doi:10.1006/jmbi.2001.5197 available online at http://www.idealibrary.com on IDEAL® J. Mol. Biol. (2001) 314, 1041–1052

JMB

Automatic Clustering of Orthologs and In-paralogs from Pairwise Species Comparisons

Maido Remm[1,2], Christian E. V. Storm[1] and Erik L. L. Sonnhammer[1*]

[1]*Center for Genomics and Bioinformatics, Karolinska Institutet, S-17177 Stockholm Sweden*

[2]*Estonian Biocentre, Riia 23 Tartu 51010, Estonia*

**Corresponding author*

Orthologs are genes in different species that originate from a single gene in the last common ancestor of these species. Such genes have often retained identical biological roles in the present-day organisms. It is hence important to identify orthologs for transferring functional information between genes in different organisms with a high degree of reliability. For example, orthologs of human proteins are often functionally characterized in model organisms. Unfortunately, orthology analysis between human and e.g. invertebrates is often complex because of large numbers of paralogs within protein families. Paralogs that predate the species split, which we call out-paralogs, can easily be confused with true orthologs. Paralogs that arose after the species split, which we call in-paralogs, however, are *bona fide* orthologs by definition.

Orthologs and in-paralogs are typically detected with phylogenetic methods, but these are slow and difficult to automate. Automatic clustering methods based on two-way best genome-wide matches on the other hand, have so far not separated in-paralogs from out-paralogs effectively.

We present a fully automatic method for finding orthologs and in-paralogs from two species. Ortholog clusters are seeded with a two-way best pairwise match, after which an algorithm for adding in-paralogs is applied. The method bypasses multiple alignments and phylogenetic trees, which can be slow and error-prone steps in classical ortholog detection. Still, it robustly detects complex orthologous relationships and assigns confidence values for both orthologs and in-paralogs. The program, called INPARANOID, was tested on all completely sequenced eukaryotic genomes. To assess the quality of INPARANOID results, ortholog clusters were generated from a dataset of worm and mammalian transmembrane proteins, and were compared to clusters derived by manual tree-based ortholog detection methods. This study led to the identification with a high degree of confidence of over a dozen novel worm-mammalian ortholog assignments that were previously undetected because of shortcomings of phylogenetic methods.

A WWW server that allows searching for orthologs between human and several fully sequenced genomes is installed at http://www.cgb.ki.se/inparanoid/. This is the first comprehensive resource with orthologs of all fully sequenced eukaryotic genomes. Programs and tables of orthology assignments are available from the same location.

Keywords: orthologs; paralogs; automatic clustering; genome comparison

Introduction

With the rapidly growing amount of sequence data, the need for automatic analysis methods for biological discovery is growing too. The sequencing of the human genome, one of the most important milestones in biology and medicine of our age, is nearly completed. Many scientists are now asking which genes the human genome has in common with other species. A particularly important question is which genes in the human genome are sharing the exact same biological function with genes in simpler organisms. There are numerous so-called model organisms that are well studied and can be used for unraveling gene functions. As

E-mail address of the corresponding author: Erik.Sonnhammer@cgb.ki.se

0022-2836/01/051041–12 $35.00/0

in vivo experiments with human genes are not feasible, the ability to infer the function of human genes from the function of corresponding genes in model organisms is highly desirable.

To be able to infer which genes have the same function, we need to understand how the genes evolved have. Genes in two species that have directly evolved from a single gene in the last common ancestor are most likely to share the function. Such genes are called orthologs.[1] Often, the sequences have duplicated after the speciation event (i.e. after the two species diverged from each other). In this case there is more than one ortholog in one or both species and the orthologs are said to have a one-to-many or many-to-many relationship. In such cases, it is non-trivial to determine which of the orthologs is functionally equivalent to the ortholog in the other species. It may be only one, but several genes could also have redundant functions.

Due to the uncertainty of functional equivalence between the orthologs deriving from a single ancestor at the time of speciation, it is important to detect all of them. As these are homologs found in the same genome, they are called paralogs.[1] However, there may also be paralogs that arose from a duplication event before the speciation. These are therefore not orthologs according to the definition. Unfortunately, there is no accepted terminology to separate paralogs that were duplicated before a speciation event from paralogs that were duplicated after it. We here propose two new terms, in analogy with the phylogenetic concepts of outgroup and in-group. Paralogs predating the speciation event that thus are not orthologs are denoted out-paralogs. Paralogs that were duplicated after the speciation event, and thus are orthologs, are denoted in-paralogs. A potential synonym for in-paralog could be co-ortholog but we prefer in-paralog because of the symmetry with out-paralog.

Automatic detection of orthologs and in-paralogs from full genomes is an important but challenging problem. As orthologs, by definition, are related through evolutionary history, phylogenetic trees are the most natural way to detect orthologs. Unfortunately, construction of phylogenetic trees involves some poorly automatable steps and demands large resources of computing power. Carrying out this approach for all genes of two or more genomes would require clustering of homologs, generation of correct multiple alignment for each group of homologous domains, construction of a phylogenetic tree for each group, and finally extraction of orthologs from these trees. Approaches for automating the final step exist,[2] but current methods for automatic generation of multiple alignments of domains still yield sub-standard quality output, which makes subsequent orthology analysis unreliable.

An alternative to phylogenetic methods is to use all-*versus*-all sequence comparison between two genomes to detect orthologs. The idea is that if the sequences are orthologs, they should score higher with each other than with any other sequence in the other genome. This method does not use multiple alignments or phylogenetic trees and therefore avoids potential errors that might be introduced at these steps. All-*versus*-all BLAST searches has recently gained popularity for finding orthologs.[3–6] Originally, it was used mainly used to detect simple one-to-one relationships.[7] With the appearance of several fully sequenced genomes, the COG database was created.[7–9] The COG database is a collection of BLAST-based ortholog groups from multiple species. The species are chosen to be from distant phylogenetic lineages, although multicellular eukaryotes are not included. According to its authors, it does not represent a comprehensive phylogenetic analysis, but still provides a fast and convenient short-cut to delineate a large number of groups that most likely consist of orthologs.[10] The members of a COG entry must belong to at least three species. It therefore represents sequences whose function is conserved across major phylogenetic lineages. This is useful in many cases, particularly if people work with diverse prokaryotic organisms. With more genomes being sequenced, the need for finding orthologous proteins between two closely related species also appears. The orthologous groups between mouse and human[11] or chimpanzee and human are very different from the orthologous groups between human and yeast or human and *Escherichia coli* (see also Results). Many investigators are interested in questions such as: which are the human orthologs of a given Drosophila gene or which are the mouse orthologs of a given human gene? These questions cannot be answered with one single answer for all species, all pairwise comparisons produce different groups of orthologs, depending on which genes their common ancestor had.

To resolve these questions without the time-consuming and error-prone phylogenetic analysis, we designed a program, INPARANOID, that identifies orthologs and in-paralogs between any given pair of genomes. INPARANOID stands for in-paralog and ortholog identification. Out-paralogs are not reported. The methodology can be seen as an extension of the all-*versus*-all technique, but with special rules for cluster analysis in order to extract all in-paralogs. The INPARANOID algorithm is presented here, as well as an assessment of its performance when tested on a set of previously validated ortholog assignments. Also, it was applied to the complete set of protein sequences from the *Caenorhabditis elegans* and *Drosophila melanogaster* genomes.

INPARANOID

Input data

The program expects two datasets of protein sequences in multiple FASTA formats. The datasets, denoted A and B, should be in two separate files. These datasets are expected to include the complete set of protein sequences from two species. Incomplete sets of genes could be used, but this may result in an incorrect list of orthologs. In principle, sequences from any phylogenetic clade could be used instead of one species. For example, in the absence of a complete human gene set, one could consider using all mammalian genes as one dataset. This would give the chance of detecting mammalian orthologs even if the appropriate human gene is not sequenced or has been lost. The program has the option to use a third dataset of sequences as an out-group species. An overview of the algorithm is shown in Figure 1.

Pairwise similarity scores

The detection of orthologs starts with calculation of all pairwise similarity scores between all studied sequences. This is usually done with the BLAST program for speed, but it could be done with any other pairwise alignment program. Pairwise similarity scores are calculated in four separate steps; A *versus* B, B *versus* A, A *versus* A, B *versus* B. Although it is not really necessary to use separate steps, this helps to organize the data and thus helps to reduce memory requirements of the algorithm. If an out-group species (dataset C) is used, the similarity scores between dataset A *versus* C and dataset B *versus* C are calculated. The BLAST program occasionally reports asymmetric scores between sequence pairs X-Y and Y-X. To avoid problems that these asymmetric scores could cause in later steps, all pairwise scores are averaged.

Two user adjustable cut-off values are applied to each pairwise match: (1) a score cut-off; and (2) an overlap cut-off. The score cut-off is necessary to separate significant scores from spurious matches. We normally use a score cut-off of 50 bits. The effect of this cut-off is mainly to avoid inclusion of insignificant hits and thereby reduce the volume of data. The choice of this cut-off has virtually no

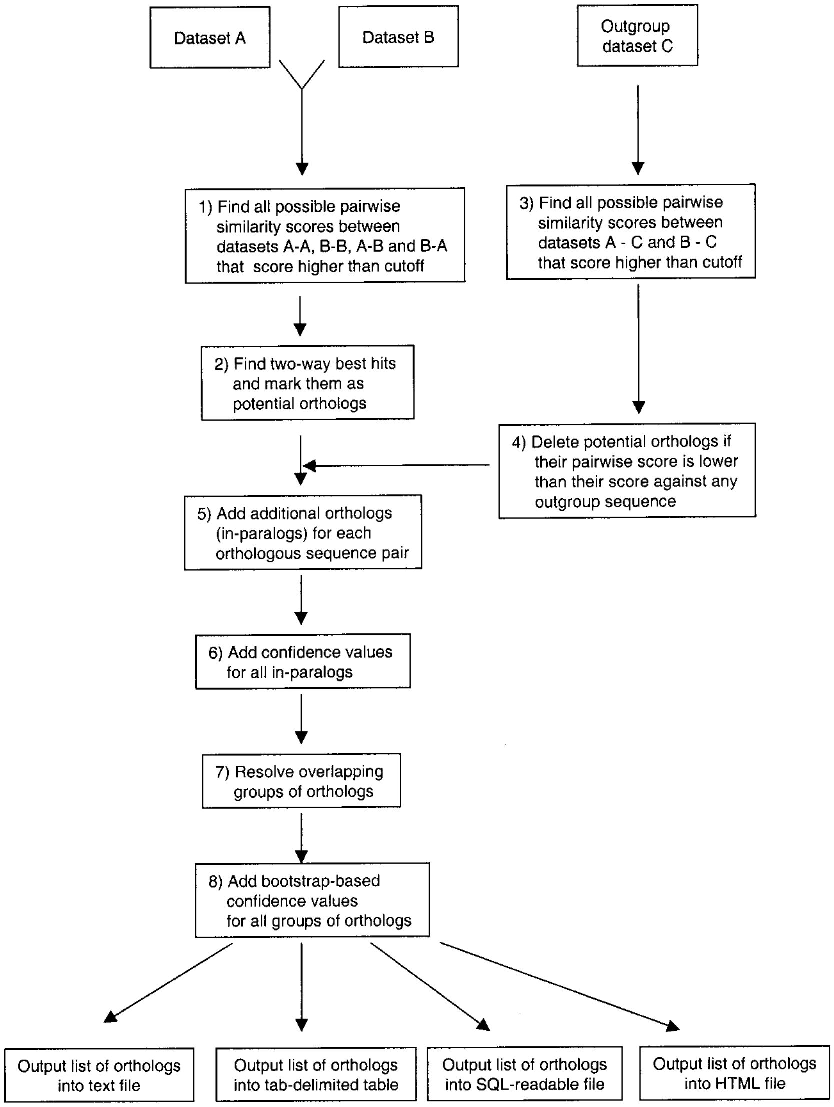

Dataset A
Dataset B
Outgroup dataset C
1) Find all possible pairwise similarity scores between datasets A-A, B-B, A-B and B-A that score higher than cutoff
3) Find all possible pairwise similarity scores between datasets A - C and B - C that score higher than cutoff
2) Find two-way best hits and mark them as potential orthologs
4) Delete potential orthologs if their pairwise score is lower than their score against any outgroup sequence
5) Add additional orthologs (in-paralogs) for each orthologous sequence pair
6) Add confidence values for all in-paralogs
7) Resolve overlapping groups of orthologs
8) Add bootstrap-based confidence values for all groups of orthologs
Output list of orthologs into text file
Output list of orthologs into tab-delimited table
Output list of orthologs into SQL-readable file
Output list of orthologs into HTML file

Figure 1. Overview of the INPARANOID algorithm. The program requires two fasta format sequence files A and B with protein sequences. All-*versus*-all BLAST search is run (1) and sequence pairs with mutually best hits are detected (2). Sequences from out-group species are optionally used to detect cases of selective loss of orthologs. The A-B sequence pairs are eliminated if either sequence A or sequence B scores higher to out-group sequence than they score to each other (3,4). Additional orthologs (in-paralogs) are clustered together with each remaining pair of potential orthologs as shown in 5Figure 2 (5,6). Overlapping clusters are resolved by a set of rules (7) shown in Figure 3. Finally, the bootstrapping technique is used to estimate the probability that a given pair of orthologs had mutual best score only by chance (8). The bootstrapping step is optional.

effect on the clustering of sequences that score above the cut-off. The overlap cut-off is applied to avoid short, domain-level matches. Orthologous sequences are expected to maintain the homology over the majority of their length. We therefore apply an overlap cut-off of 50 %, i.e. the matching segment of the longer sequence must exceed 50 % of its total length.

We also tried to use a cut-off for accepting multiple best hits within a ''grey zone'', in order to avoid selecting the best pairwise match when alternative orthologs exist that are almost as strong. However, we found that, although this cut-off reduced the number of false negatives, it resulted in more false positives, and hence we do not apply it.

Clustering algorithm

The purpose of the ortholog detection algorithm is to find non-overlapping groups of orthologous sequences using pairwise similarity scores. This is essentially a sequence clustering problem. The ortholog detection starts with finding mutually best scoring sequence pairs, bi-directionally best hits between datasets A and B. These mutually best hits are marked as the main ortholog pair of a given ortholog group. The main ortholog pairs serve as central points around which additional orthologs (in-paralogs) from both species will be clustered in later steps. The detection of additional orthologs is done independently for each ortholog group, starting with the pair with highest sequence similarity and continuing until the pair with lowest sequence similarity within the limits defined by the cut-off value.

Additional orthologs are then clustered around the main ortholog in each species separately. The basic assumption is that sequences from the same species that are more similar to the main ortholog than to any sequence from other species are in-paralogs belonging to the same group of orthologs. This principle is explained graphically in Figure 2. The program runs through all main ortholog pairs and adds in-paralogs from datasets A and B. In the case of overlap between two groups, the overlapping groups are merged, deleted, or separated depending on the type and extent of overlap (Figure 3). The rules are applied in the following order: (1) merge groups if main orthologs A_2 and B_2 are already clustered in a stronger group A_1-B_1; (2) merge groups if main ortholog B has equally best hit to two orthologs from species A, A_1 and A_2; (3) delete new group if one of the main orthologs A_2 already belongs to a much stronger group; (4) merge groups if one of the main orthologs already has a high confidence value in another group.

The clustering approach based on BLAST scores implicitly assumes equal evolutionary rate of all paralogs. This is an approximation that might cause incorrect results if the real rates of evolution vary significantly between paralogs.

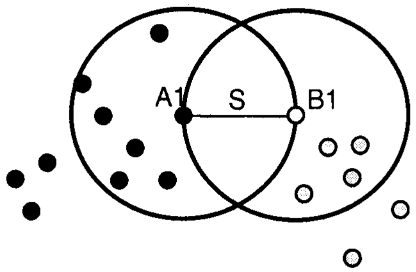


Figure 2. Clustering of additional orthologs (in-paralogs). Each circle represents a sequence from species A (black) or species B (grey). Main orthologs (pairs with mutually best hit) are denoted A_1 and B_1. Their similarity score is shown as S. The score should be thought of as reverse distance between A_1 and B_1, higher score corresponding to shorter distance. The main assumption for clustering of in-paralogs is that the main ortholog is more similar to in-paralogs from the same species than to any sequence from other species. On this graph it means that all in-paralogs with score S or better to the main ortholog are inside the circle with diameter S that is drawn around the main ortholog. Sequences outside the circle are classified as out-paralogs. In-paralogs from both species A and B are clustered independently.

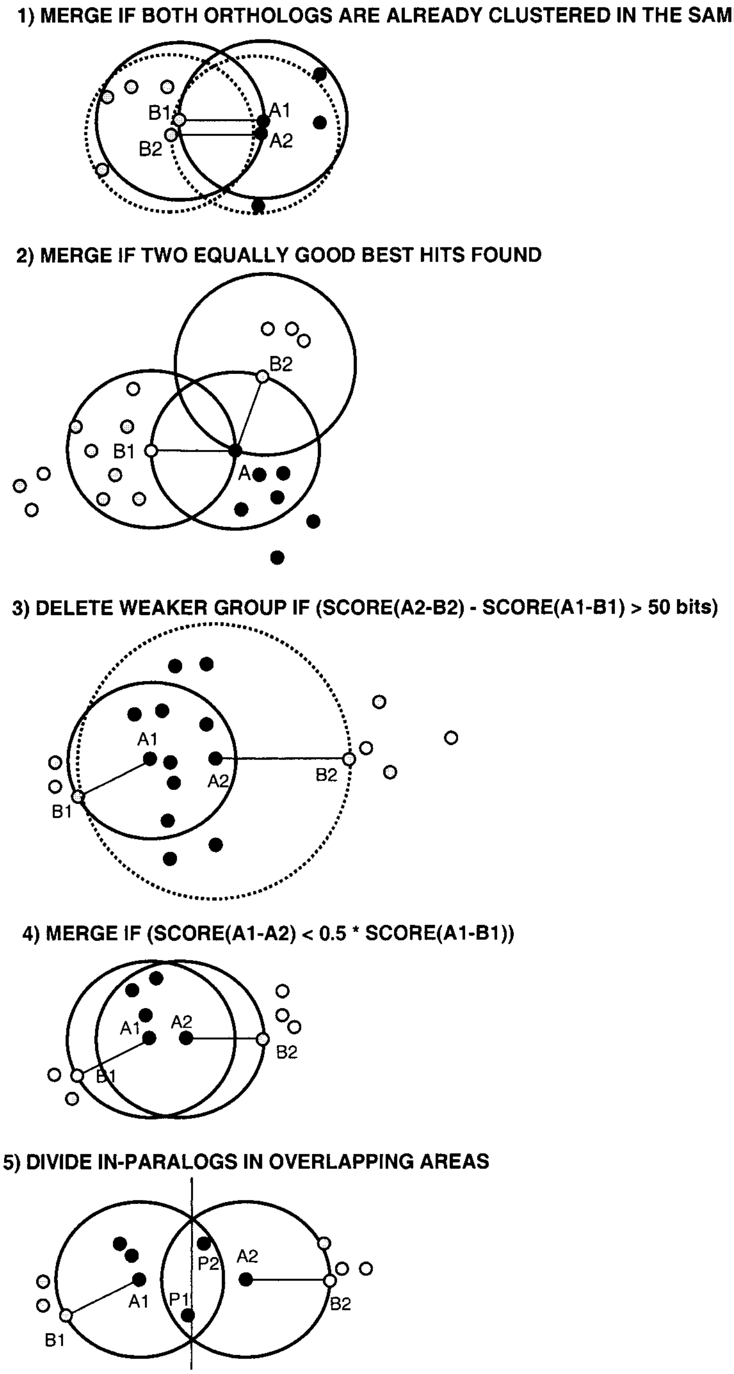
Confidence values for in-paralogs

In the case of one-to-many or many-to-many types of orthology, several in-paralogs form a cluster in which all proteins are orthologous to one or many proteins in the other species. Although all in-paralogs are considered orthologs, some may be very similar to the main ortholog, while others may be so dissimilar that they are nearly excluded from the group. We wanted to characterize this feature of in-paralogs quantitatively, assigning them a confidence value that shows ''how orthologous'' a given sequence is. The confidence value simply shows how far a given sequence is from the main ortholog of the same species on a scale between 0 % and 100 % (Figure 4). On this scale, 100 % is assigned to the main ortholog and 0 % is assigned to a sequence with the minimum similarity score required to be marked as in-paralog of a given group. A general formula to calculate this confidence value is:

$$\text{Confidence for } A_\text{p} = 100\% \times (\text{score}AA_\text{p} - \text{score}AB)/(\text{score}AA - \text{score}AB)$$

$$\text{Confidence for } B_\text{p} = 100\% \times (\text{score}BB_\text{p} - \text{score}AB)/(\text{score}BB - \text{score}AB)$$

where, A_p is an in-paralog from dataset A, B_p is an in-paralog from dataset B, A is the main ortholog from dataset A, B is the main ortholog from dataset B, scoreXY is the similarity score between protein X and Y in bits.



1) MERGE IF BOTH ORTHOLOGS ARE ALREADY CLUSTERED IN THE SAME GROUP

2) MERGE IF TWO EQUALLY GOOD BEST HITS FOUND

3) DELETE WEAKER GROUP IF (SCORE(A2-B2) - SCORE(A1-B1) > 50 bits)

4) MERGE IF (SCORE(A1-A2) < 0.5 * SCORE(A1-B1))

5) DIVIDE IN-PARALOGS IN OVERLAPPING AREAS

Figure 3. The rules for resolving overlapping groups of in-paralogs. In-paralogs are clustered in order of their similarity scores, starting with the more similar groups. The rules are applied in the following order: (1) merge groups if main orthologs A_2 and B_2 are already clustered in the same group with a stronger group A_1-B_1; (2) merge groups if main ortholog B has equally best hit to two orthologs from A, A_1 and A_2; (3) delete new group if one of the main orthologs A_2 already belongs to a much stronger group ($S_1 - S_2 > 50$ bits); (4) merge groups if one of the new ortholog candidates already has a high (>50 %) confidence value in another group; (5) all other overlapping groups of in-paralogs are separated based on their distance to the main ortholog. In the given example, the in-paralog P_1 will remain in group with A_1, but the in-paralog P_2 will be moved into the second group with A_2.

If groups of orthologs are merged in the process of clustering, they will retain their original confidence values. Thus, after merging two groups, a group of orthologs can contain more than one member with 100 % confidence.

Bootstrap values for groups of orthologs

In addition to confidence values for in-paralogs, we try to estimate the reliability of each orthologous group itself. If no other sequence competes as a main orthologs we would assign a high confidence value in the assignment, while if the main ortholog is only slightly better than competing sequences we would assign a low confidence value. We calculate a confidence estimate by using the bootstrapping technique. The bootstrap values are calculated by comparing two pairwise sequence alignments. These two alignments are between main ortholog pair A_1B_1 and between an alternative, lower-scoring alignment A_1B_2. The sequence B_2 in this case is the first alternative

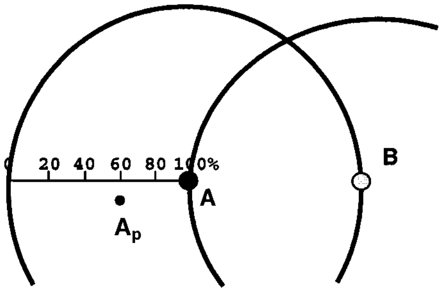


Figure 4. Confidence values are calculated for all in-paralogs. Confidence for in-paralogs is scaled between 0% and 100%, depending on their similarity to the main ortholog. The confidence value for the main ortholog is always 100%. The confidence value for in-paralog $P = 100\,\% \times ((\text{score}AA_{\text{p}} - \text{score}AB)/(\text{score}AA - \text{score}AB))$.

ortholog that is not already clustered in a given group. The columns in alignments between sequences A_1B_1 and A_1B_2 are sampled with replacement, considering an insertion as a single unit. In this way we generate, on average, the same amounts of gaps as in the original alignment. The bootstrap value is expressed as the fraction of sampled alignments that support the hypothesis that the best match to A_1 is B_1, and not B_2. The same procedure is repeated for sequence pairs A_1B_1 and A_2B_1, to test the hypothesis that the best hit to B_1 is A_1, and not A_2. Both bootstrap values are shown in the output file. There is a clear correlation between score difference between two alternative orthologs and the bootstrap value for this pair. In other words, the bootstrap value tends to be lower if there is a closely related alternative ortholog (Figure 5). As this way of calculating bootstrap values from pairwise alignments is rather novel and uncharacterized, we report it, but do not reject any group based on its bootstrap value.

Additional considerations with the results of the BLAST program

Although BLAST is fast and detects biologically relevant homologies reliably, it should be used with caution. The main problem for the presented ortholog detection algorithm is that BLAST reports local similarities. The orthologs are expected to share sequence similarity over the entire length, or at least over the majority of their length. We avoid domain-level matches by forcing the matched area to be longer than 50% of the longer sequence. This should avoid clustering sequences that share only short domains. Additional problems with the BLAST output appear with sequence pairs that have two or three separate regions of sequence similarity. This happens with many sequences whose N terminus and C terminus are conserved, but the conservation in the middle of the sequence is too low to be reported by BLAST. The BLAST output segmentation could be addressed by setting drop-off value -X on command line, which would cause the BLAST program to report longer seg-

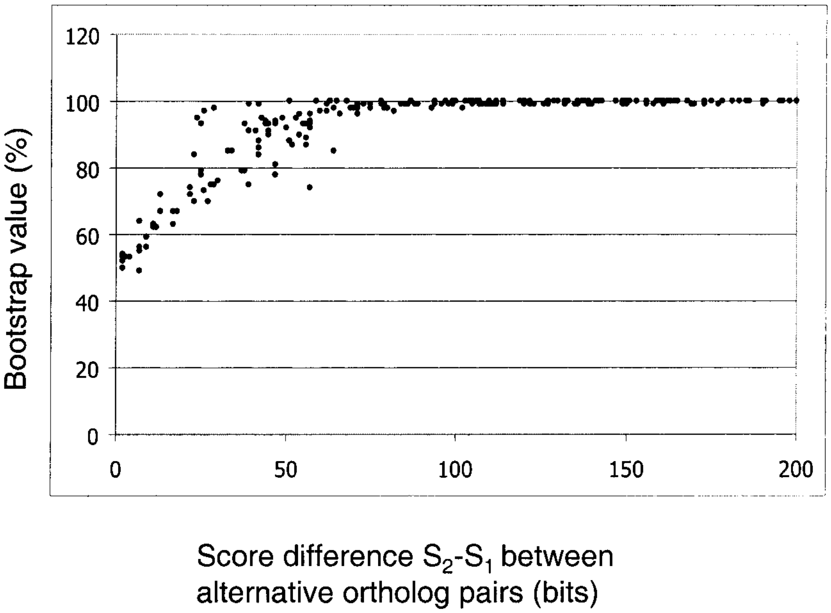


Figure 5. Correlation between bootstrap value and score difference between alternative orthologs. Data from running INPARANOID on the dataset of *C. elegans* and mammalian transmembrane proteins.

ments of similarity. We found it more realistic to ignore the non-conserved areas and sum only the conserved segment scores. The segments are checked for consistency and for overlap, after which the similarity scores of non-overlapping parts are summed.

One difference between phylogenetic methods and the BLAST program is that the BLAST program uses gap penalties. Most of the commonly used phylogenetic methods calculate distances using each column in multiple alignment independently, ignoring columns that contain gaps. The affine gap penalty model used in the BLAST program is certainly more advanced and biologically relevant. However, errors are frequent in current genome databases due to wrong gene predictions. Sometimes an exon is missing in one of the sequences, or an intron is predicted as a coding area. In such cases, BLAST would incur unrealistically high gap penalties and the real relationship between truly orthologous sequences might be missed. In our comparison to the phylogenetically detected orthologs (see Table 1), inflated BLAST gap penalties resulted in failure to detect five orthologs, while one case was missed due to overly segmented alignments in the BLAST output. It may be possible to overcome some of these problems by a more global alignment approach but, as the problems appear to be relatively rare, we consider BLAST to be a well-suited method for the task.

Implementation

The program INPARANOID that finds and reports in-paralogs and orthologs according to the described algorithm was written in PERL. The input to the program are files with pairwise similarity scores from the dataset comparisons, which can be generated by a parser for BLAST2 (BLAST_PARSER) that implements the overlap rules described above. Also included in the INPARANOID package are PERL and JAVA programs to calculate bootstrap values from pairwise alignments (BLAST2FAA.PL and SEQSTAT.JAR).

The output is given in plain text or HTML format as a sorted list of orthologous groups with all member sequences and confidence values for in-paralogs and the bootstrap support for the ortholog group itself. Additional computer-readable output is printed into a separate file in tab-delimited format or into tables that are used as input for a MYSQL database.

The time spent for calculation of orthologs is mainly dependent on speed and number of calculations for pairwise similarity scores. Ortholog detection alone takes about 20 minutes for the full set of 14,100 *D. melanogaster* and 19,105 *C. elegans* proteins on an 800 MHz PC Linux. A critical issue for the program is memory usage, due to the poor memory management of the PERL programming language: 145 MB of RAM was used by the program to calculate orthologs between all proteins from *D. melanogaster* and *C. elegans*.

Results and Discussion

Comparison to curated dataset of orthologs

In order to assess the quality of orthologs produced by INPARANOID, we ran it on a dataset of 5500 worm and mammalian proteins in which orthologs had been assigned by manual analysis of phylogenetic trees.[12] In that study, orthologs were assigned if a majority of nine different phylogenetic methods supported the orthology. This yielded 168 curated orthology assignments. We consider these assignments a trusted set of orthologs and use it as a testbench for quality assessment of other ortholog-finding methods. The INPARANOID results in comparison with the trusted dataset of reference orthologs are summarized in Table 1. A group of orthologs predicted by INPARANOID was counted positive if both main orthologs were found in one group in the set of trusted orthologs.

In general, ortholog groups reported by both methods are rather similar, although INPARANOID tends to report smaller groups or even split the reference groups into two subgroups of orthologs. INPARANOID failed to report less than 3% of the reference orthologs. However, it reported 32 additional ortholog assignments that were not present in the reference set. These may represent

Table 1. A comparison of the INPARANOID performance compared to the reference dataset of phylogenetically derived orthologs[12]

	Phylogenetic orthologs without use of out-group		INPARANOID orthologs without use of out-group	
	No.	%	No.	%
True positives	168	95	162	84
False positives	9	5	18	9
Additional orthologs			14	7
False negatives			6	3
Total	177	100	194	100

In the row Additional orthologs are listed likely ortholog groups that were missed by phylogenetic methods, but reported by INPARANOID. Many of them are partly supported by phylogenetic trees, i.e. supported with low bootstrap or by some but not a majority of the phylogenetic methods. The false positives listed under the reference dataset were detected initially on phylogenetic trees as orthologs, but removed later after additional analysis that indicated selective gene loss or long branch attraction.

either false positives or novel true orthologs. There are cases when the BLAST approach might detect orthologs that were missed by tree methods. The two main reasons for this are large protein families and different treatment of gaps. In general, tree bootstrap values tend to decrease as the number of sequences in the family increases.[13] All tree methods used to generate the trusted set of orthologs do not use gap penalties when calculating distances, which is an inappropriate evolutionary model if many insertions/deletion events have taken place. BLAST, on the other hand, does use gap penalties, so matches with many gaps will get a lower score than equivalent matches with fewer gaps.

Many of the proteins in the testbench dataset have changed since it was compiled, and new proteins have been discovered, hence some of the novel ortholog assignments are not found in the on-line version of INPARANOID, which is based on SWISS-PROT + TREMBL from May 2001. We used persistence from the old to the new dataset as a criterion to consider an additional ortholog group correct, as well as high bootstrap support. After careful analysis of the additional ortholog groups, 14 of them were considered novel ortholog groups with a high degree of confidence (Table 2). The remaining 18 groups with less supporting evidence were considered false positives, caused by shortcomings of the BLAST program. Essentially all of the novel ortholog groups have high bootstrap support and were supported by a minority of the phylogenetic tree methods. Several of these novel groups included G-protein coupled receptors (GPCRs). We believe that they represent very likely orthologous sequences that were not detected by phylogenetic methods because of prohibitively large family sizes or because gaps are not penalized.

The effect of long branch attraction and out-group for orthology detection

We compared the ability of INPARANOID to avoid false positive orthologous groups that had initially been detected by phylogenetic methods. These false positives appear among phylogenetic orthologs mainly for two reasons. First, many phylogenetic methods are sensitive to "long branch attraction". This is a known artifact of distance-based methods where sequences are clustered together, not due to their similarity but rather due to their common dissimilarity to other sequences. Careful analysis of the original dataset from Remm & Sonnhammer[12] revealed that it contained four cases of long branch attraction. INPARANOID did not detect any of these cases. This is not surprising, because sequences related by long branch attraction typically have a very low similarity score.

The other reason for false positives in phylogenetic methods is selective loss of orthologs in one or the other phylogenetic lineage. This can make out-paralogs look like orthologs. A similar effect can be observed in cases where only partially sequenced genomes are analyzed. These false positives can be detected by adding out-groups: i.e. sequences from species that are expected to be evolutionarily more distant. Such sequences will stay between the false orthologs in the tree and can thus indicate the selective loss of orthologs. To cope with the problem, INPARANOID was written to include information from an out-group species. An important criterion for orthology detection is that no sequence from an out-group should be closer to the orthologs than they are to each other. Based on that principle, INPARANOID can optionally reject orthologs that have significantly higher scoring match with an out-group sequence.

We detected five cases of selective gene loss in the original worm/mammalian ortholog set, using plant and lower species as out-group. INPARANOID reported such false positives at a lower rate than phylogenetic methods. Using either *Saccharomyces cerevisiae* or *Arabidopsis thaliana* as out-group resulted in three assignments of selective loss of orthologs. Two of these three assignments were the same sequences for both out-groups and agreed with the phylogenetic analysis, but the third assignment, which was different depending on the out-group used, was incorrect. Such false-negative assignments of selective loss can be caused by BLAST artifacts or unequal rates of evolution. Using an out-group is thus useful for removing false ortholog assignments due to selective gene loss, but there is an inherent risk of removing *bona fide* orthologs.

To estimate the frequency of selective loss of ortholog assignments by INPARANOID using different out-group species, we calculated orthologs from the fully sequenced genomes of *C. elegans* and *D. melanogaster* with different species as an out-group. In this worm-fly ortholog analysis, the total number of detected orthologs was 3331. From these, 15 (0.5%) were rejected if *E. coli* was used as an out-group, 134 (4.0%) were rejected if *S. cerevisiae* was used as an out-group, and 276 (8.3%) were rejected if *A. thaliana* was used as an out-group.

Comparison with alternative automatic approaches

Most orthology detection approaches simply identify mutually best matches. We believe that such an approach is too limited for eukaryotic genomes, and that it is important to identify additional orthologs (in-paralogs). One approach that does include paralogs is the COG system, which has much in common with our method. However, in the final COGs, no distinction is made between in-paralogs and out-paralogs. The main reason for this is that the COG database strives to flatly group orthologs from all species together, while our approach considers only two species or lineages at the time. In fact, a COG must consist of at least three species. Sequences with unidirectional

Table 2. Novel ortholog assignments made by INPARANOID in a dataset of membrane proteins from worm and mammals

Group	*C. elegans* ortholog(s)	Bootstrap value (%)	Human ortholog(s)	Bootstrap value (%)	Total family size	Number of proteins in tree
1	C16D6.2 (O62062) C53C7.1 (Q9XXU4)	61	GPRA_HUMAN O75194	92	1550	72
2	F41E7.3 (Q20275)	55	NY2R_HUMAN	78	1550	72
3	C52B11.3 (YYI3_CAEEL)	65	A1AA_HUMAN A1AB_HUMAN A1AD_HUMAN O60451 Q9UD67 Q13675 Q13729 Q9UD63 Q9H1N4	90	1550	101
4	F14D12.6 (Q19449)	72	A2AA_HUMAN A2AB_HUMAN A2AC_HUMAN A2AD_HUMAN Q9HB49	73	1550	101
5	F54D7.3 (O44731)	93	GRHR_HUMAN Q92644 O75793	80	1550	34
6	F59C12.2 (Q21034)	77	5H2A_HUMAN 5H2C_HUMAN 5H2B_HUMAN Q9P2Q9	91	1550	101
7	R106.2 (Q23033)	85	SSR2_HUMAN SSR5_HUMAN SSR3_HUMAN SSR1_HUMAN SSR4_HUMAN OPRM_HUMAN GPR8_HUMAN OPRK_HUMAN OPRD_HUMAN GPR7_HUMAN OPRX_HUMAN Q9UIY1 Q9H573	87	1550	101
8	Q9U990 Q23074 T10G3.7 (Q9TW41) B0207.12 (O01436) F11A5.10 (O17793) F25F8.2 (Q17328)	78	GRA2_HUMAN GRA3_HUMAN GRA1_HUMAN GRB_HUMAN	98	222	72
9	ZC482.1 (O18276)	72	GAB3_HUMAN Q16323 GAB2_HUMAN GAB1_HUMAN	72	222	72
10	T23D8.2 (Q9XVI4)	98	CD63_HUMAN	93	80	71
11	R04F11.4 (Q21729) C24H11.8 (Q9XVD1)	49	CIW4_HUMAN CIWA_HUMAN CIW2_HUMAN CIW5_HUMAN Q9NRT2 Q9H591	100	67	65
12	F14F4.3 (O62170)	99	MRP5_HUMAN (O14517)	99	249	30
14	K10D3.1 (Q21415)	59	GLK1_HUMAN GLK2_HUMAN GLK3_HUMAN GLK4_HUMAN GLK5_HUMAN	99	90	47

These groups were not detected by phylogenetic tree methods in our previous work[12] mainly due to large family sizes. The size of the families and subfamilies from which the trees were constructed are shown to the right. The INPARANOID analysis was done using all available mammalian proteins, but only the human proteins are shown here.

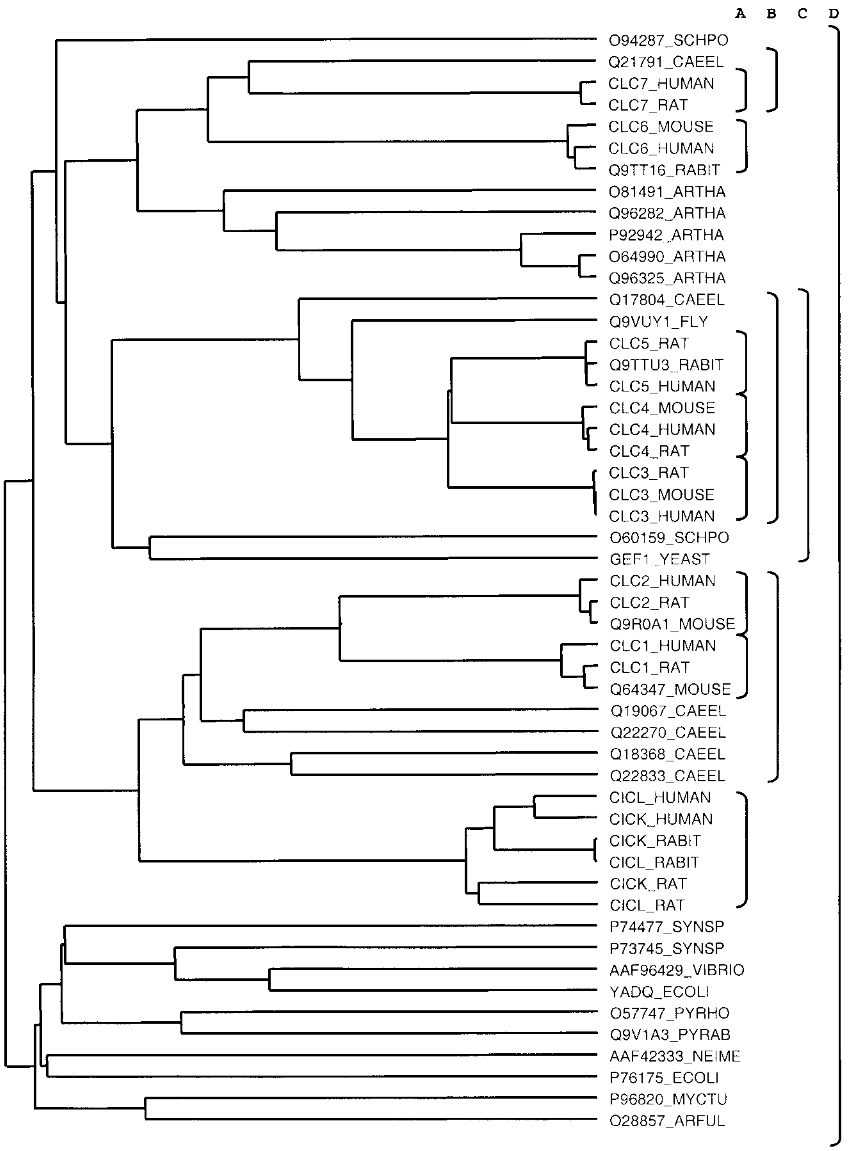


Figure 6. UPGMA tree of chloride channel proteins. The number of orthologous groups between a pair of species increases with decreasing evolutionary distance between them. There are eight groups of orthologs between human and other mammalia (A), three groups between human and *C. elegans* (B), one group between human and yeast (C), and one group of orthologs between any prokaryotic and eukaryotic species (D).

best hits to members of a COG are added later, representing potential in-paralogs. However, because the orthology in a COG is not defined to a particular evolutionary point, both in-paralogs and out-paralogs may be added.

As an example, COG0050 from the COG database† includes tufA/tufB (SWISS-PROT EFTU_ECOLI) from *E. coli* and YOR187w (SWISS-PROT EFTU_YEAST) from *S. cerevisiae*:. These are bacterial elongation factor TU (EF-TU) and its mitochondrial counterpart in *S. cerevisiae*. However, six other sequences from *S. cerevisiae* are included in this COG, all of which are out-paralogs relative to the tufA/B-YOR187w orthology, and hence are not joined to them by INPARANOID. The six other yeast sequences in COG00050 are annotated as elongation factor 1-alpha (YPR080W and YBR118W; EF1A_YEAST), EF1A-like (YKR084C; HBS1_YEAST), peptide chain release factor (YDR172W; ERF2_YEAST), translation initiation factor 2-gamma (YER025W; IF2G_YEAST), and unannotated (YOR076C; Q08491). Although some of these sequences perform functions related to the mitochondrial EF-TU, none of them can be considered orthologs to bacterial EF-TU, neither from an evolutionary nor from a functional point of view. There are thus several good reasons not to cluster all these sequences together as putative orthologs.

† http://www.ncbi.nlm.nih.gov/cgi-bin/COG/palox?COG0050

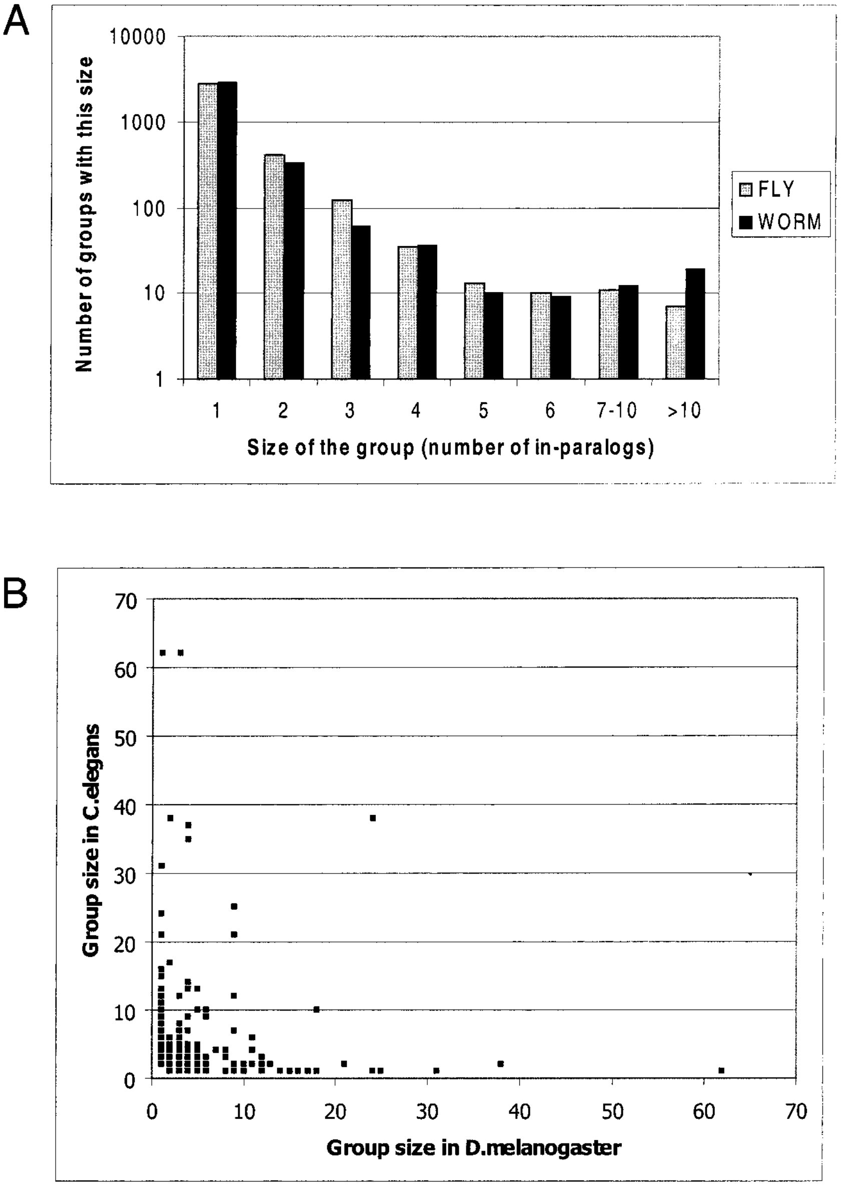


Figure 7. (a) The size distribution of orthologous groups in worm and fly. Most of the orthologous groups in both worm and fly contain only one or few in-paralogs; large in-paralog families are rare in both genomes. (b) The number of in-paralogs from worm and fly in each group is poorly correlated ($r^2 = 0.08$). This indicates that evolution has led to expansion of different gene families in these two genomes.

In contrast to COGs, our approach is limited explicitly to two species or lineages only, which allows us to define the evolutionary point of the orthology precisely, and to separate in-paralogs from out-paralogs. Defining the evolutionary point of the orthology is important, because the number of branches increases during evolution due to sequence duplication. As a result, the number of orthologous groups between closely related species is expected to be greater than the number of orthologous groups between distantly related species. As an illustration of this, see the tree of chloride channel proteins in Figure 6. There are eight groups of orthologs between human and other mammalia (A), three groups between human and *C. elegans* (B), one group between human and yeast (C), and one group between any prokaryotic and eukaryotic species (D). Any of these different lists of orthologous groups might be useful for the biologist, depending on the purpose of the study. That is the reason why we prefer to limit INPARANOID to the comparison of two species or lineages and do not attempt to create flat groups of orthologs covering many lineages or the whole tree of life. Groups of orthologs from more than two species can still achieved by considering a lineage of multiple species as a "superspecies" in INPARANOID.

Example of full genomic dataset: worm-fly orthologs

To test the performance of the INPARANOID program on the whole-genome scale, we applied it to identify orthologs from the complete set of protein sequences from *C. elegans* and *D. melanogaster* genomes. The *C. elegans* genome (wormpep98) contains 19,099 proteins and the *D. melanogaster* genome 14,100. INPARANOID detected 3331 groups of orthologs, comprising 4451 worm and

4366 fly sequences. The distribution of orthologous group sizes is similar in both species, see Figure 7(a). The largest group of orthologs contains 34 worm sequences and 26 fly sequences (a subfamily of UDP-glucuronosyltransferases). However, the families that have expanded are different in worm and fly. If one plots the number of in-paralogs in one organism against the number in the other organism for all groups (Figure 7(b)), they do not appear correlated ($r^2 = 0.08$). In all, 72 % of the groups between fly and worm represent straightforward one-to-one orthology, 23 % of the groups have a one-to-many type of relationship and only 5 % of the groups have a many-to-many type of orthology. This illustrates the fact that different living conditions of arthropods and nematodes have led to expansions of different protein families.

Acknowledgements

This work was supported by a grant from the Pharmacia Corporation (Kalamazoo, MI).

References

1. Fitch, W. M. (1970). Distinguishing homologous from analogous proteins. *Syst. Zool.* **19**, 99-113.
2. Yuan, Y. P., Eulenstein, O., Vingron, M. & Bork, P. (1998). Towards detection of orthologues in sequence databases. *Bioinformatics,* **14**, 285-289.
3. Chervitz, S. A., Aravind, L., Sherlock, G., Ball, C. A., Koonin, E. V., Dwight, S. S. *et al*. (1998). Comparison of the complete protein sets of worm and yeast: orthology and divergence. *Science,* **282**, 2022-2028.
4. Rubin, G. M., Yandell, M. D., Wortman, J. R., Gabor, Miklos G. L., Nelson, C. R., Hariharan, I. K. *et al*. (2000). Comparative genomics of the eukaryotes. *Science,* **287**, 2204-2215.
5. Wheelan, S. J., Boguski, M. S., Duret, L. & Makalowski, W. (1999). Human and nematode orthologs - lessons from the analysis of 1800 human genes and the proteome of *Caenorhabditis elegans*. *Gene,* **238**, 163-170.
6. Mushegian, A. R., Garey, J. R., Martin, J. & Liu, L. X. (1998). Large-scale taxonomic profiling of eukaryotic model organisms: a comparison of orthologous proteins encoded by the human, fly, nematode, and yeast genomes. *Genome Res.* **8**, 590-598.
7. Tatusov, R. L., Mushegian, A. R., Bork, P., Brown, N. P., Hayes, W. S., Borodovsky, M. *et al*. (1996). Metabolism and evolution of Haemophilus influenzae deduced from a whole-genome comparison with *Escherichia coli*. *Curr. Biol.* **6**, 279-291.
8. Tatusov, R. L., Koonin, E. V. & Lipman, D. J. (1997). A genomic perspective on protein families. *Science,* **278**, 631-637.
9. Tatusov, R. L., Natale, D. A., Garkavtsev, I. V., Tatusova, T. A., Shankavaram, U. T., Rao, B. S. *et al*. (2001). The COG database: new developments in phylogenetic classification of proteins from complete genomes. *Nucl. Acids Res.* **29**, 22-28.
10. Tatusov, R. L., Galperin, M. Y., Natale, D. A. & Koonin, E. V. (2000). The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucl. Acids Res.* **28**, 33-36.
11. Makalowski, W. & Boguski, M. S. (1998). Evolutionary parameters of the transcribed mammalian genome: an analysis of 2820 orthologous rodent and human sequences. *Proc. Natl Acad. Sci. USA,* **95**, 9407-9412.
12. Remm, M. & Sonnhammer, E. L. L. (2000). Classification of transmembrane protein families in the *Caenorhabditis elegans* genome and identification of human orthologs. *Genome Res.* **10**, 1679-1689.
13. Zharkikh, A. & Li, W. H. (1995). Estimation of confidence in phylogeny: the complete-and-partial bootstrap techniques. *Mol. Phylogenet. Evol.* **4**, 44-63.

Edited by F. Cohen

(Received 4 July 2001; received in revised form 12 October 2001; accepted 12 October 2001)