RJE_DBASE: Database Access in Python

Richard J. Edwards (2006)

1 Introduction

This manual is different to the usual program manuals as it covers several accessory modules concerned with downloading and processing sequence databases, rather than the running of a specific program.

Like the software itself, this manual is a 'work in progress' to some degree. If the version you are now reading does not make sense, then it may be worth checking the website to see if a more recent version is available, as indicated by the **1.1 Version** section of the manual. Good luck.

Rich Edwards, 2006.

1.1: Version

This manual is designed to accompany RJE_DBASE version 1.2, RJE_ENSEMBL version 1.1 and RJE_UNIPROT version 2.3.

The manual was last edited on 01 November 2006.

1.2: Using this Manual

As much as possible, I shall try to make a clear distinction between explanatory text (this) and text to be typed at the command-prompt etc. Command prompt text will be `written in Courier New` to make the distinction clearer. Program options, also called 'command-line parameters', will be **`written in bold Courier New`** (and coloured **`red`** for fixed portions or **`dark red`** for user-defined portions, such as file names etc.). Command-line examples will be given in (purple) *`italicised Courier New`*. Optional parameters will (where I remember) be `[`in square brackets`]`. Names of files will be marked in normal text by (dark yellow) **Bold Times New Roman**.

1.3: Getting Help

Much of the information here is also contained in the Accessory Applications website (http://www.bioinformatics.rcsi.ie/~redwards/) and the documentation of the Python modules themselves. A full list of command-line parameters can be printed to screen using the **`help`** option, with short descriptions for each one. See the PEAT Manual for more details and general information on how to run RJE Python programs.

1.4: Availability and Local Installation

See the accompanying **PEAT Appendices** document for details. The Python Modules are open source and may be changed if desired, although please give me credit for any useful bits you pillage. I cannot accept any responsibility if you make changes and the program stops working, however!

Note that the organisation of the modules and the complexity of some of the classes is due to the fact that most of them are designed to be used in a number of different tools. As a result, not all the options listed in the `__doc__()` (**`help`**) will be of relevance. If you want some help understanding the way the modules and classes are set up so you can edit them, just contact me.

2 RJE_DBASE

RJE_DBASE has the following main functions:

1. Automated downloading of sequences databases from remote FTP sites.
2. Indexing UniProt (Bairoch et al. 2005) databases for ease of local extraction.
3. Compressing EnsEMBL (Birney et al. 2006) and UniProt into a combined "EnsLoci" dataset for each genome, consisting of one protein sequence per protein-coding gene.
4. Extracting species codes for specific taxa and generating taxa-specific sequence databases from UniProt, EnsEMBL and IPI (Kersey et al. 2004) (if desired).

It is recommended that an *.ini file is created for routinely downloading and processing all the desired databases (see Downloading Databases). Once downloaded, RJE_ENSEMBL and RJE_UNIPROT can be used for specific processing as described in the relevant sections of this manual.

2.1: Setting up RJE_DBASE

By default, RJE_DBASE is designed to work with EnsEMBL, UniProt and IPI and should be given paths to their locations using the following commands:

Option	Description	Default	Modules affected
unipath=PATH	Path to UniProt files	[UniProt/]	**rje_dbase, rje_ensembl & rje_uniprot**
enpath=PATH	Path to EnsEMBL file	[EnsEMBL/]	**rje_dbase & rje_ensembl**
ipipath=PATH	Path to IPI files	[IPI/]	**rje_dbase**

By default, existing database files will be used if they are younger than the files used to generate them. Setting **force=T** will force regeneration of any existing files. Setting **ignoredate=T** will ignore the relative timestamps of files when assessing whether to regenerate and will only generate missing files.

2.2: Downloading Databases

Automated downloads are controlled using the **download=FILE** option. RJE_DBASE uses the **rje_xml.py** module to parse an (attempt at) XML formatted file which contains information on the database name, ftp site, files to download and directory to download into. E.g. for UniProt, the file looks something like this:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<database name="UniProt"
ftproot="ftp://ftp.ebi.ac.uk/pub/databases/uniprot/current_release/knowledgebase/complete/" outdir="UniProt/">

  <file path="*dat.gz">SwissProt and TrEMBL</file>

</database>
```

This downloads a database called `"UniProt"` into the directory `"UniProt/"` by going to the ftp site `"ftp://ftp.ebi.ac.uk/pub/databases/uniprot/current_release/knowledge`

base/complete/" and downloading all "*dat.gz" files. This will be unzipped and/or untarred if appropriate.

2.3: Database Processing Options

2.3.1 UniProt Index

Extraction and manipulation of UniProt is handled by the **rje_uniprot** module. However, indexing is performed by RJE_DBASE using the **index=T/F** option, which creates an index file (**uniprot.index**) for the Uniprot ***.dat** files found in the path specified by **unipath=PATH**. If this table already exists, regeneration is controlled by **force=T/F** and **ignoredate=T/F** (see **2.1: Setting up RJE_DBASE**).

2.3.2 Database Reformatting

This is now a bit obsolete and will not be described until someone asks for details! In essence, the **dbformat=T** option reformats UniProt, IPI and EnsEMBL databases to a format that is like that described in the RJE_SEQ manual (and not dissimilar to the EnsLoci format), containing a gene or database identifier, species code and accession number in the first word of the sequence name, followed by the description.

2.4: Generation of Taxa Lists and Custom Databases

2.4.1 UniProt Species Table

To extract specific taxanomic groups from databases, the **spectable=T** option, which generates a table of species codes, taxonomy and taxon_id from the UniProt ***.dat** files found in the path specified by **unipath=PATH**. If this table already exists, regeneration is controlled by **force=T/F** and **ignoredate=T/F** (see **2.1: Setting up RJE_DBASE**).

2.4.2 Custom Database Manufacture

The following Database and sub-database manufacture options will be explained in a future version of this manual:

- makedb=FILE : Makes a database from combined databases [None]
  - Note that rje_seq commandline options will be applied to this database with the addition of a goodspec=X filter applied from the taxalist=LIST
- formatdb=T/F : Whether to BLAST format database after making [True]
- useX=T/F : Whether to use certain aspects of databases, where X is: uniprot/sprot/trembl/ensembl/known/novel/abinitio/ipi [All but ipi True]
- taxalist=LIST : List of taxanomic groups to extract spec_codes for reduced database (else all) [None]
- speconly=T/F : Will simply output a list of SPECIES codes to the makedb file, rather than making dbase [False]
- inversedb=T/F : TaxaList is a list of taxanomic groups *NOT* to be in database [False]
- screenipi=T/F : Species represented by IPI databases will be screened out of UniProt and EnsEMBL. [False]
- screenens=T/F : Species represented by EnsEMBL will be screened out of UniProt [True]

- ensloci=T/F : Reduce EnsEMBL to a single protein per locus, mapping UniProt where possible [True]
- ensfilter=T/F : Run EnsEMBL genomes through rje_seq to apply filters, rather than just concatenating [False]

2.4.3 Taxanomic Databases

In addition to the single custom databases that can be made as with the options above, RJE_DBASE can also read in a taxadb XML file using the **taxadb=FILE** command. Again, **rje_xml.py** is used to parse out details from a file in the form:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<taxadb>
  <makedb makedb="TaxaDB/embl_metazoa.fas"
taxalist=":Metazoa:">Metazoa</makedb>
  <makedb makedb="TaxaDB/embl_archaea.fas"
taxalist=":Archaea:">Archae</makedb>
  <makedb makedb="TaxaDB/embl_euk_extra.fas"
taxalist=":Fungi:,:Metazoa:,:Viridiplantae:,:Bacteria:,:Bacteria.:,:
Viruses:,:Viruses.:,:Archaea:" inversedb="True">EukExtra</makedb>
</taxadb>
```

Each makedb element contains a list of commandline options from **2.4.2: Custom Database Manufacture** allowing any combination of databases to be generated.

The default for our local running makes the following taxanomic databases in a **TaxaDB** directory:

- embl_metazoa.fas = Metazoa
- embl_vertebrata.fas = Vertebrata
- embl_insecta.fas = Insecta
- embl_fungi.fas = Fungi
- embl_viridiplantae.fas = Plants
- embl_bacteria.fas = Bacteria
- embl_viruses.fas = Viruses
- embl_archaea.fas = Archae
- embl_euk_extra.fas = Everything else

The default options will include "EnsLoci" genomes for the appropriate taxa and then all UniProt sequences from the relevant file *except those represented by EnsEMBL genomes*.

3 HTML

The RJE_ENSEMBL modules is primarily designed to be used by RJE_DBASE but can be run in a standalone capacity to download EnsEMBL and, where UniProt downloads already exist, generate the "EnsLoci" genomic datasets.

3.1: Downloading EnsEMBL Databases

The local computer must have external FTP access. The program contacts ftp://ftp.ensembl.org/pub and identifies all available species with a "current_X" directory. Where these species are not recognised by the program, the presence of new species will be recorded. Species codes are extracted from UniProt.

For each species, the data/fasta/pep/ folder is entered and the known, known-ccds, novel, and abinitio ***.fa.gz** files are downloaded and unzipped into a subdirectory created in the directory specified by the **enspath=PATH** parameter. In addition, the **gene.txt.table** and **gene_stable_id.txt.table** files are downloaded from data/mysql/.

3.2: Generating "EnsLoci" Genomic Datasets

The main originality and utility of RJE_ENSEMBL lies in its ability to manufacture a single well-annotated fasta file for each genome, from the EnsEMBL and UniProt downloads. For this, the known, novel and, where available, known-ccds files are used.

The aim of this is to generate a comprehensive, non-redundant sequence dataset, containing one protein per gene, with informative gene descriptions and links to external databases where available. The disadvantage of this compared to IPI, for example, is that there are no splice variants. However, the advantage is a reduction in database size and a removal of problems for orthology identification etc. that splice variants can introduce.

The generation process is as follows:

1. The downloaded EnsEMBL fasta files and MySQL tables are parsed and the links established between:
    a. EnsEMBL peptides and EnsEMBL genes
    b. EnsEMBL genes and external database IDs

    For each EnsEMBL gene, the useful description is extracted from the MySQL table.
2. External database AccNums are extracted from UniProt, where available.
3. Each locus (protein-coding gene) is taken in turn and the "best" EnsEMBL peptide retained. If the gene maps to a SwissProt sequence, and one of the peptides mapped to that gene has an *identical sequence* to the SwissProt entry (hereafter referred to as an "exact match") then that peptide is used. In all other cases, the longest peptide is used. Sequence names are generated as described below and the sequence is output into a file **ens_SPECIES.loci.fas**, where **SPECIES** is the species code.

In addition, details of the numbers of peptides and loci etc. are output to a file **ens_loci.tdt**.

3.2.1 nsL c e enc e a n nvent n

The resulting file names have the following format:

ID_SPECIES__AccNum Description [acc:**DBAcc** pep:**EnsPep** gene:**EnsGene**]

This is comprised of the following elements (see above for details of how the file and sequence mappings were made):

- **ID.** This is the gene ID. If the sequence was mapped to SwissProt then this is the upper case SwissProt gene ID. If it was mapped exactly to TrEMBL then a lower case gene identifier from the UniProt entry is used. If it was mapped from RefSeq, then the ID is "ref". In all other cases, the ID is "ens" for an EnsEMBL Known sequence, or "nvl" for an EnsEMBL novel sequence.
- **SPECIES**. This is the UniProt species code for that species.
- **AccNum**. If the sequence was an exact Primary UniProt Accession Number mapping, then the UniProt accession number is used here. Otherwise, the EnsEMBL peptide accession number is used. The exception is yeast, which uses the standard SGD yeast protein accession number.
- **Description**. This is the description for the gene, taken from the EnsEMBL MySQL table gene.txt.table. If no description is given and the sequence maps to a UniProt sequence, the UniProt description is used. Else, if no description is given, "EnsEMBL Known" or "EnsEMBL Novel" is used.
- **DBAcc**. This is the database accession number that was used by EnsEMBL to map the "known" gene. It is either UniProt or RefSeq. Where it is neither (or missing from the MySQL table), the EnsEMBL peptide ID is used instead.
- **EnsPep**. This is the EnsEMBL peptide ID for the sequence used. This is either the SwissProt sequence, if found, or the longest peptide for a given gene.
- **EnsGene**. This is the EnsEMBL gene ID for the gene.

RJE_UNIPROT

The RJE_UNIPROT module has two main functional aims:

1. To provide **UniProt** and **UniProtEntry** classes for handling uniprot files and entries, respectively, in other Python programs.
2. To provide standalone functionality for extracting entries and useful data from UniProt files.

This section deals with the second of these functions. This documentary is still in its preliminary stages and thus explanations may be a bit thin. If you want more explanation of any areas, please contact me.

4.1: Standalone Functions

For all standalone functions, RJE_UNIPROT will use the index file in the directory specified by **unipath=PATH**. This is the index file made using RJE_DBASE (see **2.3.1: UniProt Indexing**) and by default is called **uniprot.index**. (The **dbindex=FILE** option can make the module look for an alternatively named index file.) To use a file without indexing it, the **uniprot=FILE** option can also be used to specify the input file.

4.1.1 Extracting UniProt Entries

Entries can be extracted from the specified UniProt file(s) using either **extract=LIST** or **acclist=LIST**, where LIST can be file or list of UniProt IDs/AccNums X,Y,... etc. If the list contains splice variants (e.g. AccNum-X, where X is a number) then the **splicevar=T** should be used. The name of the new UniProt DAT file of extracted sequences is specified with **datout=FILE**.

The RJE_SEQ module can then be used if desired to convert this file to a fasta format file:

```
python rje_seq.py seqin=DATFILE seqout=FASFILE reformat=fasta
```

4.1.2 Additional Outputs

In addition to making a new DAT flat file, there are three summary table output options:

- **tabout=FILE** : Table of extracted UniProt details, such as functional annotation, pubmed links etc.
- **linkout=FILE** : Table of extracted Database links
- **domtable=T/F**: Table of domains from features lines

5 Appendices

5.1: Appendix I: Command-line Options

General details on command-line options can now be found in the **PEAT Appendices** document distributed with this program. A full list of options can be found in the distributed **readme.txt** and **readme.html** files.

5.2: Appendix II: Distributed Python Modules

Details can be found in the distributed **readme.txt** and **readme.html** files as well as the accompanying **PEAT Appendices** document.

5.3: Appendix III: Log Files

See the accompanying **PEAT Appendices** document for general information on log files.

5.4: Appendix IV: Troubleshooting

There are currently no specific Troubleshooting issues arising with this software. Please see general items in the **PEAT Appendices** document and contact me if you experience any problems not covered.

5.5: Appendix VI: References

Bairoch, A. et al. (2005) The Universal Protein Resource (UniProt), *Nucleic Acids Res.*, **33**, D154-159.

Birney, E. et al. (2006) Ensembl 2006, *Nucleic Acids Res.*, **34**, D556-561.

Kersey, P.J. et al. (2004) The International Protein Index: an integrated database for proteomics experiments, *Proteomics.*, **4**, 1985-1988.