

BLAST[®] Help

Last Updated: October 24, 2023



National Center for Biotechnology Information (US)
Bethesda (MD)

National Center for Biotechnology Information (US), Bethesda (MD)

BLAST is a Registered Trademark of the National Library of Medicine

NLM Citation: BLAST[®] Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2008-.

This manual documents the BLAST (Basic Local Alignment Search Tool) command line applications developed at the National Center for Biotechnology Information (NCBI).

Table of Contents

Introduction to BLAST	1
BLAST Help Manual Overview	3
User Manuals	5
Standalone BLAST Setup for Windows PC	7
Introduction	7
Downloading	7
Installation	7
Configuration	11
Execution and validation	12
Technical Assistance	17
Standalone BLAST Setup for Unix	21
Introduction	21
Downloading	21
Installation	22
Configuration	24
Database Download	24
Execution and validation	25
Technical Assistance	27
BLAST+ Release Notes	29
BLAST+ 2.15.0: October 31, 2023	29
BLAST+ 2.14.0: April 25, 2023	29
BLAST+ 2.13.0: March 11, 2022	30
BLAST+ 2.12.0: June 28, 2021	30
BLAST+ 2.11.0: October 19, 2020	31
BLAST+ 2.10.1: June 8, 2020	31
BLAST+ 2.10.0: December 16, 2019	31
BLAST+ 2.9.0: April 1, 2019	32
BLAST+ 2.8.1: December 13, 2018	32
BLAST+ 2.8.0: March 28, 2018	33
BLAST+ 2.7.1: October 23, 2017	33
BLAST+ 2.6.0: January 09, 2017	33

BLAST+ 2.5.0: September 12, 2016	34
BLAST+ 2.4.0: June 02, 2016	34
BLAST+ 2.3.0: December 21, 2015	35
BLAST+ 2.2.31: May 18, 2015	35
BLAST+ 2.2.30: October 6, 2014	36
BLAST+ 2.2.29: January 3, 2014	36
BLAST+ 2.2.28: March 19, 2013	37
BLAST+ 2.2.27: September 10, 2012	38
BLAST+ 2.2.26: January 31, 2012	38
BLAST+ 2.2.26: March 15, 2011	39
BLAST+ 2.2.24 bug fix release: October 30, 2010	40
BLAST+ 2.2.24: August 2, 2010	40
BLAST+ 2.2.23: Feb 03, 2010	40
BLAST+ 2.2.22 Internal bug fix release: November 02, 2009	40
BLAST+ 2.2.22: Sep 27, 2009	41
BLAST+ 2.2.21: May 27, 2009	41
BLAST+ 2.2.19: November 03, 2008	41
BLAST+ 2.2.18: October 14, 2008	42
BLAST+ 2.2.17 internal release: September 24, 2008	42
BLAST+ 2.2.16 internal release: August 21, 2008	42
BLAST FTP Site	43
Subdirectories under the BLAST FTP directory	43
The <i>"db"</i> subdirectory	43
Contents of the <i>"blast/demo/"</i> subdirectory	48
Contents of the <i>"blast/documents/"</i> subdirectory	49
Contents of the <i>"blast/executables/"</i> subdirectory	49
Contents of the <i>"blast/matrices/"</i> subdirectory	50
Contents of the <i>"blast/temp/"</i> subdirectory	51
Contents of the <i>"blast/WGS_TOOLS/"</i> subdirectory	51
Getting Help	51
BLAST Glossary	53

Introduction to BLAST

BLAST Help Manual Overview

Tom Madden¹

Created: January 28, 2011.

The Basic Local Alignment Search Tool (BLAST) is the most widely used sequence similarity tool. There are versions of BLAST that compare protein queries to protein databases, nucleotide queries to nucleotide databases, as well as versions that translate nucleotide queries or databases in all six frames and compare to protein databases or queries. PSI-BLAST produces a position-specific-scoring-matrix (PSSM) starting with a protein query, and then uses that PSSM to perform further searches. It is also possible to compare a protein or nucleotide query to a database of PSSM's. The NCBI supports a BLAST web page at blast.ncbi.nlm.nih.gov as well as a network service. The NCBI also distributes stand-alone BLAST applications for users who wish to run BLAST on their own machines or with their own databases. This document describes the stand-alone BLAST applications and will concentrate on the latest generation of such applications included in the BLAST+ package.

The first two sections of this document are “quick start” guides that address setting BLAST+ up under UNIX (including LINUX and MacOSX) and Windows. The third section is a more in-depth look at the BLAST+ applications. These documents are currently being revised and may change substantially in coming releases.

User Manuals

Standalone BLAST Setup for Windows PC

Tao Tao, Ph.D.✉¹

Created: May 31, 2010; Updated: August 31, 2020.

Introduction

In addition to providing BLAST sequence alignment services on the web, NCBI also makes these sequence alignment utilities available for download through FTP. This allows BLAST searches to be performed on local platforms against databases downloaded from NCBI or created locally. These utilities run through DOS-like command windows and accept input through text-based command line switches. There is no graphic user interface.

The following tutorial discusses the steps needed to install BLAST+ and a sample NCBI database on PCs running Windows 10 Operating System.

Downloading

The BLAST+ software package is available as self-extracting archives. The archive `ncbi-blast-#.#.#+-win64.exe`, is for PCs running 64-bit Windows operating system. Here, the `"#.#.#"` denotes the current version number of the package. Archives with the same base name and version number are equivalent.

Please note that the archive with the `".tar.gz"` file extension does not have the installer function. The discussion below focuses on archives with the `".exe"` extension.

Steps

Steps to download the package are:

- Point a browser to this FTP directory:
<https://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/>
- Right click on a desired archive and select "Save target as..." from the popup menu
- In the prompt, switch to a desired directory (folder) and click the "Save" button to save the archive to the selected location on the local disk

Examples

These steps for the `"ncbi-blast-2.10.0+-win64.exe"` archive are given in Figure 1a and Figure 1b, where the first two steps are demonstrated by 1a and the last step is demonstrated by 1b.

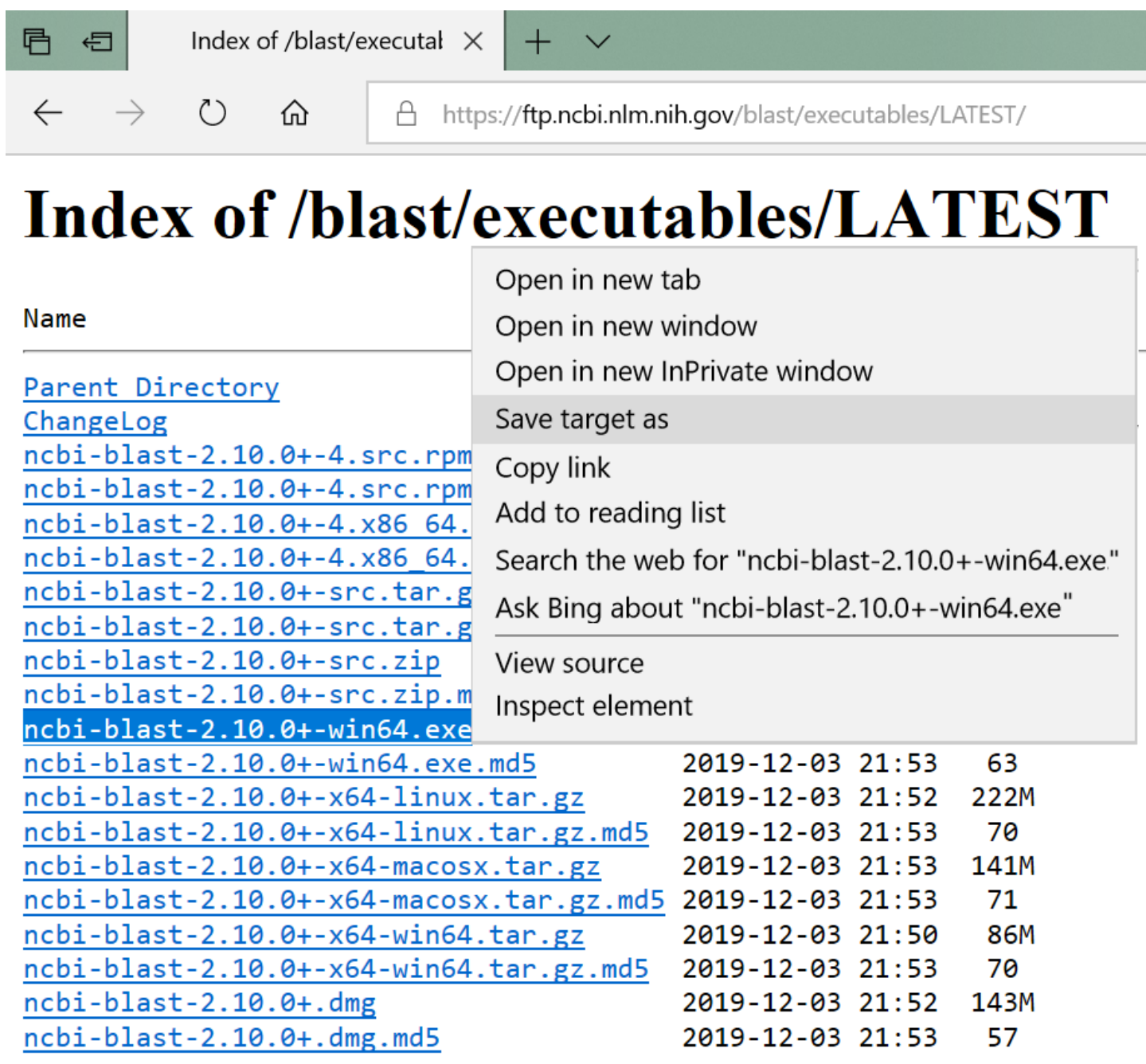
Installation

The BLAST+ archive downloaded above contains a built-in installer. Double click the file to launch the installer, accept the license agreement to specify the install location in a new prompt. In this test case `"C:\Users\taota\Desktop\blast-2.10.0+"` is set as the installation directory (Figure 2). Clicking the "Install" button, the installer will create this directory and install the following:

- a `"doc"` subdirectory with the BLAST+ user manual in PDF format
- an `"uninstaller"` for future removal of the installation, and

Author Affiliation: 1 NCBI; Email: tao@ncbi.nlm.nih.gov.

✉ Corresponding author.



Index of /blast/executables/LATEST

Name

[Parent Directory](#)

[ChangeLog](#)

[ncbi-blast-2.10.0+-4.src.rpm](#)

[ncbi-blast-2.10.0+-4.src.rpm](#)

[ncbi-blast-2.10.0+-4.x86_64.rpm](#)

[ncbi-blast-2.10.0+-4.x86_64.rpm](#)

[ncbi-blast-2.10.0+-src.tar.gz](#)

[ncbi-blast-2.10.0+-src.tar.gz](#)

[ncbi-blast-2.10.0+-src.zip](#)

[ncbi-blast-2.10.0+-src.zip.md5](#)

[ncbi-blast-2.10.0+-win64.exe](#)

[ncbi-blast-2.10.0+-win64.exe.md5](#)

[ncbi-blast-2.10.0+-x64-linux.tar.gz](#)

[ncbi-blast-2.10.0+-x64-linux.tar.gz.md5](#)

[ncbi-blast-2.10.0+-x64-macosx.tar.gz](#)

[ncbi-blast-2.10.0+-x64-macosx.tar.gz.md5](#)

[ncbi-blast-2.10.0+-x64-win64.tar.gz](#)

[ncbi-blast-2.10.0+-x64-win64.tar.gz.md5](#)

[ncbi-blast-2.10.0+.dmg](#)

[ncbi-blast-2.10.0+.dmg.md5](#)

2019-12-03 21:53 63

2019-12-03 21:52 222M

2019-12-03 21:53 70

2019-12-03 21:53 141M

2019-12-03 21:53 71

2019-12-03 21:50 86M

2019-12-03 21:53 70

2019-12-03 21:52 143M

2019-12-03 21:53 57

Open in new tab

Open in new window

Open in new InPrivate window

Save target as

Copy link

Add to reading list

Search the web for "ncbi-blast-2.10.0+-win64.exe"

Ask Bing about "ncbi-blast-2.10.0+-win64.exe"

View source

Inspect element

Figure 1a. Download a blast+ package from NCBI through a web browser: Log on to [ftp://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/](https://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/) and select "Save target as ..." after right-clicking on "ncbi-blast-2.2.29+-win64.exe".

- a "bin" subdirectory with all BLAST programs and accessory utilities

Table 1 sums up programs and utilities provided by the BLAST+ package.

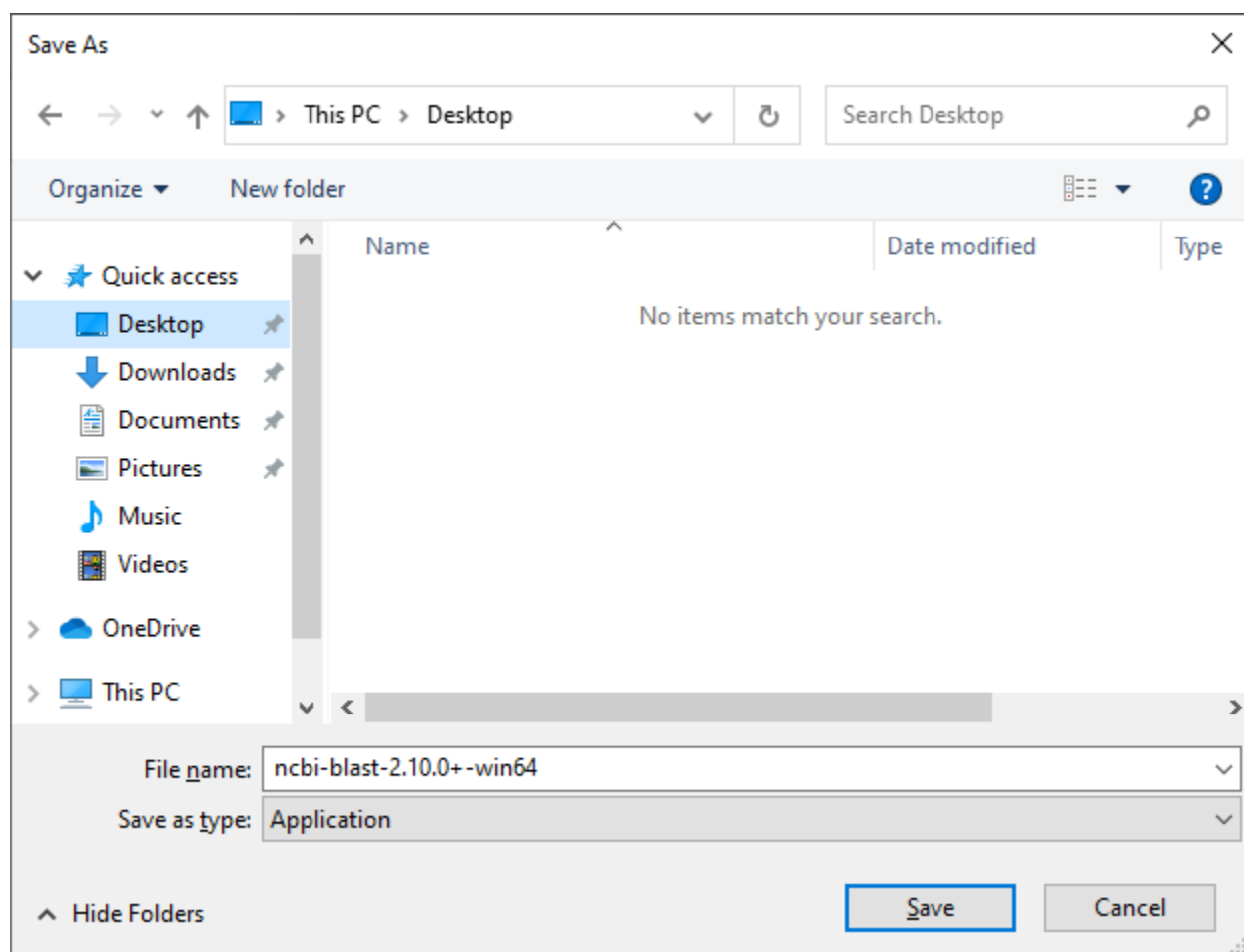


Figure 1b. Download a blast+ package from NCBI through a web browser: Change the location in the subsequent prompt to your own directory under "C:" before saving the archive to a desired location.

Table 1 Programs and utilities contained in the blast+ package

Program	Function
blastdbcheck	Examines random entries in the target BLAST database to confirm its integrity
blastdbcmd	Retrieves sequences or other information from an existing BLAST database. Specific sequence retrieval requires the database be created with <i>-parse_seqids</i> option
blastdb_aliastool	Creates database alias to tie several databases together for example or to specify a subset of sequence in a target database
blastn	Searches FASTA nucleotide queries in the input file against a nucleotide database
blastp	Searches FASTA protein queries in the input file against a protein database
blastx	Searches FASTA nucleotide queries in the input file, dynamically translated in all six frames, against a protein database and returns alignments at the protein level
blast_formatter	Formats a blast result using its assigned request ID (RID) or its saved archive file
convert2blastmask	Converts lowercase masking into makeblastdb readable data
deltablast	Searches a protein query against a protein database, using a more sensitive algorithm by taking conserved domain (delta cdd database required) matches into consideration
dustmasker	Masks the low complexity regions in the input nucleotide sequences

Table 1 continued from previous page.

Program	Function
get_species_taxids.sh	Generates a list of species-level taxids (<i>EntrezDirect</i> installation required) for the input organism name or taxonomic ids above the species-level for use with a version 5 BLAST database in search limit or exclusion
legacy_blast.pl	Converts a legacy blast search command line into blast+ counterpart and execute it
makeblastdb	Formats input FASTA file(s) into a BLAST database
makembindex	Indexes an existing nucleotide database for use with indexed megablast search
makeprofiledb	Creates a conserved domain database from a list of input position specific scoring matrices (scoremats), usually generated by psiblast
psiblast	Finds members of a protein family, identifies proteins distantly related to the query, or builds position specific scoring matrix for the query through iterative rounds of searches
rpsblast	Searches a protein query against a conserved domain database to identify functional domains present in the query
rpstblastn	Searches a nucleotide query, by dynamically translated it in all six-frames first, against a conserved domain database
segmasker	Masks the low complexity regions in input protein sequences
tblastn	Searches a protein query against a nucleotide database dynamically translated in all six frames to return alignment at the protein level
tblastx	Searches a nucleotide query, dynamically translated in all six frames, against a nucleotide database translated in the same manner to return alignment at the protein level
update_blastdb.pl	Automatically downloads and decompresses all volumes of a specified preformatted blast databases from NCBI, AWS, or GCP (default is from NCBI)
windowmasker	Masks repeats found in input nucleotide sequences

Test BLAST database

In addition to BLAST programs and accessory utilities, target database is an indispensable component of a standalone BLAST setup. The common set of pre-formatted NCBI BLAST databases is available as compressed archives from NCBI FTP site. Databases can also be prepared *de novo* from custom FASTA sequences locally using the *makeblastdb* utility. The best way to manage available BLAST databases is to place them in a dedicated directory. A subdirectory named "db" under the "C:\Users\taota\Desktop\blast-2.10.0+" directory is created for this, and its full path is "C:\Users\taota\Desktop\blast-2.10.0+\db".

Similar procedures in Figure 1 can be used to download preformatted BLAST databases from the NCBI ftp site to this dedicated database subdirectory. Here are the steps for downloading the single-volume 16S_ribosomal_RAN database databases:

- Point the browser to <https://ftp.ncbi.nlm.nih.gov/blast/db/>
- Right-click on the 16S_ribosomal_RNA.tar.gz file
- Select "Save target as ..." from the popup menu (menus may differ among browsers)
- When prompted, use the "Save in" to change the directory to "C:\Users\taota\Desktop\blast-2.10.0+\db "

Use WinZip, 7-Zip, or other decompression utility to inflate the compressed archive first, then extract the files from the resulting archive. Note that the above steps download and install a database with a single volume. Large databases, such as nt, are provided as multi-volume sets. Get compressed archives with the same base name (with different ".##" or ".###" volume numbers) when attempting to reconstitute any multi-volume database. The database alias file, such as *nt.nal* or *nr.pal*, ties all volumes together back into the complete database. Also, for

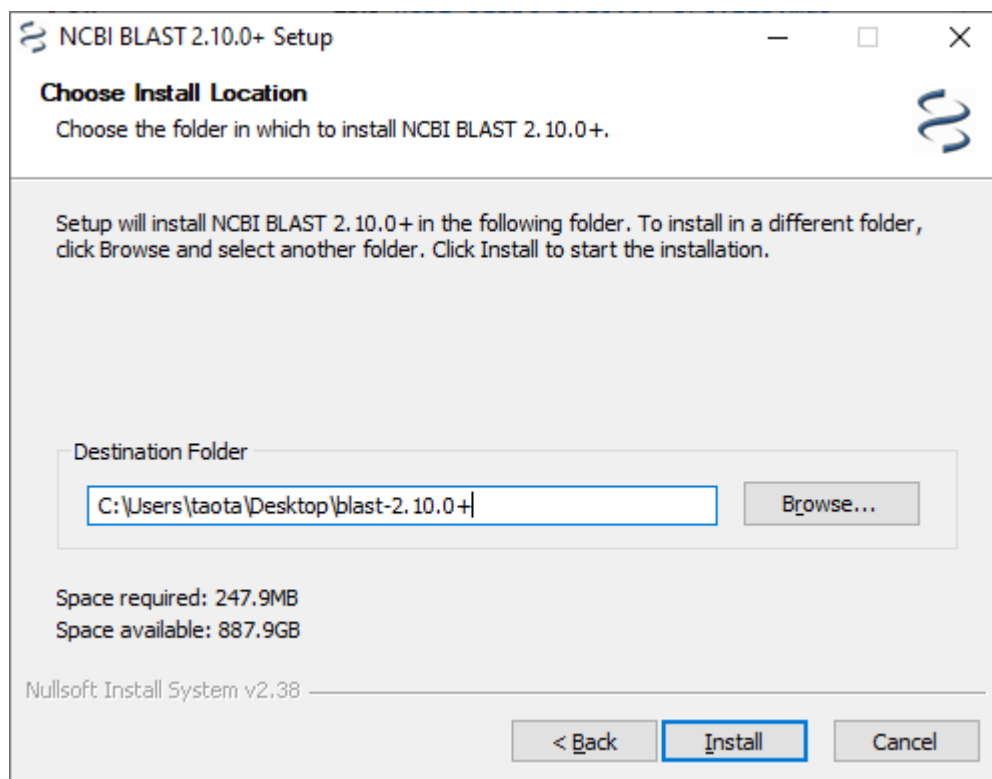


Figure 2. Windows BLAST+ installer dialog box: Use “Browse...” button to change the name and location of blast+ installation.

multi-volume databases, extra files enabling version 5 functionalities are only provided in the first volume. Figure 3 below shows an example inflation/extraction procedure using 7-Zip.

The blast+ package provides a Perl script, *update_blastdb.pl*, to help streamline the downloading of preformatted BLAST databases from NCBI. Downloading through this script requires the installation of the Perl package and execution from the command prompt under the database directory or "C:\users\taota\Desktop\blast-2.10.0+\db" in this example setup. The base command is:

```
perl update_blastdb.pl --passive --decompress base_database_name
```

where "base_database_name" is the name of the target database without the "##.tar.gz" extension (e.g., refseq_rna, or refseq_representative_genomes).

Configuration

Further configuring the PC will help facilitate the execution of blast+ programs and streamline the access of installed databases. This configuration is through information stored in special user environment variables. For 2.10.0 release of the blast+ package, three variables are needed:

- A modified **path** environment variable to indicate the location of installed blast+ programs, with "C:\users\taota\Desktop\blast-2.10.0+\bin\;" prepended to its existing value
- A new **BLASTDB** environment variable as pointer to database location, with "C:\users\taota\Desktop\blast-2.10.0+\db\" as its value
- A new **BLASTDB_LMDB_MAP_SIZE**, with **1000000** as its value (needed to optimize *makeblastdb* operation when creating new database files)

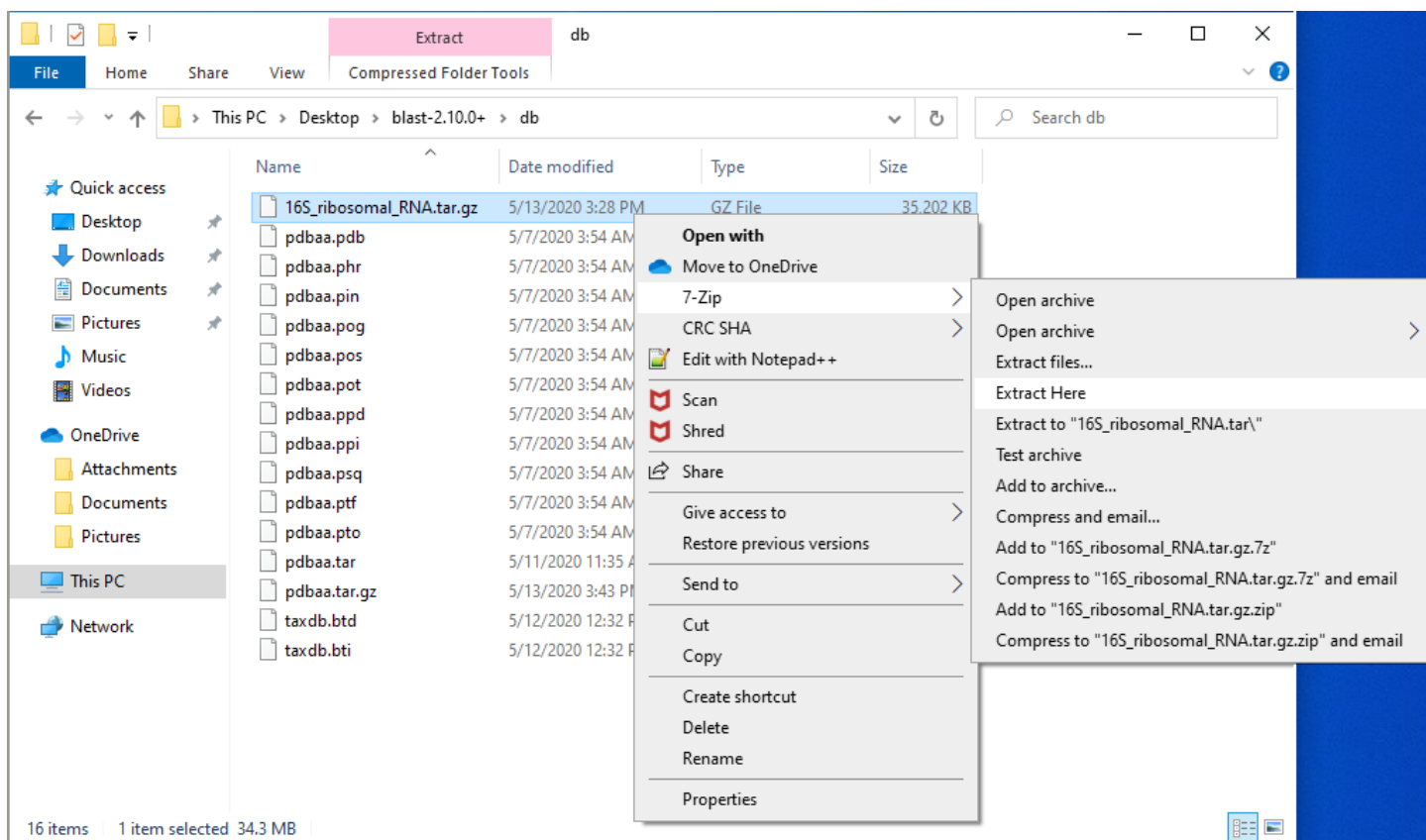


Figure 3. Extract a BLAST database archive using 7-Zip: Right click on the downloaded 16S_ribosomal_RNA.tar.gz archive, select "7-Zip" >> "Extract Here" from the cascading menu to inflate it. Right click on the inflated .tar file and select the same cascading menu to extract individual files from the archive. The screenshot also contains extracted pdbaa database files, the inflated archive (.tar), and the compressed archive (.tar.gz). The taxdb.* accessory files provides scientific name for database entries.

Environment Variables

Here are the steps to create or modify environment variables to configure blast+ installation:

- Use toolbar's search box to search for "edit user environment variables"
- Click the retrieved icon to launch the dialog box
- Click the "New..." button under the "User variable for ..." panel to create a new entry
- Type the variable name, then the absolute path (or other required input value)
- Click "OK" to close the prompts
- Highlight an existing variable, then click "Edit..." to edit its value
- Click "OK" to close the prompt and click "OK" again to exit

Example Screen Shots

Screen shots of these steps are shown, with the first two steps in Figure 4a, and the rest in Figure 4b.

Execution and validation

Standalone blast+ programs do NOT have a graphical user interface (GUI) and must be executed from a command prompt window (CMD). The easiest way to open this prompt is to locate it by searching for "cmd" or "command prompt" using the toolbar's search box as shown in Figures 5.

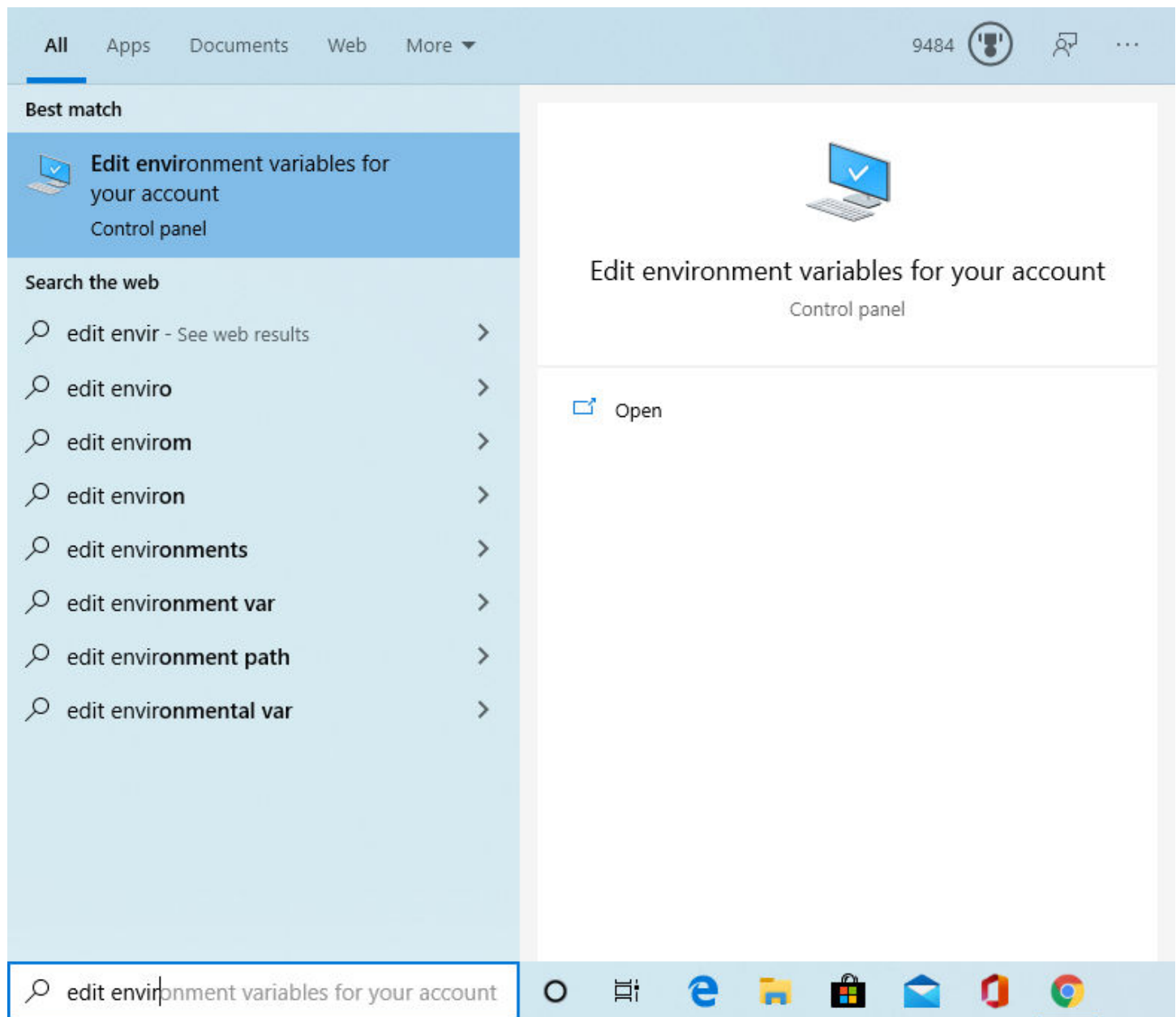


Figure 4a. Configure standalone blast+ using Windows' environment variables: Use toolbar's search box to find "Edit environment variable for your account," then click to launch it.

Example Execution

Test the installation before using this installation for actual work. The test commands groups into three categories:

- Check blast-specific settings using commands
- Call blastdbcmd for database checking and specific sequence retrieval
- Call blastn to run a quick nucleotide blast search
- Get into the /db directory and use update_blastdb.pl to download NCBI databases

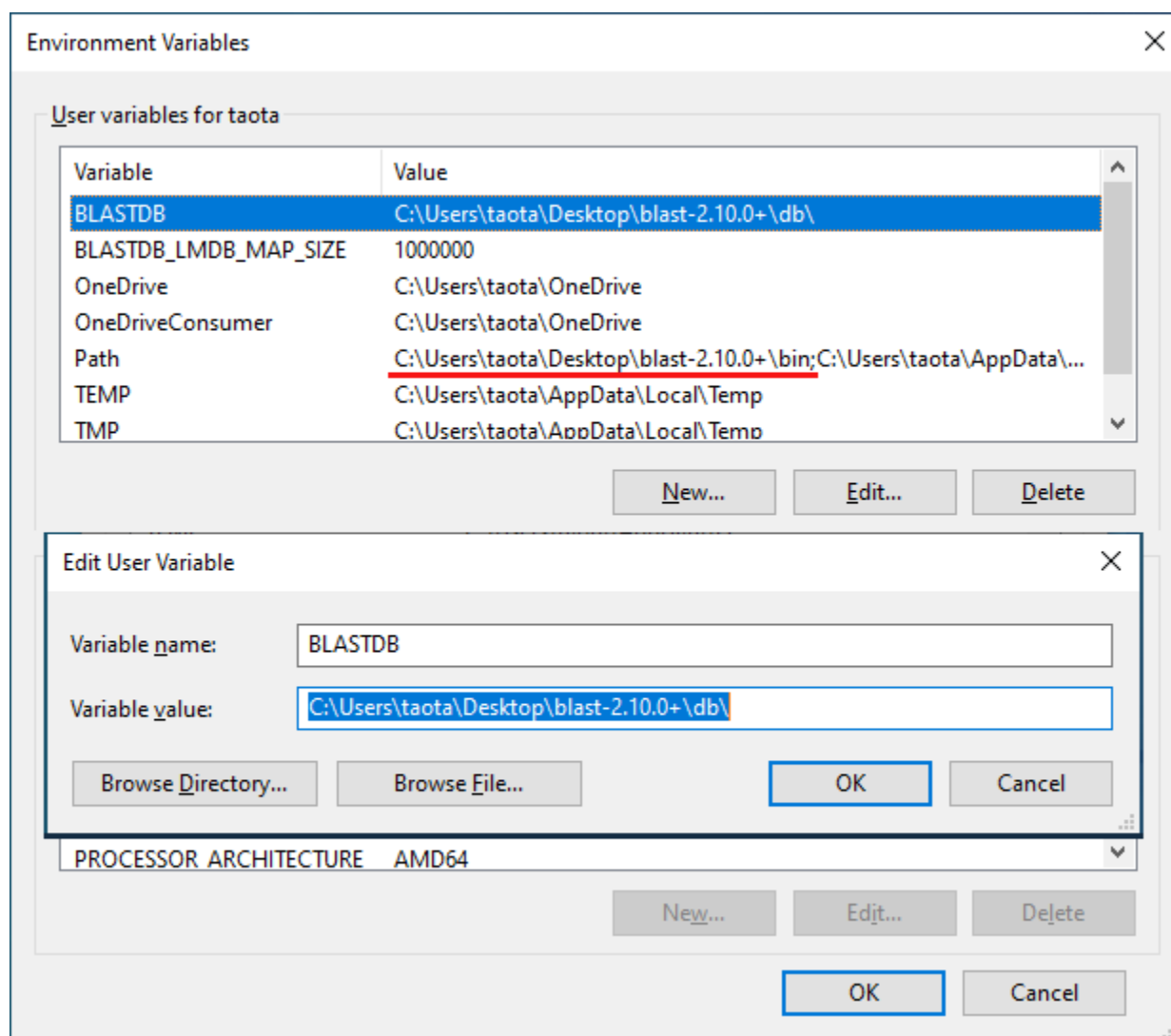


Figure 4b. Configure standalone BLAST using Windows' environment variables: Use the “User variables for taota” section at the top of the dialog box to do the configuration. The two user variables relevant to blast+ are BLASTDB and path. Clicking the “New...” button to create the BLASTDB environment variable (insert) with “C:\Users\taota\Desktop\blast-2.10.0+\bin\” as its value. Click “OK” to save it to the list. Highlight the path variable, click “Edit...” to modify it by prepending a new path (underlined) to it. This addition enables Windows to locate installed blast+ programs. The example also sets up a variable called BLASTDB_LMDB_MAP_SIZE, with a value of 1000000, which is for the optimal function of makeblastdb when making version 5 databases locally.

Detailed commands and their explanation

Technically, there are three components for a blast search, the input query, the target database, and the blast program. The test session below includes a set of representative commands, each testing a specific aspect of this example installation.

First section is on general checking and execution with marked command matching examples in Figure 6A.

Check the configuration by finding the settings of blast-specific environment variables (A).

```
set | find /I "blast"
```

The exact meaning of the command line is (from left to right) to:

- execute the set command (which dumps out all the environment variable settings)

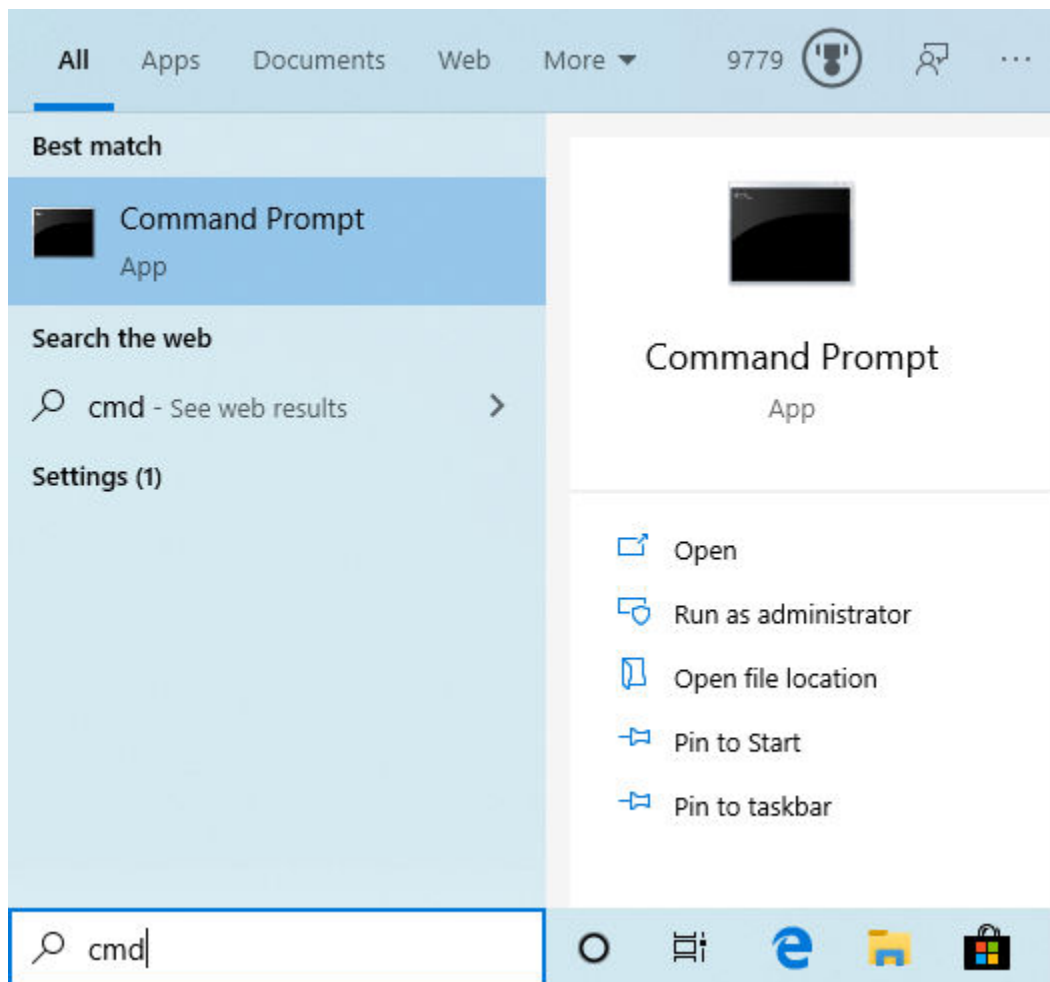


Figure 5. Open a command prompt in Windows 7: Use the toolbar's search box to search for the Command Prompt app. Click the icon to launch it.

- pass the console output ("|" pipe symbol) to next command
- execute find command in case-insensitive mode ("/I") to find lines containing blast in the output

Change the working directory to the installation (B) and check its content.

```
cd "C:\Users\taota\Desktop\blast-2.10.0+"
```

The exact meaning of the command line is to:

- a. execute cd (change directory) command
- b. set target the directory to that given in the argument
- c. follow the above with the dir command (C, not shown) to check directory content

Check the manually downloaded 16S_ribosomal_RNA database with *blastdbcmd* (D):

```
blastdbcmd -db 16S_ribosomal_RNA -info
```

This command instructs the system to:

- execute *blastdbcmd* program
- look for a database call 16S_ribosomal_RNA (first in the BLASTDB specified directory, if failed, the current working directory)
- display the general information available for that database

Retrieve a sequence from this database for use as a test query (E):

```
blastdbcmd -db 16S_ribosomal_RNA -entry NR_025000 > test_16S_query.fa
```

This command instructs the system to:

- execute *blastdbcmd* program
- look for a database named 16S_ribosomal_RNA
- retrieve a sequence with its accession (-entry NR_025000) in default fasta format
- redirect (“>” symbol) the output to a text file name test_16S_query.fa

Check the version of the installation through the blastn program (F):

```
blastn -version
```

This command instructs the system to:

- execute *blastn* program
- display its version to the console

Run a nucleotide search using blastn and the test query sequence (G):

```
blastn -db 16S_ribosomal_RNA -query test_16S_query.fa -outfmt 7 -max_target_seqs 5
```

This command instructs the system to:

- execute *blastn* program
- search against the specified database 16S_ribosomal_RNA
- use sequence(s) in the specified file test_16S_query.fa as query
- ask for tabular output with header (-outfmt 7)
- request only the top 5 hits (-max_target_seqs 5)
- let the results display to the console (without specifying -out file_name)

The following focuses on database manipulation and example session is in Figure 6b.

Change working directory to the db subdirectory (H, not shown), then use update_blastdb.pl to download the preformatted swissprot database (I).

```
perl ../bin/update_blastdb.pl --passive --decompress swissprot
```

This command instructs the system to:

- execute *perl* program (requires a separate installation)
- run the update_blastdb.pl script, which is in the parent directory’s bin subdirectory (../bin/)
- use passive mode (--passive) for FTP
- decompress the downloaded file automatically (--decompress)
- set the requested database to swissprot

Check the extracted database files using dir (J, not shown), then download a multi-volume database refseq_rna using the same base command (K, not shown).

A common need to install standalone blast+ is to search against a custom collection of sequences. For that, the file with custom sequences in FASTA format needs to be converted to a blastable database using *makeblastdb* tool from the blast+ package.

Take advantage of the installed swissprot database to retrieve set of sequences from it and save them to a file (L) for use as example input.

```
blastdbcmd -db swissprot -taxids 9606 -out sp_hs_subset.faa
```

This command instructs the system to:

- execute *blastdbcmd* program
- set the source database to swissprot
- retrieves sequences based on their taxonomic id (-taxids 9606)
- save the output sequence to a file (-out sp_hs_subset.faa)

Call *makeblastdb* to convert the newly created FASTA sequence file into a blastable database (M).

```
makeblastdb -in sp_hs_subset.faa -dbtype prot -parse_seqids -title "demo: swissprot hs  
subset without taxid" -out sp_hs_subset
```

This command instructs the system to:

- execute *makeblastdb* program
- use sp_hs_subset.faa as input
- set the database type as protein
- index sequence ids (for specific sequence retrieval by id)
- create a title using text in the quotes
- rename the output (else the full FASTA file name including file extension will be used)

Check the resulted database using *blastdbcmd* (N, not shown).

Technical Assistance

Questions, feedback, and technical assistance requests should be sent to blast-help at:

```
blast-help@ncbi.nlm.nih.gov
```

Questions on other NCBI resources should be addressed to NCBI Service Desk at:

```
info@ncbi.nlm.nih.gov
```

```

C:\Users\taota>set | find /I "blast"
BLASTDB=C:\Users\taota\Desktop\blast-2.10.0+db\
BLASTDB_LMDB_MAP_SIZE=1000000
Path=C:\Perl64\site\bin;C:\Perl64\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\OpenSSH\;C:\Users\taota\AppData\Local\Programs\Aspera\Aspera Connect\bin\;C:\Users\taota\Desktop\blast-2.10.0+bin;C:\Users\taota\AppData\Local\Microsoft\WindowsApps

C:\Users\taota>cd "c:\Users\taota\Desktop\blast-2.10.0+"
c:\Users\taota\Desktop\blast-2.10.0+>dir
Volume in drive C has no label.
Volume Serial Number is 80CF-52AE

Directory of c:\Users\taota\Desktop\blast-2.10.0+

05/21/2020  03:39 PM    <DIR>          .
05/21/2020  03:39 PM    <DIR>          ..
05/13/2020  03:20 PM    <DIR>          bin
05/19/2020  04:35 PM    <DIR>          db
05/13/2020  03:20 PM    <DIR>          doc
05/21/2020  03:39 PM                1,445 test_16S_query.fa
05/13/2020  03:20 PM        62,507 Uninstall-ncbi-blast-2.10.0+.exe
                2 File(s)          63,952 bytes
                5 Dir(s)      949,088,591,872 bytes free
c:\Users\taota\Desktop\blast-2.10.0+>blastdbcmd -db 16S_ribosomal_RNA -info
Database: 16S_ribosomal_RNA (Bacteria and Archaea type strains)
        21,030 sequences; 30,580,655 total bases

Date: May 12, 2020  12:31 PM    Longest sequence: 3,600 bases

BLASTDB Version: 5

Volumes:
        C:\Users\taota\Desktop\blast-2.10.0+db\16S_ribosomal_RNA
c:\Users\taota\Desktop\blast-2.10.0+>blastdbcmd -db 16S_ribosomal_RNA -entry NR_025000 > test_16S_query.fa
c:\Users\taota\Desktop\blast-2.10.0+>blastn -version
blastn: 2.10.0+
Package: blast 2.10.0, build Dec  3 2019 18:01:22
c:\Users\taota\Desktop\blast-2.10.0+>blastn -db 16S_ribosomal_RNA -query test_16S_query.fa -outfmt 7 -max_target_seqs 5
# BLASTN 2.10.0+
# Query: NR_025000.1 Mycobacterium kubicar strain CDC 941078 16S ribosomal RNA, partial sequence
# Database: 16S_ribosomal_RNA
# Fields: query acc.ver, subject acc.ver, % identity, alignment length, mismatches, gap opens, q. start, q. end, s. start, s. end, evalue, bit score
# 5 hits found
NR_025000.1    NR_025000.1    100.000 1321    0      0      1      1321    1      1321    0.0    2440
NR_025000.1    NR_028940.1    99.243 1321    9      1      1      1321    30     1349    0.0    2383
NR_025000.1    NR_125568.1    98.941 1322    12     2      1      1321    17     1337    0.0    2362
NR_025000.1    NR_113062.1    98.716 1324    11     6      1      1321    17     1337    0.0    2346
NR_025000.1    NR_117227.1    98.716 1324    11     6      1      1321    37     1357    0.0    2346
# BLAST processed 1 queries

```

Figure 6a. The output of a work session testing the blast+ installation: The input commands are underlined with their function explained by inserted text.


```

C:\Users\taota\Desktop\blast-2.10.0+>cd db
C:\Users\taota\Desktop\blast-2.10.0+>perl ..\bin\update_blastdb.pl --passive --decompress swissprot
Connected to NCBI
Downloading swissprot.tar.gz... [OK]
Decompressing swissprot.tar.gz ... [OK]

c:\Users\taota\Desktop\blast-2.10.0+>dir swissprot*
Volume in drive C has no label.
Volume Serial Number is 80CF-52AE

Directory of c:\Users\taota\Desktop\blast-2.10.0+>db

05/17/2020  04:40 AM           24,043,520 swissprot.pdb
05/17/2020  04:40 AM          119,868,542 swissprot.phr
05/17/2020  04:40 AM           3,792,240 swissprot.pin
05/17/2020  04:40 AM           1,896,088 swissprot.pog
05/17/2020  04:40 AM           8,857,626 swissprot.pos
05/17/2020  04:40 AM           6,037,404 swissprot.pot
05/17/2020  04:40 AM           3,792,112 swissprot.ppd
05/17/2020  04:40 AM            14,860 swissprot.ppi
05/17/2020  04:40 AM          179,714,384 swissprot.psq
05/17/2020  04:40 AM           471,040 swissprot.ptf
05/17/2020  04:40 AM          2,300,788 swissprot.pto
05/21/2020  03:48 PM              51 swissprot.tar.gz.md5
               12 File(s)      350,788,655 bytes
               0 Dir(s)      948,720,263,168 bytes free

c:\Users\taota\Desktop\blast-2.10.0+>perl ..\bin\update_blastdb.pl --passive --decompress refseq_rna
Connected to NCBI
Downloading refseq_rna (7 volumes) ...
Downloading refseq_rna.00.tar.gz...[OK]
Downloading refseq_rna.01.tar.gz...[OK]
...
Decompressing refseq_rna.00.tar.gz...[OK]
Decompressing refseq_rna.01.tar.gz...[OK]
...

c:\Users\taota\Desktop\blast-2.10.0+>blastdbcmd -db swissprot -taxids 9606 -out sp_hs_subset.faa

c:\Users\taota\Desktop\blast-2.10.0+>makeblastdb -in sp_hs_subset.faa -dbtype prot -parse_seqids -title "demo: swissprot hs subset without taxid" -out sp_hs_subset

Building a new DB, current time: 05/26/2020 23:39:07
New DB name:  c:\Users\taota\Desktop\blast-2.10.0+>db\sp_hs_subset
New DB title: demo: swissprot hs subset without taxid
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 20379 sequences in 1.12771 seconds.

c:\Users\taota\Desktop\blast-2.10.0+>blastdbcmd -db sp_hs_subset -info
Database: demo: swissprot hs subset without taxid
        20,379 sequences; 11,363,421 total residues

Date: May 26, 2020  11:39 PM    Longest sequence: 34,350 residues

BLASTDB Version: 5

Volumes:
        c:\Users\taota\Desktop\blast-2.10.0+>db\sp_hs_subset

c:\Users\taota\Desktop\blast-2.10.0+>db>

```

H. Change working directory to /db subdirectory

I. Use update_blastdb.pl script to download swissprot preformatted blast database, automatically extract it

J. Check the resulted data base files after download

K. For multi-volume databases, this script automatically downloads and decompress them


L. Dump out a subset of FASTA sequences from the installed swissport database by the encoded taxid

M. Use makeblastdb to format the output FASTA file into a blastable database

N. Check the newly created database with blastdbcmd

Figure 6b. The output of a work session focusing on database management and creation: The input commands are underlined with their function explained by inserted text.

Standalone BLAST Setup for Unix

Tao Tao, Ph.D. 

Created: May 31, 2010; Updated: August 31, 2020.

Introduction

NCBI provides command line standalone BLAST+ programs (based on the NCBI C++ toolkit) as a single compressed package. The package is available for the Linux, Mac OSX, and Windows platforms at:

<https://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/>

The archives for Linux and Mac OSX are available as gzip-compressed tar files named using the following convention:

`ncbi-blast-#.#.#+-CHIP-OS.tar.gz`

Here, the #.#.# represents the version number of the current release, CHIP indicates the chipset, and OS indicates the operating system. Equivalent .rpm and .dmg files for Linux and Mac OSX are also available. These archives and their target platforms are listed in the table below.

Table 1 Executable BLAST+ package available from NCBI

Archive Name	Content	Chipset	OS	File Type
ncbi-blast-#.#.#+-src.tar.gz	Source code	N/A	N/A	gzip'd tar archive
ncbi-blast-#.#.#+-src.zip	Source code	N/A	N/A	Zipped
ncbi-blast-#.#.#+-win64.exe	Programs	x64	64-bit Windows	Windows installer
ncbi-blast-#.#.#+-x64-win64.tar.gz	Programs	x64	64-bit Windows	gzip'd tar archive
ncbi-blast-#.#.#+-x64-linux.tar.gz	Programs	x64	64-bit Linux	gzip'd tar archive
ncbi-blast-#.#.#+-x86_64.rpm	Programs	x64	64-bit Linux	Linux RPM package
ncbi-blast-#.#.#+-x64-maxosx.tar.gz	Programs	x64	64-bit MacOSX	gzip'd tar archive
ncbi-blast-#.#.#+-dmg	Programs	x64	64-bit MacOSX	Disk image

The installation process from the disk image (.dmg) for Mac OSX and the Red Hat Package Manager (.rpm) for Linux requires administrative privileges and will not be discussed here. More information is available here: <http://www.ncbi.nlm.nih.gov/books/NBK279671/>.

Downloading

The BLAST+ packages for various platforms can be downloaded through anonymous ftp using an ftp client, wget, curl, or a web browser. The example working session below demonstrates an ftp download process using an ftp client in a Linux environment. In Mac OSX, a similar command line interface is available through the Terminal utility, which is generally under the Utilities folder.

Steps

Steps to download the package through a browser are described below.

Author Affiliation: 1 NCBI; Email: tao@ncbi.nlm.nih.gov.

 Corresponding author.

- Point a browser to this ftp directory:
<https://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/>
- Right click on a desired archive and select "Save link as..." from the popup menu
- In the prompt, switch to a desired directory (folder) and click the "Save" button to save the selected archive to a desired location on the local disk

Example

Downloading through an ftp client is shown below with input commands in bold.

```
$ ncftp ftp.ncbi.nlm.nih.gov
NcFTP 3.2.5 (Feb 02, 2011) by Mike Gleason (http://www.NcFTP.com/contact/).
Connecting to ftp.ncbi.nlm.nih.gov...

This warning banner provides privacy and security notices consistent with
applicable federal laws, directives, and other federal guidance for accessing
this Government system, which includes all devices/storage media attached to
this system. This system is provided for Government-authorized use only.
Unauthorized or improper use of this system is prohibited and may result in
disciplinary action and/or civil and criminal penalties. At any time, and for
any lawful Government purpose, the government may monitor, record, and audit
your system usage and/or intercept, search and seize any communication or data
transiting or stored on this system. Therefore, you have no reasonable
expectation of privacy. Any communication or data transiting or stored on this
system may be disclosed or used for any lawful Government purpose.

FTP Server ready.
Logging in...
Anonymous access granted, restrictions apply
Logged in to ftp.ncbi.nlm.nih.gov.
ncftp / > cd /blast/executables/LATEST/    << change directory to the LATEST
ncftp /blast/executables/LATEST > bin      << set transfer mode to binary
ncftp /blast/executables/LATEST > get ncbi-blast-2.10.1+-x64-linux.tar.gz
ncbi-blast-2.10.1+-x64-linux.tar.gz:                224.64 MB    37.27 MB/s
ncftp /blast/executables/LATEST > bye
$
```

For platforms lacking a precompiled BLAST+ package, users will need to compile from the BLAST source code. The source code archive, "ncbi-blast-#.##+-src" in either zip or gzipped tar format, is available from the same LATEST ftp directory. Instructions on compilation are available online in the BLAST+ user manual: http://www.ncbi.nlm.nih.gov/books/NBK279671/#_introduction_Source_tarball, familiarity with compilers is assumed. Questions and feedback on source code compilation should be addressed to:

toolbox@ncbi.nlm.nih.gov

Installation

To install, simply extract the downloaded package after placing it under a desired directory. This can be accomplished by a single tar command, or a combination of gunzip and tar commands.

```
$ tar xzvpf ncbi-blast-2.10.1+-x64-linux.tar.gz
```

or

```
$ gunzip -d ncbi-blast-2.10.1+-x64-linux.tar.gz
$ tar xvpf ncbi-blast-2.10.1+-x64-linux.tar
```

Successful execution of the above commands installs the package and generates a **new ncbi-blast-2.10.1+** directory under the working directory selected. This new directory contains the bin and doc subdirectories, as well as a set of informational files. The bin subdirectory contains the programs listed below.

Table 2 Programs contained in BLAST+ package

Category	Program	Function
NCBI database downloading tool	update_blastdb.pl	Downloads preformatted blast databases from NCBI
Local database manipulation tools	makeblastdb	Formats input FASTA file(s) into a BLAST database
	makembindex	Indexes an existing nucleotide database for use with megablast for indexed search
	makeprofiledb	Creates a conserved domain database from a list of input position specific scoring matrix (scoremats) generated by psiblast
	dustmasker	Masks the low complexity regions in the input nucleotide sequences, mostly for use in database preparation
	windowmasker	Masks repeats found in input nucleotide sequences
	segmasker	Masks the low complexity regions in input protein sequences, mostly for use in database preparation
	convert2blastmask	Converts lowercase masking into makeblastdb readable data
	blastdb_aliastool	Creates database alias (to tie volumes together, for example)
	blastdbcheck	Checks the integrity of a BLAST database
	blastdbcmd	Retrieves sequences or other information from a BLAST database
	Cleanup-blastdb-volumes.py	Python script to clean up blast database volumes and remove unreferenced volumes [use it with caution and at your own risk!]
Core blast search programs	blastn	Searches a nucleotide query against a nucleotide database
	blastp	Searches a protein query against a protein database
	blastx	Searches a nucleotide query, dynamically translated in all six frames, against a protein database
	tblastn	Searches a protein query against a nucleotide database dynamically translated in all six frames
	tblastx	Searches a nucleotide query, dynamically translated in all six frames, against a nucleotide database similarly translated
Specialized protein blast search programs	deltablast	Searches a protein query against a protein database, using a more sensitive algorithm
	psiblast	Finds members of a protein family, identifies proteins distantly related to the query, or builds position specific scoring matrix for the query
Conserved domain blast search programs	rpsblast	Searches a protein against a conserved domain database to identify functional domains present in the query
	rpstblastn	Searches a nucleotide query, by dynamically translating it in all six-frames first, against a conserved domain database
Command line translator	legacy_blast.pl	Converts a legacy blast search command line into blast+ counterpart and execute it
Result formatting tool	blast_formatter	Formats a blast result using its assigned request ID (RID) or its saved archive

Configuration

Using the BLAST+ package installed above without configuration will be cumbersome – it requires the installation path to be prefixed to the program and database calls since the system does not know where to look for the installed program and the specified database. To streamline BLAST searches, two environment variables, PATH and BLASTDB, need to be modified and created, respectively, to point to the corresponding directories.

Under bash, the following command appends the path to the new BLAST bin directory to the existing PATH setting:

```
$ export PATH=$PATH:$HOME/ncbi-blast-2.10.1+/bin
```

The equivalent command under csh is:

```
$ setenv PATH ${PATH}:/home/tao/ncbi-blast-2.10.1+/bin
```

The modified \$PATH can be examined using echo (added portion underlined):

```
$ echo $PATH
/usr/X11R6/bin:/usr/bin:/bin:/usr/local/bin:/opt/local/bin:/home/tao/ncbi-blast-2.10.1+/bin
```

To manage available BLAST databases, create a directory to store them:

```
$ mkdir $HOME/blastdb
```

Use the approaches described above for PATH to set the BLASTDB value under bash:

```
$ export BLASTDB=$HOME/blastdb
```

Or under csh to create it anew:

```
$ set BLASTDB=$HOME/blastdb
```

A better approach is to have the system automatically set these variables upon login, by modifying the *.bash_profile* or *.cshrc* file.

Once they are set, the system knows where to call BLAST programs, and the invoked program will know where to look for the database files. Note that with BLASTDB unspecified, BLAST+ programs only search the working directory, i.e. the directory where BLAST command is issued. For more details about configuring BLAST+, please see <http://www.ncbi.nlm.nih.gov/books/NBK279695/>.

Database Download

BLAST database is a key component of any BLAST search. To fully test the BLAST+ package, a functional database is needed. The following working session demonstrates the process of downloading and installing a single-volume database named 16S_ribosomal_RNA from NCBI, using the *update_blastdb.pl* script included in the /bin directory.

```
$ cd $HOME/blastdb

$ perl ../bin/update_blastdb.pl --passive --decompress 16S_ribosomal_RNA
Connected to NCBI
Downloading 16S_ribosomal_RNA.tar.gz... [OK]
Decompressing 16S_ribosomal_RNA.tar.gz ... [OK]
$ ls -l
total 172388
-rw-r--r-- 1 tao sdesk 1142784 Jun 6 12:05 16S_ribosomal_RNA.ndb
-rw-r--r-- 1 tao sdesk 3326945 Jun 6 12:05 16S_ribosomal_RNA.nhr
-rw-r--r-- 1 tao sdesk 252508 Jun 6 12:05 16S_ribosomal_RNA.nin
-rw-r--r-- 1 tao sdesk 170664 Jun 6 12:05 16S_ribosomal_RNA.nnd
-rw-r--r-- 1 tao sdesk 716 Jun 6 12:05 16S_ribosomal_RNA.nni
-rw-r--r-- 1 tao sdesk 84152 Jun 6 12:05 16S_ribosomal_RNA.nog
-rw-r--r-- 1 tao sdesk 424244 Jun 6 12:05 16S_ribosomal_RNA.nos
-rw-r--r-- 1 tao sdesk 252764 Jun 6 12:05 16S_ribosomal_RNA.not
-rw-r--r-- 1 tao sdesk 7761701 Jun 6 12:05 16S_ribosomal_RNA.nsq
-rw-r--r-- 1 tao sdesk 548864 Jun 6 12:05 16S_ribosomal_RNA.ntf
-rw-r--r-- 1 tao sdesk 148812 Jun 6 12:05 16S_ribosomal_RNA.nto
-rw-r--r-- 1 tao sdesk 59 Jun 11 12:14 16S_ribosomal_RNA.tar.gz.md5
-rw-r--r-- 1 tao sdesk 146879591 Jun 6 12:05 taxdb.btd
-rw-r--r-- 1 tao sdesk 15506928 Jun 6 12:05 taxdb.bti
$
```

For databases with multiple volumes, update_blastdb.pl will automatically get all of them. For databases already installed locally, update_blastdb.pl will compare it against that on the NCBI ftp site to determine if refreshing is needed.

```
$ perl ../bin/update_blastdb.pl --passive --decompress 16S_ribosomal_RNA
Connected to NCBI
The contents of 16S_ribosomal_RNA.tar.gz are up to date in your system.
$
```

Execution and validation

With the above configuration, BLAST programs installed under the "ncbi-blast-2.10.1+/bin" directory can be invoked by name from any directory. Type the command "blastn -help" (without quotes) should display the program parameters of blastn in the console as shown below.

```
$ blastn -help
USAGE
blastn [-h] [-help] [-import_search_strategy filename]
        [-export_search_strategy filename] [-task task_name] [-db database_name]
        [-dbsize num_letters] [-gilist filename] [-seqidlist filename]
        [-negative_gilist filename] [-entrez_query entrez_query]
        [-db_soft_mask filtering_algorithm] [-db_hard_mask filtering_algorithm]
        [-subject subject_input_file] [-subject_loc range] [-query input_file]
        [-out output_file] [-eval value] [-word_size int_value]
        [-gapopen open_penalty] [-gapextend extend_penalty]
        [-perc_identity float_value] [-xdrop_ungap float_value]
        [-xdrop_gap float_value] [-xdrop_gap_final float_value]
        [-searchsp int_value] [-max_hsps int_value] [-sum_statistics]
        [-penalty penalty] [-reward reward] [-no_greedy]
        [-min_raw_gapped_score int_value] [-template_type type]
        [-template_length int_value] [-dust DUST_options]
        [-filtering_db filtering_database]
        [-window_masker_taxid window_masker_taxid]
        [-window_masker_db window_masker_db] [-soft_masking soft_masking]
```

```

[-ungapped] [-culling_limit int_value] [-best_hit_overhang float_value]
[-best_hit_score_edge float_value] [-window_size int_value]
[-off_diagonal_range int_value] [-use_index boolean] [-index_name string]
[-lcase_masking] [-query_loc range] [-strand strand] [-parse_deflines]
[-outfmt format] [-show_gis] [-num_descriptions int_value]
[-num_alignments int_value] [-html] [-max_target_seqs num_sequences]
[-num_threads int_value] [-remote] [-version]

```

DESCRIPTION

Nucleotide-Nucleotide BLAST 2.2.29+

OPTIONAL ARGUMENTS

```

-h
    Print USAGE and DESCRIPTION; ignore all other parameters
-help
    Print USAGE, DESCRIPTION and ARGUMENTS; ignore all other parameters
-version
    Print version number; ignore other arguments

*** Input query options
-query <File_In>
    Input file name
    Default = '-'
-query_loc <String>
    Location on the query sequence in 1-based offsets (Format: start-stop)
-strand <String, 'both', 'minus', 'plus'>
    Query strand(s) to search against database/subject
    Default = 'both'
...

```

For installation without \$PATH modification, prefix the path to the program. For example, to execute the same command from /home/tao directory, use the following command instead, where the "./" prefix denotes the current working directory:

```
/home/tao $ ./ncbi-blast-2.10.1+/bin/blastn -help
```

Example Execution

The real test of this installation should be example searches. The working session shown below performs the following task:

- Call `blastdbcmd` to extract the sequence of NR_025000 from the installed database (16S_ribosomal_RNA) to a text file (16S_query.fa)
- Run a `blastn` search using the sequence in `test_query.fa` as query against the 16S_ribosomal_RNA database
- With extra custom settings of using `blastn` algorithm (`-task blastn`), without filter (`-dust no`), requesting custom tabular output (`-outfmt "7 delim=.` Etc), and asking only for the top 5 hits (`-max_target_seqs 5`)
- Since no `-out` is specified, the result is displayed in the console (starting from the # initialed line)

```
$ blastdbcmd -db 16S_ribosomal_RNA -entry nr_025000 -out 16S_query.fa
```

```

$ blastn -db 16S_ribosomal_RNA -query 16S_query.fa -task blastn -dust
no -outfmt "7 delim=, qacc sacc evalue bitscore qcovus pident" -max_target_seqs 5

```



```
# BLASTN 2.10.1+

# Query: NR_025000.1 Mycobacterium kansasii strain CDC 941078 16S
ribosomal RNA, partial sequence

# Database: 16S_ribosomal_RNA

# Fields: query acc., subject acc., evalue, bit score, % query coverage
per uniq subject, % identity

# 5 hits found

NR_025000.1,NR_025000,0.0,2383,100,100.000

NR_025000.1,NR_028940,0.0,2334,100,99.243

NR_025000.1,NR_125568,0.0,2320,100,98.940

NR_025000.1,NR_118110,0.0,2302,100,98.637

NR_025000.1,NR_117220,0.0,2302,100,98.637

# BLAST processed 1 queries

$
```

Note that the command lines and output wrap around.

Technical Assistance

Questions, feedback, and technical assistance requests should be sent to blast-help at:

blast-help@ncbi.nlm.nih.gov

Questions on other NCBI resources should be addressed to NCBI User Services at:

info@ncbi.nlm.nih.gov

BLAST+ Release Notes

Christiam Camacho¹ and Tom Madden²

Created: March 12, 2013; Updated: October 24, 2023.

BLAST+ 2.15.0: October 31, 2023

New features

- You can limit your search by a non-leaf taxID (e.g., all bacteria) without running a helper script.
- BLAST now selects the best threading mode (i.e., sets `-mt_mode` option) based on your query and database for BLASTN, BLASTP, BLASTX, and TBLASTN.

Improvements

- Optimized batch size for `blastx-fast` for smaller databases (e.g., Swiss-Prot) and many queries to make it 7% faster.
- Improved the error message shown if the LMDB file (part of BLAST database) cannot be opened (mentions file).
- `update_blastdb.pl` will use at most 25% of the CPUs on a machine when downloading from the cloud (adjustable with an option).

Bug fixes

- Fixed a megaBLAST crash in two-hit mode.
- Fixed a TBLASTN crash with very short database sequences and a taxonomy limit.

BLAST+ 2.14.0: April 25, 2023

New features

- BLASTP, BLASTX, and TBLASTN are now quicker when you use the "fast" tasks with a command-line argument like `"-task blastp-fast"`. You can search the non-redundant protein database(nr) over 20% faster than before. Smaller databases like `swissprot` or `pdbea` will be 2-3 times faster.

Improvements

- A FASTA file that cannot be validated by `makeblastdb` is now a warning and not an error.
- `Update_blastdb.pl` uses `curl` on a mac to download BLASTDB files instead of a PERL module.
- An error is issued if the incompatible `-remote` and `-in_pssm` options are used together.
- A formatting pre-fetch error is no longer fatal, and BLAST will try to format the report.

Bug fixes

- Fixed an issue with a space before the database name that crashed the `rpsblast` executable.
- Fixed a multi-threading consistency issue with the `-use_sw_tback` option.
- Fixed an issue that crashed the `blastn` executable.
- Fixed an issue sorting results with composition-based statistics.
- Fixed an exit code issue with `update_blast.pl`

- Corrected the expect value setting for blastn-short task
- Fixed an issue with sstrand in tabular formats with the tblastn executable.

BLAST+ 2.13.0: March 11, 2022

New features

- Blastn_vdb and tblastn_vdb included in the 2.13.0 release.
- Makeblastdb now produces a (JSON) metadata file about the database. This makes BLAST databases more Findable in the FAIR sense. See [here](#) for details.

Improvements

- TBLASTN can now handle database sequences up to 2 billion bases (was 1 billion)
- Makeblastdb default volume size is now 3 billion bases (was 1 billion)
- Dustmasker has a new option to replace low complexity regions with N's (hard masking)
- Makeblastdb will issue an error message and exit if it encounters a sequence longer than the maximum supported size (2,147,483,647 letters).

Bug fixes

- Rare problem with mutex that caused BLAST to crash.
- Memory leaks.

BLAST+ 2.12.0: June 28, 2021

For this release, we have performed a major restructuring of the module that reads the BLAST databases. For multithreaded searches, these changes reduce the number of mutex calls, result in the use of fewer file pointers, and reduce the number of calls to memory map. These changes also allow us to support a different threading model ("threading by query") that can be more efficient in some situations. See <https://www.ncbi.nlm.nih.gov/books/NBK571452/> for more information.

NOTE: The NCBI is preparing to use a larger numerical range for its GI identifier. This release provides full support for these GI's that will appear in nucleotide databases later this year.

New features

- Threading by query batch (for BLASTN, BLASTP, BLASTX, RPSBLAST, and RPSTBLASTN) may more efficiently BLAST large numbers of queries, especially if the database is small or the search is limited by taxid. Use "-mt_mode 1" to enable this option.
- Makeblastdb requires less virtual memory for smaller databases.
- Makeprofiledb creates multiple volumes for a CDD database, which allows RPSBLAST to handle a larger number of records. The number of SMP files included in a volume can be controlled with the new -new_smp_vol option.
- update_blastdb.pl now supports the "-showall pretty" option for databases hosted at the NCBI.
- update_blastdb.pl now reports the database timestamp in ISO8601 format.

Bug fixes

- Fixed phiblast core dump when -subject option is used.
- Fixed memory leak in setup procedures.

BLAST+ 2.11.0: October 19, 2020

New features

- Usage reporting - Help improve BLAST by sharing limited information about your search. Details on the information collected, how it is used, and how to opt-out at <https://www.ncbi.nlm.nih.gov/books/NBK309243>
- Threading by query batch for rpsblast/rpstblast can BLAST large numbers of queries faster. For large numbers of queries, use the -mt option to more efficiently multi-thread the search.

Bug fixes

- Fix slowdown in TBLASTN searches run without composition-based statistics on long database sequences.
- Remove necessity of a network connection for blast_formatter. This also speeds up blast_formatter if the database can be found locally.
- A core dump for RPSBLAST and RPSTBLASTN has been fixed.
- Makeblastdb for windows has been fixed to not require as much virtual memory and to not produce overly large LMDB files.

BLAST+ 2.10.1: June 8, 2020

Bug fixes

- Fix for TBLASTN Multi-Threading bug.

BLAST+ 2.10.0: December 16, 2019

New features

- Enhancements to composition-based statistics to ensure the consistency of matches if fewer than the default number of matches is selected. Read about the details in the “[Outline of the BLAST process](#)” section of the BLAST+ user manual appendix.
- Adaptive composition-based statistics may process more sequences in the CBS stage of BLAST if many matches have a similar score, increasing the likelihood of finding novel results. To enable: set the environment variable ADAPTIVE_CBS to 1. This is an experimental feature and your feedback is welcome.
- Default BLAST database version changes:
- makeblastdb generates BLAST databases in version 5 format.
- New script to clean up BLAST database volumes (cleanup-blastdb-volumes.py).
- Add support for genetic code 33 for blastx and rpstblastn.

Improvements

- Better error messages for -taxids argument.
- Consistent error reporting in [get_species_taxids.sh](#) to standard error.

Bug fixes

- Restore sum statistics (-sum_stat parameter) for BLASTN.
- Fix Blast-archive generation/ingestion when subject_bsthits flag is used.

- Fix problem with empty lines in files provided to the taxidlist argument.
- Fix blastdb_aliastool input file size overflow problem.
- Fix blastdb_aliastool problem in Windows with binary GI list files.
- Fix search failures using -remote option in BLAST+ 2.9.0.
- Fix reading from standard input in Windows.
- Fix missing space in descriptions define.
- Fix HTML BLAST report to include version in accession anchors.
- Fix segmentation fault in tabular BLAST output format when sequences have no define.
- Fix to prevent generation of local Seq-IDs in Seq-align output format when accessions are available.
- Fix blast_formatter output when searches are limited by taxonomy.

BLAST+ 2.9.0: April 1, 2019

New features

- Support for PDB biopolymer chain identifiers up to four-characters long in BLASTDB version 5 (not supported in BLASTDB version 4).
- Configurable output separator for tabular and CSV output formats (see [manual entry](#)).

Improvements

- Better error messages in [get_species_taxids.sh](#).
- Fix memory leaks in BLAST libraries and unit tests.

Bug fixes

- Fix taxID filtering combined with mask-based alias BLAST databases.
- Fix ordering of sequence IDs in BLAST report.

BLAST+ 2.8.1: December 13, 2018

NOTE: First production release to support the new [BLAST](#) database version (BLASTDBv5). This is a taxonomically aware version of the BLAST database. See notes at <https://ftp.ncbi.nlm.nih.gov/blast/db/v5/blastdbv5.pdf>

Improvements

- A new option (-subject_besthit) culls HSPs on a per subject sequence basis by removing HSPs that are completely enveloped by another [HSP](#). This is an experimental option and is subject to change.
- Allow use of the -max_target_seqs option for formats 0-4. The number of alignments and descriptions will be set to the max_target_seqs.
- Issue a warning if -max_target_seqs is set to less than five.

Bug fixes

- Disabled an overly aggressive optimization that caused problems mentioned by Shah et al. in <https://www.ncbi.nlm.nih.gov/pubmed/30247621>.
- Fixed an invalid memory error that occurred when [composition-based statistics](#) and [SEG](#) were used.
- Fixed some memory problems with the culling option.
- Nucleotide scores for even rewards are no longer rounded down to an even number when displayed.
- Blastdbcmd now reports intervals in the output [FASTA](#) if a partial sequence is requested with the range option.

BLAST+ 2.8.0: March 28, 2018

NOTE: This is an alpha release to allow users to test and comment on new features

Improvements

- Support for a new version of the BLAST database that allows you to limit search by taxonomy as well some other improvements. See description at <https://ftp.ncbi.nlm.nih.gov/blast/db/v5/blastdbv5.pdf>.
- The 2GB output file size limit for makeblastdb has been increased to 4 GB.

Bug fixes

- Fix makeblastdb problem with producing spurious files with masking information.
- Fix a problem with being unable to retrieve taxonomy information for tabular output.

BLAST+ 2.7.1: October 23, 2017

Improvements

- Provided an upper limit on the number of threads for BLAST+ search applications.
- Improved performance of taxonomic name lookups.
- Fixed Mac installers so they are interoperable with other NCBI applications.
- Reduced the amount of locking in BLASTDB reading library (CSeqDB).

Bug fixes

- Fixed race condition when using glist parameter.
- Fixed culling_limit bug with HSPs from different strands
- Fixed dustmasker bug with long region of Ns
- Fixed bl2seq problem with HTML output

Note: If building BLAST+ from source code, the [LMDB](#) library will be needed.

BLAST+ 2.6.0: January 09, 2017

New features

- Handle bare accessions on blastdb_aliastool.
- Change defaults for output formats 6, 7, and 10 to incorporate version in accessions.

Improvements

- Add support for NCBI_DONT_USE_LOCAL_CONFIG and NCBI_DONT_USE_NCBI_RC environment variables.
- Better runtime performance in blastdbcmd when the entry_batch parameter is used.
- SAM output improvements.
- Changed gapped alignment starting point to minimize the chance to produce sub-optimal alignments.
- For custom matrices absent from the util/tables source file, use BLOSUM62 for reporting number of positives.
- Added long_seqids flag to blastdbcmd to use long (legacy) NCBI Seq-id format.

Bug fixes

- Fixed issue with missing alignments in blastx.
- Fixed problem processing accession.version in makeblastdb.
- Fixed blastdbcmd problem with local IDs.
- Removed memory leak for multi-threaded runs.
- Fixed blastdbcmd crash when listing all entries and a sequence has no title.

BLAST+ 2.5.0: September 12, 2016

New features

- Composition based statistics for rpstblastn.
- Added output format for taxonomic organism report.
- Support for bare accessions in FASTA and BLAST reports.

Improvements

- -remote option connects to NCBI via HTTPS. This adds a dependency on GNUTLS (see <https://www.ncbi.nlm.nih.gov/books/NBK279690/>)
- Pre-fetch sequences for formatting.

Bug fixes

- Fixed improper functioning of output format 6 tokens ssciname, staxid, sscinames and staxids.
- tblastn core-dumps in multi-threaded mode.
- Ensure stable sorting of results in multi-threaded mode.
- Fixed incorrect percent identity in tabular format for sequences containing selenocysteine.

BLAST+ 2.4.0: June 02, 2016

New features

- Introduced multi-threaded traceback for blastp, blastx, tblastn and tblastx.
- Added new tabular format specifiers for taxonomic information (staxid, ssciname, scomname, sblastname, skingdom) that correspond to the first subject ID.

Improvements

- Speed up makeblastdb runtime performance with input consisting of many ambiguities.
- Better support for 'bare' IDs in taxid_map option to makeblastdb.
- Score U (selenocysteine) as C (not X) in protein-protein and translated searches.

Bug fixes

- Corrected E-value computation in finite-size correction.
- Removed memory leak from rpsblast.
- Made handling of ambiguities in subjects identical when using FASTA and BLAST database inputs.
- makeblastdb no longer replaces tabs in definition line by '#'.
- Corrected problem with spaces in database names on windows.
- Corrected handling of subject_loc.

BLAST+ 2.3.0: December 21, 2015

New features

- Added new PSIBLAST command line options to support saving PSSM and checkpoint files for each iteration and calculate checkpoint and PSSM for the last iteration.
- Added unique subject sequence query coverage to tabular output.
- Added support single file JSON and XML2 Blast output format.
- Beta release of SAM output format.
- Treat N subject sequences, entered with the -subject argument (bl2seq mode), as one search set rather than N sets.

Improvements

- makeblastdb ignores and warns users about empty sequences in input.
- BLAST+ only accepts "obinary" windowmasker files for performance reasons.

Bug fixes

- Best hits processing multi-threaded context.
- Fixed memory leak when invoking composition based statistics with an argument of 2.
- Return non-zero exit code when failing to write output file.
- Use relative paths in XInclude file for multi-file XML2 output format.
- Fixed memory leak in blastx.
- Fixed inconsistent XML2 output to standard output vs. file.
- Fixed psiblast incorrectly processing large input MSA.
- Fixed bug when running BLAST+ on windows with multiple threads.

BLAST+ 2.2.31: May 18, 2015

New features

- Added support for BLAST-XML2 specification.
- Added support for JSON Blast output format.

Improvements

- Improved adaptive batch size algorithm to better handle small databases.
- Preface error/warning message(s) with name of the application.
- Allow multiple defines even without GIs.
- Download more concise database information for -remote searches.

Bug fixes

- Fixed problem with makeblastdb's -max_file_sz.
- Reenabled support for word size 5 in tblastn.
- Fixed memory initialization problems.
- Use score for sorting search results if evalue less than 1.0e-180.

BLAST+ 2.2.30: October 6, 2014

New features

- Added tblastn-fast, blastp-fast, and blastx-fast tasks. These tasks make use of longer words as described by Shiryev et al. in <http://www.ncbi.nlm.nih.gov/pubmed/17921491>.
- Added new output option (outfmt 12) with Seq-Align in JSON.

Improvements

- Added new command line option qcov_hsp_perc that removes alignments below the specified query coverage.
- Added option line_length for the printing of alignment lengths (outfmt 0-4).
- Added larger gap penalties for PAM30 and PAM70 matrices.
- psiblast now accepts 0 for num_iterations to indicate iterating until convergence.
- rpsblast uses composition-based-statistics by default. Recover old behavior with "-comp_based_stats F -seg yes".
- Improved blastn multithreading performance for many queries with small databases.
- Changed cmdline option -sum_stats (formerly -sum_statistics) from flag to boolean.

Bug fixes

- Fixed spurious messages when parsing FASTA input.
- Fixed makeprofiledb handling of PSSMs created from multiple sequence alignment.
- Fixed makeblastdb handling of '-' at the end of FASTA input.
- Fixed windowmasker segmentation fault when the incorrect window size is provided.
- Fixed problem with lower-case masking and large sequences.
- Fixed makeblastdb segmentation fault on duplicate seqids.
- Allow specification of scoring matrices in lower case letters.
- Fixed exit code when disk space is not available for the output file.
- Fixed problem with using seqids list from -outfmt "6 sseqid" as input with -seqidlist.
- Fixed bug with culling_limit that excludes top hit.
- Fixed bug with max_target_seqs not working with psiblast.

BLAST+ 2.2.29: January 3, 2014

Improvements

- Improved the criteria for segging subject sequences used in composition based statistics with protein and translated searches.
- Improved blastn batch query performance.
- Improved blastdbcmd performance when retrieving taxonomic data from the BLAST databases.
- blastdb_aliastool supports reading a list of BLASTDBs from a file.
- Source releases build optimized multi-threaded binaries by default.
- Multi-threaded traceback: provides performance improvement for nucleotide-nucleotide BLAST with large (>25k) queries.
- Made makeprofiledb error messages more user friendly.
- Ungapped BLAST no longer uses sum statistics by default. Recover old behavior with -sum_statistics flag.
- Improved multithreading by better dividing the BLAST database among threads.

Bug fixes

- Allow update_blastdb.pl to work with databases containing more than 100 volumes.
- makeblastdb provides error message when -parse_seqids is used and invalid FASTA is provided.
- ASCII PSSM output for psiblast and deltablast displays two-digit scores in a more readable manner.
- Fixed negative percent identity in tabular output format
- Removed -num_threads option for binaries built without multi-threading.
- Fixed deltablast failures when searching multiple queries against multiple subject sequences.
- Fixed segmasker exception on example from BLAST cookbook.
- Fixed bogus warning about indexed megablast when using import search strategies.
- Fixed missing hits when running blastn with multiple queries, word size 7, large evalue, and no low complexity filtering.
- Fixed handling of gaps in ASN.1 input.
- Fixed Statistics_hsp-len value of 0 in XML output from blast_formatter.
- Fixed incorrect query coverage computation when sequence range was specified.
- Tabular output no longer ignores the -db_gencode argument.
- Fixed missing query sequence data in BLAST archive when -parse_deflines and FASTA with gnl ID was provided.
- Produce one XML document from BLAST archive.
- Removed 100 volume restriction in blastdb_aliastool -num_volumes.
- Fixed caption for 'query coverage' in tabular output format.
- Approximate gapped alignment in blastp is turned on/off for each query individually.
- Fixed query genetic code option.

BLAST+ 2.2.28: March 19, 2013

New features

- Composition based statistics support in rpsblast
- Support for query coverage, subject sequence title, and taxonomy data in custom tabular output format
- blastdbcmd support for batch subsequence retrieval

Improvements

- Adaptive BATCH_SIZE
- Perform incremental XML output

Bug fixes

- Formatting of asterix character in XML output
- Segmentation fault on out-of-memory
- Prevented extension of alignment into Ns
- Segmentation fault in DeltaBLAST when used with -remote and -out_ascii_pssm
- Replace tabs with spaces in FASTA deflines
- blastdbcmd displaying internal sequence ID for databases built without -parse_seqid
- blastdbcmd not fetching sequence data for complete sequence ID and -target_only
- blastn missing a hit for small word sizes
- Crash in blastn when it fetches sequence data from Genbank
- DeltaBLAST returning no hits when used with -remote option and searching more than one query
- Initialization problems for indexed megablast

- psiblast problem using -import_search_strategy
- blast_formatter displaying empty query for DeltaBLAST RID
- makeblastdb problem with ASN.1 input
- dustmasker errors with acclist and maskinfo_xml output formats
- blastx reporting of HSPs dependent on -max_target_seqs
- psiblast's display of number of queries in tabular output format
- blastx error when -ungapped and -comp_based_stats F are used

BLAST+ 2.2.27: September 10, 2012

New features

- Composition-based statistics for blastx.
- Added seedtop - a tool for searching for patterns in an input sequence or BLAST database.
- Enable remote DELTA-BLAST searches.

Improvements

- Revamped controls for the number of alignments/descriptions so that they are specific to applicable output formats (see user manual for details).
- Reduce memory usage for BLAST searches that involve (large) multiple queries.
- Speed up start-up times for BLAST databases.
- Display of new statistical parameters have been added to the BLAST results.
- Speed up runtime performance of tabular output formatting.
- Improve the placement of gaps in MegaBLAST

Bug fixes

- Fixed formatting bug when GI input format is provided to blastn.
- Fixed incorrect composition-statistics default for DELTA-BLAST.
- Bug fixes in blast_formatter, blastdbcmd.
- An asterix (stop-codon) in sequence was not rendered properly.
- The Smith-Waterman option in blastp would cause seg filtering on the subject sequence even if the composition-based statistics were not being used.
- The makeblastdb taxid_map option is broken.

BLAST+ 2.2.26: January 31, 2012

New features

- Mac executables are now Universal Binaries for 32- and 64-bit architectures; we no longer produce PPC and Intel Universal binaries. The executable archive names remain unchanged.
- Added DELTA-BLAST - a new tool for sensitive protein searches
- Added makeprofiledb - a tool for creating a database for RPS-BLAST

Improvements

- The blast_formatter application can now format bl2seq RIDs.
- PSI-BLAST can produce archive format, blast_formatter can format that output.
- PSI-BLAST has two new options that work with multiple-sequence alignments: ignore_msa_master and msa_master_idx (see BLAST+ manual).

- mkmbindex can now create masked indices from a BLAST database and ASN.1 masking data.
- An improved finite size correction is now used for blastp/blastx/tblastn/rpsblast.
- The FSC is subtracted from the query and database sequence length for the calculation of the expect value. The new FSC results in more accurate expect values, especially for alignments with a short query or target sequence. Re-enable the old size correction by setting the environment variable OLD_FSC to a non-NULL value.
- The blastdbcmd -range parameter now accepts a blank value for the second parameter to signify the end of a sequence (e.g., -range "100-")
- There was a performance improvement for long database sequences in results with many matches.

Bug fixes

- There was a blastn problem if subject_loc and lcase_masking were used together.
- There was a problem with multi-threaded blastx if the query included a long (10,000+) sequence of N's.
- The percent identity calculation was wrong if the best-hit algorithm was used.
- There was a problem with the multiple BLAST database statistics report in XML format.
- Makeblastdb failed to return an error when input was not available.
- The formatting option -outfmt "7 nident" always printed zero.
- The search strategy was not properly saving the -db_soft_mask option.
- An error message was emitted if there was a "<" in the query title.
- A problem reading lower-case masking from the query could cause a search to fail.

BLAST+ 2.2.26: March 15, 2011

New features

- Enhanced documentation, includes simplified setup instructions, available at
- <http://www.ncbi.nlm.nih.gov/books/NBK1762>

Improvements

- Added support for hard-masking of BLAST databases.
- Improve performance of makeblastdb for FASTA input with large numbers of sequences, improve error checking.
- Allow Best Hit options and XML formatting for Blast2Sequences mode
- Allow multiple query sequences for psiblast.
- Allow specification of any multiple sequence alignment sequence as the master with the -in_msa psiblast argument.
- Add an optional -input_type argument to makeblastdb.
- Added support for query and subject length to tabular output.
- Performance of -seqidlist argument improved.
- The minimum of the number of descriptions and alignments is now used for tabular and
- XML output (consistent with the behavior of the older blastall applications).

Bug fixes

- Makeblastdb and blastdbcmd problems with parsing, storing, and retrieving sequence identifiers.
- Missing subject identifiers in tabular output.
- Blast_formatter ignoring -num_alignments and -num_descriptions
- Blast archive format could be saved incorrectly with multiple queries.
- Blast_formatter established an unneeded network connection.

- Blast_formatter did not save masking information correctly.
- Rpstblastn might crash if searching many sequences.
- Indexed megablast would not run in multi-threaded mode.
- Query title in the PSSM saved by psiblast was not being stored.
- Possible failure to run in multi-threaded mode with multiple queries or large database sequences.
- Tblastn runs with database masking might miss matches.

BLAST+ 2.2.24 bug fix release: October 30, 2010

Bug fixes

- Improved makeblastdb performance and taxid_map option
- Fixed segmentation faults on blastn and megablast
- Fixed truncated output for sequence input with extra spaces in the define
- Fixed problem with MacOSX binaries on MacOSX 10.5

BLAST+ 2.2.24: August 2, 2010

- Added support for BLAST Archive format (see BLAST+ user manual)
- Added the blast_formatter application (see BLAST+ user manual)
- Added support for translated subject soft masking in the BLAST databases
- Added support for the BLAST Trace-back operations (btop) output format
- Added command line options to blastdbcmd for listing available BLAST databases
- Improved performance of formatting of remote BLAST searches
- Use a consistent exit code for out of memory conditions
- Fixed bug in indexed megablast with multiple space-separated BLAST databases
- Fixed bugs in legacy_blast.pl, blastdbcmd, rpsblast, and makeblastdb
- Fixed Windows installer for 64-bit installations

BLAST+ 2.2.23: Feb 03, 2010

- Bug fix for tabular output formatting involving BLAST databases that do not have parseable defines.
- Fixed problem displaying accessions in XML output format.
- Prevent collisions between queries and subject sequences with local identifiers.
- Fixed megablast performance regression when used with query masking.
- Fixed seg filtering failure for blastx and genomic sequences.
- Implemented saving search strategies in bl2seq mode.
- Fixed bug in tabular output format with qseq, sseq, pident and ppos keywords.
- Fixed bug with blastp-short task.
- Fixed blastdbcmd retrieval of taxids for BLAST databases without GIs.
- Added makeblastdb support for adding masking information to existing BLAST databases.

BLAST+ 2.2.22 Internal bug fix release: November 02, 2009

- Fix issue dealing with opening BLAST databases which contain references to a
- BLAST database specified with a relative path.
- Prevent collisions between queries and subject sequences with local identifiers

BLAST+ 2.2.22: Sep 27, 2009

- Added `entrez_query` command line option for restricting BLAST databases.
- Added support for `psi-tblastn` to the `tblastn` command line application via the `-in_pssm` option.
- Improved documentation for subject masking feature in user manual.
- User interface improvements to `windowmasker`.
- Made the specification of BLAST databases to resolve GIs/accessions configurable.
- `update_blastdb.pl` downloads and checks BLAST database MD5 checksum files.
- Allowing long words with `blastp`.
- Added support for overriding megablast index when importing search strategy files.
- Added support for best-hit algorithm parameters in strategy files.
- Bug fixes in `blastx` and `tblastn` with genomic sequences, subject masking, `blastdbcheck`, and the SEG filtering algorithm.

BLAST+ 2.2.21: May 27, 2009

- Added support for Best-Hit algorithm.
- Added support for `-in_msa psiblast` option.
- Performance improvements and bug fixes to subject soft masking feature (note: the file format for the files containing the masking information has changed in a non-backwards compatible way).
- Changed command line option to specify single soft masking algorithm to mask BLAST databases from `-mask_subjects` to `-db_soft_mask`.
- Masked FASTA and subject masks can be obtained via `blastdbcmd`.
- Improved error messages when `makeblastdb` processes masking information.
- Bug fixes in tabular output for translated searches.
- Bug fixes to `makeblastdb`.
- Bug fixes to search strategies and megablast.
- Bug fixes to XML output.
- Bug fixes and performance improvements to multi-threaded execution.
- Bug fixes to lower case masking in `blastx`.
- Bug fixes to ungapped searches.
- Added support for smaller lookup tables for small queries.
- Added support for partial sequence fetching during traceback.
- Fixed the 2-hit algorithm so that no overlap between two hits is allowed.
- Implemented a new method to compute effective observations and new entropy-based method to compute column-specific pseudocounts in PSI-BLAST.
- Remote BLAST database data loader is used as a fallback if local BLAST databases cannot be found.
- Bug fixes, improved error messages, and support for ASN.1 input in `makeblastdb`.
- Bug fixes and performance improvements to subject masking feature.
- Added the `update_blastdb.pl` script
- Updated BLAST+ user manual with documentation about configuring BLAST, automatic resolution of sequence identifiers, and a description of how the BLAST databases are searched.

BLAST+ 2.2.19: November 03, 2008

- Made sequence ID/title display uniform in sequence filtering applications.
- Fixed incorrect display of filtering options in XML output.
- Fixed handling of empty sequences in BLAST input.
- Fixed negative strand handling for `tblastn`/`tblastx`.

BLAST+ 2.2.18: October 14, 2008

- Added update_blastdb.pl script to distribution of BLAST+ command line applications.
- Changed a few PSI-BLAST constants for pseudo-counts.
- Bug fix in blastdbcmd to distinguish non-redundant sequence titles.
- Bug fix to display BLAST database information remotely from outside NCBI for XML output.

BLAST+ 2.2.17 internal release: September 24, 2008

- Fix to prevent initial seed extension from going beyond context boundary.
- Improvements to reduce memory usage when query splitting is applied.
- Print the accession and version for blastdbcmd's %a output format.
- gilists/negative gilists are not saved in search strategies or supported in remote blast searches.
- legacy_blast.pl fixed for MacOSX, as well as extended support for megablast formatting options (-D, -f).
- Enhancements to Mac installer to add installation path to user's PATH.
- ASN.1 output is now of type Seq-annot.
- -lcase_masking option now applies to subject sequences as well as queries.
- Bug fix for creation of masked databases with non-redundant sequences that use a BLAST database as its data source.
- Bug fix for merging masking locations.

BLAST+ 2.2.16 internal release: August 21, 2008

- First internal release

BLAST FTP Site

Tao Tao, PhD,¹ Tom Madden, PhD,² and Camacho Christiam³

Created: May 29, 2011; Updated: August 30, 2020.

The NCBI FTP server contains a BLAST-specific directory (<https://ftp.ncbi.nlm.nih.gov/blast/>). Through this directory, the standalone BLAST packages and a standard set of BLAST databases are available to the public for download through anonymous FTP. For faster download, the service is also available through the Aspera client for those users with the Aspera browser plug-in installed (<https://www.ncbi.nlm.nih.gov/public/?blast/>).

This document describes the subdirectories and file contents of the BLAST FTP directory. Technical details on how to use certain files, especially those under the db (database) subdirectory, are also provided.

Subdirectories under the BLAST FTP directory

There are several subdirectories under the BLAST FTP directory. Each stores a set of files with similar types of content. These subdirectories are summarized in Table 1.

Table 1. File content of subdirectories under the "/blast" FTP directory

Subdirectory	File content
db	Preformatted BLAST database files and FASTA sequence files (only for a few representative databases, kept under the /FASTA subdirectory)
demo	Various demonstration packages for software developers
documents	Preliminary documentation (mostly from software developers) and pointers to other documentation
executables	Different releases for standalone BLAST packages, including blast+
matrices	Different scoring matrices, only a selected subset are supported by blast
temp	Miscellaneous files
WGS_TOOLS	Perl scripts for generating WGS project-based database alias for TSA and WGS datasets, to be used with vdb blast
windowmasker_files	A collection of windowmasker files, organized into subdirectory named by the taxonomic ids

The "/db" subdirectory

This subdirectory contains a common set of preformatted BLAST database files in version 5 format. The FASTA sequences for a few widely used databases are stored under the "/FASTA" subdirectory. The contents of available preformatted databases are summarized separately according to their sequence nature and sources (for nucleotide databases). The databases provided for the cloud-based BLAST packages are under the "/cloud" subdirectory. The version 4 databases are kept in the "/v4" subdirectory, those entries will not be updated. The "/v5" subdirectory is a soft link back to the "/db" directory.

Table 2a. Preformatted protein database files

File name	Contents
landmark.tar.gz	The landmark database includes complete proteomes from a few selected representative genomes spanning a wide taxonomic range, the main database used by the SmartBLAST services.

Author Affiliations: 1 NCBI; Email: tao@ncbi.nlm.nih.gov. 2 NCBI; Email: madden@ncbi.nlm.nih.gov. 3 NCBI; Email: camacho@ncbi.nlm.nih.gov.

✉ Corresponding author.

Table 2a. continued from previous page.

File name	Contents
cdd_delta.tar.gz	Condensed conserved domain database for use with deltablast protein searches.
nr.##.tar.gz	A collection of protein sequences with entries from GenPept, Swissprot, PDB, PRF, PIR and NCBI Reference Sequence (RefSeq) project.
pataa.tar.gz	Protein sequences from patents as supplied by USPTO. These entries are EXCLUDED from the nr database.
pdbaa.tar.gz	Protein sequences from PDB structure records' protein components.
refseq_protein.##.tar.gz	Protein sequences from NCBI RefSeq project.
swissprot.tar.gz	Protein sequences from the swiss-prot sequence database (last major update).
tsa_nr.##.tar.gz	Protein sequences from the Transcriptome Shotgun Assembly. Its entries are EXCLUDED from the nr database.
env_nr.##.tar.gz	Protein sequences from large environmental sequencing projects, e.g., Sargasso Sea, Acid Mine Drainage. Its entries are EXCLUDED from the nr database.

Table 2b. Preformatted RefSeq nucleotide database files

File	Contents
16S_ribosomal_RNA.tar.gz	Microbial 16S RNA sequences from the RefSeq Targeted Loci project (https://www.ncbi.nlm.nih.gov/refseq/targetedloci/).
refseq_rna.##.tar.gz	RNA sequences from NCBI RefSeq project, also included in the nt database.
refseq_euk_rep_genomes.##.tar.gz	Eukaryotic representative genomes from NCBI RefSeq project
refseq_prok_rep_genomes.##.tar.gz	Prokaryotic representative genomes from NCBI RefSeq project
refseq_viroids_rep_genomes.##.tar.gz	Viriods representative genomes from NCBI RefSeq project
refseq_viruses_rep_genomes.##.tar.gz	Viruses representative genomes from NCBI RefSeq project
human_genome.##.tar.gz	Current refseq human genome assembly (GRCh) with various database masking
mouse_genome.##.tar.gz	Current refseq mouse genome assembly (GRCm) with various database masking

Table 2c. Preformatted non-RefSeq nucleotide and target loci databases

File	Contents
nt.##.tar.gz	The nucleotide sequence database contains entries from traditional divisions of GenBank, EMBL and DDBJ. Sequences from bulk divisions, i.e., gss, sts, pat, est, htg, wgs, con, and environmental sequences are excluded. RefSeq genomic entries are also excluded.
patnt.##.tar.gz	Nucleotide sequences from patents as supplied by USPTO to GenBank, or from EU/Japan Patent Agencies through EMBL/DDBJ. Entries are EXCLUDED from the nt database.
pdbnt.##.tar.gz	Sequences for the nucleotide components of PDB structure records.
tsa_nt.##.tar.gz	A database with earlier non-project based Transcriptome Shotgun Assembly (TSA) entries. Project-based TSA entries are NOT included. Entries are EXCLUDED from the nt database.
ITS_*.tar.gz	Databases with collection fungal or eukaryotic Internal Transcribed Spacer sequences.
LSU_*_rRNA.tar.gz	Database with large submit rRNA sequences for prokaryotes and eukaryotes.
SSU_*_rRNA.tar.gz	A database with sequences small from fungi and eukaryotes
taxdb.tar.gz	A non-sequence database file containing taxonomic information for sequences in the preformatted databases providing common and scientific names for each entry.

Getting the preformatted database files

Preformatted BLAST database files offer several advantages over the FASTA files:

- The preformatted databases are broken into smaller volumes and therefore can be downloaded more readily with fewer errors
- A convenient Perl script (***update_blastdb.pl*** found in the bin directory of a locally installed blast+ package) is available to simplify the download of these preformatted databases
- Preformatted database files remove the makeblastdb formatting steps, and saves valuable processing time and disk space
- Taxonomic information is encoded within the preformatted databases and can be used to limit the scope of a blast search, and sequence retrieval, and scientific name addition through the included taxdb files
- Sequences in FASTA format can be generated easily from the preformatted databases using the blastdbcmd utility when needed

Preformatted databases must be downloaded in binary mode, downloading through the *update_blastdb.pl* script is recommended. An example command line for getting the preformatted refseq_rna nucleotide database and the session output are given below.

```
$ perl ../bin/blast+/update_blastdb.pl --passive --decompress refseq_rna
Connected to NCBI
Downloading refseq_rna (7 volumes) ...
Downloading refseq_rna.00.tar.gz... [OK]
Downloading refseq_rna.01.tar.gz... [OK]
Downloading refseq_rna.02.tar.gz... [OK]
Downloading refseq_rna.03.tar.gz... [OK]
Downloading refseq_rna.04.tar.gz... [OK]
Downloading refseq_rna.05.tar.gz... [OK]
Downloading refseq_rna.06.tar.gz... [OK]
Decompressing refseq_rna.00.tar.gz ... [OK]
Decompressing refseq_rna.01.tar.gz ... [OK]
Decompressing refseq_rna.02.tar.gz ... [OK]
Decompressing refseq_rna.03.tar.gz ... [OK]
Decompressing refseq_rna.04.tar.gz ... [OK]
Decompressing refseq_rna.05.tar.gz ... [OK]
Decompressing refseq_rna.06.tar.gz ... [OK]
```

The complete options of this script (obtained using specific option "--help") are shown below.

```
$ perl update_blastdb.pl --help
NAME
    update_blastdb.pl - Download pre-formatted BLAST databases

SYNOPSIS
    update_blastdb.pl [options] blastdb ...

OPTIONS
    --decompress
        Downloads, decompresses the archives in the current working directory,
        and deletes the downloaded archive to save disk space, while
        preserving the archive checksum files (default: false).

    --showall
        Show all available pre-formatted BLAST databases (default: false). The
        output of this option lists the database names which should be used
        when requesting downloads or updates using this script.
```

It accepts the optional arguments: 'tsv' and 'pretty' to produce tab-separated values and a human-readable format respectively. These parameters elicit the display of additional metadata if this is available to the program. This metadata is displayed in columnar format; the columns represent:

name, description, size in gigabytes, date of last update (YYYY-MM-DD format).

--blastdb_version

Specify which BLAST database version to download (default: 4).
Supported values: 4, 5

--passive

Use passive FTP, useful when behind a firewall or working in the cloud (default: true). To disable passive FTP, configure this option as follows: --passive no

--timeout

Timeout on connection to NCBI (default: 120 seconds).

--force

Force download even if there is a archive already on local directory (default: false).

--verbose

Increment verbosity level (default: 1). Repeat this option multiple times to increase the verbosity level (maximum 2).

--quiet

Produce no output (default: false). Overrides the --verbose option.

--version

Prints this script's version. Overrides all other options.

--num_cores

Sets the number of cores to utilize to perform downloads in parallel when data comes from GCS. Defaults to all cores (Linux and macos only).

DESCRIPTION

This script will download the pre-formatted BLAST databases requested in the command line from the NCBI ftp site.

EXIT CODES

This script returns 0 on successful operations that result in no downloads, 1 on successful operations that downloaded files, and 2 on errors.

BUGS

Please report them to <blast-help@ncbi.nlm.nih.gov>

COPYRIGHT

See PUBLIC DOMAIN NOTICE included at the top of this script.

Using the preformatted BLAST database files

The `--decompress` option of `updated_blastdb.pl` automatically decompresses and extract the archives of the requested database files. When manually downloading preformatted databases, those compressed archives must be downloaded in binary format using the passive mode, then inflated with **gunzip** or other decompress utilities. The working database files can then be extracted out of the resulting tar archive using *tar* program in Unix/Linux or *WinZip* and *StuffIt Expander* on Windows and Macintosh platforms, respectively.

Large databases are formatted in multiple gigabytes-sized volumes, which are named using the "name.##.tar.gz" convention. To reconstitute a given multi-volume database, all volumes with the same base database name are required. A database alias file, with ".pal" extension for protein or ".nal" extension for nucleotide, is provided to tie the volumes together. The database can be called using the base database name. For example, binary programs from the blast+ package can call the nt database using the command line option of "**-db nt**" option argument.

For proper setup of standalone blast+, it is recommended that database files be stored in a centralized directory, with the path to this directory be encoded by the BLASTDB variable. Details are available in the setup document for Windows and Mac/Linux/Unix.

Sequence files under the "/db/FASTA/" subdirectory

This subdirectory contains sequence files in the FASTA format. With preformatted databases readily available, only a few commonly used databases are available in this format.

Table 3. Protein database files under the /db/FASTA directory

File	Content
nr.gz	The FASTA equivalent of the nr.##.tar.gz database files
swissprot.gz	The FASTA equivalent of the swissprot.tar.gz database file.
nt.gz	The FASTA equivalent of the nt.##.tar.gz database files.

For local BLAST searches, the recommendation is to use the preformatted version given in the parent directory. For those without preformatted counterparts, the FASTA sequence file first need to be inflated using gunzip or other comparable utilities, the resulting file can then be formatted by *makeblastdb* from the blast+ package. Example command lines for formatting igSeqNt and igSeqProt are given below.

```
$ makeblastdb -in swissprot -dbtype prot -parse_seqids
$ makeblastdb -in nt -dbtype nucl -parse_seqids
```

For vector screening needs, get the FASTA sequences from this ftp directory and formatted them using *makeblastdb*:

<https://ftp.ncbi.nlm.nih.gov/pub/UniVec/>

Chromosome entries are available in the refseq_euk_rep_genomes and refseq_prok_rep_genomes databases. Organism-specific sequences are available in FASTA format under the genomes FTP directory:

<https://ftp.ncbi.nlm.nih.gov/pub/genomes/all/> https://ftp.ncbi.nlm.nih.gov/pub/factsheets/Factsheet_Assembly.pdf

The NCBI Datasets tool is another way to get the genomic data from NCBI. Please refer to this page for more information: <https://www.ncbi.nlm.nih.gov/datasets>

Database files under the **"/db/v4/" subdirectory**

This subdirectory contains the preformatted blast database files in the older version 4 format. Those databases will not be updated. They are meant as a stop-gap measure to ease the transition to version 5 of the databases. Content description for this subdirectory will be skipped.

Database update

In general, BLAST databases are updated daily. There is no established incremental update scheme due to sequence removal and update. It is recommended that databases be downloaded at regular intervals to keep the content of local copy current. The ***update_blastdb.pl*** script can help streamline this download process. If the original database.##.tar.gz files are kept, this utility can automatically check the time stamps to determine if file refreshing is required or not.

Faster download through Aspera plug-in is also possible (downloadable from the [Aspera Soft site](#) under the download tab). A web interface for the NCBI FTP site is at: <https://www.ncbi.nlm.nih.gov/public/>. Aspera's commandline client ascp can be used to access Aspera indexed NCBI ftp site.

Contents of the **"/blast/demo/" subdirectory**

This directory contains technical presentations given by NCBI BLAST developers in scientific conferences and several tools and documents relevant to the BLAST service to demonstrate how specific functions from NCBI's C-toolkit code can be used.

Table 4. Contents of the /blast/demo/ subdirectory

File/Dir Name	Content
QUICKBLASTP	Standalone kmer indexing tool and the kmer blastp tool. The algorithm is used by the web protein blast search when "Quick BLASTP (Accelerated protein-protein BLAST) " option is selected
README.quickblastp	Readme for quickblastp
quickblastp.tar.gz	Kmer-based quickblastp demonstration package
benchmark	Package with sample database and query for gauging the performance of BLAST releases on different platforms
bmc	Sequences used in the generation of BLAST search data for the blast+ paper published in BMC
igblast	Directory with igblast related set of scripts, see its README for more details
magicbkast_article	Supplemental materials for the magicblast paper and the binary used to generate the specs
blast_programming.ppt	PowerPoint presentation on BLAST programming
mt_tback.tgz	Multi-threaded traceback related test code
openmp_test.tar.gz	Openmp multi-thread related test code
parse_blast_xml.tar.gz	Demo package on parsing xml styled blast output
test_suite.tar.gz	An old set of test sequences with csh script
vecscreen	Binary program for vecscreen
*.ppt, *.pdf	Slides or posters presented by the BLAST group in various conferences
*.fsa	Miscellaneous sequences

Since NCBI is migrating to C++ code base provided by the [C++ toolkit](#), some of the files from this directory could become obsolete or change without notice. Most of the functions demonstrated here are incorporated in the *blast_formatter* utility distributed in the [blast+ package](#).

Contents of the `"/blast/documents/"` subdirectory

This directory contains mostly posters and other preliminary documentation from BLAST developers. For blast+ packages, a [user manual](#) along with the [instruction for installation](#) are available through [NCBI bookshelf](#). Content description will be skipped. See the README for this directory for details: <https://ftp.ncbi.nlm.nih.gov/blast/documents/README>

Contents of the `"/blast/executables/"` subdirectory

This directory contains several subdirectories each for a set of BLAST distribution packages from a specific release. Binaries based on the new C++ toolkit are under the `/blast+` subdirectory with the latest release directly accessible through the `/LATEST` symbolic link. The only program file is **remote_fuser**, for use in database fetching in the cloud implementation.

Contents of the `"/blast/executables/LATEST/"` subdirectory

This directory is a symbolic link pointing to the LATEST release of BLAST+ programs built from the NCBI C++ toolkit. Packages for common platforms available in different formats are summarized in Table 5 below.

Table 5. File content of the `/blast/executables/LATEST/` subdirectory

File	Contents
ChangeLog	Changes introduced in this release
ncbi-blast-*.src.*	blast+ source code in different formats
ncbi-blast-#.##+.x86_64.rpm	rpm installation package for PC running 64-bit Linux
ncbi-blast-#.##+.x64-linux.tar.gz	Tar archive for PC running 64-bit Linux
ncbi-blast-*.dmg	Disk image for Macintosh running 64-bit OSX
ncbi-blast-*.macosx.tar.gz	Tar archive for Macintosh running 64-bit OSX
ncbi-blast-*.win64.exe	Installer for PC running 64-bit Windows
ncbi-blast-#.##+.x64-win64.tar.gz	Tar archive for PC running 64-bit Windows

All non-source code archives or packages are equivalent. They contain a standard collection of standalone command line programs and accessory utilities for different platforms. Installation of the package enables local BLAST searches, custom database preparation from FASTA sequences, as well as sequence retrieval from existing databases formatted with the `"-parse_seqids"` argument.

Details on individual programs from the package as well as installation procedures are available for Windows and Mac/Linux/Unix.

Note that blast+ does not provide a separate client-server tool. That function is built into individual blast programs, i.e. *blastn*, *blastp*, *blastx*, *tblastn* and *tblastx*, and can be invoked using the `"-remote"` option. The `"-remote"` option enables remote search against databases at NCBI using NCBI's computation resources. In addition, blast+ does not provide package equivalent to the decommissioned *wwwblast* package.

Contents of the `"/blast/executables/blast+/"` subdirectory

This subdirectory archives all releases of the blast+ package. Each release is under its own directory named using its version number. Available builds start at version 2.2.18, with version 2.2.20 skipped. Optional version 5 database support began in release 2.8.0alpha.

Contents of the **"/blast/executables/legacy.NOTSUPPORTED/"** subdirectory

This subdirectory archives the releases of legacy blast package based on the NCBI C Toolkit. They are meant for historical references, NCBI no longer supports them. Each release is under its own directory with the version number as its name. Available builds start at version 2.0.7.

Packages with version number 2.2.10 or newer are packaged with a built-in directory structure to better organize the distributed contents.

Contents of the **"/blast/executables/igblast/"** subdirectory

This subdirectory archives the different releases of standalone igblast package (under the release subdirectory. The separate data files and database files required by the binary programs in this package are available in subdirectories separate from each releases.

Table 6. File content of the /blast/executables/igblast/release/ subdirectory

File	Contents
##.	Different release directories
LATEST	Soft link pointing to the latest release subdirectory
database *	germline immunoglobulin gene sequences for mouse and rhesus monkey. Databases for other organisms are NOT provided due to license requirement.
Internal_data	Internal vdj gene information file for annotation need
optional_file	Additional gene annotation file
edit_imgt_file.pl *	A perl script for manipulating the defines of IMGT immunoglobulin sequences so they can be used as input to make databases through makeblastdb

* Note: Internal_data, optional_file, and database directories are included in release 1.13.0 or later; use the files downloaded with the release. Additionally, license requirement prevents NCBI from distributing the germline sequences for certain organisms as blast-ready databases. Instead, those sequences should be obtained from IMGT as FASTA. Convert the defines of the IMGT sequences using the "edit_imgt_file.pl" script first before using makeblastdb to format the file into a igblast readable database.

Contents of the **"/blast/executables/magicblast/"** subdirectory

A subdirectory for the releases of a next generation sequence read mapper from NCBI. The LATEST directory maps to the current release's directory. Refer to the README file under the subdirectory for more details.

Details technical description of the package is at: <https://ncbi.github.io/magicblast/>.

Contents of the **"/blast/executables/rmbblast/"** subdirectory

This subdirectory archives the repeat masker BLAST. Only two releases, 2.2.27 and 2.2.28, are available. Refer to the readme for more details.

Contents of the **"/blast/matrices/"** subdirectory

This directory contains an extensive list of score matrices, most of which are experimental in nature and not supported by blast programs. The matrices can be grouped into PAM family of matrices, BLOSUM family of matrices, matrices for nucleotide and other miscellaneous matrices.

Table 7. Summary of matrices found in the /blast/matrices/ subdirectory

File	Contents
BLOSUM*	BLOSUM family of score matrices for protein alignment
PAM*	PAM family of score matrices for protein alignment
DAYHOFF*, GONNET*	Variants of PAM matrices for protein alignment
MATCH, IDENTITY	Simplified score matrices for protein alignment
NUC*	Nucleotide score matrices
PAM.tar.gz	C source code for generating PAM matrices

Contents of the "/blast/temp/" subdirectory

This is a directory used for testing purposes or other special needs that do not fall into the above categories.

Contents of the "/blast/WGS_TOOLS/" subdirectory

This directory provides two database alias generating tools for WGS and TSA datasets, respectively. These tools take an input taxonomic id and generate a database alias for project-based WGS or TSA datasets - those with four letter project prefix as listed at <https://www.ncbi.nlm.nih.gov/Traces/wgs/>. The alias file produced can be used with the vdb blast tools available from the sratoolkit, which is available for common platforms: <https://www.ncbi.nlm.nih.gov/Traces/sra/?view=software>. A handout describing the general usage of these vdb blast programs is available at:

ftp://ftp.ncbi.nlm.nih.gov/pub/factsheets/HowTo_Local_SRA_BLAST.pdf

Getting Help

For details, please refer to documents under the [Help tab](#) of the BLAST homepage or the [document directory](#) under the BLAST FTP site.

Comments, questions and bug reports specifically relating to the BLAST programs and their usage should be sent to blast-help@ncbi.nlm.nih.gov.

BLAST Glossary

Jan Fassler, Ph.D.¹ and Peter Cooper, Ph.D.²

Created: July 14, 2011.

algorithm

A fixed procedure embodied in a computer program.

alignment

The process or result of matching up the nucleotide or amino acid residues of two or more biological sequences to achieve maximal levels of identity and, in the case of amino acid sequences, conservation, for the purpose of assessing the degree of similarity and the possibility of homology.

bioinformatics

Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data.

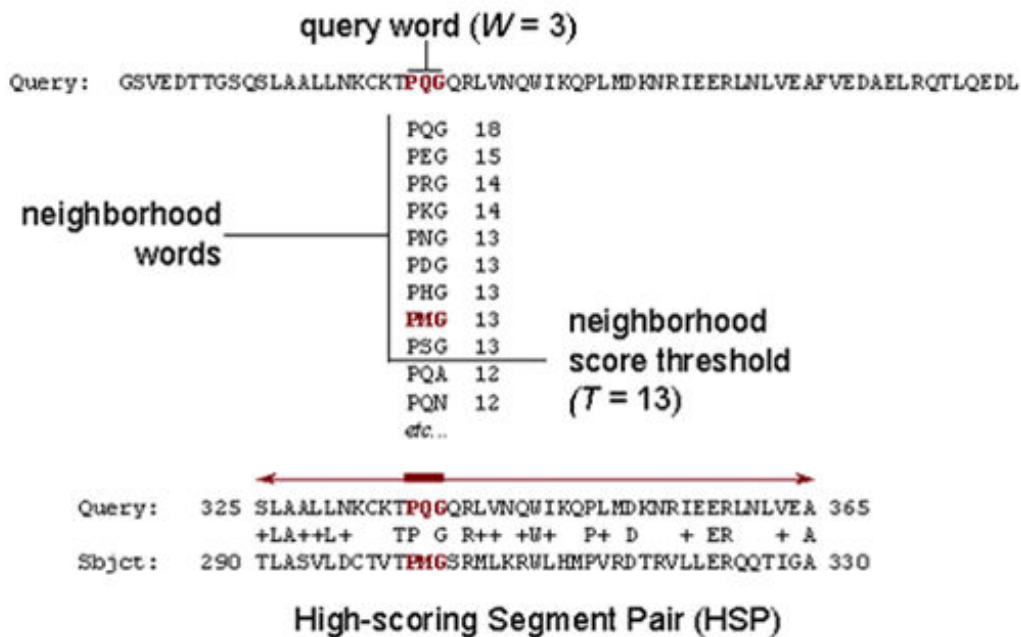
bit score

The bit score, S' , is derived from the raw alignment score, S , taking the statistical properties of the scoring system into account. Because bit scores are normalized with respect to the scoring system, they can be used to compare alignment scores from different searches.

BLAST

Basic Local Alignment Search Tool ([Altschul et al., 1990 & 1997](#)) is a sequence comparison algorithm optimized for speed used to search sequence databases for optimal local alignments to a query. The initial search is done for a word of length "W" that scores at least "T" when compared to the query using a substitution matrix. Word hits are then extended in either direction in an attempt to generate an alignment with a score exceeding the threshold of "S". The "T" parameter dictates the speed and sensitivity of the search.

The BLAST Search Algorithm



BLOSUM

A Blocks Substitution Matrix is a substitution scoring matrix in which scores for each position are derived from *observations* of the frequencies of substitutions in blocks of local alignments in related proteins. Each matrix is tailored to a particular evolutionary distance. In the BLOSUM62 matrix, for example, the alignment from which scores were derived was created using sequences sharing no more than 62% identity. Sequences more identical than 62% are represented by a single sequence in the alignment so as to avoid over-weighting closely related family members. (Henikoff and Henikoff, 1992)

	A	C	D	E	F	G	H	
A	4	0	-2	-1	-2	0	-2	
C	0	9	-3	-4	-2	-3	-3	
D	-2	-3	6	2	-3	-1	-1	
E	-1	-4	2	5	-3	-2	0	
F	-2	-2	-3	-3	6	-3	-1	
G	0	-3	-1	-2	-3	3	0	
H	-2	-3	-1	0	-1	0	4	

BLOSUM 62

composition-based statistics

These are methods applied to protein BLAST searches that adjust the significance of alignment scores by taking into account the overall amino acid composition of the query and aligned database sequences. These methods provide more accurate statistics than those originally used in protein BLAST searches (Schäffer et al., 2001; Yu and Altschul, 2005). The conditional compositional score matrix adjustment method (Yu and Altschul, 2005) is used by default on the NCBI protein BLAST service.

conserved substitution

A change at a specific position of an amino acid or, less commonly, DNA sequence that preserves the physico-chemical properties of the original residue or achieves a positive score in the governing scoring matrix.

domain

A discrete portion of a protein assumed to fold independently of the rest of the protein and possessing its own function.

DUST

A program for filtering low complexity regions from nucleic acid sequences.

E value

The Expectation value or Expect value represents the number of different alignments with scores equivalent to or better than *S* that is expected to occur in a database search by chance. The lower the E value, the more significant the score and the alignment.

FASTA

The first widely used algorithm for database similarity searching. The program looks for optimal local alignments by scanning the sequence for small matches called "words". Initially, the scores of segments in which there are multiple word hits are calculated ("init1"). Later the scores of several segments may be summed to generate an "initn" score. An optimized alignment that includes gaps is shown in the output as "opt". The

sensitivity and speed of the search are inversely related and controlled by the "k-tup" variable that specifies the size of a "word" ([Pearson and Lipman, 1988](#)).

filtering

Filtering, also known as masking, removes regions of (nucleic acid or amino acid) sequence having characteristics that may lead to spurious high scores. See SEG and DUST.

gap

A space introduced into an alignment to compensate for insertions and deletions in one sequence relative to another. To prevent the accumulation of too many gaps in an alignment, introduction of a gap causes the deduction of a fixed amount (the gap score) from the alignment score. Extension of the gap to encompass additional nucleotides or amino acid is also penalized in the scoring of an alignment.

global alignment

The alignment of two nucleic acid or protein sequences over their entire length.

H

H is the relative entropy of the target and background residue frequencies. ([Karlin and Altschul, 1990](#)). H can be thought of as a measure of the average information (in bits) available per position that distinguishes an alignment from chance. At high values of H short alignments can be distinguished by chance, whereas at lower H values a longer alignment may be necessary ([Altschul, 1991](#)).

homology

Similarity attributed to descent from a common ancestor. Homologous biological components (genes, proteins, structures) are called homologs. See also orthologs and paralogs.

HSP

A High-scoring Segment Pair (HSP) is a local alignment with no gaps that achieves one of the highest alignment scores in a given search.

identity

The extent to which two (nucleotide or amino acid) sequences have the same residues at the same positions in an alignment, often expressed as a percentage.

K

A statistical parameter used in calculating BLAST scores that can be thought of as a natural scale for search space size. The value **K** is used in converting a raw score (**S**) to a bit score (**S'**).

lambda

A statistical parameter used in calculating BLAST scores that can be thought of as a natural scale for scoring system. The value lambda is used in converting a raw score (**S**) to a bit score (**S'**).

local alignment

The alignment of a high-scoring region of two nucleic acid or protein sequences.

low complexity region

A region of biased composition in nucleic acid and protein sequences. These include homopolymeric runs, short-period repeats, and subtler over representation of one or a few residues. The SEG program is used to mask or filter low complexity regions in amino acid queries. The DUST program is used to mask or filter such regions in nucleic acid queries.

masking

Also known as filtering. The removal of repeated or low complexity regions from a sequence in order to improve the sensitivity of sequence similarity searches performed with that sequence.

motif

A short conserved region in a protein sequence. Motifs are frequently highly conserved parts of domains.

multiple sequence alignment

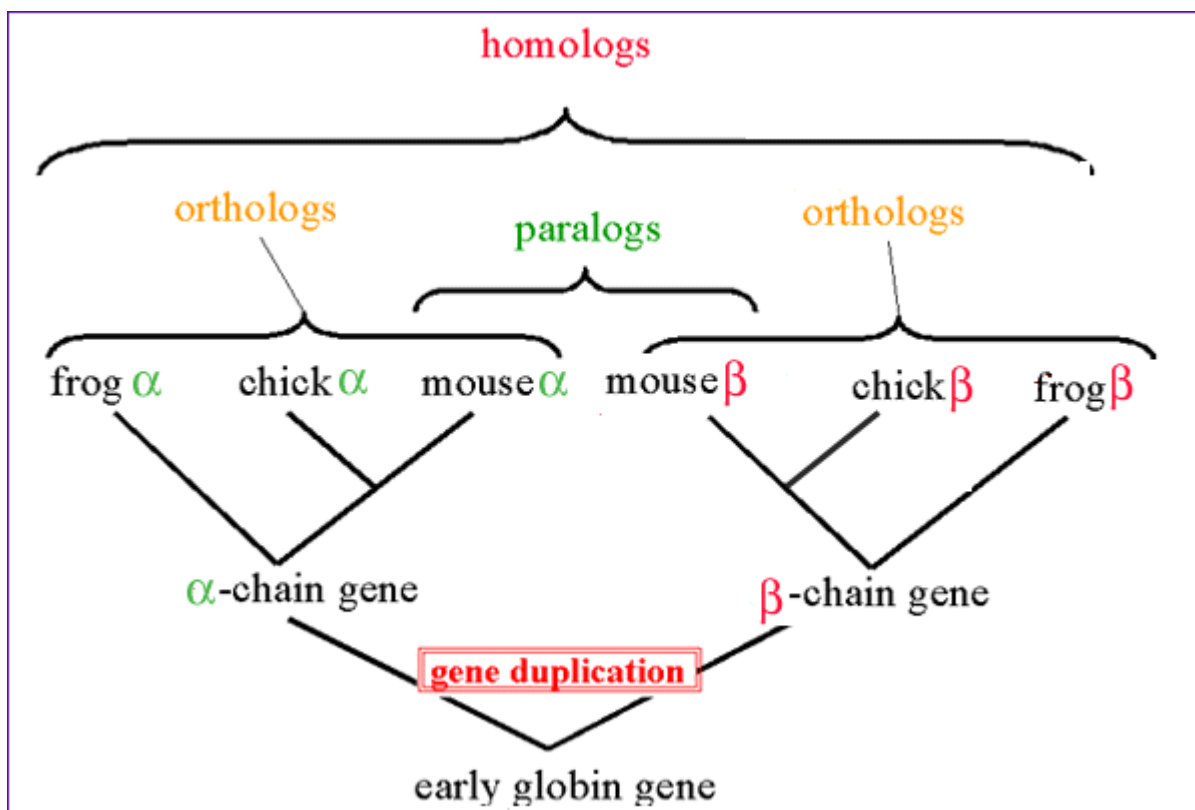
An alignment of three or more sequences with gaps inserted in the sequences such that residues with common structural positions and/or ancestral residues are aligned in the same column. Clustal W ([Thompson, Higgins, and Gibson, 1994](#)) is an example of a popular multiple sequence alignment program. The NCBI COBALT tool also produces multiple alignments of protein sequences ([Papadopoulos and Agarwala, 2007](#)).

optimal alignment

An alignment of two sequences with the highest possible score.

orthologs

Homologous biological components (genes, proteins, structures) in different species that arose from a single component present in the common ancestor of the species; orthologs may or may not have a similar function. Compare with paralogs.



p value

The probability of a chance alignment occurring with a particular score or a better score in a database search. The p value is calculated by relating the observed alignment score, *S*, to the expected distribution of HSP scores from comparisons of random sequences of the same length and composition as the query to the database. The most highly significant P values will be those close to 0. P values and E values are different ways of representing the significance of the alignment.

PAM

Percent Accepted Mutation (PAM) is unit introduced by Margaret Dayhoff and colleagues to quantify the amount of evolutionary change in a protein sequence. 1.0 PAM unit is the amount of evolution that will change, on average, 1% of amino acids in a protein sequence. A PAM(*x*) substitution matrix is a look-up table in which scores for each amino acid substitution have been calculated based on the frequency of that substitution in closely related proteins that have experienced a certain amount (*x*) of evolutionary divergence.

paralogs

Homologous biological components within a single species that arose by gene duplication. Compare with orthologs.

PHI-BLAST

Position Hit Initiated BLAST (PHI-BLAST) is a variant of PSI-BLAST that can focus the alignment and construction of the PSSM around a motif, which must be present in the query sequence and is provided as input to the program. Only database sequences that contain the motif in context will be included in the results. See also PSSM.

profile

A table that lists the frequencies of each amino acid in each position of protein sequence alignment. Frequencies are calculated from multiple alignments of sequences containing a domain of interest. See also PSSM.

proteomics

Systematic analysis of protein expression of normal and diseased tissues that involves the separation, identification and characterization of all of the proteins in a sample.

PSI-BLAST

Position-Specific Iterative BLAST (PSI-BLAST) is an iterative search using the protein BLAST algorithm. A profile is built after the initial search that is then used in subsequent searches. The process may be repeated, if desired, with new sequences found in each cycle used to refine the profile ([Altschul et al., 1997](#)).

PSSM

A Position-Specific Scoring Matrix (PSSM) is a profile that gives the log-odds score for finding a particular matching amino acid in a target sequence.

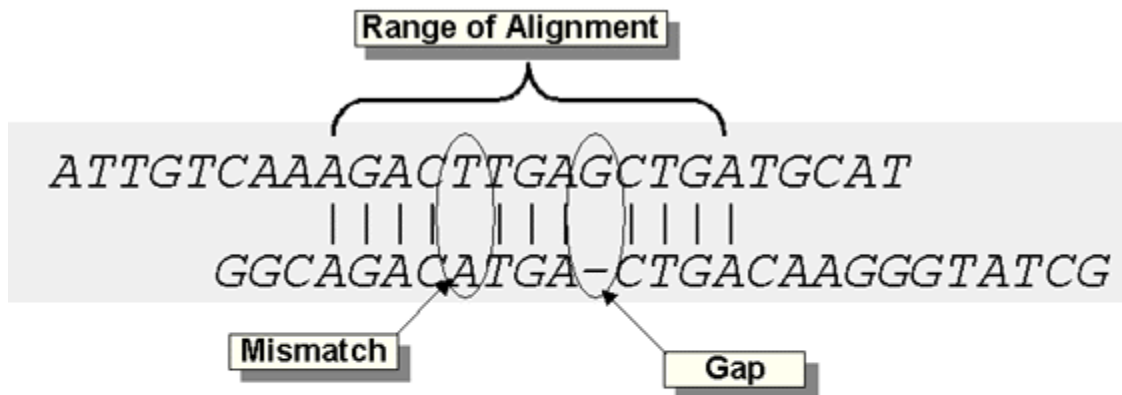
query

The input sequence (or other type of search term) to which all of the entries in a database are to be compared.

raw score

The score of an alignment, *S*, calculated as the sum of substitution and gap scores. Substitution scores are given by a look-up table (see PAM, BLOSUM). Gap scores are typically calculated as the sum of *G*, the gap opening

penalty and L, the gap extension penalty. For a gap of length n, the gap cost would be $G + Ln$. The choice of gap costs, G and L is empirical, but it is customary to choose a high value for G (10-15) and a low value for L (1-2).



$$S = \sum(\text{identities, mismatches}) - \sum(\text{gap penalties})$$

$$\text{Score} = \text{Max}(S)$$

similarity

The extent to which nucleotide or protein sequences are related. Similarity between two sequences can be expressed as percent sequence identity and/or percent positive substitutions.

SEG

A program for filtering low complexity regions in amino acid sequences ([Wootton and Federhen, 1996](#)).

Residues that have been masked are represented as "X" in an alignment. SEG filtering is no longer the default in the NCBI blastp service because of the use of compositional adjustments to estimate BLAST statistics. See composition-based statistics.

substitution

The presence of a non-identical amino acid at a given position in an alignment. If the aligned residues have similar physico-chemical properties or have a positive score in the governing scoring matrix the substitution is said to be conservative.

substitution scoring matrix

A scoring matrix containing values proportional to the probability that amino acid i mutates into amino acid j for all pairs of amino acids. Such matrices are constructed by assembling a large and diverse sample of verified pairwise alignments of protein sequences. If the sample is large enough, the resulting matrices should reflect the true probabilities of mutations occurring through a period of evolution. The BLOSUM matrices are examples of substitution scoring matrices.

unitary matrix

Also known as identity matrix. This is a scoring system in which only identical characters receive a positive score.