OBDA Access HOWTO

Aaron Mackey, *University of Virginia* [http://www.virginia.edu] <amackey@virginia.edu>

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

Peter Schattner, *UCSC* [http://www.ucsc.edu] <schattner@alum.mit.edu>

Heikki Lehvaslaiho, *EBI* [http://ebi.ac.uk] <heikki@ebi.ac.uk>

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

2002-07-14

Revision History

Revision 1.1	2003/02/20	lstein
added some OBDA documentation plus flat file index load script		
Revision 1.2	2003/02/20	lstein
check in reasonable howtos for database access and flat databases		
Revision 1.3	2003/02/20	lstein
added new heading to biodatabase access doc for Heikki to fill in		
Revision 1.4	2003/02/20	heikki
added explanation for biogetseq		
Revision 1.5	2003/03/01	peters
documentation updates		
Revision 1.6	2003/03/12	amackey
documentation updates		
Revision 1.7	2003/06/10	BIO
convert to XML/SGML, corrections		

This is a HOWTO written in DocBook (XML) that explains how to set up and use the Open Biological Database Access system.

Table of Contents

1. Introduction

Importing sequences with annotations is a central part of most bioinformatics tasks. Bioperl supports importing sequences from indexed flat-files, local relational databases and remote (internet) databases. Previously, separate programming syntax was required for each of these types of data access (see for example Section III.1 of the Bioperl tutorial). In addition, if one wanted to change one's mode of sequence-data acquisition (for example, by implementing a local relational database version of Genbank when previously the data had been stored in an indexed flat-file) one had to rewrite all of the data-access subroutines in one's application code.

The Open Biological Database Access (OBDA) System was designed so that one could use the same application code to access data from all three of the database types by simply changing a few lines in a configuration file. This makes application code more portable and easier to maintain. This document shows how to set up the required OBDA-registry configuration file and how to access data from the databases referred to in the configuration file using a perl script as well as from the command line. The Web site for OBDA is *obda.open-bio.org* [http://obda.open-bio.org].

Note

Accessing data via the OBDA system is optional in Bioperl. It is still possible to access sequence data via the old database-format-specific modules such as Bio::Index::Fasta or Bio::DB::Fasta.

2. Using the OBDA Registry System

The OBDA BioDirectory Registry is a platform-independent system for specifying how BioPerl programs find sequence databases. It uses both local and site-wide configuration files, known as the registry, which define one or more databases and the access methods to use to access them.

For instance, you might start out by accessing sequence data over the web, and later decide to install a locally mirrored copy of GenBank. By changing one line in the registry file, all Bio{Perl,Java,Python,Ruby} applications will start using the mirrored local copy automatically - no source code changes are necessary.

3. Installing the Registry File

The registry file should be named seqdatabase.ini. By default, it should be installed in one or more of the following locations:

```
$HOME/.bioinformatics/seqdatabase.ini
/etc/bioinformatics/seqdatabase.ini
```

The Bio{Perl,Java,Python,Ruby} registry-handling code will initialize itself from the registry file located in the home directory first, and then it will read the system-wide default in /etc. Windows Perl users should make sure to set the $HOME variable, otherwise the seqdatabase.ini file may not be found. Unix users need not do this since the code will use the getpwuid() method.

If a local registry file cannot be found, the registry-handling code will attempt to copy the file located at this URL to a $HOME/.bioinformatics directory:

```
http://www.open-bio.org/registry/seqdatabase.ini
```

4. Modifying the Search Path

The registry file search path can be modified by setting the environment variable OBDA_SEARCH_PATH. This variable is a semicolon-delimited string of directories and URLs, for example:

```
OBDA_SEARCH_PATH=/home/lstein/;http://foo.org/
```

Important

Note that the fact that the search path is for an entire file (seqdatabase.ini) rather than for single entry (e.g. 'genbank') means that you have to copy any default values you want to keep from the (old) default configuration file to your new configuration file.

For example, say you have been using biofetch with the default configuration file http://www.open-bio.org/registry/seqdatabase.ini for all your sequence-data retrieval. If you now install a local copy of genbank, your local seqdatabase.ini must not only have a section indicating that the genbank copy is local but it must have sections configuring the web access for all the other databases you use, since http://www.open-bio.org/registry/seqdatabase.ini will no longer be found in a registry-file search.

5. Format of the Registry File

The registry file is a simple text file, as shown in the following example:

```
VERSION=1.00

[embl]
protocol=biofetch
location=http://www.ebi.ac.uk/cgi-bin/dbfetch
dbname=embl

[swissprot]
protocol=biofetch
location=http://www.ebi.ac.uk/cgi-bin/dbfetch
dbname=swall
```

The first line is the registry format version number in the format VERSION=X.XX. The current version is 1.00.

The rest of the file is composed of simple sections, formatted as:

```
[database-name]
tag=value
tag=value

[database-name]
tag=value
tag=value
```

Each section starts with a symbolic database service name enclosed in square brackets. Service names are case-insensitive. The remainder of the section is followed by a series of tag=value pairs that configure access to the service.

Database-name sections can be repeated, in which case the client should try each service in turn from top to bottom.

The options under each section must have two non-optional tag=value lines being

```
protocol="protocol-type"
location="location-string"
```

6. The Protocol Tag

The protocol tag species what access mode to use. Currently it can be one of:

```
flat
biofetch
biosql
```

"flat" is used to fetch sequences from local flat files that have been indexed using BerkeleyDB or binary search indexing.

"biofetch" is used to fetch sequences from web-based databses. Due to restrictions on the use of these databases, this is recommended only for lightweight applications.

"biosql" fetches sequences from BioSQL databases. To use this module you will need to have an instantiated relational database conforming to the BioSQL schema, and install the bioperl-db distribution.

Important

Support for the biosql protocol is disabled as of Bioperl version 1.4. We hope to remedy this in a subsequent release.

7. The Location Tag

The location tag tells the bioperl sequence fetching code where the database is located. Its interpretation depends on the protocol chosen. For example, it might be a directory on the local file system, or a remote URL. See below for protocol-specific details.

8. Other Tags

Any number of additional tag values are allowed. The number and nature of these tags depends on the access protocol selected. Some protocols require no additional tags, whereas others will require several.

Protocol	Tag	Description
flat	location	Directory in which the index is stored. The "config.dat" file generated during indexing must be found in this location
flat	dbname	Name of database
biofetch	location	Base URL for the web service. Currently the only compatible biofetch service is http://www.ebi.ac.uk/cgi-bin/dbfetch
biofetch	dbname	Name of the database. Currently recognized values are "embl" (sequence and protein), "swall" (SwissProt + TREMBL), and "refseq" (NCBI RefSeq entries)
biosql	location	host:port
biosql	dbname	database name
biosql	driver	mysql|postgres|oracle|sybase|sqlserver|access|csv|informix|odbc|rdb
biosql	user	username
biosql	passwd	password
biosql	biodbname	biodatabase name

Table 1. OBDA Protocols

9. Installing Local Databases

If you are using the biofetch protocol, you're all set. You can start reading sequences immediately. For the flat protocol, you will need to create and initialize a local database. See the following documentation on how to do this:

flat protocol:*Flat Databases HOWTO* [http://bioperl.org/HOWTOs/html/Flat_Databases.html]

biosql protocol: BioSQL INSTALL (from the biosql-schema package, available at *obda.open-bio.org* [http://obda.open-bio.org/]). Download the Biosql tar file to view all the documentation.

Important

Support for the biosql protocol in OBDA is disabled as of Bioperl version 1.4. We hope to remedy this in a subsequent release.

Once the flat database is created you can configure your seqdatabase.ini file. Let's say that you have used the bioflat_index.pl script to create the flat database and a new directory called "ppp" has been created in your /home/sally/bioinf/ directory (and the ppp/ directory contains the config.dat file). Your sequence.ini entry should contain these lines:

```
protocol=flat
location=/home/sally/bioinf
dbname=ppp
```

The "database-name" can be any useful name, it does not have to refer to existing files or directories, but the "dbname" should be the name of the newly created directory.

10. Writing code to use the Registry

Once you've set up the OBDA registry file, accessing sequence data from within a perl script is simple. The following example shows how; note that nowhere in the script do you explicitly specify whether the data is stored in a flat file, a local relational database or a database on the internet.

To use the registry from a Perl script, use the following idiom:

```
1 use Bio::DB::Registry;
2 $registry = Bio::DB::Registry->new;
3 $db = $registry->get_database('embl');
4 $seq = $db->get_Seq_by_acc("J02231");
5 print $seq->seq,"\n";
```

In lines 1 and 2, we bring in the Bio::DB::Registry module and create a new Bio::DB::Registry object. We then ask the registry to return a database accessor for the symbolic data source "embl", which must be defined in an [embl] section in the seqdatabase.ini registry file.

The returned accessor is a Bio::DB::RandomAccessI object (see the appropriate manual page), which has just three methods:

```
$db->get_Seq_by_id($id);
$db->get_Seq_by_acc($acc);
$db->get_Seq_by_version($versioned_acc);
```

These methods return Bio::Seq objects by searching for their primary IDs, accession numbers and accession.version numbers respectively. The returned objects have all the methods defined by Bio::Seq (see the appropriate manual page, online at *doc.bioperl.org* [http://doc.bioperl.org]). In line 5, we call the sequence object's seq() method to fetch and print out the DNA or protein string.

11. Using biogetseq to Access Registry Databases

As a convenience, the Bioperl distribution includes the script biogetseq.PLS that enables one to have OBDA access to sequence data from the command line. It's located in the scripts/DB directory of the bioperl distribution (it may also have been installed in your system if you you asked for a script installation after 'make install'. Move or add it into your path to run it. You can get this help text by running it with no arguments:

```
Usage: biogetseq --dbname embl --format embl --namespace acc id [ id ... ]*
       dbname defaults to embl
       format defaults to embl
       namespace defaults to 'acc' ['id', 'acc', 'version']
       rest of the arguments is a list of ids in the given namespace
```

If you have a set of ids you want to fetch from EMBL database, you just give them as space-separated parameters:

```
  % biogetseq J02231 A21530 A10516
```

The output is directed to STDOUT, so it can be redirected to a file. The options can be given in the long "double hyphen" format or abbreviated to one-letter format ("--fasta" or "-f"):

```
 % biogetseq -f fasta -n acc J02231 A21530 A10516 > filed.seq
```