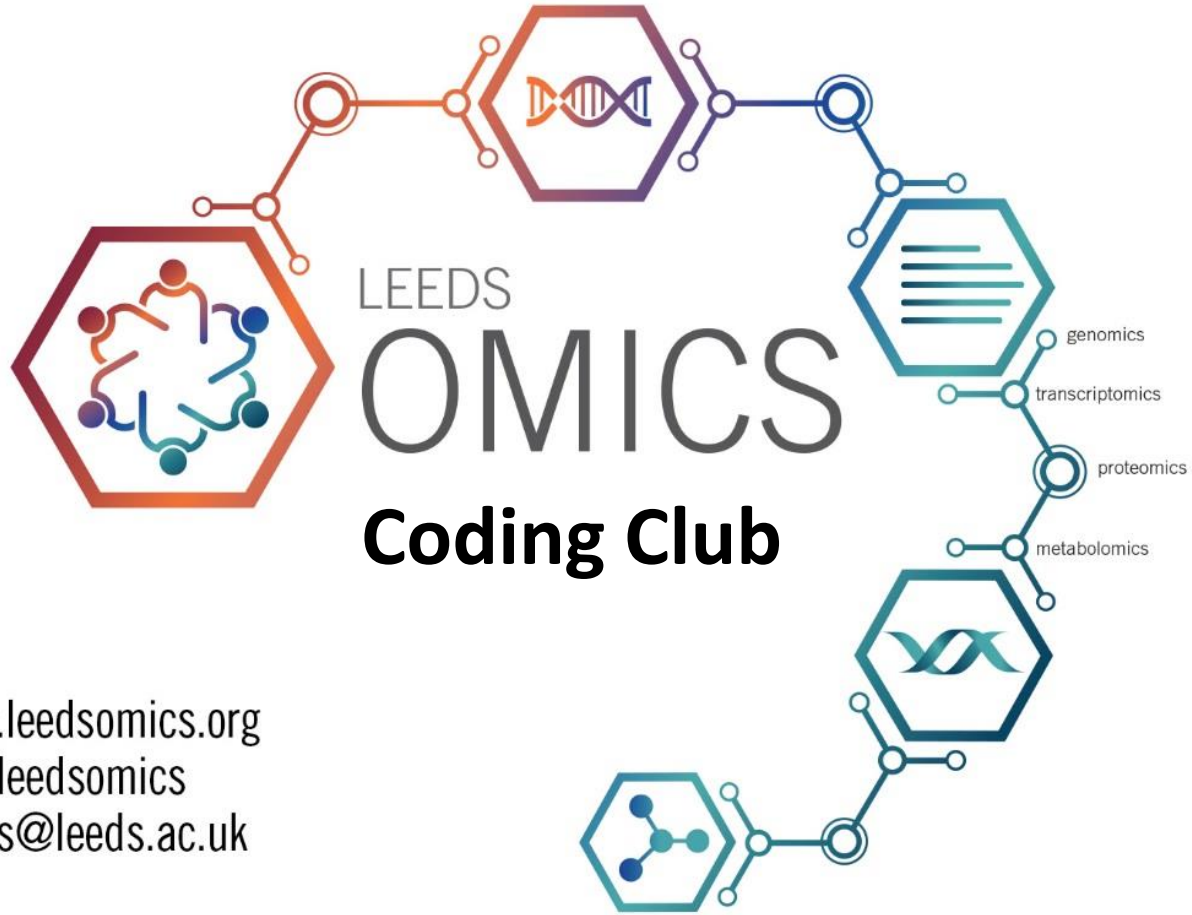


Coding Club

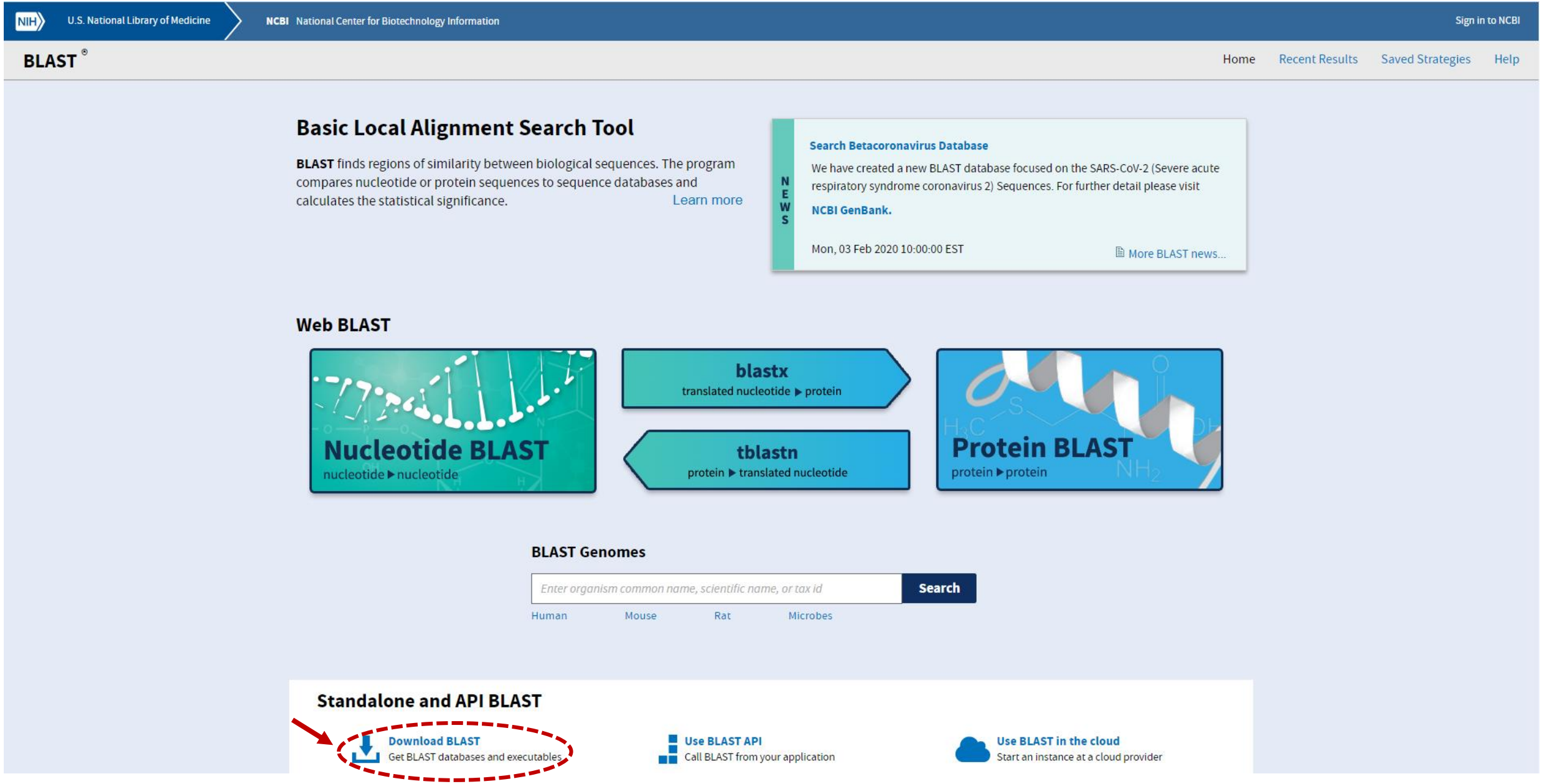
www.leedsomics.org
@leedsomics
omics@leeds.ac.uk

Using BLAST for sequence similarity search

Club Moderators: Elton Vasconcelos, Peter Mulhair, Euan McDonell, Chew Cheng, and Dapeng Wang

Topics to be addressed - **Survey Result**

https://blast.ncbi.nlm.nih.gov/Blast.cgi



Downloading the standalone package of executables

U.S. National Library of Medicine | NCBI National Center for Biotechnology Information | Sign in to NCBI

BLAST ®

Home | Recent Results | Saved Strategies | Help

Download BLAST Software and Databases

BLAST+ executables

Do you have difficulties running high volume BLAST searches? Do you have proprietary sequence data to search and cannot use the NCBI BLAST web site? Do you have access to your own server? Do you have your own research pipeline? Have security or IP concerns about sending searches outside of your organization? If you answered yes to any of these questions, read on!

The NCBI provides a suite of command-line tools to run BLAST called BLAST+. This allows users to perform BLAST searches on their own server without size, volume and database restrictions. BLAST+ can be used with a command line so it can be integrated directly into your workflow.

What are the next steps?

Download and install BLAST+. Installers and source code are available from ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/. Download the databases you need,(see database section below), or create your own. Start searching.

For more details, please see the BLAST+ user manual, the BLAST Help manual, the BLAST releases notes, and the article in BMC Bioinformatics (PubMed link). See our versioning policy.

The BLAST+ suite is the currently supported package. The older C toolkit executables are no longer supported. See our versioning policy.

We are always listening and welcome your feedback at BLAST Support Center.

Index of /blast/executables/blast+/LATEST/

[parent directory]

Name	Size	Date Modified
ChangeLog	85 B	04/12/2019, 02:52:00
ncbi-blast-2.10.0+-4.src.rpm	19.4 MB	04/12/2019, 02:50:00
ncbi-blast-2.10.0+-4.src.rpm.md5	63 B	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-4.x86_64.rpm	175 MB	04/12/2019, 02:50:00
ncbi-blast-2.10.0+-4.x86_64.rpm.md5	66 B	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-src.tar.gz	24.4 MB	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-src.tar.gz.md5	64 B	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-src.zip	28.5 MB	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-src.zip.md5	61 B	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-win64.exe	86.6 MB	04/12/2019, 02:49:00
ncbi-blast-2.10.0+-win64.exe.md5	63 B	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-x64-linux.tar.gz	222 MB	04/12/2019, 02:52:00
ncbi-blast-2.10.0+-x64-linux.tar.gz.md5	70 B	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-x64-macosx.tar.gz	141 MB	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-x64-macosx.tar.gz.md5	71 B	04/12/2019, 02:53:00
ncbi-blast-2.10.0+-x64-win64.tar.gz	86.3 MB	04/12/2019, 02:50:00
ncbi-blast-2.10.0+-x64-win64.tar.gz.md5	70 B	04/12/2019, 02:53:00
ncbi-blast-2.10.0+.dmg	143 MB	04/12/2019, 02:52:00
ncbi-blast-2.10.0+.dmg.md5	57 B	04/12/2019, 02:53:00

Running BLAST executables

 # Formatting DBs

```
$ makeblastdb -in targetDB.fasta -dbtype [nucl|prot]
```

 # Running a regular blastn analysis

```
$ blastn -task blastn -query file1.fasta -db targetDB -out file1--vs--targetDB.BlastN.txt
```

 # Changing parameters for a more sensitive search and reporting only the top10 hits

```
$ blastn -task blastn -query file1.fasta -db targetDB -out file1--vs--targetDB.BlastN-F-W7-1e-3-top10.txt -dust no -word_size 7 -evalue 1e-3 -num_descriptions 10 -num_alignments 10 -num_threads 12
```

 # Same as above, but generating a tabular output file rather than the long default output format

```
$ blastn -task blastn -query file1.fasta -db targetDB -out file1--vs--targetDB.BlastN-F-W7-1e-3-top10.tsv -dust no -word_size 7 -evalue 1e-3 -num_descriptions 10 -num_alignments 10 -outfmt [6|7] -num_threads 12
```

 # Running remotely (using NCBI nr or nt databases), just add the "-remote" option at the end of the cmd line

```
$ nohup blastp -query file1.fasta -db nr -out test10-vs-nr.BlastP_remote-top5.txt -num_descriptions 5 -num_alignments 5 -remote &
```

 # NOTE: nt.nal or nr.pal file must exist in the current directory, for either non-redundant nucleotide or protein DBs, respectively

 # They can be obtained by "untaring" one of the n[tr].##.tar.gz files from the ftp://ftp.ncbi.nlm.nih.gov/blast/db/

Tweaking tabular output format options

```
*** Formatting options
-outfmt <String>
  alignment view options:
    0 = Pairwise,
    1 = Query-anchored showing identities,
    2 = Query-anchored no identities,
    3 = Flat query-anchored showing identities,
    4 = Flat query-anchored no identities,
    5 = BLAST XML,
    6 = Tabular,
    7 = Tabular with comment lines,
    8 = Seqalign (Text ASN.1),
    9 = Seqalign (Binary ASN.1),
   10 = Comma-separated values,
   11 = BLAST archive (ASN.1),
   12 = Seqalign (JSON),
   13 = Multiple-file BLAST JSON,
   14 = Multiple-file BLAST XML2,
   15 = Single-file BLAST JSON,
   16 = Single-file BLAST XML2,
   18 = Organism Report
```

```
Options 6, 7 and 10 can be additionally configured to produce
a custom format specified by space delimited format specifiers.
The supported format specifiers are:
          qseqid means Query Seq-id
             qgi means Query GI
            qacc means Query accesion
         qaccver means Query accesion.version
            qlen means Query sequence length
          sseqid means Subject Seq-id
       sallseqid means All subject Seq-id(s), separated by a ';'
            sgi means Subject GI
          sallgi means All subject GIs
            sacc means Subject accession
         saccver means Subject accession.version
         sallacc means All subject accessions
            slen means Subject sequence length
          qstart means Start of alignment in query
            qend means End of alignment in query
          sstart means Start of alignment in subject
            send means End of alignment in subject
            qseq means Aligned part of query sequence
            sseq means Aligned part of subject sequence
          evalue means Expect value
        bitscore means Bit score
           score means Raw score
          length means Alignment length
          pident means Percentage of identical matches
          nident means Number of identical matches
        mismatch means Number of mismatches
        positive means Number of positive-scoring matches
         gapopen means Number of gap openings
            gaps means Total number of gaps
            ppos means Percentage of positive-scoring matches
          frames means Query and subject frames separated by a '/'
          qframe means Query frame
          sframe means Subject frame
            btop means Blast traceback operations (BTOP)
          staxid means Subject Taxonomy ID
        ssciname means Subject Scientific Name
        scomname means Subject Common Name
      sblastname means Subject Blast Name
       sskingdom means Subject Super Kingdom
         staxids means unique Subject Taxonomy ID(s), separated by a ';'
                  (in numerical order)
       sscinames means unique Subject Scientific Name(s), separated by a ';'
       scomnames means unique Subject Common Name(s), separated by a ';'
      sblastnames means unique Subject Blast Name(s), separated by a ';'
                  (in alphabetical order)
      sskingdoms means unique Subject Super Kingdom(s), separated by a ';'
                  (in alphabetical order)
          stitle means Subject Title
      salltitles means All Subject Title(s), separated by a '<>'
         sstrand means Subject Strand
           qcovs means Query Coverage Per Subject
         qcovhsp means Query Coverage Per HSP
          qcovus means Query Coverage Per Unique Subject (blastn only)
When not provided, the default value is:
'qaccver saccver pident length mismatch gapopen qstart qend sstart send
evalue bitscore', which is equivalent to the keyword 'std'
Default = `0'
```

NOTE: *One may see those instructions by typing any of the blast executable commands (blastn, blastp, blastx, tblastn, or tblastx) followed by "-help"*

Important filtering options for better results' interpretation

#Percent identity

-perc_identity <Real, 0..100>

#Percent query coverage per hsp

-qcov_hsp_perc

<Real, 0..100>

#***NOTE:*** *Another straightforward alternative is being quite loose (less stringent) on a first BLAST run, and then filtering a large tabular output file with the "awk" bash command, setting desired thresholds for both columns 3 and 4 (%identity and alignment_length, respectively). It is worthwhile (and faster) when one needs to test different thresholds.*

Bring your issues on!