NCBI E-utilities

CBMG 688p

Mihai Pop

NCBI E-utils

- A way to access NCBI databases directly, through software
- i.e. No need to download databases

- Caveats:
  - access through web server (not ideal for large volumes of data)
  - utilities can be brittle (build in retries and checkpointing)
  - NCBI gets cross if you abuse the system (e.g. too many requests)
  - WARNING: the whole campus can be blacklisted because of a bug in your code!

Additional resources

- http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html

User Requirements
Do not overload NCBI's systems. Users intending to send numerous queries and/or retrieve large numbers of records from Entrez should comply with the following:

* Run retrieval scripts on weekends or between 9 pm and 5 am Eastern Time weekdays for any series of more than 100 requests.
* Send E-utilities requests to http://eutils.ncbi.nlm.nih.gov, not the standard NCBI Web address.
* Make no more than 3 requests every 1 second.
* Use the URL parameter email, and tool for distributed software, so that we can track your project and contact you if there is a problem.
* NCBI's Disclaimer and Copyright notice must be evident to users of your service. NLM does not claim the copyright on the abstracts in PubMed; however, journal publishers or authors may. NLM provides no legal advice concerning distribution of copyrighted materials, consult your legal counsel.

Blast client

- Netblast (blastcl3)
  - Same as the old version of NCBI blast (blastall) but runs against the NCBI database directly (no need to download data)
- Note: netblast is obsolete
- Current blast version (blast+) support "-remote" command-line option to run against NCBI database

NCBI programmatic access

- http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html
  - must write your own HTTP client (LWP Perl module helps)
  - queries go directly to web server
  - data returned in XML
- http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=show&f=doc&m=obtain&s=stips
  - stub script provided (query_tracedb)
  - queries still go through web server
  - data returned in a variety of user selected formats
- For both, limits are set on the amount of data retrieved, e.g. less than 40,000 records at a time
- Download procedure:
  - figure out # of records to be retrieved ("count" query)
  - read data in allowable chunks
  - combine the chunks

Example: query_tracedb

- ftp://ftp.ncbi.nih.gov/pub/TraceDB/misc/query_tracedb
- http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=show&f=doc&m=obtain&s=stips
- Note: it just connects to database through the web

```
#!/usr/bin/perl -w
use strict;
use LWP::UserAgent;
use HTTP::Request::Common 'POST';

$ENV{'LANG'}='C';
$ENV{'LC_ALL'}='C';

my $query = join ' ', @ARGV;
$query = 'help' if $query =~ /^(\-h|\-\-help|\-)$/;
$query = join('', <STDIN>) if ! $query;

my $req = POST 'http://trace.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=raw', [query=>$query];
my $res =  LWP::UserAgent->new->request($req, sub { print $_[0] });
die "Couldn't connect to TRACE server\n" if ! $res->is_success;
```

query_tracedb...cont

- Note: NCBI puts a limit of 40,000 records at a time
- First, figure out how many sequences you will retrieve

```
query_tracedb "query count species_code='AEDES AEGYPTI'"

122116
```

NCBI query string

- Second, get the identifiers in batches (note...commands wrap)

```
query_tracedb "query page_size 40000 page_number 0 binary
species_code='AEDES AEGYPTI'" > page1.bin
query_tracedb "query page_size 40000 page_number 1 binary
species_code='AEDES AEGYPTI'" > page2.bin
...
```

- Then, retrieve all the data

```
(echo -n "retrieve_tgz all 0b"; cat page1.bin) | query_tracedb >
data1.tgz
...
```

NCBI query string

Esearch and Efetch

- http://www.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?
- Simple URL, you just need to add the actual command
  - http://www.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=nr&term=YOURQUERY
- Example:
  - http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=nucleotide&term=biomol+trna[prop]
- Note: results returned in XML (not easy for you to read)
- Efetch can retrieve the data in a more useful format

Esearch and Efetch

- Esearch returns IDs and Efetch retrieves the actual data
- Just like query_tracedb, may need to limit the number of records retrieved

  ?rettype=count – retrieves the number of records

  ?retstart=1000&retmax=1000 -retrieves records from 1000 to 2000

- Linking to Efetch: use cookies

  ?usehistory=yes   - will return a cookie

```
<QueryKey>1</QueryKey>
–
<WebEnv>
NCID_1_48550085_130.14.22.28_9001_1290566364_1241742595
</WebEnv>
```

  – the cookie can be used to retrieve the corresponding results

Esearch and Efetch

http://www.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=pubmed

&**WebEnv=%3D%5DzU%5D%3FIJIj%3CC%5E%5DA%3CT%5DEACgdn%3DF%5E%3Eh**

GFA%5D%3CIFKGCbQkA%5E_hDFiFd%5C%3D

&**query_key=6**

- Efetch parameters (depend on database)

  rettype=xml – will return XML output (many other options exist)

  retmode=fasta – will return results in FASTA format

- Note: if you don't use the WebEnv, you will need to use the actual ids:

  ?id=id1,id2,...,idn

Through Bio::Perl

- Eutils can be a pain (you will need to write a fair amount of fancy code)
- Perl to the rescue

```
use Bio::DB::GenBank;
$gb = Bio::DB::GenBank->new();

$seq = $gb->get_Seq_by_id('MUSIGHBA1'); # Unique ID
# note: actual Bio::Perl sequence record
```

More Bio::Perl

```
my $query = Bio::DB::Query::GenBank->new
        (-query   =>'Oryza sativa[Organism] AND EST',
         -reldate => '30',
         -db      => 'nucleotide');
my $seqio = $gb->get_Stream_by_query($query);

 while( my $seq =  $seqio->next_seq ) {
     print "seq length is ", $seq->length,"\n";
}
```

- Note: Bio::Perl has utilities for other databases as well (EMBL, etc.)

Getting data from the Short Read Archive

- Short read archive has lots of data...takes a long time to download datasets
- A better approach: Aspera client (tweaks internet parameters to make transfer fast)
- http://www.ncbi.nlm.nih.gov/viewvc/v1/trunk/sra/doc/SRA/Aspera_Transfer_Guide.pdf?view=co
- Step 1: install Aspera software
- You can download through the browser
- Location of command-line client depends:

  Windows: C:\Program Files\Aspera\Aspera Connect\bin\ascp.exe

  Linux: ~/.aspera/connect/bin/ascp

  MacOS: /Applications/Aspera Connect.app/Contents/Resources/ascp

Aspera command

- ascp -i asperaweb_id_dsa.putty <path_to_file> .

- similar to "cp" command in UNIX
- Note: path to file looks like:

```
ascp -i $ETCPATH/asperaweb_id_dsa.putty anonftp@ftp-private.ncbi.nlm.nih.gov:/sra/static/SRX018/SRX018273/SRR042027_2.fastq.gz .
```

- Note the "asperaweb_id_dsa.putty" file – this is an encryption/authentication key provided with the aspera code
  - Windows: C:\Program Files\Aspera\Aspera Connect\etc\asperaweb_id_dsa.putty
  - Linux: ~/.aspera/connect/etc/asperaweb_id_dsa.putty
  - Mac: /Applications/Aspera Connect.app/Contents/Resources/asperaweb_id_dsa.putty

Homework

- Due Friday, December 3rd at Midnight
- Submit as assignment 12 on grace system

    - Write a simple script that wraps around the query_tracedb script from NCBI and allows a user to download an entire data-set without worrying about “page limits”.

      Note: set your page limit to 4000 records (rather than the 40,000 allowed) so as not to overload the NCBI servers.