



## Standalone BLAST Setup for Unix

Tao Tao, Ph.D. <sup>✉1</sup>

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

The installation process from the disk image (`.dmg`) for Mac OSX and the Red Hat Package Manger (`.rpm`) for Linux requires administrative privileges and will not be discussed here. More information is available here: <http://www.ncbi.nlm.nih.gov/books/NBK569861/>.

## 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.

- 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](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 zxvpf 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

Table 2 continued from previous page.

Category	Program	Function
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 <https://www.ncbi.nlm.nih.gov/books/NBK569858/>.

## 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] [-evalue evalue] [-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 kubicae 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
```