BLAST Help
BLAST Help
BLAST Help
BLAST Help

BLAST Command Line Applications User Manual

Christiam Camacho
NCBI
camacho@ncbi.nlm.nih.gov

---

1. Introduction

This manual documents the BLAST (Basic Local Alignment Search Tool) command line applications developed at the National Center for Biotechnology Information (NCBI). These applications have been revamped to provide an improved user interface, new features, and performance improvements compared to its counterparts in the NCBI C Toolkit. Hereafter we shall distinguish the C Toolkit BLAST command line applications from these command line applications by referring to the latter as the BLAST+ applications, which have been developed using the NCBI C++ Toolkit (http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=toolkit.TOC&depth=2).

2. Installation

The BLAST+ applications are distributed in executable and source code format. For the executable formats we provide installers as well as tarballs, the source code is only provided as a tarball. These are freely available at ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/

2.1 Windows

Download the executable installer ncbi-blast-2.2.18+.exe and double click on it. After accepting the license agreement, select the install location and click “Install” and then “Close”

2.2 MacOSX

Download the ncbi-blast-2.2.18+.dmg installer and double click on it. Double click the newly mounted ncbi-blast-2.2.18+ volume, double click on ncbi-blast-2.2.18+.pkg and follow the instructions in the installer. After accepting the license agreement, users without administrator privileges must change the install location by clicking on “Change Install Location”, selecting the proper partition, and choosing a folder (or creating a new one) for installation. Following this step, users with or without administrator privileges can click on “Install” and continue following the installer instructions.

2.3 RedHat Linux

Download the appropriate *.rpm file for your platform and either install or upgrade the ncbi-blast+ package as appropriate using the commands:

```
Install:
rpm -ivh ncbi-blast-2.2.18-1.x86_64.rpm
Upgrade:
rpm -Uvh ncbi-blast-2.2.18-1.x86_64.rpm
```

Note: one must have root privileges to run these commands.

2.4 Other Unix platforms

Download the tarball and expand it in the location of your choice.

BLAST Help
BLAST Help
BLAST Help
BLAST Help

2.5 Source tarball

Use this approach if you would like to build the BLAST+ applications yourself. Download the tarball, expand it and in the expanded directory type the following commands:

```
./configure --without-debug --with-mt --with-flat-makefile \
--with-build-root=ReleaseMT
cd c++/ReleaseMT/build
make -f Makefile.flat
```

The compiled executables will be found in c++/ReleaseMT/bin.

3. Quick start

3.1 For users of NCBI C Toolkit BLAST

The easiest way to get started using these command line applications is by means of the legacy_blast.pl Perl script which is bundled along with the BLAST+ applications. To utilize this script, simply prefix it to the invocation of the C toolkit BLAST command line application and append the --path option pointing to the installation directory of the BLAST+ applications. For example, instead of using

```
    blastall -i query -d nr -o blast.out
```

use

```
    legacy_blast.pl blastall -i query -d nr -o blast.out
--path /opt/blast/bin
```

For more details, refer to the section titled Backwards compatibility script.

3.2 For users of Web BLAST (http://blast.ncbi.nlm.nih.gov)

Users of Web BLAST can take advantage of the search strategies to quickly get started using the BLAST+ applications, as these intend to allow seamless integration between the Web and command line BLAST tools. For more details, refer to the section on BLAST search strategies.

3.3 For new users of BLAST

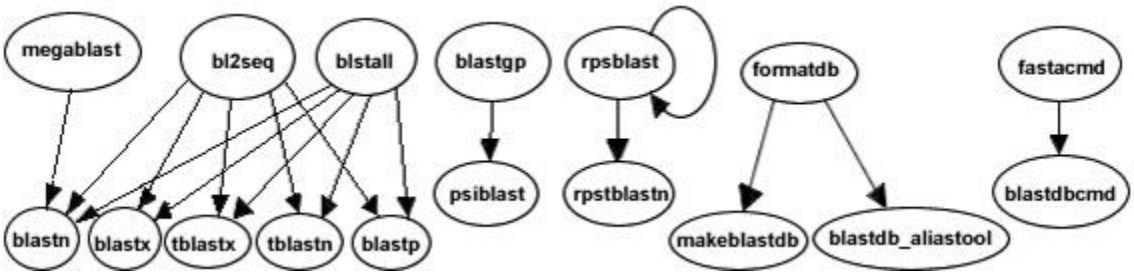
An introduction to BLAST is outside the scope of this manual, more information on this subject can be found on http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs. Nonetheless, new users will benefit from the examples in the cookbook as well as reading the user manual.

4. User manual

4.1 Functionality offered by BLAST+ applications

The functionality offered by the BLAST+ applications has been organized by program type, as to more closely resemble Web BLAST. The following graph depicts a correspondence between the NCBI C Toolkit BLAST command line applications and the BLAST+ applications:

BLAST Help
BLAST Help
BLAST Help
BLAST Help



As an example, to run a search of a nucleotide query (translated "on the fly" by BLAST) against a protein database one would use the blastx application instead of blastall. The blastx application will also work in "Blast2Sequences" mode (i.e.: accept FASTA sequences instead of a BLAST database as targets) and can also send BLAST searches over the network to the public NCBI server if desired.

The blastn, blastp, blastx, tblastx, tblastn, psiblast, rpsblast, and rpstblastn are considered search applications, as they execute a BLAST search, whereas makeblastdb, blastdb_aliastool, and blastdbcmd are considered BLAST database applications, as they either create or examine BLAST databases.

There is also a new set of sequence filtering applications described in the section Sequence filtering applications and an application to build database indices that greatly speed up megablast in some cases (see section titled Megablast indexed searches).

Please note that the NCBI C Toolkit applications seedtop and blastclust are not available in this release.

4.2 Common options

The following is a listing of options that are common to the majority of BLAST+ applications followed by a brief description of what they do:

4.2.1 db: Name of BLAST database to search the query against.

4.2.2 dbsize: Effective length of the database.

4.2.3 dbtype: Molecule type stored or to store in a BLAST database.

4.2.4 culling_limit: Ensures that more than the specified number of HSPs are not aligned to the same part of the query. This option was designed for searches with a lot of repetitive matches, but if possible it is probably more efficient to mask the query to remove the repetitive sequences.

4.2.5 evalue: Expectation value threshold for saving hits.

4.2.6 export_search_strategy: Name of the file where to save the search strategy (see section titled BLAST search strategies).

4.2.7 gapextend: Cost to extend a gap.

4.2.8 gapopen: Cost to open a gap.

4.2.9. gilist: File containing a list of GIs to restrict the BLAST database to search.

4.2.10 h: Displays the application's brief documentation.

BLAST Help

BLAST Help

BLAST Help

BLAST Help

4.2.11 help: Displays the application's detailed documentation.

4.2.12 html: Enables the generation of HTML output suitable for viewing in a web browser.

4.2.13 import_search_strategy: Name of the file where to read the search strategy to execute (see section titled BLAST search strategies).

4.2.14 lcase_masking: Interpret lowercase letters in query sequence(s) as masked.

4.2.15 mask_subjects: Filtering algorithm IDs to apply from the database specified as soft masking for subjects. These algorithm IDs for a given BLAST database can be obtained by invoking blastdbcmd with its -info flag (only shown if such filtering in the BLAST database is available). For more details see the section Masking in BLAST databases.

4.2.16 matrix: Name of the scoring matrix to use.

4.2.17 max_target_seqs: Maximum number of aligned sequences to keep from the BLAST database.

4.2.18 negative_gilist: File containing a list of GIs to exclude from the BLAST database.

4.2.19 num_alignments: Number of alignments to show in the BLAST output.

4.2.20 num_descriptions: Number of one-line descriptions to show in the BLAST output.

4.2.21 num_threads: Number of threads to use during the search.

4.2.22 out: Name of the file to write the application's output. Defaults to stdout.

4.2.23 outfmt: Allows for the specification of the search application's output format. A listing of the possible format types is available via the search application's -help option. If a custom output format is desired, this can be specified by providing a quoted string composed of the desired output format (tabular, tabular with comments, or comma-separated value), a space, and a space delimited list of output specifiers. The list of supported output specifiers is available via the -help command line option. Unsupported output specifiers will be ignored.

4.2.24 parse_deflines: Parse the query and subject deflines.

4.2.25 query: Name of the file containing the query sequence(s), or '-' if these are provided on standard input.

4.2.26 query_loc: Location of the first query sequence to search.

4.2.27 remote: Instructs the application to submit the search to NCBI for remote execution.

4.2.28 searchsp: Effective length of the search space.

4.2.29 seg: Arguments to SEG filtering algorithm (use 'no' to disable).

4.2.30 show_gis: Show NCBI GIs in deflines in the BLAST output.

4.2.31 soft_masking: Apply filtering locations as soft masks (i.e.: only when finding alignment seeds).

4.2.32 subject: Name of the file containing the subject sequence(s) to search.

BLAST Help BLAST Help BLAST Help BLAST Help

4.2.33 subject_loc: Location of the first subject sequence to search.

4.2.34 strand: Strand(s) of the query sequence to search.

4.2.35 threshold: Minimum word score such that the word is added to the BLAST lookup table.

4.2.36 ungapped: Perform ungapped alignments only.

4.2.37 version: Displays the application's version.

4.2.38 window_size: Size of the window for multiple hits algorithm, use 0 to specify 1-hit algorithm.

4.2.39 word_size: Word size for word finder algorithm.

4.2.40 xdrop_gap: X-dropoff value (in bits) for preliminary gapped extensions.

4.2.41 xdrop_gap_final: X-dropoff value (in bits) for final gapped alignment.

4.2.42 xdrop_ungap: X-dropoff value (in bits) for ungapped extensions.

4.3 Backwards compatibility script

The purpose of the legacy_blast.pl Perl script is to help users make the transition from the C Toolkit BLAST command line applications to the BLAST+ applications. This script produces its own documentation by invoking it without any arguments.

The legacy_blast.pl script supports two modes of operation, one in which the C Toolkit BLAST command line invocation is converted and executed on behalf of the user and another which solely displays the BLAST+ application equivalent to what was provided, without executing the command.

The first mode of operation is achieved by specifying the C Toolkit BLAST command line application invocation and optionally providing the --path argument after the command line to convert if the installation path for the BLAST+ applications differs from the default (available by invoking the script without arguments). See example in the first section of the Quick start.

The second mode of operation is achieved by specifying the C Toolkit BLAST command line application invocation and appending the --print_only command line option as follows:

```
$ ./legacy_blast.pl megablast -i query.fsa -d nt -o mb.out --
print_only
/opt/ncbi/blast/bin/blastn -query query.fsa -db "nt" -out mb.out
$
```

4.4 Exit codes

All BLAST+ applications have consistent exit codes to signify the exit status of the application. The possible exit codes along with their meaning are detailed in the table below:

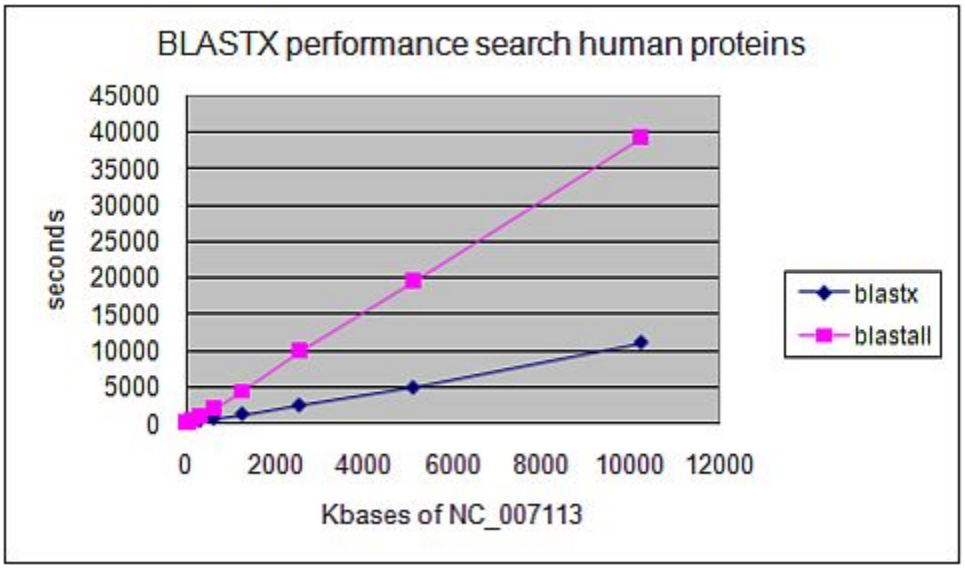
BLAST Help
BLAST Help
BLAST Help
BLAST Help

Exit Code	Meaning
0	Success
1	Error in query sequence(s) or BLAST options
2	Error in BLAST database
3	Error in BLAST engine
255	Unknown error

In the case of BLAST+ database applications, the possible exit codes are 0 (indicating success) and 1 (indicating failure).

4.5 Improvements over C Toolkit BLAST command line applications

4.5.1 Query splitting—This new feature in the BLAST+ applications provides substantial performance improvements, particularly for blastx searches. Below is a graph comparing the runtime of blastall and blastx when searching different size excerpts of NC_007113 (varying from 10 kbases to about 10 Mbases) against the human genome database (experiments performed in July 2008):



4.5.2 Tasks— The concept of tasks has been added to support the notion of commonly performed tasks via the -task command line option in blastn and blastp. The following tasks are currently available:

Program	Task Name	Description
blastp	blastp	Traditional BLASTP to compare a protein query to a protein database
	blastp-short	BLASTP optimized for queries shorter than 30 residues
blastn	blastn	Traditional BLASTN requiring an exact match of 11
	blastn-short	BLASTN program optimized for sequences shorter than 50 bases
	megablast	Traditional megablast used to find very similar (e.g., intraspecies or closely related species) sequences
	dc-megablast	Discontiguous megablast used to find more distant (e.g., interspecies) sequences

BLAST Help BLAST Help BLAST Help BLAST Help

4.5.3 Megablast indexed searches—Indexed searches for megablast are available and are faster than regular megablast. The application to generate the database indices is called makembindex, which is included in this distribution. More information about it can be found at ftp://ftp.ncbi.nlm.nih.gov/pub/agarwala/indexed_megablast/README.usage.

4.5.4 Partial sequence retrieval from BLAST databases—Improvements to the BLAST database reading module allow it to fetch only the relevant portions of the subject sequence that are needed in the gapped alignment stage, providing a substantial improvement in runtime. The following example compares 103 mouse EST sequences against the human genome shows (example run in July 2008 after the database had already been loaded into memory):

```
$ time megablast -d 9606_genomic -i est.500.in -r 2 -q -3
-F mL -m 9 -o old.out -W 11 -t 18 -G 5 -E 2
341.455u 65.242s 6:47.99 99.6% 0+0k 0+0io 0pf+0w
$ time blastn -task dc-megablast -db 9606_genomic -query
est.500.in -outfmt 7 -out new.out
218.540u 11.632s 3:50.53 99.8% 0+0k 0+0io 0pf+0w
```

Similar gains in performance should be expected in BLAST databases which contain very large sequences and many very short queries.

4.5.5 BLAST search strategies—BLAST search strategies are files which encode the inputs necessary to perform a BLAST search. The purpose of these files is to be able to seamlessly reproduce a BLAST search in various environments (Web BLAST, command line applications, etc).

4.5.5.1 Exporting search strategies on the Web BLAST

Click on "download" next to the RID/saved strategy in the "Recent Results" or "Saved Strategies" tabs.

4.5.5.2 Exporting search strategies with BLAST+ applications

Add the -export_search_strategy along with a file name to the command line options.

4.5.5.3 Importing search strategies on Web BLAST

Go to the "Saved Strategies" tab, click on "Browse" to select your search strategy file, then click on "View" to load it into the submission page.

4.5.5.4 Importing search strategies with BLAST+ applications

Add the -import_search_strategy along with a file name containing the search strategy file. Note that if provided, the –query and -db command line options will override the specifications of the search strategy file provided.

4.5.6 Negative GI lists—Negative GI lists are available on search applications and they provide a means to exclude GIs from a BLAST database search. For an example, see the cookbook.

4.5.7 Masking in BLAST databases —It is now possible to create BLAST databases which contain filtered sequences (also known as masking information or masks). This filtering

BLAST Help BLAST Help BLAST Help BLAST Help

information can be used as soft masking for the subject sequences. For a instructions on creating masked BLAST databases, please see the cookbook.

4.5.8 Custom output formats for BLAST searches—The BLAST+ search command line applications support custom output formats for the tabular and comma-separated value output formats. For more details see the common options as well as the cookbook.

4.5.9 Custom output formats to extract BLAST database data—blastdbcmd supports custom output formats to extract data from BLAST databases via the -outfmt command line option. For more details see the blastdbcmd options as well as the cookbook.

4.5.10 Improved software installation packages—The BLAST+ applications are available via Windows and MacOSX installers as well as RPMs (source and binary) and unix tarballs. For more details about these, refer to the installation section.

4.5.11 Sequence filtering applications—The BLAST+ applications include a new set of sequence filtering applications, namely segmasker, dustmasker, and windowmasker. segmasker is an application that identifies and masks low complexity regions of protein sequences. The dustmasker and windowmasker applications provide similar functionality for nucleotide sequences (see ftp://ftp.ncbi.nlm.nih.gov/pub/agarwala/dustmasker/README.dustmasker and ftp://ftp.ncbi.nlm.nih.gov/pub/agarwala/windowmasker/README.windowmasker for more information).

4.6 Options by program type

4.6.1 blastp—*4.6.1.1 task*: Specify the task to execute. For more details, refer to the section on tasks.

4.6.1.2 comp_based_stats: Select the appropriate composition based statistics mode (applicable only to blastp and tblastn). Available choices and references are available by invoking the application with -help option.

4.6.1.3 use_sw_tback: Instead of using the X-dropoff gapped alignment algorithm, use Smith-Waterman to compute locally optimal alignments

4.6.2 blastn—*4.6.2.1 task*: Specify the task to execute. For more details, refer to the section on tasks.

4.6.2.2 penalty: Penalty for a nucleotide mismatch.

4.6.2.3 reward: Reward for a nucleotide match.

4.6.2.4 use_index: Use a megablast database index.

4.6.2.5 index_name: Name of the megablast database index.

4.6.2.6 perc_identity: Minimum percent identity of matches to report

4.6.2.7 dust: Arguments to DUST filtering algorithm (use ‘no’ to disable).

4.6.2.8 filtering_db: Name of BLAST database containing filtering elements (i.e.: repeats)

4.6.2.9 window_masker_taxid: Enable WindowMasker filtering using a Taxonomic ID [experimental]

BLAST Help BLAST Help BLAST Help BLAST Help

4.6.2.10 window_masker_db: Enable WindowMasker filtering using this repeats database. [experimental]

4.6.2.11 no_greedy: Use non-greedy dynamic programming extension.

4.6.2.12 min_raw_gapped_score: Minimum raw gapped score to keep an alignment in the preliminary gapped and traceback stages.

4.6.2.13 template_type: Discontiguous megablast template type.

4.6.2.14 template_length: Discontiguous megablast template length.

4.6.3 blastx—*4.6.3.1 query_gencode*: Genetic code to use to translate the query sequence(s).

4.6.3.2 frame_shift_penalty: Frame shift penalty for use with out-of-frame gapped alignments

4.6.3.3 max_intron_length: Length of the largest intron allowed in a translated nucleotide sequence when linking multiple distinct alignments (a negative value disables linking).

4.6.4 tblastx—*4.6.4.1 db_gencode*: Genetic code to use to translate database/subjects.

4.6.4.2 max_intron_length: Identical to blastx.

4.6.5 tblastn—*4.6.5.1 db_gencode*: Identical to tblastx.

4.6.5.2 frame_shift_penalty: Identical to blastx.

4.6.5.3 max_intron_length: Identical to blastx.

4.6.5.4 comp_based_stats: Identical to blastp.

4.6.5.5 use_sw_tback: Identical to blastp.

4.6.5.6 in_pssm: Checkpoint file to initiate PSI-TBLASTN.

4.6.6 psiblast—*4.6.6.1 comp_based_stats*: Identical to blastp with the exception that only composition based statistics mode 1 is valid when a PSSM is the input (either when restarting from a checkpoint file or when performing multiple PSI-BLAST iterations).

4.6.6.2 gap_trigger: Number of bits to trigger gapping.

4.6.6.3 use_sw_tback: Identical to blastp.

4.6.6.4 num_iterations: Number of iterations to perform.

4.6.6.5 out_pssm: Name of the file to store checkpoint file containing a PSSM.

4.6.6.6 out_ascii_pssm: Name of the file to stire ASCII version of PSSM.

4.6.6.7 in_msa: Name of the file containing multiple sequence alignment to restart PSI-BLAST.

4.6.6.8 in_pssm: Checkpoint file to re-start PSI-BLAST.

4.6.6.9 pseudocount: Pseudo-count value used when constructing the PSSM.

BLAST Help

BLAST Help

BLAST Help

BLAST Help

4.6.6.10 inclusion_ethresh: E-value inclusion threshold for pairwise alignments to be considered to build the PSSM.

4.6.6.11 phi_pattern: Name of the file containing a PHI-BLAST pattern to search.

4.6.7 rpstblastn—*4.6.7.1 query_gencode*: Identical to blastx.

4.6.8 makeblastdb—This application serves as a replacement for formatdb.

4.6.8.1 in: Input file or BLAST database name to use as source; the data type is automatically detected. Note that multiple input files/BLAST databases can be provided (each must be separated by white space in a quoted string).

4.6.8.2 title: Title for the BLAST database to create

4.6.8.3 parse_seqids: Parse the Seq-id(s) in the FASTA input provided.

4.6.8.4 hash_index: Enables the creation of sequence hash values. These hash values can then be used to quickly determine if a given sequence data exists in this BLAST database.

4.6.8.5 mask_data: Comma-separated list of input files containing masking data to apply to the sequences being added to the BLAST database being created. For more information, see Masking in BLAST databases and the examples.

4.6.8.6 out: Name of the BLAST database to create.

4.6.8.7 max_file_sz: Maximum file size for any of the BLAST database files created.

4.6.8.8 logfile: Name of the file to which the program log should be redirected (stdout by default).

4.6.8.9 taxid: Taxonomy ID to assign to all sequences.

4.6.8.10 taxid_map: Name of file which provides a mapping of sequence IDs to taxonomy IDs.

4.6.9 blastdb_aliastool—This application replaces part of the functionality offered by formatdb. It supports two modes of operation:

1) Gi file conversion:

Converts a text file containing GIs (one per line) to a more efficient

binary format. This can be provided as an argument to the -gilist option

of the BLAST search command line binaries or to the -gilist option of

this program to create an alias file for a BLAST database (see below).

2) Alias file creation:

Creates an alias for a BLAST database and a GI list which restricts this

database. This is useful if one often searches a subset of a database

(e.g., based on organism or a curated list). The alias file makes the

BLAST Help BLAST Help BLAST Help BLAST Help

search appear as if one were searching a regular BLAST database rather

than the subset of one.

4.6.9.1 gi_file_in: Text file to convert, should contain one GI per line.

4.6.9.2 gi_file_out: File name of converted GI file

4.6.9.3 title: Title for BLAST database.

4.6.9.4 gilist: Name of the file containing the GIs to restrict the database provided in -db.

4.6.9.5 out: Identical to makeblastdb.

4.6.9.6 logfile: Identical to makeblastdb.

4.6.10 blastdbcmd—This application is the successor to fastacmd. The following are its supported options:

4.6.10.1 entry: A comma-delimited search string of sequence identifiers, or the keyword 'all' to select all sequences in the database.

4.6.10.2 entry_batch: Input file for batch processing, entries must be provided one per line. If input is provided on standard input, a '-' should be used to indicate this.

4.6.10.3 pig: PIG to retrieve.

4.6.10.4 info: Print BLAST database information (overrides all other options).

4.6.10.5 range: Selects the range of a sequence to extract.

4.6.10.6 strand: Strand of nucleotide sequence to extract.

4.6.10.7 outfmt: Output format string. For a list of available format specifiers, invoke the application with its -help option. Note that for all format specifiers except %f, each line of output will correspond to a single sequence.

4.6.10.8 target_only: The definition line of the sequence should contain target GI only.

4.6.10.9 get_dups: Retrieve duplicate accessions

4.6.10.10 line_length: Line length for output (applicable only with FASTA output format).

4.6.10.11 ctrl_a: Use Ctrl-A as the non-redundant defline separator (applicable only with FASTA output format).

5. Cookbook

5.1 Query a BLAST database with a GI, but exclude that GI from the results

```
$ blastdbcmd -entry all -db ecoli -dbtype nucl | head -1 | tee
exclude_me
1786181
$ blastn -db ecoli -negative_gilist exclude_me -show_gis
-num_alignments 0 \  -query exclude_me | grep `cat exclude_me`
```

```
Query= gi|1786181|gb|AE000111.1|AE000111
$
```

5.2 Create a masked BLAST database

To create a masked BLAST database, use one of the filtering programs (dustmasker, segmasker, or windowmasker) to create the masks and specify the maskinfo_asn1_bin output format (-outfmt maskinfo_asn1_bin) and then use the output of the filtering program when creating the BLAST database. This can be done in two ways:

Using an existing BLAST database

```
$ segmasker -in ecoli -out ecoli.masks.asn -infmt blastdb \
  -outfmt maskinfo_asn1_bin
$ makeblastdb -in ecoli -dbtype prot -parse_seqids \
  -title "Masked ecoli" -mask_data ecoli.masks.asn \
  -out ecoli.masked
Building a new DB, current time: 07/11/2008 13:28:16
New DB name:   ecoli.masked
New DB title:  Masked ecoli
Sequence type: Protein
Keep Linkouts: T
Keep MBits: T
Maximum file size: 1073741824B
Mask file: ecoli.masks.asn
Adding sequences from raw db source; added 4289 sequences
in 0.756123 seconds.
$ blastdbcmd -info -db ecoli.masked
Database: Masked ecoli
        4,289 sequences; 1,358,990 total residues
Date: Jul 11, 2008  1:28 PM     Longest sequence: 2,383 residues
Available filtering algorithms applied to database sequences:
Algorithm ID  Algorithm name
Options
    21        seg                 window=12; locut=2.2;
hicut=2.5
Volumes:
        /tmp/ecoli.masked
$
```

Using FASTA as input:

```
$ segmasker -in ecoli.fsa -out ecoli.masks.asn \
  -outfmt maskinfo_asn1_bin
$ makeblastdb -in ecoli.fsa -dbtype prot -parse_seqids \
  -title "Masked ecoli" -mask_data ecoli.masks.asn \
  -out masked.ecoli
Building a new DB, current time: 07/11/2008 13:31:33
New DB name:   masked.ecoli
New DB title:  Masked ecoli
Sequence type: Protein
Keep Linkouts: T
```


BLAST Help BLAST Help BLAST Help BLAST Help

BLAST Help BLAST Help BLAST Help BLAST Help

```
Keep MBits: T
Maximum file size: 1073741824B
Mask file: ecoli.masks.asn
Adding sequences from FASTA; added 4289 sequences in
1.11871 seconds.
$ blastdbcmd -info -db masked.ecoli
Database: Masked ecoli
        4,289 sequences; 1,358,990 total residues
Date: Jul 11, 2008  1:31 PM     Longest sequence: 2,383 residues
Available filtering algorithms applied to database sequences:
Algorithm ID  Algorithm name
Options
    21        seg                 window=12; locut=2.2;
hicut=2.5
Volumes:
        /tmp/masked.ecoli
$
```

5.3 Extract all human sequences from the nr database

Although one cannot select GIs by taxonomy from a database, a combination of unix command line tools will accomplish this:

```
$ blastdbcmd -db nr -entry all -outfmt "%g %T" | \
   awk ' { if ($2 == 9606) { print $1 } } ' | \
   blastdbcmd -db nr -entry_batch - -out human_sequences.txt
```

The first blastdbcmd invocation produces 2 entries per sequence (GI and taxonomy ID), the awk command selects from the output of that command those sequences which have a taxonomy ID of 9606 (human) and prints its GIs, and finally the second blastdbcmd invocation uses those GIs to print the sequence data for the human sequences in the nr database.

5.4 Display BLAST search results with custom output format

The following example shows how to display the results of a BLAST search using a custom output format, in this case, the tabular output format with comments is requested along with the query accession, subject accession, evalue, query start, query stop, subject start, and subject stop. For brevity, only the first 10 lines of output are shown:

```
$ echo 1786181 | ./blastn -db ecoli -outfmt '7 qacc sacc evalue
qstart qend sstart send'
# BLASTN 2.2.18+
# Query: gi|1786181|gb|AE000111.1|AE000111
# Database: ecoli
# Fields: query acc., subject acc., evalue, q. start, q. end, s.
 start, s. end
# 85 hits found
AE000111        AE000111        0.0     1       10596   1
10596
AE000111        AE000174        8e-30   5565    5671    6928    6821
AE000111        AE000394        1e-27   5587    5671    135     219
AE000111        AE000425        6e-26   5587    5671    8552    8468
```

```
AE000111        AE000171        3e-24   5587    5671    2214    2130
$
```

BLAST Help
BLAST Help
BLAST Help
BLAST Help