2010-08-09

Model Detection Based upon Amino Acid Properties

Kit J. Menlove
Brigham Young University - Provo

Homology Searching and Protein Model Detection

Utilizing Amino Acid Properties

Kit J. Menlove

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Keith A. Crandall, Chair
Mark J. Clement
Thomas A. Knotts

Department of Biology

Brigham Young University

December 2010

ABSTRACT

Homology Searching and Protein Model

Detection Utilizing Amino Acid Properties

Kit J. Menlove

Department of Biology

Master of Science

Similarity searches are an essential component to most bioinformatic applications. They form the bases of structural motif identification, gene identification, and insights into functional associations. With the rapid increase in the available genetic data through a wide variety of databases, similarity searches are an essential tool for accessing these data in an informative and productive way. In our chapter, we provide an overview of similarity searching approaches, related databases, and parameter options to achieve the best results for a variety of applications. We then provide a worked example and some notes for consideration.

Homology detection is one of the most basic and fundamental problems at the heart of bioinformatics. It is central to problems currently under intense investigation in protein structure prediction, phylogenetic analyses, and computational drug development. Currently discriminative methods for homology detection, which are not readily interpretable, are substantially more powerful than their more interpretable counterparts, particularly when sequence identity is very low. Here I present a computational graph-based framework for homology inference using physiochemical amino acid properties which aims to both reduce the gap in accuracy between discriminative and generative methods and provide a framework for easily identifying the physiochemical basis for the structural similarity between proteins. The accuracy of my method slightly improves on the accuracy of PSI-BLAST, the most popular generative approach, and underscores the potential of this methodology given a more robust statistical foundation.

Keywords: similarity searching; fold recognition; homology modeling; sequence profiles;

BLAST; sequence alignment; protein evolution; threading

ACKNOWLEDGEMENTS

I would like to thank my committee members and my former advisor, Dr. David McClellan for their remarkable patience, helpful comments and guidance, passion for their research, and stalwart examples of how to conduct meaningful research. I am be eternally grateful to Dr. Crandall, Dr. Clement, and Dr. McClellan for their instrumental role in developing in me an interest in computational biology and bioinformatics, as well as establishing the Bioinformatics program at BYU. I thank the Cancer Research Center at BYU for their support and training opportunities. I thank my mother and grandparents for their encouragement and unwavering support. Most of all I am indebted to my wonderful wife and my Father in Heaven who have faithfully and lovingly stood by me through the seemingly countless long days of thesis work.

Table of Contents

List of Tables

List of Figures

Chapter 1 Similarity searching using BLAST

Kit J. Menlove, Mark Clement, and Keith A. Crandall

Published in *Bioinformatics for DNA Sequence Analysis* (Menlove, Clement et al. 2009)

1. Introduction

1.1. An introduction to nucleotide databases

Perhaps THE central goal of genetics is to articulate the associations of phenotypes of interest with their underlying genetic components and then to understand the relationship between genetic variation and variation in the phenotype. This goal has been buoyed by the tremendous increase in our ability to obtain molecular genetic data, both across populations and species. As methods of gathering information about various aspects of biological macromolecules arose, biological information became abundant and the need to consolidate and make this information accessible became increasingly apparent. In the early 1960's, Margaret Dayhoff and colleagues at the National Biomedical Research Foundation (NBRF) began collecting information on protein sequences and structure into a volume entitled *Atlas of Protein Sequence and Structure (Dayhoff, Eck et al. 1965)*. Since that beginning, databases have been an important and essential part of biological and biochemical research. By 1972, the size of the Atlas was becoming unwieldy, so Dr. Dayhoff, a pioneer of bioinformatics, developed a database infrastructure into which this information could be funneled. Though nucleotide information was included in the Atlas as early as 1966 *(Hersh 1967)*, its bulk was comprised of amino acid sequences with structural annotation.

1.2. International Nucleotide Sequence Database Collaboration: DDBJ, EMBL, and GenBank

It was not until 1982 that databases were developed with the express purpose of storing nucleotide sequences by the European Molecular Biology Laboratory (EMBL: http://www.embl.org/) in Europe and the National Institutes of Health (NIH – NCBI: http://www.ncbi.nlm.nih.gov/) in North America. Japan followed suit with the creation of their DNA Databank (DDBJ: http://www.ddbj.nig.ac.jp/) in 1986. A sizeable amount of sharing naturally occurred between these three databases and the Genome Sequence Database, also in North America, a condition that led to their coalition in 1988 under the title International Nucleotide Sequence Database Collaboration (INSDC). They still remain very distinct entities, but in the 1988 meeting, established policies to govern the formatting of and stewardship over the sequences each receives. Their current policies include unrestricted access and use of all data records, proper citation of data originators, and the responsibilities of submitters to verify the validity of the data and their right to submit it. The INSDC currently contains approximately 80 billion base pairs (not including whole-genome shotgun sequences) and nearly 80 million sequence entries. Including shotgun sequences (HTGS), it passed the 100 gigabase mark on August 22, 2005 and contains approximately 200 billion base pairs as of September 2007. For more than ten years, the amount of data in these databases doubled approximately every 18 months. This expansion has begun to level off as our capacity for high-throughput sequencing is gradually reaching a maximum. The next redoubling of the data is expected to occur in approximately 4 years (**Fig. 1.1**).

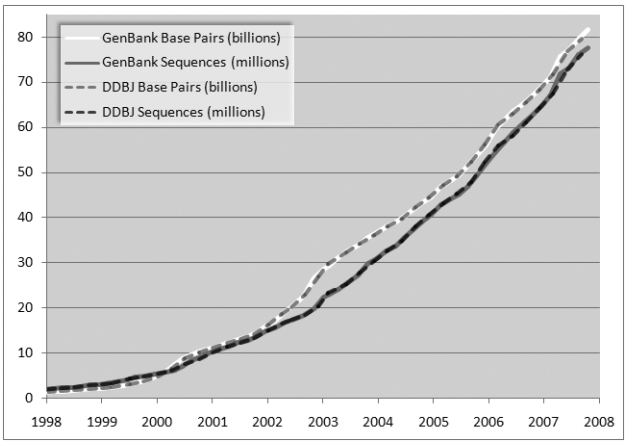


Figure 1.1 Growth of GenBank and DDBJ genetic databases over the past ten years. The INSDC databases have grown, over the past 10 years, approximately 168 fold in total number of base pairs. While in the past the number of entries in INSDC databases doubled approximately every two years, a simple second-order polynomial regression (R^2=0.9995) of the data over the past ten years indicates that the next redoubling will take a little over four years. This graph does not include HTG data.

1.3. Other nucleotide sequence databases

Since the first nucleotide databases were initiated by EMBL and NIH (now held by NCBI), many DNA databases have been formed to cater to the needs of specialized research groups. The 2007 Database issue of Nucleic Acid Research contained 109 nucleotide sequence databases that met the standards to be included in its listing *(Galperin 2007)*. These databases are typically developed to include ancillary data associated with the genetic data, such as patient or specimen information, including clinical information, images, downstream analyses, etc. Many do not meet the standards of "quality, quantity and originality of data as well as the quality of the web interface" that are required to be considered for the issue *(Batemen 2007)*. Even more are privately held to permit access of costly data to a select few. All in all, the number of DNA

databases is astounding and steadily increasing as we find new, powerful ways to gather, store, and utilize the pieces that comprise the puzzle of life.

2. Program Usage

2.1. Database file formats

One of the largest sources of diversity among DNA databases lies in their file formats. While great efforts have been made to standardize file formats, the various types and purposes of sequence information and annotation entreat customized file types.

2.1.1. FASTA format

First used with Pearson and Lipman's FASTA program for sequence comparison *(Pearson and Lipman 1988)*, the FASTA file format is the simplest of the widely-used formats available through the INSDC. It is composed of a definition or description line followed by the sequence. The definition line begins with a greater-than sign (>) and marks the beginning of each new entry. The information following the greater-than symbol varies according to its source. Generally, an identifier follows (**Table 1.1**), after which optional description words may be

Database name	**Identifier syntax**
GenBank	gb|*accession.version*
EMBL	emb|*accession.version*
DDBJ	dbj|*accession.version*
NCBI RefSeq	ref|*accession.version*
PDB	pdb|*entry*|*chain*
Patents	pat|*country*|*number*

NBRF PIR	pir||*entry*
SWISS-PROT	sp|*accession*|*entry*
Protein Research Foundation	prf|*name*
GenInfo Backbone Id	bbs|*number*
General database identifier	gnl|*database*|*identifier*
Local Sequence identifier	lcl|*identifier*

Table 1.1. FASTA File Sequence Identifiers. Information from the NCBI Handbook (Madden 2002).

included. If the sequence is retrieved through NCBI's databases, a GI number precedes the identifier. Though it is recommended that the definition line be no greater than 80 characters, various types and levels of information are often included. The definition line is followed by the DNA sequence itself, in single or multi-line format. Nucleotides are represented by their standard IUB/IUPAC codes, including ambiguity codes (**Table 1.2**).

A	**a**denosine	M	A or C (a**m**ino)	V	A, C or G
C	**c**ytidine	K	G or T (**k**eto)	H	A, C or T
G	**g**uanine	R	A or G (pu**r**ine)	D	A, G or T
T	**t**hymidine	Y	C or T (p**y**rimidine)	B	C, G or T
U	**u**ridine	S	A or T (**s**trong)	–	gap of indeterminate length
		W	C or G (**w**eak)	N	A, C, G or T (a**n**y or unk**n**own)

Table 1.2. IUB/IUPAC nucleotide and ambiguity codes.

2.1.2. Flat file format

GenBank, EMBL, and DDBJ each have their own flat file format, but contain basically the same information. They are all based upon the Feature Table, which can be found at http://www.ncbi.nlm.nih.gov/collab/FT. For references to these file types, see *(León and Markel 2003).*

2.1.3. Accession numbers, version numbers, locus names, database identifiers, etc.

The standard for identifying a nucleotide sequence record is by an *accession*.*version* system where the *accession number* is an identifier of two letters followed by six digits and the *version* is an incremental number indicating the number of changes that have been made to the sequence since it was first submitted. Locus names (*see* **Note 1**) are older, less standardized identifiers whose original purpose was to group entries with similar sequences. The original locus format was intended to hold information about the organism and other common group characteristics (such as gene product). That 10-character format is no longer able to hold such information for the large number and variety of sequences now available, so the locus has become yet another unique identifier often set to be the same value as the accession number. Database identifiers are simply two or three-character strings that serve to indicate which database originally received and stored the information. The database identifier is the first value listed in the FASTA identifier syntax (**Table 1.1**).

When a sequence is first submitted to GenBank, it is submitted with several defined features associated with the sequence. Some include CDS (coding sequence), RBS (ribosome binding site), rep_origin (origin of replication), and tRNA (mature transfer RNA) information. A translation of protein coding nucleotide sequences into amino acids is provided as part of the features section. Likewise, labeling of different open reading frames, introns, etc. are all part of the table of features. A list of features and their descriptions, formats, and conventions that were agreed upon by INSDC can be found in the Feature Table (*see* section 2.1.2).

2.2. Smith-Waterman and Dynamic Programming

In 1970, Needleman and Wunsch adapted the idea of dynamic programming to the difficult problem of global sequence alignment *(Needleman and Wunsch 1970).* In 1981, Smith and Waterman adapted this algorithm to local alignments *(Smith and Waterman 1981).* A global alignment attempts to align two sequences throughout their entire length, whereas a local alignment aligns regions of two sequences where high similarity is observed. Both methods involve initializing, scoring, and tracing a matrix where the rows and columns correspond to the bases or residues of the two sequences being aligned (**Fig. 1.2**). In the local alignment case, the first row and first column are filled with zeroes. The remaining cells are filled with a metric value recursively derived from neighboring values:

$$\max\begin{cases} 0 \\ \text{left neighbor} + \text{gap penalty} \\ \text{top neighbor} + \text{gap penatly} \\ \text{top - left neighbor} + \text{match/mismatch score} \end{cases}$$

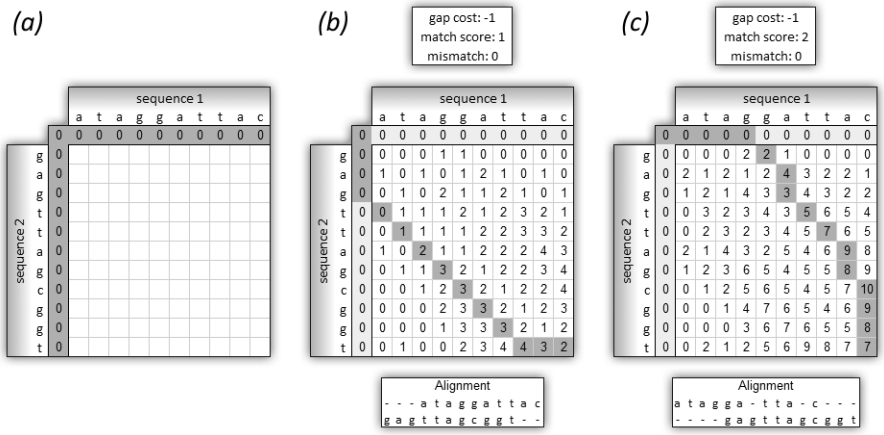


Figure 1.2 Smith-Waterman local alignment example. A shows an empty matrix, initialized for a Smith-Waterman alignment. B and c are alignments calculated using the specified scoring parameters. The

alignment produced in b is drastically different from that in c, though they only differ slightly in their scoring parameters, one using a match score of 1 and the other 2.

If the current cell corresponds to a match (identical bases), the match score is added to the value from the diagonal neighbor, otherwise the mismatch score is used. The gap penalty and mismatch scores are generally zero or a small, negative number while the match score is a positive number larger in magnitude. This method is used recursively, starting from the upper left corner of the matrix and proceeding to the lower right corner. Figs. 1.2b and 1.2c show matrices from two different sets of gap and match scores.

To find a local alignment, one simply finds the largest number in the matrix and traces a path back until a zero is reached, each step moving to a cell that was responsible for the current cell's value. While this method is robust and is guaranteed to give the best alignment(s) for a given set of scores and penalties, it is important to note that often multiple paths and therefore multiple alignments are possible for any given matrix when these parameters are used. As an example, Figs. 2b and 2c only differ slightly in their gap and match scores, but produce very different alignments. In addition, the set of scores and penalties used dramatically affect the alignment and finding the optimal set is neither trivial nor deterministic. Weight matrices for protein-coding sequences were developed in the late 1970s in an attempt to overcome these challenges.

2.3. Weighting/Models

2.3.1. PAM Matrices

In order to increase the specificity of alignment algorithms and provide a means to evaluate their statistical significance, it was necessary to implement a meaningful scoring scheme for nucleotide and amino acid substitutions. This was especially true when dealing with protein (or protein-coding) sequences. In 1978, Dayhoff et al. developed the first scoring or weighting matrices created from substitutions which have been observed during evolutionary history *(Dayhoff, Schwartz et al. 1978).* These substitutions, since they have been allowed or accepted by natural selection, are called accepted point mutations (PAM). For Dayhoff's PAM matrices, groups of proteins with 85% or more sequence similarity were analyzed and their 1571 substitutions were cataloged. Each cell of a PAM matrix corresponds to the frequency in substitutions per 100 residues between two given amino acids. This frequency is referred to as one PAM unit. Back in the 1970's, when they were created, however, there was a limited number and variety of protein sequences available, so they are biased towards small, globular proteins. It is also important to note that each PAM matrix corresponds to a specific evolutionary distance and that each is simply an extrapolation of the original. For example, a PAM250 (**Fig. 1.3**) matrix is constructed by multiplying the PAM1 matrix by itself 250 times and is viewed as a typical scoring matrix for proteins that have been separated by 250 million years of evolution.

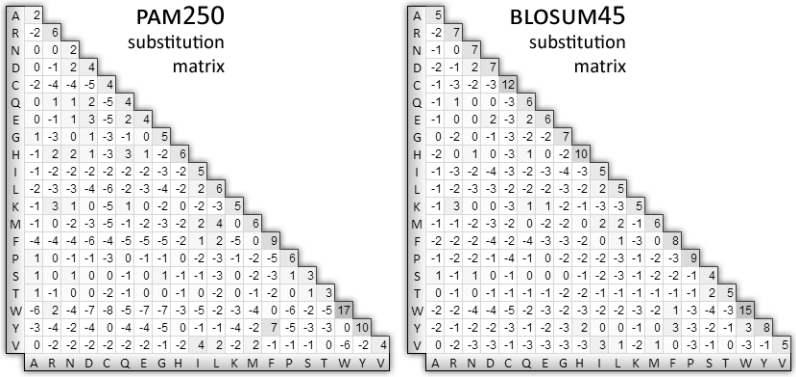


Figure 1.3 PAM250 and BLOSUM45 substitution matrices.

2.3.2. BLOSUM Matrices

To overcome some of the drawbacks of PAM matrices, Henikoff and Henikoff developed the BLOSUM matrices in 1992 *(Henikoff and Henikoff 1992)*. These matrices were based on the BLOCKS database, which organizes proteins into blocks, where each block, defined by an alignment of motifs, corresponds to a family. Whereas the original PAM matrix was calculated with proteins with at least 85% identity, BLOSUM matrices are each calculated separately using conserved motifs at or below a specific evolutionary distance. This diversity of matrices coupled with being based on larger datasets makes the BLOSUM matrices more robust at detecting similarity at greater evolutionary distances and more accurate, in many cases, at performing local similarity searches *(Baxevanis and Ouellette 2005)*.

2.3.3. Choosing a Matrix

When choosing a matrix, it is important to consider the alternatives. Do not simply choose the default setting without some initial consideration. In general, finding similarity at increasing

divergence corresponds to increasing PAM matrices (PAM1, PAM40, PAM120, etc.) and decreasing BLOSUM matrices (BLOSUM90, BLOSUM80, BLOSUM62, etc.) (Wheeler 2003). PAM matrices are strong at detecting high similarity due to their use of evolutionary information. However, as evolutionary distance increases, BLOSUM matrices are more sensitive and accurate than their PAM counterparts. **Table 1.3** includes a list of suggested uses.

Alignment size	Best at detecting:	% Similarity	PAM	BLOSUM
Short	**Similarity within a** species	**75–90**	PAM30	**BLOSUM95**
"	**Similarity within a** genus	**60–75**	PAM70	**BLOSUM85**
Medium	**Similarity within a** family	**50–60**	**PAM120**	BLOSUM80
"	**The** largest range **of similarity**	**40–50**	**PAM160**	BLOSUM62
Long	**Similarity within a** class	**30–40**	**PAM250**	BLOSUM45
"	**Similarity within the** twilight zone	**20–30**		**BLOSUM30**

Table 1.3. Suggested uses for common substitution matrices. The matrices highlighted in bold are available through NCBI's Blast web interface. Blosum62 has been shown to provide the best results in Blast searches overall due to its ability to detect large ranges of similarity. Nevertheless, the other matrices have their strengths. For example if your goal is to only detect sequences of high similarity to infer homology within a species, the pam30, blosum90, and pam70 matrices would provide the best results. This table was adapted from results obtained by David Wheeler (Wheeler 2003).

2.4. Blast Programs

Nucleotide-nucleotide searches are beneficial because no information is lost in the alignment. When a codon is translated from nucleotides to amino acid, approximately sixty-nine percent of the complexity is lost (64 possible nucleotide combinations mapped to 20 amino acids). In contrast, however, the true physical relationship between two coding sequences is best captured in the translated view. Matrices that take into account physical properties, such as PAM and BLOSUM, can be used to add power to the search. Additionally, in a nucleotide search, there

are only four possible character states compared to 20 in an amino acid search. Thus the probability of a match due to chance versus a match due to common ancestry (identify in state versus identical by descent) is high.

The Basic Local Alignment and Search Tools (BLAST) are the most widely used and among the most accurate in detecting sequence similarity *(Altschul, Gish et al. 1990)* (*see* **Note 2**). The standard BLAST programs are Nucleotide BLAST (blastn), Protein BLAST (blastp), blastx, tblastn, and tblastx. Others have also been developed to meet specific needs. When choosing a BLAST program, it is important to choose the correct one for your question of interest. Some of the most common mistakes in similarity searching come from misunderstandings of these different applications.

nucleotide blast: Compares a nucleotide query against a nucleotide sequence database

protein blast: Compares an protein query against a protein sequence database

blastx: Compares a nucleotide query translated in all 6 reading frames against a protein database

tblastn: Compares a protein query against a nucleotide sequence database dynamically translated in all 6 reading frames

tblastx: Compares a nucleotide query in all 6 reading frames against a nucleotide sequence database in all six reading frames

The BLAST algorithm is an heuristic program, one that is not guaranteed to return the best result. It is, however, quite accurate. BLAST works by first making a look-up table of all the "words" and "neighboring words" of the query sequence. Words are short subsequences of length *W* and

neighboring words are words that are highly accepted in the scoring matrix sense, determined by a threshold *T*. The database is then scanned for the words and neighboring words. Once a match is found, extensions with and without gaps are initiated there both upstream and downstream. The extension continues, adding gap existence (initiation) and extension penalties, and match and mismatch scores as appropriate as in the Smith-Waterman algorithm until a score threshold *S* is reached. Reaching this mark flags the sequence for output. The extension then continues until the score drops by a value *X* from the maximum, at which point extension stops and the alignment is trimmed back to the point where the maximum score was hit. Understanding this algorithm is important for users if they are to select optimal parameters for BLAST. The interaction between the parameters *T, W, S, X,* and the scoring matrix allows the user to find a balance between sensitivity and specificity, alter the running time, and tweak the accuracy of the algorithm. The interactions among these variables will be discussed in section 2.8.

2.5. Query Sequence

Query sequences may be entered by uploading a file or entering one manually in the text box provided (**Fig. 1.4**). The upload option accepts files containing a single sequence, multiple sequences in FASTA format, or a list of valid sequence identifiers (accession numbers, GI numbers, etc.). In contrast to previous versions of BLAST on the NCBI website, the current version allows the user to specify a descriptive job title. This allows the user to track any adjustments or versions of a search as well as its purpose and query information. This is especially important when sequence identifiers are not included in the uploaded file.

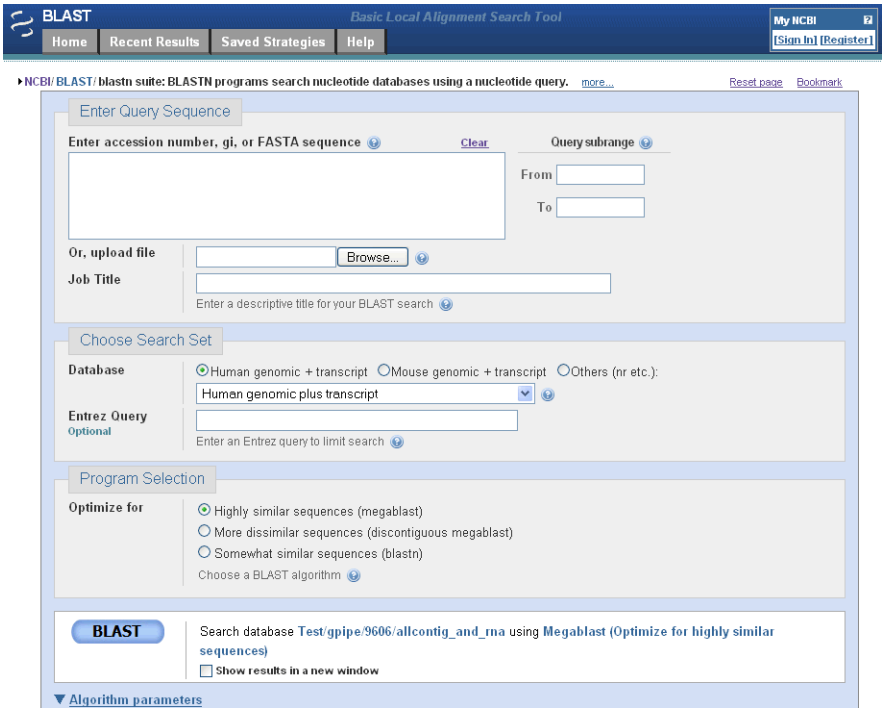


Figure 1.4. NCBI Nucleotide BLAST Interface.

2.6. Search Set

2.6.1. Databases

When choosing a database, it is important to understand their purpose, content, and limitations. The list of nucleotide databases is divided into *Genomic plus Transcript* and *Other Databases* sections. Some of the databases, composed of reference sequences, come from the RefSeq database, a highly-curated, all-inclusive, non-redundant set of INSDC (EMBL + GenBank + DDBJ) DNA, mRNA, and protein entries. RefSeq sequences have accession numbers of the form AA_###### where AA is one of the following combination of letters (**Table 1.4**) and ###### is a unique number representing the sequence.

Experimentally Determined and Curated		Genome annotation (computational predictions from DNA)	
NC	**C**omplete genomic molecules		
NG	Incomplete **g**enomic region		
NM	**m**RNA	XM	Model **m**RNA
NR	**R**NA (non-coding)		
NP	**p**rotein	XP	Model **p**rotein

Table 1.4. RefSeq Categories

A description of the nucleotide databases is included below. A list of protein databases accessible through BLAST's web interface can be found at http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml.

Human genomic plus transcript: Contains all human genomic and RNA sequences.

Mouse genomic plus transcript: Contains all mouse genomic and RNA sequences.

Nucleotide collection (nr/nt): Contains INSDC + RefSeq nucleotides + PDB sequences, not including EST, STS, GSS, or unfinished HGT sequences. The Nucleotide collection is the most comprehensive set of nucleotide sequences available through BLAST.

Reference mRNA sequences (refseq_rna): Contains the non-redundant RefSeq mRNA sequences.

Reference genomic sequences (refseq_genomic): Contains the non-redundant RefSeq genomic sequences.

Expressed sequence tags (est): Contains short, single reads from mRNA sequencing (via cDNA). These cDNA sequences represent the mRNA in a cell at a particular moment in a particular tissue.

Non-human, non-mouse ESTs (est_others): The previous database with human and mouse sequences removed.

Genomic survey sequences (gss): Contains random genomic sequences obtained from single-pass genome surveys, cosmids, BACs, YACs, and other survey methods. Their quality varies.

High-throughput genomic sequences (HTGS): Contains sequences obtained from high-throughput genome centers. Sequences in this database contain a phase number, 0 being the initial phase and 3 being the finished phase. Once finished, the sequences move to the appropriate division in their respective database.

Patent sequences (pat): Contains sequences from the patent offices at each of the INSDC organizations.

Protein data bank (pdb): The nucleotide sequences from the Brookhaven Protein Data Bank managed by the Research Collaboratory for Structural Bioinformatics (http://www.rcsb.org/pdb).

Human ALU repeat elements (alu_repeats): Contains a set of ALU repeat elements that can be used to mask repeat elements from query sequences. ALU sequences are regions subject to cleavage by Alu restriction endonucleases, around 300 base pairs long, and estimated to constitute about 10% of the human genome (Roy-Engel, Carroll et al. 2001).

Sequence tagged sites (dbsts): A collection of unique sequences used in PCR and genome mapping that identify a particular region of a genome.

Whole-genome shotgun reads (wgs): Contain large-scale shotgun sequences, mostly unassembled and non-annotated.

Environmental samples (env_nt): Contains sets of whole-genome shotgun reads from many sampled organisms, each set from a particular location of interest. These sets allow researchers to look into the genetic diversity existing at a particular location and environment.

2.6.2. Organism

The organism box allows the user to specify a particular organism to search. It automatically suggests organisms when you begin typing. This option is not available when Genomic plus Transcript databases are selected (**Fig. 1.5**).

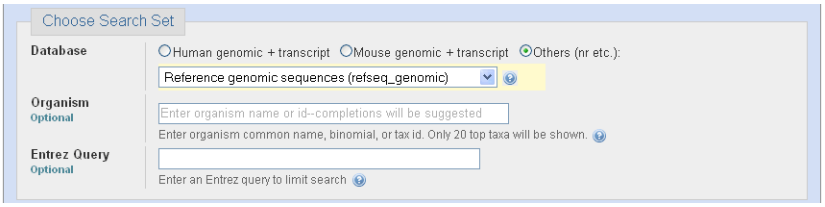


Figure 1.5. Organism Selection when Searching a Multi-organism Database.

2.6.3. Entrez Queries

Entrez queries provide a way to limit your search to a specific type of organism or molecule. It is an efficient way to filter unwanted results by excluding organisms or defining sequence length criteria. In addition, Entrez queries allow the user to find sequences submitted by a particular author, from a particular journal, with a particular property or feature key, or submitted or modified within a specific date range. For help with Entrez queries, see the Entrez Help document at http://www.ncbi.nlm.nih.gov/entrez/query/static/help/helpdoc.html.

2.7. Blast Search Parameters

In addition to entering a query sequence, choosing a search set, and selecting a program, several additional parameters are available which allow you to fine-tune your search to your needs. These parameters are available by clicking the "Algorithm parameters" link at the bottom of the BLAST page (**Fig. 1.6**) (*see* **Notes 3 and 4**).

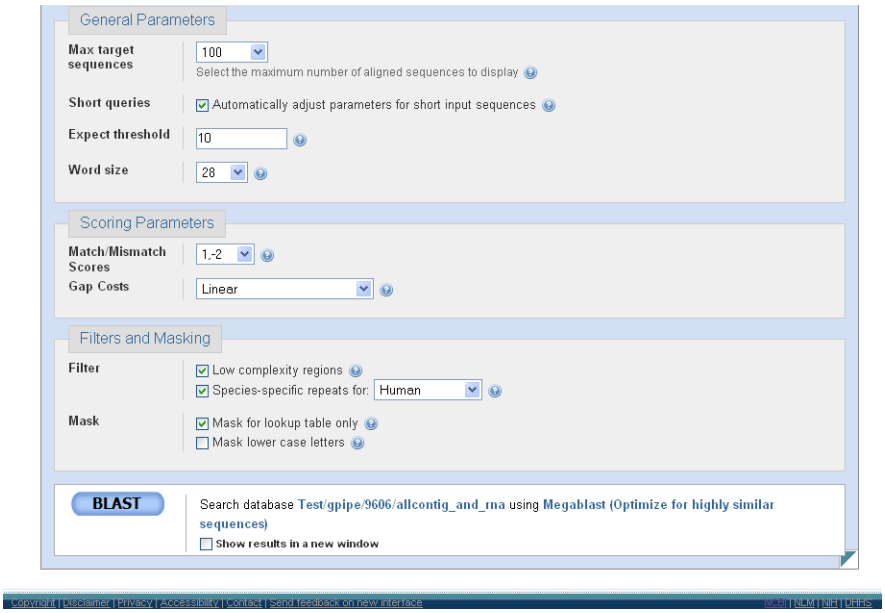


Figure 1.6. NCBI Nucleotide BLAST Algorithm parameters.

2.7.1. Max Target Sequences

The maximum target sequences parameter allows you to select the number of sequences you would like displayed in your results. Lower numbers do not reduce the search time, but do reduce the time to send the results back. This is generally only an issue over a slow connection.

2.7.2. Short Queries

When using short queries (of length 30 or less), the parameters must be adjusted or you will not receive statistically significant results. Checking the "short queries" box automatically adjusts the parameters to return valid responses for a short query sequence.

2.7.3. Expect Threshold

The expect threshold limits the results displayed to those with an *E*-value lower than it. This value corresponds to the number of sequence matches that are expected to be found merely by chance.

2.7.4. Word Size

The word size, *W*, as discussed earlier determines the length of the words and neighboring words used as initial search queries. Increasing the word size generally results in fewer extension initializations, increasing the speed of the BLAST search but decreasing its sensitivity.

2.7.5. Scoring Parameters

The scoring parameters of a nucleotide search are the match and mismatch scores and gap costs. In protein searches, the match and mismatch scores are indicated by a scoring matrix (*see* section 2.3). A limited set of suggested match and mismatch scores are available from the dropdown

menu on NCBI's BLAST search form. Increasing the ratio in the following fashion (match, mismatch): (1,-1) → (4,-5) → (2,-3) → (1,-2) → (1,-3) → (1,-4) prevents mismatched nucleotides from aligning, increasing the number of gaps, but decreasing mismatches. The greater divergence you expect in sequences you are looking for, the larger the ratio you should choose. NCBI has provided the guidelines found in **Table 1.5**. Additionally, decreasing the gap existence and extension penalties will increase gap incidence.

Match/Mismatch Ratio	% Similarity
-0.33 (1/-3)	**99%**
-0.5 (1/-2)	**95%**
-1 (1/-1)	**75%**

Table 1.5. Suggested scoring parameters for nucleotide-nucleotide Blast searches. When performing a nucleotide-nucleotide Blast search, these general guidelines may be used to choose a match/mismatch score, based upon the degree of conservation you expect to see in your results. If you are searching for sequences with a high degree of similarity (i.e. within a species), the default parameters of (match +1, mismatch -2) would be appropriate. If, however, you are searching for sequences between very distant organisms (a worm and a mouse, for example), a smaller ratio would be more appropriate (for example, -1). Information provided by NCBI .

2.7.6. Filters

The low complexity regions filter removes regions of the sequence with low complexity, preventing those segments from producing statistically significant but uninformative results. The DUST program by Tatusov and Lipman (unpublished) is used for nucleotide BLAST searches. Often, when a search takes much longer than expected, the query contains a low-complexity region that is being matched with many similar but unrelated sequences. It is important to note, however, that turning this filter on may remove some interesting and informative matches from

the results. In nucleotide searches, it is also possible to remove species-specific repeats by checking the "Species-specific repeats for:" box and selecting the appropriate species. This prevents repeats that are common in a particular species from producing false-positives with other parts of its own or closely related genomes.

2.7.7. Masks

The "Mask for lookup table only" option allows the user to mask the low-complexity regions (regions of biased composition including homopolymeric runs, short-period repeats, etc.) during the seeding stage, where words and neighboring words are scanned, but unmask them during the extension phases. This prevents the *E*-values from being affected in biologically interesting results while preventing regions of low-complexity from slowing the search down and introducing uninteresting results.

The "Mask lower case letters" option gives the user the option to annotate his or her sequence by using lower case letters where masking is desired.

2.8. Interpreting the Results

By default, BLAST results contain five basic sections: a summary of your input (query and parameters), a graphical overview of the top results, a table of sequences producing significant alignments, the best 100 alignments, and result statistics. The number of hits shown in the graphical overview as well as the number of alignments, among other options, may be changed

by clicking “Reformat these results” at the top of the results page or by clicking “Formatting options” on the Formatting Results page (the page that appears after you click BLAST and before the results appear).

In the third section, the results table contains eight columns: accession, description, max score, total score, query coverage, E value, max ident, and links. The **Accession** number provides a link to detailed information about the sequence. The **description** provides information about the species and the kind of sample the hit was generated from. The **max score** provides a metric for how good the best local alignment is. The **total score** indicates how similar the sequence is to the query, accounting for all local alignments between the two sequences. If the max score is greater than the total score, then more than one local alignment was found between the two sequences. Higher scores are correlated with more similar sequences. Both of these scores, reported in bits, are calculated from a formula that takes into account matches (or similar residues, if doing a protein search) and mismatch penalties along with gap insertion penalties. Bit scores are normalized so that they can be directly compared even though the alignments between different sequences may be of different lengths. The expectation value or **E-value** provides an estimate of how likely it is that this alignment occurred by random chance. An E-value of 2e-02 indicates that similarity found in the alignment has a 2 in 100 chance of occurring by chance. The lower the E-value, the more significant the score. An appropriate cutoff E-value depends on the users goals. The **max identity** field shows the percentage of the query sequence that was identical to the database hit. The **links** field provides links to UniGene, the Gene Expression Omnibus, Entrez Gene, Entrez’s Related Structures (for protein sequences), and the Map Viewer (for genomic sequences).

2.9. Future of Similarity Searching

Since both PAM and BLOSUM matrices are experimentally derived from a limited set of sequences in a database that was available at the time they were created, they will almost certainly not provide optimal values for searches with new sequence families. Current research is being performed to determine which chemical properties are changing in a sequence in order to provide a magnitude of change that is independent of scoring matrices.

Current techniques to find promoter regions are severely lacking in accuracy (Tompa, Li et al. 2005). Techniques will arise in the future that may improve current methods by using BLAST-like algorithms to assess the similarity of a sequence to known promoter elements, thus helping to identify it as a promoter.

3. Examples

This section will provide three examples of common BLAST uses: a nucleotide-nucleotide BLAST, a position specific iterated BLAST, and a BLASTX.

3.1. Nucleotide-Nucleotide Blast for allele finding

Here we present an example of using BLAST to search for the known alleles of a given nucleotide sequence. This approach can be used to answer the question: What are the known variants of my gene of interest (within its species)? Our example will be to find all known variants of a Tp53

nucleotide sequence (accession number AF151353) from a mouse. While this sequence does code for a protein, non-coding sequences would work just as well using this approach.

We will start by going to the BLAST homepage at http://www.ncbi.nlm.nih.gov/BLAST/ and selecting **nucleotide blast**. In the "Enter Query Sequence" box, we type the accession number: AF151353. You will notice that the "Job Title" box automatically fills in a title for you "AF151353:*Mus musculus* tumor suppressor p53...". If we were to paste a sequence instead of an accession number or gi, we would want to enter a job title to help us keep track of our results. Under "Choose Search Set", we select the "Nucleotide collection (nr/nt)" database since it is the most comprehensive database (remember that nr is no longer non-redundant). For a complete search, we should also perform a search on the "Expressed sequence tags (est)" database. In the Organism box, we chose type "mouse" and select "mouse (taxid:10090)", which corresponds to *Mus musculus*, the house mouse. Since we are searching for alleles, we select "Highly similar sequences (megablast)" in the "Program Selection" box.

Next, let's change the algorithm parameters. Click "Algorithm parameters" to display them. Since the sequence is 1409 base pairs in length, we deselect the "Automatically adjust parameters for short input sequences" box. Since we expect that the p53 protein is a well conserved protein (due to its critical function), we set the expect threshold to a low value. Let's choose 1e-8. For a word size, we are not concerned about speed in this case, so the number of extensions performed is not a concern. Let's select a word size of 20 to make sure we don't miss any matches (although in this case a larger word size shouldn't make much difference). As for

the scoring parameters, we choose the largest ratio, corresponding to the greatest identity: “1,-4”. Since this is a protein-coding sequence, we don’t expect repeats to be a factor, so we leave the Filters and Masking section at the default settings.

The results indicate that 108 hits were found on the query sequence. Looking at the graphical alignment (**Fig. 1.7**), we notice that only about 2/3 of them span a good portion of the query. When we scroll down to the gene descriptions, most of the last fourth are pseudogenes (partial sequence) (**Fig. 1.8**), which may offer insight into different alleles and their corresponding phenotypes, but which were not sequenced experimentally. Performing a search on the EST database with the same parameters results in 101 additional hits.

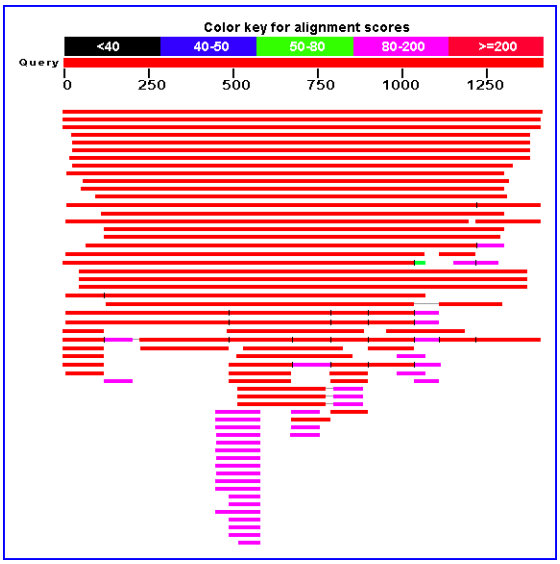
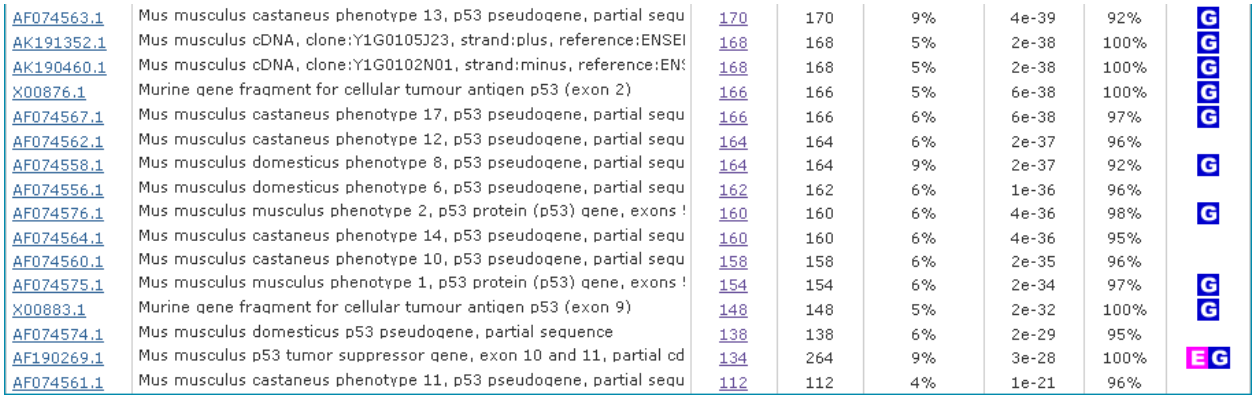


Figure 1.7. Graphical Distribution of top 100 BLAST hits.



AF074563.1	Mus musculus castaneus phenotype 13, p53 pseudogene, partial sequ	170	170	9%	4e-39	92%	G
AK191352.1	Mus musculus cDNA, clone:Y1G0105J23, strand:plus, reference:ENSEI	168	168	5%	2e-38	100%	G
AK190460.1	Mus musculus cDNA, clone:Y1G0102N01, strand:minus, reference:EN	168	168	5%	2e-38	100%	G
X00876.1	Murine gene fragment for cellular tumour antigen p53 (exon 2)	166	166	5%	6e-38	100%	G
AF074567.1	Mus musculus castaneus phenotype 17, p53 pseudogene, partial sequ	166	166	6%	6e-38	97%	G
AF074562.1	Mus musculus castaneus phenotype 12, p53 pseudogene, partial sequ	164	164	6%	2e-37	96%	
AF074558.1	Mus musculus domesticus phenotype 8, p53 pseudogene, partial sequ	164	164	9%	2e-37	92%	G
AF074556.1	Mus musculus domesticus phenotype 6, p53 pseudogene, partial sequ	162	162	6%	1e-36	96%	
AF074576.1	Mus musculus musculus phenotype 2, p53 protein (p53) gene, exons !	160	160	6%	4e-36	98%	G
AF074564.1	Mus musculus castaneus phenotype 14, p53 pseudogene, partial sequ	160	160	6%	4e-36	95%	
AF074560.1	Mus musculus castaneus phenotype 10, p53 pseudogene, partial sequ	158	158	6%	2e-35	96%	
AF074575.1	Mus musculus musculus phenotype 1, p53 protein (p53) gene, exons !	154	154	6%	2e-34	97%	G
X00883.1	Murine gene fragment for cellular tumour antigen p53 (exon 9)	148	148	5%	2e-32	100%	G
AF074574.1	Mus musculus domesticus p53 pseudogene, partial sequence	138	138	6%	2e-29	95%	
AF190269.1	Mus musculus p53 tumor suppressor gene, exon 10 and 11, partial cd	134	264	9%	3e-28	100%	E G
AF074561.1	Mus musculus castaneus phenotype 11, p53 pseudogene, partial sequ	112	112	4%	1e-21	96%	

Figure 1.8. Last 16 sequences producing significant alignments from a mouse p53 gene Nucleotide BLAST search. Nineteen of the last twenty-six reported sequences are pseudogenes.

3.2. PSI-Blast for distant homology searching

When searching for distantly related sequences, two BLAST options are available. One is the standard nucleotide-nucleotide BLAST with discontiguous BLAST, a method very similar to Ma et al's work (Ma, Tromp et al. 2002), selected as the program. The other is to use a more sensitive approach, PSI-BLAST, which performs an iterative search on a protein sequence query. Though the second approach will only work if you are dealing with protein-coding sequences, it is more sensitive and accurate than the first.

In this example, we will search for relatives of the cytocrome b gene of the Durango night lizard (*Xantusia extorris*). We start by selecting **protein blast** from the BLAST home page and entering the accession number, ABY48155, into the query box. If your sequence is not available as a protein sequence, you will need to translate it. This can easily be done using a program such as MEGA (Tamura, Dudley et al. 2007), available at http://www.megasoftware.net, or an online

tool such as the JustBio Translator (http://www.justbio.com/translator/) or the ExPASy Translate Tool (http://www.expasy.org/tools/dna.html).

Once again, the "Job Title" box is filled with "ABY48155:cytochrome b [Xantusia extorris]". We will choose the "Reference proteins (refseq_protein)" database, which is more highly curated and non-redundant than (per gene) than the default nr database. We do not specify an organism because we want results from any and all related organisms. For the algorithm, we select PSI-BLAST due to its ability to detect more distantly related sequences. We hope to include as many sequences as possible in our iterations, so we choose 1000 as the max target sequences. We can, once again, remove the "Automatically adjust parameters for short input sequences" check since our sequence is sufficiently long (380 amino acids). Since we wish all related sequences, we keep the expect threshold at its default of 10. While decreasing it may remove false positives, it may also prevent some significant results from being returned. Since we do not have a particular scope in mind (within the genus or family, for example), we will use the BLOSUM62 matrix due to its ability to detect homology over large ranges of similarity.

The first iteration results in 1000 hits on the query sequence, all of which cover at least 93% of the query sequence and have an E-value of 10^{-126} or less. We leave all of the sequences selected and press the "Run PSI-Blast iteration 2" button. The second iteration likewise returns 1000 hits, but this time they have E-values less than 10^{-99} and cover at least 65% of the query sequence (all but 6 cover 90% or more). We uncheck the last hit, Bi4p [*Saccharomyces cerevisiae*], since we are unsure of its homology, and iterate one last time.

At this point, it would be helpful to view the taxonomy report of the results. You can do so by clicking "Taxonomy Reports" near the bottom of the first section of the BLAST report. You will notice that we have a good selection of organisms, ranging from bony fishes to proteobacteria. While this list would need to be narrowed to produce a good taxonomy, it would be a good starting point if you wished to perform a broad phylogenetic reconstruction. To perform a search of more closely related sequences, you would likely perform a standard blastp (protein-protein BLAST) instead of a PSI-BLAST and use the PAM 70 or PAM 30 matrix.

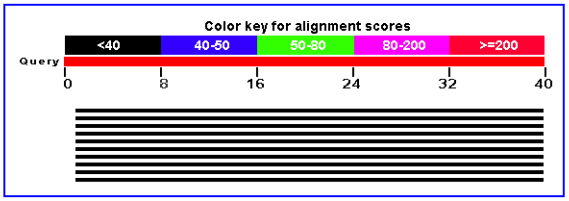
3.3. BlastX for EST identification

What if you have a nucleotide sequence such as an expressed sequence tag and wish to know if it codes for a known protein? You can search the nucleotide database or take the more direct approach of BLASTX. BLASTX allows you to search the protein database using a nucleotide query which it first translates into all six reading frames. In this example, we will perform a blastx on the following sequence: TCTCTATAGTTATGGTGTTCTGAATCAGCCTTCCCTCATA

Since the sequence is only 40 base pairs long, we need to be careful with our parameters. We start by selecting blastx from the BLAST homepage. We then enter the sequence into the query box and enter a relevant job title, such as "EST BlastX Search 1". We will search the "Non-redundant protein sequences (nr)" database since it has the largest number of annotated nucleotide sequences. Under "Algorithm parameters", we need to choose an appropriate expect threshold and matrix. If we choose too low of an expect threshold, we might not find anything.

Likewise, if we choose the wrong matrix we may not obtain significant results due to the short length of our sequence. We will choose 10 (the default) as our expect threshold and PAM70 as our matrix, since corresponds to finding similarity at or below the family/genus level. Since we do not know what our sequence is, we want to filter regions of low complexity to ensure that if our sequence contains such regions, they will not return deceptively significant results.

Our search produces a large number (more than 1000) results with an E-value of 0.079 (**Fig. 1.9**). If we were to use the PAM70 matrix, essentially the same results would be obtained, but each with an E-value of 3.0. Since all of the 2,117 results are different entries of the nuclecapsid protein of the Influenza A virus, we can be somewhat confident that our protein is related, especially if we had any prior knowledge that would support our findings.



Legend for links to other resources: U UniGene E GEO G Gene S Structure M Map Viewer

Sequences producing significant alignments:
(Click headers to sort columns)

Accession	Description	Max score	Total score	Query coverage	E value	Max ident	Links
ABY81430.1	nucleocapsid protein [Influenza A virus (A/swine/Iowa/1/1986(H1N1))	38.3	38.3	97%	0.079	92%	
ABQ43788.1	nucleocapsid protein [Influenza A virus (A/mallard/ON/499/2005(H5N:	38.3	38.3	97%	0.079	92%	
CAN89845.1	nucleoprotein [Influenza A virus (A/wild boar/Germany/R169/2006(H:	38.3	38.3	97%	0.079	92%	
ABQ12377.1	nucleocapsid protein [Influenza A virus (A/mallard/Manitoba/458/200	38.3	38.3	97%	0.079	92%	
ABV53582.1	nucleoprotein [Influenza A virus (A/chicken/Nigeria/1071-30/2007(H5	38.3	38.3	97%	0.079	92%	
ABV53572.1	nucleoprotein [Influenza A virus (A/chicken/Nigeria/1071-29/2007(H5	38.3	38.3	97%	0.079	92%	
ABV53532.1	nucleoprotein [Influenza A virus (A/chicken/Nigeria/1071-10/2007(H5	38.3	38.3	97%	0.079	92%	
ABV53512.1	nucleoprotein [Influenza A virus (A/chicken/Nigeria/1071-7/2007(H5N	38.3	38.3	97%	0.079	92%	
ABV53492.1	nucleoprotein [Influenza A virus (A/chicken/Nigeria/1071-4/2007(H5N	38.3	38.3	97%	0.079	92%	
ABV53482.1	nucleoprotein [Influenza A virus (A/chicken/Nigeria/1071-3/2007(H5N	38.3	38.3	97%	0.079	92%	

Figure 1.9. BlastX Results showing E-values of 0.079 for the top ten hits, all of which are nucleocapsid proteins or nucleoproteins.

4. Notes

One of the options NCBI provides from their homepage is to search across their databases using an identifier (accession number, sequence identification number, Locus ID, etc…). This option can be rather straightforward if you are using an identifier unique to a particular sequence; however, if you are searching for a locus across organisms or individuals, you may need to pay close attention to the search terms you are using. For example, since the Cytochrome b/b6 subunit is known by the terms "Cytochrome b", "Cytochrome b6", "cyt-b", "cytb", "cyb" "COB", "COB1", "cyb6", "petB", "mtcyb", and "mt-cyb", in a search for all possible homologs of this subunit it is necessary to search for all of its names and abbreviations used in the organisms of interest. Since research groups studying different organisms create their own unique locus names for the same gene, it is important to use all of them in your search. IHOP (www.ihop-net.org) is an excellent resource for protein names (Hoffmann and Valencia 2004). In addition, you will want to perform a BLAST search to make sure you have everything!

In addition to the BLAST program provided by NCBI, other BLAST programs exist which have improved the BLAST algorithm in various ways. Dr. Warren Gish at Washington University in St. Louis has developed WU-BLAST, the first BLAST algorithm that allowed gaped alignments with statistics (Gish 1996-2004). It boasts speed, accuracy, and flexibility, taking on even the largest jobs. Another program, FSA-BLAST (Faster Search Algorithm), was developed to implement recently published improvements to the original BLAST algorithm (Cameron, Williams et al. 2004-2006). It promises to be twice as fast as NCBI's and just as accurate. WU-BLAST is free for academic and non-profit use and FSA-BLAST is open source under the BSD license agreement.

My NCBI is a tool that allows you to customize your preferences, save searches, and set up automatic searches that send results via e-mail. If you find yourself performing the same searches (or even similar searches) repeatedly, you may want to take advantage of this option! To register, go to the NCBI home page and click the "My NCBI" link under "Hot Spots". Once you have registered and signed in, a new option will be available to you on all BLAST and Entrez searches (**Fig. 1.10**).

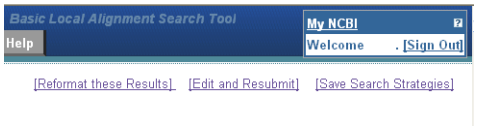


Figure 1.10. Save Search Strategies. The new *My NCBI* interface allows users to save search strategies to assist with repetitive search tasks.

To save a BLAST search strategy, simply click the "Save Search Strategies" link on the results page. This will add the search to your "Saved Strategies" page, which is available through a tab on the top of each page in the BLAST website when you are logged in to My NCBI. Doing so will not save your results, but it will save your query and all parameters you specified for your search so you can run it later to retrieve updated results.

Chapter 2 Model Detection based upon Amino Acid Properties

1. Introduction

Protein structure prediction, over the past two decades, has become the *holy grail* of computational biology. The ability to predict the structure of a protein often precedes our ability to determine its functions and the sites at which it performs each function. Knowing the structure of a protein whose sequence has been mutated is essential to understanding its effects. Since 1973, when Anfinsen showed that a protein's native structure was determined, with few exceptions, from its amino acid sequence alone (Anfinsen 1973), many algorithms have been created in the attempt to predict the final protein structure from its amino acid sequence. To date, the best methods are based upon homology modeling, also known as threading (Kryshtafovych, Fidelis et al. 2007); however some *ab initio* methods, while extremely expensive computationally, have shown encouraging success with shorter proteins (Jauch, Yeo et al. 2007).

Despite the many methods that have been applied, it has proven difficult to predict the structure from a protein given only its amino acid sequence due to immense number (approx. 10^N, where N is the number of amino acids) of possible conformations (Zwanzig, Szabo et al. 1992), particularly when the protein is large and homologous proteins are not available or difficult to detect. This is especially true within the "twilight zone", the region surrounding 25% amino acid similarity where structural homology is still quite elusive. For example, the protein adenylate kinase has essentially the same structure and function in all species, but has low sequence identity (around 20%) in some sections of the protein (Onuchic, Luthey-Schulten et al. 1997). Additionally, while it is estimated that there are less than 4000 distinct protein folds in nature,

many of these folds are yet to be identified and characterized, and methods of recognizing them solely from a sequence of amino acids are encouraging at best.

An increasingly popular method, sometimes referred to as partial-threading, for structure prediction involves a combination of low-resolution prediction and high-resolution refinement (made popular by (Das, Bin et al. 2007)). First, a large number of low-resolution models, typically accurate to 3.5 or 4 Å, are generated. The first criterion for an *optimal* low-resolution model is that it falls within the radius of convergence of the high resolution maximum. The radius of convergence defines the area of the potential energy surface which, upon energy minimization refinement, converges to the global minimum. Each of these models is then refined to a high-resolution state, potentially accurate to 1.5 Å, a process which requires substantial computational power. Therefore, increasing the accuracy of the low-resolution model(s), thereby reducing the number that need to be refined in order to find an optimal structure, is basal to both better and faster predictions.

Recognizing this weakness, Chivian & Baker (Chivian and Baker 2006) developed a systematic way called K*Sync to incorporate a few protein features, such as how obligate a sequence region is to the protein fold, into the dynamic programming alignments used previously (Bellman 1952). While this method outperforms previous ones in most cases, there is nevertheless substantial room for improvement. Other methods have used frequency profiles to search for distant homologs (Jaroszewski, Rychlewski et al. 2000; Yona and Levitt 2002; Edgar and Sjolander 2004), fold recognition methods (Jaroszewski, Rychlewski et al. 1998; Jones 1999;

Panchenko, Marchler-Bauer et al. 2000), and ensemble generation methods (Jaroszewski, Li et al. 2002; Contreras-Moreira, Fitzjohn et al. 2003; John and Sali 2003) to find structurally related areas of proteins where sequence similarity is low.

Amino acid properties have been around for decades, but as of 2008 have not been utilized in the detection of remote homologues. In the 1990s, a list of 31 amino acid properties was compiled with their empirical values for use with TreeSAAP (Woolley, Johnson et al. 2003). In 2000, Kawashima and colleagues created a similar, but more comprehensive, list entitled AAindex (Kawashima, Ogata et al. 1999; Kawashima and Kanehisa 2000). TreeSAAP's creators then used this list to generate an alternate TreeSAAP-formatted list of 515 properties. The AAindex database has now been expanded to include 544 properties in version 9.1 (Kawashima, Pokarowski et al. 2008). Additionally, an alternate dataset of 243 properties is available, but not as comprehensive as that offered in AAindex (Mathura and Kolippakkam 2005).

Here we present an alternative method for model detection based upon the signatures of amino acid properties found in particular domains. The advantages of this method include relatively straightforward interpretation, rapid searching, and accuracy comparable to today's most commonly used methods. This new framework for structural homology determination and functional classification will assist in one of the greatest challenges facing prediction algorithms: "The difficulty in extracting the meaning from protein sequences is in discerning what features are common to all sequences, what features are specific to protein-like sequences, and what features are specific to a given structure." (Onuchic, Luthey-Schulten et al. 1997)

To detect distantly related proteins who share similar structure (but where the structure of at least one of them is not known), we will rely upon highly conserved "property regions." By singling out specific conserved property regions, we seek to capture the important information from a scoring matrix thereby reducing the amount of noise seen by the search algorithm. The method creates a network of property regions representing the query sequence, which will facilitate further investigation on the effects of amino acid properties on functional domains. In contrast to discriminative methods such as support vector machines, graph-based approaches allow for relatively straightforward interpretation, particularly when based upon well understood physiochemical properties. Here we show that such a network-based approach based upon physically meaningful amino acid properties provides an effective alternative to current generative approaches.

2. Methods

2.1. Scoring physiochemical properties according to their biological relevance

Many of the 544 properties found in the AAindex are highly correlated with one another or unimportant in sequence conservation. To reduce the number of properties used in our study, we begin by making use of protein sequence alignment benchmarking datasets created from a combination of methods. Current versions of publicly available datasets include BAliBASE 3 (Thompson, Plewniak et al. 1999; Bahr, Thompson et al. 2001; Thompson, Koehl et al. 2005), OXBench (Raghava, Searle et al. 2003), PREFAB v4 (Edgar 2004), HOMSTRAD (Mizuguchi, Deane et al. 1998; de Bakker, Bateman et al. 2001; Stebbings and Mizuguchi 2004), SABmark 1.65 (Van Walle, Lasters et al. 2005), and SMART 4.0 (Letunic, Copley et al. 2004). Each of

these databases is based on a different combination of manual curation, automation, structural alignment methods (see Table 1.1), sequence alignments, and hidden Markov models. For example, while OXBench is not manually curated and based on automatically created structure and sequence alignments, HOMSTRAD uses a consensus method solely based upon structural alignment programs and is slightly curated. BAliBASE, on the other hand, is highly curated by new experts. Each of these three databases will be used in our study due to their varying levels of automation and curation and excellent sampling across known protein families. By using these datasets, we are able to get a feel for the properties that are most conserved in structural alignments and therefore are likely to display the most signal in protein sequence alignments. In addition, we look at the influence of gaps on conserved amino acid properties.

We began by parsing through the 515 properties compiled in 2006 for TreeSAAP to remove errors and duplicates. There were three errors in property name and approximately 12 duplicates where the name was similar and the empirical, numerical, values were exactly the same. After removal of these duplicates, 503 properties remained, including six pairs where the values were very similar, but not equivalent. Most of the six were simply measurements taken by different groups or the same group at different times. These properties were noted, but preserved in the list for this analysis.

The second task was to shift the range of each property so that the different values could be compared. We began by translating each property scale to range from 0 to 1. Unfortunately, though expectedly, this ended up highly favoring properties with low standard deviations. To

help offset this bias, we scaled each range by the inverse of its standard deviation. While this did not completely eliminate the bias, it significantly reduced it, as we will discuss later.

The third task was to read in the reference property file, a simple tab-delimited file of property values for each amino acid. This was done by creating a Perl module (Properties.pm). The next task was related – that of reading in the alignment files of each database. Again, a Perl module was created (Alignments.pm) to read in the varying formats of the HOMSTRAD, OXBench, and BAliBASE datasets. Each alignment was stored as an array of sites, where each site was a collection of single amino acid codes or '-' for gaps. Ambiguous characters, such as B (asparagines or aspartic acid), J (leucine or isoleucine), Z (glutamine or glutamic acid), and X (unspecified or unknown), which were only found in the BAliBASE dataset, were treated as gaps as they could add unwanted error to our results. By treating them as gaps, we effectively remove them from the analysis under our protocol.

Two approaches were used in this analysis in order to see the effect of gaps in amino acid properties, one where all sites that included gaps (and ambiguous characters) were removed and one that treated them as sites with fewer characters (sequences). For the ungapped analysis, the Perl Data Language was used due to its built-in statistical functions and efficient matrix operations. For the gapped analysis, where matrix operations could not be applied, a statistical module entitled Statistics::Descriptive was used for its standard deviation and mean functions.

In either approach, the following pseudo-code summarizes what is calculated for each alignment:

For each of the 503 properties

For each site in the alignment

Calculate the standard deviation if more than one data point exists

(i.e. is not a gap or ambiguous amino acid code)

Calculate the average standard deviation over all sites

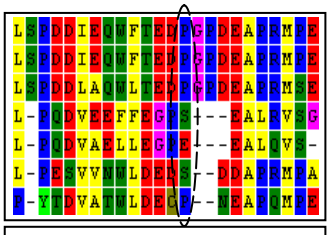
In general, the mean standard deviation for each property is calculated, by which the properties are ranked in increasing order. The top 10 properties for each alignment are then tallied independently for each of the three databases and then the three scores are combined for a total score. For a few sequences, once gaps are removed, there are no differences in the amino acid sequence. In such cases, the alignments are dropped and not included in the tally. The analysis is then repeated to calculate a tally for the top 1, top 5, top 25, and top 50 properties for each alignment.

A copy of the ranking program and associated modules is available from the author upon request.

2.2. Overview of the Property Profiling Method

Using PSI-BLAST, a MSA and its corresponding position-specific scoring matrix (PSSM) with sequences who share high sequence identity is constructed. From the given alignment we construct a "property profile" by scanning the alignment for regions of high conservation. Any sites where the protein of interest (i.e. reference sequence) contains a gap in the alignment are ignored. This profile is then used to search for homologous coding sequences (of local similarity to the given protein) who share a high degree of similarity in the conserved properties. This method is not meant to find distantly related proteins where little or no structural similarity is

retained; however, it will allow us to answer the question "what properties are important where?" and provide measures of property conservation within a family of protein sequences.



Construct an MSA

For each site, determine the mean value and variance for each property

Search for regions where properties have been conserved (variance below a given threshold) and of similar magnitude relative to the range in that particular property across the amino acids.

Construct a "property profile" for the alignment, consisting of "property regions" where a particular property has been conserved.

Search structure databases for proteins that match at least a portion of this property profile, scoring each based upon several criteria.

	P	P	P	S	E	S	P	ξ	s^2
alpha-CH chemical shifts	4.44	4.44	4.44	4.5	4.29	4.5	4.44	4.436	0.004929
Hydrophobicity index	1.95	1.95	1.95	0.05	0.47	0.05	1.95	1.196	0.904629
Signal sequence helical potential	0.76	0.76	0.76	0.97	0.11	0.97	0.76	0.727	0.083857
Membrane-buried preference param.	0.76	0.76	0.76	0.81	0.23	0.81	0.76	0.699	0.043248
Average flexibility indices	0.509	0.509	0.509	0.507	0.497	0.507	0.509	0.507	1.92E-05
Residue volume (Bigelow)	73.6	73.6	73.6	54.9	84.7	54.9	73.6	69.84	120.6295
Transfer free energy to surface	-0.98	-0.98	-0.98	-0.39	-0.3	-0.39	-0.98	-0.714	0.110729
Apparent partial specific volume	0.73	0.73	0.73	0.594	0.632	0.594	0.73	0.677	0.004506
Steric parameter	0.36	0.36	0.36	0.53	0.68	0.53	0.36	0.454	0.016329
Polarizability parameter	0.131	0.131	0.131	0.062	0.151	0.062	0.131	0.114	0.001322
Free energy of solution in water	-2.24	-2.24	-2.24	-0.524	1.77	-0.524	-2.24	-1.177	2.342917

Figure 2.1. A flowchart outlining the Property Profiling Method.

The back end of the homology detection program was built in C++ for speed, efficiency, and availability of mpiCC for parallel computing.

2.3. Constructing the Multiple Sequence Alignment

Using a perl script, PSI-BLAST (Altschul, Madden et al. 1997) is run and the resulting binary checkpoint (.chk) file is parsed into a text-based, tab-delimited PSSM. The advantage to using PSI-BLAST's checkpoint file over a standard sequence alignment is that the checkpoint file takes into account sequence weight, giving more weight to more divergent sequences, and gap frequency, placing zeroes in matrix columns with greater than 50% gap observance. It is expected that the improvements in the new property-based method ChemALIGN, implemented in the open source bioinformatics package PSODA, will replace PSI-BLAST in this protocol in the future (Snell 2007; Carroll 2009), though the adaptations mentioned would need to be incorporated to limit skewing by sampling bias. The increased accuracy of ChemALIGN will improve the detection of conserved regions, increasing the accuracy of our network model.

2.4. Constructing the Property Profile

The following definitions will be used throughout the remainder of the paper:

- A property region is a stretch of one or more amino acids where a single property exhibits high conservation
- A property profile is a collection of property regions representing a single protein sequence or subsequence

Property regions should be composed of sites where the property of interest is highly conserved, or, rather, has a low variance across the amino acids observed throughout evolutionary history at that site. The sites should also exhibit a similar magnitude. Here, the "importance" of a property region is based upon the variance of property magnitudes found at the corresponding residues of

structurally similar proteins. Therefore, to define our initial set of property regions, we scan through the columns of a PSSM (or checkpoint matrix in the case of PSI-BLAST) and locate regions where the variance of the amino acid property, weighted by the PSSM, is below a given threshold. The weighted variance is calculated as

$$\sigma^2_{r,weighted} = \sum_{i=1}^{20} w_{r,i}\left(x_i - \bar{x}_{weighted}\right)^2$$

where $w_{r,i}$ is the value in the normalized PSSM of amino acid i at position r and x_i is the rescaled property value of the amino acid. The property regions are then obtained using a seed-and-expand approach, where a stringent threshold is applied to seed the property region, and it is expanded on each side as long as the average variance is below a second, slightly relaxed, threshold (**Fig. 2.2**). These threshold parameters, like most of the other parameters used, are not statistically based at this point but rather given as inputs to the program, obtained using a training dataset.

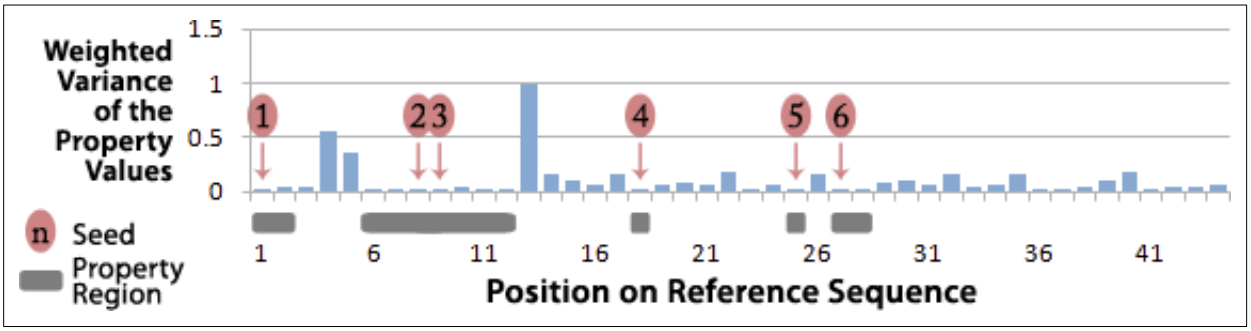


Figure 2.2. This figure illustrates the creation of property regions for a single amino acid physiochemical property index. Property regions are created by first finding a seed site where a property value is ultraconserved and then expanding the region until the average weighted variance of the property value

being studied surpasses a given threshold. Regions may contain more than one seed site such as seed sites 2 and 3 which are both in the second property region.

The goal in constructing a property profile is to create a data structure that is accurate and robust at detecting distant homology, informative at identifying property regions that are conserved throughout evolution, and fast at performing searches on a large database of sequences (with or without associated structures). The framework should also allow for the following information to be stored: the arrangement (in structure) or relative order (in sequence) of property regions, the distance (with some flexibility) between them, and their correlation with one another (for example, in the same domain). In order to meet these goals, a network-based approach is used to link property regions with one another. Since we are unable, at this point, to assign correlation directly to sequences, currently regions within a specified distance are linked together in a hierarchical fashion based upon the importance of each region (**Fig. 2.3**). This accounts for their relative order and distance, but not necessarily the grouping (as sub-networks) of regions into domains. The profile is constructed in the following manner:

1. connect all of the nodes within a distance *t* of one another (typically *t* = 2-4 residues)
2. Select top 8% of nodes ranked by "importance" (where "importance" = $1/\sigma^2_{r,weighted}$) to be the root nodes
3. Using an algorithm based on Dijkstra's Queue, find the shortest path from each node to a root node and remove all other connections to that node

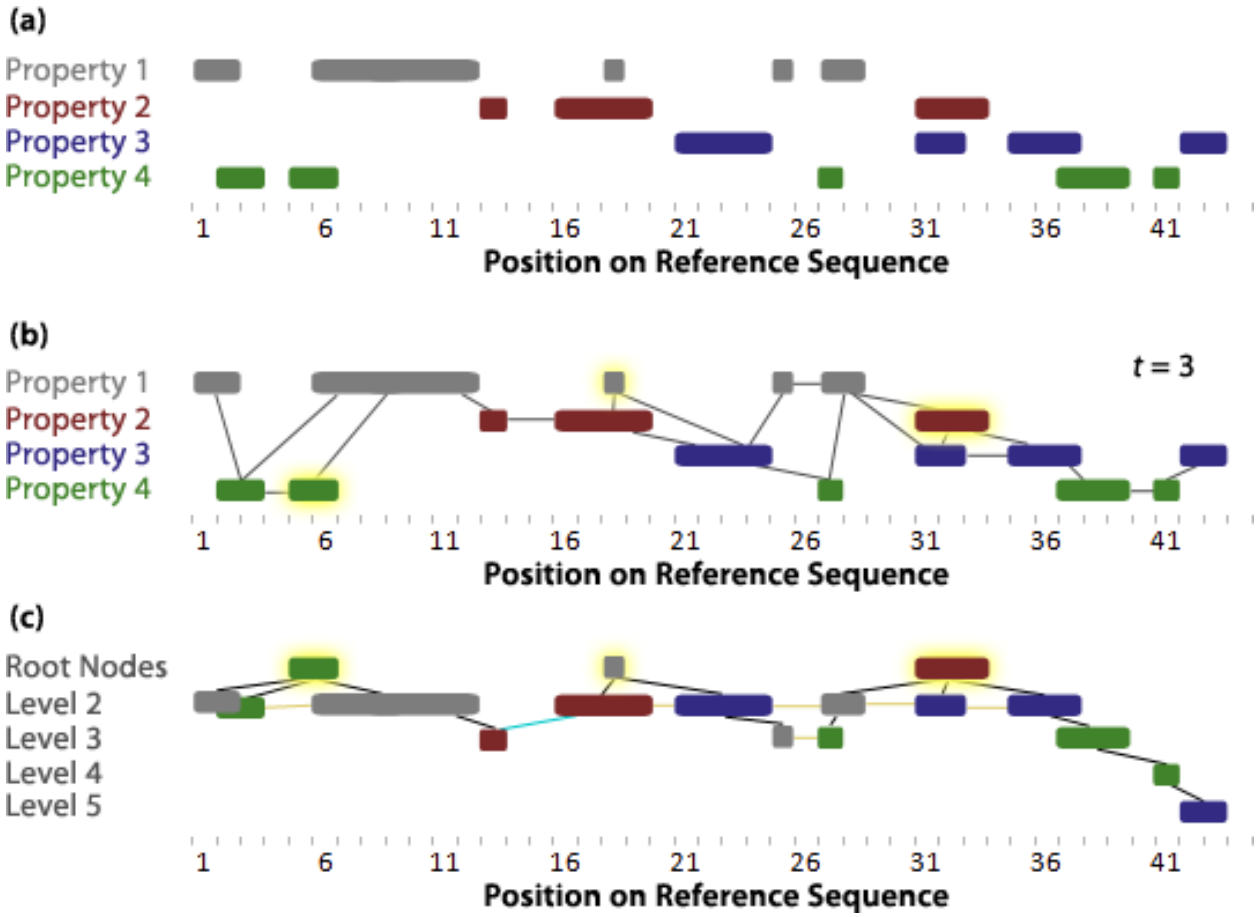


Figure 2.3: Property profiles are created from a set of property regions (a) by first linking nearby property regions within a distance *t* of one another (b), selecting ultraconserved regions to be the root nodes (highlighted in yellow), and removing sibling (beige lines) and foster parent (teal line) links (c). What remains is a set of rooted trees that can then be used in a fast top-down search.

In order to run the program from the command line, several parameters must be passed. First, the program requires a sequence or PSI-BLAST checkpoint file. If a PSI-BLAST checkpoint file is provided, it is used instead of re-running PSI-BLAST. This allows the user to modify the parameters of the property profile without having to re-run the much more intensive PSI-BLAST requirement of the program. Second, the program requires a property file. A default file is

included with the program, which will be used if one is not specified. Property files must be tab delimited files, with the first column containing property names, subsequent columns containing property values for each of the two amino acids, and the last column containing property weights (used to make the properties comparable to one another). Additional parameters that may be passed to the program include the tightness bound on the variance of each property region (default is 0.08), the distance cutoff for linked regions in the profile (default is 3 residues), and a flexibility multiplier (default is 3.2). The higher the tightness bound, in general, the larger each region will be. The higher the flexibility multiplier, the more property regions will be included in the profile, slightly increasing sensitivity but decreasing speed.

To obtain default parameter values, the program was trained on the Twilight Zone set from the SABmark database, which contains both positive and negative pairwise sequence alignments based on pairwise reference alignments from the consensus of SOFI and CE structural alignment programs. A genetic algorithm was run multiple times, utilizing the thorough search protocol (see below), for two to four days per run, until each parameter had converged to roughly the same value three times. Eleven runs were required to achieve this convergence.

2.5. Searching against a protein structure database

Using the property profile to search for homologous proteins performs a depth-first search of the "nodes", or regions, of the profile. It begins with regions of highest importance (the "root" nodes of the network. From there, it branches out, accumulating a higher score, based upon the importance of each region, as it locates additional linked regions within the sequence. Once two

consecutive regions are not found within a path, the path is abandoned and the next path is searched. The top *n* sequences with the highest scores are returned.

3. Results

3.1. Scoring physiochemical properties according to their biological relevance

The top results from both the ungapped and gapped analyses are reported in Tables 2.2 and 2.3. These results indicate the combined score of the three databases. The results are out of 1896 possible alignments: 1031 from HOMSTRAD, 672 from OXBench, and 193 from BAliBASE. The results indicate that in both ungapped and gapped analyses, the properties of *partition coefficient* and *alpha-NH chemical shifts* are both highly conserved in structural alignments. In all three databases, they were ranked number one and two respectively. From there, though the scores differ slightly from database to database and from ranking to ranking (number of times in top 10 vs. number of times in top 50, for example), the results reported in Tables 2.2 and 2.3 consistently were among those ranked in the top 20.

While the results of ungapped and gapped analyses differ slightly, in general they do not differ dramatically. The only possibly significant difference lies in the property *alpha helix propensity of position 44 in T4 lysozyme*, but even there we would need to perform a significance test to ensure its signal. It appears that properties that are significantly constrained/conserved in gapped regions are not significantly different from those of ungapped regions.

3.2. Benchmarking against a database of homologous proteins

To evaluate the usefulness and accuracy of the Property Profiler method, we used a dataset derived from the SCOP classification. It was obtained in a manner similar to that performed by Liao and Noble in 2003 to benchmark their SVM-Pairwise approach (Liao and Noble 2003), specifically by selecting sequences with less than 95% identity from the Astral database (http://astral.berkeley.edu), yielding 16,712 sequences grouped into families, superfamilies, and folds.

The benchmarking analysis used the 240 families that contained at least 15 members each. For each of those families, we calculated a mean ROC (receiver operating characteristic) value. ROC scores are calculated as the area under the curve of true positives as a function of false positives. Sequences in the same superfamily but not in the same family were removed and each family member was then compared to the remaining sequences using our method. The ROC score was then computed for each family member and the average value for the family was obtained (**Fig. 2.4**). This procedure was then used on the latest versions of the PSI-BLAST and SVM-Pairwise programs, two popular approaches for remote homology detection. In the SVM-Pairwise case, the removed superfamily sequences were used as a positive training set for each family. The mean ROC score for the three programs were calculated as 0.684, 0.833, and 0.691 for PSI-BLAST, SVM-Pairwise, and Property Profiler respectively, where a higher value indicates a more accurate separation of positive versus negative examples (0.5 being completely random). A discussion of these results will follow.

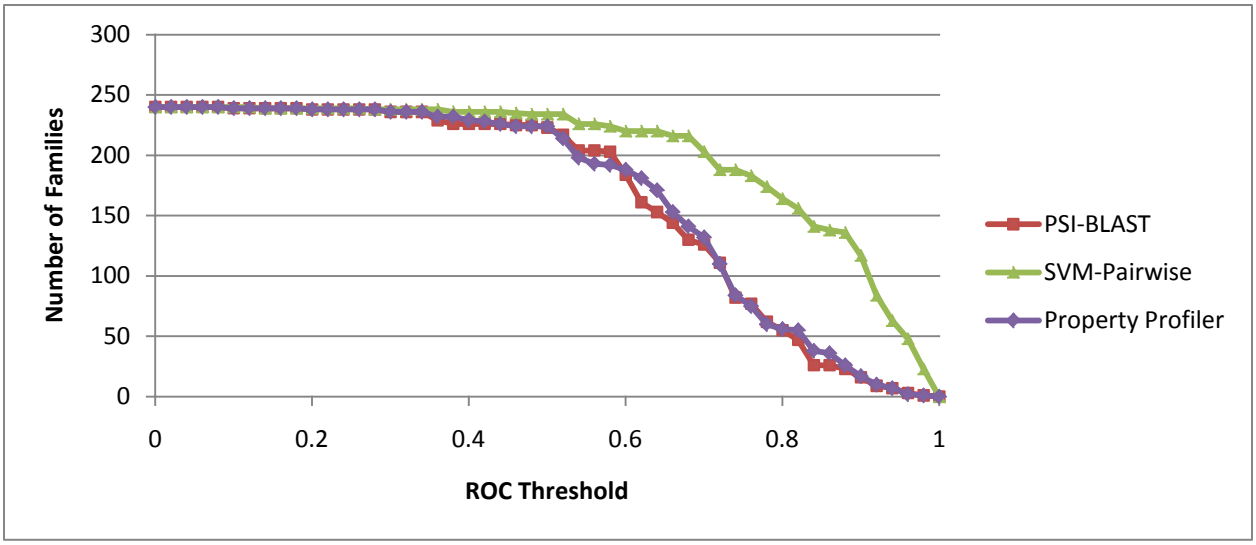


Figure 2.4: A comparison of the ROC (receiver operating characteristic) score distributions for three remote homology detection programs run on our dataset of 240 families.

4. Discussion

It is important to note that our handling of gaps reduces the number of sequences used in the calculation of some sites. Since the standard deviation is divided by *n*-1 (in contrast to *n*), the fewer the number of sites, the larger the standard deviation is likely to be. This effect did not seem to have a great effect on our results, but may be room for further studies where the magnitude of property differences is not the primary concern, but rather the significance of any differences. In addition, we would like to address questions on the effect of our property value scaling method on the results. Were the properties scaled enough to remove the bias in their distribution? When mapping the top ranked properties to a line, no obvious patterns are apparent which would lead to a suspicion of continuing bias (**Fig. 2.5**), but this needs to be statistically tested to be sure.

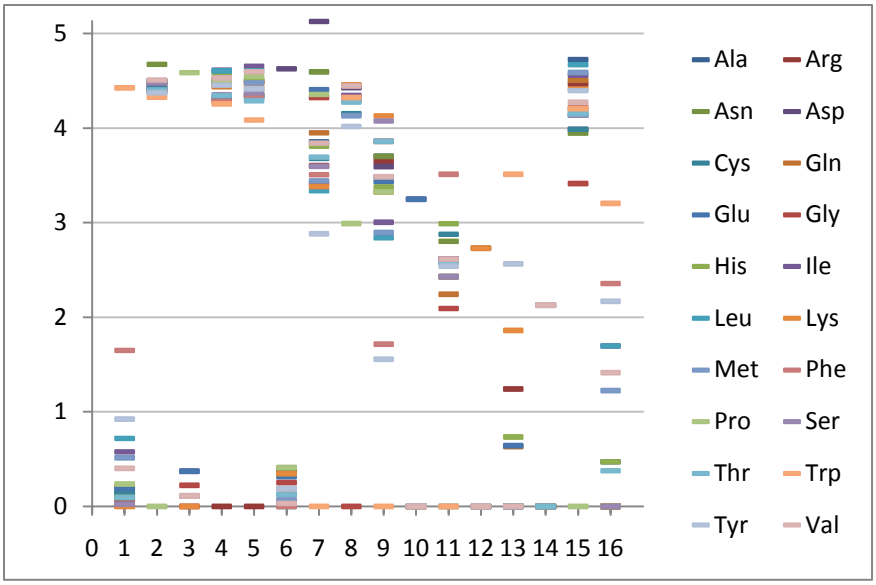


Figure 2.5: Property values from the combined results of the ungapped analysis. Columns 1–16 refer to the 16 properties in Table 2.2.

The success of the algorithm, as one would expect, is dependent upon the input. The robustness of the PSSM (returned by PSI-BLAST or other alignment algorithm) largely determines how accurate the resulting property profile will be and is the obvious reason for the similar results seen across many families between the PSI-BLAST and Property Profiler. Indeed, the more similar sequences available for a given protein of interest, the more accurate and powerful its property profile will be at detecting distant homologues and identifying regions of property conservation. On top of this dependence on a robust PSSM, the program lacks a robust statistical framework upon which to create an accurate property profile network. This is a common challenge in bioinformatics, and is a particular weakness in this approach. The heuristic of depending upon a given "threshold percentile" to determine whether a region is conserved or not is a quite rough approximation. By substituting this with a more rigorous statistical method, I believe the accuracy would increase substantially. Ongoing work will investigate the effect of

the threshold percentile and other variables on both profile creation and profile matching. Ultimately, I believe a main contribution of this work is to show that an alignment may be accurately represented by a network of property regions and that the two are practically interchangeable, at least in the case of homology detection.

The results from this paper provide a good starting point for a PCA analysis to determine which properties to use to achieve the greatest conservative signal in the least number of properties. Several properties that are highly correlated with another highly ranked property may likely be removed.

Eliminating highly correlated properties from use in the analysis (such as Averaged turn propensities in a transmembrane helix and Negative charge) would decrease the number of redundant property regions in each profile, increasing the search speed. A principal components analysis also has the potential to improve the efficiency of the program as the last PCA that was performed on amino acid properties was done over 40 years ago when the list of properties was sparse (Sneath 1966).

Table 2.1: Publicly available structural alignment programs

STAMP* (1992–1999) (Russell and Barton 1992)	http://www.compbio.dundee.ac.uk/Software/Stamp/stamp.html
MNYFIT* (Sutcliffe, Haneef et al. 1987)	http://www-cryst.bioc.cam.ac.uk/~joy/mnyfit.html
SSAP (1989–1996) and its successor, **SAP** (2000) (Taylor and Orengo 1989; Orengo and Taylor 1996; Taylor 2000)	http://mathbio.nimr.mrc.ac.uk

COMPARER (1990–1992) and its successor, **BATON** (Sali and Blundell 1990; Zhu, Sali et al. 1992)	http://www-cryst.bioc.cam.ac.uk/COMPARER
Structal (Subbiah, Laurents et al. 1993; Gerstein and Levitt 1998)	http://molmovdb.mbb.yale.edu/align
Protein3Dfit (Lessel and Schomburg 1994)	
DALI and **DaliLite** (Holm and Sander 1993; Holm and Park 2000)	http://ekhidna.biocenter.helsinki.fi/dali_server http://www.ebi.ac.uk/DaliLite
Pairwise Superposition of Protein 3D Structures (Boutonnet, Rooman et al. 1995)	http://wwwsup.scmbb.ulb.ac.be/~ocha/wwwsup1/wwwsup.cgi
ProSup (1996–2000) and its successor, **TopMatch** (2007) (Feng and Sippl 1996; Lackner, Koppensteiner et al. 2000; Sippl and Wiederstein 2008)	http://topmatch.services.came.sbg.ac.at
VAST (Gibrat, Madej et al. 1996)	http://www.ncbi.nlm.nih.gov/Structure/VAST
LSQMAN (Kleywegt 1996)	http://portray.bmc.uu.se/dejavu http://xray.bmc.uu.se/usf/dejavu.html
CE (Shindyalov and Bourne 1998)	http://cl.sdsc.edu
KENOBI (2000), **K2** (2002), and **K2SA** (Szustakowski and Weng 2000; Szustakowski and Weng 2002)	http://zlab.bu.edu/k2sa
FATCAT (2003–2006) (Ye and Godzik 2003; Ye and Godzik 2004; Ye and Godzik 2004)	http://fatcat.burnham.org
SSM (2003–2004) (Krissinel and Henrick 2004)	http://www.ebi.ac.uk/msd-srv/ssm
LGA: Local-Global Alignment (Zemla 2003)	http://PredictionCenter.llnl.gov/local/lga
GANGSTA (2003–2006) (Kolbeck, May et al. 2006)	http://gangsta.chemie.fu-berlin.de
SALIGN* (Marti-Renom, Madhusudhan et al. 2004)	http://salilab.org/DBAli/?page=tools&action=f_salign
MALECON* (Ochagavia and Wodak 2004)	
SuperPose* (Maiti, Van Domselaar et al. 2004) **TOPOFIT** (Ilyin, Abyzov et al. 2004)	http://wishart.biology.ualberta.ca/SuperPose http://mozart.bio.neu.edu/topofit
MultiProt* (Shatsky, Nussinov et al. 2004)	http://bioinfo3d.cs.tau.ac.il/MultiProt
CE-MC* (Guda, Lu et al. 2004)	http://pathway.rit.albany.edu/~cemc
POSA: Partial Order Structure Alignment* (Ye and Godzik 2005)	http://fatcat.burnham.org/POSA

FAST (Zhu and Weng 2005)	http://biowulf.bu.edu/FAST
3d-SS (Sumathi, Ananthalakshmi et al. 2006)	http://cluster.physics.iisc.ernet.in/3dss
Angle-Curve Alignment (Zhi, Krishna et al. 2006)	http://pops.burnham.org/curve
MUSTANG* (Konagurthu, Whisstock et al. 2006)	http://www.cs.mu.oz.au/~arun/mustang
OPAAS (Shih, Gan et al. 2006)	http://opaas.ibms.sinica.edu.tw
CPalign (Dundas, Binkowski et al. 2007)	http://bleezer.bioengr.uic.edu/salign
PROMALS3D* (Pei, Kim et al. 2008)	http://prodata.swmed.edu/promals3d

* supports multiple structures

Table 2.2: Combined results of ungapped analysis

Property	**1**	**5**	**10**	**25**	**50**
Partition coefficient (Garel et al.)	841	1679	1766	1787	1790
alpha-NH chemical shifts	472	1498	1664	1730	1754
Helix initiation parameter at position i,i+1,i+2	177	1452	1644	1725	1747
Activation Gibbs energy of unfolding, pH7.0	136	758	1062	1372	1510
Activation Gibbs energy of unfolding, pH9.0	30	626	1005	1346	1493
Averaged turn propensities in a transmembrane helix	37	340	730	1242	1425
Optimized propensity to form reverse turn	0	256	1013	1607	1714
Side chain angle theta(AAR)	26	313	683	1187	1386
RF value in high salt chromatography	0	227	950	1652	1752
Negative charge	26	278	655	1148	1353
Spin-spin coupling constants 3JHalpha-NH	0	202	898	1566	1685
Positive charge	29	226	629	1153	1396
Amphiphilicity index	0	219	768	1417	1631
Bitterness	14	237	624	1258	1535
Alpha helix propensity of position 44 in T4 lysozyme	0	199	696	1387	1573
Transfer energy, organic solvent-water	0	158	687	1537	1726

Table 2.3: Combined results of gapped analysis

Property	**1**	**5**	**10**	**25**	**50**
Partition coefficient (Garel et al.)	948	1696	1767	1787	1789
alpha-NH chemical shifts	414	1469	1632	1723	1748

Helix initiation parameter at posision i,i+1,i+2	127	1395	1621	1716	1742
Activation Gibbs energy of unfolding, pH7.0	141	809	1110	1411	1538
Activation Gibbs energy of unfolding, pH9.0	31	676	1063	1384	1528
RF value in high salt chromatography	0	279	1041	1672	1753
Optimized propensity to form reverse turn	0	268	1033	1610	1713
Averaged turn propensities in a transmembrane helix	34	302	675	1182	1369
Spin-spin coupling constants 3JHalpha-NH	0	199	945	1577	1694
Side chain angle theta(AAR)	26	290	618	1133	1321
Negative charge	24	273	643	1121	1327
Amphiphilicity index	0	233	815	1465	1647
Positive charge	28	243	634	1156	1378
Bitterness	13	243	663	1353	1643
Transfer energy, organic solvent-water	0	166	742	1589	1726

References

(June 7, 2007). "EMBL Nuleotide Sequence Database User Manual." Retrieved November 24, 2007, 2007, from http://www.ebi.ac.uk/embl/Documentation/User_manual/usrman.html.

(August 7, 2007). "Explanation of DDBJ flat file Format." Retrieved November 24, 2007, 2007, from http://www.ddbj.nig.ac.jp/sub/ref10-e.html.

(October 15, 2007). "NCBI-GenBank Flat File Release 162.0." Distribution Release Notes Retrieved November 20, 2007, 2007, from ftp://ftp.ncbi.nih.gov/genbank/gbrel.txt.

(October 23, 2006). "Sample GenBank Record." Retrieved November 24, 2007, 2007, from http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html.

. "Web BLAST page options." Retrieved January 4, 2008, 2008, from http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml#Reward-penalty.

Altschul, S. F., W. Gish, et al. (1990). "Basic Local Alignment Search Tool." Journal of Molecular Biology **215**(3): 403-410.

Altschul, S. F., T. L. Madden, et al. (1997). "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." Nucleic Acids Research **25**(17): 3389-3402.

Anfinsen, C. B. (1973). "Principles that Govern the Folding of Protein Chains." Science **181**(4096): 223-230.

Bahr, A., J. D. Thompson, et al. (2001). "BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations." Nucleic Acids Research **29**(1): 323-326.

Batemen, A. (2007). "Editorial." Nucleic Acids Research **35**(suppl_1): D1-2.

Baxevanis, A. D. and B. F. F. Ouellette (2005). Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins. Hoboken, New Jersey, John Wiley & Sons, Inc.

Bellman, R. (1952). "On the Theory of Dynamic Programming." Proc Natl Acad Sci U S A **38**(8): 716-719.

Boutonnet, N. S., M. J. Rooman, et al. (1995). "Optimal protein structure alignments by multiple linkage clustering: Application to distantly related proteins." Protein Engineering **8**(7): 647-662.

Cameron, M., H. E. Williams, et al. (2004-2006, March 8, 2006). "FSA BLAST." Retrieved January 3, 2008, 2008, from http://www.fsa-blast.org.

Carroll, H. (2009). ChemAlign: Biologically Relevant Multiple Sequence Alignment Using Physicochemical Properties.

Chivian, D. and D. Baker (2006). "Homology modeling using parametric alignment ensemble generation with consensus and energy-based model selection." Nucleic Acids Res **34**(17): e112.

Contreras-Moreira, B., P. W. Fitzjohn, et al. (2003). "In silico protein recombination: enhancing template and sequence alignment selection for comparative protein modelling." J Mol Biol **328**(3): 593-608.

Das, R., Q. Bin, et al. (2007). "Structure prediction for CABP7 targets using extensive all-atom refinement with Rosetta@home." Proteins-Structure Function and Bioinformatics **69**: 118-128.

Dayhoff, M. O., R. V. Eck, et al. (1965). Atlas of Protein Sequence and Structure. Silver Spring, MD., National Biomedical Research Foundation.

Dayhoff, M. O., R. M. Schwartz, et al. (1978). A Model of Evolutionary Change in Proteins. Atlas of protein sequence and structure. N. B. R. Foundation. Silver Spring, MD, National Biomedical Research Foundation. **5:** 345-358.

de Bakker, P. I. W., A. Bateman, et al. (2001). "HOMSTRAD: adding sequence information to structure-based alignments of homologous protein families." Bioinformatics **17**(8): 748-749.

Dundas, J., T. A. Binkowski, et al. (2007). "Topology independent protein structural alignment." BMC Bioinformatics **8**: 388.

Edgar, R. C. (2004). "MUSCLE: multiple sequence alignment with high accuracy and high throughput." Nucleic Acids Research **32**(5): 1792-1797.

Edgar, R. C. and K. Sjolander (2004). "COACH: profile-profile alignment of protein families using hidden Markov models." Bioinformatics **20**(8): 1309-1318.

Feng, Z. K. and M. J. Sippl (1996). "Optimum superimposition of protein structures: Ambiguities and implications." Folding & Design **1**(2): 123-132.

Galperin, M. Y. (2007). "The Molecular Biology Database Collection: 2007 update." Nucleic Acids Research **35**: D3-D4.

Gerstein, M. and M. Levitt (1998). "Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins." Protein Science **7**(2): 445-456.

Gibrat, J. F., T. Madej, et al. (1996). "Surprising similarities in structure comparison." Current Opinion in Structural Biology **6**(3): 377-385.

Gish, W. (1996-2004, March 22, 2006). "WU BLAST 2.0." Retrieved January 3, 2008, 2008, from http://blast.wustl.edu.

Guda, C., S. F. Lu, et al. (2004). "CE-MC: a multiple protein structure alignment server." Nucleic Acids Research **32**: W100-W103.

Henikoff, S. and J. G. Henikoff (1992). "Amino Acid Substitution Matrices from Protein Blocks." Proceedings of the National Academy of Sciences of the United States of America **89**(22): 10915-10919.

Hersh, R. T. (1967). "Reviews." Systematic Zoology **16**(3): 262-263.

Hoffmann, R. and A. Valencia (2004). "A Gene Network for Navigating the Literature." Nature Genetics **36**(7): 664-664.

Holm, L. and J. Park (2000). "DaliLite workbench for protein structure comparison." Bioinformatics **16**(6): 566-567.

Holm, L. and C. Sander (1993). "Protein structure comparison by alignment of distance matrices." Journal of Molecular Biology **233**(1): 123-138.

Ilyin, V. A., A. Abyzov, et al. (2004). "Structural alignment of proteins by a novel TOPOFIT method, as a superimposition of common volumes at a topomax point." Protein Science **13**(7): 1865-1874.

Jaroszewski, L., W. Li, et al. (2002). "In search for more accurate alignments in the twilight zone." Protein Sci **11**(7): 1702-1713.

Jaroszewski, L., L. Rychlewski, et al. (2000). "Improving the quality of twilight-zone alignments." Protein Sci **9**(8): 1487-1496.

Jaroszewski, L., L. Rychlewski, et al. (1998). "Fold prediction by a hierarchy of sequence, threading, and modeling methods." Protein Sci **7**(6): 1431-1440.

Jauch, R., H. C. Yeo, et al. (2007). "Assessment of CASP7 structure predictions for template free targets." Proteins **69 Suppl 8**: 57-67.

John, B. and A. Sali (2003). "Comparative protein structure modeling by iterative alignment, model building and model assessment." Nucleic Acids Res **31**(14): 3982-3992.

Jones, D. T. (1999). "GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences." J Mol Biol **287**(4): 797-815.

Kawashima, S. and M. Kanehisa (2000). "AAindex: Amino acid index database." Nucleic Acids Research **28**(1): 374-374.

Kawashima, S., H. Ogata, et al. (1999). "AAindex: Amino Acid Index Database." Nucleic Acids Research **27**(1): 368-369.

Kawashima, S., P. Pokarowski, et al. (2008). "AAindex: amino acid index database, progress report 2008." Nucleic Acids Research **36**: D202-D205.

Kleywegt, G. J. (1996). "Use of non-crystallographic symmetry in protein structure refinement." Acta Crystallographica Section D-Biological Crystallography **52**: 842-857.

Kolbeck, B., P. May, et al. (2006). "Connectivity independent protein-structure alignment: a hierarchical approach." Bmc Bioinformatics **7**.

Konagurthu, A. S., J. C. Whisstock, et al. (2006). "MUSTANG: A multiple structural alignment algorithm." Proteins-Structure Function and Bioinformatics **64**(3): 559-574.

Krissinel, E. and K. Henrick (2004). "Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions." Acta Crystallographica Section D-Biological Crystallography **60**: 2256-2268.

Kryshtafovych, A., K. Fidelis, et al. (2007). "Progress from CASP6 to CASP7." Proteins **69 Suppl 8**: 194-207.

Lackner, P., W. A. Koppensteiner, et al. (2000). "ProSup: a refined tool for protein structure alignment." Protein Engineering **13**(11): 745-752.

León, D. and S. Markel (2003). Sequence Analysis in a Nutshell. Sebastopol, CA, O'Reilly & Associates, Inc.

Lessel, U. and D. Schomburg (1994). "Similarities between protein 3-D structures." Protein Engineering **7**(10): 1175-1187.

Letunic, I., R. R. Copley, et al. (2004). "SMART 4.0: towards genomic data integration." Nucleic Acids Research **32**: D142-D144.

Liao, L. and W. S. Noble (2003). "Combining pairwise-sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships." Journal of Computational Biology **10**(6): 857-868.

Ma, B., J. Tromp, et al. (2002). "PatternHunter: faster and more sensitive homology search." Bioinformatics **18**(3): 440-445.

Madden, T. (2002, August 13, 2003). "The BLAST Sequence Analysis Tool." The NCBI Handbook Retrieved January 4, 2008, 2008, from http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=handbook.

Maiti, R., G. H. Van Domselaar, et al. (2004). "SuperPose: a simple server for sophisticated structural superposition." Nucleic Acids Research **32**: W590-W594.

Marti-Renom, M. A., M. S. Madhusudhan, et al. (2004). "Alignment of protein sequences by their profiles." Protein Science **13**(4): 1071-1087.

Mathura, V. S. and D. Kolippakkam (2005). "APDbase: Amino acid Physico-chemical properties Database." Bioinformation **1**(1): 2-4.

Menlove, K. J., M. Clement, et al. (2009). "Similarity Searching Using BLAST." Bioinformatics for DNA Sequence Analysis: 1-22.

Mizuguchi, K., C. M. Deane, et al. (1998). "HOMSTRAD: A database of protein structure alignments for homologous families." Protein Science **7**(11): 2469-2471.

Needleman, S. B. and C. D. Wunsch (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins." Journal of Molecular Biology **48**(3): 443-453.

Ochagavia, M. E. and H. Wodak (2004). "Progressive combinatorial algorithm for multiple structural alignments: Application to distantly related proteins." Proteins-Structure Function and Bioinformatics **55**(2): 436-454.

Onuchic, J. N., Z. Luthey-Schulten, et al. (1997). "Theory of protein folding: The energy landscape perspective." Annual Review of Physical Chemistry **48**: 545-600.

Orengo, C. A. and W. R. Taylor (1996). SSAP: Sequential structure alignment program for protein structure comparison. Computer Methods for Macromolecular Sequence Analysis. **266:** 617-635.

Panchenko, A. R., A. Marchler-Bauer, et al. (2000). "Combination of threading potentials and sequence profiles improves fold recognition." J Mol Biol **296**(5): 1319-1331.

Pearson, W. R. and D. J. Lipman (1988). "Improved tools for biological sequence comparison." Proceedings of the National Academy of Sciences of the United States of America **85**(8): 2444-2448.

Pei, J., B.-H. Kim, et al. (2008). "PROMALS3D: a tool for multiple protein sequence and structure alignments." Nucleic Acids Research **36**(7): 2295-2300.

Raghava, G. P. S., S. M. J. Searle, et al. (2003). "OXBench: A benchmark for evaluation of protein multiple sequence alignment accuracy." Bmc Bioinformatics **4**.

Roy-Engel, A. M., M. L. Carroll, et al. (2001). "Alu insertion polymorphisms for the study of human genomic diversity." Genetics **159**(1): 279-290.

Russell, R. B. and G. J. Barton (1992). "Multiple Sequence Alignment from Tertiary Structure Comparison: Assignment of Global and Residue Confidence Levels." Proteins: Structure Function and Genetics **14**(2): 309-323.

Sali, A. and T. L. Blundell (1990). "Definition of general topological equivalence in protein structures - a procedure involving comparison of properties and relationships through simulated annealing and dynamic-programming." Journal of Molecular Biology **212**(2): 403-428.

Shatsky, M., R. Nussinov, et al. (2004). "A method for simultaneous alignment of multiple protein structures." Proteins-Structure Function and Bioinformatics **56**(1): 143-156.

Shih, E. S. C., R. C. R. Gan, et al. (2006). "OPAAS: a web server for optimal, permuted, and other alternative alignments of protein structures." Nucleic Acids Research **34**: W95-W98.

Shindyalov, I. N. and P. E. Bourne (1998). "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path." Protein Engineering **11**(9): 739-747.

Sippl, M. J. and M. Wiederstein (2008). "A note on difficult structure alignment problems." Bioinformatics **24**(3): 426-427.

Smith, T. F. and M. S. Waterman (1981). "Identification of common molecular subsequences." Journal of Molecular Biology **147**(1): 195-197.

Sneath, P. H. A. (1966). "Relations between chemical structure and biological activity in peptides." Journal of Theoretical Biology **12**(2): 157-195.

Snell, H. C. M. E. M. C. Q. (2007). "PSODA: Better Tasting and Less Filling Than PAUP." Proceedings of the Biotechnology and Bioinformatics Symposium **3**(1).

Stebbings, L. A. and K. Mizuguchi (2004). "HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database." Nucleic Acids Research **32**: D203-D207.

Subbiah, S., D. V. Laurents, et al. (1993). "Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core." Current Biology **3**(3): 141-148.

Sumathi, K., P. Ananthalakshmi, et al. (2006). "3dSS: 3D structural superposition." Nucleic Acids Research **34**: W128-W132.

Sutcliffe, M. J., I. Haneef, et al. (1987). "Knowledge based modeling of homologous proteins, part I: Three-dimensional frameworks derived from the simultaneous superposition of multiple structures." Protein Engineering **1**(5): 377-384.

Szustakowski, J. D. and Z. P. Weng (2000). "Protein structure alignment using a genetic algorithm." Proteins-Structure Function and Genetics **38**(4): 428-440.

Szustakowski, J. D. and Z. P. Weng (2002). Protein structure alignment using evolutionary computing. Evolutionary Computation in Bioinformatics.

Tamura, K., J. Dudley, et al. (2007). "MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) software version 4.0." Mol Biol Evol **24**(8): 1596-1599.

Taylor, W. R. (2000). Protein structure prediction: methods and protocols. Methods in Molecular Biology. D. M. Webster. Totowa, N.J., Humana Press, Inc. **143:** 19-32.

Taylor, W. R. and C. A. Orengo (1989). "Protein structure alignment." Journal of Molecular Biology **208**(1): 1-22.

Thompson, J. D., P. Koehl, et al. (2005). "BAliBASE 3.0: Latest developments of the multiple sequence alignment benchmark." Proteins-Structure Function and Bioinformatics **61**(1): 127-136.

Thompson, J. D., F. Plewniak, et al. (1999). "BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs." Bioinformatics **15**(1): 87-88.

Tompa, M., N. Li, et al. (2005). "Assessing computational tools for the discovery of transcription factor binding sites." Nature Biotechnology **23**(1): 137-144.

Van Walle, I., I. Lasters, et al. (2005). "SABmark - a benchmark for sequence alignment that covers the entire known fold space." Bioinformatics **21**(7): 1267-1268.

Wheeler, D. G. (2003). "Selecting the right protein scoring matrix." Current Protocols in Bioinformatics: 3.5.1-3.5.6.

Woolley, S., J. Johnson, et al. (2003). "TreeSAAP: Selection on Amino Acid Properties using phylogenetic trees." Bioinformatics **19**(5): 671-672.

Ye, Y. and A. Godzik (2003). "Flexible structure alignment by chaining aligned fragment pairs allowing twists." Bioinformatics **19**(suppl_2): ii246-255.

Ye, Y. Z. and A. Godzik (2004). "Database searching by flexible protein structure alignment." Protein Science **13**(7): 1841-1850.

Ye, Y. Z. and A. Godzik (2004). "FATCAT: a web server for flexible structure comparison and structure similarity searching." Nucleic Acids Research **32**: W582-W585.

Ye, Y. Z. and A. Godzik (2005). "Multiple flexible structure alignment using partial order graphs." Bioinformatics **21**(10): 2362-2369.

Yona, G. and M. Levitt (2002). "Within the twilight zone: a sensitive profile-profile comparison tool based on information theory." J Mol Biol **315**(5): 1257-1275.

Zemla, A. (2003). "LGA: a method for finding 3D similarities in protein structures." Nucleic Acids Research **31**(13): 3370-3374.

Zhi, D. G., S. S. Krishna, et al. (2006). "Representing and comparing protein structures as paths in three-dimensional space." Bmc Bioinformatics **7**.

Zhu, J. H. and Z. P. Weng (2005). "FAST: A novel protein structure alignment algorithm." Proteins-Structure Function and Bioinformatics **58**(3): 618-627.

Zhu, Z. Y., A. Sali, et al. (1992). "A variable gap penalty-function and feature weights for protein 3-D structure comparisons." Protein Engineering **5**(1): 43-51.

Zwanzig, R., A. Szabo, et al. (1992). "Levinthal's paradox." Proc Natl Acad Sci U S A **89**(1): 20-22.