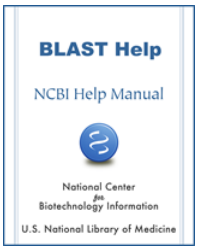
U.S. National Library of Medicine
National Center for Biotechnology Information

NLM Citation: Camacho C. Building a BLAST database with your (local) sequences. 2008 Jun 23 [Updated 2024 Aug 16]. In: BLAST® Command Line Applications User Manual [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2008-.
Bookshelf URL: https://www.ncbi.nlm.nih.gov/books/



Building a BLAST database with your (local) sequences

Christiam Camacho[1]

Created: June 23, 2008; Updated: August 16, 2024.

If you would like to search the BLAST databases NCBI offers, please see Get NCBI BLAST databases

The makeblastdb application produces BLAST databases from FASTA files. It is possible to use completely unstructured (or even blank) FASTA definition lines, but this is not the recommended procedure. Assigning a unique identifier to every sequence in the database allows you to retrieve the sequence by identifier and allows you to associate every sequence with a taxonomic node (through the taxid of the sequence). The unique identifier can be a simple string (as in the example below) or could be actual accession of the sequence if the sequence comes from a public database (e.g., GenBank). Being able to associate a database sequence with a taxonomic node is especially powerful for the version 5 databases that BLAST can use to limit the search by taxonomy. The identifier should begin right after the ">" sign on the definition line and contain no spaces and the -parse_seqids flag should be used. In general, you should not use a "|" (bar) in your identifier. The "|" (bar) is a reserved character for the NCBI FASTA ID parser and makeblastdb will return an error unless the bar is used in a specific manner described at https://ncbi.github.io/cxx-toolkit/pages/ch_demo#ch_demo.T5

An example FASTA file is:

```
$ cat test.fsa
>seq1
MSFSTKPLDMATWPDFAALVERHNGVWGGCWCMAFHAKGSGAVGNREAKEARVREGSTHAALVFDGSACVGWCQFGPTGE
LPRIKHLRAYEDGQAVLPDWRITCFFSDKAFRGKGVAAAALAGALAEIGRLGGGTVESYPEDAQGRTVAGAFLHNGTLAM
>seq2
MKAIDLKAEEKKRLIEGIQDFFYEERNEEIGIIAAEKALDFFLSGVGKLIYNKALDESKIWFSRRLEDISLDYELLYK
>seq3
MTLAAAAQSATWTFIDGDWYEGNVAILGPRSHAMWLGTSVFDGARWFEGVAPDLELHAARVNASAIALGLAPNMTPEQIV
GLTWDGLKKFDGKTAVYIRPMYWAEHGGYMGVPADPASTRFCLCLYESPMISPTGFSVTVSPFRRPTIETMPTNAKAGCL
YPNNGRAILEAKARGFDNALVLDMLGNVAETGSSNIFLVKDGHVLTPAPNGTFLSGITRSRTMTLLGDYGFRTTEKTLSV
RDFLEADEIFSTGNHSKVVPITRIEGRDLQPGPVAKKARELYWDWAHSASVG
>seq4
MRSFFHHVAAADPASFGVAQRVLTIPIKRAHIEVTHHLTKAEVDALIAAPNPRTSRGRRDRTFLLFLARTGARVSEATGV
NANDLQLERSHPQVLLRGKGRRDRVIPIPQDLARALTALLAEHGIANHEPRPIFIGARQERLTRFGATHIVRRAAAQAVT
IKPALAHKPISPHIFRHSLAMKLLQSGVDLLTIQAWLGHAQVATTHRYAAADVEMMRKGLEKAGVSGDLGLRFRPNDAVL
QLLTSI
>seq5
MTISRVCGSRTEAMLTNGQEIAMTSILKSTGAVALLLLYTLTANATSLMISPSSIERVAPDRAAVFHLRNQMDRPISIKV
RVFRWSQKGGVEKLEPTGDVVASPISAQLSPNGNRAVRVVRVSKEPLRSEEGYRVVIDEADPTRNTPEAESLSARHVLPV
LFRPPDVLGPEIELSLTRSDGWLMLVVENKGASRLRRSDVTLAQGSAGIARREGFVGYVLPGLTRHWRVGREDSYSGGIV
TVSANSSGGAIGEQLVVSGR
```

Author Affiliation: 1 Email: camacho@ncbi.nlm.nih.gov

```
>seq6
TTLLLQVPIGWGVLHQGGALVVLGFAIAHWRGFVGTYTRDTAIEMRD
```

An additional (optional) file mapping the identifiers to taxids (a number identifying a taxonomic node) may be used to associate each sequence with a taxonomic node.

```
$ cat test_map.txt
seq1 68287
seq2 2382161
seq3 68287
seq4 382
seq5 382
seq6 382
```

The taxid for a taxonomic node can be looked up via https://www.ncbi.nlm.nih.gov/Taxonomy/TaxIdentifier/tax_identifier.cgi. Additionally, the NCBI provides other resources. The files in https://ftp.ncbi.nlm.nih.gov/pub/taxonomy/accession2taxid/ provide a mapping from accession to taxid (useful if the sequences are from a public database). Information on other taxonomy files is available at https://ncbiinsights.ncbi.nlm.nih.gov/2018/02/22/new-taxonomy-files-available-with-lineage-type-and-host-information/ . Please note that the accession2taxid files may have a different format than what makeblastdb expects for its taxid_map argument. In some cases, there are more columns than expected and the header line must be removed, e.g.:

```
cut -f 2,3 prot.accession2taxid | sed 1d > protein-taxid-map.txt
```

Makeblastdb can be invoked for the FASTA and (optional) taxid mapping files as below. We use the -blastdb_version parameter to construct a version 5 database and the -taxid_map parameter to associate each sequence with a taxonomic node. Note that we also use -parse_seqids.

```
$ makeblastdb -in test.fsa -parse_seqids -blastdb_version 5 -taxid_map test_map.txt
-title "Cookbook demo" -dbtype prot


Building a new DB, current time: 02/06/2019 17:08:14
New DB name:   test.fsa
New DB title:  Cookbook demo
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 6 sequences in 0.00222588 seconds.
$
```

If you do add the taxids to your database, make sure you have the BLAST taxonomy data files (taxdb.bt[di]) which are available from https://ftp.ncbi.nlm.nih.gov/blast/db/ but also packaged with most BLAST databases distributed by the NCBI.

If all of the sequences in your database have the same taxid, you can simply use the -taxid flag on makeblastdb to associate all sequences with that taxid rather than needing to prepare a file.

For releases prior to BLAST+ 2.9.0, ad hoc identifiers (as shown in our example above) should be prefixed with "lcl|" (e.g., lcl|seq1 in place of seq1) for the taxid mapping file.

The NCBI makes databases that are searchable on the NCBI web site (such as nr, refseq_rna, and swissprot) available on its FTP site. It is better to download the preformatted databases rather than starting with FASTA. The databases on the FTP site contain taxonomic information for each sequence, include the identifier indices for lookups, and can be up to four times smaller than the FASTA. The original FASTA can be generated from the BLAST database using blastdbcmd.

Starting with the 2.10.0 release, makeblastdb produces version 5 databases by default, which uses LMDB. LMDB requires virtual memory (at least 600 GB, but 800 GB is recommended). Virtual memory is just that (virtual) and doesn't depend on the hardware in your system. In general, we recommend that BLAST users simply set the virtual memory to unlimited.