Computational Methods for Remote Homolog Identification

Article *in* Current Protein and Peptide Science · January 2006

DOI: 10.2174/138920305774933231 · Source: PubMed

2 authors:

Xiu-Feng Wan
Mississippi State University
181 PUBLICATIONS 2,824 CITATIONS
SEE PROFILE

Dong Xu
University of Missouri
623 PUBLICATIONS 14,184 CITATIONS
SEE PROFILE

Computational Methods for Remote Homolog Identification

Xiu-Feng Wan[†] and Dong Xu[*]

Digital Biology Laboratory, Department of Computer Science, University of Missouri - Columbia, Columbia, MO 65211, USA

Abstract: As more and more protein sequences are available, homolog identification becomes increasingly important for functional, structural, and evolutional studies of proteins. Many homologous proteins were separated a very long time ago in their evolutionary history and thus their sequences share low sequence identity. These remote homologs have become a research focus in bioinformatics over the past decade, and some significant advances have been achieved. In this paper, we provide a comprehensive review on computational techniques used in remote homolog identification based on different methods, including sequence-sequence comparison, and sequence-structure comparison, and structure-structure comparison. Other miscellaneous approaches are also summarized. Pointers to the online resources of these methods and their related databases are provided. Comparisons among different methods in terms of their technical approaches, their strengths, and limitations are followed. Studies on proteins in SARS-CoV are shown as an example for remote homolog identification application.

Keywords: Remote homolog, homolog identification, evolution, sequence analysis, sequence profile, threading.

1. INTRODUCTION

1.1 Why Remote Homolog is Important

Homolog identification is becoming more and more important in modern biology. Traditional biological studies have been focused extensively on model systems, and these studies provide tremendous resources to investigate other species. The most used model systems include *E. coli,* budding yeast (*Saccharomyces cerevisiae*), fission yeast (*Schizosaccharomyces pombe*), *Caenorhabditis elegans*, *Drosophila melanogaster*, zebrafish (*Danio rerio), Arabidopsis thaliana*, and mouse (*Mus musculus*) [1]. Most of the biological knowledge that has been accumulated so far is related to these model organisms. A convenient way to study the functions and structures of a new gene is to identify homologs (evolutionary relationships) in model organisms, from which one can infer structure, function and mechanism of the new gene. Such an approach becomes very popular nowadays, given the surge in biological sequence data due to the breakthroughs in large-scale sequencing technologies and various genome projects.

Homolog identification can be conducted through computationally matching a query sequence to similar sequences in the database. However, this matching process is not trivial since two homologous proteins could have been separated a very long time ago in their evolutionary history and thus their evolutionary relationship may be very difficult to detect. Such distantly related proteins are called remote homologs. A large proportion, typically 30-40% of the predicted protein coding genes do not have specific function assignments since we cannot relate these proteins to any protein with known function in the database. This is the case even in well-studied model organisms. For example, at present, 2,280 genes out of 6,324 genes in budding yeast *S. cerevisiae* (The Gene Ontology Consortium, 2000; http://www.geneontology.org) have not been annotated with any functions. Many of these "unknown" genes probably have remote homologs whose functions are well characterized. The existing methods for detecting homology relationships via sequence similarity might fail since their amino acid sequences have diverged so much.

2.1 Homology and Evolution

When two genes are evolutionarily related (i.e., descended from a common ancestor), they are called homologs. A homolog may be categorized into ortholog, paralog or xenolog based on different evolutionary events (Fig. **1A**) [2]. **Orthologs** are homologs that diverged after speciation events, **paralogs** are homologs that diverged after gene duplication events, and **xenologs** are homologs that diverged after lateral (horizontal) gene transfer events [3]. Although homologs only indicate their evolutionary relationship and may not have the same function, the homolog categorization may facilitate function prediction. For instance, orthologs typically preserve the gene functions. In contrast, paralogs and xenologs often have related/overlapping but no identical biological functions because gene duplications and horizontal transfer are frequently accompanied by functional divergence [4,5]. In general, homologs have similar protein structures, but vice versa are not true. Proteins sharing similar structures but without detectable evolutionary/functional relationship are called **analogs**.

The evolutionary relationship between genes is complicated. The evolutionary connection can go beyond the one-

[*] Address correspondence to this author at the 201 Engineering Building West, Department of Computer Science, University of Missouri, Columbia, Missouri, MO 65211, USA; Tel: 1-573-882-7064; Fax: 1-573-882-8318; Email:xudong@missouri.edu.

[†] *Current address: Department of Microbiology, Miami University, Oxford, OH 45056, USA; E-mail: wanx@muohio.edu*

to-one relationship and have mixed homology. As shown in (Fig. **1B**), a new protein may have multiple domains sharing homology to the domains of different proteins. This phenomenon happens frequently when domains are duplicated, inserted, or permuted [6]. They can make the remote homology detection even more challenging and lead to misleading computational results. Some computer packages were specially designed to handle such cases, e.g., DIVCLUS, which detects families of duplication modules from a protein sequence database [7]. Some databases use domains as a basis to study the homologous relationship between proteins, e.g. PRODOM [8].

1.3 Challenges and Progresses in Remote Homolog Identification

A major challenge for computational identification of remote homolog is the low signal-to-noise ratio. Since remote homologs were separated through a long evolutionary history, similarity due to convergence is generally limited to small regions of genes (signal) and other parts of sequences were diverged too much to have any relationship (noise). When aligning whole sequences or even just domains, most parts of the sequences are forced to align together so that the noise buries the signal. Hence, remote homologs typically cannot be identified through straightforward pairwise sequence comparison using tools such as BLAST [9] or FASTA [10].

In the past ten years or so, active research has been carried out for remote homolog identification, and some significant advances have been achieved. Although most of the available methods utilize the information contained within the alignments of multiple closely related sequences (sequence profiles), protein structure prediction and structure-structure comparison also became a useful tool for identification of remote homologs as the protein three-dimensional structure is more conserved than sequence [11,12]. Mutation, insertion or deletion of residues in the sequence often still maintains the structure and function of a protein (e.g. an enzyme domain may have a similar structure but diverse sequences). In general, methods for remote homolog identification can be grouped into four categories based on signal retrieval techniques: (1) sequence-sequence comparison, (2) sequence-structure comparison, (3) structure-structure comparison, and (4) other methods that do not depend on protein sequence or structure (e.g., using gene location in the genomic sequence or microarray data).

Here, we provide a comprehensive review for currently available remote homolog identification methods based on these four categories. Section 2 introduces the methods based on protein sequence-sequence comparison. Section 3 focuses on the methods using sequence-structure comparison, i.e., the threading method. Section 4 discusses other computational methods that do not rely on protein sequence for remote homolog identification. The comparisons among different methods will be summarized in section 5. Section 6 illustrates the remote homolog identification by using the example of SARS-CoV. This paper ends with some discussions in Section 7. Box **1** collects some frequently used resources for remote homolog identification. The tools and databases listed here are by no means comprehensive; especially since many new tools/databases are being generated. The collection only provides a sampling for the types of resources that a user may find.

2. SEQUENCE-SEQUENCE COMPARISON METHODS

The major methods for remote homolog identification are based on sequence-sequence comparison since a protein sequence encodes all the information related to its structure and function. Comparing protein sequences through pairwise alignments often provides the most straightforward relationship between the proteins. Sophisticated methods in identifying remote homologs, such as threading, are often based on the sequence profile resulted from searching a query sequence against a sequence database. The comparison of the advantages and disadvantages of these methods are shown in Table **1**.

2.1 Pairwise Sequence Comparison

Using dynamic programming [13,14], pairwise sequence alignment can retrieve the optimal match between two query sequences based on a selected substitution matrix, which defines the weight for substitution of the amino acid type *i* with *j*. The substitution matrix is essential for the pairwise sequence alignment to identify the true homolog. During the past several decades, many substitution matrices have been

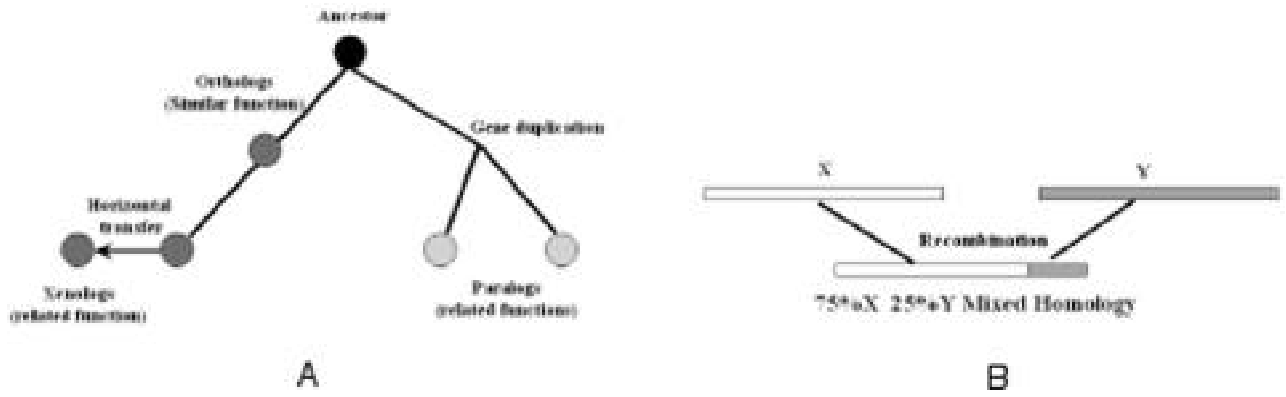

Fig. (1). Evolutionary relationship among proteins. A. Gene homologs; B. Gene recombination.

Box 1.   Remote Homolog Identification Resources

Tool or Database Name	Functionality summary	URL	Reference
GENERAL DATABASE			
Comprehensive Microbial Repository (CMR)	A comprehensive microbial resource	http://www.tigr.org/tigr-scripts/CMR2/CMRHomePage.spl	175
DDBJ	DNA Data Bank of Japan	http://www.ddbj.nig.ac.jp/	176
Evolutionary Lineage Inferred from Structural Analysis (ELISA)	An online database that combines functional annotation with structure and sequence homology modeling to place proteins into sequence-structure-function "neighborhoods"	http://romi.bu.edu/elisa	177
GeneCards	A database of human genes, their products and their involvement in diseases.	http://bioinfo.weizmann.ac.il/cards/	178
Gene Ontology (GO) Consortium	GO provides three structured networks of defined terms to describe gene product attributes, i.e., biological process, molecular function and cellular component	http://www.geneontology.org	179
Genome Channel	A genome annotation and visualization resource	http://compbio.ornl.gov/channel/	180
KEGG	A suite of databases and associated software, integrating information related to molecular interaction networks, pathways, and gene function	http://www.genome.ad.jp/kegg/	181
Mouse Genome Informatics (MGI) database	Provides integrated access to data on the genetics, genomics, and biology of the laboratory mouse.	http://mouseblast.informatics.jax.org/	182
OWL	non-redundant composite of 4 publicly-available primary sources: *SWISS-PROT, PIR (1-3), GenBank* (translation) and *NRL-3D*.	http://umber.sbs.man.ac.uk/dbbrowser/OWL/	183
PIR	a comprehensive database for protein sequence searching and protein classification	http://pir.georgetown.edu	184
Plasmodium database	A plasmodium genome resource	http://plasmodb.org/	185
PRF	A protein sequence database	http://www.genome.ad.jp/htbin/www_bfind?prf	181
PDB	The single worldwide repository for the processing and distribution of 3D biological macromolecular structure data.	http://www.rcsb.org/pdb/	82
STACK	Comprehensive representation of the human expressed genes	http://www.sanbi.ac.za/Dbases.html	186
SwissProt	A curated protein sequence database with extensive information collected from the literature	http://www.expasy.ch/sprot/sprot-top.html	**Error! Bookmark not defined.**
SYSTERS	Large-scale protein clustering based on sequence similarity	http://systers.molgen.mpg.de/	187
TAIR	An Arabidopsis resource	http://www.arabidopsis.org/	188
TIGR Gene Indices	A comprehensive resource for various genomes	http://www.tigr.org/tdb/tgi/	189
UniGene	automatically partitioning sequences into a non-redundant set of gene-oriented clusters.	http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene	190
Worm database	Genomic information related to C. elegans	http://www.wormbase.org	191
Yeast database	molecular biology and genetics of the yeast *Saccharomyces cerevisiae*	http://www.yeastgenome.org/	192
PROTEIN FAMILY CLASSIFICATION			
AARSDB	An aminoacyl-tRNA synthetases database	http://rose.man.poznan.pl/aars/	193
AraC/XylS database	A database on a family of helix-turn-helix transcription factors from bacteria	http://www.eez.csic.es/arac-xyls/	194
ASPD	A curated database of selected proteins and peptides from randomized pools.	http://wwwmgs.bionet.nsc.ru/mgs/gnw/aspd/	195
BLOCKS	The highly conserved regions in groups of proteins represented in the PROSITE database	http://www.psc.edu/general/software/packages/blocks/blocks.html	99
CATH	Protein structure classification	http://www.biochem.ucl.ac.uk/bsm/cath/	83
COG	Phylogenetic classification of proteins encoded in complete genomes	http://www.ncbi.nlm.nih.gov/COG/	196
CSDBase	An interactive database for cold shock domain	http://www.chemie.uni-marburg.de/~csdbase/	197
DIVCLUS	**A protein sequence domain clustering program**	http://www.mrc-lmb.cam.ac.uk/ genomes/divclus_home.html	7
DOMO	A protein domain database	http://www.infobiogen.fr/services/domo/	198
EF-hand CaBP	An collection of EF-hand calcium-binding proteins	http://structbio.vanderbilt.edu/cabp_database/	199

(Box 1) contd….

Tool or Database Name	Functionality summary	URL	Reference
ENZYME	A repository of information relative to the nomenclature of enzymes	http://us.expasy.org/enzyme/	200
FSSP	Database of families of structurally similar proteins	ftp://ftp.embl-heidelberg.de/pub/databases/protein_extras/fssp/	84
GPCRDB (receptors)	An information system for G protein-coupled receptors	http://www.gpcr.org/7tm/	201
Homeobox Page	A collection of information relevant to homeobox genes	http://www.biosci.ki.se/groups/tbu/homeo.html	202
InterPro	An integrated resource of protein families, domains, and functional sites created to handle the data from various protein family sites such as PROSITE, Pfam, PRINTS, ProDom, SMART and TIGRFAMs into a single, comprehensive resource	http://www.ebi.ac.uk/interpro/	203
MEROPS (peptidase)	An information resource for peptidases	http://merops.sanger.ac.uk/	204
MHCPEP (peptides)	A database of MHC binding peptides	http://wehih.wehi.edu.au/mhcpep/	205
Nuclear Protein Database (NPD)	A searchable database of information on proteins that are localised to the nucleus of vertebrate cells	http://npd.hgu.mrc.ac.uk/	206
O-GlycBase	Collections of experimentally verified O- or C-glycosylation site	http://www.cbs.dtu.dk/databases/OGLYCBASE/	207
Pfam	A large collection of multiple sequence alignments and hidden Markov models covering many common protein domains and families.	http://pfam.wustl.edu/	208
Protein Kinase Resource (PKR)	A resource for the protein kinase family of enzymes	http://pkr.sdsc.edu/	209
RNase P	A compilation of RNase P sequences	http://www.mbio.ncsu.edu/RNaseP/home.html	210
PRINTS	A compendium of protein fingerprints	http://umber.sbs.man.ac.uk/dbbrowser/PRINTS/	112, 113
PROCAT	A database of 3D enzyme active site templates	http://www.biochem.ucl.ac.uk/bsm/PROCAT/PROCAT.html	109
ProClass	A non-redundant protein database organized according to family relationships	http://pir.georgetown.edu/gfserver/proclass.html	
PRODOM	A comprehensive set of protein domain families automatically generated from the SWISS-PROT and TrEMBL sequence databases	http://protein.toulouse.inra.fr/prodom/current/html/home.php	8
PROSITE	Database of protein sequence motifs	http://au.expasy.org/prosite/	110, 111
SBASE	A support vector machines domain prediction system	http://hydra.icgeb.trieste.it/~kristian/SBASE/	212
Sentra	A database of sensory signal transduction proteins	http://www-wit.mcs.anl.gov/sentra/	213
SCOP	A structural classification of proteins database for the investigation of sequences and structures	http://scop.mrc-lmb.cam.ac.uk/scop/	85
Tumor Gene	A resource for cancer-causing mutations; proto-oncogenes and tumor supressor genes	http://condor.bcm.tmc.edu/oncogene.html	214,215
TSGDB	A resource for tumor suppressor genes	http://www.cise.ufl.edu/~yy1/HTML-TSGDB/Homepage.html	216
VIDA	A collection of homologous protein families derived from open reading frames from complete and partial virus genomes.	http://www.biochem.ucl.ac.uk/bsm/virus_database/VIDA.html	217
Wnt gene Homepage	A family of highly conserved secreted signaling molecules that regulate cell-to-cell interactions during embryogenesis	http://www.stanford.edu/~rnusse/wntwindow.html	218
Pairwise comparison			
BLAST	Basic local alignment search tool	http://www.ncbi.nlm.nih.gov/blast/	9
FASTA	Compares a protein sequence to another protein sequence or to a protein database, or a DNA sequence to another DNA sequence or a DNA library	http://fasta.bioch.virginia.edu/	10
WU-BLAST	Basic local alignment search tool developed by Washington University at St. Louis	http://blast.wustl.edu	25
Sequence-profile comparison			
FPS	A method for scoring a query sequence against a family of sequences	http://fps.sdsc.edu/	40
HMMER	Profile hidden Markov model for remote homolog identification	http://hmmer.wustl.edu/	33
IMPALA	A program that searches a protein query sequence against a multiple alignment database represented as a collection of PSI-BLAST checkpoint files	http://blocks.fhcrc.org/blocks/impala.html	28
META-MEME	A software toolkit for building and using motif-based hidden Markov models of DNA and proteins	http://metameme.sdsc.edu/	35

(Box 1) contd….

Tool or Database Name	Functionality summary	URL	Reference
PSI-BLAST	Iterated profile search methods	http://www.ncbi.nlm.nih.gov/BLAST/	25
SAM	A collection of flexible software tools for creating, refining, and using linear hidden Markov models for biological sequence analysis	http://www.cse.ucsc.edu/research/compbio/sam.html	34
Profile-profile comparison			
COACH	HMM-HMM comparison	http://www.drive5.com/lobster	46
COMPASS	Multiple alignment vs. Multiple alignment comparison	ftp://iole.swmed.edu/pub/compass/	44, 45
FFAS	Multiple alignment vs. Multiple alignment comparison	http://ffas.ljcrf.edu/ffas-cgi/cgi/documentn.pl?	219
FORTE	Multiple alignment vs. Multiple alignment comparison	http://mbs.cbrc.jp/htbin/forte-cgi/forte_form.pl	48
HHsearch	HMM-HMM comparison	http://protevo.eb.tuebingen.mpg.de/cgi-bin/download/download.pl?NOW_DIR=/cluster/data/download/HHsearch	47
HMAP	applying sequence, secondary and tertiary information in protein structures into profiles for remote homolog identification	http://honiglab.cpmc.columbia.edu/hmap/	49
LAVA	Block vs. Block search	http://blocks.fhcrc.org/blocks-bin/LAMA_search.sh	220
PROF_SIM	Multiple alignment vs. Multiple alignment comparison	Unavailable	41
SP3	Fold recognition using structural-derived sequence profile-profile method	http://theory.med.buffalo.edu/	50
Other sequence-sequence comparison			
PropSearch	Incorporating information from the amino acid composition instead, molecular weight, content of bulky residues, content of small residues, average hydrophobicity, average charge, and other protein properties (totally 114)	http://abcis.cbs.cnrs.fr/propsearch/propsearch.html	68
Protein Hydrophilicity/Hydrophobicity Search and Comparison Server	Tools for plotting, comparing and searching a library for similarities, of a protein's hydropathy profile	http://bioinformatics.weizmann.ac.il/hydroph/	67
T-HMM	Incorporating phylogenetic information into the HMM	Unavailable	52, 53
STRUCTURE-BASED METHODS			
123D	Comparison and recognition of protein structures	http://123d.ncifcrf.gov/	88
3D-PSSM	A tool for protein threading	http://www.sbg.bio.ic.ac.uk/~3dpssm/	221
CE	Protein structure comparison tool	http://cl.sdsc.edu/ce.html	120
DALI	Protein structure comparison tool	http://www.ebi.ac.uk/dali/	119
FUGUE	A tool for protein threading	http://www-cryst.bioc.cam.ac.uk/~fugue/	222
Genomic Threading Database	Contain structural annotations for the genomes of over 100 recently sequenced organisms	http://bioinf.cs.ucl.ac.uk/GTD	223
GenTHREADER	A tool for protein threading	http://bioinf.cs.ucl.ac.uk/psipred/	170
ORFeus	Hybrid threading approaches	http://BioInfo.PL/Meta/	224
PROSPECTOR	A tool for protein threading	http://www.bioinformatics.buffalo.edu/new_buffalo/services/threading.html	225
PROSPECT	A threading-based protein structure prediction system	http://compbio.ornl.gov/structure/prospect/	98
ProteinDBS	Protein global-to-global tertiary structure matching	http://proteindbs.rnet.missouri.edu/	121
RAPTOR	A threading-based protein structure prediction system	http://www.bioinformaticssolutions.com/products/raptor.php	99
SAS	A tool to bridge the gap between protein sequence and structural analysis.	http://www.biochem.ucl.ac.uk/bsm/sas	91
TOPITS	A tool for protein threading	http://cubic.bioc.columbia.edu/predictprotein/	90
UCLA-DOE Structure Prediction Server	An integration tool for BLAST, PSI-BLAST, PSI-PRED, SDP, and DASEY	http://fold.doe-mbi.ucla.edu/	86

constructed. Generally, these matrices can be classified into two groups: (1) the matrices generated empirically using multiple sequence alignments, such as PAM [15], mPAM [16], Gonnet [17], and JTT [18]; (2) the matrices generated from blocks of local alignments in related proteins, such as BLOcks SUbstitution Matrix (BLOSUM) [19], and Probability Matrix from Blocks (PMB) [20]. Relative to PAM, BLOSUM was shown to improve the sensitivity and accuracy of the alignments significantly, especially for those sequences with a large divergence [19,20].

Table 1. Comparison of Sequence-Sequence Comparison Methods

Methods	Advantages	Disadvantages
pairwise sequence comparison	straightforward; with solid theoretical foundation; fast	insensitive
sequence-profile comparison	more sensitive than pairwise sequence comparison	depends on whether the query sequence has many close homologs
profile-profile comparison	up to 30% more sensitive than sequence-profile comparison	higher computational cost; more false positives predicted
phylogenetic analysis	directly related to evolutionary relationship	computationally expensive; higher possibility of misleading results due to the complexity of phylogenetic tree construction
other methods (intermediate sequence, secondary structure, hydrophobicity profile, protein composition, frame-shift information, etc.)	may improve the remote homology detection sensitivity.	high false positive rates; no good method for prediction confidence assessment

The dynamic programming technique has been implemented to achieve either local (e.g. BLAST) [9] or global optimality (e.g. FASTA) [10]. However, different from Needleman-Wunsch [13] and Smith-Waterman [14] algorithms, both BLAST and FASTA employ heuristic approaches to identify the similarity between two query sequences [21]. As a result, these two methods run much faster. However, neither BLAST nor FASTA can guarantee the optimal alignment. To compare the reliability of homolog identification, the statistical assessment was integrated into pairwise sequence alignments [9,10,22-24]. For example, the Expectation Value (E-value) returned by BLAST reflects the reliability of homologous relationship between two sequences. Although remote homologs are often missed by pairwise sequence alignment due to its insensitivity, this method is typically the first step in the search for remote homologs.

Since the first development of the BLAST program, different versions of BLAST have been implemented. These BLAST programs have also been integrated into different online databases, which provide convenient tools, especial for experimental biologists, who are mainly PC users. WU-BLAST (http://blast.wustl.edu) [25] is the first BLAST package with gapped alignments, and it is different from the original version of gapless BLAST alignment [9], which is usually called NCBI-BLAST. WU-BLAST is more sensitive for remote homology identification but may be slower than NCBI-BLAST when using the similar parameters. The WU-BLAST Web servers are listed at http://blast.wustl.edu. For instance, WU-BLAST2 server at the European Bioinformatics institute (http://www.ebi.ac.uk/blast2/) is one of the most frequently used ones [26]. The databases that use WU-BLAST include not only general databases, such as the *nr* at NCBI (ftp.ncbi.nlm.nih.gov), but also many specific databases, such as the comprehensive microbial database at TIGR (http://tigrblast.tigr.org/cmr-blast/), the mouse database at Jackson Laboratory (http://mouseblast. informatics.jax.org/), the yeast database at Stanford University (http://seq.yeastgenome.org/cgi-bin/nph-blast2sgd), and the plasmodium database at the University of Pennsylvania (http://plasmodb.org/). The databases (protein, DNA, and masking databases) available for BLAST searches are listed at http://www.ch.embnet.org/software/blastdb_help.html.

2.2 Sequence-Profile Comparison

2.2.1. PSI-BLAST

A protein sequence profile (position specific weight matrix), which is generated by aligning a group of closely related protein sequences, illustrates the probability of occurrence for each amino acid at each position of the multiple sequence alignments. A sequence profile better reflects the information from a protein family than a single sequence. Thus, sequence-profile alignment is typically more sensitive for remote homolog identification than pairwise sequence alignment [27]. Popular sequence-profile alignment tools include PSI-BLAST [25] and PSI-BLAST derived IMPALA [28]. The challenge for sequence-profile alignment resides in the selection of “closely” related sequences in the database for the profile construction. Previous reports demonstrated that the selection of the first query to initiate the profile search affects that sequence-profile homolog identification results [29].

2.2.2. Hidden Markov Model

Hidden Markov Model (HMM) illustrates the probability distribution for a finite set of states. Each state will have an associated transition or emission probability. The output will be the joint probability for each state. However, the state is not visible to an external observer, thus it is called Hidden Markov Model. HMM can be first-order, second order, or *m*-order HMM. The *order* of a HMM is the context length, and the context will determine the probabilities of the next state. For example, a state in second order HMM depends upon its two previous states.

Profile Hidden Markov Models (HMMs) are among those most powerful approaches for remote homolog identification. Unlike multiple sequence alignments in PSI-BLAST, HMMs construct the profile based on HMM, which contains a probability matrix for position transitions and an emissions matrix for amino acid changes [30,31]. Due to complexity for profile construction, HMMs are at least 10 times slower than PSI-BLAST [32]. On the other hand, HMM based approaches often have a better performance in sensitivity than PSI-BLAST based approaches [32]. HMMER [33], SAM [34], and META-MEME [35] are the ones used most frequently. Among these three models, HMMER and SAM have been the most successful profile HMMs. One report in 2002 showed HMMER is faster than SAM but SAM is more sensitive for remote homology identification [34]. However, since then, both SAM and HMMER have been updated. Other recent HMM based methods for remote homolog identification include [36,37].

2.2.3. Other Sequence-Profile Comparison Methods

In order to avoid the loss of sensitivity and precision from heuristic searches, Jumping Alignment (JA) algorithm was developed based on the exhaustive searches as in Smith-Waterman algorithm [38]. Different from the normal sequence-profile alignment, JA aligns a single protein sequence with multiple sequence alignments by aligning the query sequence with each sequence in the multiple alignments whereas the profile approaches generally consider the counts for each amino acid at each column. To resolve the alignment, JA algorithm may jump from one reference to another reference sequence, thus it is similar to fragment based alignment methods, such as DIALIGN [39]. The study in [38] claims that for the median false positive counts (discrimination quality), JA had a higher precision than other methods using sequence profiles or hidden Markov models. However, no independent test for this claim is available. The Family Pairwise Search (FPS) is another algorithm not based on position specific score but based on the multiple pairwise alignments between the sequence in the query set and those sequences in the database [40].

2.3. Profile-Profile Comparison

Profile-profile comparison detects the remote homolog by comparing two profiles which are constructed in the similar way as the (multiple alignments or HMM) profile described in section 2.2. It has been shown that profile-profile comparison is significantly more sensitive in detecting remote homologs than the sequence-profile based search programs, such as PSI-BLAST and profile HMMs [41,42]. The profile-profile comparison may be able to achieve the same order of magnitude as the improvement of PSI-BLAST over BLAST [41]. One assessment demonstrated that profile-profile comparison outperformed sequence-profile comparison over 30% [43]. Nevertheless, the profile-profile approaches are not used so frequently as sequence-profile tools. A major problem for profile-profile methods is that they require two confident profiles. In most cases, users without extensive experience in bioinformatics cannot provide high-quality profiles easily. Thus, generally one has to apply PSI-BLAST or some other sequence-profile methods to generate profiles first, and the final accuracy will be affected dramatically by this profile construction step. In addition, as the number of the entries in some databases (e.g. nr) is very large, the processes of profile construction and profile-profile comparison may be very time consuming.

The available profile-profile comparison tools include COMPASS [44,45], PROF_SIM [41], COACH [46], HHsearch [47], FORTE [48], HMAP [49], and SP^3 [50]. Among these methods, COMPASS, PROF_SIM, and FORTE are multiple alignment-multiple alignment comparison based approaches, whereas COACH and HHsearch are HMM-HMM based methods. The software evaluation shows that COACH is only slightly better than COMPASS, but COACH is much faster (5 times) than COMPASS according to searching speed [46]. Similar to COACH, HHsearch generated 1.2, 1.7, and 3.3 times more good alignments than COMPASS at the family, superfamily, and fold level [47]. HHsearch is also 10 times faster than PROF_SIM and 17 times faster than COMPASS [47].

HMAP [49] uses profile-to-profile alignment with position-dependent weights for both sequence and structural information. Since the core structural elements and secondary structural information are only used in the weighting function and the profile itself is sequence-based, HMAP is still a sequence comparison method, instead of a threading method as described below. The results of HMAP suggest that incorporation of structural information brings substantial improvement over the pure sequenced based profile-to-profile approaches. SP^3 [50] is another structural-derived sequence profile-profile method. SP^3 performs profile-profile comparison for: (1) evolutionary derived sequence profile of the query sequence and that of the template sequence; (2) the query sequence profile and the template profile derived from the aligned structures of the templates. The information on the depth of residues from protein surface was incorporated to compensate the potentially lost information by fragmented structural alignment. This strategy proposed in SP^3 was shown to improve the performance significantly than previous version SP (sequence profile alone) and SP^2 (sequence plus secondary structure profiles).

Among these methods, COMPASS adapted the statistical methods from pairwise sequence comparison, Extreme Value Distribution (EVD) [22] to estimate the homology significance for two protein profiles to compare. The E values will be assigned after the profile comparison. The lower E values, the more confident the compared protein profiles.

2.4. Phylogenetic Analysis

The phylogenetic relationship of the sequences directly reflects the evolutionary relationship between sequences. Thus, it will be able to directly detect or validate the remote homolog. But in practice, it may not be efficient to use phylogenetic analysis to identify remote homolog because (1) the evolutionary distance is too far to detect, (2) current phylogenetic tree construction strategies cannot guarantee the optimal tree, and (3) as the number of sequences increase, the chance of obtaining “wrong” tree increases exponentially. Despite these disadvantages, some research has been carried out for the applications of phylogenetic analysis in remote homolog identification. For example, Rehmsmeier and Vingron [51] used the length of the edge in the tree to judge the homology between query sequence and a family. Recently, in T-HMM [52], the phylogenetic information has been incorporated into the profile HMM and resulted in an improved accuracy. The new version of T-HMM with iterative algorithm for model refinement was shown to have a better performance [53].

Based on evolutionary relationship, derivation of the ancestral sequences may enhance the remote homolog identification since those ancestral sequences may be more similar to some remote homologs. ANCESCON [54] and GASP [55] have been developed specifically for ancestral protein sequence reconstruction.

2.5 Other Approaches

All the above methods rely on protein sequence alignments, either pairwise or multiple. There are alternative methods that are also based on sequence comparison. These methods may not have good specificity, i.e., the prediction reliability may not be as high as the methods above. However, they may be more sensitive so that they can detect some remote homologs that cannot be identified by the direct sequence alignment of amino acids. Six methods based on protein related features are summarized below. The other gene attributes for nucleotide sequence, such as GC content and codon usage, may also be useful in remote homolog identification, and this will not be discussed in the paper.

2.5.1. Homology Detection Through Intermediate Sequences

This approach [56] is based on the idea that two sequences might be so divergent that a direct comparison between them may not yield any meaningful results, while both of the sequences may be similar to a third one which will act as a transitive sequence between the original two. This is the approach implemented in the DOUBLE-BLAST tool, where the BLAST hits of a query protein sequence are used as a new set of queries for the next BLAST run. The work by Park *et al.* [56] also suggests that the ISS (Intermediate Sequence Search) approach works better in prediction sensitivity than profile-sequence approaches such as PSI-BLAST. Other implementations of homology detection through ISS are also available [57-59].

2.5.2 Incorporation of Secondary Structure Information

This approach [60,61] is based on the fact that secondary structures of proteins are more likely to be conserved than their sequences. In addition, the accuracy of secondary structure prediction is close to 80%. Hence, one can predict the secondary structures for two proteins under alignment, and then use the predicted secondary structures as an additional scoring function. It was found that if the predicted secondary structures of two compared sequences match by more than 50%, then these sequences are more likely to be structurally related (also likely to be homologs). Even when the sequence identity was below 20%, homology could still be detected using the secondary structure comparison [62].

Wallqvist *et al.* [63] combined both PSI-BLAST and secondary structure prediction to detect remote homology and found that this combination generated higher prediction sensitivity than PSI-BLAST alone. Structure-based ALignment TOol (SALTO) [64] is another recent sequence-sequence comparison software incorporating the secondary structural information, which was derived from the conserved domains in the NCBI's Conserved Domain Database. DescFold [65] further integrated PSI-based descriptor, the predicted secondary structure descriptor, and the PROSITE motif-based descriptor by using a support vector machine-learning algorithm [66]. DescFold was shown to have better prediction accuracy than any method using one of these three descriptors.

2.5.3 Search Remote Homolog Using Hydrophobicity Profile

This approach is based on hydrophobicity profile extrapolated from the hydropathy scales of residues along a protein sequence. Hydropathy scale is defined as a measure of hydrophobicity of an amino acid and comes in several different sets. A hydrophobicity profile value at a certain sequence position is obtained by averaging the hydrophobicity scales of several neighboring residues. In some cases, the two remote homologs do not share any significant sequence similarity, but they share similar hydrophobicity profiles. Detecting a similar hydrophobicity profile for the query protein in a protein sequence database can be an alternative approach for possible remote homolog identification. Such strategy is implemented in the Protein Hydrophilicity/Hydrophobicity Search and Comparison Server (http://bioinformatics.weizmann.ac.il/hydroph/) [67].

2.5.4 Homolog Detection Using Compositional Properties of Protein Sequence

In this approach, as implemented in PropSearch [68], a fundamentally different measure of similarity between proteins is used – protein dissimilarity is defined as a weighted sum of differences of compositional (physico-chemical) properties such as singlet/doublet residue composition, molecular weight, and isoelectric point. This approach can use either a single sequence or multiple sequences as a query to the database. In case of multiple sequences being used as a query, they can be reconciled into a consensus sequence describing the "average composition" of the protein family. PropSearch does not require alignments and is very fast when scanning a preprocessed database. The searches use reduced information from protein sequence, and hence, more false positives are expected than sequence-alignment methods. Nevertheless, the tool provides a useful alternative for further remote homolog identification when traditional sequence/profile-based methods fail.

2.5.5 Homologous Relationship with Frame-Shift

This approach [69] is based on the assumption that some nucleotide frameshifts in DNA/RNA that result in changes in protein are responsible for the divergence between protein sequences. For example, if two genes are duplicated from the same ancestral DNA, due to point mutation and in-frame insertions/deletions for one or both of these two genes, the resulting protein sequences may be entirely different. However, they may maintain the similarity at the nucleotide level. In these cases, the classical sequence comparison methods using protein sequences, which only consider insertions, deletions and mutations in protein sequences, will not be able to detect such evolutionary relationships. To account for frame-shifts, sequences were compared using special amino acid substitution matrices for the alternate frames of translation. Such a method provides a sensitive approach for detecting a different type of remote homologs. Since these methods are based on an assumption completely different from other protein sequence based methods, the performance of these methods cannot be compared with other methods in general. Instead, this method may complement other sequence-based approaches and discover the remote homologs missed by protein sequence based approaches.

2.5.6 Homologous Detection Through Machine Learning Approaches

Machine learning approaches can classify the query sequences into two classes, either remote homologs or not. The advantages of machine learning approaches include its flexibility for incorporation of statistical assessment (e.g. probability) as well as its power for integrating various features from a protein sequences. Generally these methods use BLAST, PSI-BLAST, or some other sequence comparison approaches to collect the first homologous sequences and then apply machine-learning approaches to construct rules for the classification. The disadvantages for these methods are that it is usually very hard to obtain solid and large-enough training and testing datasets. The rapidly increasing data in the databases have increased the potential of machine learning approaches in identifying remote homology. For example, Homologous Induction (HI) [70] is such an algorithm. HI first collects possible homologous sequences using PSI-BLAST and then accumulates information on these homologous sequences, such as amino acid distribution for singlets and pairs of residues, the description in SWISS-PROT (function, keywords, organism, molecular weight, and database reference) [71], secondary structures, predicted cleavage sites, hydrophobicity, and length and starting point of local PSI-BLAST alignments. Then, HI applies an inductive logic programming [72] to construct the rules for classification based on learning. Their results demonstrate that HI was more sensitive than PSI-BLAST. SVM-HMMSTR [73] is also a method integrating the sequence information and structural motifs, which are represented using Hidden Markov Model. SVM-HMMSTR employs Support Vector Machines [66] to predict remote homolog.

3. SEQUENCE-STRUCTURE COMPARISON METHODS

Since the 3D structures of proteins have been better conserved during evolution than their sequences, protein structure prediction often provides a more sensitive approach to identify distant evolutionary relationships (remote homology) than sequence-comparison methods. Among the protein-structure prediction methods, threading is the most suitable for remote homolog identification [74-77]. The idea of threading was derived from the observations that proteins with no apparent sequence similarity could have similar structural folds and that the total number of different structural folds in nature may be small (possibly a few thousand) [78]. Thus, a structure prediction problem can be reduced to a recognition problem, i.e., given a query protein sequence, searching for the most compatible structural fold based on sequence-structure relationships. Sequence-structure relationships include the notion that different amino acids may prefer different structural environments. For instance, a hydrophobic amino acid tends to be in the interior of a globular protein, and proline rarely occurs in an _-helix. Once a structural template for the query sequence is identified, the template can serve as a basis for function inference of the query protein, although the template can be an analog of the query protein (i.e., the query and the template do not share biological function or have evolutionary relationship). In this section, we will first introduce the four components of threading. Then we will discuss how to use the predicted structural template for function inference of the query protein. Readers can find more details about protein structure prediction based on threading in [79,80].

3.1. Threading Components

A threading method typically consists of four components [81]: (1) a library of representative 3D protein structures as templates; (2) an energy function for measuring the fitness between a query sequence and a template structure in the library; (3) a threading algorithm for searching the lowest energy among the possible alignments for a given sequence-template pair; (4) a criterion for estimating the confidence level of the predicted structure. The following discussion addresses each aspect in detail.

3.1.1. Fold template Library

A fold template library is intended to represent all the experimentally determined protein structures in the database PDB [82]. As many proteins in PDB are similar to each other in both sequence and structure, it is not necessary to include all of them in a fold library. Typically, only the representative proteins based on protein structure classification are used. Most template libraries of the existing threading programs are based on three widely used databases of protein structure classifications: CATH [83], FSSP [84], and SCOP [85]. CATH is a hierarchical classification of protein domain structures. FSSP contains a hierarchical classification of protein chains as well as sequence neighbors and multiple structure alignments. SCOP represents a hierarchical classification based on protein domains, and each protein in SCOP is classified into family, superfamily, and fold. Since the classification of folds by these three databases differs somewhat due to their different classification criteria (e.g., classification on a whole chain or a structure domain) and structure-structure comparison methods, the number of unique folds differs among these three databases.

After the templates are selected, some processing will be carried out for each template to include derived information from the structure, such as protein secondary structure and solvent accessibility, both of which are needed for threading calculation.

3.1.2. Scoring Function

The scoring function describes how favorable an alignment between a query sequence and a template structure is. Threading generally uses knowledge-based scoring functions rather than physical energies, since physical energies are too sensitive to small displacement of atomic coordinates, making them less suitable for threading and too time-consuming for computing. A typical threading scoring function has the following form:

$$S\_total = S\_mutate + S\_gap + S\_single + S\_pair \qquad \textbf{(1)}$$

The mutation score *S_mutate* describes the compatibility of substituting one amino acid type by another in sequence; *S_gap* is the alignment gap penalty; the singleton score

S_single represents a residue's preference to its local secondary structures (_-helix, _-strand, and loop) and its preference to being in a certain solvent environment (either exposed to solvent or in the interior of the protein); *S_pair* is the pairwise score between spatially close residues that are not neighbors in the protein sequence. The mutation score and the alignment gap penalty are similar to the ones used in sequence alignments. It has been shown that PAM250 is one of the best substitution matrices available for threading [86,87]. The gap penalty is often a linear function of the gap size, with a penalty for opening a gap and a smaller penalty for each extension thereafter. Both *S_single* and *S_pair* are typically derived from Boltzmann statistics using a non-redundant protein database [88]. The basic idea is that if an amino acid is frequently observed in the interior of protein structure, a favorable score value will be rewarded when it is aligned to an interior position of a template.

The score function can also integrate some other attributes beyond the sequence similarity, such as structure-function relationship. MANIFOLD [89] utilizes a non-linear ranking scheme (a simple two-layer neural network) to integrate three scores from secondary structure, sequence similarity, and enzyme classification.

3.1.3. Alignment Algorithm

The alignment algorithm in the context of threading means the computational methods to identify a sequence-structure alignment with the best threading score as described in Equation (1). If we do not consider the pairwise score, a threading problem is essentially the same as a sequence alignment problem. Such a problem can be solved efficiently by a dynamic programming approach [13,14]. There are a number of computer programs which essentially use dynamic programming for their threading problem, e.g., 123D [88], TOPITS [90], SAS [91], the UCLA-DOE Structure Prediction Server [86]. The speed of the threading algorithm will be improved if pairwise score is not considered. However, a benchmark shows that without pairwise interactions, the threading accuracy is compromised [92]. On the other hand, threading with pairwise terms and alignment gaps is generally considered to be a very difficult problem [93]. Two previously existing threading programs with rigorous solutions have exponential time complexity [94,95]. To overcome the computational expense, several methods that use statistical sampling have been proposed [75,76,96,97]. Such methods do not guarantee to find the globally optimal threading alignment. To solve the globally optimal threading problem efficiently, a unique threading algorithm (PROSPECT) based on a divide-and-conquer algorithm [98] was developed under the assumption that the pairwise term needs to be considered only between spatially close pairs in threading.

RAPTOR [99] formulates the protein threading problem as a large scale integer/linear programming problem to handle pairwise interactions. RAPTOR extracts the optimal alignment using a branch-and-bound method and a standard integer programming solver. It was convincingly demonstrated, through applications of these programs at the CASP contests [100], that threading programs with guaranteed global optimality had an advantage over programs without this property.

3.1.4. Assessment of Threading Results

A threading score between a query sequence and a template structure may not provide enough information to determine whether the template is the "correct" fold. This is because the scores from separate folding processes are not generally normalized to the same scale. Hence, from the threading scores for a query and a pool of templates, we generally cannot tell if the correct fold template for the query is in the pool nor can we always identify the correct fold through ranking the raw scores even if it is present in the pool. There have been a number of attempts to "normalize" the threading scores so that they can be compared with each other. An early attempt was to use *z-score* [101]. There have also been attempts to use the P-value schema [22,23] as a way to assign a meaning to a threading score. P-value, which estimates the probability of having a particular alignment score between two random sequences, have been successfully applied to sequence alignment, thanks to Karlin's seminal work on a rigorous model of gapless alignments [22,23]. Due to the lack of a rigorous model for threading, the P-values are typically estimated through compiling a "large" number of threading scores between a query sequence and a template after randomly shuffling its residues [96]. While some usefulness of the estimated P-value has been demonstrated, the problem of developing a rigorous P-value scheme for threading still remains as an open challenge. A practical way to "normalize" the threading scores is to feed the threading raw scores along with various normalization factors such as sequence length to a neural network, which has learned to "optimally" combine these factors based on a training set [102,103].

In the above four components, the template library is straight forward, while others components are still under further development. More sophisticated energy functions, faster alignment algorithms, and more effective confidence assessments for threading results are being explored. On the other hand, the pure threading approach in the classical sense (i.e., using structure template information only) is not as widely used as sequence-based methods; instead we observe a trend of integrating threading with other methods for remote homolog identification and protein structure prediction in different directions: (1) mixing threading methods with sequence-comparison methods that use sequence profiles, as discussed above for HMAP [49]; (2) combining various prediction methods to seek consensus of prediction results, which often is more accurate than the prediction from any single tool [104]; (3) integrating threading into a computational pipeline with various tools for protein structure and function analyses [105]; (4) incorporating the threading idea into *ab initio* protein structure prediction, i.e., the mini-threading approach [106].

3.2. Identification of Remote Homolog from Structural Relationship

Even when the predicted 3D structure has a poor quality due to a wrong alignment between the query protein and the template, the identified fold template often represents a remote homolog of the query protein, so that some evolutionary and functional relationship can be inferred between the query and the template. Given that threading often produces inaccurate alignment, it may be more useful in remote homolog identification than in 3D structure prediction.

Although remote homolog may be identified from the predicted structures, a relationship in structure does not guarantee a homology relation. The relationship between the proteins can be classified at different hierarchical levels according to structural, functional, and evolutionary relationships. A widely used classification schema consists of three levels of groups: fold, family, and superfamily, as shown in the SCOP [85], which currently has 2,630 families, 1,447 superfamilies, and 887 folds (1.67 release, May 2005). A *family* consists of proteins that have a significant sequence identity (often 25% or higher) between each other and share a common evolutionary ancestor (close homolog). Proteins of different families sharing a common evolutionary origin (reflected by their common structural and functional features), typically remote homologs, are placed in the same *superfamily*. Different superfamilies, i.e., analogs, are grouped into a *fold* family if their proteins have the same major secondary structures in the same arrangement and with the same topological connections. The structural similarities among proteins from the same fold family (but not the same superfamily) may arise just from the protein energetic favoring certain packing arrangements instead of a common evolutionary origin.

Although proteins with the same fold may not be the homologs, one can suggest a possible homolog of a query sequence from its predicted fold using the SCOP database. When a predicted structural fold contains multiple superfamilies, it is possible to predict the most likely homolog for the query proteins among all the superfamilies based on threading results. For example, PROSPECT [92] calculates a z-score that measures the reliability of the structure prediction and the possible homology relationship [107], as shown in Table **2**. The z-score is the threading score in standard deviation unit relative to the average of the threading raw score distribution of random sequences with the same amino acid composition and sequence length against the same structural templates. In practice, the average and the standard deviation are estimated by repeated threading between a template and a large number of randomly shuffled query sequences. When the z-score of the prediction is higher, the query and the template are more likely to be homologs, and one can simply select the superfamily with the highest z-score among all the superfamilies in the predicted fold as the (remote) homolog. When the z-score is low, the predicted fold may not represent a homolog at all.

To further pin down whether a query protein and a predicted template are homologs, one can check functional motifs. If the predicted structure contains a functional motif (conserved residues at a particular position in the 3D structure, not necessarily close to each other on the protein sequence) of a protein in the template, the query protein and the template are probably homologs. Zhang *et al.* [108] have constructed a database of functional motifs for known structures (e.g., EF-hand motif for calcium binding), called SITE. Currently, SITE contains identified functional motifs from about 50% of the SCOP superfamilies. One can also search the predicted protein structure against PROCAT [109], which is a database of 3-D enzyme active site templates.

Although the structural motifs are more general, the comparison between the query protein and the template in terms of the motifs depends on the alignment accuracy, which may be difficult to achieve by threading. Hence, one can also carry out sequence-based motif searches using PROSITE [110,111], PRINTS [112,113], and BLOCKS [114]. A good example is the target T0053 of CASP3 [115]. Using PROSPECT, we successfully identified a native-like fold (1ak1) of T0053 in PDB, as shown in (Fig. **2**). T0053 and 1ak1 have only 11.2% sequence identity in the sequence independent structure-structure alignment. Without additional information, it is difficult to determine whether T0053 and 1ak1 are remote homologs. Using the BLOCK search [116], we found that the two proteins share the same sequence block with a conserved active site at His-183 in 1ak1 and His-145 in T0053. This information allowed us to determine that the two proteins are remote homologs. Our prediction turned out to be obviously correct when the experimental structure of T0053 was determined (PDB code: 1qgo).

4. SEQUENCE-INDEPENDENT METHODS

Both the sequence-comparison methods and protein-structure predictions for remote-homolog identification use the information from the query-protein sequence. In some

Table 2. Interpretation of the z-Scores from PROSPECT

z-score interval	Probability to be correct	Confidence level	Homology
< 6	<0.3	Unlikely	analogs/unrelated
6 - 8	0.35	Low	superfamily/analogs
8 - 10	0.63	Medium	superfamily/fold
10 - 12	0.85	High	Superfamily
12 - 20	0.96	very high	family/superfamily
> 20	>0.99	Certain	Family

The first column represents the z-score range. The second column shows the probability of a sequence-template pair sharing the same fold within a certain z-score range. The third column shows a corresponding qualitative confidence level. The fourth column provides a possible homologous relationship between the query and template protein in terms of the SCOP protein family classification, *family, superfamily, and fold* [**Error! Bookmark not defined.**].

cases, the information about a remote homolog is also revealed in other sources, such as structure-structure comparison, evolutionary footprints, and gene expression. In this section, we will address these sequence-independent methods for remote homolog identification.

Fig. (2). A comparison between the predicted structure (left) and the experimental one (right) for the CASP-3 target t0053. The cylinders indicate α-helices, the strands indicate β-sheets, and the lines indicate loops.

4.1. Structure-Structure Comparison

When the structure of a protein is known, one can use the structure to identify its homologs through comparing with other known structures. This is similar to the threading method in the sense of using structural information, while the structure-structure comparison is far more reliable than sequence-sequence comparison or sequence-structure comparison in identifying remote homologs. Protein structure-structure comparison had limited applications in remote homolog identification in the past, as not many protein structures were available. With the advent of new technologies such as synchrotron radiation sources and high-resolution nuclear magnetic resonance (NMR), a great number of new protein structures have been determined in recent years. In particular, in the recent effort of structural genomics [117,118], where protein structures are being determined on a large scale, the structures of many proteins were determined without knowing their function. Structure comparison provides a useful tool to identify remote homologs for these proteins, and further predict functions based on the homologs.

Thus, when the structure of a protein is available, one can use the structure to search against the database of known protein structures, i.e., PDB, and the hits with similar structures are potential remote homologs of the query protein. Popular tools for comparing a query protein structure against all the structures in the PDB are DALI [119] and CE [119]. A much faster search engine for protein structure comparison, ProteinDBS [121], was developed recently. Based on the alignment between the query protein structure and the hits in the structure database, one may find biologically interesting similarities that are not detectable by sequence comparison or threading. For example, common structural motifs between two aligned structures can be found. Using such information, one can tell whether two proteins of the same fold are remote homologs or merely analogs. Several protein-structure classification databases, such as SCOP, CATH, and FSSP, as discussed in Section 3, can facilitate the search for remote homolog using structure comparison. These structural databases provide a useful resource for systematically checking the common features of structural motifs and sequence patterns among proteins in the same superfamily, and these features can help to tell whether two proteins of the same structural fold are homologs or analogs.

4.2. Paralog Relationship in High-Throughput Biological Data

During evolution, some genes may be duplicated and then diverged (i.e., paralog as defined in Section 1). The paralogs often have some trace in various high-throughput biological data, including genomic sequence data, gene expression data, and genetic interaction data. Although these traces alone are typically insufficient for predicting paralog relationships, they can help remote paralog identification in conjunction with other methods. In particular, when such traces occur, a paralog prediction from other methods would have an increased confidence level. The following three types of traces can offer some support of possible paralog relationships:

4.2.1 Adjacent Genes in Genomic Sequence

Many gene duplications occur in tandem. Hence, it is not surprising that many paralogs were also found in adjacent positions of a genome [122,123]. These parallel functional modules increase cellular flexibility and robustness [124]. Probably the simplest approach is to apply traditional homolog identification and then correlate the identified homologs with positions. In prokaryotes, it may be relatively easier due to the general existing operon structure is likely to preserve both functions and positions. Many operon prediction approaches may be effectively adapted to paralog identification [125-128]. However, for eukaryotes, it is much more challenging since there is no operon structure for functional module. Through comparative genomics strategy, Li *et al.* [124] took a four-step approach to predict adjacent gene remote homologs: (1) calculate functional linkages for all possible protein pairs in the query genome by comparing them with proteins in other genomes; (2) construct a matrix of functional linkages for the query genome and group proteins based on the similarity of their functional linkage patterns in other genomes using a hierarchical clustering algorithm; (3) visually search for off-diagonal clusters within the functional linkage map; (4) manually match the module partners from each subgroup and identify functional linkages that result from paralogous relationship. Through this strategy, they identified 37 cellular systems of parallel functional modules from 10 genomes, including a number of previously reported parologs.

The large-scale experimental protein interaction may be applied in paralog identification in the same genome [129]. However, when compared to computational methods, they are much more time consuming. Probably a combination of both computational methods and bench work may be the most efficient and effective approach for paralog identification in the genomic scale.

4.2.2 Correlated Microarray Gene Expression Patterns

The neighboring genes due to this duplication mechanism often show similar expression patterns, since these adjacent genes share a single upstream activating sequence in many cases. The correlated gene expression pattern also relates to the distance between the genes on the genomic sequence, as it was found that the expression similarity was correlated to the physical distances in both prokaryotes and eukaryotes, such as *Saccharomyces cerevisiae* and *Arabidopsis thaliana* [123]. This means that if two neighboring genes on a genomic sequence share similar gene expression pattern, they are likely to be paralogs. The correlated gene expression pattern among paralogs also extends to orthologs [130].

4.2.3 Genetic Interactions Based on Synthetic Lethality Screening

The synthetic lethality screening is a very powerful method for finding “genetic interaction” between gene products [131]. It identifies lethal deletions of two genes at the same time, while either deletion alone is not lethal. A systematic high-throughput synthetic lethal analysis was carried out in yeast *Saccharomyces cerevisiae* for 4,700 viable mutants [132]. Between two genes with such a genetic interaction, one may be a backup of the other, and hence, the two genes may be paralogs.

5. ASSESSMENT OF COMPUTATIONAL METHODS

Given that so many methods are available for remote homolog identification, it is very important to compare these methods based on some benchmark tests. However, such a comparison is not trivial. If we look into any particular paper discussing an individual method, typically the paper shows that the method outperforms others. The results depend on what criteria are used and how the comparisons are performed. At least the following six criteria can be considered when comparing different methods for remote homolog identification:

(1) Sensitivity of remote homolog identification, i.e., how many true remote homologs can be identified as top hits from all remote homologs in the database? For example, if *k* remote homologs are in the database of a query protein sequence, how many of them rank as top *n*.

(2) Specificity of remote homolog identification, i.e., among the top hits, how many of them represent true homologs? For example, if top *n* hits in the database are selected for a query protein sequence, how many of them are true homologs?

(3) Reliable confidence assessment, i.e., to what extent can the prediction result of homolog identification be trusted? Does such assessment reflect the prediction accuracy well?

(4) Alignment accuracy, i.e., in an alignment between the query protein and the correctly identified remote homolog, how many alignment positions are biologically true? A true biological alignment is typically represented by the structure-structure alignment between the two proteins.

(5) Applicability, i.e., what conditions does a method require? For example, for the threading method, it requires that the structure of a remote homolog for the query sequence is available in the database. Some methods do not have explicit requirements, but they tend to work poorly in certain cases, e.g., HMMs do not work well when the query protein does not have any close homolog to build profiles.

(6) Computational efficiency, i.e., the computing time and memory requirement, and their dependence on the query protein size. This turns out to be very important in practice, especially because many related computations are carried out in large (genome) scale. For example, it is known that many other methods are more accurate in identifying remote homologs, but PSI-BLAST is still the most popular method for remote homolog identification, given that it is very fast and has a linear computational complexity time.

Even though constant improvements have been made for remote homolog identification methods, there is still much room for improvement along the six criteria. It has been shown [133] that for close homolog identification (with sequence similarity over 30%), almost all the methods work very well, with insignificant differences for criteria 1-4. However, when predicting remote homologs, none of the methods consistently outperforms others in all of the six criteria. Hence, although PSI-BLAST is the most popular method, many other computational tools are also widely used at the same time.

Some systematic benchmarks to compare different sequence-comparison methods have been constructed. These comparisons often use SCOP as the gold standard and focus on whether a method can detect remote homologs in the same superfamily but in different families and also how well the sequence alignment compares with the structural alignment. It was found [133] that while comparing sequences below 30% identity (many of them are in the same family), less than 50% of remote homologs could be detected using tools like BLAST, FASTA or the Smith-Waterman SSEARCH. However, even for the identified homologs, the study in [133] suggested that the P-values generated by BLAST seem to underestimate the errors and the alignments are often inaccurate. Another study [134] compared the performance for sequence-alignment accuracy against structure-structure alignment among a pairwise alignment method (BLAST), a sequence-profile method (PSI-BLAST), and an intermediate-sequence-search method (DOUBLE-BLAST). On sequence similarities between 10% and 15%, BLAST, PSI-BLAST, and DOUBLE-BLAST correctly aligned 28%, 40%, and 46% of these sequences, respectively. This indicates that all methods have much room for improvement in alignment accuracy.

Another set of benchmarks comes from the protein-structure-prediction community. Although protein structure prediction focuses on structure instead of homology, the dominant method is to identify homologs in the protein structure database PDB and use the homologs as templates to build protein tertiary structures. As a result, the structure-prediction assessment is also applicable to remote-homolog prediction. To assess objectively the state of the art in prediction tools for protein structures, the computational structural biology community has agreed on an evaluation system called CASP (Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction, http://predictioncenter.llnl.gov/). CASP was initiated in 1994 and has been a biannual event since its inception. In each CASP, participants were given tens of protein sequences whose experimental structures were being solved or had been solved but not published. CASP participants then predicted their structures blindly, either in an automated fashion or with manual adjustment. A group of invited assessors evaluated how well each predicted structure matched the experimental structure. At the end of the prediction season, the performance of each team was ranked. The CASP exercises provide an objective way to assess related computational methods, particularly for Criteria 1, 4, and 5. The strengths and weaknesses of each method are often revealed. For example, even though remote homolog identification methods have been consistently improved, the alignment accuracy had little improvement over the past few years [135,136]. Two observations from the CASPs are (1) manual process (the human knowledge) can help improve prediction significantly, and (2) using consensus approach, i.e., to find common hits from different methods, can outperform any individual method substantially [137]. Based on such findings, computational pipelines [138-140] or expert systems [141] have been developed to incorporate various methods and human knowledge to improve the prediction accuracy. Some hybrid methods using various types of information together were also developed [142,143].

Other than manual predictions and evaluations in CASP, some fully automated servers for protein structure predictions and evaluations were developed. Such efforts complement CASP and provide useful information for assessments of computational tools themselves (instead of human experts). One of them is CAFASP [144], which was carried out in parallel with CASP, using the same set of prediction targets. The third CAFASP in 2003 showed that several best automated prediction servers using the consensus approaches achieved comparable performance as human CASP predictors. This result shows that significant progress has been achieved in automatic structure prediction. Another automated evaluation server is MaxBench (http://www.sanger.ac.uk/Users/lp1/MaxBench/) [145]. This system makes it easy for developers both to compare the performance of their methods to standard algorithms and to investigate the results of individual comparisons. Two large-scale evaluation servers using updated PDB entries as test cases are LiveBench (http://bioinfo.pl/LiveBench/) [146] and EVA (http://cubic.bioc.columbia.edu/eva/) [147]. The evaluation is updated automatically when sequences of newly available protein structures in PDB are sent to the servers and their predictions are collected. The predictions are then compared to the experimental structures automatically and the results are published on the Web pages. Over time, the two servers have accumulated prediction results for a large number of proteins with various prediction methods and they provide useful information to developers as well as users of these prediction tools.

6. IDENTIFICATION OF REMOTE HOMOLOGS OF SARS-COV GENES

Here we show an example using threading to identify remote homologs of SARS-CoV genes. We applied threading in conjunction with other computational methods to predict protein structure and identify homologs. Using PROSPECT, we performed a global analysis of the structural folds and homologs for the SARS-CoV proteins and a detailed study of the S, M, and N-aminopeptidase proteins [148].

6.1. Overview of SARS and SARS-CoV

In 2003, a new unusual pneumonia, Severe Acute Respiratory Syndrome (SARS), attacked us as “a new apocalyptic horseman” [149], which was caused by a novel coronavirus, SARS-CoV [151]. After the first case of this disease was identified in November 2002 in Guangdong Province, China, this disease spread to more than 32 countries and areas around the world. Until August 7, 2003, 8,422 persons worldwide have been infected by SARS, with the vast majority occurred in Mainland China (5,327 infected; 349 deaths), Hong Kong (1,755; 300), Taiwan (665; 180), Canada (251, 41), Singapore (238; 33), and Vietnam (63; 5). There were about 33 cases but no deaths in the United States (see the “WHO SARS case summary”, http://www.who.int /csr/sars/country/en/country2003_08_15.pdf). This disease resulted in mortality of 11% in general. Mortality in persons older than 60 years was reported to be more than 40% [151]. In addition, the economic loss caused by SARS reached billions of dollars. The SARS has been gradually under control since July 2003 although occasional cases are still witnessed around the world. However, we still face the potential challenges from possible future occurrence of this disease, which could be extraordinarily severe.

SARS-CoV is a novel coronavirus, which is similar in genome organization but distantly related to previously characterized coronaviruses in gene sequences [152-154]. Among the identified open reading frames (ORFs), replicase ORF1ab, spike [S], envelope [E], membrane [M] and nucleopcapsid [N] are found in other known coronaviruses with a conserved genome organization. In addition to this common genome organization, this novel virus also has a number of nonstructural proteins with unknown functions [152,153,155].

6.2. Computational Survey of All of ORFs in SARS-CoV

Knowledge of structures and functions of the proteins in SARS-CoV are crucial in understanding the SARS disease. Although several sequence analyses have been published [152,153,156], there has been no comprehensive structural and functional analysis for SARS-CoV. In particular, the amino acid sequence homology between SARS-CoV and any other known coronaviruses are generally less than 40-50% [152]. This finding suggests that SARS-CoV has gone through a substantial evolution from other known coronaviruses. This novel coronavirus may possess many unique unknown features. A step towards characterizing the genes in

SARS-CoV is an in-depth computational analysis of protein structure and function. We used the PROSPECT pipeline to survey the 11 Open Reading Frames (ORFs) in SARS-CoV strain Urbani (GenBank ID: 30027617), one of the first SARS-CoV genomes [152]. Table **3** shows part of our survey results, which give the possible signal peptide and trans-membrane regions for each ORF.

6.3. Structural Prediction of S, M and N-Aminopeptidase in SARS-CoV

We applied PROSPECT to predict the structures for the M protein as well as the S1 and S2 domains. (Table **4**) summarizes the structure prediction results. The prediction results can be evaluated through Z-scores. As described in (Table **2**), if the Z-score is above 10, the prediction is highly confident, and typically no manual assessment is needed. This is the case for the S2 domain. If the Z-score is less than 6, the prediction confidence level is low and manual analyses are necessary. For the M protein, our manual analyses indicated a good confidence level that the M protein adopts structural templates of 1boy or 1edha, both of which share the same structural fold, i.e., immunoglobulin (Ig)-like beta sandwich. Initial manual analyses did not yield a confident structure prediction for the S1 domain, and further studies are needed.

Interestingly, both the S2 domain and the M protein are predicted to adopt the fold of Ig-like beta sandwich (see Figs. **3A, 3B**). The structural similarity suggests that the S2 domain and the M protein may be evolutionarily related through gene fusion and duplication, although their sequences do not have significant similarity anymore after the long period of evolution. Such a phenomenon often occurs among the proteins related to the same biological pathway [85]. Our results might explain how the M protein interacts with the S2 domain, for virus assembly; since the S2 domain with the fold of Ig-like beta sandwich can interact with the S1 domain, the M protein with the same fold can probably interact with the S1 domain in the mode. This suggests that the S1 domain may act as an on-off switch between the S2 domain and the M protein. Such a mechanism may suggest that the M protein could be also involved in the virus-host cell interaction. After we made the prediction, we found that our suggestion was supported by a recent study in the murine hepatitis coronavirus study, which showed the glycosylation of the M protein affected the interferogenic capacity of the virus [157].

N-aminopeptidases in many organisms act as a cell surface receptor for coronaviruses including TGEV, FIPV, FeCV, Human-CoV, and PRCV [158,159]. It has been shown that the N-aminopeptidase interacts with the TGEV S

Table 3. A Computational Survey of All the ORFs in SARS-CoV

ORF	Protein length	Swiss-Prot entry	Signal P (start…end, *p*)	Trans-membrane
AAP13450 (X5)	84	-	-	Soluble protein
AAP13449 (X4)	122	-	15…16, 0.637	Membrane protein Secondary helix: 1-23; Primary helix: 96-117
AAP13448 (X3)	63	-	39…40, 0.106	Membrane protein Primary helix: 12-34
AAP13447 (X2)	154	-	-	Soluble protein
AAP13446 (X1)	274	1	61…62, 0.435	Membrane protein Primary helices: 40-62, 77-99; Secondary helices: 108-130
AAP13445 (N)	422	18	-	Soluble protein
AAP13444 (M)	221	12	39…40, 1.000	Membrane protein Primary helices: 46-68, 78-100; Secondary helices: 14-36
AAP13443 (E)	76	-	43…44, 0.880	Membrane protein Primary helices: 11-33, 37-59
AAP13441 (S)	1,255	18	13…14, 0.421	S1 domain: Soluble protein S2 domain: Membrane protein Primary helix: 531-553
AAP13440 (non-structural polyprotein)	2,695	9	-	Soluble protein
AAP13439 (non-structural polyprotein)	4,382	2	-	16 helices

The different columns in the table show the gene identification/gene name, number of amino acids, the number of homologs in SwissProt, the cleavage site of the predicted signal peptide (two boundary residues and prediction confidence), and predicted trans-membrane segments (primary helix means that the helix is stable in membrane by itself; the secondary helix requires interacting other trans-membrane helix/helices to keep it stable in membrane).

Table 4. Structural Analyses of the M and S Proteins, and the N-Aminopeptidase

PDB template (SARS-CoV ORF)	Z score	Class	Fold	Family	Function
1vfaa (S2)	15.45	All beta protein	Ig-like beta sandwich	V set domains (antibody variable domain-like)	acts as mouse monoclonal antibody.
1boy (M)	<6	All beta protein	Ig-like beta sandwich	Fibronectin type III	plays a role in initiating the cell- surface assembly and propagation of the coagulation protease cascade.
1edha (M)	<6	All beta protein	Ig-like beta sandwich	Cadherin	Cadherins are cell adhesion proteins interacting with themselves in a homophilic manner in order to connect cells.
1has6 (aminopeptidase)	73.62	Alpha/beta Protein	Zincin-like	Leukotriene A4 hydrolase catalytic domain	hydrolyzes an epoxide moiety of leukotriene A4 to leukotriene B4. The enzyme also has some peptidase activity.

The columns in the table indicate the PDB code of the structural template (and chain name in the fifth letter if available), the z-score estimated from PROSPECT for the prediction, and the class, fold, family, and function of the predicted structural template. The sequence identity between the query SARS protein and template sequence is below 25% in all cases.

protein in a specific manner [158]. However, the receptors may be different even for the coronaviruses of the same hosts. Human-CoV -229E uses hAPN as the virus receptor whereas Human-CoV-OC43 uses MHC I as its receptor [160,161,]. Bovine coronavirus (BCV) uses 9-O-actylneuraminic acid as its receptor [162], and murine coronavirus use CEACAM as its receptor [163].

The aminopeptidases are a group of universal peptidases with various functions [164-166]. For example, besides functioning in the cell adhesion and amino acid scavenging, this enzyme can serve as the receptor of Human CoV 229E as stated above. Although it is possible that SARS-CoV may utilize other receptors, the N-aminopeptidase or a similar structural fold of N-aminopeptidase might also be the receptor of SARS-CoV [154]. Before predicting the human receptors for SARS-CoV systematically, we first predicted the structure of N-aminopeptidase as a basis for understanding its interaction with the S protein. As shown in (Table **4**), the structure prediction for N-aminopeptidase has a high Z score, which indicates a good confidence of the structure prediction. The graphical view of the predicted structural fold is shown in Fig. **3C**. Independent of our study, Spiga *et al.* [167] also predicted the structural of Spike protein of SARS-CoV. They predicted the S2 is an Ig-like beta sandwich structure with the template of *C. botulinum* neurotoxin B (PDB code: 1g9d), which is similar to our result. Fig. **4** is the template of *C. botulinum* neurotoxin B.

6.4. Structural Prediction of ORF-X2 in SARS-CoV

Unlike any other coronavirus, SARS-CoV has 5-11 novel open reading frames. Most recently, the largest of ORF in SARS (U274, X1 in Table **3**) has been expressed and found to be involved in the apoptosis via a caspase-dependent pathway [168,169]. However, the functions of all the other ORFs are still not known. Here we use ORF-X2 as an example for remote homolog identification.

We applie170], and GenTHREADER [170] to identify the remote homolog for ORF-X2. Both PROSPECT and

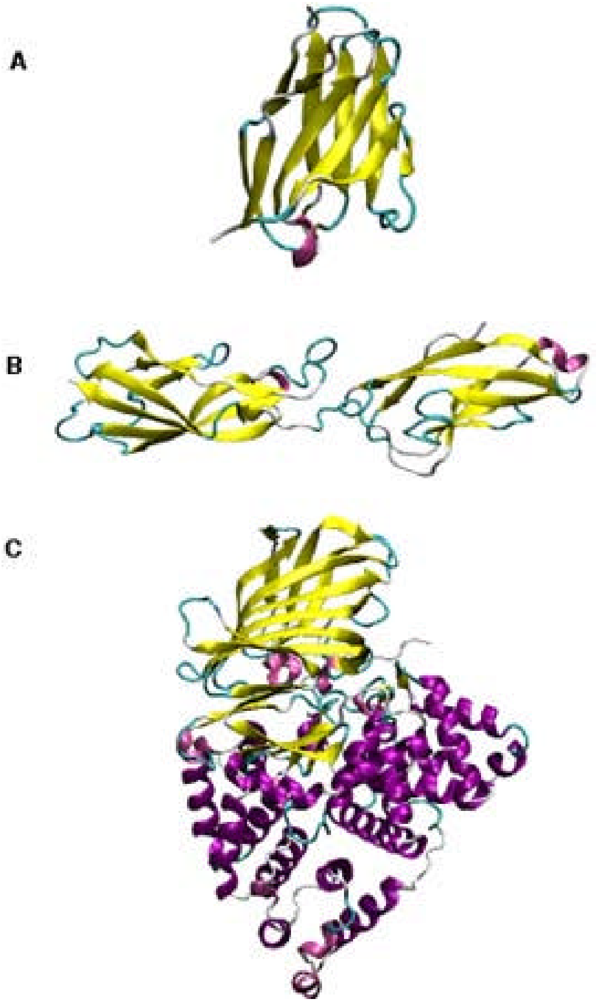

Fig. (3). A. Structural template 1vfa for the S2 domain. B. Structural template 1edha for the M protein. C. Structural template 1hs6a for the N-aminopeptidase.

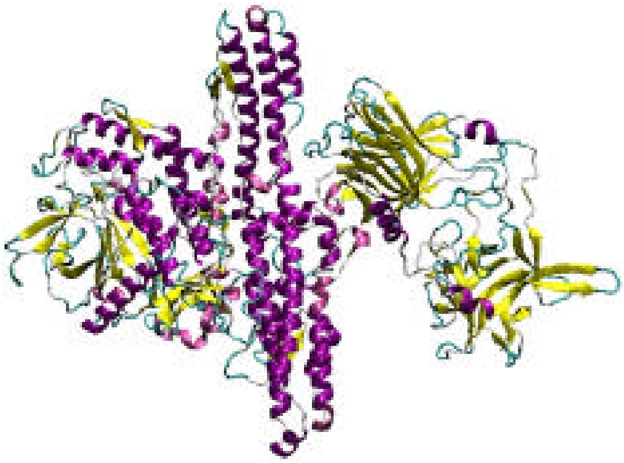

Fig. (4). The 3D structure of *C. botulinum* neurotoxin B (PDB code: 1g9d).

SAM returned the same top hit as Interleukin 2 related homolog (PDB code: 3ink). However, GenTHREADER indicated ORF-X2 may be a remote homolg for protein-tyrosine kinase (PDB code: 1k04). These predicted functions of ORF-X2 provide some hypotheses for further experimental verification.

7. DISCUSSIONS

In summary, significant advances have been made for computational identification of protein remote homologs in the past decade. Various methods have pushed our limit to find distantly related homologs that are unidentifiable from simple pairwise sequence comparisons. These methods often utilize the evolutionary information in the sequence database effectively, in particular through building multiple sequence profiles or identifying evolutionary intermediates. Many methods also use protein structural information, including integrating protein secondary structure prediction into the process of sequence comparison, searching through sequence-structure comparisons (threading), and performing structure-structure alignments. More recently, mega-servers using multiple methods to find consensus solutions have been developed. These servers often show significant improvement over any single method. All these developments have a big impact on the field of post-genomic biology, especially for genome annotation, comparative genomics, structural genomics, and functional genomics. Not only computational biologists but also experimentalists benefit tremendously from these tools, which often provide useful information about the structure and function of a protein through its (remote) homologs. The computational results can help develop biological hypothesis for new experiments and also help the interpretation of experimental data.

However, these computational tools should not be used blindly for inferring remote homologies. It should be noted that even when the sequence similarity between two proteins is high, it might not always correspond to homology. There is always a possibility that the sequence similarity was by chance, rather than due to biological relationship. When more sensitive methods for remote-homolog identification are used, the confidence level of a comparison result can be low, and it is not rare that false positive predictions are generated. Also, homology may not imply function conservation. Many remote homologs, especially paralogs, have divergent functions, although their functions are often related in a broader category. To best take advantage of the available computational tools and reduce the chance of wrong prediction, it is important to use multiple tools to check for consensus solutions and differences between various results. It is also important to use other computational approaches [171], such as prediction of signal peptide cleavage sites, subcellular localization, protein domain prediction, prediction of transmembrane helices, and sequence motif prediction, to predict the properties and functions of the proteins so that one can better assess the potential homologous and functional relationships between proteins of interest, as illustrated in our example in Section 6. Furthermore, when high-throughput data (e.g., gene expression data and protein-protein interaction data) are available, it is very useful to utilize these experimental data to confirm and extend the homologous relationship identified from sequence or structure based methods [172,173]. An example of using various methods in conjunction with homolog identification is described in a recent paper [174]. Finally, additional experiments are generally needed to confirm the predictions.

There are still many challenging problems in remote homolog identification and the related research is very active. More sensitive methods are needed for difficult homolog identifications. Still in many cases, people know that homolog of protein X with well characterized function in species A should be present in species B, since species B shows the same phenotype related to protein X as does species A. However, current methods may not be sensitive enough to detect the homolog of protein X in species B. Another challenge is the opposite. Sometimes there are many homologs detected in species X, but it is unknown which one represents the true ortholog. Other than the sensitivity issues, current confidence assessment methods of the homolog identification results need further improvement. Some tools, such as BLAST, PSI-BLAST, and FASTA, have good confidence assessment methods, but they often overestimate statistical significance. Many other tools have primitive assessment methods or no assessment at all. As a result, generally, remote homolog identification has poor prediction specificity, i.e., false positives are frequently predicted. In addition, the current alignment accuracy between remote homologs is typically poor, and there is much room for improvement.

ACKNOWLEDGEMENTS

We are grateful to critical comments and suggestions from two anonymous reviewers. We would like to thank Drs. Nickolai Alexandrov and Ying Xu for helpful discussions. We also like to thank Ognen Duzlevski and Demet Ataman for technical assistance. This work is supported in part by the US Department of Energy's Genomes to Life program (http://www.doegenomestolife.org) under project, “Carbon Sequestration in *Synechococcus* Sp.: From Molecular Machines to Hierarchical Modeling” (www.genomes-to-life.org). It was also supported by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, under Contract DE-AC05-00OR22725, managed by UT-Battelle, LLC.

REFERENCES

[1] Bahls, C.; Weitzman, J. and Gallagher, R. **(2003)** *The Scientist 17, Suppl 1,* http://www.the-scientist.com/yr2003/jun/ feature_030602.html.

[2] Fitch, W.M. **(1970)** *Syst Zool. 19,* 99–113.

[3] Koonin, E. V.; Makarova, K. S. and Aravind, L. **(2001)** *Annu. Rev. Microbiol. 55,* 709–742.

[4] Baumberger, N.; Steiner, M.; Ryser, U.; Keller, B. and Ringli, C. **(2003)** *Plant J. 35,* 71–81.

[5] Kobor, M. S.; Venkatasubrahmanyam, S.; Meneghini, M. D.; Gin, J. W.; Jennings, J. L.; Link, A. J.; Madhani, H. D. and Rine, J. **(2004)** *PLoS Biol. 2,* E131.

[6] Russell, R. B. and Ponting, C. P. **(1998)** Protein fold irregularities that hinder sequence analysis. *Curr. Opin. Struct. Biol. 8,* 364–371.

[7] Park, J. and Teichmann, S. A. **(1998)** *Bioinformatics 14,* 144–150.

[8] Corpet, F.; Servant, F.; Gouzy, J. and Kahn, D. **(2000)** *Nucleic Acids Res. 28,* 267–269.

[9] Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W. and Lipman, D. J. **(1990)** *J. Mol. Biol. 215,* 403–410.

[10] Pearson, W. R. and Lipman, D. J. **(1988)** *Proc. Natl. Acad. Sci. U. S. A. 85,* 2444–2448.

[11] Branden, C. and Tooze, J. **(1999)** Introduction to Protein Structure. Garland Publishing, 2nd edition.

[12] Lesk, A. **(2001)** Introduction to Protein Architecture: The Structural Biology of Proteins Oxford University Press.

[13] Needleman, S. B. and Wunsch, C. D. **(1970)** *J. Mol. Biol. 48,* 443–453.

[14] Smith, T. F. and Waterman, M. S. **(1981)** *J. Mol. Biol. 147,* 195–197.

[15] Dayhoff, M.O.; Schwartz, R.M. and Orcutt, B.C. **(1978)** Atlas of Protein Sequence and Structure (National Biomedical Research Foundation, Washington DC.), vol. *5,* pp345–358.

[16] Xu, W. and Miranker, D. P. **(2004)** *Bioinformatics 20,* 1214–1221.

[17] Gonnet, G. H.; Cohen, M. A. and Benner SA. **(1992)** *Science 256,* 1443–1445.

[18] Jones, D. T.; Taylor, W. R. and Thornton, J. M. **(1992)** *Comput. Appl. Biosci. 8,* 275–282.

[19] Henikoff, S. and Henikoff, J.G. **(1992)** *Proc. Natl. Acad. Sci. U.S.A. 89,* 10915–10910.

[20] Veerassamy, S.; Smith, A. and Tillier, E. R. **(2003)** *J. Comput. Biol. 10,* 997–1010.

[21] Vingron, M. **(1996)** *Curr. Opin. Struct. Biol. 6,* 346–352.

[22] Karlin, S. and Altschul, S. F. **(1990)** *Proc. Natl. Acad. Sci. U.S.A. 87,* 2264–2268.

[23] Karlin, S.; Dembo, A. and Kawabata, T. **(1999)** *Ann. Statistics 18,* 571–581.

[24] Olsen, R.; Bundschuh, R. and Hwa, T. **(1999)** *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 211–222.

[25] 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.

[26] Lopez, R.; Silventoinen, V.; Robinson, S.; Kibria, A. and Gish, W. **(2003)** *Nucleic Acids Res. 31,* 3795–3798.

[27] Tatusov, R. L.; Altschul, S. F. and Koonin, E. V. **(1994)** *Proc. Natl. Acad. Sci. U.S.A. 91,* 12091–12095.

[28] Schaffer, A. A.; Wolf, Y. I.; Ponting, C. P.; Koonin, E. V.; Aravind, L. and Altschul, S. F. **(1999)** *Bioinformatics 15,* 1000–1011.

[29] Aravind, L. and Koonin, E. V. **(1999)** *J. Mol. Biol. 287,* 1023–1040.

[30] Krogh, A.; Brown, M.; Mian, I. S.; Sjolander, K. and Haussler, D. **(1994)** *J Mol Biol. 235,* 1501–1531.

[31] Eddy, S. R. **(1996)** *Curr. Opin. Struct. Biol. 6,* 361–365.

[32] Madera, M. and Gough, J. **(2002)** *Nucleic Acids Res. 30,* 4321–4328.

[33] Eddy, S. R. Profile hidden Markov models. **(1998)** *Bioinformatics 14,* 755–763.

[34] Karplus, K.; Barrett, C. and Hughey, R. **(1998)** *Bioinformatics 14,* 846–856.

[35] Grundy, W. N.; Timothy, L. B.; Charles, P. E. and Baker, M. E. **(1997)** *Comput. Appl. Biosci. 13,* 397–406.

[36] Loytynoja, A.; Milinkovitch, M. C. **(2003)** *Bioinformatics 19,* 1505–1513.

[37] Wistrand, M.; Sonnhammer, E. L. **(2004)** *J. Mol. Biol. 338,* 847–854.

[38] Spang, R.; Rehmsmeier, M. and Stoye, J. **(2002)** *J. Comput. Biol. 9,* 747–760.

[39] Morgenstern, B.; Frech, K.; Dress, A. and Werner, T. **(1998)** *Bioinformatics* **14**, 290–294.

[40] Grundy, W. N. and Bailey, T. L. **(1999)** *Bioinformatics 15,* 463–470.

[41] Yona, G. and Levitt, M. **(2002)** *J. Mol. Biol. 315,* 1257–1275.

[42] von Ohsen, N.; Sommer, I. and Zimmer, R. **(2003)** *Pac. Symp. Biocomput.* 252–263.

[43] Panchenko, A. R. **(2003)** *Nucleic Acids Res. 31,* 683–689.

[44] Sadreyev, R. I.; Baker, D. and Grishin, N. V. **(2003)** *Protein Sci. 12,* 2262–2272.

[45] Sadreyev, R. and Grishin, N. **(2003)** *J. Mol. Biol. 326,* 317–336.

[46] Edgar, R. C. and Sjolander, K. **(2004)** *Bioinformatics 20,* 1309–1318.

[47] Soding, J. **(2005)** *Bioinformatics 21,* 951–960.

[48] Tomii, K. and Akiyama, Y. **(2004)** *Bioinformatics 20,* 594–595.

[49] Tang, C. L.; Xie, L.; Koh, I. Y. Y.; Posy, E. A. and Honig, B. **(2003)** *J. Mol. Biol.* **334**, 1043–1062.

[50] Zhou, H. and Y. Zhou. **(2005)** *Proteins* **58**, 321–328.

[51] Rehmsmeier, M. and Vingron, M. **(2001)** *Proteins 45,* 360–3671.

[52] Qian, B. and Goldstein, R. A. **(2003)** *Proteins 52,* 446–453.

[53] Qian, B. and Goldstein, R. A. **(2004)** *Bioinformatics 20,* 2175–2180.

[54] Cai ,W.; Pei, J.; Grishin, N. V. **(2004)** *BMC Evol Biol 4,* 33.

[55] Edwards, R.J. and Shields, D. C. **(2004)** *BMC Bioinformatics 5,* 123

[56] Park, J.; Karplus, K.; Barrett, C.; Hughey, R.; Haussler, D.; Hubbard, T. and Chothia, C. **(1998)** *J. Mol. Biol. 284,* 1201–1210.

[57] Gerstein, M. **(1998)** *Bioinformatics 14,* 707–714.

[58] Zhu, J., Luthy, R. and Lawrence, C. E. **(1999)** *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 297–305.

[59] John, B. and Sali, A. **(2004)** *Protein Sci. 13,* 54–62.

[60] Geourjon, C.; Combet, C.; Blanchet, C. and Deleage, G. **(2001)** *Protein Sci. 10,* 788–797.

[61] Ginalski, K. and Rychlewski, L. **(2003)** *Proteins* 53 Suppl *6,* 410–417.

[62] Geourjon, C.; Combet, C.; Blanchet, C. and Deleage, G. **(2001)** *Protein Sci. 10,* 788–797.

[63] Wallqvist, A.; Fukunishi, Y.; Murphy, L. R. and Levy, R. M. **(2000)** *Bioinformatics 16,* 988–1002.

[64] Kann, M. G.; Thiessen ,P. A.; Panchenko, A. R.; Schaffer, A. A.; Altschul, S. F. and Bryant, S. H. **(2005)** *Bioinformatics 21,* 1451–1456..

[65] Zhang, Z.; Kochhar, S. and Grigorov, M. G. **(2003)** *Protein Sci. 14,* 431–44.

[66] Cristiani, N. and Shawe-Taylor J. **(2000)** *An introduction to support vector machines.* Cambridge, UK: Cambridge University Press.

[67] Prilusky, J.; Hansen, D.; Pilpel, T. and Safran, M. **(1999)** *The Protein Hydrophilicity/Hydrophobicity Search and Comparison Server.* Weizmann Institute of Science, Rehovot, Israel.

[68] Hobohm, U. and Sander, C. **(1995)** *J. Mol. Biol. 251,* 390–399.

[69] Pellegrini, M. and Yeates, T. O. **(1999)** *Proteins 37,* 278–283.

[70] Karwath, K. and King, R. D. **(2002)** *BMC Bioinformatics 3,* 11.

[71] Gasteiger, E.; Jung, E. and Bairoch, A. **(2001)** *Curr. Issues Mol. Biol. 3,* 47–55.

[72] Muggleton, S. **(1990)** *Inductive logic programming.* In: Proceedings of the first conference on algorithm learning theory, Tokyo, Ohmsha.

[73] Hou, Y.; Hsu, W.; Lee, M. L. and Bystroff, C. **(2004)** *Proteins 57,* 518–530.

[74] Bowie, J. U.; Luthy, R. and Eisenberg, D. **(1991)** *Science 253,* 164–170.

[75] Sippl, M. J. and Weitckus, S. **(1992)** *Proteins 13,* 258–271.

[76] Jones, D. T.; Taylor, W. R. and Thornton, J. M. **(1992)** *Nature 358,* 86–89.

[77] Xu, Y.; Xu, D. and Uberbacher, E. C. **(1998)** *J. Comput. Biol. 5,* 597–614.

[78] Finkelstein, A. V. and Ptitsyn, O. B. **(1987)** *Prog. Biophys. Mol. Biol. 50,* 171–190.

[79] Webster, D. M. **(2000)** Protein structure prediction: methods and protocols. Humana Press, 1st edition,

[80] Tsigelny, I.F. **(2002)** *Protein structure prediction: bioinformatic approach.* International University Line publishers (IUL), La Jolla, CA.

[81] Smith, T. F.; Lo Conte, L.; Bienkowska, J.; Gaitatzes, C.; Rogers, R. G. Jr. and Lathrop, R. **(1997)** *J. Comput. Biol. 4,* 217–225.

[82] Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N. and Bourne, P. E. **(2000)** *Nucleic Acids Res. 28,* 235–242.

[83] Orengo, C. A.; Michie, A. D.; Jones, S.; Jones, D. T.; Swindells, M. B. and Thornton, J. M. **(1997)** *Structure 5,* 1093–1108.

[84] Holm, L. and Sander, C. **(1996)** *Science 273,* 595–603.

[85] Murzin, A. G.; Brenner, S. E.; Hubbard, T. and Chothia, C. **(1995)** *J. Mol. Biol. 247,* 536–540.

[86] Fischer, D.; Elofsson, A.; Bowie, J. U. and Eisenberg, D. **(1996)** In: *Biocomputing: Proceedings of the 1996 Pacific Symposium,* (Hunter, L. and Klein, T., eds) pp. 300–318. World Scienti.c Publishing Co. Singapore.

[87] Abagyan, R. A. and Batalov, S. **(1997)** *J. Mol. Biol. 273,* 355–368.

[88] Alexandrov, N. N.; Nussinov, R. and Zimmer, R. M. **(1996)** *Pac. Symp. Biocomput.* 53–72.

[89] Binderwald, E.; Cestaro, A.; Hesser, J.; Heiler, M. and Tosatto, S. C. E. **(2003)** *Protein Eng. 16,* 785–789.

[90] Rost, B. **(1995)** Proc. Int. Conf. Intell. Syst. Mol. Biol. *3,* 314–321.

[91] Milburn, D.; Laskowski, R. A. and Thornton, J. M. **(1998)** *Protein Eng. 11,* 855–859.

[92] Xu, Y. and Xu, D. **(2000)** *Proteins 40,* 343–354.

[93] Lathrop, R. H. **(1994)** *Protein Eng. 7,* 1059–1068.

[94] Bryant, S. H. and Lawrence, C. E. **(1993)** *Proteins 16,* 92–112.

[95] Lathrop, R. H. and Smith, T. F. **(1996)** *J. Mol. Biol. 255,* 641–665.

[96] Bryant, S. H. and Altschul, S. F. **(1995)** *Curr. Opin. Struct. Biol. 5,* 236–244.

[97] Crawford, O. H. **(1999)** *Bioinformatics 15,* 66–71.

[98] Xu, Y.; Xu, D. and Uberbacher, E. C. **(1998)** *J. Comput. Biol. 5,* 597–614.

[99] Xu, J.; Li, M.; Kim, D. and Xu, Y. **(2003)** *J. Bioinformatics and Comp. Biol. 1,* 95-117.

[100] Xu, J. and Li, M. **(2003)** *Proteins* 53 Suppl *6,* 579–584.

[101] Flockner, H.; Braxenthaler, M.; Lackner, P.; Jaritz, M.; Ortner, M. and Sippl, M. J. **(1995)** *Proteins 23,* 376–386.

[102] Jones, D. T. **(1999)** *J. Mol. Biol. 287,* 797–815.

[103] Xu, Y.; Xu, D. and Olman, V. **(2002)** *Statistica Sinica* (special issue in bioinformatics) *12,* 159–177.

[104] Lundstrom, J.; Rychlewski, L.; Bujnicki, J. and Elofsson, A. **(2001)** *Protein Sci. 10,* 2354–2362.

[105] Shah, M.; Passovets, S.; Kim, D.; Ellrott, K.; Wang, L.; Vokler, I.; LoCascio, P.; Xu, D. and Xu, Y. **(2003)** *Bioinformatics 19,* 1985–1996.

[106] Kim, D. E.; Chivian, D. and Baker D. **(2004)** *Nucleic Acids Res.* 32(Web Server issue), W526–531.

[107] Kim, D.; Xu, D.; Guo, J. T.; Ellrott, K. and Xu, Y. **(2003)** *Protein Eng. 16,* 641–650.

[108] Zhang, B.; Rychlewski, L.; Pawlowski, K.; Fetrow, J. S.; Skolnick, J. and Godzik, A. **(1999)** *Protein Sci. 8,* 1104–1115.

[109] Wallace, A. C.; Laskowski, R.A. and Thornton, J. M. **(1996)** *Protein Science 5,* 1001–1013.

[110] Bucher, P. and Bairoch, A. **(1994)** *Proc. Int. Conf. Intell. Syst. Mol. Biol. 2,* 53–61.

[111] Hulo, N.; Sigrist, C. J.; Le Saux, V.; Langendijk-Genevaux, P. S.; Bordoli, L.; Gattiker, A.; De Castro, E.; Bucher, P. and Bairoch, A. **(2004)** *Nucleic Acids Res. 32,* D134–7.

[112] Attwood, T. K. and Beck, M. E. **(1994)** *Protein Eng. 7,* 841–848.

[113] Attwood, T. K.; Bradley, P.; Flower, D. R.; Gaulton, A.; Maudling, N.; Mitchell, A. L.; Moulton, G.; Nordle, A.; Paine, K.; Taylor, P.; Uddin, A. and Zygouri, C. **(2003)** *Nucleic Acids Res. 31,* 400–402.

[114] Henikoff, J. G. and Henikoff, S. **(1996)** *Methods Enzymol. 266*, 88–105.

[115] Xu, Y.; Xu, D.; Crawford, O. H.; Einstein, L. F; Uberbacher, E.; Unseren, M. A. and Zhang, G. **(1999)** *Protein Eng. 12,* 899–907.

[116] Henikoff, S. and Henikoff, J. G. **(1994)** *Genomics 19,* 97–107.

[117] Zhang, C. and Kim, S. H. **(2003)** *Curr. Opin. Chem. Biol. 7,* 28–32.

[118] Brenner, S. E. **(2001)** *Nat. Rev. Genet. 2,* 801–809.

[119] Holm, L. and Sander, C. **(1993)** *J. Mol. Biol. 233,* 123–138.

[120] Shindyalov IN and Bourne PE **(1998)** *Protein Engineering* **11**(9) 739–747.

[121] Shyu, C. R.; Chi, P. H.; Scott, G. and Xu, D. **(2004)** *Nucleic Acid Res. 32,* W572–W575.

[122] Cohen, B. A.; Mitra, R. D.; Hughes, J. D. and Church, G. M. **(2000)** *Nat. Genet. 26,* 183–186.

[123] Volkmuth, W. and Alexandrov, N. **(2002)** *Pac. Symp. Biocomput.* 247–258.

[124] Li, H.; Pellegrini, M. and Eisenberg, D. **(2005)** *Nat Biotechnol. 23,* 253–260.

[125] Chen, X.; Su, Z.; Dam, P.; Palenik, B.; Xu, Y. and Jiang, T. **(2004)** *Nucleic Acids Res. 32,* 2147–2157.

[126] Bockhorst, J.; Craven, M.; Page, D.; Shavlik, J. and Glasner, J. **(2003)** *Bioinformatics 19,* 1227–1235.

[127] Zheng, Y.; Szustakowski, J.D.; Fortnow, L.; Roberts, R.J. and Kasif, S. **(2002)** *Genome Res. 12,* 1221–1230.

[128] Ermolaeva, M. D.; White, O. and Salzberg, S. L. **(2001)** *Nucleic Acids Res. 29,* 1216–1221.

[129] Kelley, B. P. *et al.* (**2003**) *Proc. Natl. Acad. Sci. U. S. A. 100,* 11394–11399.

[130] Jimenez, J. L.; Mitchell, M. P. and Sgouros, J. G. **(2003)** *Genome Biol. 4,* R4.

[131] Simons, A. H.; Dafni, N.; Dotan, I.; Oron, Y. and Canaani, D. **(2001)** *Nucleic Acids Res. 29,* E100.

[132] Tong, A. H.; Evangelista, M.; Parsons, A. B.; Xu, H.; Bader, G. D.; Page, N.; Robinson, M.; Raghibizadeh, S.; Hogue, C. W.; Bussey, H.; Andrews, B.; Tyers, M. and Boone, C. **(2001)** *Science 294,* 2364–2368.

[133] Brenner, S. E.; Chothia, C. and Hubbard, T. J. **(1998)** *Proc. Natl. Acad. Sci. U. S. A. 95,* 6073–6078.

[134] Sauder, J. M.; Arthur, J. W. and Dunbrack, R. L. Jr. **(2000)** *Proteins* 40(1):6–22.

[135] Venclovas, C. **(2003)** *Proteins* 53 Suppl *6,* 380–388.

[136] Tramontano, A. and Morea, V. **(2003)** *Proteins* 53 Suppl. *6,* 352–368.

[137] Ginalski, K. and Rychlewski, L. **(2003)** *Proteins* 53 Suppl *6,* 410–417.

[138] Xu, D.; Kim, D.; Dam, P.; Shah, M.; Uberbacher, E. C. and Xu, Y. **(2003)** In *Genetic Engineering, Principles and Methods,* edited by Setlow, J. K. Kluwer Academic/Plenum Publishers, New York. 269–293.

[139] Shah, M.; Passovets, S.; Kim, D.; Ellrott, K.; Wang, L.; Vokler, I.; LoCascio, P.; Xu, D. and Xu, Y. **(2003)** *Bioinformatics 19,* 1985–1996.

[140] Ginalski, K.; Elofsson, A.; Fischer, D. and Rychlewski, L. **(2003)** *Bioinformatics 19,* 1015–1018.

[141] Guo, J. T.; Ellrott, K.; Chung, W. J.; Xu, D.; Passovets, S. and Xu, Y. **(2004)** *Nucleic Acid Res. 32,* W522–CW525.

[142] Tang, C. L.; Xie, L.; Koh, I. Y.; Posy, S.; Alexov, E. and Honig, B. **(2003)** *J. Mol. Biol. 334,* 1043–1062.

[143] Raval, A.; Ghahramani, Z. and Wild, D. L. **(2002)** *Bioinformatics 18,* 788–801.

[144] Fischer, D.; Rychlewski, L.; Dunbrack, R. L. Jr.; Ortiz, A. R. and Elofsson, A. **(2003)** *Proteins* 53 Suppl *6,* 503–516.

[145] Leplae, R. and Hubbard, T. J. **(2002)** *Bioinformatics 18,* 494–495.

[146] Rychlewski, L.; Fischer, D. and Elofsson, A. **(2003)** *Proteins* 53 Suppl *6,* 542–547.

[147] Koh, I. Y.; Eyrich, V. A.; Marti-Renom, M. A.; Przybylski, D.; Madhusudhan, M. S.; Eswar, N.; Grana, O.; Pazos, F.; Valencia, A.; Sali, A. and Rost, B. **(2003)** *Nucleic Acids Res. 31,* 3311–3315.

[148] Wan, X.-F.; Ataman, D. and Xu, D. **(2004)** In *Progress in Bioinformatics*, Nova Science Publishers, Inc. In press.

[149] Maki, D. G. **(2003)** SARS: 1918 revisited? *Mayo Clin. Proc. 78,* 813–816.

[150] Ksiazek, T. G.; Erdman, D.; Goldsmith, C. S.; Zaki, S. R.; Peret, T.; Emery, S.; Tong, S.; Urbani, C.; Comer, J. A.; Lim, W. *et al.* SARS Working Group. **(2003)** *N. Engl. J. Med. 348,* 1953–1966.

[151] Donnelly, C. A.; Ghani, A. C.; Leung, G. M.; Hedley, A. J.; Fraser, C.; Riley, S.; Abu-Raddad, L. J.; Ho, L. M.; Thach, T. Q.; Chau, P. *et al.* **(2003)** *Lancet 361,* 1761–1766.

[152] Rota, P. A.; Oberste, M. S.; Monroe, S. S.; Nix, W. A.; Campagnoli, R.; Icenogle, J. P.; Penaranda, S.; Bankamp, B.; Maher, K.; Chen, M. H. *et al.* **(2003)** *Science 300,* 1394–1399.

[153] Marra, M. A.; Jones, S. J.; Astell, C. R.; Holt, R. A.; Brooks-Wilson, A.; Butterfield, Y. S.; Khattra, J.; Asano. J. K.; Barber, S. A.; Chan, S. Y. *et al.* **(2003)** *Science 300,* 1399–1404.

[154] Yu, X. J.; Luo, C.; Lin, J. C.; Hao, P.; He, Y. Y.; Guo, Z. M.; Qin, L.; Su, J.; Liu, B. S.; Huang, Y. *et al.* **(2003)** *Acta Pharmacol. Sin. 24,* 481–488.

[155] Zeng, F. Y.; Chan, C. W.; Chan, M. N.; Chen, J. D.; Chow, K. Y.; Hon, C. C.; Hui, K. H.; Li, J.; Li, V. Y. *et al.* **(2003)** *Exp. Biol. Med. (Maywood) 228,* 866–873.

[156] Ruan, Y. J.; Wei, C. L.; Ee, A. L.; Vega, V. B.; Thoreau, H.; Su, S. T.; Chia, J. M.; Ng, P.; Chiu, K. P.; Lim, L. *et al.* **(2003)** *Lancet 361,* 1779–1785.

[157] de Haan C. A.; Stadler, K.; Godeke, G. J.; Bosch, B. J. and Rottier, P. J. **(2004)** *J. Virol. 78,* 6048–6054.

[158] Delmas, B.; Gelfi, J.; L'Haridon, R.; Vogel, L. K.; Sjostrom, H.; Noren, O. and Laude, H. **(1992)** *Nature 357,* 417–420.

[159] Tresnan, D. B. and Holmes, K. V. **(1998)** *Adv. Exp. Med. Biol. 440,* 69–75.

[160] Yeager, C. L.; Ashmun, R. A.; Williams, R. K.; Cardellichio, C. B.; Shapiro, L. H.; Look, A. T. and Holmes, K. V. **(1992)** *Nature 357,* 420–422.

[161] Collins, A. R. **(1993)** *Immunol. Invest. 22,* 95–103.

[162] Schultze, B. and Herrler, G. **(1993)** *Adv. Exp. Med. Biol. 342,* 299–304.

[163] Tan, K.; Zelus, B. D.; Meijers, R.; Liu, J. H.; Bergelson, J. M.; Duke, N.; Zhang, R.; Joachimiak, A.; Holmes, K. V. and Wang, J. H. **(2002)** *EMBO J. 21,* 2076–2086.

[164] Riemann, D.; Kehlen, A. and Langner, J. **(1999)** *Immunol. Today 20,* 83–88.

[165] Wan, X.; Branton, S. L.; Hughlett, M. B.; Hanson, L.A. and G. T. Pharr. **(2004)** *Int. J. of Poultry Sci. 3,* 70–74.

[166] Wan, X.; Branton, S. L.; Hanson, L. A. and Pharr, G. T. **(2004)** *Curr. Microbiol. 48,* 32–38.

[167] Spiga, O.; Bernini, A.; Ciutti, A.; Chiellini, S.; Menciassi, N.; Finetti, F.; Causarono, V.; Anselmi, F.; Prischi, F. and Niccolai, N. **(2003)** *Biochem. Biophys. Res. Commun. 310,* 78–83

[168] Tan, Y. J.; Fielding, B. C.; Goh, P. Y.; Shen, S., Tan, T. H.; Lim, S. G. and Hong, W. **(2004)** *J. Virol. 78,* 14043–14047.

[169] Tan, Y. J.; Teng, E.; Shen, S.; Tan, T. H.; Goh, P. Y.; Fielding, B. C.; Ooi, E. E.; Tan, H. C.; Lim, S. G. and Hong, W. **(2004)** *J. Virol. 78,* 6723–6734.

[170] Jones, D. T. **(1999)** *J. Mol. Biol. 287,* 797–815.

[171] Xu, D.; Xu, Y. and Uberbacher, E. C. **(2000)** *Curr. Protein Pept. Sci. 1,* 1–21.

[172] Chen, Y. and Xu, D. **(2003)** *Curr. Protein Pept. Sci. 4,* 159–181.

[173] Chen, Y.; Joshi, T.; Xu, Y. and Xu, D. **(2003)** Proceeding of the 3rd IEEE Symposium on Bioinformatics and Bioengineering 18–*25,* IEEE/CS Press.

[174] Qu, K.; Lu, Y.; Lin, N.; Singh, R.; Xu, X.; Payan, D. G. and Xu, D. **(2004)** *Curr. Med. Chem. 11,* 569–582.

[175] Peterson, J.D.; Umayam, L.A.; Dickinson, T.; Hickey, E.K. and White, O. **(2001)** *Nucleic Acids Res. 29,* 123–125.

[176] Okayama, T.; Tamura, T.; Gojobori, T.; Tateno, Y.; Ikeo, K.; Miyazaki, S.; Fukami-Kobayashi, K. and Sugawara, H. **(1998)** *Bioinformatics 14,* 472–478.

[177] Shakhnovich, B. E. *et al.* **(2003)** *BMC Bioinformatics 4,* 34.

[178] Rebhan, M.; Chalifa-Caspi, V.; Prilusky, J. and Lancet, D. **(1997)** *GeneCards: encyclopedia for genes, proteins and diseases.* Weizmann Institute of Science, Bioinformatics Unit and Genome Center (Rehovot, Israel).

[179] Harris, M. A. *et al.* **(2004)** *Nucleic Acids Res.* 32: D258–261

[180] Mural, R. J.; Parang, M.; Shah, M.; Snoddy, J. and Uberbacher, E. C. **(1999)** *Trends Genet. 15,* 38–39.

[181] Kanehisa, M. and Goto, S. **(2000)** *Nucleic Acids Res. 28,* 27–30.

[182] Eppig, J. T. et al. **(2005)** *Nucleic Acids Res.* 33(Database issue), D471–475.

[183] Bleasby, A.J.; Akrigg, D. and Attwood, T.K. **(1994)** *Nucleic Acids Res. 22,* 3574–3577.

[184] McGarvey, P.B.; Huang, H.; Barker, W.C.; Orcutt, B.C.; Garavelli, J.S.; Srinivasarao, G.Y.; Yeh, L.L.; Xiao, C. and Wu, C. **(2000)** *Bioinformatics 16,* 290–291.

[185] Kissinger, J.C., *et al.* (**2002**) *Nature 419,* 490-492

[186] Christoffels, A.; van Gelder, A.; Greyling, G.; Miller, R.; Hide, T. and Hide, W. **(2001)** *Nucleic Acids Res. 29,* 234–238.

[187] Meinel, T.; Krause, A.; Luz, H.; Vingron, M. and Staub, E. **(2005)** *Nucleic Acids Res.* 33: Database issue D226–D229.

[188] Zhang, P.; Foerster, H.; Tissier, C.; Mueller, L.; Paley, S.; Karp, P. and Rhee, S.Y. **(2005)** *Physiology*, 138: 27–37.

[189] Lee, Y. *et al.* **(2005)** *Nucleic Acids Res.* 33(Database issue), D71–74.

[190] Boguski MS, Schuler GD **(1995)** *Nature Genetics* 10: 369–371.

[191] Chen, N. *et al.* **(2005)** *Nucleic Acids Res.* 33(Database issue), D383–389.

[192] Dwight, S. S. *et al.* **(2004)** *Brief Bioinform. 5,* 9–22.

[193] Szymanski, M.; Deniziak M.A.; and Barciszewski, J. **(2001)** *Nucleic Acids Res. 29,* 288–290.

[194] Tobes, R. and Ramos, J. L. **(2002)** *Nucleic Acids Res. 30,* 318–321.

[195] Afonnikov, D.A.; Valuev, V.P.; Kashinskaya, Ju. O. and Orlov, Yu. L. **(2000)** *Computational Technologies 5,* S75–S78.

[196] Tatusov, R. L.; Koonin, E.V. and Lipman, D. J. **(1997)** *Science 278,* 631–37.

[197] Michael, H.W.; Fricke, I. ; Doll, N. and Marahiel, M.A. **(2002)** *Nucleic Acids Res. 30,* 375–378.

[198] Gracy, J. and Argos, P. **(1998)** *Bioinformatics 14,* 163–173.

[199] Nelson, M.R.; Thulin, E.; Fagan, P.A.; Forsen, S. and Chazin, W. J. **(2002)** *Protein Sci. 11,* 198–205.

[200] Bairoch, A. **(2000)** *Nucleic Acids Res.* 28:304–305.

[201] Horn, F.; Weare, J.; Beukers, M.W.; Horsch, S.; Bairoch, A.; Chen, W.; Edvardsen, O.; Campagne, F. and Vriend, G. **(1998)** *Nucleic Acids Res.* 1:275–279

[202] Bürglin, T.R. **(1997)** *Nucleic Acids Res. 25,* 4173–4180.

[203] Mulder, N. J. *et al.* **(2005)** *Nucleic Acids Res. 33,* D201–205.

[204] Rawlings, N.D.; Tolle, D.P. and Barrett, A.J. **(2004)** *Nucleic Acids Res.* 32 Database issue, D160–D164.

[205] Brusic, V.; Rudy, G.; Kyne, A.P., and Harrison, L.C. **(1998)** *Nucleic Acids Res. 26,* 368–371

[206] Dellaire G, Farrall R, Bickmore WA. **(2003)** Nucl. Acids Res. 31: 328-330

[207] Gupta, R., Birch, H., Rapacki, K., Brunak, S., and Hansen, J.E. **(1999)** Nucleic Acids Research, 27: 370-372.

[208] Bateman, A. *et al.* **(2004)** *Nucleic Acids Res.* 32: D138–141.

[209] Petretti, C. and Prigent, C. **(2005)** *Biol Cell 97,* 113–118.

[210] Brown, J. W. **(1999)** *Nucleic Acids Res. 27,* 314.

[211] Wu, C., Shivakumar S., and Huang, H. **(1999)**. Nucleic Acids Research, 27 (1), 272–274.

[212] Vlahovicek, K.; Kajan, L.; Agoston, V. and Pongor, S. **(2005)** *Nucleic Acids Res.* 33(Database issue), D223–225

[213] Maltsev, N.; Marland, E; Yu, G.X.; Bhatnagar, S. and Lusk, R. **(2002)** *Nucleic Acids Res. 30,* 349–350.

[214] Baasiri, R.A.; Glasser, S.R.; Steffen, D.L. and Wheeler, D.A. **(1999)** *Oncogene 18,* 7958–7965.

[215] Levine, A. E. and Steffen, D. L. **(2001)** *Nucleic Acids Res. 29,* 300–302.

[216] Yang, Y. and Fu, L. M. **(2003)** *Bioinformatics 19,* 2311–2312.

[217] Albà, M., Lee, D., Pearl, F.M.G., Shepherd, A.J., Martin, N., Orengo, C.A. and Kellam, P. **(2001)** Nuleic Acids Research 29: 133–136.

[218] Nusse, R. **(2005)** *Cell Res. 15,* 28–32.

[219] Jaroszewski, L.; Rychlewski, L. and Godzik, A. **(2000)** *Protein Science* **9**, 1487–1496

[220] Pietrokovski, S. **(1996)** *Nucleic Acids Res. 24,* 3836–3845.

[221] Kelley LA, MacCallum RM & Sternberg MJE **(2000).** J. Mol. Biol. 299(2), 501-522

[222] Shi, J., Blundell, T. L., and Mizuguchi, K. **(2001)**. J. Mol. Biol., *310,* 243-257.

[223] McGuffin, L. J.; Street, S.; Sorensen, S. A. and Jones, D. T. **(2004)** *Bioinformatics 20,* 131–132.

[224] Ginalski, K.; Pas, J., Wyrwicz, L.S.; von Grotthuss, M.; Bujnicki, J.M.& Rychlewski, L. **(2003)** *Nucleic Acids Res.* **31**, 3804–3807.

[225] Skolnick, J. and Kihara, D. **(2001)** Defrosting the frozen approximation: PROSPECTOR - a new approach to threading. *Proteins* **42**, 319–331.