Using NCBI BLAST

Naomi A. Stover[1] and Andre R.O. Cavalcanti[2]

[1]Bradley University, Peoria, Illinois
[2]Pomona College, Claremont, California

BLAST is the most widely used software in bioinformatics research. Its main function is to compare a sequence of interest, the query sequence, to sequences in a large database. BLAST then reports the best matches, or "hits," found in the database. This simple program has two primary applications. First, if the function of the query sequence is unknown, it may be possible to infer its function based on the recognized functions of similar sequences. Second, if the researcher has a query sequence with a known function, it may be possible to identify sequences in the database that have similar functions. The utility of BLAST therefore depends on the researcher's choice of query sequence and database. An appreciation for the functions and limitations of BLAST is vital to using this program effectively. This unit will introduce the basic concepts behind BLAST, walk through BLAST searching protocols, and interpret common results.

Keywords: BLAST • sequence alignment • sequence analysis • sequence annotation

How to cite this article:
Stover, N. A. and Cavalcanti, A. R. O. 2017. Using NCBI BLAST
Curr. Protoc. Essential Lab. Tech. 14:11.1.1-11.1.34.
doi: 10.1002/cpet.8

OVERVIEW AND PRINCIPLES

While the information contained in strings of nucleotide and protein sequences is meaningful to cells, no human being can glean information from these sequences without the use of a computer and bioinformatics software. BLAST (Basic Local Alignment Search Tool) is a quick and easy program that aligns any sequence with those in a large database and then scores the strength of the alignment. A simple BLAST search usually provides enough information to perform two of the simple tasks that are nearly universal when working with genes and genome sequences: (1) determining the putative function of a newly identified DNA or protein sequence; and (2) determining if a genome (or other large database of sequences) contains a sequence similar to a known gene.

Introduction to BLAST

The sequence of a protein, RNA, or DNA is a window into its function and its history. The unique order, length, and composition of subunits in a protein (its "primary structure") largely determine its binding and catalytic properties. These same features of a nucleic acid in turn dictate the protein it encodes. Nucleic acid and protein sequences change very little over time, thanks to the fidelity of the enzymes involved and the phenotypic consequences of mistakes (natural selection). The collective efforts of the scientific community to catalog the functions of proteins and genes from a vast array of organisms now ensures that many newly sequenced proteins and genes will resemble a sequence that has been previously described. The ability to infer the properties of new proteins and genes based on sequence comparison is one of the bases of the field of bioinformatics.

Current Protocols Essential Laboratory Techniques 11.1.1-11.1.34, May 2017
Published online May 2017 in Wiley Online Library (wileyonlinelibrary.com).
doi: 10.1002/cpet.8

BLAST (Altschul et al., 1990, 1997) is a program that searches for regions of sequence similarity between two nucleotide or protein sequences. Most commonly, BLAST is used to compare a researcher's sequence (the "query" sequence) to sequences in a large database, such as an entire genome sequence, or even the collection of all sequences submitted to GenBank (see *UNIT 11.2*; Chang et al., 2016). BLAST then reports to the researcher the best matches, or "hits," found in the database. Because so many different inferences can be made upon determining specific relationships between two sequences, BLAST searches are often a simple and effective way to reveal a variety of gene and protein features. Common questions that can be answered by a BLAST search will be covered later in this unit. First, how does BLAST work?

BLAST Scoring and Alignments

In order to evaluate the degree of identity that a query sequence shares with each sequence in a database, the BLAST program must first align these sequences correctly. Not every sequence in a database has the same length or begins at the same point as the query sequence. Obviously, scoring the number of nucleotides or amino acids shared by two sequences is dependent on first correctly identifying which residues should be compared, then judging how similar two residues are to one another.

A variety of scoring systems have been used over the years to determine how different residues are scored to find the best alignment between two sequences. The simplest possible system would score one point for each column that contained an identical nucleotide or amino acid; the alignment that places the maximum number of identical residues side by side would have the highest score. A very similar scoring system is still used today for comparing nucleotide sequences.

For amino acids, such a scoring system is too naïve—proteins are made of 20 different amino acid subunits with very different chemical properties. While alignments could be scored on amino acid identities alone, many amino acids can be substituted for one another with little effect on the function of the protein. While these are not strict identities, substitutions of similar amino acids for one another provide additional information that can be used to find the best alignment.

For an example, look at the first two protein alignments in Figure 11.1.1. They are identical with the exception of the second alignment position. In the alignment shown in Figure 11.1.1A, the second column has an Ala in the first sequence and an Arg in the second sequence. In the alignment shown in Figure 11.1.1B, the first sequence has an Ala in the second column, and the second sequence has a Ser. A quick check in any biochemistry textbook will show that Ala resembles Ser far more than it does Arg. Alignment algorithms thus need to include a metric to score how similar two amino acids are.

Since there are many ways to define similarities and differences between amino acids (e.g., hydrophobicity, size, chemical composition, etc.), which property should be used when aligning two sequences? Over the years, a number of scoring matrices have been

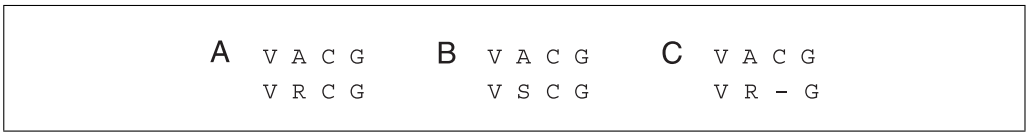

Figure 11.1.1 Example alignments. (**A**) The two sequences differ in the second column by the change of an Ala to an Arg. (**B**) The two sequences differ in the second column by the change of an Ala to a Ser. (**C**) The Cys in the third column of the first sequence is aligned with a gap in the second sequence. This could be caused either by an insertion in the first sequence or by a deletion in the second.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	−1	−2	−2	0	−1	−1	0	−2	−1	−1	−1	−1	−2	−1	1	0	−3	−2	0
R	−1	5	0	−2	−3	1	0	−2	0	−3	−2	2	−1	−3	−2	−1	−1	−3	−2	−3
N	−2	0	6	1	−3	0	0	0	1	−3	−3	0	−2	−3	−2	1	0	−4	−2	−3
D	−2	−2	1	6	−3	0	2	−1	−1	−3	−4	−1	−3	−3	−1	0	−1	−4	−3	−3
C	0	−3	−3	−3	9	−3	−4	−3	−3	−1	−1	−3	−1	−2	−3	−1	−1	−2	−2	−1
Q	−1	1	0	0	−3	5	2	−2	0	−3	−2	1	0	−3	−1	0	−1	−2	−1	−2
E	−1	0	0	2	−4	2	5	−2	0	−3	−3	1	−2	−3	−1	0	−1	−3	−2	−2
G	0	−2	0	−1	−3	−2	−2	6	−2	−4	−4	−2	−3	−3	−2	0	−2	−2	−3	−3
H	−2	0	1	−1	−3	0	0	−2	8	−3	−3	−1	−2	−1	−2	−1	−2	−2	2	−3
I	−1	−3	−3	−3	−1	−3	−3	−4	−3	4	2	−3	1	0	−3	−2	−1	−3	−1	3
L	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4	−2	2	0	−3	−2	−1	−2	−1	1
K	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5	−1	−3	−1	0	−1	−3	−2	−2
M	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5	0	−2	−1	−1	−1	−1	1
F	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6	−4	−2	−2	1	3	−1
P	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7	−1	−1	−4	−3	−2
S	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4	1	−3	−2	−2
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5	−2	−2	0
W	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11	2	−3
Y	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	−1
V	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

Figure 11.1.2 The Blosum62 scoring matrix used by default by BLAST to align two protein sequences. The score for aligning two amino acids can be found by the intersection of the rows and columns of the matrix. Positive numbers mean that the two amino acids substitute one another frequently; negative numbers mean that the two amino acids rarely substitute one another. The score of an alignment can be calculated by summing the scores of each column of the alignment. Using this matrix, the alignment in Figure 11.1.1A has a score of 18 (V–V = 4; C–C = 9; A–R = –1; G–G = 6), while that in Figure 11.1.1B has a score of 20 (V–V = 4; C–C = 9; A–S = 1; G–G = 6).

developed in order to properly align protein sequences; these strategies are covered in depth in Wheeler (2003). In 1978, Margaret Dayhoff developed a simple scoring matrix by using closely related sequences that were easy to align, and then determined the frequency with which each amino acid substituted another in real sequences (Dayhoff et al., 1978). Using this approach completely avoids the problem of choosing a metric. Two amino acids that frequently substitute one another evolutionarily are considered similar, whereas amino acids that almost never substitute each other are considered dissimilar. Her efforts generated the PAM matrices still widely used. Henikoff and Henikoff (1992) developed another set of matrices, the BLOSUM matrices, which were also based on observed substitution frequencies between amino acids in biological sequences. The derivation of these matrices is beyond the scope of this unit, but because each PAM and BLOSUM matrix is derived using a slightly different methodology, specific matrices are better suited for aligning either diverged or more similar sequences. Figure 11.1.2 shows the BLOSUM62 matrix, which is the default matrix used by BLAST when aligning proteins. For a nice introduction on how the BLOSUM matrices are derived, see Eddy (2004a).

Each PAM or BLOSUM matrix is followed by a number, for example, PAM250 or BLOSUM62. Although the significance of these numbers is beyond the scope of this unit, it is important to know that PAM matrices with a larger number are better suited to align more diverged sequences and BLOSUM matrices with larger numbers are better suited to align closely related sequences. For example, the PAM250 matrix is equivalent to the BLOSUM45, while the PAM120 is equivalent to the BLOSUM80 (see Wheeler, 2003, and references therein).

Besides determining the probability of substituting one amino acid by another, substitution matrices also need to account for the possibility of aligning an amino acid with a gap, as illustrated in Figure 11.1.1C. Biologically, this happens when a sequence has an

insertion or a deletion. Ideal gap values and gap scoring schemes have been empirically determined and their derivation is beyond the scope of this unit.

Now that a reward or penalty for observing two amino acids (or an amino acid and a gap) in the same position in the two sequences can be determined, how can the equivalent positions in the two sequences be determined? Needleman and Wunsch (1970) developed a mathematically rigorous method, based on a technique called dynamic programming, which is guaranteed to find the best alignment between two sequences given a certain substitution matrix. The Needleman and Wunsch method aligns the entirety of two sequences; Smith and Waterman (1981) modified this method to find similar regions between two sequences. Instead of aligning the full sequences, the Smith-Waterman algorithm compares segments of the two sequences and returns the most similar segments between the sequences (for an introductory review on dynamic programming, see Eddy, 1978). Theoretically, this method could be used to search a database for sequences similar to the one of interest.

In practice, considering the enormous number of sequences in the databases typically queried in a BLAST search, the exact Smith-Waterman algorithm would require too much computer time. The BLAST programs, thus, have been optimized in such a way that a search of even the largest databases, like the nonredundant (nr) database maintained by GenBank, can take mere seconds. BLAST achieves this by using some approximations (heuristics) to keep computing times manageable. Although these approximations mean that we cannot mathematically prove that BLAST will return the best hit in the database, the BLAST algorithm has been shown to work very well and is able to detect similar sequences in the database very fast and reliably (Altschul et al., 1997).

In the Smith-Waterman method, we only need to worry about one parameter, the substitution matrix. The algorithm does all the rest and returns the best possible alignment of the sequences under that scoring scheme. Because BLAST uses heuristics to achieve its speed, it introduces several parameters that have to be optimized. Normally we will not need to worry about these extra parameters because the default values in the BLAST Web server work for most applications, and the program even knows how to change the parameters to give better results. Thus, we will not discuss any of these advanced parameters, and will instead use BLAST with default parameters throughout this unit. For a more detailed review of alignment algorithms, scoring matrices, and the BLAST heuristics, see Korf et al. (2003) or read Chapter 3 in *Current Protocols in Bioinformatics* (Bateman et al., 2017), which contains several units covering alignment algorithms and BLAST.

Comparing BLAST Results

When a sequence is submitted to a BLAST server, the BLAST program compares the query with the first "target" sequence in the database. The program first quickly determines if the sequences have any region of similarity; if they do not, the target sequence is not further analyzed. If the query and target share some sequence similarities, they are aligned, and this alignment is scored according to the selected scoring matrix and an elaborate system that takes into account the number and spacing of shared residues. The score of the alignment between the query and the first sequence is saved, and BLAST moves on to the next sequence. The top scoring sequences are displayed to the researcher at the end of the BLAST search.

For each target sequence, also known as a "BLAST hit," BLAST displays two types of scores to the user. The first is the Bit Score, which is calculated directly from the alignment of the query to the target sequence by summing the scores of each column of the alignment. The second score reported by BLAST is the E-value, or "expected" value.

The E-value represents how many sequences you would expect to find in the database with a Bit Score as large as, or larger than, the one of the hit by pure chance. The E-value is calculated using the Bit Score and takes into account the length of the query sequence and the size of the database. However, unlike the Bit Score, where a larger number represents greater similarity between two sequences, a hit with a lower E-value more closely resembles your query sequence.

It is important to understand the difference between the Bit Score and the E-value. The Bit Score refers to an alignment between two sequences (the query and the hit), and should be the same regardless of the database used. The E-value depends on the database and query size. If you search a small database with a query sequence, there is a defined probability that you will find a hit with a given score by pure chance. If you search a larger database using the same query, you are more likely to find a hit with the same score because there are more opportunities for your sequence to align in a much larger pool of sequences. The E-value reflects exactly this; a given Bit Score will generate a larger (less significant) E-value when searching a larger database.

Usually, when interpreting BLAST results, the most important factor is the E-value of the hits. The lower the E-value, the more likely the query sequence is related by common ancestry to the hit and the more confident you should be in making inferences about the query based on its hits.

In the remainder of this unit, we will present several protocols on how to run BLAST searches and how to use the different databases available at NCBI. We will show how to run BLAST using a protein or a nucleotide sequence as a query against a protein or nucleotide database. We will also explain how to interpret the results of a BLAST search and discuss some special cases and troubleshooting techniques to help you obtain useful results. Keep in mind that the best way to learn how to use BLAST is by trying it, experimenting with different searches and parameters, following the links in the results pages, and really getting your feet wet.

STRATEGIC QUESTIONS

Before starting your BLAST search, have answers to the following questions (detailed in Strategic Planning):

1. What information are you hoping to get from a BLAST search?
2. Which BLAST program is most suitable?
3. Which sequence database do you want to search?

STRATEGIC PLANNING

Information from BLAST Searches

BLAST searches are performed for two primary reasons. First, BLAST can be used to identify the function of an unknown sequence. For example, when a researcher sequences a newly identified or cloned DNA fragment, a BLAST search is typically performed to determine the identity of the sequence. If the DNA in question is a mutant generated in the laboratory, a BLAST search may simply confirm that certain residues have been changed. If the DNA was identified in a library screen, the search results may allow the researcher to predict a function for the gene. In both cases, a BLAST search quickly makes sense of the otherwise meaningless string of nucleotides. The other use for a BLAST search is to determine if a certain gene or protein sequence is present in a database. For example, the sequence of a specific kinase protein in a mouse can be used to search for a similar protein in a list of protein sequences encoded in the human genome. Selecting the proper BLAST program and database are key to utilizing BLAST for these and other purposes.

Many Web sites offer users the chance to BLAST their sequences against specific datasets. Model organism databases contain curated genome (see *UNIT 11.4*; Engel and McPherson, 2016), coding, and protein sequence datasets for a single species. However, while searching the mouse genome, for example, is often useful or desirable, many BLAST searches must be performed against much larger databases in order to return more accurate or meaningful hits. The largest sequence databases in the world can be searched at the National Center for Biotechnology Information Web site (NCBI; *http://www.ncbi.nlm.nih.gov*). The databases at NCBI are populated in a number of ways, including scheduled uploads from sequencing centers, curated collections of existing sequences, and deposits by researchers from around the world (see *UNIT 11.2*; Chang et al., 2016). Each database at NCBI contains either DNA or protein sequences, and by choosing the appropriate BLAST program, most databases can be searched with either a nucleotide or protein sequence. NCBI provides anyone with an Internet connection with the ability to search any of their databases free of charge. These features and others make NCBI the premier Web site for searching any publicly available sequence data, and the protocols in this unit demonstrate how to use the tools available there.

Choosing a BLAST Program

The many different sequence databases at NCBI can be searched using one of five different BLAST programs. These programs compare either nucleotide sequences or protein sequences using the scoring matrices described earlier. Some programs also translate either the query sequence or the sequences of a nucleotide database before performing protein alignments. This versatility allows a researcher to compare any sequence to any database, and a basic tool such as BLAST can be used creatively to answer many questions. However, different types of questions may require different searching strategies. For example, if you are trying to determine whether a new primate gene you sequenced is more similar to a chimpanzee or a human sequence, you should compare nucleotide sequences directly; many human and chimpanzee protein sequences are identical to one another but are encoded using different codons. Alternatively, searching for homologs of your gene in very distantly related organisms requires a protein sequence comparison. Not only can the identical amino acids be encoded by different codons, but, over time, the proteins in different species may have evolved to contain different amino acids. Nucleotide sequences that encode distantly related proteins may contain little trace of their common ancestry, and may not even show up in your search results.

In addition, because some amino acids have more than one codon (the genetic code is degenerate), the same protein sequence can be encoded by very different nucleotide sequences. Since selection on protein-coding genes acts primarily on the sequences of proteins, the nucleotide sequences encoding homologous proteins in different species may be quite different. This means that when searching for homologs to a protein coding sequence, it is usually better to use the protein sequence itself instead of the nucleotide sequence.

The five different BLAST programs are named in a manner that describes how the query and database sequences are handled during the search. The BLASTN, BLASTP, and BLASTX programs search either nucleotide (BLASTN) or protein (BLASTP, BLASTX) databases directly using a nucleic acid or a protein query sequence. The programs with a "T" prefix (TBLASTN, TBLASTX) can only be used to search nucleotide databases; however, the sequences in the chosen database will be translated to the corresponding protein sequences prior to searching. Similarly, programs that end in "X" (BLASTX, TBLASTX) translate the query sequence.

With these considerations in mind, some examples of how to use each program to find different types of sequences are listed below.

BLASTN: This type of BLAST is used to search for sequences similar to a nucleotide query in a nucleotide sequence database. If this nucleotide sequence codes for a protein and you can determine the protein sequence, you should use the protein sequence and BLASTP instead. This is because the genetic code is degenerate and different nucleotide sequences can code for the same protein. You can use this program to: (1) identify noncoding RNA genes (rRNA, tRNA, snRNA, etc.), (2) identify conserved promoter elements, (3) find cDNAs corresponding to a genome sequence, and (4) identify a polymorphism or mutation in a sequence.

BLASTX: If you have a nucleotide sequence that encodes a protein, but you do not know which reading frame encodes it, you can input your nucleotide sequence into BLASTX. This program will translate the sequence in the six possible reading frames and search for sequences similar to these sequences in a protein database. It is typically used to: (1) determine the function of the protein encoded by a cloned cDNA or genomic DNA fragment, (2) determine the locations of introns in a genomic DNA sequence, and (3) identify frame shift errors in a DNA sequence.

BLASTP: This type of BLAST is used to search for sequences similar to a protein query in a protein sequence database. It can be used to: (1) search putative proteins encoded by a genome for a specific homolog, and (2) identify domains or motifs in a protein.

TBLASTN: Sometimes a protein sequence of interest has not been annotated in a genome or other large nucleotide database. This is common in the early stages of a genome-sequencing project, or in an EST project. In these cases, you can use your protein sequence to search the nucleotide database. BLAST will translate the database in all six reading frames and search for sequences similar to your protein in all frames.

TBLASTX: In this type of search, BLAST will translate a nucleotide query in all six reading frames, and will translate all the sequences in a nucleotide database in all reading frames. It then searches all possible translations of the database using all possible translations of your query. This type of search is useful when you have a nucleotide sequence that you know codes for a protein and you want to search for similar sequences in an EST database of another genome, or when you want to search an unannotated nucleotide database with a low-quality nucleotide sequence.

Choosing a Database

NCBI allows you to search many databases using the BLAST programs listed above. Two of the largest and most useful of these are the nucleotide and protein "nr" (nonredundant) databases. These databases contain every nucleotide or protein sequence submitted to the three international sequence depositories (GenBank, EMBL, and DDBJ), and are often the first databases many people search with a new sequence. However, while the nr database may be the only database that contains the sequences you are searching for, it may also contain far too many closely related sequences for your BLAST search to return meaningful results. For example, a BLAST search can be used to identify conserved and variable amino acids in a protein. However, searching nr with a sequence from an organism with many well-studied relatives (*Drosophila*, *Saccharomyces*) may return 100 nearly identical sequences from these relatives, providing no new information. As the number of sequences in databases like nr grows larger, the choice of database therefore becomes even more important. Searching a large database like nr also takes longer than necessary and wastes computing power on community servers.

Because searching large databases can lead to problems such as these, limiting the size of your BLAST database is often an important strategy. NCBI has created a smaller, curated database called RefSeq in which most of the identical sequences that arise from sequencing errors and mutational analyses have been removed. Though this database lacks data from some of the more exotic species, the sequences present are well annotated and still cover a broad sampling of organisms. RefSeq is therefore particularly useful for identifying a function for a newly sequenced gene from any organism.

Another way to limit the size of the BLAST database searched is to use an Entrez term. Entrez is the search engine used to select entries in all of the different databases at NCBI. Each entry in the NCBI sequence databases is tagged with information such as its source species, sequence length, type of molecule sequenced, and more. Including an Entrez term in a BLAST search will ensure that the search returns only sequences that satisfy the search term. For example, performing a BLASTP search of the RefSeq database using the yeast Eno1p (enolase) protein sequence while including the Entrez term "Homo sapiens" yields all of the human enolase genes, related human genes, and a handful of splicing variants. A similar search of RefSeq without including the Entrez term returns the maximum number of sequences (100 is the default), only three of which belong to humans. Because of the popularity of limiting a database by organism, the BLAST Web page even has a separate search term box that will recognize and auto-complete taxonomic and common names of many organisms.

Sequences in four very important sequence collections are not included in the "nr" database: the EST (Expressed Sequence Tag), GSS (Genomic Survey Sequence), WGS (Whole Genome Shotgun), and Trace Archive databases. The EST and GSS databases contain sequences generated by high-throughput cDNA (EST) and genome fragment (GSS) sequencing projects. These two databases contain sequences from many obscure organisms that are poorly represented in other datasets. Many interesting genes that have not been analyzed in depth can be found in these sequence collections, making them prime resources for discovering entirely new types of genes. The WGS database contains data from partially completed genome projects. Genomes whose coding regions have not been determined, or are only partially assembled, are available for searching in this database until they are completed. It can take years between assembly of the individual shotgun sequencing reads and publication of a genome sequence. Placing the sequences in the WGS database while the project is ongoing allows access to these sequences during this time. The Trace Archive contains raw data from many completed and ongoing sequencing projects. Searching this database allows immediate access to sequence data before it is assembled into chromosomes. It also contains leftover sequences from genome projects that could not be assembled into completed chromosomes. Unlike the other databases, which can be chosen in the pull-down menu on the individual BLAST program pages, the Trace Archive must be accessed from *https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=NucleotidesBLAST_SPEC=Trace Archive* and can only be searched using BLASTN and other nucleotide-versus-nucleotide programs.

Some of the databases available for searching at NCBI are summarized below. A list of NCBI databases along with additional information on how to use them can be found at *ftp://ftp.ncbi.nlm.nih.gov/pub/factsheets/HowTo_BLASTGuide.pdf*, or by clicking the question mark button beside the database selection box.

Nucleotide databases

Nucleotide collection (nr/nt): Comprehensive database that includes all sequences found in the GenBank and RefSeq databases.

Reference RNA sequences (refseq_rna): Cloned cDNA entries from the Reference Sequence database (fully contained in nr).

Reference genomic sequences (refseq_genomic): Genomic entries from the Reference Sequence database (fully contained in nr).

Expressed sequence tags (est): High-throughput cDNA sequences collected from a single species or cell type. EST projects are an efficient way to identify the proteins and protein-coding sequences in an organism without the cost of sequencing noncoding DNA, and are often performed in lieu of complete genome sequencing for many exotic organisms. This database contains all EST sequences in GenBank.

Genomic survey sequences (gss): GSS projects strive to sequence a random portion of a genome. If the genome of an exotic organism is thought to be very gene dense or if the noncoding regions are of interest, a GSS project may be performed rather than an EST project.

Whole-genome shotgun contigs (wgs): Contains all the reads used to assemble genomes. It is a database with many low-quality sequences and it should only be used if you cannot find an ortholog of your gene in any other databases.

Protein databases

Non-redundant protein sequences (nr): Contains translations of all GenBank coding sequences, plus sequences from the RefSeq Proteins collection and PDB.

Reference proteins (refseq_protein): Protein sequences from the Reference Sequence database (fully contained in nr).

Protein Data Bank proteins (pdb): Sequences from the 3-D structure database Brookhaven Protein Data Bank (fully contained in nr). Very useful if you are interested in finding proteins similar to your query that have a three-dimensional structure available.

PROTOCOLS

Basic Protocol 1: Selecting a Sequence Using Entrez

Entrez is a search engine that searches for keywords associated with entries in multiple different databases at NCBI. These databases include the Genbank nucleotide and protein sequence databases, plus non-sequence databases such as PubMed, OMIM, and Taxonomy. To execute this protocol, you will need to have an idea of what sequence interests you.

1. Point your browser to the NCBI home page: *http://www.ncbi.nlm.nih.gov*. This page contains a siingle text box in which to enter one or more keywords. It also contains a pull-down menu to specify the database to be searched (Fig. 11.1.3).
2. Select either the Nucleotide or the Protein sequence database from the pull-down menu.

   Note that it is not necessary to select a database before searching. The default, All Databases, will return the number of hits found in each NCBI database. The Nucleotide and Protein databases provide the best combination of database size and keyword annotation.

3. Enter a search term in the text box. Entrez can be searched with standard Boolean terms, like `AND`, `OR`, and `NOT`. Example: `Saccharomyces[organism] AND formaldehyde dehydrogenase`.

Figure 11.1.3 Screenshot of the NCBI Web page (*http://www.ncbi.nlm.nih.gov/*) showing the databases available for an Entrez search.

4. Select the link next to the sequence you wish to download; this will take you to the sequence's entry page.
5. Select the format of the sequence (see Support Protocols 2 and 3).
6. Copy the sequence to the clipboard or save the sequence as a file.

   These options can be selected from the pull-down menu near the top of the page. Alternatively, you can select the sequence or a portion of the sequence manually and choose Copy from your browser's Edit menu.

Basic Protocol 2: Search a Nucleotide Database Using a Nucleotide Query: Nucleotide BLAST (BLASTN)

The first BLAST search protocol will enable you to find sequences similar to a nucleotide query sequence in a nucleotide database. If this nucleotide sequence codes for a protein and you can determine the protein sequence, you should consider using the protein sequence and BLASTP instead (Support Protocol 1). This is because the genetic code is degenerate and different nucleotide sequences can code for the same protein.

NOTE: To execute this protocol, you will need a nucleotide sequence in FASTA (see Support Protocol 2) or GenBank format (see Support Protocol 3), or the accession number of your sequence of interest.

1. Point your browser to the BLAST Web page at NCBI: *http://blast.ncbi.nlm.nih.gov/Blast.cgi*. You will see a screen like that shown in Figure 11.1.4.
2. Choose nucleotide BLAST. You will see a page like that shown in Figure 11.1.5.

Figure 11.1.4 Screenshot of the BLAST program selection homepage.

3. Paste your nucleotide sequence into the text box in one of the following formats:

   FASTA: Copy and paste a FASTA-formatted sequence (Basic Protocol 1), including the description line started with >. Alternatively, you can omit the description line, pasting only your sequence. If a definition line is included, this will be used to title your search.

   GenBank: Copy and paste the sequence area of a GenBank entry of interest (Support Protocol 3). BLAST will remove the numbers and spaces and submit only the sequence. Make sure to only paste the sequence part of a GenBank entry; otherwise, your search will fail.

   GI or Accession number: To BLAST a sequence already present in GenBank, paste the GI (GenInfo Identifier) or Genbank Accession number into the search box and BLAST will retrieve the query sequence for you.

   Alternatively, you can upload a file containing the sequence in FASTA format.

4. If desired, limit the region of the sequence used to query the database by entering "From" and "To" values into the Query Subrange boxes.

   For an example of using the Query subrange option, see A Practical Example.

5. Choose the database to be queried.

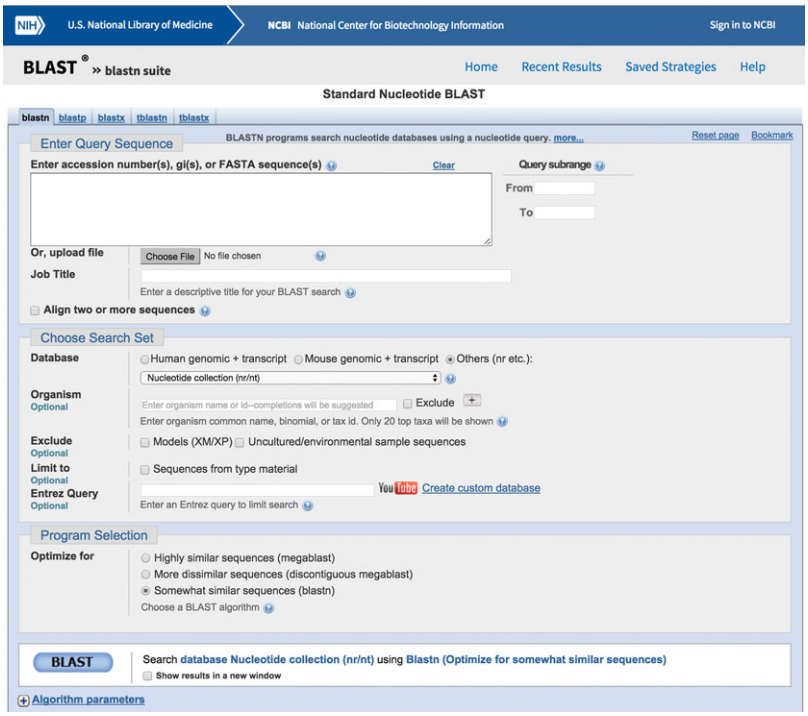

Figure 11.1.5 Screenshot of the nucleotide BLAST form.

BLAST allows you to choose from many databases. As the number of sequences in the databases grows larger, the choice of database becomes even more important; depending on the database that you use, your results may be dominated by a heavily studied gene and its variants (see Strategic Planning).

Two databases are readily available if you are interested in searching for your genes in humans or mouse: Human genomic + transcript and Mouse genomic + transcript. These databases are restricted to human or mouse sequences, respectively. Unless you are interested only in human or mouse results, you will probably need to use one of the other databases.

Information about each of the databases is available by clicking the question mark button beside the drop-down box and following the link there. Depending on the database you choose, another text box may appear that allows you to restrict the BLAST search to sequences of a given organism. This text box is dynamic and automatically completes the organism name as you are typing.

Another option is to use an Entrez query to further limit the database searched. See Basic Protocol 1 on how to use Entrez.

6. Select the type of BLAST search to be performed.

   For most applications, you should use the "Somewhat similar sequences (blastn)" option. We will not discuss the other nucleotide BLAST types in this unit. If you click the "?" button, you can access information about all of the available options. A good guide to determine which BLAST type is appropriate for you can be accessed at ftp://ftp.ncbi.nlm.nih.gov/pub/factsheets/HowTo_BLASTGuide.pdf. Also, see Ladunga (2009).

7. *If desired, set advanced search parameters*: Most of the time, you will perform BLAST using the default parameters; however, sometimes you can improve your search results by modifying these parameters. You can set advanced options by clicking in the arrow to the left of "Algorithm parameters" and change the following parameters as desired (Fig. 11.1.6):

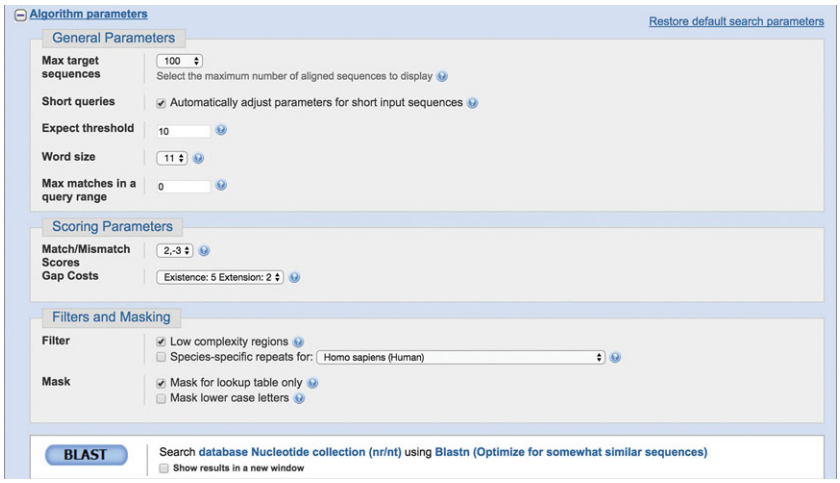

Figure 11.1.6 Example screenshot of the advanced options available under "Algorithm parameters" for a BLASTN search. The page for the other programs is slightly different, but the parameters discussed in this unit are the same for all BLAST programs.

Max target sequences: Because the databases are so large, it is common that the sequence you are interested in is not one of the 100 best hits (the default number of BLAST hits returned) and thus will not be reported by BLAST. This very useful option allows you to retrieve many more hits to your query (up to 5000; see A Practical Example).

Short queries: Very short query sequences usually do not return significant hits, since random matches easily happen by chance. This checkbox is marked by default; leaving the checkbox marked tells BLAST to adjust its parameters to try to find hits to a short sequence.

Expect threshold: BLAST will only return hits with an E-value smaller than the expect threshold. By default, this number is set to 10. Increasing this value will deliver a greater number of less significant hits. Decreasing this value will bring fewer, more significant hits.

The number of hits returned by BLAST is determined by both the Expect threshold and the Max target sequences. BLAST will return a maximum of Max target sequences. If there are less than Max target sequences hits with E-values smaller than the Expect threshold, BLAST will only return the hits with an E-value smaller than the Expect threshold.

All the other options affect parameters of the BLAST search algorithms, and they will not be covered in detail in this unit. For a review of these parameters, see Chapter 3 of Current Protocols in Bioinformatics (Bateman et al., 1992), or see Korf et al. (2003).

8. Click on BLAST to run your search.

9. Wait until the BLAST server completes the search and displays a results page similar to that shown in Figure 11.1.7.

The BLAST search may take several minutes. Once the search is finished, you can return to this results page instantly by entering the Request ID (RID) assigned to your search that is listed near the top of the results page.

Support Protocol 1: Search a Protein Database Using a Protein Query: Protein BLAST (BLASTP)

This BLAST search protocol will enable you to find sequences similar to a protein query in a protein database.

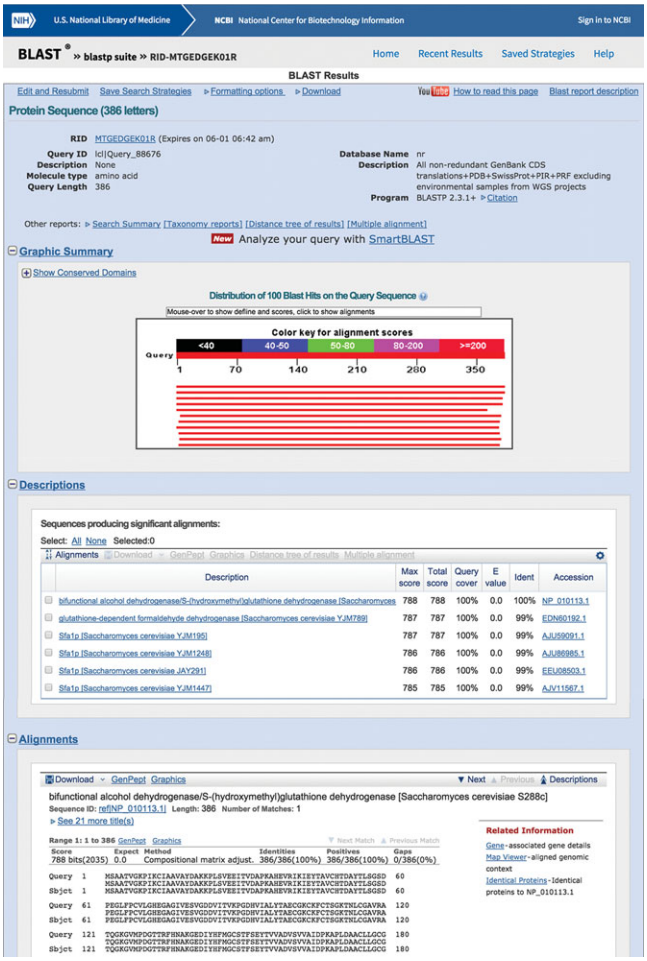

Figure 11.1.7 Example screenshot of a BLAST results page.

NOTE: To execute this protocol, you will need a protein sequence in FASTA (see Support Protocol 2) or GenBank format (see Support Protocol 3), or the accession number of your sequence of interest.

1. Point your browser to the BLAST Web page at NCBI: *http://blast.ncbi.nlm.nih.gov/Blast.cgi*. You will see a screen like that shown in Figure. 11.1.4
2. Choose protein BLAST. You will see a page similar to that shown in Figure 11.1.8.
3. Paste your protein sequence into the textbox.

   You can paste your sequence in a variety of formats. Consult step 3 of Basic Protocol 2 and Support Protocols 2 and 3 for sequence formatting options.

4. If desired, limit the region of the sequence used to query the database by entering "From" and "To" values into the Query Subrange boxes.

   For an example of using the "Query subrange" option, see A Practical Example.

5. Choose the database to be queried.

   BLAST allows you to choose many databases (see Strategic Planning). You can use the textbox below the database drop-down box to restrict your search to sequences of a given organism, or use an Entrez query to further limit the database searched.

6. Select the type of BLAST search to be performed.

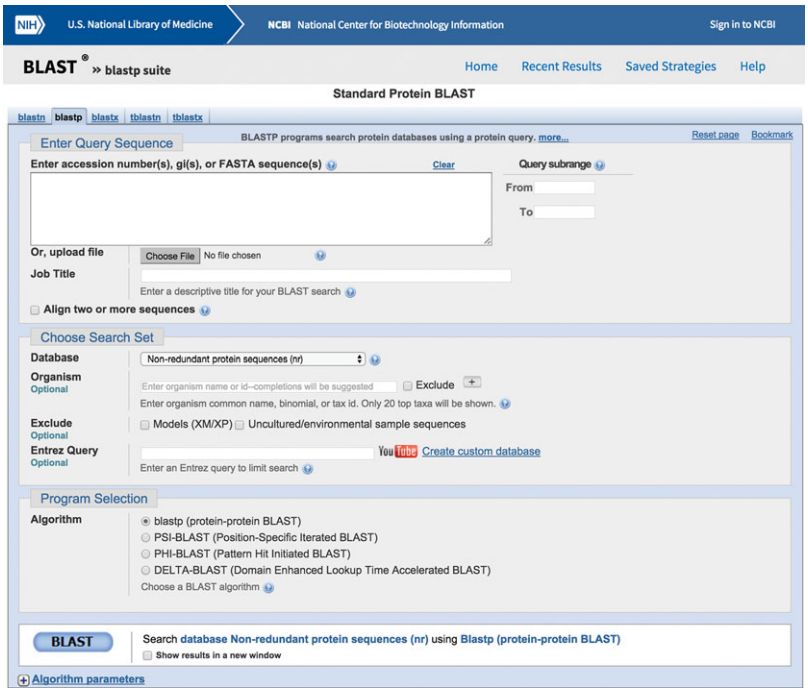

Figure 11.1.8 Screenshot of the protein BLAST form.

For simple applications, you should use the "BLASTp (protein-protein BLAST)" option. We will not discuss the other protein BLAST types in this unit. If you click the "?" button, you can access information about all the available options. A good guide to determine which BLAST type is appropriate for you can be accessed at ftp://ftp.ncbi.nlm.nih.gov/pub/factsheets/HowTo_BLASTGuide.pdf. The other protein BLAST programs, especially PSI-BLAST, give lots of extra power for your searches if you are not able to find the hits you expected.

7. Click on BLAST to run your search using the default parameters. These defaults work well for most searches and usually do not need to be changed.

   Alternatively, you can set advanced options by clicking on the arrow to the left of "Algorithm parameters." The parameters you can choose are very similar to those in step 7 of Basic Protocol 2, and we refer you there for an explanation of these options.

8. Click on BLAST to run your search.

9. Wait until the BLAST server completes the search and displays a results page similar to that shown in Figure 11.1.7.

   The BLAST search may take several minutes. Once the search is finished, you can return to this results page instantly by entering the Request ID (RID) assigned to your search that is listed near the top of the results page.

Basic Protocol 3: Search a Protein Database Using a Translated Nucleotide Query: BLASTX

If you have a nucleotide sequence that encodes a protein, but you do not know which reading frame encodes it, you can input your nucleotide sequence into BLASTX. This program will translate the sequence in the six possible reading frames and search for sequences similar to these sequences in a protein database.

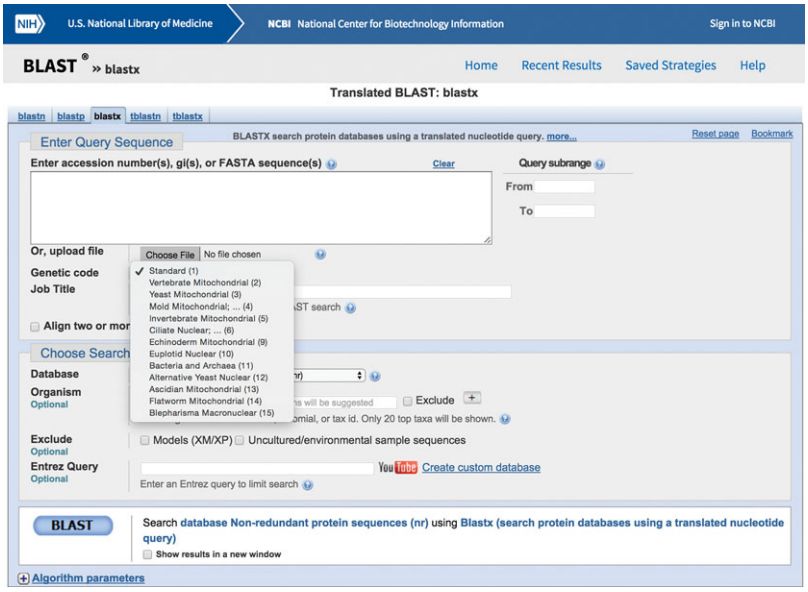

Figure 11.1.9 Screenshot of the BLASTX form, showing the genetic codes available for the query sequence.

1. Point your browser to the BLAST Web page at NCBI: *http://blast.ncbi.nlm.nih.gov/Blast.cgi*. You will see a screen like that shown in Figure 11.1.4.
2. Choose blastx. You will see a page like that shown in Figure 11.1.9.
3. Paste your nucleotide sequence into the textbox.

   You can paste your sequence in a variety of formats. Consult step 3 of Basic Protocol 2 for sequence formatting options.

4. If desired, limit the region of the sequence used to query the database by entering "From" and "To" values into the Query Subrange boxes.

   For an example of using the "Query subrange" option, see A Practical Example.

5. Select a Genetic Code to translate your nucleotide query sequence.

   Before performing a translated search, BLAST will translate your sequence in all six reading frames using the genetic code selected here. By default, this is the standard genetic code. Alternatives to the standard genetic code are used by many mitochondria and some organisms, especially ciliated protozoans. If you know that your query sequence is translated using an alternative code, select the appropriate genetic code in this box before performing your search.

6. Choose the database to be queried.

   BLAST allows you to choose many databases (see Strategic Planning). You can use the textbox below the database drop-down box to restrict your search to sequences of a given organism, or use an Entrez query to further limit the database searched.

7. Click on BLAST to run your search using the default parameters. These defaults work well for most searches and usually do not need to be changed.

   Alternatively, you can set advanced options by clicking on the arrow to the left of "Algorithm parameters." The parameters you can choose are very similar to those in step 7 of Basic Protocol 2, and we refer you there for an explanation of these options.

8. Wait until the BLAST server completes the search and displays a results page similar to that shown in Figure 11.1.7.

   The BLAST search may take several minutes. Once the search is finished, you can return to this results page instantly by entering the Request ID (RID) assigned to your search that is listed near the top of the results page.

Basic Protocol 4: Search a Translated Nucleotide Database Using a Protein Query: TBLASTN

Sometimes a protein sequence of interest has not been annotated in a genome or other large nucleotide database. This is common in the early stages of a genome-sequencing project, or in an EST project. In these cases, you can use your protein sequence to search the nucleotide database. BLAST will translate the database in all six reading frames and search for sequences similar to your protein in all frames.

NOTE: To execute this protocol, you will need a protein sequence in FASTA (see Basic Protocol 1) or GenBank format (see Basic Protocol 2), or the accession number of your sequence of interest.

1. Point your browser to the BLAST Web page at NCBI: *http://blast.ncbi.nlm.nih.gov/Blast.cgi*. You will see a screen like that shown in Figure 11.1.4.
2. Choose tblastn. You will see a page like that shown in Figure 11.1.10.
3. Paste your protein sequence into the textbox.

   You can paste your sequence in a variety of formats. Consult step 3 of Basic Protocol 2 for sequence formatting options.

4. If desired, limit the region of the sequence used to query the database by entering "From" and "To" values into the Query Subrange boxes.

   For an example of using the Query subrange option, see A Practical Example.

   IMPORTANT NOTE: *It is not necessary to specify a genetic code when using TBLASTN. The genetic codes needed to translate the sequences in the database have already been*

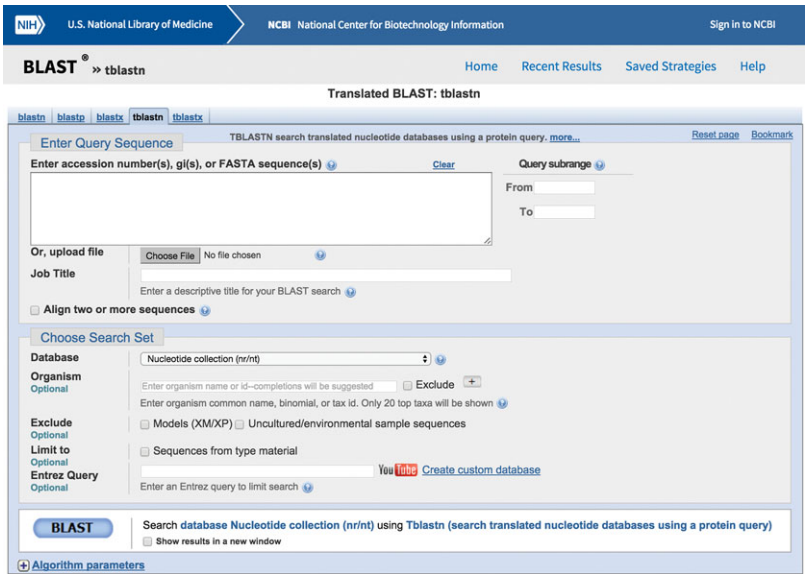

Figure 11.1.10 Screenshot of the TBLASTN form.

identified for each sequence. BLAST will automatically use the proper genetic code to perform the translation.

5. Choose the database to be queried.

   BLAST allows you to choose many databases (see Strategic Planning). You can use the textbox below the database drop-down box to restrict your search to sequences of a given organism, or use an Entrez query to further limit the database searched.

6. Click on BLAST to run your search using the default parameters. These defaults work well for most searches and usually do not need to be changed.

   Alternatively, you can set advanced options by clicking in the arrow to the left of "Algorithm parameters." The parameters you can choose are very similar to those in step 7 of Basic Protocol 2, and we refer you there for an explanation of these options.

7. Wait until the BLAST server completes the search and displays a results page similar to that shown in Figure 11.1.7.

   The BLAST search may take several minutes. Once the search is finished, you can return to this results page instantly by entering the Request ID (RID) assigned to your search that is listed near the top of the results page. You are done performing your BLAST search.

Basic Protocol 5: Search a Translated Nucleotide Database Using a Translated Nucleotide Query: TBLASTX

In this type of search, BLAST will translate your nucleotide query in all six reading frames, and will translate all the sequences in a nucleotide database in all reading frames. It then searches all possible translations of the database you choose, using all possible translations of your query. This type of search is useful when you have a nucleotide sequence that you know codes for a protein in an organism, and want to search for similar sequences in an EST project of another genome. This type of search can also be useful if you still have no hits to your nucleotide query after using BLASTN (Basic Protocol 3). Because the same amino acid can be coded by different codons, two protein sequences that are similar can be encoded by fairly different nucleotide sequences, and thus the program might detect similarities in the protein sequences that are difficult to detect in the nucleotide sequences.

NOTE: To execute this protocol, you will need a nucleotide sequence in FASTA (see Basic Protocol 1) or GenBank format (see Basic Protocol 2), or the accession number of your sequence of interest.

1. Point your browser to the BLAST Web page at NCBI: *http://blast.ncbi.nlm.nih.gov/Blast.cgi*. You will see a screen like that shown in Figure 11.1.4.

2. Choose tblastx. You will see a page like that shown in Figure 11.1.11.

3. Paste your nucleotide sequence into the textbox.

   You can paste your sequence in a variety of formats. Consult step 3 of Basic Protocol 2 for sequence formatting options.

4. If desired, limit the region of the sequence used to query the database by entering "From" and "To" values into the Query Subrange boxes.

   For an example of using the "Query subrange" option, see A Practical Example.

5. Select a Genetic Code to translate your nucleotide query sequence.

   Before performing a translated search, BLAST will translate your sequence in all six reading frames using the genetic code selected here. By default, this is the standard genetic code. Alternatives to the standard genetic code are used by many mitochondria and some organisms, especially ciliated protozoans. If you know that your query sequence

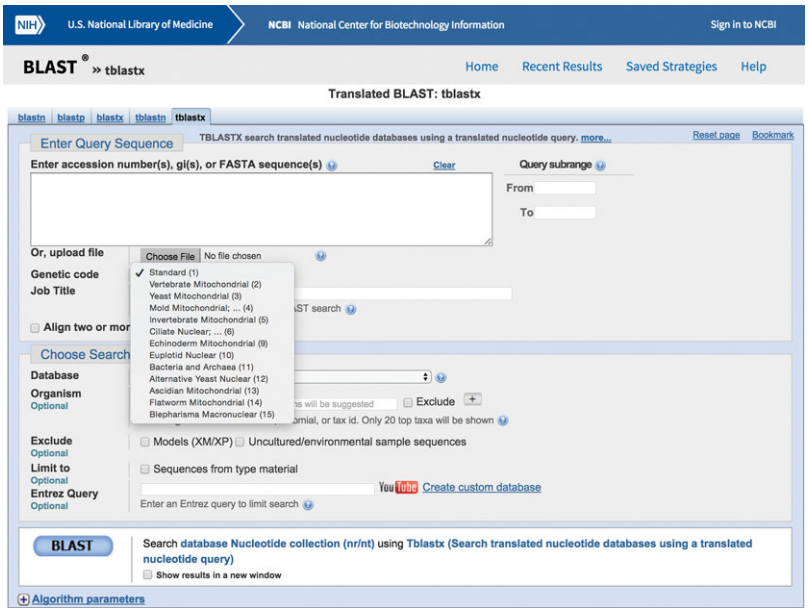

Figure 11.1.11 Screenshot of the TBLASTX form, showing the genetic codes available for the query sequence.

is translated using an alternative code, select the appropriate genetic code in this box before performing your search.

6. Choose the database to be queried.

   BLAST allows you to choose many databases (see Strategic Planning). You can use the textbox below the database drop-down box to restrict your search to sequences of a given organism, or use an Entrez query to further limit the database searched.

7. Click on BLAST to run your search using the default parameters. These defaults work well for most searches and usually do not need to be changed.

   Alternatively, you can set advanced options by clicking in the arrow to the left of "Algorithm parameters." The parameters you can choose are very similar to those in step 7 of Basic Protocol 2, and we refer you there for an explanation of these options.

8. Wait until the BLAST server completes the search and displays a results page similar to that shown in Figure 11.1.7.

   The BLAST search may take several minutes. Once the search is finished, you can return to this results page instantly by entering the Request ID (RID) assigned to your search, which is listed near the top of the results page. You are done performing your BLAST search.

Support Protocol 2: Preparing a Sequence in FASTA Format

The FASTA format (Fig. 11.1.12) is perhaps the most frequently used sequence format in bioinformatics, and it is one of the simplest. Because of its simplicity, it is supported by almost all bioinformatics software. This versatility comes with a price, however; a FASTA file contains very little information about the sequence besides the sequence itself.

A FASTA file may contain one or multiple sequences. Each sequence begins with a description, which is a single line that starts with a > ("greater than" character) and can contain any description of the sequence. The end of the description must be indicated by a newline character. The nucleotide or protein sequence that follows is written in standard IUPAC one-letter notation for nucleotides or amino acids, and it may contain additional

```
>Sequence 1
MTSATAGKPIECVAAVAYEAGKPLTVEKIIVDAPKAHEVRVQVTHTAVCH
TDAYTLSGVDPEGAFPSILGHEGAGIVESVGDGVTNVKVGDHVVLLYTAE
CGKCKFCKSNKTNLCGSVRATQGKGVMPDGTTRFHNLKGEPLLHFMGCST
FSQYTVVADVSLVTIDPSAPLSSVCLLGCGVTTGYGAAVKTANVQEGDTV
AVFGAGTVGLSVVQGAKSRGASKIIVVDVNDQKKQWSMDFGATGFVNPLK
DLKEGETIVSKLIDMTDGGLDFTFDCTGNVKVMRDALEACHKGWGQSIII
GVAAAGEEISTRPFQLITGRVWKGSAFGGIKGRSEMGQLVTSYQKGDLKV
DDFITHKRPFTEIN---------NAFEDLHHGDCLRTVLDLAN--
>Sequence 2
MSAATVGKPIKCIAAVAYDAKKPLSVEEITVDAPKAHEVRIKIEYTAVCH
TDAYTLSGSDPEGLFPCVLGHEGAGIVESVGDDVITVKPGDHVIALYTAE
CGKCKFCTSGKTNLCGAVRATQGKGVMPDGTTRFHNAKGEDIYHFMGCST
FSEYTVVADVSVVAIDPKAPLDAACLLGCGVTTGFGAALKTANVQKGDTV
AVFGCGTVGLSVIQGAKLRGASKIIAIDINNKKKQYCSQFGATDFVNPKE
DLAKDQTIVEKLIEMTDGGLDFTFDCTGNTKIMRDALEACHKGWGQSIII
GVAAAGEEISTRPFQLVTGRVWKGSAFGGIKGRSEMGGLIKDYQKGALKV
EEFITHRRPFKEIN---------QAFEDLHNGDCLRTVLKSDEIK
>Sequence 3
-MASTVGKTITCKAAIAWGAGQELSYEDVEVAPPKAHEVRIQIKHTGVCH
TDAYTLSGKDPEGAFPVILGHEGAGIVESVGEGVTNVKPGDHVIALYTPE
CKECKFCKSGKTNLCGKIRATQGRGVMPDGTSRFR-ARGQDILHFMGTST
FSQYTVVADISVVAVNPEAPMDRTCLLGCGITTGYGAATITANVEKGSTV
AIFGAGCVGLSVIQGAVANGASKIIAVDVNPSKEEWSRKFGATDFVNP-S
TLPEGQSVVDKLIELTDGGCDYTFDCTGNVKVMRAALEACHKGWGQSIII
GVAAAGQEISTRPFMLVTGRVWRGSAFGGVKGRSQLPGLVEDYLNGKIKV
DELITHRKKLAEIN---------NAFEVMHQGDCVRAVVDMS---
```

Figure 11.1.12 A FASTA file containing the alignment of three protein sequences. Note that some sequences have hyphens (-) to indicate gaps. Each sequence begins with a line started by a greater than symbol (>).

newline characters to make the sequence easier to read. Gaps are represented by a hyphen (-), and an asterisk (*) in a protein sequence indicates translation termination. The end of the file signifies the end of the sequences. If multiple sequences are stored in the same file, the next sequence begins at the next line started by a > character.

To create a FASTA file, you can use any text editor; however, make sure to save your file as plain text, as the formatting characters used by many word processors will make the file unreadable by bioinformatics software programs. In Windows, it is preferable that you use Notepad to create your FASTA files, as it is a plain text editor. With the Macintosh, you can use TextEdit, but make sure to click on the Make Plain Text option in the Format menu; this will guarantee that TextEdit will save your file as plain text.

Support Protocol 3: Formatting a Sequence in GenBank/GenPept

These formats are the default for viewing the entry pages for nucleotide and protein sequences in GenBank, respectively. These formats list many useful facts related to the sequence, including its length, identifiers, species of origin, submission author, links to corresponding protein/nucleotide sequences, and papers published about the sequence. The sequence itself is found at the bottom of the page and is formatted with a space between every tenth residue, sixty residues per line; each sequence line starts with numbers representing the current position in the sequence. The end of an entry is signified by a line containing only the // characters. A single file can contain multiple sequences separated by // (Fig. 11.1.13). Note that this sequence can be copied directly from the page and submitted to a BLAST server; the numbers and spaces will be stripped out by the program. A full description of these formats is beyond the scope of this unit, but you can get a full description of the formats at the NCBI site at *http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html*, or in Leonard et al. (2006).

```
LOCUS       AY639878               2227 bp    DNA     linear   INV 15-JUN-2005
DEFINITION  Sterkiella histriomuscorum formaldehyde
            dehydrogenase/S-formylglutathione hydrolase fusion protein (FSF1)
            gene, nanochromosome complete sequence; macronuclear.
ACCESSION   AY639878
VERSION     AY639878.1  GI:55669336
KEYWORDS    .
SOURCE      Sterkiella histriomuscorum (Oxytricha trifallax)
  ORGANISM  Sterkiella histriomuscorum
            Eukaryota; Alveolata; Ciliophora; Intramacronucleata; Spirotrichea;
            Stichotrichia; Stichotrichida; Oxytrichidae; Sterkiella.
REFERENCE   1  (bases 1 to 2227)
  AUTHORS   Stover,N.A., Cavalcanti,A.R., Li,A.J., Richardson,B.C. and
            Landweber,L.F.
  TITLE     Reciprocal fusions of two genes in the formaldehyde detoxification
            pathway in ciliates and diatoms
  JOURNAL   Mol. Biol. Evol. 22 (7), 1539-1542 (2005)
   PUBMED   15858209
REFERENCE   2  (bases 1 to 2227)
  AUTHORS   Stover,N.A., Cavalcanti,A.R.O., Li,A.J., Richardson,B.C. and
            Landweber,L.F.
  TITLE     Direct Submission
  JOURNAL   Submitted (27-MAY-2004) Ecology and Evolutionary Biology, Princeton
            University, Guyot Hall, Princeton, NJ 08544-1003, USA
FEATURES             Location/Qualifiers
     source          1..2227
                     /organism="Sterkiella histriomuscorum"
                     /macronuclear
                     /mol_type="genomic DNA"
                     /db_xref="taxon:94289"
                     /note="nanochromosome
                     synonym: Oxytricha trifallax"
     misc_feature    1..28
                     /note="telomere"
     gene            117..2132
                     /gene="FSF1"
     CDS             117..2132
                     /gene="FSF1"
                     /note="Fsf1p"
                     /codon_start=1
                     /transl_table=6
                     /product="formaldehyde dehydrogenase/S-formylglutathione
                     hydrolase fusion protein"
                     /protein_id="AAV54596.1"
                     /db_xref="GI:55669337"
                     /translation="MESQGTQGQVIRCKAAVAWEANKPLDICEIEVAPPQKGEIRVRV
                     VANALCHTDIYTLDGHDPEGLFPCILGHEATAIVESIGEGVTSVKVGDTVIPCYTPQC
                     MERDCVFCMSQKTNLCPKIRATQGKGVMPDGTTRFSKDGKPIYHFMGCSTFSEYTVIA
                     EISAAKINPTADLNKVCMLGCGVSTGWGAAMVNPEVKPGTAVAVWGLGAVGLAVIQAA
                     KLQGAGKIYGFDVNHDKFDHAKKLGADECFNPMESDSKDWLLQREKWGVHYTYDCTGN
                     VAVMRTALEAAHRGYGESCIIGVAAAGKEISTRPFQLVTGRQWKGTAFGGWKSREDVP
                     KLVNKVVVGELNVDDFITHYFDGLDQVNESIDILHSGKCLRAVVKISSTDVQESHNVK
                     VLQSQKYQGGVLKTVQHWSNVNNCEMKFMIFLPNETIKEQRGKAYPALYFLSGLTATH
                     ENAAIKSHFGAFAKKHNIAMIFPDTSPRGVEIEGIKENWWFGESAGYYLNATEGKWSK
                     NFNMYSYINEELPQVVERHFHVDGSRKSITGLSMGGMGALQIYLKNSEKYRSVSAFSP
                     IANPSECQWGQDAFNGFLGSVEAGAQYDPTLLVKDFQGRKTPILIDQGSHDKFLKDLL
                     PENFLKAADQSGVEVEYTMRDGYGHDFFFVSTFIENHIDFHARYLKA"
     misc_feature    2208..2227
                     /note="telomere"
```

Figure 11.1.13 A GenBank-formatted flat file containing two sequences. Note that this format gives much more information about the sequence than the FASTA format. Each entry in a GenBank formatted file ends in a line containing only `//`. Multiple entries can be present in a single file; if that is the case, entries are separated by `//`.

```
ORIGIN
        1 ccccaaaacc ccaaaacccc aaaaccccat ggataagcat tttaagtgat gcttgatttg
       61 agaatcttaa taaataggaa taattaatta caaatatctt ttttaatcaa gtcacaatgg
      121 aaagtcaagg tactcaaggc caagtcatta gatgcaaagc tgcagttgca tgggaagcaa
      181 ataaaccatt agatatctgc gagattgaag ttgctccacc ataaaaaggt gaaatcagag
      241 tgagagttgt tgctaatgca ctatgtcata cagatattta tactttagat ggtcatgatc
      301 cagaaggatt gttcccatgt attcttggac acgaagctac agctattgtc gagagtattg
      361 gtgagggagt cacttcagtc aaagttggag atactgttat cccttgctat actcctcaat
      421 gtatggagag agattgtgtc ttctgcatga gttaaaagac aaatctttgc cccaagatcc
      481 gtgctactca aggaaaagga gtgatgccag atggaacaac cagattctca aaggatggaa
      541 aacccattta tcattttatg ggatgctcta ctttcagtga atacacagtt attgctgaaa
      601 tctcagctgc taaaatcaat ccaactgccg atcttaacaa ggtttgcatg ctaggttgtg
      661 gagttagcac tggatgggga gctgcaatgg taaacccaga agttaaacca ggcacagctg
      721 ttgctgtttg gggtttagga gctgtaggtc ttgctgtaat ctaagctgct aagcttcaag
      781 gtgctggaaa gatctatgga tttgatgtta atcatgataa atttgatcac gcaaagaagc
      841 taggtgccga tgagtgcttt aaccctatgg aatcagattc taaggattgg ttactctaaa
      901 gagaaaagtg gggtgttcat tatacctacg attgcacagg aaatgttgct gtcatgagaa
      961 ctgctctaga agctgctcat agaggatatg gggaaagctg tatcattggt gttgctgctg
     1021 caggtaaaga aatctcaact agacctttct aacttgttac tggaagatag tggaaaggaa
     1081 cagcattcgg aggatggaag agtagagaag atgtcccaaa gcttgtcaat aaagttgttg
     1141 taggagaatt gaatgttgat gatttcatta ctcattattt cgatggtctt gactaagtaa
     1201 acgaatcaat tgatatcctt cactctggca aatgtctaag agctgtagtc aaaatctcat
     1261 caacagatgt ttaagagtct cataatgtca aagttttgta aagctagaaa tatcaaggag
     1321 gtgtactaaa gactgtacaa cattggagta atgtaaacaa ctgtgaaatg aaattcatga
     1381 ttttccttcc aaatgaaaca attaaagaac aaagaggtaa ggcataccca gctctatatt
     1441 tcttatctgg acttactgca actcacgaaa atgcagcaat caagagtcac tttggtgcat
     1501 ttgccaagaa acacaatatt gctatgattt tcccagatac ctccccaaga ggtgttgaaa
     1561 ttgaaggcat aaaagaaaat tggtggtttg gtgagagtgc tggctactat ttaaatgcta
     1621 ctgaaggaaa atggtcaaag aacttcaata tgtatagcta tattaatgag gagttacctt
     1681 aagttgttga aagacacttc catgttgatg gatccagaaa atcaattact ggtttgagta
     1741 tgggaggaat gggtgctctt caaatctact tgaagaattc agaaaaatat agatcagttt
     1801 ctgctttcag ccctatcgct aaccctagtg agtgttaatg ggggtaagat gctttcaatg
     1861 gattccttgg ttcagttgaa gctggagcac aatacgatcc aactttgtta gtcaaggact
     1921 tctaaggccg caagacacct attctcattg attaaggttc tcacgataaa ttcttgaaag
     1981 atctcttgcc agagaacttc ttgaaggctg cagatcaatc tggagttgaa gttgagtaca
     2041 ctatgagaga tggttatggt catgacttct tctttgtctc tactttcatt gaaaatcaca
     2101 ttgatttcca tgctagatat cttaaggcat gaattagaaa atgagaaatc tttaaatgct
     2161 tgtttattta ttttatttcg atatcattct ttctttgaga ttattaaggg gttttggggt
     2221 tttgggg
//
```

Figure 11.1.13 *Continued.*

You will rarely need to create files in this format, except when you submit your sequences to GenBank. In this case, you will use the program Sequin to generate GenBank-formatted files (see *UNIT 11.2*; Chang et al., 2016).

COMMENTARY

Understanding Results

The first step in a BLAST search should be to copy the query sequence from a file on your computer, or to choose a query sequence from a database as described in Basic Protocol 1. In this first example, we will describe how to use the formaldehyde dehydrogenase protein from yeast (*SFA1*) to query a protein database. We have used this exact search in a research setting to identify formaldehyde dehydrogenase enzymes in a variety of organisms.

The accession number of the *Saccharomyces cerevisiae* (yeast) formaldehyde dehydrogenase enzyme is NP_010113. Searching for this accession number in the Entrez protein database at NCBI leads to a results page containing a link to the GenBank entry for the protein. At the bottom of this page, you can copy the amino acid sequence of the protein. This sequence can be pasted directly into the textbox of the protein BLAST form (Fig. 11.1.8). Alternatively, you can simply

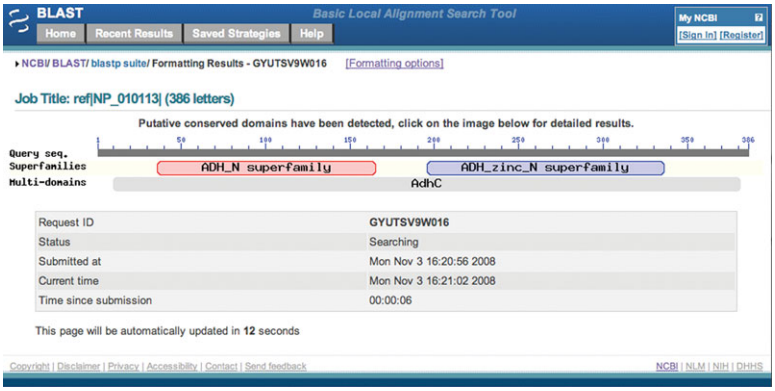

Figure 11.1.14 Screenshot of the temporary page that is loaded after you submit your BLAST search. This is a dynamic page that is automatically updated until your results are ready. In this example, our search was assigned the Request ID GYUTSV9W016. The page also shows the results of a conserved domain search.

Figure 11.1.15 Screenshot of the header area of a BLAST result page. At the top are links to change how the results are displayed and to download the results. On the left, BLAST shows the ID and description of the query sequence (note that this will only be shown if you search using a GI or accession number). On the right, BLAST displays the database and program used in the search. At the bottom are links for the "Taxonomy reports" and the "Distance tree of results" pages.

enter the accession number NP_010113 in the sequence box. BLAST will recognize this number and use the protein sequence it refers to in the search.

Please note that the databases we discuss here are continuously growing and changing. Thus, your results will not be identical to the ones described here.

After the BLAST search is submitted using the default values, you will be directed to a temporary page that informs you of the progress of your search. This page shows the Request ID for your search, a unique identifier that can be used to retrieve the results up to 36 hr later. In the example shown in Figure 11.1.14, the Request ID is GYUTSV9W016. This page also shows the results of a Conserved Domain Database search, which will be discussed below.

This page will reload periodically until the search is completed, at which point the results page will be automatically loaded. If you prefer, you can retrieve your results at a later time by going to the Recent Results tab of the BLAST Web page, entering the Request ID in the text box, and clicking Go.

The BLAST results page

The BLAST results page is divided in four main areas: Header, Graphics Summary, Descriptions, and Alignments. The links at the top of the results page provide options to save and reformat the results (Fig. 11.1.15).

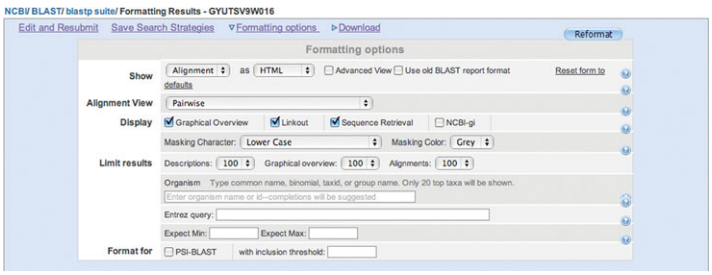

Figure 11.1.16 Screenshot of the formatting options in a BLAST result page.

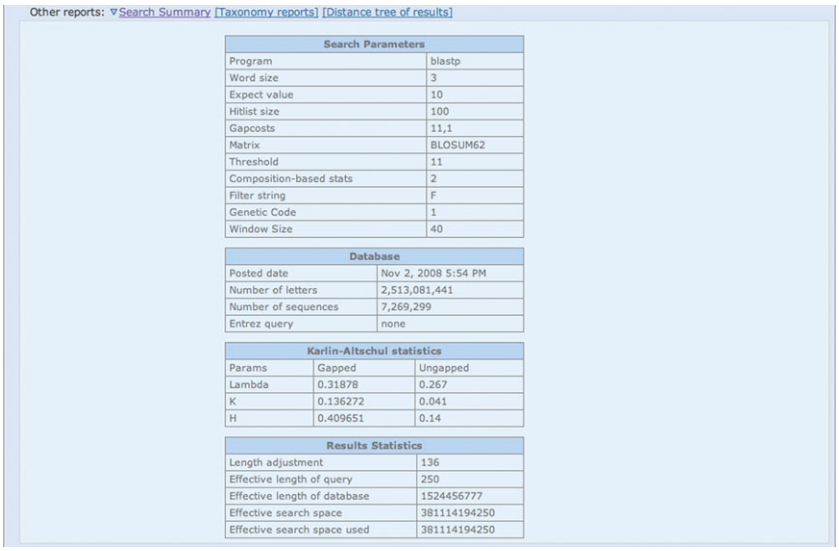

Figure 11.1.17 Screenshot of the Search Summary showing the different parameters used by BLAST in the search.

Clicking on the arrow to the left of Formatting options allows you to change the number of hits displayed in the graphical overview, determine how many description lines are shown, and determine how many alignments are displayed (see below). Since the complete results are all stored after each BLAST search, selecting which results to display does not require running a new search. Likewise, you can use other formatting options to limit your results to hits within a single organism or taxon, hits that fulfill a certain Entrez query, or hits within a certain E-value range (Fig. 11.1.16).

Note that the limit for the maximum number of hits that the BLAST search returns is set before the program is run (see step 7 of Basic Protocol 2). The Formatting options can be used only to limit how many of the hits resulting from the run are displayed. For example, if you ran BLAST with a Max target sequences (step 7 of Basic Protocol 2) of 100, this is the maximum number of hits that will be displayed regardless of what you enter in the Formatting options form.

If you click on the arrow to the left of **Download**, you can download your results. BLAST gives a variety of formats for download. Usually you will be interested in downloading the results as **Text**, which is just a text file with the results, or as a hit table, which consists of a table with information about all the hits with the values separated by commas [**Hit Table(csv)**], or tabs [**Hit Table(text)**]. The hit table formats are suitable for opening in a worksheet program like Microsoft Excel. The other formats are useful as input to other computer programs.

A summary of the parameters used in the search is shown by clicking the arrow beside **Search Summary** (Fig. 11.1.17). Some of these values have been discussed previously in this unit, but the others are beyond the scope of this introduction.

Clicking on **Taxonomy Reports** opens a new page showing the phylogenetic

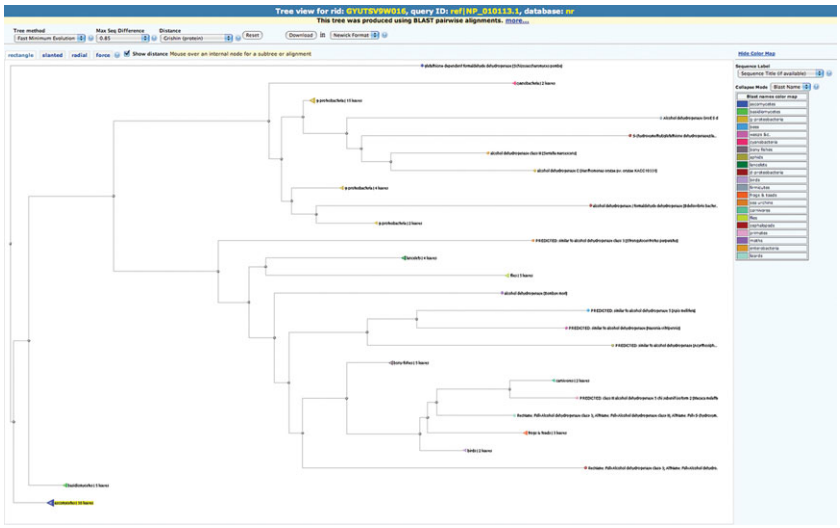

Figure 11.1.18 Screenshot of the "Distance tree of the results" showing a phylogenetic tree of the hits obtained in the BLAST search of the yeast formaldehyde dehydrogenase gene (SFA1) against the nr protein database.

distribution of your hits. **Distance tree of results** takes this one step further and shows a phylogenetic tree of the hits (Fig. 11.1.18). Please note that although this tree is useful for a quick overview, it is based only on the BLAST pairwise alignment scores rather than a multiple alignment. It is important to stress that superior methods of phylogenetic analysis exist and may return results significantly different from this tree. Simple phylogenetics methods are described in greater detail in *UNIT 11.3* (Zufall, 2017).

Graphics summary

The graphics summary has two main components. The first is a figure showing the conserved domains found in your sequence (Fig. 11.1.19). BLASTP searches always compare your sequence to the Conserved Domain Database (CDD) maintained by NCBI; you will be alerted if your sequence matches any of the consensus sequences found in this database. Conserved domains can be very useful when trying to determine the function of a protein. Proteins that share particular domains or combinations of domains often perform similar functions. If you click on the figure, you will be directed to a CDD Web page, where you can learn more about the domains that comprise your protein.

In our yeast formaldehyde dehydrogenase example, the enzyme contains two conserved domains, an alcohol dehydrogenase domain and a zinc-binding dehydrogenase domain. Visiting the CDD leads to more information about each of these domains and examples of other proteins with similar structures.

The second part of the graphics summary shows the location of the BLAST hits returned in your search. Each line corresponds to a hit and shows its location relative to the query sequence. The color of a bar represents the score of the hit. A key to the scores is shown above the thick red bar representing the query sequence. In this example, you can see many high-scoring hits along the entire length of the sequence (Fig. 11.1.19). If you mouse over a hit, its description is shown in the text box above the figure. If you click on a hit, you will be taken down the page to the alignment of the hit and query sequences.

Since formaldehyde dehydrogenase is a very important protein whose sequence is highly conserved in a wide range of species, our example BLASTP search returns many strong hits (with scores higher than 200) that extend for the entire length of the query sequence. The graphic results of a more interesting search are shown in Figure 11.1.20. Here we have performed a BLASTP search of the protein nr database using a receptor tyrosine kinase called HTK30, from the cnidarian *Hydra vulgaris* (accession number: AAC34124). This protein contains a highly conserved intracellular tyrosine kinase domain and an unusual extracellular region that contains a domain found only in one other tyrosine kinase.

The CDD search reveals a PTK (protein tyrosine kinase) domain as expected (Fig. 11.1.20). The alignment figure shows

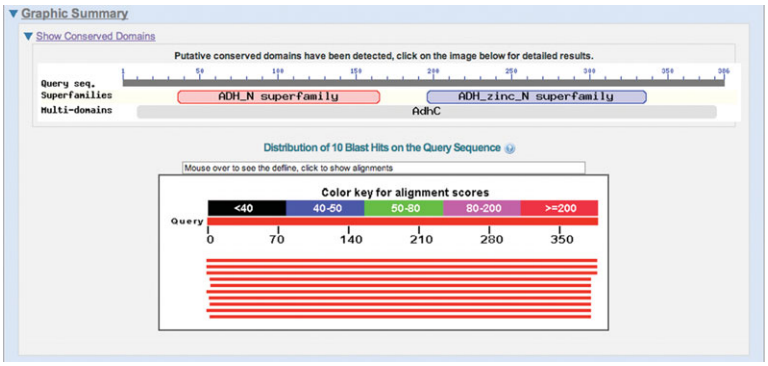

Figure 11.1.19 Screenshot of the Graphics Summary of the BLASTP search of the yeast formaldehyde dehydrogenase gene (SFA1; accession number: NP_010113) against the protein NR database.

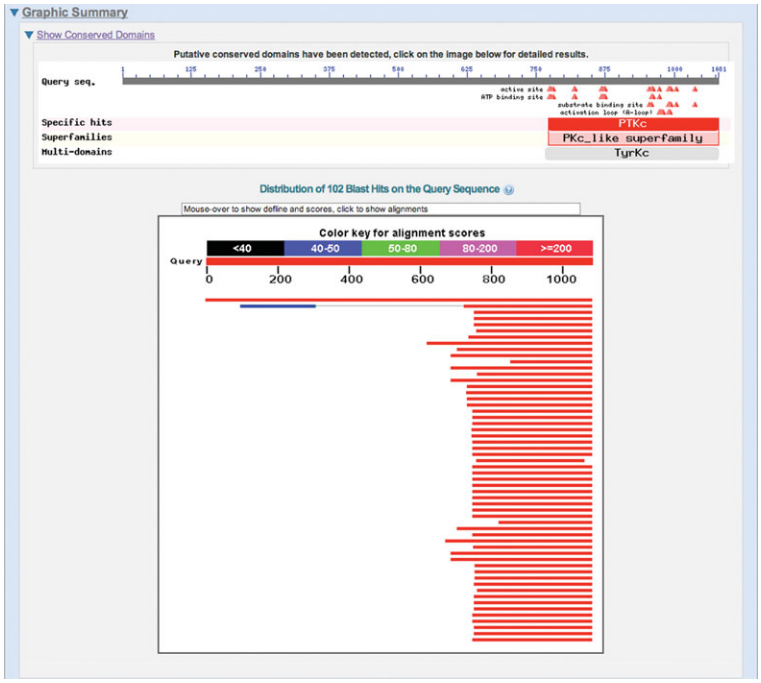

Figure 11.1.20 Screenshot of the Graphics Summary of the BLASTP search of the hydra tyrosine kinase HTK30 (accession number: AAC34124) against the protein nr database.

database hits below the red bar aligned to the query sequence. The stronger hits are shown closest to the query. In this case, there is one high-scoring database match that aligns to most of the query sequence. This is the HTK30 sequence, which has been deposited into the nr database. The next bar represents a lower-scoring match that aligns to two regions of the query, from about residues 100 to 300 and residues 700 to 1100. The line connecting these regions indicates that the two regions of similarity are on the same protein, but that this intervening region does not match. The remaining bars show lower-scoring alignments.

The Graphic Summary is very useful for a quick look at the distribution of the hits in the query sequence, but to get a better idea of the quality of the hits, it is important to check the Alignments section of the results page (see below).

Descriptions

The third section of the results page shows the hits to the query, listed from highest score to lowest. Each line shows the accession number and description line of the hit, plus its score and E-value. The accession number of the hit is hyperlinked to its GenBank page with more

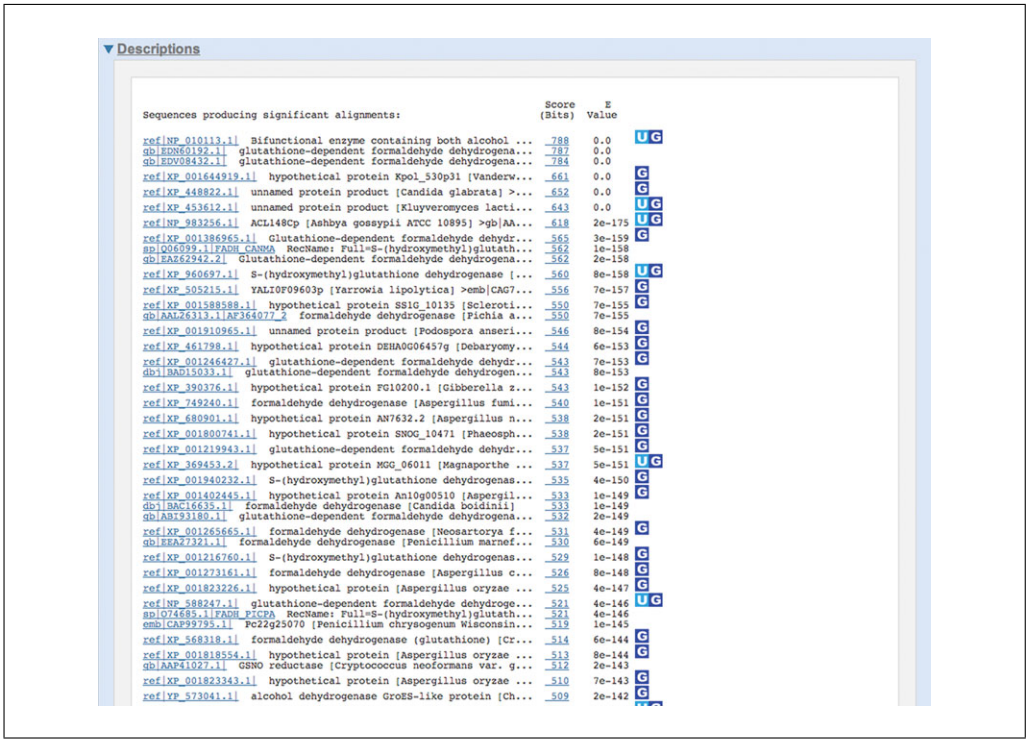

Figure 11.1.21 Screenshot of part of the Descriptions section of the BLASTP search of the yeast formaldehyde dehydrogenase gene (SFA1; accession number: NP_010113) against the protein NR database.

information about the sequence. The BLAST score for each is hyperlinked to the alignment of the hit and query farther down the page (Fig. 11.1.21).

Alignments

The last section of the BLAST results page shows the alignments of the query sequence against the hits listed in the Descriptions section. The default format for these alignments is a pairwise alignment in which each hit is shown aligned with the query separately. Using the **Alignment view** drop-down box in the **Formatting options** link from the Search Summary section (Fig. 11.1.16), you can choose different formats for the alignments, each more indicated for a given application.

The default **Pairwise** alignment (Fig. 11.1.22A) is indicated when you are interested in evaluating the similarity between the query and a target sequence, and thus is ideal for most searches. The alignment is given with the query sequence on top and the target sequence on the bottom. The middle line between query and target indicates if the residues in a given position are identical (using the amino acid code), if they are similar (by a +), or if they are dissimilar (by a space).

An alternative pairwise alignment that some find easier to interpret is the **Pairwise with dots for identities** option. Figure 11.1.22B shows the same alignment as Figure 11.1.22A using this alternative display. Only the query and the target sequence are shown using this option. Identical residues are represented by dots in the target sequence, and residues that differ are represented in red by the letters of their amino acid code, making them easy to spot against the background of dots.

Sometimes it is more interesting to see how individual residues of the query sequence are conserved among several sequences. The query-anchored formats are more appropriate in these cases. Figure 11.1.23A shows an example alignment using the **Flat query-anchored with dots for identities** format. The target sequences are all shown aligned in relation to the query. Identical residues are represented as dots, while mismatches are represented by the amino acid code. When a target sequence contains a residue missing in the query, BLAST will add a gap at that location in the query.

In contrast, the **Query-anchored with dots for identities** format (Fig. 11.1.23B) does not insert gaps into the query when aligning sequences of different lengths. When a target sequence contains an insertion relative to the query, the insertion point is indicated by a vertical bar (|) and a slash sign (/), and the

Figure 11.1.22 Screenshots of the alignment between the yeast formaldehyde dehydrogenase with the human alcohol dehydrogenase 5 (accession number: AAV38636.1) using different types of pairwise alignments from BLAST. (**A**) Pairwise. (**B**) Pairwise with dots for identity.

inserted amino acids are shown below. Figure 11.1.23B shows that several of the target sequences have the insertion of an alanine at a position corresponding to the 64th residue in the query.

Each of the query-anchored formats also has the option of using letters to show identities instead of dots. The dot formats are particularly useful for identifying similarities and differences between all the results in your BLAST search, and it is highly recommended that you use this format when comparing all of the sequences at once.

The query-anchored formats are very useful for studying allelic variations or to determine which residues are highly conserved in a family of proteins. The alignments shown in Figure 11.1.23 were made using BLASTP to compare the yeast formaldehyde dehydrogenase against the protein nr database and limiting the search to human sequences. As can be seen in Figure 11.1.23A, position 35 in the yeast sequence has a histidine, and several human sequences have a histidine in the same position, but some show a tyrosine and yet others show lysine. Position 26 has a valine in the yeast query sequence, while all human sequences shown have an isoleucine in the same place.

Finally, BLAST also allows you to display your results as a **Hit Table**. This format is only appropriate if you are going to use the output of your BLAST search as the input for other bioinformatics programs and that will not be covered here. Note that if you choose **Hit Table**, the output page does not show the descriptions or the graphical overview.

Troubleshooting

BLAST searches sometimes return surprising results. Some of the most common problems, with possible solutions or explanations, are shown in Table 11.1.1.

Unfortunately, not all problems with BLAST searches have simple, straightforward answers. As with any task, mistakes can be made by a person performing a search, a person submitting a sequence to a database, or

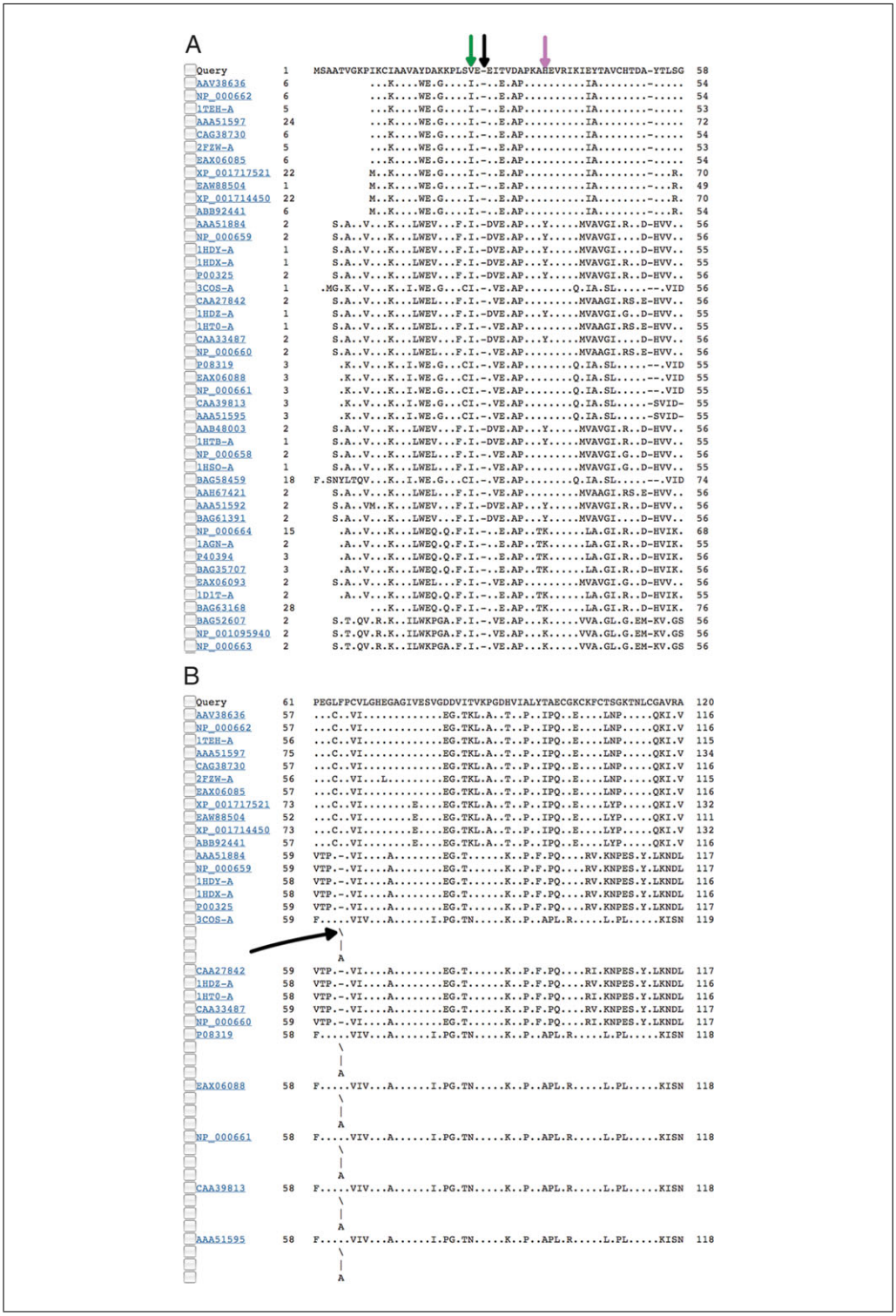

Figure 11.1.23 Screenshots of the query-anchored alignment between the yeast formaldehyde dehydrogenase with its hits in Homo sapiens. (**A**) Flat query-anchored with dots for identities. The green arrow points to position 26 in the yeast query sequence; while the yeast sequence has a V at this position, all the human sequences have an I. The pink arrow points to position 37 in the yeast query sequence; the yeast sequence has an H at this position; several human sequences also have an H (represented by dots as they match the residue in the query sequence) while other human hits have a Y or a K. Note the gap between residues 27 and 28 in the yeast query sequence (black arrow). (**B**) Query-anchored with dots for identity. Note that many human sequences have the insertion of an A between positions 64 and 65 in the yeast sequence (black arrow).

Table 11.1.1 Troubleshooting Guide for BLAST Searches

Problem	Possible causes and solutions
Query returned no hits, or the E-values of the hits are very poor (~1e-05 or higher)	Obtain additional sequence information and repeat the search
	The sequence you are looking for may not exist in some databases. Perform your search using a larger database, such as the nr or EST databases.
	The sequence may not be conserved among different species
	Use Psi-BLAST (Altschul et al., 1997)
Query returned many hits, but did not return a hit from a particular species of interest	Limit your query by including the species name in the Entrez text box. Note that the gene may not be sequenced yet in your species of interest, that the gene sequence may have diverged too much, or that the gene may have been lost in the lineage of that species.
The top hits returned are to unexpected or distantly related species	The sequence may be identical in many different species, producing identical scores in the search. The order in which these sequences are listed on the results page is not significant.
	The gene may have been horizontally transferred between distant organisms
	The vector sequence often found at the beginning or end of a DNA sequence read may still be attached
	Either the query sequence or the hit may have been attributed to the wrong organism
The hits align to only a portion of the sequence	The sequence may contain a highly conserved domain. To obtain information about the remainder of the sequence, remove the domain from the sequence and repeat the search. You may also limit the portion of the sequence used in the search (see A Practical Example).
	The vector sequence often found at the beginning or end of a DNA sequence read may still be attached

sometimes even an organism (i.e., a mutation). In the case of a BLAST search, these mistakes can sometimes lead to very confusing results. After some investigation, we were able to attribute the following explanations to some of the oddest BLAST results we have encountered over the past decade. Consider these cases if your BLAST search shows something completely unexpected.

1. Genes in which two protein-coding domains have been fused together (see below);
2. An RNA gene submitted to Genbank in the wrong direction;
3. A horizontally transferred gene that still most closely resembles sequences in its organism of origin;
4. cDNA sequences in which the researchers forgot to strip off the cloning vector at the ends of the sequence;
5. Trans-spliced mRNAs that are encoded by genes in different parts of the genome;
6. Sequences that were attributed to the wrong organism;
7. Genes that have been incorrectly annotated as two separate, adjacent genes in a genome (instead of a single gene).

Consider these cases if your BLAST search shows something completely unexpected. Describing how we came to each of these conclusions would be beyond the scope of this unit, but the following example shows how we came to discover something very interesting based on an odd BLAST result.

A practical example

In this example, we will show how BLAST searches of various types can be used in a research setting to identify and describe an interesting gene.

The ciliate *Tetrahymena thermophila* is a single-celled organism that lives in freshwater environments all over the world. This organism has likely encountered a huge variety of toxins and pollutants over the years, and its worldwide success suggests it may have genes that enable it to withstand a large number of chemicals. Let us consider *T. thermophila*'s response to the toxic molecule formaldehyde. Our objective will be to use BLAST to learn as much as we can about formaldehyde detoxification in this organism.

We will start by using the sequence of the yeast formaldehyde dehydrogenase

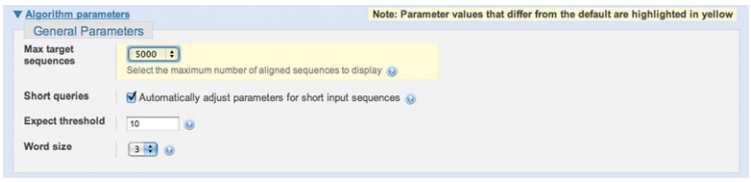

Figure 11.1.24 Screenshot showing how to alter the **Algorithm parameters** to return 5000 hits to the query sequence.

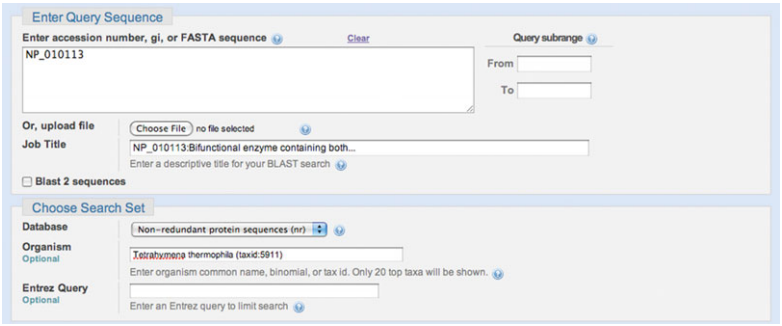

Figure 11.1.25 Screenshot showing how to limit your BLAST search to *Tetrahymena thermophila* sequences in the database.

protein that we discussed in the previous section. Yeast is a model organism whose response to formaldehyde is well known, and the *SFA1* gene encodes the enzyme that directly oxidizes formaldehyde. Since we know that the accession number for this gene is NP_010113, we can enter this number into the protein BLAST Web page and run the search using default parameters. Alternatively, we could use the Entrez search engine at NCBI (Basic Protocol 1) to search for `Saccharomyces[organism] AND formaldehyde detoxification`. NP_010113 is the first entry in the list of results. We could then use this sequence to perform a protein BLAST search.

Unfortunately, no Tetrahymena proteins appear in the list of results when this protein BLAST search is performed. A scan through the results shows that many of the sequences listed in the BLAST results come from closely related fungi and animals. It is likely that Tetrahymena does contain a homolog of this gene, but there are many sequences that are more closely related to our yeast query sequence. There are several ways to fix this problem. First, we could rerun the query increasing the **Max target sequences** option in the **Algorithm parameters** section of the BLASTP page to 5000 (Fig. 11.1.24; see step 7 of Basic Protocol 2). However, simply viewing 5000 description lines will not allow us to find a link to the Tetrahymena protein. In this particular example, the name of the organism does not appear in the abbreviated description line of the protein. If we use the **Reformat results** option (see the Header section under Understanding results) to display 5000 alignments, we can find the Tetrahymena protein (accession number: XP_001013202). The score of this alignment is 426, while the E-value is 2 × 10-117.

A much easier way to find the Tetrahymena homolog of this protein is to use the Organism textbox. This very useful feature in BLAST allows you to enter the name of the taxon or organism you are interested in as an Entrez term. As you enter the term Tetrahymena, the Web page will auto-complete the name for you (Fig. 11.1.25).

Repeating the BLAST search using `Tetrahymena thermophila` as the Organism, the Tetrahymena protein from before is returned as the best hit (XP_001013202; Fig. 11.1.26), with a score of 426 and an E-value of 2 × 10-119. As expected, the score of the hit is the same—remember that this value depends only on the alignment between the two sequences and not on the size of database. The E-value for the second search, however, is much lower than in the first. In the second search, we queried against a database composed of only the Tetrahymena sequences in the nr database. Because the second database

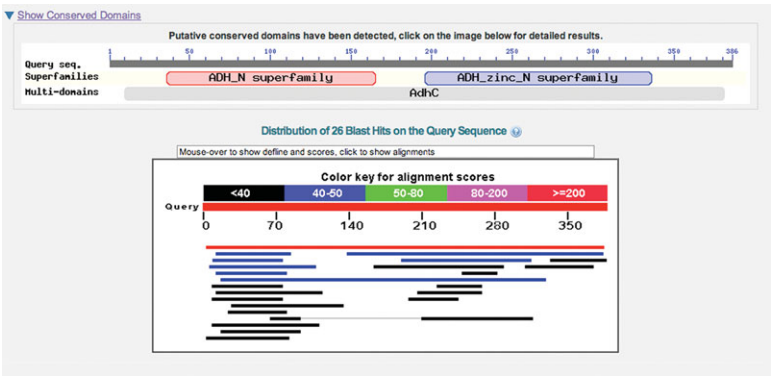

Figure 11.1.26 Graphical results of a BLAST search of the yeast formaldehyde dehydrogenase gene (accession number: NP_010113) against the protein nr database limited to the organism *Tetrahymena thermophila* (see Fig. 11.1.25).

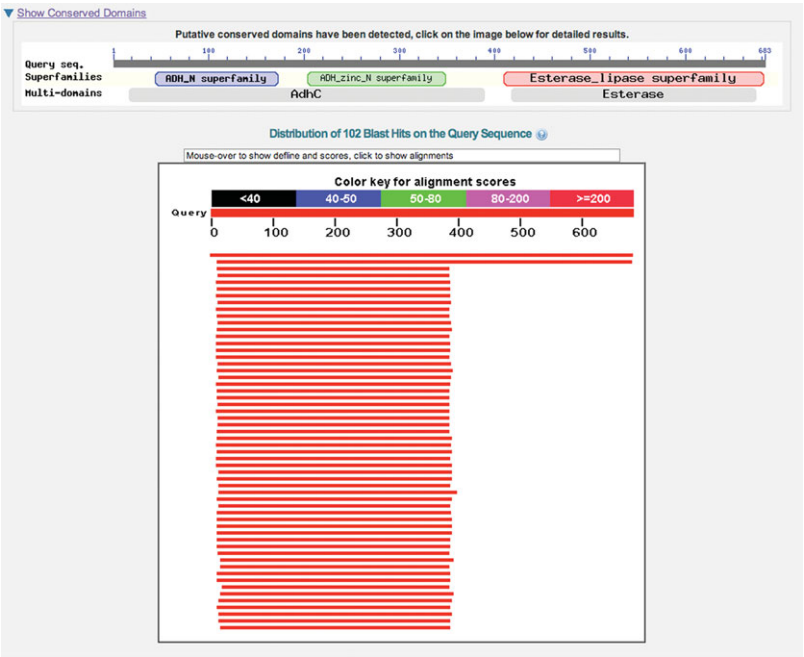

Figure 11.1.27 Graphical result of a BLASTP search of the Tetrahymena formaldehyde dehydrogenase protein (accession: XP_001013202) against the nr protein database.

is significantly smaller, finding a hit with such a high score is less likely than it is when searching the larger, more complete database. This results in a smaller E-value for the second search, indicating a more significant hit.

Clicking on the accession number leads to the GenBank entry page for the Tetrahymena protein. Interestingly, this protein has 683 amino acids; the original yeast protein had only 386 amino acids. We decided to do a BLASTP search of the nr database using the Tetrahymena protein as a query, to see if we could explain the discrepancy in size.

Notice that the conserved domain search shows three conserved domains rather than the two conserved domains found in the yeast gene. The first two domains in the Tetrahymena protein correspond to the two found in the yeast protein. However, the Tetrahymena protein also has an esterase domain (Fig. 11.1.27). Interesting!

The Graphical overview does not really help make sense of this result. Aside from two hits that extend the full length of the query (both corresponding to closely related ciliate proteins), all of the other hits are located in the first 400 amino acids of

Figure 11.1.28 Using the "Query subrange" boxes to limit the BLAST search to only part of your sequence.

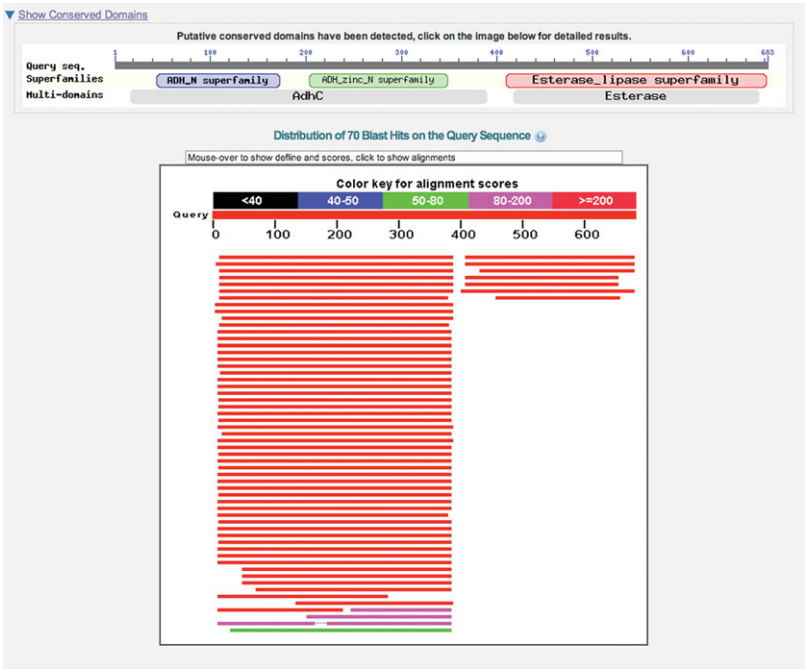

Figure 11.1.29 Graphical result of a BLASTP search of the Tetrahymena formaldehyde dehydrogenase protein (accession: XP_001013202) against all the human sequences in the nr protein database.

the Tetrahymena gene. What is going on here?

Because the nr database is so large, sometimes the hits to one part of the sequence overload the results page and mask hits to the rest of the sequence. There are several ways to solve this problem without simply increasing the number of hits returned by BLAST.

First, you can re-run the search using only a subrange of the query (Fig. 11.1.28), which will show all the target sequences that hit the protein between the selected residues, in this case between 400 and 683. The graphical overview now shows many hits to esterases, just as expected based on the conserved domain search (figure not shown; we encourage you to try this search yourself to see that it works).

A second solution to the problem is to limit the search to a given organism, for example humans, using the Organism textbox. It is easy to see in the graphical overview (Fig. 11.1.29) that the Tetrahymena protein seems to be a fusion of two proteins: a formaldehyde dehydrogenase and an *S*-formylgluthatione hydrolase. It is important to note that this approach may not always work; either one or both of the proteins might not be present in the genome of the organism you are searching. If that is the case, searching using the subrange option will give you better results.

Our next step is to research the functions of formaldehyde dehydrogenase and S-formylgluthatione hydrolase. These enzymes catalyze sequential steps in the formaldehyde detoxification pathway. This is an interesting result, as it suggests that the Tetrahymena gene could be a fused gene able to perform both steps of the pathway. Further experiments confirmed that this fusion gene is expressed in

Tetrahymena cells (Stover et al., 2005). Gene fusions and other events that produce unexpected BLAST results are rare, but they do happen occasionally. Only experience running many BLAST searches can teach you if odd results, like these, are significant new findings, or if they are simply artifacts from sequencing or annotation mistakes—so get BLASTing!

Acknowledgements

This work was funded in part by National Institutes of Health Grant Number 5P40OD010964-12.

Literature Cited

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. , and Lipman, D.J. 1990. Basic local alignment search tool. *J. Mol. Biol.* 215:403-410. doi: 10.1016/S0022-2836(05)80360-2

Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z. , Miller, W., and Lipman, D.J. 1997. Gapped BLAST and PSI–BLAST: A new generation of protein database search programs. *Nucleic Acids Res*. 25:3389-3402. doi: https://doi.org/10.1093/nar/25.17.3389

Bateman, A., Pearson, W.R., Stein, L.D., Stormo, G.D., and Yates, J.R. III. (eds.) 2017. Current Protocols in Bioinformatics. Chapter 3. John Wiley & Sons, Hoboken, N.J.

Chang, W.-J., Zaila, K.E., and Coppola, T.W. 2016. Submitting a sequence to GenBank. *Curr. Protoc. Essen. Lab. Tech*. 12:11.2.1–11.2.24. doi: 10.1002/9780470089941.cpet8s12

Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. 1978. A model of evolutionary change in proteins. *In*Atlas of Protein Sequence and Structure. Vol. 5 (M.O.Dayhoff, ed.) pp. 345-352. National Biomedical Research Foundation, Washington, D.C.

Eddy, S.R. 2004a. Where did the BLOSUM62 alignment score matrix come from? *Nat. Biotechnol.* 22:1035-1036. doi: 10.1038/nbt0804-1035

Engel, S.R., and MacPherson, K.A. 2016. Using model organism databases (MODs). *Curr. Protoc. Essen. Lab. Tech*. 13:11.4.1–11.4.22. doi: 10.1002/cpet.4

Eddy, S.R. 2004b. What is dynamic programming? *Nat. Biotechnol.* 22:909-910. doi: 10.1038/nbt0704-909

Henikoff, S. and Henikoff, J. 1992. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U.S.A.* 89:10915-10919.

Korf, I., Yandell, M., and Bedell, J. 2003. BLAST. O'Reilly Media, Inc., Sebastopol, Calif.

Ladunga, I. 2009. Finding similar nucleotide sequences using network BLAST searches. *Curr. Protoc. Bioinform.* 26:3.3.1-3.3.26. doi: 10.1002/0471250953.bi0303s26

Leonard, S.A., Littlejohn, T.G., and Baxevanis, A.D. 2006. Common file formats. *Curr. Protoc. Bioinform.* 16:A.1B.1-A.1B.9. doi: 10.1002/0471250953.bia01bs16

Needleman, S.B. and Wunsch, C.D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48:443-453. http://dx.doi.org/10.1016/0022-2836(70)90057-4

Smith, T.F. and Waterman, M.S. 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147:195-197. http://dx.doi.org/10.1016/0022-2836(81)90087-5

Stover, N.A., Cavalcanti, A.R., Li, A.J., Richardson, B.C., and Landweber, L.F. 2005. Reciprocal fusions of two genes in the formaldehyde detoxification pathway in ciliates and diatoms. *Mol. Biol. Evol.* 22:1539-1542. doi: 10.1093/molbev/msi151

Wheeler, D. 2003. Selecting the right protein–scoring matrix. *Curr. Protoc. Bioinform.* 00:3.5.1-3.5.6. doi: 10.1002/0471250953.bi0305s00

Zufall, R.A. 2017. Beyond simple homology searches: Multiple sequence alignments and phylogenetic trees. *Curr. Protoc. Essen. Lab. Tech*. 1:11.3.1–1.3.17. doi: 10.1002/9780470089941.et1103s01