Flat Databases HOWTO

Lincoln Stein, *Cold Spring Harbor Laboratory* [http://www.cshl.org] <lstein-at-cshl.org>

Brian Osborne, *Cognia Corporation* [http://www.cognia.com] <brian-at-cognia.com>

Heikki Lehväslaiho, *European Bioinformatics Institute* [http://www.ebi.ac.uk] <heikki-at-ebi.co.uk>

2003-02-26

Revision History

Revision 1.0	2003-02-26	LS
	First version	
Revision 1.1	2003-10-17	BIO
Revision 1.2	2003-10-17	HL
	from txt reformatted into SGML	
Revision 1.3	2004-05-20	BIO
	Add section on custom indexes	

The Open Biological Database Access (OBDA) standard specifies a way of generating indexes for entry-based sequence files (e.g. FASTA, EMBL) so that the entries can be looked up and retrieved quickly. These indexes are created and accessed using the `Bio::DB::Flat` module.

Table of Contents

1. Creating OBDA-Compliant Indexed Sequence Files

`Bio::DB::Flat` has the same functionality as the various `Bio::Index` modules. The main reason to use it is if you want to use the BioSequence Registry system (see the OBDA Access HOWTO at *http://bioperl.org/HOWTOs*), or if you want to share the same indexed files among scripts written in other languages, such as those written with BioJava or BioPython.

There are four steps to creating a `Bio::DB::Flat` database:

1. Select a Root Directory

   Select a directory in which the flat file indexes will be stored. This directory should be writable by you, and readable by everyone who will be running applications that access the sequence data.

2. Move the Flat Files Into a Good Location

The indexer records the path to the source files (e.g. FASTA, or local copies of GenBank, Embl or SwissProt). This means that you must not change the location or name of the source files after running the indexer. Pick a good stable location for the source files and move them there.

3. Choose a Symbolic Name for the Database

   Choose a good symbolic name for the database. For example, if you are mirroring GenBank, "genbank" might be a good choice. The indexer will create files in a subdirectory by this name located underneath the root directory.

4. Run the bioflat_index.pl Script

   The final step is to run the bioflat_index.PLS script. This script is located in the BioPerl distribution, under scripts/DB. For convenience, you are offered the option to copy it to /usr/bin or another system-wide directory on 'make install' (and its name will be changed to bioflat_index.pl).

2. Choosing Your Options

The typical usage is as follows:

```
bioflat_index.pl -c -l /usr/share/biodb -d genbank -i bdb -f fasta data/*.fa
```

The following command line options are required:

Option	Description
-c	create a new index
-l	path to the root directory
-d	symbolic name for the new database
-i	indexing scheme (discussed below)
-f	source file format

Table 1. bioflat_index command-line options

The `-c` option must be present to create the database. If the database already exists, `-c` will reinitialize the index, wiping out its current contents.

The `-l` option specifies the root directory for the database indexes.

The `-d` option chooses the symbolic name for the new database. If the `-c` option is specified, this will cause a new directory to be created underneath the root directory.

The `-i` option selects the indexing scheme. Currently there are two indexing schemes supported: "bdb" and "flat." "bdb" selects an index based on the Berkeley DB library. It is generally the faster of the two, but it requires that the Berkeley DB library (from Linux RPM or from *www.sleepycat.com* [http://www.sleepycat.com], version 2 or higher) and the Perl BerkeleyDB module be installed on your system. The Perl DB_File module will not work.

"flat" is a sorted text-based index that uses a binary search algorithm to rapidly search for entries. Although not as fast as bdb, the flat indexing system has good performance for even large databases, and it has no requirements beyond Perl itself. The disadvantage of "flat" is that the indexing of large databases will be slower and more memory-intensive than the indexing using "bdb".

Once an indexing scheme has been selected there is no way to change it other than recreating the index from scratch using the `-c` option.

The `-f` option specifies the format of the source database files. It must be one of the many formats that BioPerl supports, including "genbank", "swiss", "embl" or "fasta". Consult the `Bio::SeqIO` documentation for the complete list. All files placed in the index must share the same format.

The indexing script will print out a progress message every 1000 entries, and will report the number of entries successfully indexed at the end of its run.

To update an existing index run bioflat_index.pl without the `-c` option and list the files to be added or reindexed. The `-l` and `-d` options are required, but the indexing scheme and source file format do not have to be specified for updating as they will be read from the existing index.

For your convenience, bioflat_index.pl will also take default values from the following environment variables:

ENV variable	**description**
OBDA_FORMAT	format of sequence file (`-f`)
OBDA_LOCATION	path to directory in which index files are stored (`-l`)
OBDA_DBNAME	name of database (-d)
OBDA_INDEX	type of index to create (`-i`)

Table 2. Environmental variables

3. Moving Database Files

If you must change the location of the source sequence files after you create the index, there is a way to do so. Inside the root directory you will find a subdirectory named after the database, and inside that you will find a text file named "config.dat." An example config.dat is shown here:

```
index flat/1
fileid_0 /share/data/alnfile.fasta 294
fileid_1 /share/data/genomic-seq.fasta 171524
fileid_2 /share/data/hs_owlmonkey.fasta 416
fileid_3 /share/data/test.fasta 804
fileid_4 /share/data/testaln.fasta 4620
primary_namespace ACC
secondary_namespaces ID
format URN:LSID:open-bio.org:fasta
```

For each source file you have moved, find its corresponding "fileid" line and change the path. Be careful not to change anything else in the file or to inadvertently replace tab characters with spaces.

4. Secondary or custom namespaces

The bioflat_index script creates what could be called a default index using sequence files in a few different formats. Each of these formats has its own default primary key, the key that can be used as a a query. Consider this fasta entry:

```
>P84139 gi|443893
MHHLLHGRHHRRQRKKKITEWSVKKKIIACNMIIRRIIIIIAHYTRQWPLMNVDS
```

The Bio::DB::Flat modules will use the first "word" in the fasta header as the primary key, "P84139", equivalent to this regular expression:

```
>(\S+)
```

This value turns out to be same value returned by Sequence object's display_id() method. The table below shows the default primary keys of the 4 Bio::DB::Flat formats.

Format	Seq method	Regular expression
fasta	display_id()	>(\S+)
swiss	display_id()	^ID\s+(\S+)
genbank	primary_id()	^LOCUS\s+(\S+)
embl	display_id()	^ID\s+(\S+)

Table 3. Default primary keys

What if you wanted to use some other part of the entry as a key, like the GI number? This could also be called specifying another namespace, or a secondary namespace. No problem, but now you've gone beyond the capabilities of the bioflat_index script, you'll need to write your own. It would look something like this:

```
use strict;
use Bio::DB::Flat::BinarySearch;

# use single quotes so you don't have to write
# regular expressions like "gi\\|(\\d+)"
my $primary_pattern = '^>(\S+)';
# one or more patterns stored in a hash:
my $secondary_patterns = {GI => 'gi\|(\d+)'};

my $db = Bio::DB::Flat::BinarySearch->new(
                     -directory          => "/home/bio",
 -dbname             => "ppp",
        -write_flag          => 1,
 -primary_pattern    => $primary_pattern,
 -primary_namespace  => 'ACC',
 -secondary_patterns => $secondary_patterns,
 -verbose            => 1,
 -format             => 'fasta'  );

$db->build_index("ppp.fa");
```

This code will index the file "ppp.fa" and place the indexes in the directory "/home/bio/ppp". Then you can retrieve sequences like this:

```
$db->secondary_namespaces("GI");
my $acc_seq = $db->get_Seq_by_id("P84139");
my $gi_seq = $db->get_Seq_by_secondary("GI",443893);
```

5. More information

For more information on using your indexed flat files please see the *OBDA Access HOWTO* [http://bioperl.org/HOWTOs/html/OBDA_Access.html].