

GPU-BLAST 1.1: Installation instructions and user guide (February 2016)

GPU-BLAST is designed to accelerate the gapped and ungapped protein sequence alignment algorithms of the NCBI-BLAST implementation. GPU-BLAST is integrated into the NCBI-BLAST code and produces identical results. It has been tested on CentOS 5.5, 6.7, 7.2 and Fedora 10 with an NVIDIA Tesla C1060, an NVIDIA Tesla C2050 and an NVIDIA Tesla K40 GPU.

GPU-BLAST is free software.

Please cite the authors in any work or product based on this material:

Panagiotis D. Vouzis and Nikolaos V. Sahinidis, "GPU-BLAST: Using graphics processors to accelerate protein sequence alignment," Vol. 27, no. 2, pages 182-188, Bioinformatics, 2011 (Open Access).

For any questions and feedback about GPU-BLAST, contact sahinidis@cmu.edu.

I. Supported features

GPU-BLAST 1.1 supports protein alignment and is integrated in the "blastp" executable produced after installation. GPU-BLAST 1.1 does not support PSI BLAST.

II. Installation instructions

GPU-BLAST modifies NCBI-BLAST in order to add GPU functionality. These modifications do not alter the standard NCBI-BLAST when the GPU is not used. In addition, the results of GPU-BLAST are identical to those from NCBI-BLAST, irrespective of whether the GPU is used for calculations or not.

GPU-BLAST has been developed and tested on NVIDIA GPUs Tesla C1060, C2050 and K40. The present version requires a CUDA-capable NVIDIA GPU and the CUDA nvcc compiler.

The following sequence of commands are on a CentOS 7.2 with CUDA 7.5 and gcc v4.8.2. The login shell is bash.

The installation files are placed on an existing folder named "blast".

There are two possible ways to install GPU-BLAST: (i) install NCBI-BLAST and add GPU-BLAST later (Option 1), or (ii) directly install NCBI-BLAST and GPU-BLAST (Option 2).

Option 1: Install NCBI-BLAST (maybe it is already installed) and add GPU-BLAST later

NOTE: If you want to directly install both NCBI-BLAST and GPU-BLAST skip to option 2.

Step 1. NCBI-BLAST installation

NOTE: If you have already installed NCBI-BLAST, go to step 2.

```
[ploskas@anaximander blast]$ wget
ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.2.28/ncbi-blast-2.2.28+-src.tar.gz
--2016-02-09 13:13:32-- ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.2.28/ncbi-blast-
2.2.28+-src.tar.gz
=> 'ncbi-blast-2.2.28+-src.tar.gz'

Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 130.14.250.7, 2607:f220:41e:250::10
Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|130.14.250.7|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done. ==> PWD ... done.
==> TYPE I ... done. ==> CWD (1) /blast/executables/blast+/2.2.28 ... done.
==> SIZE ncbi-blast-2.2.28+-src.tar.gz ... 13468751
==> PASV ... done. ==> RETR ncbi-blast-2.2.28+-src.tar.gz ... done.
Length: 13468751 (13M) (unauthoritative)

100%[=====] 13,468,751 3.89MB/s in 3.3s
```

2016-02-09 13:13:36 (3.89 MB/s) - 'ncbi-blast-2.2.28+-src.tar.gz' saved [13468751]

```
[ploskas@anaximander blast]$ ls
ncbi-blast-2.2.28+-src.tar.gz
```

```
[ploskas@anaximander blast]$ tar -xvzf ncbi-blast-2.2.28+-src.tar.gz
[ploskas@anaximander blast]$ rm -r ncbi-blast-2.2.28+-src.tar.gz
[ploskas@anaximander blast]$ ls
ncbi-blast-2.2.28+-src
```

```
[ploskas@anaximander blast]$ cd ncbi-blast-2.2.28+-src/c++
[ploskas@anaximander c++]$ ./configure
[ploskas@anaximander c++]$ make
[ploskas@anaximander c++]$ make install
```

Depending on the version of your gcc compiler, NCBI-BLAST creates a directory inside "ncbi-blast-2.2.28+-src/c++". For example, if you use gcc v4.8.2 the directory will be named "GCC482-Debug64". Inside this directory, a directory named "bin" exists where all the needed executables are placed, e.g. makeblastdb and blastp. Please change accordingly the name of this folder to the following commands.

Step 2. GPU-BLAST installation

Adding GPU functionality to an existing installation of NCBI-BLAST.

```
[ploskas@anaximander blast]$ ls
ncbi-blast-2.2.28+-src
```

```
[ploskas@anaximander blast]$ wget http://thales.cheme.cmu.edu/gpublast/gpu-blast-1.1_ncbi-
blast-2.2.28.tar.gz
--2016-02-09 13:56:41-- http://thales.cheme.cmu.edu/gpublast/gpu-blast-1.1_ncbi-blast-
2.2.28.tar.gz
Resolving thales.cheme.cmu.edu (thales.cheme.cmu.edu)... 128.2.55.81
Connecting to thales.cheme.cmu.edu (thales.cheme.cmu.edu)|128.2.55.81|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 252460 (247K) [application/x-gzip]
Saving to: 'gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz'

100%[=====] 252,460  --.-K/s in 0.02s
```

```
2016-02-09 13:56:41 (11.5 MB/s) - 'gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz' saved [252460/252460]
```

```
[ploskas@anaximander blast]$ ls
gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz ncbi-blast-2.2.28+-src
```

```
[ploskas@anaximander blast]$ tar -xzf gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz  
[ploskas@anaximander blast]$ rm gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz  
[ploskas@anaximander blast]$ ls  
gpu_blast install ncbi-blast-2.2.28+-src README
```

```
[ploskas@anaximander blast]$ sudo sh ./install  
Do you want to install GPU-BLAST on an existing installation of "blastp" [yes/no]  
yes: you will be asked for the installation directory of the "blastp" executable  
no: will download and install "ncbi-blast-2.2.28+-src"  
yes  
Please input the installation directory of "blastp" of "ncbi-blast-2.2.28+-src"  
/home/ploskas/blast/ncbi-blast-2.2.28+-src/c++/GCC482-Debug64/bin/  
"blastp" version 2.2.28+ is compatible  
Continuing with the installation of GPU-BLAST...
```

Modifying NCBI BLAST files

Compiling CUDA code

....

Building NCBI BLAST with GPU BLAST

Option 2: Directly install both NCBI-BLAST and GPU-BLAST

```
[ploskas@anaximander blast]$ wget http://thales.cheme.cmu.edu/gpublast/gpu-blast-1.1_ncbi-  
blast-2.2.28.tar.gz  
--2016-02-09 15:15:34-- http://thales.cheme.cmu.edu/gpublast/gpu-blast-1.1_ncbi-blast-  
2.2.28.tar.gz  
Resolving thales.cheme.cmu.edu (thales.cheme.cmu.edu)... 128.2.55.81  
Connecting to thales.cheme.cmu.edu (thales.cheme.cmu.edu)|128.2.55.81|:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
Length: 252460 (247K) [application/x-gzip]
Saving to: 'gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz'
```

```
100%[=====>] 252,460  --.-K/s in 0.02s
```

```
2016-02-09 15:15:34 (11.8 MB/s) - 'gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz' saved [252460/252460]
```

```
[ploskas@anaximander blast]$ ls
gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz
```

```
[ploskas@anaximander blast]$ tar -xzf gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz
[ploskas@anaximander blast]$ ls
gpu_blast gpu-blast-1.1_ncbi-blast-2.2.28.tar.gz install README
```

```
[ploskas@anaximander blast]$ sh ./install
```

```
Do you want to install GPU-BLAST on an existing installation of "blastp" [yes/no]
```

```
yes: you will be asked for the installation directory of the "blastp" executable
```

```
no: will download and install "ncbi-blast-2.2.28+-src"
```

```
no
```

```
Continuing with the downloading of ncbi-blast-2.2.28+-src...
```

```
Downloading NBCI BLAST
```

```
--2016-02-09 15:29:18-- ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.2.28/ncbi-blast-
2.2.28+-src.tar.gz
```

```
=> 'ncbi-blast-2.2.28+-src.tar.gz'
```

```
Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 130.14.250.11, 2607:f220:41e:250::7
```

```
Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|130.14.250.11|:21... connected.
```

```
Logging in as anonymous ... Logged in!
```

```
==> SYST ... done. ==> PWD ... done.
```

```
==> TYPE I ... done. ==> CWD (1) /blast/executables/blast+/2.2.28 ... done.
```

```
==> SIZE ncbi-blast-2.2.28+-src.tar.gz ... 13468751
```

```
==> PASV ... done. ==> RETR ncbi-blast-2.2.28+-src.tar.gz ... done.
```

```
Length: 13468751 (13M) (unauthoritative)
```

100%[=====] 13,468,751 8.39MB/s in 1.5s

2016-02-09 15:29:20 (8.39 MB/s) - 'ncbi-blast-2.2.28+-src.tar.gz' saved [13468751]

Extracting NCBI BLAST

Configuring ncbi-blast-2.2.28+-src with options:

```
../configure --without-debug --with-mt --without-sybase --without-ftds --without-fastcgi --without-ncbi-c --without-sssdb --without-sss --without-geo --without-sp --without-orbacus --without-boost
```

If you want to change these options edit the file "install", delete the directory ncbi-blast-2.2.28+-src, and rerun install

Modifying NCBI BLAST files

Compiling CUDA code

Building NCBI BLAST with GPU BLAST

```
[ploskas@anaximander blast]$ ls
configure.output gpu_blast.output modify.output      ncbi-blast-2.2.28+-src.tar.gz README
gpu_blast      install      ncbi-blast-2.2.28+-src ncbi_blast.output
```

Depending on the version of your gcc compiler, NCBI-BLAST creates a directory inside ncbi-blast-2.2.28+-src/c++. For example, if you use gcc v4.8.2 the directory will be named "GCC482-ReleaseMT64". Inside this directory, a directory named bin exists where all the needed executables are placed, e.g. makeblastdb and blastp.

NOTE: if you used Option 1 to install GPU-BLAST after installing NCBI-BLAST, the name of this folder would be "GCC482-Debug64". Please change accordingly the name of this folder to the following commands.

III. How to use GPU-BLAST

If the above process is successful, the NCBI-BLAST installed will offer the additional option of using GPU-BLAST. The interface of GPU-BLAST is identical to the original NCBI-BLAST interface with the following additional options for "blastp":

```
*** GPU options
-gpu <Boolean>
    Use GPU for blastp
    Default = 'F'

-gpu_threads <Integer, 1..1024>
    Number of GPU threads per block
    Default = '64'

-gpu_blocks <Integer, 1..65536>
    Number of GPU block per grid
    Default = '512'

-method <Integer, 1..2>
    Method to be used
    1 = for GPU-based sequence alignment (default),
    2 = for GPU database creation
    Default = '1'
```

* Incompatible with: num_threads

Typing "./blastp -help" will print the above options towards the end of the output.

GPU-BLAST also adds the option "-sort_volumes" in the "makeblastdb" executable in "/ncbi-blast-2.2.28+-src/c++/GCC482-ReleaseMT64/bin". "makeblastdb" is an executable that formats a FASTA database to the appropriate format to be used by NCBI-BLAST. The option "-sort_volumes" sorts the database volumes produced according to length.

NOTE: "makeblastdb" has the option to split input databases in chunks, called volumes, according to the users choice.

When you type "./makeblastdb -help", you will see the additional option listed in comparison to the original NCBI-BLAST "makeblastdb"

*** Miscellaneous options

-sort_volumes

Sort the sequences according to length in each volume

The following example uses the protein FASTA formatted database, called "env_nr". Create a directory "database" inside directory "blast" and download the database env_nr inside this directory.

```
[ploskas@anaximander blast]$ mkdir database  
[ploskas@anaximander blast]$ cd database  
[ploskas@anaximander database]$ wget ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/env_nr.gz  
[ploskas@anaximander database]$ gunzip env_nr.gz  
[ploskas@anaximander database]$ rm enc_nr.gz
```

Then, create a directory "queries" inside directory "blast" and download some sample queries inside this folder.

```
[ploskas@anaximander blast]$ cd ..  
[ploskas@anaximander blast]$ mkdir queries  
[ploskas@anaximander blast]$ cd queries
```

```
[ploskas@anaximander queries]$ wget http://thales.cheme.cmu.edu/gpublast/queries.tar.gz
[ploskas@anaximander database]$ tar -xzf queries.tar.gz
[ploskas@anaximander database]$ rm queries.tar.gz
```

The tree structure of the folder "blast" should look like:

```
blast
|-- database
|   |-- env_nr
|-- gpublast
|   |-- ncbi_blast_files
|       |-- .
|       |-- .
|       |-- .
|   |-- gpu_blastp.c
|   |-- .
|   |-- .
|   |-- .
|   |-- .
|   |-- ncbi-blast-2.2.28+-src/
|       |-- c++
|           |-- compilers
|               |-- .
|               |-- .
|               |-- .
|               |-- .
|               |-- GCC482-ReleaseMT64
|                   |-- bin
|                   |-- blastp
|                   |-- makeblastdb
|                   |-- .
|                   |-- .
|                   |-- .
|                   |-- .
|                   |-- build
|                   |-- .
|                   |-- .
|                   |-- .
|                   |-- .
```

```
| | | |-- inc
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- lib
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- status
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- .
| | |-- include
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- .
| | |-- scripts
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- .
| | | |-- config.log
| | |-- configure
| | |-- Makefile
|-- queries
| |-- SequenceLength_00000002.txt
| |-- .
| |-- .
| |-- .
| |-- SequenceLength_00004998.txt
|-- install
```

|-- README

To use GPU-BLAST, first sort the protein database according to sequence length, create a database for the GPU and execute blastp on the GPU.

Step 1. Sorting a database

To sort a database use the option "-sort_volumes" with "makeblastdb".

"-sort_volumes" works only with protein FASTA-formatted databases. Assuming the database "env_nr" is saved in the directory "/blast/database", type:

```
[ploskas@anaximander blast]$ cd ncbi-blast-2.2.28+-src/c++/GCC482-ReleaseMT64/bin/  
[ploskas@anaximander bin]$ ./makeblastdb -in ../../../../database/env_nr -out  
../../../../database/sorted_env_nr -dbtype prot -sort_volumes -max_file_sz 500MB
```

Building a new DB, current time: 02/09/2016 16:16:57

New DB name: ../../../../../../database/sorted_env_nr

New DB title: ../../../../../../database/env_nr

Sequence type: Protein

Keep Linkouts: T

Keep MBits: T

Maximum file size: 500000000B

Sorting 2538598 sequences

Writing to Volume

Done Writing to Database

Sorting 2614567 sequences

Writing to Volume

Done Writing to Database

Adding sequences from FASTA; added 6964945 sequences in 277.507 seconds.

Sorting 1811780 sequences

Writing to Volume

Done Writing to Database

This will produce the following files inside the database directory:

```
[ploskas@anaximander bin]$ ls ../../../../database  
env_nr sorted_env_nr.00.phr sorted_env_nr.00.pin sorted_env_nr.00.psq  
sorted_env_nr.01.phr sorted_env_nr.01.pin sorted_env_nr.01.psq sorted_env_nr.02.phr  
sorted_env_nr.02.pin sorted_env_nr.02.psq sorted_env_nr.pal
```

Since this process only sorts the database, the "sorted_env_nr" will produce identical alignments with the original "env_nr", no matter whether GPU-BLAST is used or not.

NOTE: By using the option "-sort_volumes", "makeblastdb" stores in a vector each volume of the produced database in order to sort it. This may require large amounts of memory, depending on the database size and the options used with "makeblastdb". If, by using "-sort_volumes", the program runs out of memory, consider using the option "-max_file_sz" to produce smaller database volumes compared to the default NCBI size which is 1 GB.

```
./makeblastdb -in ../../../../../../database/env_nr -out ../../../../../../database/sorted_env_nr  
-dbtype prot -sort_volumes -max_file_sz 200MB"
```

to produce database volumes with size up to 200 MB.

Step 2. Creating a GPU database

To create the database in the appropriate GPU-BLAST format from the sorted database created in the previous step, treat the sorted database as any database that you would use to align a sequence against; i.e., format and sort the input database with "makeblastdb" (see step 1). Then, execute "./blastp" with the added options "-gpu T -method 2 -gpu_blocks <Integer, 1..65536> -gpu_threads <Integer, 1..1024>".

For example, to create a database for GPU-BLAST of the "sorted_env_nr", type

```
[ploskas@anaximander bin]$ ./blastp -query ../../../../../../queries/SequenceLength_00000100.txt -db  
../../../../database/sorted_env_nr -gpu t -method 2 -gpu_blocks 256 -gpu_threads 32  
../../../../database/sorted_env_nr.00.pin: 2538598  
../../../../database/sorted_env_nr.01.pin: 2614567
```

```
../../../../database/sorted_env_nr.02.pin: 1811780
num_volumes = 3
Done with creating the GPU Database file (../../../../database/sorted_env_nr.00.gpu)
Done with creating the GPU Database file (../../../../database/sorted_env_nr.01.gpu)
Done with creating the GPU Database file (../../../../database/sorted_env_nr.02.gpu)
BLASTP 2.2.28+
```

Reference: Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Reference for composition-based statistics: Alejandro A. Schaffer, L. Aravind, Thomas L. Madden, Sergei Shavirin, John L. Spouge, Yuri I. Wolf, Eugene V. Koonin, and Stephen F. Altschul (2001), "Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements", Nucleic Acids Res. 29:2994-3005.

Database: ../../../../../../database/env_nr
6,964,945 sequences; 1,385,475,744 total letters

The options "-gpu_blocks" and "-gpu_threads" are optional when creating the GPU database. If they are not specified, their default values are 512 and 64, respectively.

NOTE: the option "-method" is incompatible with "-num_threads".

This will create the files "sorted_env_nr.gpu" and "sorted_env_nr.gpuinfo" in the same directory with "sorted_env_nr".

Step 3. Executing GPU-BLAST

You are now ready to use GPU-BLAST. To search the database with the query "SequenceLength_00000100.txt", type

```
[ploskas@anaximander bin]$ ./blastp -query ../../queries/SequenceLength_00000100.txt -db  
../../database/sorted_env_nr -gpu t  
BLASTP 2.2.28+
```

Reference: Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Reference for composition-based statistics: Alejandro A. Schaffer, L. Aravind, Thomas L. Madden, Sergei Shavirin, John L. Spouge, Yuri I. Wolf, Eugene V. Koonin, and Stephen F. Altschul (2001), "Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements", Nucleic Acids Res. 29:2994-3005.

Database: ../../database/env_nr
6,964,945 sequences; 1,385,475,744 total letters

Query= sp|Q91G65|032R_IIV6 Uncharacterized protein 032R OS=Invertebrate
iridescent virus 6 GN=IIV6-032R PE=4 SV=1

Length=100

Score E

Sequences producing significant alignments:	(Bits)	Value
ECC82777.1 hypothetical protein GOS_5867893, partial [marine me...]	61.6	6e-11
GAH00199.1 unnamed protein product [marine sediment metagenome]	48.5	3e-06
EBG35402.1 hypothetical protein GOS_9447047 [marine metagenome]	47.4	5e-06
OIR15560.1 hypothetical protein GALL_38830 [mine drainage metag...]	40.4	0.003
GAF70722.1 unnamed protein product [marine sediment metagenome]	38.9	0.014
EDE03225.1 hypothetical protein GOS_1201654 [marine metagenome]	38.1	0.021
ECN02724.1 hypothetical protein GOS_4064323, partial [marine me...]	38.1	0.026
KKK92759.1 hypothetical protein LCGC14_2699710, partial [marine...]	36.6	0.088
OIR08757.1 hypothetical protein GALL_91410 [mine drainage metag...]	35.0	0.25
KKK91461.1 hypothetical protein LCGC14_2712760 [marine sediment...]	33.9	0.77
EBJ79500.1 hypothetical protein GOS_8838565, partial [marine me...]	32.0	4.3
ECT65541.1 hypothetical protein GOS_5098259, partial [marine me...]	31.6	5.0
EBB41044.1 hypothetical protein GOS_238181 [marine metagenome]	31.6	5.3
EBB29372.1 hypothetical protein GOS_257812, partial [marine met...]	31.2	7.6
KKM82403.1 hypothetical protein LCGC14_1319930 [marine sediment...]	31.2	8.0
EBE52939.1 hypothetical protein GOS_9748365, partial [marine me...]	31.2	8.0
CBI09278.1 conserved hypothetical protein [mine drainage metage...]	31.2	8.5
EBQ63088.1 hypothetical protein GOS_7721906, partial [marine me...]	31.2	9.5
EDI73951.1 hypothetical protein GOS_382468, partial [marine met...]	31.2	9.5
EDG96131.1 hypothetical protein GOS_690898, partial [marine met...]	31.2	9.6
EBV21546.1 hypothetical protein GOS_6939265, partial [marine me...]	30.8	9.7

> ECC82777.1 hypothetical protein GOS_5867893, partial [marine metagenome]

Length=137

Score = 61.6 bits (148), Expect = 6e-11, Method: Compositional matrix adjust.

Identities = 32/90 (36%), Positives = 53/90 (59%), Gaps = 1/90 (1%)

Query 10 KKKEVGQAVLQ-KERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCCCKLAIPKI 68

++ VGQ+A+L+ ++IFY+VTK K Y KP L+ ++ L + ++A+PKI

Sbjct 46 QRAGVGQIAILKCHGQIIFYLVTKSYYEKPNLWSIHASLLQLRRYMDHCLTQIAMPKI 105

Query 69 GCCLDRLYWKTVKNIIDKLCKKGIEVVVY 98

GC LDR+ W+ V+N++ I V +Y

Sbjct 106 GCGLDRMAWEDVENLLWSVFDNHNILVTIY 135

> GAH00199.1 unnamed protein product [marine sediment metagenome]

Length=156

Score = 48.5 bits (114), Expect = 3e-06, Method: Compositional matrix adjust.

Identities = 22/92 (24%), Positives = 52/92 (57%), Gaps = 1/92 (1%)

Query 10 KKKEVGQAVLQKERL-IFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCCKLAIPKI 68

+ ++G L+++ + IFY+VTK+ + KPT + ++++++ ++ +++P+I

Sbjct 65 QPHQIGDTPYLERDGIVIFYLVTKKLYHQKPTYDSIEKSLETRDIMIQKHIHDISMPRI 124

Query 69 GCCLDRLYWKTVKNIIDKLCKKGIEVVVYI 100

GC LD+ W ++ I+ I++ VY +

Sbjct 125 GCGLDKKNWTEIEKILGRVFKDIDIKVYSL 156

> EBG35402.1 hypothetical protein GOS_9447047 [marine metagenome]

Length=140

Score = 47.4 bits (111), Expect = 5e-06, Method: Compositional matrix adjust.

Identities = 24/72 (33%), Positives = 39/72 (54%), Gaps = 0/72 (0%)

Query 26 IFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCCKLAIPKIGCCLDRLYWKTVKNIII 85

++ ++TK+K KPT ++ + N L ++A+PKIGC LD+L W V+ II

Sbjct 66 VYNLITKKYSGKPTYQTIRMSLVMRNHALANNTRIAMPKIGCGLDKLQWAMVRAIIH 125

Query 86 DKLCKKGIEVV 97

+ IE+ V

Sbjct 126 ELFEDTDIEIRV 137

> OIR15560.1 hypothetical protein GALL_38830 [mine drainage metagenome]

Length=163

Score = 40.4 bits (93), Expect = 0.003, Method: Compositional matrix adjust.

Identities = 28/102 (27%), Positives = 48/102 (47%), Gaps = 8/102 (8%)

Query 5 KFCYNKKKEVGQAVLQKE--RLIFYIVTKEKSYL---KP---TLANFSNAIDSPLYNECL 56

+C + + G++ R + + T++ +Y KP TL + +++++ +L N

Sbjct 48 HYCQTQHPKSGELWTWMSADGRYLVLNLFTQDAAYAHGSKPGNATLHHINHSLHALRNFVQ 107

Query 57 LRKCCKLAIPKIGCCLDRLYWKTVKNIIDKLCKKGIEVVVY 98

K LA+P++ C + L W VK II L GI V VY

Sbjct 108 KEKV GSL ALPRLACGVGGLNWDDVKLIIEKHLGDLGIPVYVY 149

> GAF70722.1 unnamed protein product [marine sediment metagenome]

Length=199

Score = 38.9 bits (89), Expect = 0.014, Method: Compositional matrix adjust.

Identities = 28/100 (28%), Positives = 45/100 (45%), Gaps = 6/100 (6%)

Query 5 KFCYNKKKEVG----QAVLQKERLIFYIVTKEKSYLKPTLANFSNAIDSLYNECLLRK 59

K C +KK G Q+ +L + + I TK K L + +++L E

Sbjct 48 KVCKDKKLHPGMVYTYQLLILGRTQYIINFPTKRHWKGSKLEDIQQGLEALAQEIKRLG 107

Query 60 CCKLAIPKIGCCLDRLYWKTVKNIIDKLCK-KGIEVVVY 98

++IP +GC L L W+TV+ I+ L +EV +Y

Sbjct 108 IRSISIPPLGCGLGLLNWETVRPIMESALVSLTDVEVDIY 147

> EDE03225.1 hypothetical protein GOS_1201654 [marine metagenome]

Length=158

Score = 38.1 bits (87), Expect = 0.021, Method: Compositional matrix adjust.

Identities = 21/79 (27%), Positives = 39/79 (49%), Gaps = 0/79 (0%)

Query 20 LQKERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCKLAIPIKGCCCLDRWKT 79

++ E IF + T++ K TL + ++ S+ + + IP+IG L LW

Sbjct 66 VEGENTIFNLGTQKTWRTKATLDAVATSLGMLAIAQAKGIKCGIPRIGAGLGLVWDD 125

Query 80 VKNIIIDKLCKKGIEVVVY 98

VK II ++ K + ++V+

Sbjct 126 VKAIIQEEAQKSDVTLIVF 144

> ECN02724.1 hypothetical protein GOS_4064323, partial [marine metagenome]

Length=206

Score = 38.1 bits (87), Expect = 0.026, Method: Compositional matrix adjust.

Identities = 20/64 (31%), Positives = 33/64 (52%), Gaps = 8/64 (13%)

Query 43 NFSNAIDSPLYNECLLRKCKLAIPIKGCCCLDRW-----KTVKNIIIDKLCKKGIEV 95

N N+ D YN L+ K + IPK+G D +Y + ++N +++KL KGI

Sbjct 86 NRKNSAD-FYNHLLVEKIHDVGIPKVGDNRDHVYQMYTITVEEKIRNEVVEKLNSKGIGA 144

Query 96 VVYY 99

V++

Sbjct 145 SVHF 148

> KKK92759.1 hypothetical protein LCGC14_2699710, partial [marine sediment metagenome]

Length=226

Score = 36.6 bits (83), Expect = 0.088, Method: Compositional matrix adjust.

Identities = 19/65 (29%), Positives = 30/65 (46%), Gaps = 0/65 (0%)

Query 20 LQKERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCKLAIPKIGCCLDRLYWKT 79

LQ R + TK K + + + + + SL E R +AIP +GC L LW

Sbjct 68 LQLPRYVINFPTKRHWKGKSRIEDIESGLNSLITEVRQRNIESIAIPPLGCGLGGLDWGV 127

Query 80 VKNII 84

V+ ++

Sbjct 128 VRPMV 132

> OIR08757.1 hypothetical protein GALL_91410 [mine drainage metagenome]

Length=162

Score = 35.0 bits (79), Expect = 0.25, Method: Compositional matrix adjust.

Identities = 23/102 (23%), Positives = 48/102 (47%), Gaps = 8/102 (8%)

Query 5 KFCYNKKKEVGQAVLQKE--RLIFYIVTKEKSY--LKP---TLANFSNAIDSPLYNECL 56

+C + + G++ R + + T++ +Y KP TL++ ++ + +L +

Sbjct 48 HYCQTQHPKPGELWTWMSADGRYLVNLFTQDGAYDHGSKPGHATLSHVNVHTLHALRSFAQ 107

Query 57 LRKCKLAIPKIGCCLDRLYWKTVKNIIIDKLCKKGIEVVVY 98

K LA+P++ C ++ L W VK +I L I + +Y

Sbjct 108 KEKPSLALPRLSCGINGLDWNDVKPLIEKHLGDLNIPYIY 149

> KKK91461.1 hypothetical protein LCGC14_2712760 [marine sediment

metagenome]

Length=150

Score = 33.9 bits (76), Expect = 0.77, Method: Compositional matrix adjust.

Identities = 22/82 (27%), Positives = 38/82 (46%), Gaps = 1/82 (1%)

Query 17 QAVLQKERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCCCKLAIPKIGCCLDRY 76

V + RL+ + V K + + L ++ + L C K K+A+P+ GC RL

Sbjct 65 VHTFGQYRLMTPV-KYHWHEEADLDLIRHSCEQLRGLCYTLKMAKVAMPRPGCGNGRLD 123

Query 77 WKTVKNIIDKLCKKGIEVVVY 98

W V+ ++ + L E +VY

Sbjct 124 WDDVRPVLEETLGDSRTEFLVY 145

> EBJ79500.1 hypothetical protein GOS_8838565, partial [marine

metagenome]

Length=419

Score = 32.0 bits (71), Expect = 4.3, Method: Composition-based stats.

Identities = 18/67 (27%), Positives = 34/67 (51%), Gaps = 0/67 (0%)

Query 16 QAVLQKERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCCCKLAIPKIGCCLDRY 75

QVA+ + +F V KE++Y LA+ SNA+ ++ + ++ C K + +

Sbjct 198 QVAIALENAQLFEEVAKERTYSDSMLASMSNAVVTINEDGIIATCNKAGLKIFRVSAQEI 257

Query 76 YWKTVKN 82

KTV++

Sbjct 258 IGKTVED 264

> ECT65541.1 hypothetical protein GOS_5098259, partial [marine

metagenome]

Length=145

Score = 31.6 bits (70), Expect = 5.0, Method: Compositional matrix adjust.

Identities = 15/47 (32%), Positives = 27/47 (57%), Gaps = 0/47 (0%)

Query 16 QAVLQKERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCC 62

QVA+ + +F V KE++Y LA+ SNA+ ++ + ++ C K

Sbjct 99 QVAIALENAQLFEEVAKERTYSDSMLASMSNAVVTINEDGIIATCNK 145

> EBB41044.1 hypothetical protein GOS_238181 [marine metagenome]

Length=215

Score = 31.6 bits (70), Expect = 5.3, Method: Compositional matrix adjust.

Identities = 15/47 (32%), Positives = 23/47 (49%), Gaps = 0/47 (0%)

Query 54 ECLLRKCCCKLAIPKIGCCLDRLYWKTVKNIIIDKLCKKGIEVVVYYI 100

E + + +P + L R+ WKT ++IDKL GI + YI

Sbjct 136 EYGFKNLSDIHVPNLFKNLSRVMWKTYDELLIDKLIVNGIANQIYI 182

> EBB29372.1 hypothetical protein GOS_257812, partial [marine metagenome]

Length=166

Score = 31.2 bits (69), Expect = 7.6, Method: Compositional matrix adjust.

Identities = 20/77 (26%), Positives = 38/77 (49%), Gaps = 1/77 (1%)

Query 25 LIFYIVTKEKSYLKPTLANFSNAIDS-LYNECLLRKCCCKLAIPKIGCCLDRLYWKTVKNI 83

L+ Y+ ++ Y+ + S+ I S L+E + ++P + L R+ WKT +

Sbjct 57 LLSYFIYYKRQYIDINIFKRSSRIYSILSSEYGFKNLSDIHVPNLFKNLSRVMWKTYDEL 116

Query 84 IIDKLCKKGIEVVVYYI 100

IDKL GI ++++

Sbjct 117 FIDKLIVNGIANQIHHV 133

> KKM82403.1 hypothetical protein LCGC14_1319930 [marine sediment metagenome]

Length=345

Score = 31.2 bits (69), Expect = 8.0, Method: Composition-based stats.

Identities = 28/105 (27%), Positives = 48/105 (46%), Gaps = 10/105 (10%)

Query 4 YK-FCYNKKKEVGQVAVLQKERLI----FYIV---TKEKSYLKPTLANFSNAIDSLYNE 54

YK C ++ GQ+ V + L Y+V TK K ++ + +D+L +

Sbjct 47 YKSLCDGNELQPGQMVFDTKSLFEAEGPRYLVNFPTKAHWRSKSKISYVEDGLDALVST 106

Query 55 CLLRKCCCKLAIPKIGCCLDRLYWKTVKNIIIDKLCK-KGIEVVVY 98

+ IP +GC L W VK +I+ KL +G+++VV+

Sbjct 107 IREYGIKSIGIPPLGCGNGGLDWAQVKPLIVSKLSGLEGVDIVVF 151

> EBE52939.1 hypothetical protein GOS_9748365, partial [marine

metagenome]

Length=492

Score = 31.2 bits (69), Expect = 8.0, Method: Composition-based stats.

Identities = 19/67 (28%), Positives = 33/67 (49%), Gaps = 0/67 (0%)

Query 16 QVAVLQKERLIFYIVTKEKSYLKPTLANFSNAIDSLYNECLLRKCCCKLAIPKIGCCLDR 75

QVA+ + +F V KE++Y LA+ SNA+ ++ E + C K + +

Sbjct 55 QVAIALENAQLFEEVAKERTYSDSMLASMSNAVVTINEEGKIATCNKAGLKIFRVTQE 114

Query 76 YWKTVKN 82

KTV++

Sbjct 115 VGKTVED 121

> CBI09278.1 conserved hypothetical protein [mine drainage metagenome]

Length=368

Score = 31.2 bits (69), Expect = 8.5, Method: Composition-based stats.

Identities = 16/36 (44%), Positives = 19/36 (53%), Gaps = 0/36 (0%)

Query 63 LAIPKIGCCLDRLYWKTVKNIIIDKLCKKGIEVVVY 98

LAIP +GC L W V I+ KL GI V +Y

Sbjct 108 LAIPPLGCGNGGLEWTLVGPIMYQKLASLGISVDIY 143

> EBQ63088.1 hypothetical protein GOS_7721906, partial [marine metagenome]

Length=475

Score = 31.2 bits (69), Expect = 9.5, Method: Composition-based stats.

Identities = 19/67 (28%), Positives = 33/67 (49%), Gaps = 0/67 (0%)

Query 16 QVAVLQKERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCCCKLAIPKIGCCLDRL 75

QVA+ + +F V KE++Y LA+ SNA+ ++ E + C K + +

Sbjct 112 QVAIALENAQLFEEVAKERTYNDMSMLASMSNAVVTINEEGKIATCNKAGLKIFRVSTPEI 171

Query 76 YWKTVKN 82

KTV++

Sbjct 172 VGKTVED 178

> EDI73951.1 hypothetical protein GOS_382468, partial [marine metagenome]

Length=601

Score = 31.2 bits (69), Expect = 9.5, Method: Composition-based stats.

Identities = 19/67 (28%), Positives = 33/67 (49%), Gaps = 0/67 (0%)

Query 16 QVAVLQKERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCCCKLAIPKIGCCLDRL 75

QVA+ + +F V KE++Y LA+ SNA+ ++ E + C K + +

Sbjct 349 QVAIALENAQLFEEVAKERTYSDSMLASMSNAVVTINEERKIATCNKAGLKIFRVSTPEI 408

Query 76 YWKTVKN 82

KTV++

Sbjct 409 VGKTVED 415

> EDG96131.1 hypothetical protein GOS_690898, partial [marine metagenome]

Length=696

Score = 31.2 bits (69), Expect = 9.6, Method: Composition-based stats.

Identities = 17/67 (25%), Positives = 34/67 (51%), Gaps = 0/67 (0%)

Query 16 QVAVLQKERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCCKLAIPKIGCCLDRL 75

QVA+ + +F V+E++Y LA+ SNA+ ++ + ++ C K + +

Sbjct 376 QVAIALENAQLFEEVARERTYSDSMLASMSNAVVTINEDGIIATCNKAGLKIFRVSAQEI 435

Query 76 YWKTVKN 82

KTV++

Sbjct 436 VGKTVED 442

> EBV21546.1 hypothetical protein GOS_6939265, partial [marine

metagenome]

Length=282

Score = 30.8 bits (68), Expect = 9.7, Method: Compositional matrix adjust.

Identities = 17/60 (28%), Positives = 31/60 (52%), Gaps = 4/60 (7%)

Query 3 VYKFCYNKKKEVGQAVLQKERLIFYIVTKEKSYLKPTLANFSNAIDSPLYNECLLRKCC 62

+YK+ +N K E+G+ A+ T +KS+LK + + N D ++E L+R +

Sbjct 57 MYKYIFNPTELGEKAITSWSEY----YTTDKSFLKDSQKVYKNMKDFPFDESLIRNMLE 112

Lambda K H a alpha

0.327 0.143 0.446 0.792 4.96

Gapped

Lambda K H a alpha sigma

0.267 0.0410 0.140 1.90 42.6 43.6

Effective search space used: 33829893504

Database: ../../database/env_nr

Posted date: Mar 4, 2017 6:26 PM

Number of letters in database: 1,385,475,744

Number of sequences in database: 6,964,945

Matrix: BLOSUM62

Gap Penalties: Existence: 11, Extension: 1

Neighboring words threshold: 11

Window for multiple hits: 40

NOTE: if during execution you get the message

WARNING: Not enough GPU global memory to process volume No. 00 of the database.

Continuing without the GPU...

Consider splitting the input database in volumes with smaller size by using the option "-max_file_size <String>" (e.g. -max_file_sz 500MB).

Consider using fewer GPU blocks and then fewer GPU thread when formatting the database with (e.g. "-gpu_blocks 256" and/or "-gpu_threads 32") to reduce the GPU global memory requirements.

then your GPU does not have enough global memory to carry out the sequence alignment of the current database volume under the current GPU database configuration. To overcome this problem, you can reformat the database with "makeblastdb" with a value for "-max_file_sz" smaller than your previous choice. Another option is to recreate the GPU database by using the options "-method 2" and smaller values for "-gpu_blocks" and/or "-gpu_threads" than your previous choices (or use smaller values than 512 and 64 if you let GPU-BLAST create the GPU database with the default values as shown in the example above).

Timing the execution time

```
[ploskas@anaximander bin]$ time ./blastp -query ../../../../queries/SequenceLength_00000100.txt -  
db ../../../../database/sorted_env_nr -gpu t > gpu_output.txt  
  
real 0m5.564s  
user 0m4.237s  
sys 0m1.331s  
  
[ploskas@anaximander bin]$ time ./blastp -query ../../../../queries/SequenceLength_00000100.txt -  
db ../../../../database/sorted_env_nr -gpu f > cpu_output.txt  
  
real 0m9.970s  
user 0m9.778s  
sys 0m0.200s  
  
[ploskas@anaximander bin]$ diff cpu_output.txt gpu_output.txt
```