Introduction to bioinformatics

BLAST

Basic Local Alignment Search Tool

A CRITICAL GUIDE

Version: 1 August 2018

BLAST

Overview

This Critical Guide provides an overview of the BLAST similarity search tool, briefly examining the underlying algorithm and its rise to popularity. Several Web-based and stand-alone implementations are reviewed, and key features of typical search results are discussed.

Teaching Goals & Learning Outcomes

This Guide introduces concepts and theories embodied in the sequence database search tool, BLAST, and examines features of search outputs important for understanding and interpreting BLAST results. On reading this Guide, you will be able to:

- search a variety of Web-based sequence databases with different query sequences, and alter search parameters;
- explain a range of typical search parameters, and the likely impacts on search outputs of changing them;
- analyse the information conveyed in search outputs and infer the significance of reported matches;
- examine and investigate the annotations of reported matches, and their provenance; and
- compare the outputs of different BLAST implementations and evaluate the implications of any differences.

1 Introduction

From the advent of the first molecular sequence repositories in the 1980s, tools for searching databases became essential. Database searching is essentially a 'pairwise alignment' problem, in which the best match (alignment) is sought between a query sequence and target sequences in the database. Alignments can generally be thought about in two ways: one is to consider similarities along the full length of the compared sequences; the other is to consider similarities that are localised to particular regions – the former are termed 'global', the latter 'local' alignments. This distinction is important because, usually, sequences aren't uniformly similar; there's therefore little point in seeking the best global alignment between sequences that share only local similarity.

Amongst the early global and local pairwise alignment algorithms were the **Needleman and Wunsch**[1] and **Smith-Waterman**[2]. These employ **dynamic programming** approaches, which generally seek solutions to problems by reducing them to smaller, more tractable sub-problems. Between any two sequences, there are usually many possible alignments; the 'best' solution is the one that links together the sub-problems most effectively to create the final alignment. Finding the best match between pairs of sequences in this way is non-trivial; and scaling up to compare a single sequence efficiently against an entire database has become increasingly challenging. When search algorithms were originally developed, databases were relatively small: *i.e.*, contained a few hundred or a few thousand sequences. Even then, algorithms like Needleman and Wunsch and Smith-Waterman were only practicable for aligning small numbers of sequences; for the multi-millions of sequences typical of today's 'post-genome' repositories, they are prohibitively time-consuming.

As databases grew larger, efforts were made to improve search efficiency. One solution was to adapt the available algorithms for specialised hardware; another was to adapt local-similarity algorithms to find short identical matches that could contribute to a total match, and to use **heuristics** to speed the search. **FastA**[3], developed in 1988, was the first such program, based on the idea of finding short words – **k-tuples** – common to the sequences being compared, and using heuristics to join those closest to each other, including the short mis-matched regions between them.

BLAST[4] was the second major example of this type of algorithm, and rapidly exceeded the popularity of FastA, owing to its efficiency and built-in statistics. This Guide briefly introduces the algorithm and a range of its most popular Web-based and stand-alone implementations. From a practical standpoint, it also outlines some of the general features of typical BLAST outputs.

2 About this Guide

The following sections review some of the main features of the BLAST algorithm and of its various implementations. The Guide does not exhaustively describe Web-based front-ends to BLAST, as Web interfaces change frequently; rather, it gives a high-level overview of typical components of 'standard' Web-based BLAST input forms and search outputs. Exercises are provided to help recall the principal differences between some of the many different implementations of BLAST, and to understand and interpret some of the most important output features. Throughout the text, key terms – rendered in **bold** type – are defined in boxes.

KEY TERMS

Dynamic programming: a programming technique that resolves large, often intractable, problems into smaller, solvable sub-problems, combining the results to give an overall solution

FastA: a program for nucleotide/protein sequence database searching

Heuristics: a pragmatic problem-solving 'short-cut' technique that provides a 'close enough' solution, not necessarily the optimal solution

K-tuple: an exact 'word' match between a query & target database sequences (by default, 2 residues in proteins, 6 bases in DNA)

Needleman and Wunsch: an early dynamic-programming algorithm for creating global alignments between biological sequences

Smith-Waterman: an early dynamic-programming algorithm for creating local alignments between biological sequences

3 What is BLAST?

BLAST has enjoyed enormous popularity since the algorithm was first published in 1990, largely because it was more time-efficient than FastA; it was also optimised to work with parallel Unix architectures from an early stage. In consequence, sequence searches performed on public servers were very rapid, a vital attribute for current genome-scale sequence databases.

3.1 The BLAST algorithm

Like FastA, the algorithm uses a heuristic approach that approximates the exhaustive Smith-Waterman algorithm, calculating a solution that, while not guaranteed to be optimal, is nevertheless satisfactory for all practical purposes. Similar to FastA's 'k-tuples' (which represent short, shared 'words'), BLAST also begins by seeking short words, W (for proteins, these are typically 3 letters in length; for DNA, 11). These are required to have an aligned score greater than or equal to a threshold value, T, calculated using a **scoring matrix** (for proteins, by default, this is generally likely to be **BLOSUM62** – although the original 1990 algorithm used **PAM120**). Scanning a target sequence, the algorithm quickly determines whether it contains fixed-length words that can pair with the query sequence to produce word pairs – **segment pairs** – with a score at least equal to the threshold. Any such matches are extended in each direction; these extensions are stopped when the score falls more than a given value, X, below the best score reached so far. In its original implementation, the algorithm returned such High-Scoring Pairs (HSPs) as *un-gapped* local alignments, which were, at the time, characteristic of BLAST outputs.

3.2 Gapped BLAST

In 1997, to reduce execution time and enhance its sensitivity to weak similarities, the basic BLAST algorithm was modified in two significant ways: the first innovation was the introduction of a new criterion for triggering the extension of word matches; the second was a heuristic for generating *gapped* alignments.

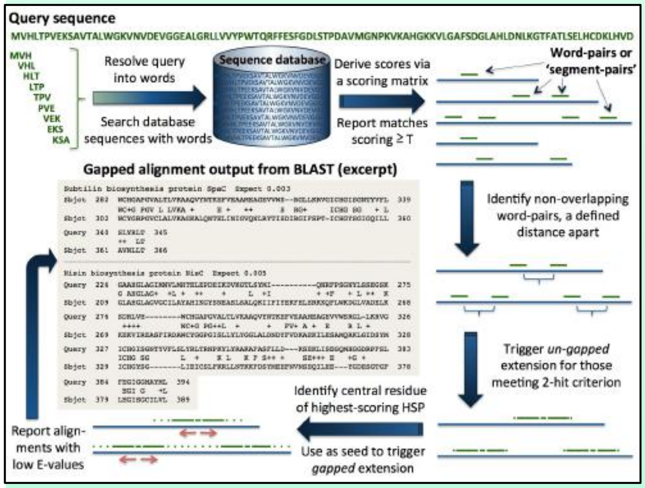

Query sequence
MVHLTPVEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVD
MVH
VHL
HLT
LTP
TPV
PVE
VEK
EKS
KSA
Resolve query into words
Search database sequences with words
Sequence database
Derive scores via a scoring matrix
Report matches scoring ≥ T
Word-pairs or 'segment-pairs'
Gapped alignment output from BLAST (excerpt)
Subtilin biosynthesis protein SpaC Expect 0.003
Nisin biosynthesis protein NisC Expect 0.005
Identify non-overlapping word-pairs, a defined distance apart
Trigger *un-gapped* extension for those meeting 2-hit criterion
Identify central residue of highest-scoring HSP
Use as seed to trigger *gapped* extension
Report alignments with low E-values

Figure 1 ***Schematic illustration of the gapped BLAST algorithm.*** *Word pairs achieving at least a threshold score & lying within a defined distance of one another trigger gapped extensions. From the central residue of the* ***maximal-scoring pair****, gapped alignments are generated.*

The original algorithm looked for word pairs with aligned score at least equal to a threshold value, T. Each of these matches was then extended to determine whether it could yield an HSP; this was the most time-consuming step. The modified algorithm therefore required *two* non-overlapping word pairs to be found within a defined distance of one another before invoking the extension step.

For word pairs satisfying the 'two-hit' criterion, and scoring greater than threshold T, an un-gapped extension is triggered to create HSPs. An HSP achieving a score greater than or equal to a defined value, S, then triggers a *gapped* extension: within the HSP, a central pair of aligned residues is used as the seed, from which gapped extension is performed in each direction using dynamic programming; extension is terminated when the alignment score drops from its maximum achieved value or on reaching the end of one of the sequences. The statistical significance of sequences matched in this way is computed as an **E-value**, denoting the number of random matches expected to achieve the same or greater score in a database of the same size and composition. Although gapped extensions are more time-consuming to compute than un-gapped ones, few are performed, so the total execution time is kept relatively low. Overall, then, rather than calculating several un-gapped alignments, a single gapped alignment is constructed.

In summary, by contrast with the original implementation, the gapped BLAST algorithm[5] requires two non-overlapping hits to achieve at least a threshold score, T, and lie within a defined distance of one another, to stimulate an un-gapped extension; if the HSP generated by this process achieves greater than or equal to threshold score, S, then a gapped alignment is invoked – see Figure 1. The final gapped alignment is only reported if its E-value is sufficiently low to be of interest.

EXERCISES

1 Name two differences between FastA & BLAST.

2 What is an HSP?

3 Name two differences between the original BLAST algorithm & the modified algorithm in gapped BLAST.

4 Give two advantages of using gapped BLAST over the original BLAST. Explain your answers.

KEY TERMS

BLOSUM62: one of a series of scoring matrices whose scores are derived from empirical substitution frequencies observed in locally aligned blocks in the BLOCKS database; in BLOSUM62, sequences in the aligned blocks on which the scores are based share no more than 62% identity

E-value: the number of matches with scores greater than or equal to that of the retrieved match that are expected to occur by chance in a database of the same size & composition, using the same scoring system (the closer the value to 0, the more significant the score)

Maximal scoring pair: an HSP that achieves a maximal score

PAM120: one of a series of scoring matrices whose scores represent the probabilities for all possible amino acid substitutions; PAM matrices are derived from the 1PAM matrix – this gives substitution probabilities for sequences in which 1 point mutation has occurred *per* 100 amino acids

Scoring matrix: a scoring table whose scores (expressed in terms of observed residue frequencies, mutation probabilities, *etc.*) quantify residue similarities at equivalent positions in sequence alignments

Segment pair: a pair of same-length sub-sequences that form an un-gapped alignment between two compared sequences; if the aligned score can't be improved by extending or shortening the segments, this is a locally optimal or High-scoring Segment Pair (HSP)

4 Implementations of BLAST

BLAST can be used in different modes[6] (*e.g.*, via the Web or stand-alone), and for different purposes depending, say, on whether a user's interest is in the analysis of protein or nucleotide sequences. The following sections outline some of the principal Web-based and stand-alone options, and how to access them.

4.1 Web-based BLAST

There are numerous implementations of BLAST on the Web: some are designed to search nucleotide sequence databases with nucleotide sequence queries (blastn), or to perform searches of protein sequence databases with protein sequence queries (blastp); others allow searches of protein sequence databases with 6-frame translated nucleotide sequence queries (blastx), or searches of 6-frame translated nucleotide sequence databases with either protein sequence queries (tblastn) or translated nucleotide sequence queries (tblastx) – see **Figure 2**. There are also options to search genomes of various model organisms (including human, mouse, rat and microbial genomes).

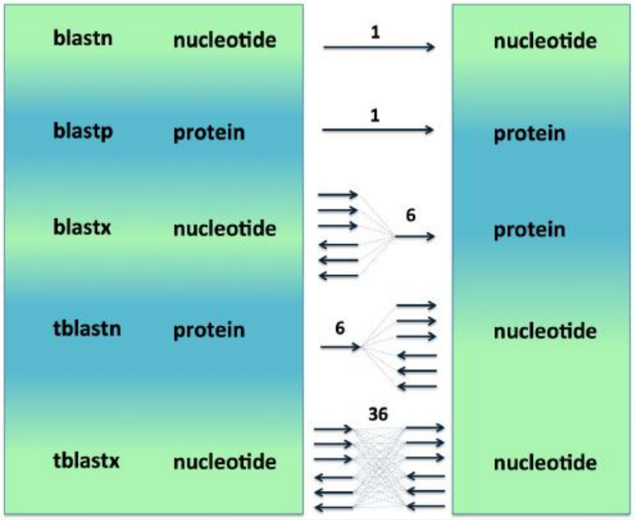


Figure 2 ***Illustration of some common BLAST programs.*** *The type of input sequence is shown in the left-hand panel, & the target database type in the right-hand panel; the central panel shows the number of searches performed by the encoded algorithms.*

For users interested in proteins, further variants of the algorithm have been designed to allow users to i) build **Position-Specific Scoring Matrices** (PSSMs) from results of initial BLAST searches, and to run new searches weighted via the PSSMs (so-called Position-Specific Iterated BLAST, or PSI-BLAST[5]) – successive searches gain greater power as more related (**true-positive**) sequences contribute to the scoring matrix, but search results quickly degrade if **false-positive** sequences are included; ii) perform BLAST searches, but limiting the resulting alignments to those that match a '**pattern**' in the query sequence (Pattern Hit Initiated BLAST, or PHI-BLAST[7]); iii) construct PSSMs from **Conserved Domain Database** search results, and again, to run new searches weighted via the PSSMs (Domain-Enhanced Lookup Time Accelerated BLAST, or DELTA-BLAST[8]); or iv) run an accelerated version of blastp, optimised to match target sequences with 50% or more identity to the query (so-called accelerated protein-protein BLAST, or QuickBLASTP[9]).

For users interested in nucleic acids, the principal variants are i) megablast, which has been optimised to match target sequences with 95% or more identity to the query; and ii) discontiguous megablast, which uses an initial seed that allows mis-matches (ignoring some bases), and was designed to facilitate cross-species comparisons. An overview of some of the different implementations of BLAST available at the **NCBI** is given in **Figure 3**.

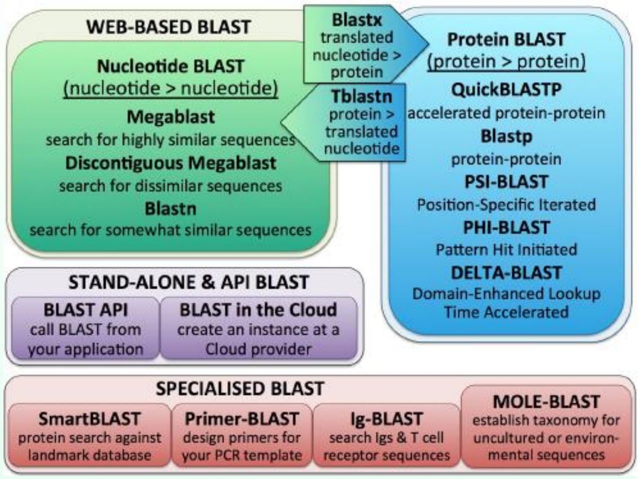

WEB-BASED BLAST
Nucleotide BLAST
(nucleotide > nucleotide)
Megablast
search for highly similar sequences
Discontiguous Megablast
search for dissimilar sequences
Blastn
search for somewhat similar sequences
Blastx
translated nucleotide > protein
Tblastn
protein > translated nucleotide
Protein BLAST
(protein > protein)
QuickBLASTP
accelerated protein-protein
Blastp
protein-protein
PSI-BLAST
Position-Specific Iterated
PHI-BLAST
Pattern Hit Initiated
DELTA-BLAST
Domain-Enhanced Lookup Time Accelerated
STAND-ALONE & API BLAST
BLAST API
call BLAST from your application
BLAST in the Cloud
create an instance at a Cloud provider
SPECIALISED BLAST
SmartBLAST
protein search against landmark database
Primer-BLAST
design primers for your PCR template
Ig-BLAST
search Igs & T cell receptor sequences
MOLE-BLAST
establish taxonomy for uncultured or environmental sequences

Figure 3 ***Snapshot of some of the many implementations of BLAST.*** *Web-based, stand-alone & specialised versions of BLAST are available, allowing users to perform diverse searches online or in-house.*

EXERCISES

1 Name two different Web-based BLAST programs. What type of input sequence & what type of target database do they each require? How many searches does the encoded algorithm perform?

2 The sensitivity of BLAST searches may be enhanced in various ways. One possibility is to iteratively build a potent scoring matrix by running successive database searches. What is the name of the BLAST program used to do this? Generally, why is this a more sensitive approach? What might compromise search results?

3 Name a Web-based BLAST program that allows 'quick & dirty' database searches. Explain why this is a less sensitive approach.

KEY TERMS

Conserved Domain Database (CDD): a resource that uses PSSMs to facilitate the identification of structural domains & to classify protein sequences based on their domain architectures

False-positive: a match to a search query in a particular data-set that is not a true member of the data-set

NCBI: National Center for Biotechnology Information, part of the National Library of Medicine, a branch of the National Institutes of Health, based at Bethesda, Maryland, USA

Pattern: a consensus pattern of amino acid residues or nucleotide bases within a sequence alignment, used as a characteristic signature of the region or family of sequences in which it is found

Position-Specific Scoring Matrix (PSSM): a scoring table that provides numerical values (expressed in terms of observed residue frequencies, mutation probabilities, *etc.*) to quantify the 'similarity' of residues at equivalent positions in sequence alignments

True-positive: a match to a search query in a particular data-set that is a true member of the data-set

4.2 Specialised BLAST

Accompanying the numerous standard BLAST options on the Web are various specialised implementations, customised for different research scenarios. Amongst these are i) SmartBLAST, which allows protein sequence searches against the **Landmark Database**[10]; ii) Primer-BLAST[11], which allows users to find **primers** specific to a **PCR** template; iii) IgBLAST[12], which was initially designed to facilitate the analysis of **immunoglobulin variable domain** sequences, and has been extended to allow the analysis of **T cell receptor** sequences; and iv) MOLE-BLAST[13], which was developed to help identify the closest neighbours to a query sequence, by computing multiple sequence alignments and generating phylogenetic trees.

4.3 Stand-alone and API BLAST

Side-by-side with the many different Web-based implementations, there are various stand-alone BLAST applications for users who prefer, or need, to run searches on their own computers or against their own in-house databases.

For example, stand-alone BLAST+[14] allows separate installation and configuration of the application to users' local environments: this provides greater flexibility for customising and fine-tuning searches, including batch-processing of large query sequences, and the possibility to integrate input and output data into custom **workflows**. In addition, the NCBI provides two similar **RESTful Web Services** to allow submission of BLAST searches programmatically: i) QBLAST or BLAST URL API[15], which offers similar flexibility to stand-alone BLAST+, but allows searches to be performed remotely; and ii) CloudBlast[16] (hosted at Cloud providers such as Amazon Web Services, Google Compute Engine and Microsoft Azure), which allows users to run stand-alone searches with the BLAST+ applications, submit searches using a subset of the BLAST URL API, and perform searches via a simplified Web page – this is particularly helpful for projects requiring large numbers of searches or that involve the use of custom databases.

4.3 Where to find BLAST

The home of BLAST and source of a comprehensive range of BLAST services is the NCBI: blast.ncbi.nlm.nih.gov/Blast.cgi. These services provide search interfaces to a variety of databases, such as **nr** (the default database for blastp), high-throughput genomic sequences, UniProtKB/Swiss-Prot, the **PDB**, patents, *etc*. The NCBI also provides a variety of tutorials and training materials[17], giving an overview of BLAST services, guides to BLAST results, explanations of E-values, how to install stand-alone BLAST applications, and so on.

A subset of BLAST services is also available at the **EBI**: www.ebi.ac.uk/services. These have been set up primarily to search EBI resources, such as **ENA**, vectors, **UniProtKB** (the default for blastp), UniProtKB Taxonomic subsets, sequences of protein structures from **PDBe**, patents, *etc*. In addition, for convenience, embedded BLAST searches are possible directly from individual sequence records within UniProtKB.

EXERCISES

1 Name two specialised implementations of BLAST, & describe the type of search for which they have been customised.

2 Name two advantages of stand-alone BLAST applications.

3 Describe some of the principal differences between BLAST services implemented at the NCBI & those available at the EBI.

4 In what way does the default database for blastp searches performed at the NCBI & at the EBI differ? What impact might this have on search results?

5 Interpreting BLAST results

Clearly, there are many different types of BLAST application. It is beyond the scope of this Guide to describe all the features of the various stand-alone, specialised and standard (Web-based) BLAST services, the minutiae of their interfaces, and facets of their different outputs; nevertheless, a brief overview of some of the principal components of a standard BLAST Web-input form, and of a corresponding output file will serve as an exemplar – a more detailed description of BLAST results is available from the NCBI's tutorials[17,18].

Some features of a standard NCBI protein BLAST Web-input form are shown in **Figure 4** – the blastp tab at the top of the page has been chosen (*n.b.*: the figure is not a screen-shot, but an illustration of selected elements only). At its simplest, the form requires input of a sequence (in **FastA format**) or an **accession number** (*not* an **identifier**). If required, a sub-range of the sequence can be specified,

KEY TERMS

Accession number: a unique computer-readable code that identifies a given entry in a given database

EBI: European Bioinformatics Institute, the EMBL hub dedicated to the provision of bioinformatics services, based at Hinxton, UK

ENA: European Nucleotide Archive, a database comprising the Sequence Read Archive, the Trace Archive & EMBL-Bank

FastA format: a text-based file format for amino acid or nucleotide sequences; the file's first line contains a '>' symbol, followed by the accession number (& sometimes the ID) & sequence title, the rest of the file containing the sequence in single-letter notation

Identifier: a unique 'human friendly' ID code that identifies a given entry in a given database

Immunoglobulin: a major protein component of the immune systems of higher animals; most immunoglobulins possess specific antibody activity, as determined by the structure adopted by variable regions within their amino acid sequences

Immunoglobulin variable domain: part of an immunoglobulin molecule whose amino acid sequence varies in characteristic ways, ultimately to determine the structure of the antibody-combining site

Landmark Database: database housing proteomes from 27 genomes, used to provide a taxonomically diverse, non-redundant set of proteins from well-studied reference species for BLAST searches

nr: a non-redundant database that includes translations of coding sequences in GenBank, PDB sequences, UniProtKB/Swiss-Prot, *etc*.

PCR: Polymerase Chain Reaction, a method used to amplify & reliably replicate specific segments of DNA

PDB: Protein Data Bank, a database of 3D molecular structures (including proteins, nucleic acids & protein-nucleic acid complexes)

PDBe: Protein Data Bank Europe, the European database of macromolecular structures

Primer: a short sequence of RNA or DNA used to initiate DNA synthesis

RESTful: REpresentational State Transfer (REST) is an architectural style that encapsulates how Web-based applications should operate

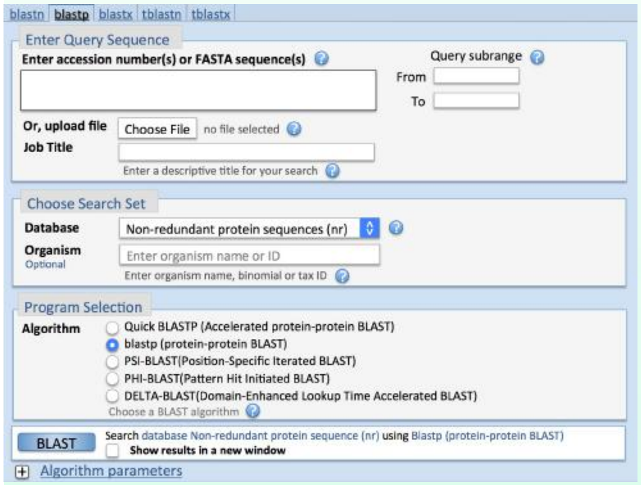
T cell receptor: a T cell-surface molecule that recognises fragments of major histocompatibility complex-bound antigens

UniProtKB: UniProt Knowledgebase, a protein sequence database comprising UniProtKB/Swiss-Prot & UniProtKB/TrEMBL

Web Service: the server component of a client/server architecture that exploits the Web's infrastructure to exchange data or request the execution of functions

Workflow: the sequence of tasks or processes that lead, stepwise, from initiation to completion of a particular analysis or project to focus the search on a particular segment of the query sequence. Alternatively, an input file can be uploaded from a user's local drive. For each option, associated '?' icons provide more information.

The form also allows selection of the desired target database – by default, this is nr, a non-redundant compilation that includes GenBank coding sequence translations and sequences from the PDB, UniProtKB/SwissProt, *etc.*; but a variety of alternative resources can be specified using the toggle button. Optionally, searches may also be narrowed to particular organisms. Importantly, users may choose the required search program, depending on the nature of their query sequence: for proteins, the default is a standard protein-protein search – blastp; for nucleotides, the default is megablast.



Enter Query Sequence

Enter accession number(s) or FASTA sequence(s)

Query subrange
From
To

Or, upload file no file selected

Job Title
Enter a descriptive title for your search

Choose Search Set

Database: Non-redundant protein sequences (nr)

Organism (Optional): Enter organism name or ID
Enter organism name, binomial or tax ID

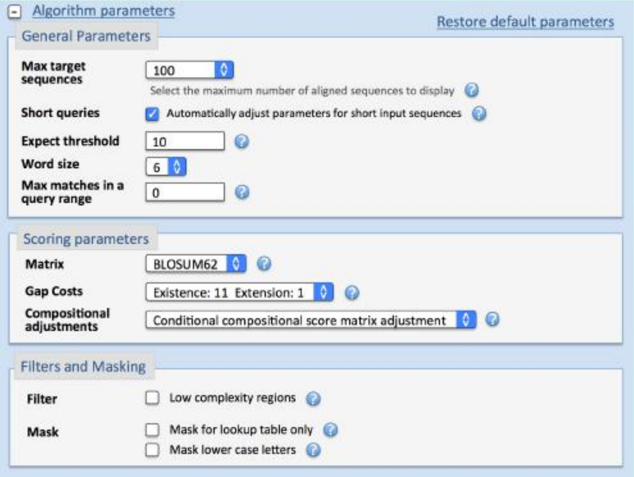
Program Selection

Algorithm
○ Quick BLASTP (Accelerated protein-protein BLAST)
● blastp (protein-protein BLAST)
○ PSI-BLAST(Position-Specific Iterated BLAST)
○ PHI-BLAST(Pattern Hit Initiated BLAST)
○ DELTA-BLAST(Domain-Enhanced Lookup Time Accelerated BLAST)
Choose a BLAST algorithm

Search database Non-redundant protein sequence (nr) using Blastp (protein-protein BLAST)
□ Show results in a new window

Figure 4 ***Typical input fields for Web-based protein BLAST.*** *The form allows input of a query sequence or its accession number, & selection of the target database & search algorithm.*

Figure 4 illustrates the form's default parameters (here, searching the nr database using blastp), as summarised beside the BLAST initiation button (at the bottom of the form). Accepting the defaults can be effective for many users; however, searches may be further refined by toggling the Algorithm parameters button (base of **Figure 4**) and modifying the defaults – these are illustrated in **Figure 5**.



General Parameters

Max target sequences: 100
Select the maximum number of aligned sequences to display

Short queries: ☑ Automatically adjust parameters for short input sequences

Expect threshold: 10

Word size: 6

Max matches in a query range: 0

Scoring parameters

Matrix: BLOSUM62

Gap Costs: Existence: 11 Extension: 1

Compositional adjustments: Conditional compositional score matrix adjustment

Filters and Masking

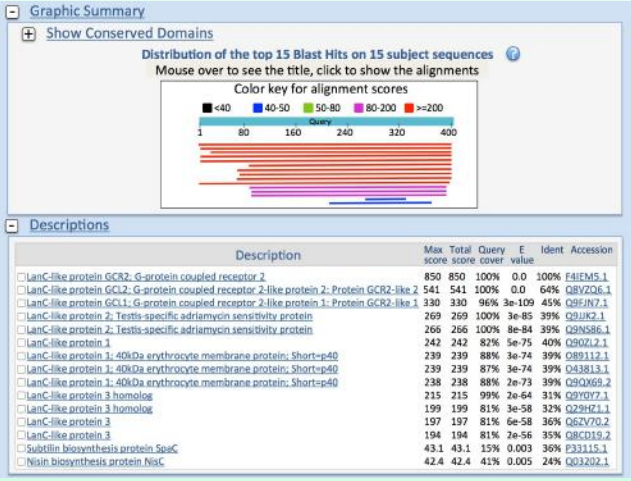
Filter: □ Low complexity regions

Mask: □ Mask for lookup table only
□ Mask lower case letters

Figure 5 ***Expanded Web-form for modifying BLAST parameters.*** *Various general & scoring parameters may be adjusted, & filters applied, to modify both the algorithm's behaviour & the display of its results.*

Amongst other things, via this expanded form, users may choose i) the number of aligned sequences to display in the output; ii) the expected number of matches to achieve a given score by chance (those with E-values above the threshold – *i.e.*, that are less significant – are filtered out); iii) the word size (smaller sizes result in more sensitive – 'noisy' – searches); iv) the scoring matrix (adjusting scores to the degree of evolutionary descent – the larger the BLOSUM number, the higher the sequence similarity and hence the smaller the evolutionary distance); v) penalties for creating and extending alignment gaps (such penalties being related to the choice of scoring matrix); and vi) to mask out compositionally biased regions (*i.e.*, those enriched with a particular amino acid(s)) within the query sequence that could generate spurious matches in the output.

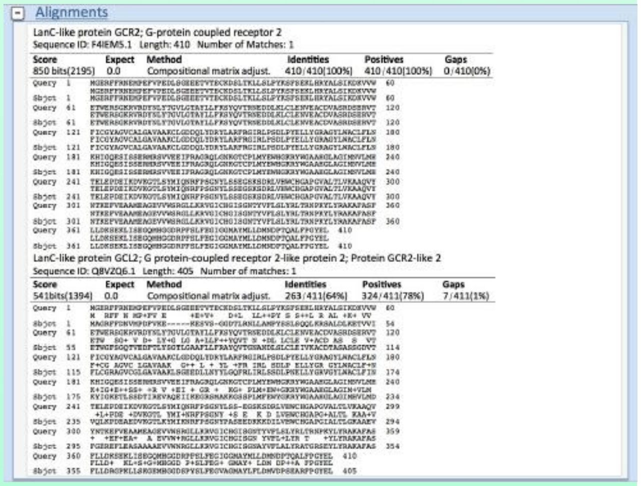
Some features of the output from a standard NCBI protein BLAST against UniProtKB/Swiss-Prot (but otherwise using default parameters) are shown in **Figures 6** and **7**. The first element is a graphical overview of the matched sequences, colour-coded to show the overall quality of each match in terms of its score and length (coverage). Below the graphical summary, the titles of the sequences are tabulated, together with the match scores, the query coverage, the match E-values, percent identities and database accession numbers. This table is followed by pairwise alignments of the Query with each of the matched database sequences (Sbjct), including details of the number of residue identities, similarities (positives) and gaps.



Distribution of the top 15 Blast Hits on 15 subject sequences
Mouse over to see the title, click to show the alignments

Description	Max score	Total score	Query cover	E value	Ident	Accession
□ LanC-like protein GCR2; G-protein coupled receptor 2	850	850	100%	0.0	100%	F4IEM5.1
□ LanC-like protein GCL2; G-protein coupled receptor 2-like protein 2; Protein GCR2-like 2	541	541	100%	0.0	64%	Q8VZQ6.1
□ LanC-like protein GCL1; G-protein coupled receptor 2-like protein 1; Protein GCR2-like 1	330	330	96%	3e-109	45%	Q9FJN7.1
□ LanC-like protein 2; Testis-specific adriamycin sensitivity protein	269	269	100%	3e-85	39%	Q9JJK2.1
□ LanC-like protein 2; Testis-specific adriamycin sensitivity protein	266	266	100%	8e-84	39%	Q9NS86.1
□ LanC-like protein 1	242	242	82%	5e-75	40%	Q90ZL2.1
□ LanC-like protein 1; 40kDa erythrocyte membrane protein; Short=p40	239	239	88%	3e-74	39%	O89112.1
□ LanC-like protein 1; 40kDa erythrocyte membrane protein; Short=p40	239	239	87%	3e-74	39%	O43813.1
□ LanC-like protein 1; 40kDa erythrocyte membrane protein; Short=p40	238	238	88%	2e-73	39%	Q9QX69.2
□ LanC-like protein 3 homolog	215	215	99%	2e-64	31%	Q9Y0Y7.1
□ LanC-like protein 3 homolog	199	199	81%	3e-58	32%	Q29HZ1.1
□ LanC-like protein 3	197	197	81%	6e-58	36%	Q6ZV70.2
□ LanC-like protein 3	194	194	81%	2e-56	35%	Q8CD19.2
□ Subtilin biosynthesis protein SpaC	43.1	43.1	15%	0.003	36%	P33115.1
□ Nisin biosynthesis protein NisC	42.4	42.4	41%	0.005	24%	Q03202.1

Figure 6 ***Typical features of Web-based protein BLAST output.*** *Details of each match (e.g., title, coverage, E-value, percent identity, accession number) follow a graphical overview that shows the match quality.*



LanC-like protein GCR2; G-protein coupled receptor 2
Sequence ID: F4IEM5.1 Length: 410 Number of Matches: 1

Score	Expect	Method	Identities	Positives	Gaps
850 bits(2195)	0.0	Compositional matrix adjust.	410/410(100%)	410/410(100%)	0/410(0%)

```
Query  1    MGERFFRNEMPEFVPEDLSGEEETVTECKDSLTKLLSLPYKSFSEKLHRYALSIKDKVVW  60
            MGERFFRNEMPEFVPEDLSGEEETVTECKDSLTKLLSLPYKSFSEKLHRYALSIKDKVVW
Sbjct  1    MGERFFRNEMPEFVPEDLSGEEETVTECKDSLTKLLSLPYKSFSEKLHRYALSIKDKVVW  60
Query  61   ETWERSGKRVRDYSLYTGVLGTAYLLFKSYQVTRNEDDLKLCLENVEACDVASRDSERVT  120
            ETWERSGKRVRDYSLYTGVLGTAYLLFKSYQVTRNEDDLKLCLENVEACDVASRDSERVT
Sbjct  61   ETWERSGKRVRDYSLYTGVLGTAYLLFKSYQVTRNEDDLKLCLENVEACDVASRDSERVT  120
Query  121  FICGYAGVCALGAVAARCLGDDQLYDRYLARFRGIRLPSDLPYELLYGRAGYLWACLFLN  180
            FICGYAGVCALGAVAARCLGDDQLYDRYLARFRGIRLPSDLPYELLYGRAGYLWACLFLN
Sbjct  121  FICGYAGVCALGAVAARCLGDDQLYDRYLARFRGIRLPSDLPYELLYGRAGYLWACLFLN  180
Query  181  KHIGQESISSERMRSVVEEIFRAGRQLGNKGTCPLMYEWHGKRYWGAAHGLAGIMNVLMH  240
            KHIGQESISSERMRSVVEEIFRAGRQLGNKGTCPLMYEWHGKRYWGAAHGLAGIMNVLMH
Sbjct  181  KHIGQESISSERMRSVVEEIFRAGRQLGNKGTCPLMYEWHGKRYWGAAHGLAGIMNVLMH  240
Query  241  TELEPDEIKDVKGTLSYMIQNRFPSGNYLSSEGSKSDRLVHWCHGAPGVALTLVKAAQVY  300
            TELEPDEIKDVKGTLSYMIQNRFPSGNYLSSEGSKSDRLVHWCHGAPGVALTLVKAAQVY
Sbjct  241  TELEPDEIKDVKGTLSYMIQNRFPSGNYLSSEGSKSDRLVHWCHGAPGVALTLVKAAQVY  300
Query  301  NTKEFVEAAMEAGEVVWSRGLLKRVGICHGISGNTYVFLSLYRLTRNPKYLYRAKAFASF  360
            NTKEFVEAAMEAGEVVWSRGLLKRVGICHGISGNTYVFLSLYRLTRNPKYLYRAKAFASF
Sbjct  301  NTKEFVEAAMEAGEVVWSRGLLKRVGICHGISGNTYVFLSLYRLTRNPKYLYRAKAFASF  360
Query  361  LLDKSEKLISEGQMHGGDRPFSLFEGIGGMAYMLLDMNDPTQALFPGYEL  410
            LLDKSEKLISEGQMHGGDRPFSLFEGIGGMAYMLLDMNDPTQALFPGYEL
Sbjct  361  LLDKSEKLISEGQMHGGDRPFSLFEGIGGMAYMLLDMNDPTQALFPGYEL  410
```

LanC-like protein GCL2; G protein-coupled receptor 2-like protein 2; Protein GCR2-like 2
Sequence ID: Q8VZQ6.1 Length: 405 Number of matches: 1

Score	Expect	Method	Identities	Positives	Gaps
541bits(1394)	0.0	Compositional matrix adjust.	263/411(64%)	324/411(78%)	7/411(1%)

```
Query  1    MGERFFRNEMPEFVPEDLSGEEETVTECKDSLTKLLSLPYKSFSEKLHRYALSIKDKVVW  60
            M  RFF N MP+FV E       +E+V+    D+L  LL++PY S S++L R AL +K+ VV
Sbjct  1    MAGRFFPNVMPDFVKE-----KESVS-GDDTLRNLLAMPYSSLSQQLKRSALDLKETVVI  54
Query  61   ETWERSGKRVRDYSLYTGVLGTAYLLFKSYQVTRNEDDLKLCLENVEACDVASRDSERVT  120
            ETW  SG+ V D+ LY+G LG A+LLF++YQVT N +DL LCLE V+ACD AS  S  VT
Sbjct  55   ETWGFSGQTVEDFTLYSGTLGAAFILFRAYQVTGSANDLSLCLEIVKACDTASASSGDVT  114
Query  121  FICGYAGVCALGAVAARCLGDDQLYDRYLARFRGIRLPSDLPYELLYGRAGYLWACLFLN  180
            F+CG AGVC LGAVAAK  G++ L + YL +FR IRL SDLP ELLYGR GYLWACLF+N
Sbjct  115  FLCGRAGVCGLGAVAAKLSGEEDLLNYYLGQFRLIRLSSDLPNELLYGRVGYLWACLFIN  174
Query  181  KHIGQESISSERMRSVVEEIFRAGRQLGNKGTCPLMYEWHGKRYWGAAHGLAGIMNVLMH  240
            K+IG+E++SS+ +R V +EI + GR +  KG+ PLM+EW+GKRYWGAAHGLAGIM+VLM
Sbjct  175  KYIGKETLSSDTIREVAQEIIKEGREMAKKGSSPLMFEWYGKRYWGAAHGLAGIMHVLMD  234
Query  241  TELEPDEIKDVKGTLSYMIQNRFPSGNYLSS-EGSKSDRLVHWCHGAPGVALTLVKAAQV  299
             +L+PDE +DVKGTL YMI+NRFPSGNY +S E  K D LVHWCHGAPG+ALTL KAA+V
Sbjct  235  VQLKPDEAEDVKGTLKYMIKNRFPSGNYPASEEDKKDILVHWCHGAPGIALTLGKAAEV  294
Query  300  YNTKEFVEAAMEAGEVVWNRGLLKRVGICHGISGNTYVFLSLYRLTRNPKYLYRAKAFAS  359
            +  +EF+EA+  A EVVWN+GLLKRVGICHGISGN YVFL+LYR T    +YLYRAKAFAS
Sbjct  295  FGEREFLEASAAAAEVVWNKGLLKRVGICHGISGNAYVFLALYRATGRSEYLYRAKAFAS  354
Query  360  FLLDKSEKLISEGQMHGGDRPFSLFEGIGGMAYMLLDMNDPTQALFPGYEL  410
            FLLD+  KL+S+G+MHGGD P+SLFEG+ GMAY+ LDN DP++A FPGYEL
Sbjct  355  FLLDRSPKLLSKGEMHGGDSPYSLFEGVAGMAYLFLDMVDPSEARFPGYEL  405
```

Figure 7 ***Typical features of Web-based protein BLAST output.*** *For each match, a pairwise alignment with the query sequence is provided.*

EXERCISES

1 Run blastp at the NCBI using the sequence F4IEM5 as input, & selecting UniProtKB/Swiss-Prot as the target database, to generate the full version of the output illustrated in Figures 6 & 7. Save the result. Now repeat the search using default parameters. What is the main difference between the outputs? What are the top hits from each search? Are they the same (follow the hyperlinked accession numbers to retrieve the database entries & find out more)? How might you account for any differences?

2 Repeat the first search above with word size 2. Do the results differ? Are all the retrieved matches significant? Explain your answer.

3 Repeat the first search in (1) with BLOSUM45 as the scoring matrix. Do the results differ from the previous search? Are all the retrieved matches significant? Explain your answer.

4 Repeat the first search in (1), selecting PSI-BLAST as the algorithm. Do the results differ from the first search? If so, why might this be so? Press the 'Go' button to run iteration 2. How do the results differ from the first iteration? Were more significant matches located?

TAKE HOMES

1 NCBI BLAST is a popular similarity search tool for protein & nucleotide sequences owing to its efficiency & built-in statistics;

2 The original algorithm (which produced un-gapped alignments) was superseded by 'gapped BLAST'; many innovations have since followed, including PSI-BLAST, PHI-BLAST, DELTA-BLAST & megablast;

3 Alongside Web-based applications are various specialised & standalone versions of BLAST; the latter allow configuration of the application to run in users' local environments;

4 Web-based BLAST allow users to upload or paste in query sequences or accession numbers, & select the target database & program;

5 Choice of database, program & algorithm parameters (word size, scoring matrix, *etc*.) make notable differences to search results – it's therefore important to customise results by changing the defaults;

6 Search outputs provide graphical overviews of matched sequences, & alignments of those sequences with the query; each sequence is hyperlinked to its originating database record, allowing verification of the provenance of the match & of its annotations;

7 Annotations of matches may be misleading & should be verified against their database records & with respect to other matches.

6 References & further reading

1 Needleman SB & Wunsch CD. (1970) **A general method applicable to the search for similarities in the amino acid sequences of two proteins.** *J. Mol. Biol.*, **48**, 443-453.

2 Smith TF & Waterman MS. (1981) **Identification of common molecular subsequences.** *J.Mol. Biol.*, **147**, 195-197.

3 Lipman DJ & Pearson WR. (1985) **Rapid and sensitive protein similarity searches.** *Science*, **227**, 1435-1441.

4 Altschul SF *et al.* (1990) **Basic local alignment search tool.** *J. Mol. Biol.*, **215**(3), 403-410.

5 Altschul SF *et al.* (1997) **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.** *Nucleic Acids Res.*, **25**(17), 3389-3402.

6 **BLAST Guide: Overview of the various NCBI BLAST services & reports: ftp.ncbi.nih.gov/pub/factsheets/HowTo_BLASTGuide.pdf**

7 Zhang Z *et al.* (1998) **Protein sequence similarity searches using patterns as seeds.** *Nucleic Acids Res.*, **26**(17), 3986-3990.

8 Boratyn GM *et al.* (2012) **Domain enhanced lookup time accelerated BLAST**. *Biol. Direct*, **7**, 12.

9 **QuickBLASTP adds pre-processing to BLAST search: ncbi-insights.ncbi.nlm.nih.gov/2017/05/17/quickblastp-adds-pre-processing-to-blast-search**

10 **Landmark Database: blast.ncbi.nlm.nih.gov/smartblast/smartBlast.cgi?CMD=Web&PAGE_TYPE=BlastDocs#searchSets**

11 Ye J *et al.* (2012) **Primer-BLAST: a tool to design target-specific primers for polymerase chain reaction.** *BMC Bioinformatics*, **13**, 134.

12 Ye J *et al.* (2013) **Ig-BLAST: an immunoglobulin variable domain sequence analysis tool.** *Nucleic Acids Res.*, **41**(Web Server issue), W34-W40.

13 **MOLE-BLAST: A tool for clustering multiple sequences with their database neighbors: ftp.ncbi.nih.gov/pub/factsheets/Factsheet_MOLE-BLAST.pdf**

14 **BLAST+ user manual – How to run stand-alone BLAST searches. www.ncbi.nlm.nih.gov/books/NBK279690**

15 **Common URL API: ncbi.github.io/blast-cloud/dev/api.html**

16 **NCBI BLAST Cloud Documentation: ncbi.github.io/blast-cloud**

17 **NCBI Tutorials: www.ncbi.nlm.nih.gov/home/tutorials**

18 **The New BLAST Results Page: Enhanced graphical presentation and added functionality: ftp.ncbi.nih.gov/pub/factsheets/HowTo_NewBLAST.pdf**

7 Acknowledgements & funding

GOBLET Critical Guides marry ideas from the Higher Apprenticeship specification for college-level students in England (**www.contentextra.com/lifesciences/unit12/unit12home.aspx**) with the EMBnet Quick Guide concept. A Quick Guide to NCBI BLAST was written by Eduardo Fernandes Formighieri *et al.* in 2004: **www.embnet.org/shared/quickguides/09-guideBLAST.pdf**.

This Guide was developed with the support of a donation from EMBnet to the GOBLET Foundation.

Design concepts and the Guide's front-cover image were contributed by CREACTIVE.

8 Licensing & availability

The Guide is freely available for download via the GOBLET portal (**www.mygoblet.org**) and EMBnet website (**www.embnet.org**).

9 Disclaimer

Every effort has been made to ensure the accuracy of this Guide; GOBLET cannot be held responsible for any errors/omissions it may contain, and cannot accept liability arising from reliance placed on the information herein.

About the organisations

GOBLET

GOBLET (Global Organisation for Bioinformatics Learning, Education & Training) was established in 2012 to unite, inspire and equip bioinformatics trainers worldwide; its mission, to cultivate the global bioinformatics trainer community, set standards and provide high-quality resources to support learning, education and training.

GOBLET's ethos embraces:

- ***inclusivity***: *welcoming all relevant organisations & people*
- ***sharing***: *expertise, best practices, materials, resources*
- ***openness***: *using Creative Commons Licences*
- ***innovation***: *welcoming imaginative ideas & approaches*
- ***tolerance***: *transcending national, political, cultural, social & disciplinary boundaries*

Further information about GOBLET and its Training Portal can be found at **www.mygoblet.org** and in the following references:

- Attwood *et al.* (2015) **GOBLET: the Global Organisation for Bioinformatics Learning, Education & Training.** *PLoS Comput. Biol.*, **11**(5), e1004281.
- Corpas *et al.* (2014) **The GOBLET training portal: a global repository of bioinformatics training materials, courses & trainers.** *Bioinformatics*, **31**(1), 140-142.

GOBLET is a not-for-profit foundation, legally registered in the Netherlands: CMBI Radboud University, Nijmegen Medical Centre, Geert Grooteplein 26-28, 6581 GB Nijmegen. For general enquiries, contact **info@mygoblet.org**.

EMBnet

EMBnet, the Global Bioinformatics Network, is a not-for-profit organisation, founded in 1988 as a network of institutions, to establish and maintain bioinformatics services across Europe. As the network grew, its reach expanded beyond European borders, creating an international membership to support and deliver bioinformatics services across the life sciences: **www.embnet.org**.

Since its establishment, a focus of EMBnet's work has been bioinformatics Education and Training (E&T), and the network therefore has a long track record in delivering tutorials and courses worldwide. Perceiving a need to unite and galvanise international E&T activities, EMBnet was one of the principal founders of GOBLET. For more information and general enquiries, contact **info@embnet.org**.

CREACTIVE

CREACTIVE, by Antonio Santovito, specialises in communication and Web marketing, helping its customers to create and manage their online presence: **www.gocreactive.com**.

About the author

Teresa K Attwood (orcid.org/0000-0003-2409-4235)

Teresa (Terri) Attwood is a Professor of Bioinformatics with more than 25 years' experience teaching introductory bioinformatics, in undergraduate and postgraduate degree programmes, and in *ad hoc* courses, workshops and summer schools, in the UK and abroad.

With primary expertise in protein sequence analysis, she created the PRINTS protein family database and co-founded InterPro (her particular interest is in the analysis of G protein-coupled receptors). She has also been involved in the development of software tools for protein sequence analysis, and for improving links between research data and the scientific literature (most notably, Utopia Documents).

She wrote the first introductory bioinformatics text-book; her third book was published in 2016:

- Attwood TK & Parry-Smith DJ. (1999) ***Introduction to Bioinformatics.*** Prentice Hall.
- Higgs P & Attwood TK. (2005) ***Bioinformatics & Molecular Evolution.*** Wiley-Blackwell.
- Attwood TK, Pettifer SR & Thorne D. (2016) ***Bioinformatics challenges at the interface of biology and computer science: Mind the Gap.*** Wiley-Blackwell.

Affiliation

School of Computer Science, The University of Manchester, Oxford Road, Manchester M13 9PL (UK).