Program Options for blastall

Tao Tao, Ph.D.
NCBI User Service

TOC

1. Introduction

BLAST is the acronym for "**B**asic **L**ocal **A**lignment **S**earch **T**ool", which is a **local** alignment search tool first described in 1990 by Altschul et al [1]. NCBI started providing sequence alignment service to the public using BLAST in 1992, first through its blast email server (decommissioned in 2002) and later through the web (1997).

[1] Altschul SF, Gish W, Miller W, Myers EW and Lipman DJ (1990) J Mol Biol 215:403-410.

BLAST now refers to a family of different programs based on the same heuristics. These programs along with the accessory tools are made available to the public as a single binary package. Despite this expansion, blastall - the original program with enhenced features, remains to be the core of the BLAST package.

1.1 What does blastall do?

The all in "blastall" reveals some of the capability of this program. It can perform one of five different searches depending on the input specified through the -p option. The acceptable inputs and the type of search they performs are:

blastn compares nucleotide queries to a nucleotide database
blastp compares protein queries to a protein database
blastx compares the translated products of nucleotide queries to a protein database
tblastn compares protein queries to the translated products of a nucleotide database
tblastx compares translational products of nucleotide queries to the translational products

from a nucleotide database

1.2. How does blastall find the alignment?

BLAST finds the optimal alignment by using the "word matching" algorithm, in which BLAST does the search in several distinctive phases: 1) generating overlapping words from the input query, 2) scanning the database for word matches (hits), and 3) extending word hits to produce (local) alignments through three steps of extension.

During the first phase, BLAST breaks the input query into short overlapping segments, or "words," and stores them a hash table. BLAST takes those query words and scans the target database for initial matches in the second phase. The nucleotide BLAST algorithm looks for any single exact word match. The protein BLAST algorithm uses a scoring threshold cutoff to identify matches. In addition, protein BLAST algorithm also requires two word hits within a certain distance in order to proceed to the next step.

In the third phase, those initial matches or word hits are used as seeds to generate the alignments in through three extension step:

a. The first step is the un-gapped extension, in which BLAST extends the word hits in both directions to generate initial alignments without the introduction of gap.
b. The second step is the gapped extension, in which BLAST extends those initial un-gapped alignments further by introducing gaps in the alignments.
c. The last step is the final gapped extension with trace-back, in which BLAST does a final gapped extension attempt with trace-back to generate the actual alignments.

It is important to remember that limit on the number of alignments, number of descriptions, and e-value cut-off are applied at each of the three extension steps. Very stringent settings may cause BLAST to miss the best hits since that hit may not be above the cutoff during the un-gapped extension step.

1.3 Where can we find the blast program?

Currently, the binaries for all these standalone blast programs and accessory tools are provided to the public as a single downloadable archive at ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/. Most of the common platforms - combination of chipset and OS - are supported.

Table 1.3 Command Line BLAST Archives and Their Target Platform

Archive Name	Target Platform	Instruction Set
blast-#.#.#-axp64-tru64.tar.gz	HP Alpha with tru64 OS	Big endian
blast-#.#.#-ia32-freebsd.tar.gz	Pentium Compatible PC with FreeBSD OS	Little endian
blast-#.#.#-ia32-linux.tar.gz	Pentium Compatible PC with Linux OS	Little endian
blast-#.#.#-ia32-solaris#.tar.gz	Pentium Compatible PC with Solaris OS version #	Little endian
blast-#.#.#-ia32-win32.exe	Pentium Compatible PC with Windows OS	Little endian
blast-#.#.#-mips64-irix.tar.gz	64 bits MIPS processor with IRIX OS	Big endian
blast-#.#.#-ppc64-aix.tar.gz	64 bits PowerPC running IBM AIX OS	Big endian
blast-#.#.#-ppc32-macosx.tar.gz	PowerPC processor with Max OSX	Big endian
blast-#.#.#-universal-macosx.tar.gz	ia32 and PowerPC32 running Max OSX	Big endian
blast-#.#.#-sparc64-solaris#.tar.gz	64 bits Sparc processor wiht Solaris OS version #	Big endian
blast-#.#.#-x64-linux.tar.gz	X64(Amd64/em64) running 64bit Linux OS	Little endian
Note: rpsblast databases are platform dependent.		

2. Installation and execution

The installation of blast package can be divided into the following four steps:

1) Downloading
2) Extraction (installation)
3) Setup
4) Execution

Since Windows version differs significantly from the Unix/Linux versions, it will be described separately first.

2.1 For PC platform running Windows OS

2.1.1 Downloading

You can download the latest blast binary package for Windows from:

ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/

The Windows standalone blast archive is a self extracting archive named with the following convention:

blast-#.#.#-ia32-win32.exe

Here #.#.# represent the current version number or the patch release date if bug fixes were made after the official release.

2.1.2 Installation

To keep all the relevant files/directories from the archive in one place, we need to fisrt create a new subdirectory under E:\ directory (or other directory of choice to you), named as blast-#.#.#, with #.#.# indicating the version number. For 2.2.13, we can name this subdirectory blast-2.2.13. Move the saved blast archive to this subdirectory and double click to extract the programs and other files. We will see a DOS terminal window open briefly with file names flashing by.

This installs the blast package and creates three subdirectories under the blast-2.2.13: bin, data, and doc. The bin subdiretory contains the following programs:

```
bl2seq.exe      blastall.exe     blastclust.exe      blastpgp.exe
copymat.exe     fastacmd.exe     formatdb.exe        formatrpsdb.exe
impala.exe      makemat.exe      megablast.exe       rpsblast.exe
seedtop.exe
```

The documents on individual programs are in the doc subdirectory. The data subdirectory contains matrices for scoring protein alignments. Files needed by other NCBI programs, such as Cn3D and Sequin, are also included. For better management of database files, we also need to create a subidrectory named db under blast-2.2.13 to keep our blast databases.

2.1.3 Setup

To ensure the smooth execution of blast programs, we need to set up a BLAST configuration file, named ncbi.ini to instruct blast the location of the data directory and db directory. In the above setup, the path to data directory and db directory should be specified the follwing way:

```
[NCBI]
DATA=E:\blast-2.2.13\data

[BLAST]
BLASTDB=E:\blast-2.2.13\db
```

Steps needed to created this file on PC are the following:

- launch notepad
- copy paste the above section
- relace the path (E:\blast-2.2.13\) with the actual one for your setup
- name the file as ncbi.ini using "save as". Double quote the name to prevent Windows from adding

the .txt extension
- move this resulted ncbi.ini file to your Windows folder

We need to place the resulted ncbi.ini file under the Windows, winnt, or the system directory (exact location will depend on the version of Windows installed). BLAST programs will read this file upon start to get the path information it needs during the search.

2.1.4 Execution

To run the program, you need first launch a DOS command prompt as depicted by the screenshot. DOS prompt is generally kept under:

Start ⇨ Program ⇨ Accessories ⇨ Command Prompt

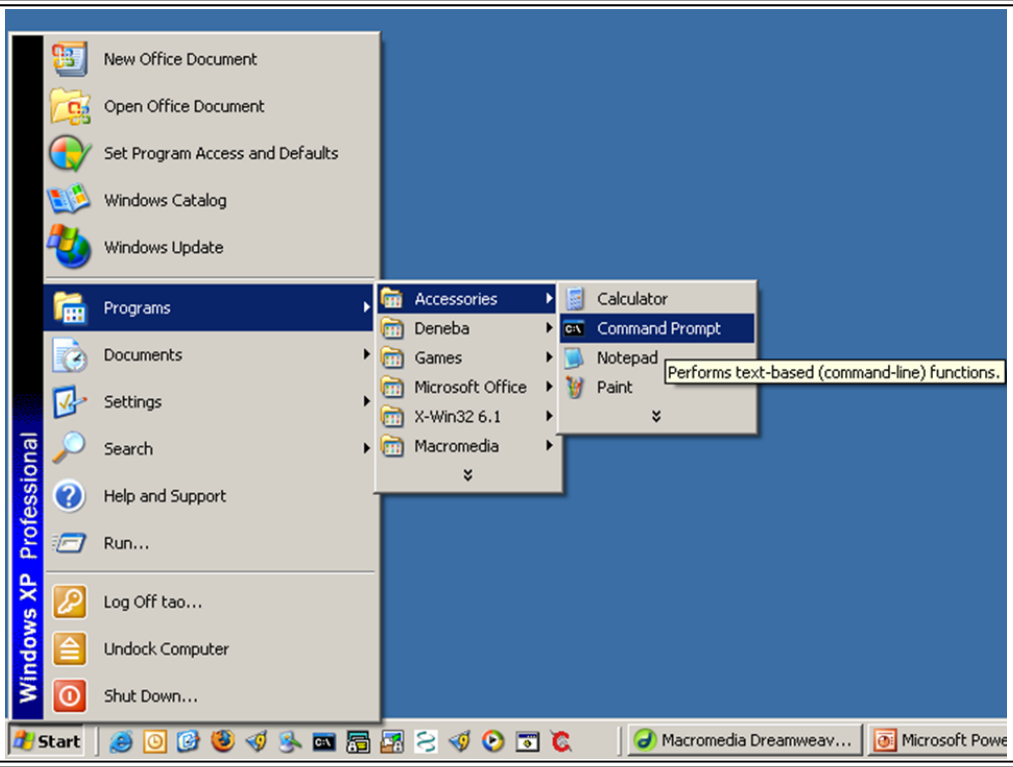

Figure 2.1.4 Openning a DOS command prompt under window (XP)

This will open a command window. In this window, change the working directory to "E:\blast-2.2.13" by typing "E:" and enter key stroke, followed by "cd blast-2.2.13" and enter key stroke. Create the db directory using "mkdir db" command. Sample commad lines are listed in Table 2.1.4 for your reference.

Table 2.1.4 Sample command lines

Command Prompt Display	Command Meaning
C:\Documents and Settings\tao>*e:*	Change directory to E:\ drive
E:\>*cd blast-2.2.13*	Change directory to blast-2.2.13
E:\blast-2.2.13>*mkdir db*	Create db directory under blast-2.2.13
E:\blast-2.2.13>*dir*	List files under blast-2.2.13
Volume in drive E has no label.	
Volume Serial Number is EC96-7870	
Directory of E:\blast	
12/14/2005 10:28 PM <DIR> .	System file
12/14/2005 10:28 PM <DIR> ..	System file
12/21/2005 12:05 AM <DIR> bin	bin directory
12/04/2005 03:51 PM <DIR> data	data directory
12/04/2005 03:52 PM <DIR> db	db directory

12/04/2005 03:52 PM <DIR> doc 6 Dir(s) 13,327,708,160 bytes free E:\blast-2.2.13>	doc directory

NOTE:
Command inputs are colored in red.

To make windows aware of the location of blast programs, you need to add the path "E:\blast-2.2.13\bin\" to the PATH environment variable under "Start ⇨ Control Panel ⇨ System ⇨ Environment Variables ⇨ Path." Terminate the PATH string with a semicolon (;), then append "E:\blast2211\bin\" after it without the double quotes.

After this, we will be able to call the programs using their name followed by the appropriate option value pairs as input from any directory in the computer. For example, to call blastall, use its blastn subprogram, search against refseq_rna database with fasta_query.txt as query, and save the result in output.txt, we will use the following command line:

```
blastall -p blastn -d refseq_rna -i fasta_query.txt -o ouput.txt
```

To further customize the search, we can manipulate the relevant search parameters by referring to the parameter list in Section 3 below. Note, refseq_rna is a NCBI provided database. It is available from the db subdirectory (ftp.ncbi.nih.gov/blast/db) in preformatted form. See this web document for more information: ftp.ncbi.nih.gov/blast/documents/blastdb.html.

2.2 For machines running Unix or Linux

2.2.1 Downloading

You can download the latest blast binary package from the same directory as for Windows:

ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/

The binaries for various platforms other than windows have ".tar.gz" file extension. They are named using blast-#.#.#-CHIPSET-OS.tar.gz convention, where the #.#.# represents the current version number, or the patch release date if bug fixed version was made. The CHIPSET represents the processor. And the OS represents the operating system.

You need to select the appropriate archive for your platform (machine/OS combination) and save it to local disk by right click on the file and choose "Save target as …" within a browser. Or you can use FTP client to download it in binary form.

Note that NCBI does not support asymmetric protocol and requires that an FTP connection be made in PASSIVE mode. One should use ftp client with such capability, ncftp is one of such a client available.

2.2.2 Installation

To install, you need to inflate the downloaded archive first using the "gunzip -d archive.tar.gz" command line. The resulted tar file should be extracted using "tar -xvpf archive.tar" command line. Successful execution of the above two steps for the 2.2.13 archive will create a "blast-2.2.13" directory with the following subdirectories within it: bin, doc, data.

For better file management, you can also create an additional db subdirectory for database files using "mkdir db" command when you are under the blast-2.2.12 directory.

2.2.3 Setup

To set up, you should create a file named .ncbirc to store the configuration information. For example, if you keep the blast-2.2.13 directory under your home directory and you also created a db subdirectory within the blast-2.2.13, then you can specify the path to data and db directory the following way:

```
[NCBI]
DATA=~/blast-2.2.13/data

[BLAST]
BLASTDB=~/blast-2.2.13/db
```

We recommend the use of absolute path if the blast directory is kept in places other than the user's home directory. You should place the .ncbirc file in your own home directory.

2.2.4 Execution

If the path to blast-2.2.13/bin directory is not included in the PATH environment variable, you would need to call blast programs by appending the path before the program name. For example, under blast-2.2.13 directory, you can call the program using the following command line, where "./bin" instructs shell to look in the bin subdirectory under the current directory:

```
./bin/blastall -
```

This command line example will display all the command line parameters for blastall on the screen. Actual search will require additional option/value inputs. The path to matrix and database will not be needed once the .ncbirc is configured properly. Make sure that your downloaded databases are placed in the db subdirectory.

2.3 For MacOSX

2.3.1 Downloading and Installation

Blast binary for MacOSX is under the same directory as for Windows and Unix platform. It is named with the following convention, with ppc32-macosx indicating the platform, e.g. powerPC based mac with OSX:

```
blast-#.#.#-ppc32-macosx.tar.gz
```

We will make binaries for Intel chip based Macs.

Installation for Macintosh is essentially the same as under other Unix systems if one choose to download it using ftp client and performs the installation using command line. If you choose to download it using browser, StuffIt Expander will automatically expand the archive to install the folder, named blast-2.2.13 for current release, on the desktop with the preconfigured directory structure. After this, you should create a db folder and move the blast-2.2.13 folder to your own home directory, which generally resides at:

```
Computer icon ⇨ Home icon ⇨ icon with your name
```

2.3.2 Setup

You need to launch Terminal program (mostly under the Utilities folder) to get a terminal prompt. By default, the working directory of terminal window is your home directory. Once there, you can create the .ncbirc file to put in the configuration information using pico editor and 'pico .ncbirc" command line (without quotes). The file should contain the following information:

```
[NCBI]
DATA=/Users/name/blast-2.2.13/data

[BLAST]
BLASTDB=/Users/name/blast-2.2.13/db
```

Here the "name" is your home directory's name. You can get the complete path by type "pwd". If your system file structure is different, you need to change the path to point to the right directory.

2.3.3 Execution

You can call the program using the instructions given for other Unix/Linux platforms given above. You can

create a query file in this directory and search it against the database. For example if your working directory is blast-2.2.13, the following command will search sequences in an input query file "my_sequences" against a database called "nr". The output is to be saved in my_output:

```
./bin/blastall -p blastn -d nr -i my_sequence -o my_output
```

If you want to be able to call blast programs from any directory, you need to append the path "/User/name/blast-2.2.13/bin" to your $PATH environment variable. You may need to check with your system admin on what the exact path is and how you should modify that setting.

3. Program parameters for blastall

blastall is a command line program with no graphic user interface (GUI). We control the program through command line option switches issued in a terminal window. Those options instruct blastall what program function, query, and database to use. They also control the search sensitivity, format the result is return in, and to which file the result should be saved, etc. Typing "blastall -" followed by a return key stroke will display the blastall parameters along with short descriptions on each one.

3.1. Presentation format

The following section lists all the available options for blastall and provides functional explanation as well as usage examples. For clarity, each option is listed in its own table and all tables follow the same general format. Example provided for each option shows a valid option/value combination as they should appear in an actual command line. The options are based on the current 2.2.13 version. The first four options (-p, -i, -d, -o) are mandatory parameters for most searches.

3.2. Individual command line options of blastall

Table 3.2.1	
Parameter	**-p**
Function	Specifies the type of search
Default	**None**
Input format	String
Example	To instructs blastall to run blastn program, use: -p blastn
Note	Program strings and types of searches they specify Program: blastn | blastp | blastx | tblastn | tblastx Query: Nuc. Acid | Protein | Nuc. Acid | Protein | Nuc. Acid DB: Protein | Protein | Protein | Nuc. Acid | Nuc. Acid

Table 3.2.2	
Parameter	**-d**
Function	Specifies the target database(s)
Default	**nr**
Input format	String
Example	To search database named est_human, use: -d est_human
Note	Use database file name WITHOUT the file extension. Even though multiple databases can be specified in command line, using -d "nr est", database alias file is a much better way to do this. See formatdb.html for more information. Search multiple large databases together may encounter memory related problems.

Table 3.2.3	
Parameter	**-i**
Function	Specifies input query file
Default	**stdin**

Input format	String
Example	To search query file my_query.txt, use: -i my_query.txt
Note	Use the complete file name WITH its extension. The query should be in FASTA format. If multiple entries are in the input file, all queries will be searched. To use stdin default, omit the -i and redirect the file using '<'.

Table 3.2.4	
Parameter	**-o**
Function	Specifies output file
Default	**stdout (dump to screen)**
Input format	String
Example	To save result in a file named out.txt, use: -o out.txt
Note	-p, -i, -d, -o are mandatory parameters for most searches.

Table 3.2.5	
Parameter	**-e**
Function	Specifies Expectation value cutoff
Default	**10**
Input format	Real
Example	To specify an e-value cutoff of 2 × 10-20, use: -e 2e-20
Note	Accepted formats are integer (100), fraction (1/500), decimal (0.005), and exponential (5e-5).

Table 3.2.6	
Parameter	**-m**
Function	Specifies alignment view
Default	**0**
Input format	Integer
Example	To get blast result in XML format, use: -m 7
Note	Input value for -m and the alignment view they specify Value — Alignment Display Format 0 — traditional pairwise 1 — query-anchored showing identities 2 — query-anchored no identities 3 — flat query-anchored, show identities 4 — flat query-anchored, no identities 5 — query-anchored no identities and blunt ends 6 — flat query-anchored, no identities and blunt ends 7 — XML Blast output 8 — tabular (not post processing, no sorting) 9 — tabular with comment lines (post-processed, sorted) 10 — ASN, text 11 — ASN, binary

Table 3.2.7	
Parameter	**-F**
Function	Specifies filter(s) to be used to mask query sequence

Default	**T (DUST for NT, SEG for PRT)**
Input format	String
Example	To inactivate filter, use: -F F
Note	Accepted strings/syntax: T, F, D, L, R, V, S, C, and m. m in -F stands for masking for lookup table only, which enables blast to extend through the masked region. L stands for Low complexity, D for DUST, R for human Repeats, and V for Vector. S stands for SEG, which has other user specifiable values: -F "S 10 1.0 1.5" SEG filter: window=10; low cut=1; high cut=1.5. C stands for COIL, which also has user specifiable values [2, 3]: -F "C 28 40 32" COIL filter: window=22; cutoff=40; linker=32. To run SEG and COIL filter together, use: -F "S; C" To mask lookup table only, add m: -F "m S; C" To mask repeat sequences use: -F R or -F "m R" To combine all together, use: -F "m L;R" To mask vector filter, use: -F "V" To use -F R, repeat libraries from www.girinst.org are needed.

[2] Lupas et al. (1991). Science 252: 1162 - 1164
[3] Wilson et al. (1995). J. Gen. Virol. 76: 2923 - 2932. (Implemented by John Kuzio).

Table 3.2.8	
Parameter	**-G**
Function	Specifies the gap opening cost
Default	**0 (zero invokes default: 5 for blastn, varies for others)**
Input format	Integer
Example	To reduce gap opening penalty to 3, use: -G 3
Note	See Table 5.1 and 5.2 for more information.

Table 3.2.9	
Parameter	**-E**
Function	Specifies the gap extension cost
Default	**0 (zero invokes default: 2 for blastn, varies for others)**
Input format	Integer
Example	To reduce gap extension penalty to 1, use: -E 1
Note	See Table 5.1 and 5.2 for more information.

Table 3.2.10	
Parameter	**-I (capital i)**
Function	Show GI's in definition lines
Default	**F**
Input format	T or F
Example	To show GI's in the deflines, use: -I T
Note	Output comparison -I F: ref|NP_001005339.1| Regulator of G-protein ... -I T: **gi|52694755|**ref|NP_001005339.1| Regulator of G-protein ...

Table 3.2.11	
Parameter	**-q (for blastn only)**
Function	Penalty for a nucleotide mismatch
Default	**-3**
Input format	Integer
Example	To change nucleotide mismatch penalty to -2, use: -q -2
Note	Different -r/-q ratio is optimal for alignments with different percent of identities. Restrictions on input values for -r/-q as well as -G/-E were instroduced since 2.2.13 release. See Table 5.2 for details.

Table 3.2.12							
Parameter	**-r (for blastn only)**						
Function	Reward for a nucleotide match						
Default	**1**						
Input format	Integer						
Example	To increase the nucleotide match reward to 2, use: -r 2						
Note	Different -r/-q ratio is optimal for finding alignment with different percent identity. Suggested -r/-q settings for different sequence identity						
	Percent id	99	98	95	90	85-80	75	65
	Ratio	1/-3	2/-5	1/-2	2/-3	4/-5	1/-1	5/-4

Table 3.2.13	
Parameter	**-v**
Function	Number of one line description of database sequences shown
Default	**500**
Input format	Integer
Example	To set the upper limit of description to 2000, use: -v 2000
Note	This sets the number of one-line description displayed at the beginning of blast result (for -m 0 to -m 6). Web counterpart is "Descriptions".

Table 3.2.14	
Parameter	**-b**
Function	Number of database sequence alignments shown
Default	**250**
Input format	Integer
Example	To set the number of alignment to 2000, use: -b 2000
Note	Upper limit is 200000. Web counterpart is "Alignments". This is NOT the total number of alignment segments or high scoring pairs (HSPs). Rather it is the number of database sequences with HSP(s) to the query, which is ≤ to the number of HSP(s).

Table 3.2.15	
Parameter	**-f**
Function	Threshold for extending word hits
Default	**0 (zero invokes default: blastp 11, blastx 12, tblastn 13, tblastx 13)**
Input format	Integer
Example	To increase the threshold to 13, use: -f 13

Note	Smaller value increases the search sensitivity. Not used by blastn and megablast 0, specified by -W setting instead.

Table 3.2.16	
Parameter	**-g**
Function	Performs gapped alignment (not available with tblastx)
Default	**T**
Input format	T or F
Example	To do ungapped alignment, use: -g F
Note	Default is gapped alignment.

Table 3.2.17	
Parameter	**-Q**
Function	Query Genetic code to use (for blastx and tblastx only)
Default	**1**
Input format	Integer
Example	To use non-universal translation code of 13, use: -Q 13
Note	Determines which translation table to use on query in translated BLAST search. Default 1 is the universal translation table, see: www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi?mode=c

Table 3.2.18	
Parameter	**-D**
Function	DB Genetic code (for tblast[nx] only)
Default	**1**
Input format	Integer
Example	To use the non-universal translation code 13, use: -D 13
Note	Determines which translation table to use for the database entries in translated BLAST search tblastn and tblastx. Default is to use the universal code. See link in Table 3.2.17 for more information.

Table 3.2.19	
Parameter	**-a**
Function	Number of CPUs/processors to use
Default	**1**
Input format	Integer, smaller or equal to the number of CPUs available
Example	To use four CPU's, use: -a 4
Note	Unlike Solaris, Linux displays multithreaded or a forked process as multiple individual processes.

Table 3.2.20	
Parameter	**-O**
Function	Generate SeqAlign output
Default	**Optional**
Input format	[File Out]
Example	To save the SeqAlign to align.asn, use: -O align.asn
	When used, BLAST will save the ASN.1 SeqAlign object to the file specified. You

Note	can use this file to reformat the result into vrious formats. See files under ftp.ncbi.nlm.nih.gov/blast/demo/ for more information. This requires the NCBI toolkit and the corresponding BLAST database.

Table 3.2.21	
Parameter	**-J**
Function	Believe the query defline
Default	**F**
Input format	T or F
Example	To trust or believe the query defline, use: -J T
Note	Default is false since query deflines may not follow NCBI format. Must be set to T to use -O output.

Table 3.2.22	
Parameter	**-M**
Function	Specifies which protein scoring matrix to use
Default	**BLOSUM62**
Input format	String
Example	To score alignment with PAM30 matrix, use: -M PAM30
Note	For protein alignment, BLAST only supports the following five matrices: BLOSUM45, BLOSUM62, BLOSUM80, PAM30, and PAM70. BLOSUM matrices were derived from actual ungapped block alignments of sequences with indicated percentage of similarities. PAM matrices are derived from closely related sequences with over 90% similar. Less stringent PAM matrices were derived from PAM1 using mathematical extrapolation.

Table 3.2.23	
Parameter	**-W**
Function	Word size
Default	**0**
Input format	Integer
Example	To set word size to 27, use: -W 27
Note	Zero invokes default: blastn 11, megablast 28, all others 3. Only two options are available for protein alignment.

Table 3.2.24	
Parameter	**-z**
Function	Effective length of the database
Default	**0**
Input format	Real
Example	To set the effective length of database to 200000, use: -z 200000
Note	Use zero for the actual database size.

Table 3.2.25	
Parameter	**-K**
Function	Number of best hits from a region to keep
Default	**0**
Input format	Integer

Example	To use a value of 100, use: -K 100
Note	This selects only the specified number of best hits for a given region of the query for further evaluation. Zero turns it off. If used, 100 is the recommended value.

Table 3.2.26	
Parameter	**-P (protein alignment only: blastp, blastx, tblastn, tblastx)**
Function	Single versus multiple hit
Default	**0**
Input format	Integer
Example	To activate single hit mode, use: -P 1
Note	Under default for non-blastn search, BLAST will only extend word hits if there are two word hits within the distance specified by -A. Zero invokes multiple hit, 1 for single hits. Not used by blastn or megablast.

Table 3.2.27	
Parameter	**-Y**
Function	Effective length of the search space
Default	**0**
Input format	Real
Example	To set search space to 10000000, use: -Y 1000000
Note	Use zero for actual search space size. This is the product of effective query length and effective database length, which are actual length corrected for the edge effect.

Table 3.2.28	
Parameter	**-S**
Function	Query strands to search against database
Default	**3**
Input format	Integer
Example	To search only the reverse complement of the query, use: -S 2
Note	For blastn, blastx, and tblastx only: 1 = input strand; 2 = reverse complement; 3 = both

Table 3.2.29	
Parameter	**-T**
Function	Produce HTML output
Default	**F**
Input format	T or F
Example	To generate HTML formatted output, use: -T T
Note	If the database is from NCBI, the matched subject sequences will be hyperlinked to corresponding record in Entrez.

Table 3.2.30	
Parameter	**-I (lowercase L)**
Function	Restrict search of database to subset specified by GIs in the input file
Default	**Optional**
Input format	String

Example	To restrict the search to GIs in my_gi.txt, use: -l my_gi.txt
Note	Argument is the name of a text file that contains a list of GIs. blastall will restrict the search to this subset of sequences in the database. This file should be in the same directory as the database, or in the directory that BLAST is called from. You can only use this with NCBI databases. For commonly used subset, database alias is a better alternative. GI numbers can be obtained from Entrez Nucleotide or Entrez Protein with appropriate query terms. Refer to the Help link on the left sidebar for details.

Table 3.2.31	
Parameter	**-U**
Function	Use lower case filtering of FASTA sequence
Default	**F**
Input format	T or F
Example	To turn "lowercase masking" on, use: -U T
Note	This option specifies that lower-case letters in the input FASTA file should be masked. Make sure that only the "undesirable" portions of the sequence to be filtered are in lowercase.

Table 3.2.32				
Parameter	**-y**			
Function	X dropoff value for ungapped extensions (in bits)			
Default	**0**			
Input format	Real			
Example	To set ungapped extension X dropoff to 20, use: -y 20			
Note	-y default (invoked by 0 input value)			
	Program	blastn	megablast	all others
	Default -y value	20	10	7

Table 3.2.33					
Parameter	**-X**				
Function	X dropoff value for gapped alignment (in bits)				
Default	**0**				
Input format	Integer				
Example	To set gapped extension X dropoff to 25, use: -X 25				
Note	-X default (invoked by 0 input value)				
	Program	blastn	megablast	tblastx	all others
	Default value	30	20	Not used	15

Table 3.2.34	
Parameter	**-Z**
Function	X dropoff value for final gapped alignment (in bits)
Default	**0**
Input format	Integer
Example	To set final gapped extension X dropoff to 75, use: -Z 75
	-Z default (invoked by 0 input value)

Note	Program / Default value: blastn 50; megablast 50; all others 25

Table 3.2.35	
Parameter	**-R**
Function	Use a PSI-TBLASTN checkpoint file as the scoring matrix
Default	**Optional**
Input format	String
Example	To read in a checkpoint file query.chk, use: -R query.chk
Note	This parameter takes the checkpoint file generated by the -C parameter of blastpgp (standalone PSI-BLAST) and uses it as matrix in the tblastn search. The same query must be used.

Table 3.2.36	
Parameter	**-n**
Function	Activate MEGABLAST algorithm for blastn search
Default	**F**
Input format	T or F
Example	To invoke megablast algorithm, use: -n T
Note	When used, -W is reset to 28 if not explicitly specified.

Table 3.2.37	
Parameter	**-L**
Function	Location on query sequence (subsequence of the query to use)
Default	**Optional**
Input format	String
Example	To search subsequence between 100-400, use: -L "100,400"
Note	Convention is -L "start,end", where the start and end are the coordinates for the subsequence. Start position is marked as 1.

Table 3.2.38	
Parameter	**-A**
Function	Multiple hits window size
Default	**0**
Input format	Integer
Example	To specify non-default value of 30, use: -A 30
Note	Distance between two hits Program / Default value: blastn/megablast 0 (not used); discontiguous megablast 50; all others 40

Table 3.2.39	
Parameter	**-w**
Function	Frame shift (Out Of Frame, OOF) penalty

Default	**0**
Input format	Integer
Example	To activate OOF penalty with desired value (5), use: -w 5
Note	Non-zero setting invokes OOF (Out Of Frame) algorithm for blastx

Table 3.2.40	
Parameter	**-t**
Function	Length of the largest intron allowed in tblastn for linking HSPs
Default	**0**
Input format	Integer
Example	To allow link of HSPs 10000 bases apart, use: -t 10000
Note	BLAST will use the value specified to link hits. Zero disables linking.

Table 3.2.41	
Parameter	**-B**
Function	Specifies the number of queries to concatenate for blastn, tblastn
Default	**N/A**
Input format	Integer
Example	To concatenate 100 queries for each database scan, use: -B 100
Note	A feature is similar in principle, but different in implementation, to that found in megablast. It allows query concatenation for non-megablast blastn and tblastn searches and decreases the time needed for the search since database is scanned only once. It should not be combined with -g F. The argument to -B option must be equal to the number of sequences in the FASTA input file. When the -B option is used, the results may differ from the ones produced with individual queries due to the heuristic nature of BLAST. It is guaranteed that matching sequences will appear in the same order when they are tied in evalue and are part of the output both with and without -B. When the -B option is used, the summary statistics at the bottom of the output are for the combined set of queries, not tabulated for the individual queries in a multiple-query input.

Table 3.2.42	
Parameter	**-V**
Function	Force use of old engine
Default	**F**
Input format	[T/F]
Example	
Note	Optional. Since 2.2.13, the default is to use new BLAST engine.

Table 3.2.43	
Parameter	**-C**
Function	Use composition-based statistics for tblastn
Default	**D**
Input format	[String]
Example	To turn on unconditional composition-based statistics, use: -C 3
	Available Input Values to -C D or d: default (equivalent to F) 0 or F or f: no composition-based statistics

Note	1 or T or t: Composition-based statistics [4] 2: Composition-based score adjustment conditioned on sequence properties [5] 3: Composition-based score adjustment unconditionally [5] For programs other than tblastn, it must be set to D, F or 0.

[4] Nuc Acid Res (2001) 29: 2994-3005.
[5] Bioinformatics (2005) 21: 902-911.

Table 3.2.44	
Parameter	**-s**
Function	Compute locally optimal Smith-Waterman alignments
Default	**F**
Input format	[T/F]
Example	To compute locally optimal Smith-Waterman alignments, use: -s T
Note	This option is only available for gapped tblastn.

4. General Usage

4.1 Nucleotide vs nucleotide search using blastn

Nucleotide vs nucleotide search can be used to identify the input sequences, find related sequences, map mRNA to its genomic counterpart, and map primers to their annealing target. The following command line searches the input nucleotide query in my_query file against the nt database using megablast algorithm with word size set to 56. The results are saved in my_output:

```
blastall -p blastn -i my_query -d nt -n T -W 56 -o my_output
```

The -n T and -W 56 increase the stringency and the speed of the search. It is good for matching highly similar sequences - such as in mapping mRNAs to their genomic counterparts.

4.2 Nucleotide vs protein search with blastx

This search involves the translation of the query and the actual alignment is performed at the protein level. It is useful in identify the potential protein product(s) that might be encoded by a nucleotide entry especially when the query nucleotide sequence still has certain errors. Also because the comparison is done at the more sensitive protein level, it is a useful search to identify homologous sequences that may escape the direct nucleotide search.

In the following command line, it searches nucleotide sequences in my_query against the protein database nr. The upper limit for returned description and alignments is set to 100 and the search results are saved in my_output:

```
blastall -p blastx -i my_query -d nr -v 100 -b 100 -o my_output
```

4.3 Protein vs protein search with blastp

This search directly compares the input protein queries against a protein database. It is useful in finding other proteins that share sequence similarities to the input query. The matching sequences found by blastp can help identify the function of the input query. The following command line searches the input protein query in my_query file against the refseq_protein database. In attempt to identify the exact matches, if present in the databases, the filter function is inactivated. The result is saved in my_output:

```
blastall -p blastp -i my_query -d refseq_protein -F F -o my_output
```

4.4 Protein vs nucleotide search with tblastn

Often the organism of interest only has a limited number of protein sequences available and the target gene of interest is not found there. One way to identify the gene sequence is through translated tblastn search against the nucletide sequences from the target organism. The query can be a known protein

homolog from a well studied organism as human, mouse, or other model organisms.

The following search is an attempt to identity the gene sequences from Lactobacillus casei wgs that encodes the mismatch repair protein mutL. The query is Escherichia coli K12 mutL protein. The wgs search is restricted to the Lactobacillus casei entries using -l lactobacillus_gi [*].

```
blastall -i e_coli_mutL -d wgs -l lactobacillus_gi -p tblastn -o lacto_mutL.out
```

[*] wgs is the preformatted NCBI database and lactobacillus is a gi list generated using entrez query term: wgs[prop] AND Lactobacillus casei[orgn]

4.5 Nucleotide vs nucleotide search with tblastx

This search is generally reserved as a last resort when all the other searches (blastn, blastp, blastx, and tblastn) fail to return any useful information. Due to the computational intensity, it takes much more time to complete. In addition, tblastx searches tend to generate much more spurious hits and have a much higher noise to signal ratio. This makes the subsequent result interpretation very difficult and time consuming. It should be used with caution.

The following commandline attempts to find the mutL counterpart in Ferroplasma through tblastx search. The search is limited to Ferroplasma portion through the input Ferroplasma_gi file[**].

```
blastall -i e_coli_mutL.ORF -d wgs -l Ferroplasma_gi -p tblastx -o Ferroplasma_mutL.out
```

[**] wgs is the preformatted NCBI database and lactobacillus is a gi list generated using entrez query term: wgs[prop] AND Ferroplasma[orgn]

4.6 Additional information and on searching with short queries

The most commonly adjusted parameters are -F, -e, -b, -v, and -m. The remaining parameters are for more complex searches, which often require optimization that deviates from the default settings. For example, to try to get longer alignment, your can try increasing the X-dropoff values specified by -X, -y, and -Z parameters.

As mentioned in the beginning, -e, -b, -v cutoff limits are applied at each of the three alignment steps. The side-effect of stringent settings in those parameters is that certain HSPs may not be included in the final result since the HSPs from ungapped extension step fall below the cutoff and was not carried to the gapped extension steps.

When BLAST searching with short query sequences, there will not be reliable way to gauge the statistical significance of the matches due the short nature and biased composition of the query. To make this type of searches working, special option/value pairs will be needed. For short nulcoetide query, we recommend adding the following option/value pairs to the command line:

```
-F F -e 1000 -W 7
```

For short peptide queries, we recommend addition the following option/value pairs to the command line:

```
-F F -e 20000 -W 2 -M PAM30
```

5. Additional information on scoring matrices and gap penalties

5.1 Nucleotide scoring marices and their -G/-E values

Nucleodite blast searches through megablast, blastall, and bl2seq have until now allowed users to select arbitrary gap existence and extension penalties for a blastn type search. This has been convenient for users but has led to the unfortunate situation that searches with some parameter sets were significantly overestimating the statistical significance of matches. The parameters that might cause an issue here are -r, -q, -G, and -E.

To address this, the proper statistical parameters for a number of -r /-q/-G/-E values have been calculated starting from version 2.2.13. Note that above a certain gap existence and extension penalty any value is

permitted, as the statistics for ungapped searches can be used. These are marked as "ungapped threshold" below.

Table 6.1 Supported Nucleotide Score Matrices and Their Allowed -G/-E Inputs

Matrix	G	E	Matrix	G	E	Matrix	G	E	Matrix	G	E
-r 1 -q -4	1	2	-r 1 -q -3	1	2	-r 2 -q -7	2	4	-r 2 -q -5	2	4
	0	2		0	2		0	4		0	4
	2	1		2	1		4	2		4	2
	1	1		1	1		2	2		2	2
	2	**2**		**2**	**2**		**4**	**4**		**4**	**4**
Matrix	**G**	**E**	**Matrix**	**G**	**E**	**Matrix**	**G**	**E**	**Matrix**	**G**	**E**
-r 1 -q -2	1	2	-r 1 -q -1 *	3	2	-r 2 -q -3	4	4	-r 4 -q -5	6	5
	0	2		2	2		2	4		5	5
	3	1		1	2		0	4		4	5
	2	1		0	2		3	3		3	5
	1	1		4	1		6	2		**12**	**8**
	2	**2**		3	1		5	2			
				2	1		4	2	-r 5 -q -4	10	6
				4	**2**		2	2		8	6
							6	**4**		**25**	**10**

NOTE:
Values in red are threshold for ungapped statisitics any value higher than them will be supported.
* megablast does not support this subset.

5.2 Protein scoring marices and their -G/-E values

NCBI BLAST programs support five protein scoring matrices: BLOSUM45, BLOSUM62, BLOSUM80, PAM30, and PAM70. To ensure correct statisitcal evaluation of the found matches, only a limitted number of -G/-E sets. These matrices with their supported -G/-E values are summarized in the table below.

Table 6.2 Supported -G/-E pairs for protein scoring matrices

Matrix	G	E	Matrix	G	E	Matrix	G	E
PAM30	5	2	BLOSUM62	7	2	BLOSUM45	10	3
	6	2		8	2		11	3
	7	2		9	2		12	3
	8	1		10	1		13	3
	9	**1**		**11**	**1**		12	2
	10	1		12	1		13	2
Matrix	**G**	**E**	**Matrix**	**G**	**E**		14	2
PAM70	6	2	BLOSUM80	6	2		**15**	**2**
	7	2		7	2		16	2
	8	2		8	2		15	1
	9	1		9	1		16	1
	10	**1**		**10**	**1**		17	1
	11	1		11	1		18	1
							19	1

NOTE:
Defaults are in red.
To use custom matrix, name it as one of the supported matrix file and place it in the data directory. The statistics of the alignments identified with custom matrix will NOT be reliable.

6. Feedback

For questions and comments on this document and BLAST in general, please send them to:

blast-help@ncbi.nlm.nih.gov

Questions and comments on other NCBI resources should be addressed to:

info@ncbi.nlm.nih.gov

Updated on 12/17/2007 23:44:07