Introduction to EMBOSS

EMBnet

What is EMBOSS?

- Wisconsin package, GCG
- Widely used, sources available for inspection
- 1988 - EGCG - academic add-on started
- GCG commercial - sources **not** freely available!
- 1999 - EGCG split from GCG to become **EMBOSS**

What is EMBOSS!

- A new suite of programs
- Open source software - sources available
- Public domain (GNU Public Licence)
- Written by HGMP/Sanger/EBI/Norway … etc

What it aims to do

- A useful, integrated set of programs
- They share a common look and feel
- Incorporates many small and large programs
- Easy to run from the command line
- Easy to call from other programs (e.g. perl)
- Easy to set up behind GUIs and Web interfaces

Scope of applications

- There are many EMBOSS programs (200+)
- See:

http://www.emboss.org

- Many sequence analysis & display programs.
- Protein 3D structure prediction being developed.
- Other assorted programs, eg: enzyme kinetics.

An example EMBOSS program

- It is easy to forget the name of a program.
- To find EMBOSS programs, use **wossname**
- **wossname** finds programs by looking for keywords in the description or the name of the program.

Running at the command-line

- Type **wossname** at the Unix % prompt

  Unix % **wossname**
- Displays one-line description.
- Prompts you for information:

```
Finds programs by keywords in their one-line documentation
Keyword to search for: restrict

SEARCH FOR 'RESTRICT'
recode          Remove restriction sites but maintain the
   same translation
remap           Display a sequence with restriction cut
   sites, translation
```

etc…..

Optional parameters

```
Unix % wossname -opt
Finds programs by keywords in their one-line
   documentation
Keyword to search for: protein
Output program details to a file [stdout]: myfile
Format the output for HTML [N]: Y
String to form the first half of an HTML link:
String to form the second half of an HTML link:
Output only the group names [N]:
Output an alphabetic list of programs [N]:
Use the expanded group name [N]:
```

Help

```
Unix % wossname -help
 Mandatory qualifiers:
  [-search]    string      Enter a word or words here.

 Optional qualifiers (* if not always prompted):
   -outfile    outfile this program will write the program names

 Advanced qualifiers:
   -[no]emboss bool        EMBOSS program
                           documentation will be searched.
```

- Mandatory - required, are often parameters (in '[]')
- Optional - use **-opt** to be prompted for these.
- Advanced - things that are not often used!

Writing to the screen

- Note that the default output file for wossname was:

  stdout (Standard output)
- Use this whenever prompted for an output file.
- This is a ‘magic’ file name.
- It displays the output on the screen, not a file.

Practical

- Try running **wossname**
- Can you find a program to:

- Display multiple alignments.
- Find ORFs (Open Reading Frames).
- Translate a sequence.
- Find restriction enzyme sites
- Find the isoelectric point of a protein.
- Do global alignments.

Working with sequences

- EMBOSS reads sequences from **files** or **databases**.
- It automatically recognises the input sequence format.
- You can easily specify many output formats.

Getting sequences from the databases

- Database single entry (ID)
  - **database:entry**
  - For example embl:hsfau
- Wildcarded entries (Query)
  - **database:hs***
- All entries
  - **database:***
- Most databases will support all 3 methods - some may not.

showdb

Unix % **showdb**

Displays information on the currently available databases

```
#Name          Type ID  Qry All Comment
#====          ==== ==  === === =======
pir             P    OK  OK  OK  PIR/NBRF
remtrembl       P    OK  OK  OK  REMTREMBL sequences
sptrembl        P    OK  OK  OK  SPTREMBL sequences
swissprot       P    OK  OK  OK  SWISSPROT sequences
embl            N    OK  OK  OK  EMBL sequences
emblnew         N    OK  OK  OK  New EMBL sequences
est             N    OK  OK  OK  EMBL EST sequences
```

seqret

- Reads in a sequence, and writes it out.

```
Unix % seqret
Reads and writes (returns) a sequence

Input sequence: embl:xlrhodop
Output sequence [xlrhodop.fasta]:

unix % more xlrhodop.fasta

>XLRHODOP L07770 Xenopus laevis rhodopsin
ggtagaacagcttcagttgggatcacaggcttctagggatcctttgggcaaaaaagaaac
acagaaggcattctttctatacaagaaaggactttatagagctgctaccatgaacggaac
        .
        .
```

seqret from the command line

- Give **seqret** all of its data on the command-line.
- It doesn't need to prompt for anything else.

Unix % **seqret embl:xlrhodop -outseq xlrhodop.fasta**

- The **'-outseq'** can be abbreviated to **'-out'**.
- Any abbreviation must be unique.

- Even shorter, leave out the qualifier:

Unix % **seqret embl:xlrhodop xlrhodop.fasta**

Changing output formats (reformatting)

- **seqret** can reformat sequences by specifying the output format:

Unix % **seqret embl:xlrhodop xlrhodop.fasta -osformat gcg**

Unix % **more xlrhodop.gcg**

```
!!NA_SEQUENCE 1.0
Xenopus laevis rhodopsin mRNA, complete cds.
XLRHODOP  Length: 1684  Type: N  Check: 9453 ..
    1 ggtagaacag cttcagttgg gatcacaggc ttctagggat cctttgggca
   51 aaaaagaaac acagaaggca ttctttctat acaagaaagg actttataga
      .
      .
```

Reading sequences from files

- Just give the name of the file:

Unix % **seqret myclone.seq gcg::myclone.gcg**

- You may specify the input format (not required):

Unix % **seqret gcg::myclone.gcg clone2.seq**

- A sequence from a file of many sequences:

Unix % **seqret allclones.seq:52H12 52H12.seq**

List files (files of file names)

- A quick way of grouping sequences to work on, like a private database.
- Any valid sequence specification can be used, not just file names.
- One entry per line in a file.
- Comment lines start with a '#'
- Indicate that it is a list file by starting it with a '@':

  Unix % **infoseq @mylist**
- Many programs (infoseq, fuzznuc, fuzzpro) can write out list files from a search (use '**-usa**' option)

Multiple sequences, single file

- **EMBOSS** writes many sequences to a single file.
- Most sequence formats can deal with this:
  - Fasta, EMBL, PIR, MSF, Clustal, Phylip, etc.
- BUT NOT: Plain, Staden and GCG
- **EMBOSS** reads many sequences from a single file.
- Use **filename:entryname** if you wish to specify a single sequence.
- If there is only one sequence, or you wish to read all entries, use just the **filename.**

Multiple sequences, many files

- If you wish to write one sequence per file, use:
  '-ossingle'

Unix % **seqret "embl:hsf*" dummy -ossingle**

- The output filenames will be based on the sequence entry names.

- The program **seretsplit** will split an existing multiple sequence file into many files.

Asterisk on the command line

- You can't use a ‘*’ on the UNIX command-line.
- UNIX tries to match it to filenames.
- Use it quoted, either with quotes or a backslash:

  "embl:*"

  embl:*

- For example:

Unix % seqret “embl:hsf*” hsf.seq

Practical

- Try running **showdb**, **seqret** and **infoseq:**

- Show just the nucleic databases
- Get the sequence entry '**hsfau**' from the EMBL database into the file '**this.seq'**.
- Ditto, but into the file '**this.gcg**' in GCG format.
- Display information on the sequence in '**this.seq**'.
- Display information on all sequences whose name starts with '**10**' in the SwissProt database.

GUIs

- There are many interfaces available or coming soon:
- **wEMBOSS** - web interface
- **EMBOSSgui** - web interface
- **spin** - from the Staden team
- many others, also in commercial packages

Conclusion - help

- If in doubt, use:

  wossname

  program -help

  program -opt

  tfm program

Conclusion - sequence data

- For database information, use

  showdb

- Uniform Sequence Addresses (USAs):
  - **database**
  - **database:entry_name** or **database:accession_number**
  - **database:wildcard**
  - **filename**
  - **filename:entry**
  - **format::filename**
  - **@list**

Conclusion - other qualifiers

- **-sbegin** sequence begin position
- **-send** sequence end position
- **-sreverse** reverse complement the sequence
- **-slower** change sequence to lower case
- **-supper** change sequence to upper case
- **-osformat** output sequence format
- **-help** show help
- **-options** ask for optional parameters
- **-auto** run silently (for use in scripts, e.g. perl)

Training training training training!

- When at home read again the tutorials, repeat the concept explanations, learn and remember the difference between the different alignment methods
- Learn about biological database characteristics and limitations. Remember all databases are “man made”!