically on the issue of low complexity as an error source in protein homology detection. This is important for applications such as the InParanoid orthology database (Berglund *et al*., 2008), which relies on finding as many correct homology relationships as possible between two proteomes. We wish to compare available low-complexity handling approaches to determine which method most effectively avoids false positive homology assignments while retaining as many *bona fide* homologs as possible. To this end, we needed a reliable dataset that covers a wide range of protein types and contains both true positive and true negative relationships.

*To whom correspondence should be addressed.

Many homology detection methods have been evaluated using the SCOP/ASTRAL dataset as a benchmark (Altschul *et al*., 2005; Chandonia *et al*., 2004; Wistrand and Sonnhammer, 2005; Yu *et al*., 2006). SCOP is a domain definition database derived from experimentally determined protein 3D structures, and groups domains in those proteins into superfamilies if there are structural reasons for considering them homologous (Murzin *et al*., 1995). ASTRAL is a database of the corresponding amino acid sequences (Chandonia *et al*., 2004). Generally, ASTRAL sequences belonging to the same superfamily can reliably be considered homologous, whereas those belonging to different superfamilies are non-homologous. For our purposes, this is insufficient for several reasons. First, sequences in ASTRAL contain a far smaller fraction of low-complexity sequences than a dataset containing all human protein sequences in ENSEMBL (Wootton and Federhen, 1996). This is not unexpected, as SCOP/ASTRAL is a domain dataset rather than a full-length protein dataset, but also because it is biased towards easily crystallizable and well-characterised proteins. Second, it is limited in size and coverage to the proteins represented in the PDB. While the SUPERFAMILY (Gough *et al*., 2001) database allows assignment of SCOP domains via HMMs to other proteins, it is equally biased to protein domains that have been structurally determined.

To avoid this bias and obtain a more representative dataset, we here exploit the domain architecture of multi-domain proteins in Pfam (Finn *et al*., 2008). We define trusted positive and negative training sets as proteins with either identical or entirely different domain architectures. Our comparison is made in the context of whole organism versus organism protein homology searches. Genome coverage and expandability of the benchmark dataset is important to us, as well as avoiding any bias against particular categories of proteins, for instance membrane proteins. Because of this, we present a complementary benchmark test for homology detection approaches based on the Pfam database, including the recent addition of Pfam clans, a higher-level hierarchy of evolutionary related domain families.

2 METHODS

2.1 Protein domains

Our work is based on using the distant evolutionary relationships represented in proteins where the domain architectures are the same. Protein domains are recurring sequence or structure elements found in several different contexts, and several frameworks for classifying and recognizing instances of a domain family have been developed. In this work, we employed specifically version 22.0 of the Pfam database (Finn *et al*., 2006, 2008; Sonnhammer *et al*., 1998), which employs hidden Markov model (Durbin *et al*., 1998; Krogh *et al*., 1994) profile techniques for defining and detecting domains; these profiles are built using manually curated alignments of sequences considered to belong to the same domain.

2.2 Defining a benchmark dataset for homology detection

For defining a set of high-confidence homologous proteins, our basic approach was the assumption that two multi-domain proteins, with exactly the same sequence of domains, are unlikely not to have a common ancestor. We based this on the relative rarity with which domain architectures arise more than once in evolution. While in some cases non-homologous proteins may have identical domain architectures, it is sufficiently uncommon for us to disregard at this point (Forslund *et al*., 2008; Gough, 2005). However, it is probable that a training set limited to only proteins with exactly the same domain architecture may be unrealistically restrictive, and unable to represent situations where weak yet significant indication of homology exists. Because of this, we also considered as homologs pairs of proteins with the same domain architecture, where we classified two domains as equal if they were members of the same Pfam clan.

Defining a negative test set for homology, however, is more difficult. Again, we employed the concept of protein domains. Using sophisticated techniques such as the Pfam hidden Markov models, common domain assignment is highly sensitive and applicable at far larger evolutionary separation than direct sequence comparison methods such as protein BLAST (Madera and Gough, 2002; Wistrand and Sonnhammer, 2005). If two proteins both have well-defined domain architectures, but share no domains at all, chances are low that they should be homologs. However, as Pfam domains from different families are known to be sometimes related, we further required that the proteins contain no domains which are part of the same Pfam clan (Finn *et al*., 2006).

To make the benchmark more robust, we required that proteins in both the negative and positive set contained at least two domains. For any Pfam family classified as Repeat or Motif, any sequences of such domains were collapsed into a single pseudo-domain, as it is known that the number of elements in such repeat regions vary significantly across even short evolutionary distances.

2.3 Sequence sources

We considered on one hand the situation of finding homologous genes between two species, and on the other, of finding within-species homologs. The reason for this is that the amino acid composition and frequency of low complexity regions often have species-dependent features. For instance in InParanoid, we observed very different filtering needs for intra-species versus inter-species comparisons. For the purpose of this benchmark, we selected a small number of species that represent different evolutionary contexts, as well as some species where unique issues are a factor. We have strived to represent species with a varying degree of evolutionary separation from *Homo sapiens*. Thus, the list of species included are human, chimpanzee (*Pan troglodytes*), worm (*Caenorhabditis elegans*), yeast (*Saccharomyces cerevisiae*), slime mold (*Dictyostelium discoideum*), the plant *Arabidopsis thaliana* and the bacterium *Escherichia coli*, K12 strain. *Dictyostelium* was chosen specially, as it contains a vast number of low-complexity regions such as single-residue or pair repeats, which present a large risk of false positive homology assignments unless corrected for (Eichinger *et al*., 2005)

The proteome sequences for these species were acquired from the appropriate model organism databases, after which Pfam domain detection was performed using the HMMER version 2.3.2 (Eddy, 2008) software and the HMMs corresponding to version 22.0 of Pfam. The analysis was performed with an all versus all query within each species included, and with the human proteome used as a query against all other species, representing a situation where we seek all homologs in model organisms.

2.4 Methods evaluated

Our analysis is limited to approaches that attempt to correct for low-complexity regions in the context of a protein BLAST database search, as these are the circumstances under which the issue might arise for the InParanoid orthology assignment framework.

Classically, protein BLAST provides either ‘hard’ or ‘soft’ sequence masking using the SEG filter (Wootton and Federhen, 1996), which can be tuned differently from its default parameters. When SEG flags a region as having low complexity, under soft masking, BLAST local alignments cannot originate in that region, but may extend across it if doing so increases the alignment score. Under hard masking, residue letters in regions flagged by SEG are changed into X, which penalizes alignment across those regions.

In recent versions of the BLAST software, a feature has been added enabling the adjustment of the substitution matrix according to the amino acid distributions of the two sequences being compared. This may be applied to all comparisons, or merely to comparisons that fulfill specific risk criteria, such as very similar composition or very different lengths (Altschul *et al*., 2005; Schäffer *et al*., 2001; Yu and Altschul, 2005; Yu *et al*., 2003). Under the adjusted substitution matrix, similarities deriving merely from compositional bias will receive a lower score.

2.5 The MSPcrunch method

The MSPcrunch BLAST parser (Sonnhammer and Durbin, 1994) uses a different approach to avoid biased composition matches. For each BLAST local alignment, the expected score is computed for two random sequences with the same amino acid composition and gap distribution as the query and database sequence. If the actual score is not significantly higher than this expected random score, that particular local alignment is excluded from the search. We here update the original MSPcrunch method to also model the gap penalty.

The filtering works as follows. Consider a BLAST High-scoring Segment Pair (HSP). Let Q and D be vectors of local amino acid frequencies for the query and database sequence, respectively. It can then be shown that the expected score of a BLAST alignment of two random sequences with these amino acid compositions is

$$S_{\exp} = L \sum_i \sum_j Q_i D_j M_{ij}$$

where L is the length of the specific HSP being analyzed and M is the scoring (substitution) matrix. The local frequency vectors Q and D for the HSP are estimated using pseudocounts as

$$Q_i = \frac{Qc_i + \alpha p_i}{L+\alpha}; \quad D_i = \frac{Dc_i + \alpha p_i}{L+\alpha}.$$

given raw amino acid counts Qc and Dc for the query and database HSP sequences, respectively, and prior amino acid frequencies p. α is a scaling factor empirically set to 5. The biased composition ratio β is then defined as

$$\beta = \frac{S - S_{\exp}}{S - LM_{\exp}}$$

where $M_{\exp}$ is the frequency-weighted expected score of random sequences when using score matrix M, as

$$M_{\exp} = \sum_i \sum_j p_i p_j M_{ij}$$

For M = BLOSUM62, which was used for all alignments in the present work, $M_{\exp}$ is -0.945. The ratio β represents the score increase above what is expected given the composition, normalised by the score increase relative to random sequences. The filtering is then performed on an HSP per HSP basis. If β falls below a certain threshold, the composition of the HSP is considered too biased, and it is excluded from the alignment of the query and database proteins as a whole. The threshold $\beta_{\min}$ was set to 0.8 in previous applications, but in the present work, we also evaluated a wider range of parameter values.

Extending the algorithm to take gaps into account is straightforward. We also evaluated this strategy (data not shown), and found that using gaps increased MSPcrunch sensitivity, but decreased precision even more, resulting in relatively lower Matthew's correlation coefficient (MCC) scores. Hence, the ungapped version was used in the tests reported in this work.

2.6 Test strategies

All of these approaches may be tuned in different ways, by supplying different parameters. We evaluated their performance under a range of parameters on the dataset in question by performing an all versus all homology search both within and between the proteomes. For each method, we investigated its ability to detect true homolog pairs and its rate of reported non-homologous proteins above a given cutoff. See Supplementary Table S1 for details on the tested parameter settings.

Homology searches within the proteome datasets, using the same proteome as batch query, were performed using version 2.2.18 of the NCBI BLAST standalone application. For each query-database sequence pair, the highest scoring HSP was chosen. Other HSPs were considered for inclusion in order of descending scores. These were retained if they were compatible (preserving relative sequence order) with all the previously retained HSPs, and did not overlap with any of those by more than 5%. For all HSPs retained, the sum of their bit scores was taken. Only hits with a bit score sum of at least 40 were considered. The same software and parameters were used for the proteome versus proteome searches.

2.7 Performance evaluation metrics

A positive match is defined as a significant hit under the scheme in question, whereas a negative is the absence of a hit. Let TP be the number of true positives, TN the number of true negatives, FP the number of false positives and FN the number of false negatives. Sensitivity (or recall) is defined as TP/(TP + FN) and precision (or positive predictive value) is defined as TP/(TP + FP). MCC is defined as $(\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}) / \sqrt{[(\mathrm{TP}+\mathrm{FP}) \cdot (\mathrm{TP}+\mathrm{FN}) \cdot (\mathrm{TN}+\mathrm{FP}) \cdot (\mathrm{TN}+\mathrm{FN})]}$.

ROC curves for the methods tested are displayed in Figure 3. These were generated by ranking all homology assignments in each dataset by bit score. The hits were then pooled across all datasets for each method, and the resulting meta-dataset was sorted according to rank in the original datasets. Cumulative counts of true and false hits were plotted as ROC curves showing the performance of the various method for successively lower-scoring homology assignments.

2.8 Effects of low-complexity filters on alignment lengths

Even for homology assignments that are retained when low-complexity filters are applied, the length of the region aligned by BLAST, may be shorter. To investigate the degree to which this occurs, as well as how the effect might differ between methods, we performed an analysis as follows. For each low-complexity filtering method, we considered every true positive homology assignment that it shares with the results achieved by running unfiltered BLAST (our negative control), under the assumption that unfiltered yields the longest possible alignment. For each such assignment, we considered the maximum aligned length, which is the longest aligned residue span on either the query or database sequence. If the match contained multiple consistent HSPs using InParanoid's consistency rules (Remm *et al*., 2001), then the distance from the start of the first HSP to the end of the last HSP was used. We then recorded the number of assignments where the maximum segment length was half the length or shorter with the filter applied than without. The results of this analysis for each combination of method and dataset are shown in Supplementary Tables S3A and B.

3 RESULTS

A benchmark dataset was generated for each genome versus genome comparison by flagging pairs of proteins as homologs or non-homologs depending on their Pfam domain family and clan assignments. Only proteins with domains from at least two different families were included. Pairs where the sequence of domains was the same (or where corresponding domains belonged to the same clan) were considered true homologs, whereas pairs where no domain in the one protein was the same as or shared a clan with any domain in the other protein were considered true non-homologs. This was done for human and six other species, ranging from closely related (chimpanzee) to distantly related (*E.coli*). All species but one were

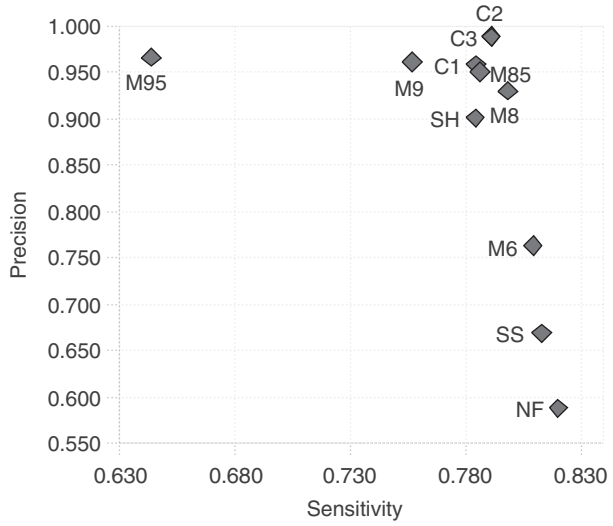

Fig. 1. Intraspecies benchmark result. The average precision and sensitivity across all species comparisons (each species versus itself) is plotted for each method. M<X>: MSPcrunch β_{min}=0.<X>; NF: No filtering; SS: SEG soft masking; SH: SEG hard masking; C1: 2001 version of score matrix adjustment; C2: compositional score matrix adjustment, conditional; C3: compositional score matrix adjustment, unconditional.

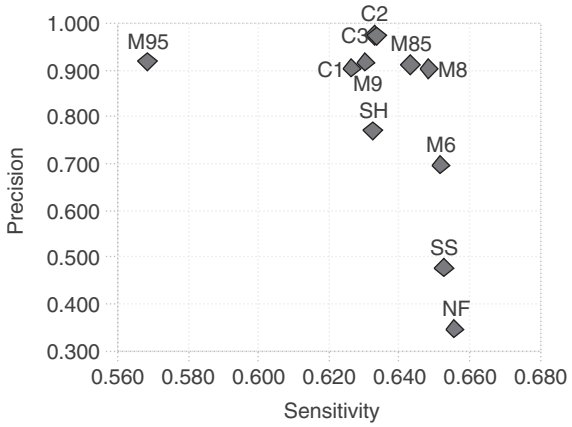

Fig. 2. Interspecies benchmark results. The average precision and sensitivity across all species comparisons (*H.sapiens* versus other species) is plotted for each method. M<X>: MSPcrunch β_{min}=0.<X>, NF: No filtering; SS: SEG soft masking; SH: SEG hard masking; C1: 2001 version of score matrix adjustment; C2: compositional score matrix adjustment; conditional, C3: compositional score matrix adjustment, unconditional.

eukaryotes, as these tend to have far more low complexity sequences than prokaryotes. The number of positive and negative pairs in each species comparison are shown in Supplementary Table S2.

In total, 11 methods were benchmarked for their ability to avoid biased composition matches. The methods were all parameter variations of either the NCBI BLAST program or of an updated version of the MSPcrunch BLAST post-processing application. BLAST was run either without filter, with two variants of the SEG filter, or with three composition-based score matrix adjustment methods. MSPcrunch was run with five different cutoffs. Note that the score matrix adjustment methods and MSPcrunch always were paired with SEG soft masking.

The full details of the benchmark results are listed in Supplemental Table 1, providing sensitivity and precision values for each species. In general, the precision showed much stronger dependence on the filtering method than the sensitivity. There were substantial differences between the performances on different species comparisons. For instance, *E. coli* and *A. thaliana* yielded high intraspecies precision with all methods, even without any filtering, but for all others species filtering methods were needed to obtain this. For *Dictyostelium*, which contains a large fraction of low-complexity regions, paralog assignments are more or less useless (precision 1.7%, sensitivity 87.0%) without low-complexity filtering. However, it is also clear that the score matrix adjustment approach is sufficient to avoid these errors even in such extreme cases without sacrificing much sensitivity.

In order to summarize the results, we plotted for each method the arithmetic mean of the precision versus sensitivity values in Figures 1 (intraspecies comparisons) and 2 (interspecies comparisons). In both these plots, the score matrix adjustment method (conditional or unconditional) yields the highest precision at relatively modest loss of sensitivity.

Another way to summarize the results is to calculate the Matthew Correlation Coefficient (MCC), which reflects both precision and sensitivity at the same time (see Tables 1 and 2). On this test, the new score matrix adjustment methods performed on average the best. Next best was either the old-score matrix adjustment method (intraspecies) or MSPcrunch $\beta = 0.85$ (interspecies). However, the MCC difference between these methods results was small. The overall highest MCC in the test was obtained by MSPcrunch, and in several of the species comparisons (four intraspecies and two interspecies) MSPcrunch outperformed the score matrix adjustment approaches. A drawback with MSPcrunch is that no β parameter was consistently the best choice, but the optimum was generally found in the range 0.8–0.9.

By pooling results across all species comparisons, we also generated receiver operating characteristic (ROC) curves for the various methods, as shown in Figure 3. Also here the score matrix adjustment methods had the best trade-off between precision and sensitivity, but could not fully achieve the same coverage as the other methods.

In summary, based on the range of species comparisons evaluated here, if avoiding false positives is the main objective then the score matrix adjustment methods are most reliable. However, they do come with a higher false negative rate, so if sensitivity is more important then MSPcrunch can be a better choice. Obviously, not filtering at all gives the highest sensitivity, but runs the danger of an extreme false positive rate. Whether or not compositional bias is applied universally or conditionally (corresponding to the -C2 and -C3 options in NCBI BLAST, respectively) made very little difference in our benchmark, implying that the application criteria as described by Altschul *et al*. (2005) were strict enough to capture most cases where score matrix adjustment would cause drastic differences in outcome.

An additional factor to consider is the degree to which biologically significant alignments may be truncated by the low complexity filtering methods. We found that up to 6% of the true positive homology assignments were reported with alignments truncated

Table 1. Intraspecies MCC for each method across all species versus itself comparisons

	EC–EC	AT–AT	SC–SC	DD–DD	CE–CE	PT–PT	HS–HS	Average
No filtering	0.895	0.869	0.685	0.109	0.872	0.534	0.511	0.639
SEG soft masking	**0.894**	0.886	0.762	0.173	0.896	0.632	0.616	0.694
SEG hard masking	**0.884**	0.893	0.823	0.805	0.865	0.798	0.797	0.838
Score matrix adjustment, 2001 version	0.885	0.894	0.835	0.873	0.893	0.839	0.842	0.866
Score matrix adjustment, unconditional	0.886	0.897	0.843	**0.889**	0.897	**0.883**	**0.888**	**0.884**
Score matrix adjustment, conditional	0.886	0.897	0.843	**0.889**	0.897	**0.883**	**0.888**	0.883
MSPcrunch, β= 0.6	0.894	0.898	0.826	0.409	**0.912**	0.717	0.704	0.766
MSPcrunch, β= 0.8	**0.894**	**0.901**	**0.844**	0.789	0.899	0.846	0.847	0.860
MSPcrunch, β= 0.85	0.890	0.898	0.838	0.825	0.880	0.854	0.855	0.863
MSPcrunch, β= 0.9	0.865	0.891	0.825	0.823	0.852	0.851	0.853	0.852
MSPcrunch, β= 0.95	0.755	0.856	0.748	0.712	0.795	0.816	0.817	0.786

The highest MCC for each species comparison is bolded.

Table 2. Interspecies MCC for each method across all species versus *H.sapiens* comparisons

	HS–EC	HS–AT	HS–SC	HS–DD	HS–CE	HS–PT	Average
No filtering	0.567	0.424	0.484	0.242	0.519	0.489	0.454
SEG soft masking	0.616	0.514	0.579	0.325	0.618	0.594	0.541
SEG hard masking	0.665	0.676	0.709	0.540	0.778	0.773	0.690
Score matrix adjustment, 2001 version	0.667	0.715	0.744	0.703	0.841	0.819	0.748
Score matrix adjustment, unconditional	0.672	**0.749**	0.773	**0.750**	**0.878**	**0.870**	**0.782**
Score matrix adjustment, conditional	0.672	0.748	0.774	0.749	0.876	0.869	0.781
MSPcrunch, β= 0.6	0.666	0.656	0.693	0.557	0.736	0.682	0.665
MSPcrunch, β= 0.8	0.685	0.740	**0.783**	0.687	0.839	0.825	0.760
MSPcrunch, β= 0.85	**0.686**	0.738	0.779	0.686	0.843	0.835	0.761
MSPcrunch, β= 0.9	0.683	0.733	0.770	0.677	0.840	0.833	0.756
MSPcrunch, β= 0.95	0.652	0.677	0.728	0.646	0.808	0.798	0.718

The highest MCC for each species comparison is bolded.

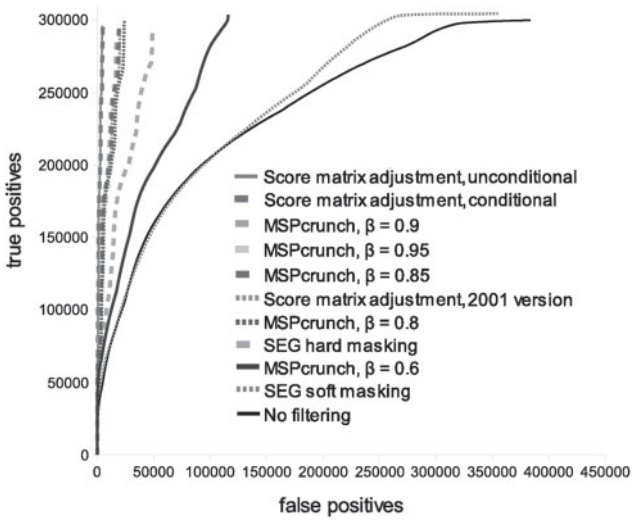

Fig. 3. ROC curves for methods based on results pooled across all species in the benchmark.

more than 50% by the filtering (Supplementary Table 2). This can have large consequences for applications where the match length is an important parameter, e.g. orthology analysis (Remm *et al*., 2001).

The truncation varied between datasets, but was generally highest for the SEG hard masking and score matrix adjustment methods, whereas it was low for SEG soft masking and most MSPcrunch settings.

4 DISCUSSION

We have presented a basic approach to defining a benchmark dataset for homology inference evaluation, based on domain assignments. As such data is easily made available for any set of amino acid sequences, scaling up or regenerating such a benchmark dataset is trivial and immediately available. A potential concern is that if the Pfam clan system is incomplete, some negatives in our benchmark would be true homologs. While this is certainly a potential source of error, we feel that our approach is justified given the lack of better options for large-scale benchmarks of this type. As previously stated, a SCOP benchmark, while clearly very accurate, fails to capture method performance in the types of situations we are interested in. An alternative could be to use random pairs as negatives, but as the present approach could be seen as an exhaustive enumeration of random pairs where the majority of potential homologs are removed, we estimate that the frequency of false negatives in the benchmark should always be lower than with a random pair-based negative set.

Our negative homology dataset could be further improved by considering Pfam-A domains that as yet share no clan as potential

clanmates for this purpose if they are similar enough with respect to an HMM–HMM comparison, even if they are not currently classified as such. Another possible improvement could come from removing any protein pairs with shared Pfam-B domains from the set of non-homologs. In addition, the negative homology dataset could be restricted only to proteins where there are no long unassigned regions, though this might also remove many low-complexity regions that we are interested in.

We are aware that the choice of species in the benchmark will influence its outcome. In this case, however, the species pairs defined by our selection span the entire range of evolutionary separation, as well as a wide variety of different types of organisms. Expanding the dataset further would be trivial. However, we saw sufficient agreement between species for the general trends, and therefore did not choose to include more species for the purpose of the present analysis. While there was variation in the results we achieve between different species comparison, some general patterns seemed to hold true.

No benchmarked method (unfiltered BLAST excluded) was able to reach more than ∼87% sensitivity, whereas the precision reached almost to 100%. Looking at the homolog pairs not detected for the comparisons with the highest sensitivity, about half of these appeared to be cases where domains belonged to different families but the same clan. As Pfam clans represent a large evolutionary separation, usually detectable only with profile-based methods such as hidden Markov models (Madera and Gough, 2002; Wistrand and Sonnhammer, 2005), this is hardly unexpected. A quarter of the false negatives involved proteins consisting only of domains classified in Pfam as type Motif/Repeat. As these are frequently short and variable in copy number it is perhaps not surprising that these are hard to find by BLAST.

Our results show once more how absolutely crucial handling of biased sequence composition is in any form of sequence similarity-based homology assignment, especially for genomes with uncommon features such as a large incidence of simple repeats. We compared some approaches thus far presented for handling this problem, including past and present default options for the popular BLAST search algorithm. In conclusion, the recently added score matrix adjustment approach appears to be the most reliable solution among those tested, for the range of species comparisons we evaluated. However, MSPcrunch can achieve higher sensitivity, though at a relatively larger trade-off in precision, than score matrix adjustment, particularly with a β threshold adapted for the species in question using a benchmark strategy such as the one outlined here.

Homology searching is often used to extract information from the obtained alignment, for instance for domain analysis. For ortholog identification it is common to require the match to span more than 50% of the sequence to be considered. This may be compromised when using strong complexity filters, particularly the score matrix adjustment methods and hard masking. When using such approaches it may be necessary to re-align the hits using less or no filtering in a second phase to obtain the full alignment extent. Given that the sequences are considered true homologs, there is little reason to truncate the alignment relative to the unfiltered version. An option in BLAST to remove the score matrix adjustment for the final alignment would therefore be very useful.

ACKNOWLEDGEMENTS

The authors would like to thank the NCBI Blast staff for useful support during the course of this work.

Funding: Grant from Pharmacia.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*., **25**, 3389–3402.

Altschul,S.F. *et al*. (2005) Protein database searches using compositionally adjusted substitution matrices. *FEBS J*., **272**, 5101–5109

Berglund,A.C. *et al*. (2008) InParanoid 6: eukaryotic ortholog clusters with inparalogs. *Nucleic Acids Res*., **36**, D263–D266.

Chandonia,J.M. *et al*. (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res*., **32**, D189–D192.

Durbin,R. *et al*. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, UK.

Eddy,S.R. (2008) A probabilistic model of local sequence alignment that simplifies statistical significance estimation. *PLoS Comput. Biol*., **4**, e1000069.

Eichinger,L. (2005) The genome of the social amoeba Dictyostelium discoideum. *Nature,* **435**, 43–57.

Finn,R.D. *et al*. (2006) Pfam: clans, web tools and services. *Nucleic Acids Res.,* **34**, D247–D251.

Finn,R.D. *et al*. (2008) The Pfam protein families database. *Nucleic Acids Res*., **36**, D281–D288.

Forslund,K. *et al*. (2008) Domain tree-based analysis of protein architecture evolution. *Mol. Biol. Evol*., **25**,254–264.

Gough,J. (2005) Convergent evolution of domain architectures (is rare). *Bioinformatics,* **21**, 1464–1471.

Gough,J. *et al*. (2001) Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *J. Mol. Biol*., **313**, 903–919.

Krogh,A. *et al*. (1994). Hidden Markov models in computational biology. Applications to protein modeling. *J. Mol. Biol*., **235**, 1501–1531.

Madera,M. and Gough,J. (2002) A comparison of profile hidden Markov model procedures for remote homology detection. *Nucleic Acids Res*., **30**, 4321–4328.

Murzin,A.G. *et al*. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol*., **247**, 536–540.

Remm,M. *et al*. (2001) Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.*, **314**, 1041–1052.

Schäffer,A.A. *et al*. (2001). Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.*, **29**, 2994–3005.

Sonnhammer,E.L. and Durbin,R. (1994) An expert system for processing sequence homology data. *ISMB,* **2**, 363–368.

Sonnhammer,E.L. *et al*. (1998) Pfam: multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Res*., **26**, 320–322.

Wistrand,M. and Sonnhammer,E.L. (2005) Improved profile HMM performance by assessment of critical algorithmic features in SAM and HMMER. *BMC Bioinformatics*, **6**, 99.

Wootton,J.C. and Federhen,S. (1996) Analysis of compositionally biased regions in sequence databases. *Methods Enzymol*., **266**, 554–571.

Yu,Y.K. and Altschul,S.F. (2005) The construction of amino acid substitution matrices for the comparison of proteins with non-standard compositions. *Bioinformatics,* **21**, 902–911.

Yu,Y.K. *et al*. (2003) The compositional adjustment of amino acid substitution matrices. *Proc. Natl. Acad. Sci. U S A,* **100**, 15688–15693.

Yu,Y.K. *et al*. (2006) Retrieval accuracy, statistical significance and compositional similarity in protein sequence database searches. *Nucleic Acids Res*., **34**, 5966–5973.