BLAST Search WGS, TSA and SRA Data Locally

A step-by-step guide of using NCBI blast+ for local search against WGS, TSA & SRA data

https://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/

National Center for Biotechnology Information • National Library of Medicine • National Institutes of Health • Department of Health and Human Services

Introduction

Starting with version 2.13.0, the blast+ release will include local vdb blast-related programs. These programs allow search against sequence data stored in the Virtual Database (vdb) format, and interact with those data files for sequence retrieval. The sequence data are from NCBI's Sequence Read Archive (SRA) database, plus NCBI's Nucleotide database's Whole Genome Shotgun (WGS) and Transcriptome Sequence Assembly (TSA) projects. This handout focuses on the practical usage of those vdb-specific programs under a Linux environment.

Download and Installation

The latest release of the blast+ packages for Linux, Mac, and PC platforms are available from the NCBI FTP site. The download, installation, and configuration of the blast+ package are described in existing documents listed here: https://www.ncbi.nlm.nih.gov/books/NBK1762

The installation's bin subdirectory now includes a set of vdb-specific programs. Their name and functions are summarized in the table. These vdb-specific programs work out-of-box, interacting with datasets from NCBI remotely without caching the data locally. This default behavior can be configured through the installation and configuration of NCBI's sratoolkit. More on this at the end of this handout.

Vdb Programs	Function
blastn_vdb	blastn equivalent, use it to search nucleotide query against nucleotide sequence data (SRA, WGS, and TSA) stored in vdb format
tblastn_vdb	tblastn equivalent, use it to search protein query against dynamically translated nucleotide sequence data (SRA, WGS, and TSA) stored in vdb format
blast_vdb_cmd	blastdbcmd equivalent, use it to retrieve sequence from nucleotide sequence data (SRA, WGS, and TSA) store in vdb format
blast_formatter_vdb	blast_formatter equivalent, use it to reformat blast search result saved in archive format, or for formatting web SRA, WGS, or TSA search results using their unexpired RID

Practical Usage of Different vdb Programs

Use Case 1: Fishing for gyrb genes from WGS assemblies using blastn_vdb

1. Retrieve the E. coli K-12 strain's gyrb gene sequence for use as query

```
$ curl 'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nucleotide&rettype=fasta&id=NC_000913.3&seq_start=
3877705&seq_stop=3880119&strand=2' > k12_gyrb.fna
```

The command invokes curl and saves the output to a file (> k12_gyrb.fna). The request in quotes specifies the function to use (efetch.fcgi?), the database to target (db=nucleotide), the record format to get (rettype=fasta), and the record to retrieve (id=NC_000913.3). It also specifies the subsequences of the chromosomal region's start (from=3877705) and stop (to=388011) coordinates, in reverse orientation (strand=2).

2. Create a database list

A random set of WGS projects is selected from a nucleotide database search (https://go.usa.gov/xuaV2), then displayed in accession list format. For the BLAST use, we need to convert them to BLAST WGS database naming convention. See next step and the table below.

3. Search against the selected WGS project using blastn_vdb

```
$ blastn_vdb -query k12_gyrb.fna -db 'JAETIL01 JAETIP01 JAETLP01 JAETLE01 JAETNF01
JAKFZI01' -outfmt 7
```

The above command invokes blastn_vdb program, uses the gyrb sequence file generated above as the query (-query k12_gyrb.fna), searches against the set of WGS databases (-db 'JAETIL01 JAETIP01 JAETLP01 JAETLE01 JAETNF01 JAKFZI01', in quotes with space as separator), and asks for tabular output with column header (-outfmt 7). The result header and one of the six hits are shown below.

Accession.Version	BLAST WGS database
JAETIL000000000.1	JAETIL01
JAETIP000000000.1	JAETIP01
JAETLP000000000.1	JAETLP01
JAETLE000000000.1	JAETLE01
JAETNF000000000.1	JAETNF01
JAKFZI000000000.1	JAKFZI01

```
# BLASTN 2.13.0+
# Query: NC_000913.3:c3880119-3877705 Escherichia coli str. K-12 substr. MG1655, complete genome
# Database: JAETIL01 JAETIP01 JAETLE01 JAETLP01 JAETNF01 JAKFZI01
# Fields: query acc.ver, subject acc.ver, % identity, alignment length, mismatches, gap opens, q. start, q. end, s.
start, s. end, evalue, bit score
# 6 hits found
NC_000913.3:c3880119-3877705    JAETIP010000115.1       100.000 2415    0       0       1 2415      83299    80885    0.0
4460
```

Use Case 1 (cont.)

The command works by fetching the specified databases remotely from NCBI, performs the search on the local machine, and returns the result in requested format to specified location (console, pipe, or output file). When invoked from a cloud instance, the database is fetched from the cloud source (GCP or AWS).

Use Case 2: Finding candidate biodegradation enzymes in environmental WGS assemblies using tblastn_vdb

The degradation of complex organic compounds is a key process in bioremediation. Versatile Pseudomonas species contains many enzymes that could be useful in this process. In this case, we use the more sensitive tbalstn_vdb to search for candidate genes from bioreactor sludge WGS projects, using the protein sequence of a hcdA gene from Pseudomonas nandelii.

1. Get the protein sequence for use as the input query

```
$ curl 'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=protein&rettype=fasta&id=QRH16652.1' >hcdA.faa
```

This command invokes curl to submit a URL request in quotes to retrieve and save the requested protein sequence to a file (> hcdA.faa). The URL in the quotes specifies the function to use (efetch.fcgi), the database to target (db=protein), the format of the result (rettype=fasta), and the record to retrieve (&id=QRH16652.1).

2. Identify a desired set of WGS projects for use as database and construct the database list

The following web search identifies a set of WGS projects for use as the target databases: https://go.usa.gov/xu8nH. We can use programs provided by EDirect (https://www.ncbi.nlm.nih.gov/books/nbk179288, to be installed separately since it is not part of blast+). The corresponding commands for EDirect are:

```
$ esearch -db nucleotide -query 'wgs_master[prop] AND metagenomes[orgn] AND bioreactor AND sludge' | \
  esummary | xtract -pattern DocumentSummary -element AccessionVersion | \
  perl -ne 'chomp; s/0+\./0/; print "$_ ";'
```

It invokes esearch to search the nucleotide database (-db nucleotide) with the query in quotes (-query 'terms'), passes the output to retrieve the summary (| esummary), then processes the output with xtract (| xtract) to retrieve the accession.version from each record (-element AccessionVersion). The last Perl onliner converts each accession.version to a space-delimited database list in a single line, readable directly by vdb blast programs:

```
JAEAMF01 JAEAME01 JAEAMG01 JADHEF01 JADHEE01 JADHED01 CAAAGI01 CAAADM01 CAAAGG01 CAAAGD01 CAAAAV01 UOWD01 UOPZ01 UOPQ01
UOPT01 UOPS01 UOQD01 UOPR01 UOWF01 UOPO01 UOPP01 UOPN01 UOPL01 UOWI01 UOWC01 UOPM01 UOPK01 UOWB01 UOVZ01 UOWM01 UOWA01
UOWH01 UOVT01 UOVX01 UOVO01 UOVM01 UOVH01 UOVW01 UOVL01 UOVR01 UOVG01 UOVQ01 UOVP01 UOVJ01 UOVU01 UOVN01 UOVB01 UOVK01
UOUX01 UOVD01 UOVC01 UOVI01 UOVA01 UOVS01 UOVF01 UOVE01 AMZC01 MCND01 JTFP01 JTFO01 JTFN01 AATO01 AATN01
```

We can wrap the list into a database alias file to simplify the command line. Simply save the follow block of text into a file named **bioreactor_sludge.nvl**. The .nvl extension is important; it tells vdb blast tools that it is a nucleotide database (n) in vdb format (v), provided as an alias (l).

```
# manually created vdb alias for bioreactor & sludge wgs project list
TITLE Bioreactor and Sludge WGS projects
VDBLIST JAEAMF001 JAEAME001 JAEAMG001 JADHEF001 JADHEE001 JADHED001 CAAAGI001 CAAADM001 CAAAGG001 CAAAGD001 CAAAAV001
UOWD01 UOPZ01 UOPQ01 UOPT01 UOPS01 UOQD01 UOPR01 UOWF01 UOPO01 UOPP01 UOPN01 UOPL01 UOWI01 UOWC01 UOPM01 UOPK01 UOWB01
UOVZ01 UOWM01 UOWA01 UOWH01 UOVT01 UOVX01 UOVO01 UOVM01 UOVH01 UOVW01 UOVL01 UOVR01 UOVG01 UOVQ01 UOVP01 UOVJ01 UOVU01
UOVN01 UOVB01 UOVK01 UOUX01 UOVD01 UOVC01 UOVI01 UOVA01 UOVS01 UOVF01 UOVE01 AMZC01 MCND01 JTFP01 JTFO01 JTFN01 AATO01
AATN01
```

3. Search against this collection of WGS projects using tblastn_vdb

```
$ tblastn_vdb -query hcdA.faa -db bioreactor_sludge -outfmt 6 -out hcdA_vs_sludge.txt
```

This command invokes tblastn_vdb, specifies the input query (-query hcdA.faa), calls the database through the alias file without the .nvl extension (-db bioreactor_sludge), asks for the tabular output without header (-outfmt 6), and saves the results to a file (-out hcdA_vs_sludge.txt).

```
$ head -8 hcdA_vs_sludge.txt
QRH16652.1      UOWD01037391.1  44.204  509     244     6       3       505     1012    2436       7.40e-130       409
QRH16652.1      UOWD01078049.1  46.667  435     209     4       3       428     234     1496       2.18e-127       389
QRH16652.1      UOVZ01068180.1  49.013  304     140     2       22      320     885     4 1.78e-97 305
QRH16652.1      JADHED010018450.1       42.574  404     228     3       1       403     1606       2808    1.58e-93
312
QRH16652.1      UOPL01012726.1  37.549  514     284     9       1       501     2099    631.61e-91 314
QRH16652.1      UOWD01002877.1  38.812  505     262     12      2       501     22650   24038      6.63e-91        313
QRH16652.1      UOVZ01000486.1  38.812  505     262     12      2       501     30524   31912      6.77e-91        313
QRH16652.1      UOWF01001583.1  39.087  504     265     14      1       501     10359   11753      2.02e-86        300
```

Use Case 3: Evaluating the expression of hcdA in environmental RNA-seq datasets from the SRA database

1. Identify sample datasets from SRA and collect the runs for use as database

The NCBI SRA databases contains a huge volume of primary sequences from various sources. This search identifies a subset of RNA-seq datasets from environmental source: https://go.usa.gov/xu9nE

We use EDirect provided tools to collect the SRR accessions:

```
$ esearch -db sra -query 'metagenomes[orgn] AND biomol_rna[prop] AND polyethylene degrade bacteria Flora' | \
  esummary | xtract -pattern DocumentSummary -element Run@acc | \
  perl -ne 'chomp; print "$_ ";' > environmental_rna_runs.nvl
```

The command invokes esearch to search against sra (-db sra) with the specified terms (-term ‘metagenomes[orgn] AND biomol_rna[prop] AND polyethylene degrade bacteria Flora’). It passes the output to retrieve the summary (| esummary), and process the output with xtract XML parser (| xtract) to read each record (-pattern DocumentSummary) and extract the run accession (-element Run@acc). The added Perl oneliner, processes the output further into a space delimited list, and redirects the output to a file (> environmental_rna_runs.nvl) for use as a database alias after edit. The final alias file content is given below, with optional NSEQ and LENGTH fields derived by summing Run@total_spots (times 2 for paired layout) and Run@total_bases, respectively:

```
#Sample environmental run list, manually created on April 23, 2022
TITLE Environmental rna-seq run, with six accessions
VDBLIST SRR13045237 SRR13045238 SRR13045239 SRR13045240 SRR13045241 SRR13045242
NSEQ 44901478
LENGTH 6735221699
```

2. Search the hcdA protein sequences against this collection of SRR runs

```
$ tblastn_vdb -query hcdA.faa -db environmental_rna_runs -seg no -comp_based_stats 0 -outfmt 11 -out hcdA_sra_matches.asn
```

This command invokes tblastn_vdb to use the sequence file from Use Case 2 as query (-query hcdA.faa), searches against the set of SRA runs (-db environmental_rna_runs), disables seg filter (-seg no) and composition-based statistics (-comp_based_stats 0), requests result in a versatile archive format (-outfmt 11), and saves it to a file (-out hcdA_sra_matches.asn, a text file with content in asn.1 format). Using blast_formatter_vdb, we can convert this archive into other human readable format without repeating the search, a significant saving in computational resources.

3. Format the saved archive into other format using blast_formatter_vdb

```
$ blast_formatter_vdb -archive hcdA_sra_matches.asn -outfmt 6 -max_target_seqs 5
```

The above command invokes blast_formatter_vdb, reads in the saved archive file (-archive hcdA_sra_matches.asn), requests the result in tabular format (-outfmt 6), and asks for the top 5 hits (-max_target_seqs 5). This prints the result to the console:

```
QRH16652.1      SRA:SRR13045242.870978.1        48.000  50      26      0       283     332     150     1       1.61e-05
48.1
QRH16652.1      SRA:SRR13045242.2856694.2       46.000  50      27      0       284     333     1       150     2.20e-05
47.8
QRH16652.1      SRA:SRR13045239.2879291.1       48.000  50      26      0       284     333     150     1       2.20e-05
47.8
QRH16652.1      SRA:SRR13045242.3419984.2       46.000  50      27      0       284     333     1       150     4.11e-05
47.0
QRH16652.1      SRA:SRR13045242.2916717.2       46.000  50      27      0       284     333     1       150     4.11e-05
47.0
```

4. Retrieve fasta sequences for matches using their seqids

```
$ blast_formatter_vdb -archive hcdA_sra_matches.asn -outfmt 6 |cut -f 2 > hcdA_matched_seqids.txt
```

The above command invokes blast_formatter_vdb, reads in a saved archive file (-archive hcdA_sra_matches.asn), and generates the tabular result without header. It passes the output to cut (| cut) and cuts out the sequence ids in the second column (-f 2), then saves the output to a file (hcdA_matched_seqids.txt) for use for sequence retrieval.

```
$ blast_vdb_cmd -db environmental_rna_runs -entry_batch hcdA_matched_seqids.txt -out matched_reads.fna
```

This command calls blast_vdb_cmd, specifies the database (-db environmental_rna_runs), reads in a set of seqids (-entry_batch hcdA_matched_seqids.txt), gets the default fasta output (without -outfmt specification), and saves the output to the specified file (-out matched_reads.fna). The output file can be used in further downstream processing.

Integration of blast+ and sratoolkit

Installation of blast+ (2.13.0 or later version required) offers a simple way to access the large volumes of WGS, TSA, and SRA datasets deposited to NCBI's Nucleotide and SRA database. Combining the blast+ installation with an installation of the NCBI sratoolkit offers additional advantage. The table below sums up the basic characteristics of these two setups. The default behavior of vdb programs from blast+ can be modified by the configuration of sratoolkit, installed on top of blast+. This will allow the download of WGS, TSA, or SRA runs, and search against those datasets saved locally.

Setup	Characteristics
Blast+ alone	• Accesses target data through remote fetching • Does not cache target datasets locally • Uses network bandwidth every time a search or retrieval is run • Cannot download datasets in more compact vdb format • Contains no tools to extract FASTQ or SAM format from SRA datasets
With sratoolkit	• Allows local datasets caching through prefetch when the option is enabled • Does not require network bandwidth when search against local cache • Provides tools to extract SRA data in FASTQ and SAM format • Requires large storage and active management of downloaded datasets

Installation and configuration of sratoolkit

1. Download and install the sratoolkit

Note: Always check the SRA software download page (https://go.usa.gov/xuXkh) for link to the latest version.

```
$ wget -O sratoolkit.tar.gz 'ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/3.0.0/sratoolkit.3.0.0-
centos_linux64.tar.gz'
$ tar zxpf sratoolkit.tar.gz
```

The first command, wget, requests the package from the quoted URL and saves it to a local file (-O sratoolkit.tar.gz). The second command inflates and extracts the downloaded archive to install the package in the current working directory, under a subdirectory named sratoolkit.3.0.0-centos_linux64. Programs are under the /bin subdirectory under it.

2. Configure the installed sratoolkit

- Modify PATH environment variable

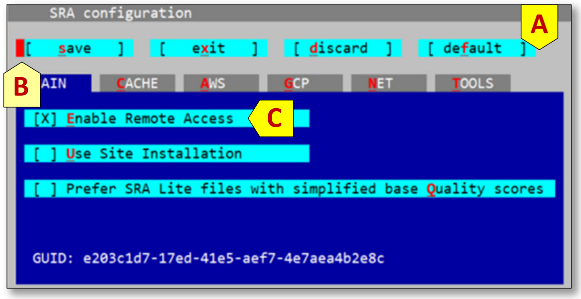
```
$ export PATH=$PATH:~/sratoolkit.3.0.0-centos_linux64/bin
```

This bash command exports a variable called PATH, by setting it to existing value ($PATH), then append (:) an additional path (~/sratoolkit.3.0.0-centos_linux64/bin, where ~ marks the home directory). This informs the system where to find sratoolkit's programs.

- Configure the sratoolkit

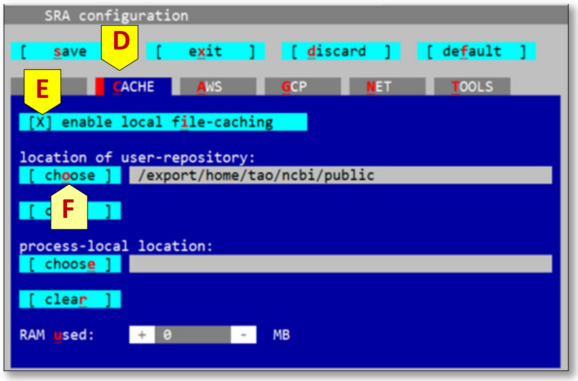
```
$ vdb-config -i
```

This launches an interactive window (**A**). There the active tab or field is marked by a red rectangle (**B**). Use tab key to cycle through different options, or use a red letter or mouse click to jump to a specific option.



```
[X] Enable Remote Access
[ ] Use Site Installation
[ ] Prefer SRA Lite files with simplified base Quality scores

GUID: e203c1d7-17ed-41e5-aef7-4e7aea4b2e8c
```

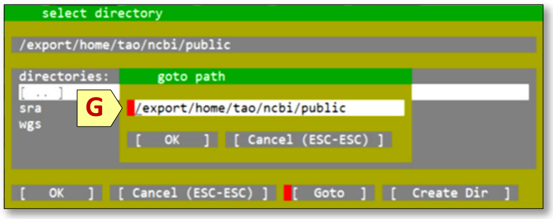


```
[X] enable local file-caching

location of user-repository:
[ choose ] /export/home/tao/ncbi/public

process-local location:
[ choose ]

RAM used: + 0 - MB
```



```
/export/home/tao/ncbi/public
goto path: /export/home/tao/ncbi/public
```

Specific setting changes:

1) Toggle on/off remote access. Type "**e**" to toggle off/on the "Enable Remote Access" checkbox. Default is on (**C**, keep it as is).
2) Enable local file-caching. Type "**c**" to go to the Cache menu (**D**), and toggle on "enable local file-caching" by typing "I" (i.e., enable the X, **E**).
3) Edit the Cache directory if needed. Type "**o**" to go to the "choose" line, hit enter to invoke the select directory prompt (**F**). Tab to "Goto" option and hit enter to edit the path in the the "goto path" prompt (**G**). Tab to "OK" and hit enter to return to previous prompt, confirm the change when prompted.
4) Save the changes. Type "**s**" to save, then "**x**" to exist the configuration.

Installation and configuration of sratoolkit (cont.)

3. Examine the existing configuration

```
$ vdb-config -a
```

This command invokes vdb-config and uses the -a option to generate the sratoolkit’s configuration XML to the console, which details the setup of this installation.

The relevant section pertaining to the local data cache directories is excerpted to the right (H), which indicates that the local cache files will be stored in a set of subdirectories under the specified ncbi directory.

```
<cache-disabled>false</cache-disabled>
<default-path>/home/tao/ncbi</default-path>
<main>
 <public>
  <apps>
   <file>
    <volumes>
     <flat>files</flat>
    </volumes>
   </file>
   <nakmer>
    <volumes>
     <nakmerFlat>nannot</nakmerFlat>
    </volumes>
   </nakmer>
   <nannot>
    <volumes>
     <nannotFlat>nannot</nannotFlat>
    </volumes>
   </nannot>
   <refseq>
    <volumes>
     <refseq>refseq</refseq>
    </volumes>
   </refseq>
   <sra>
    <volumes>
     <sraFlat>sra</sraFlat>
    </volumes>
   </sra>
   <wgs>
    <volumes>
     <wgsFlat>wgs</wgsFlat>
    </volumes>
   </wgs>
  </apps>
  <cache-enabled>true</cache-enabled>
   <root>/export/home/tao/ncbi/public</root>
```

Example Usage of Locally Cached vdb Sequence Datasets

Use Case 4: Download sra data sets for local cache and search

1. Download the SRR runs to cache locally

With sratoolkit installed, we can download WGS and TAS projects, or SRA runs to store commonly used datasets locally. This conserves network bandwidth and avoids issues caused by network interruption. For simplicity and comparison, we will reuse the list of runs in Use Case 3, and use EDirect tools to collect the SRR accessions:

```
$ esearch -db sra -query 'metagenomes[orgn] AND biomol_rna[prop] AND
polyethylene degrade bacteria Flora' | \
  esummary | xtract -pattern DocumentSummary -element Run@acc >
environmental_rna_runs.txt
```

The above command drops the Perl oneliner. It sends a list SRR accessions, one record per line to the output file, which can be used as input to prefetch for batch download.

```
$ prefetch --option-file environmental_rna_runs.txt -p
```

This command invokes prefetch, reads in an accession list generated above (--option-file environmental_rna_runs.txt), and prints the progress to the console (-p). Portion of the message concerning one of the runs is shown below.

```
2022-04-25T00:13:12 prefetch.3.0.0: Current preference is set to retrieve SRA Normalized Format files with full base
quality scores.
2022-04-25T00:13:12 prefetch.3.0.0: 1) Downloading 'SRR13045237'...
2022-04-25T00:13:12 prefetch.3.0.0: SRA Normalized Format file is being retrieved, if this is different from your
preference, it may be due to current file availability.
2022-04-25T00:13:12 prefetch.3.0.0:  Downloading via HTTPS...
|-------------------------------------------------- 100%
```

The above process saves the downloaded SRA datasets to the ncbi/public/sra subdirectory:

```
$ ls ncbi/public/sra/
SRR12800266.sra  SRR13786427.sra  SRR16965134.sra  SRR16965136.sra
SRR13479238.sra  SRR16965133.sra  SRR16965135.sra
```

2. Search the hcdA gene coding sequence against the downloaded SRR runs

```
$ efetch -db protein -format fasta_cds_na -id AZM97552.1 > unk_cds.fna
```

This command invokes efetch, sets protein as the database (-db protein), requests the coding cds (-format fasta_cds_na) of AZM97552.1 (-id AZM97552.1). It is used in the following blastn_vdb search.

```
$ blastn_vdb -query unk_cds.fna -db environmental_rna_runs -dust no -outfmt 6 -max_target_seqs 5 -out unk_cds_out.txt

lcl|CP034367.1_cds_AZM97552.1_1 SRA:SRR13045242.3455804.2       100.000 150     0    675      824      1       150
1.71e-71        278
lcl|CP034367.1_cds_AZM97552.1_1 SRA:SRR13045242.3444683.2       100.000 150     0    480      629      1       150
1.71e-71        278.
```

Use Case 4: Download sra data sets for local cache and search (cont.)

3. Retrieve the hits from the locally stored runs

Use the same approach described in Use Case 3 to retrieve hits of interest by parsing out the seqids from the second column.

```
$ cut -f2 unk_cds_out.txt  > blastn_matched_runs

$ blast_vdb_cmd -db environmental_rna_runs -entry_batch blastn_matched_runs
>gnl|SRA|SRR13045242.3455804.2 Length: 150
GTTTGATCCGCCCTTCCATCCCAATCAATCGGTGCTGCTGCACAAAGAACCCGACAATGTGTGGCGTATCGACTTCCAGC
TAGGCTGGGACGCTGACCCGGAAGAAGAAAAGAAAGAAGAGAACATTCGCCCGCGCGTGCAGGCGATGCT
… …
```

Testing with disabled remote access setting through vdb-config -i will confirm the fact that above search and retrieval are against the locally cached datasets.

Use Case 5: Download WGS Datasets for Local Cache and Search

1. Download WGS datasets used in Use Case 2

```
$ esearch -db nucleotide -query 'wgs_master[prop] AND metagenomes[orgn] AND bioreactor AND sludge' | \
  esummary | xtract -pattern DocumentSummary -element AccessionVersion | \
  perl -ne 'chomp; s/0+\./0/; print "$_\n";' > WGS_list.txt
```

The above command retrieves the accession of the WGS masters. The Perl oneliner converts the accessions into BLAST database naming convention and sends the result to an output file, one record per line, for use with prefetch download. The following is the prefetch command and part of its console progress report (enabled by -p switch):

```
$ prefetch --option-file WGS_list.txt -p

2022-04-25T03:57:11 prefetch.3.0.0: Current preference is set to retrieve SRA Normalized Format files with full base
quality scores.
2022-04-25T03:57:11 prefetch.3.0.0: 63) Downloading 'AATN01.4'...
2022-04-25T03:57:11 prefetch.3.0.0:  Downloading via fasp...
2022-04-25T03:57:15 prefetch.3.0.0:  FASP download succeed
2022-04-25T03:57:15 prefetch.3.0.0: 63) 'AATN01.4' was downloaded successfully
2022-04-25T03:57:15 prefetch.3.0.0: 'AATN01' has 0 unresolved dependencies
```

2. BLAST search against downloaded WGS projects

```
$ curl 'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?
db=nuccore&rettype=fasta&id=AP013070.1&seq_start=4709361&seq_stop=4710368&strand=2' >putida_rpoS.fna
```

The above command retrieves the coding sequence of the rpoS sigma factor from Pseudomonas putida for use as query in a blastn_vdb search:

```
$ blastn_vdb -query putida_rpoS.fna -db bioreactor_sludge -outfmt 6
```

The above command uses the same WGS database alias, but the search is targeting the locally cached datasets since we have it downloaded already. Partial output of the search is shown below.

```
AP013070.1:c4710368-4709361     JAEAMG010001275.1       90.972  1008    91      0     1008     7390    8397    0.0
1358
AP013070.1:c4710368-4709361     JAEAMF010001220.1       90.972  1008    91      0     1008     7390    8397    0.0
1358
```

Additional Information

The NCBI BLAST team provides a PERL script that can generate a custom WGS database alias for a specific organism. The alias file can then be used with local vdb-specific blast searches. See this README file for more details: https://ftp.ncbi.nlm.nih.gov/blast/WGS_TOOLS/README_BLASTWGS.txt

Some users may be on a system that has a pre-existing installation of the NCBI sratoolkit that they are not aware of. If that sratoolkit has the remote access disabled, vdb programs from the blast+ will return error when the specified datasets are not present in locally. An example error is given below. Enabling remote access through vdb-config -i will resolve the issue.

```
    T0 "/home/coremake/release_build/build/PrepareRelease_Linux64-
Centos_JSID_01_660219_130.14.18.128_9008__PrepareRelease_Linux64-Centos_1643834072/c++/compilers/unix/../../src/algo/
blast/vdb/vdb2blast_util.cpp", line 257: Error: (CException::eUnknown) ncbi::CVDBBlastUtil::x_MakeVDBSeqSrc() - Error
opening the following db(s): UOWM01
```