

Entrez Direct

BLAST+ и EMBOSS

Иван Русинов

EMBOSS

EMBOSS

European Molecular Biology Open Software Suite

Пакет консольных биоинформатических программ.

- ▶ унифицированный интерфейс
- ▶ общий формат для задания адреса последовательностей (USA)
- ▶ есть программы для большинства повседневных задач, возникающих при работе с биологическими последовательностями
- ▶ пакет перестал развиваться в 2013, программы устаревают

Помощь по программам

Можно получить справку в командной строке:

Краткое описание основных опций:

```
$ 'any-emboss-util' -help
```

Описание всех имеющихся опций:

```
$ 'any-emboss-util' -help -verbose
```

Подробное описание команды:

```
$ tfm 'any-emboss-util'
```

Поиск программы по описанию:

```
$ wosname 'alignment'
```

У всех программ есть man, по объему это примерно -help

```
$ man 'any-emboss-util'
```

Или можно читать описания в интернете:

<http://emboss.open-bio.org/> путанный официальный сайт

<http://emboss.sourceforge.net/> лучше организован, но у меня постоянно висит

Унифицированный адрес последовательности (USA)

Uniform Sequence Address

```
DB:entry[start:end:reverse]  
format::file:entry[start:end:reverse]  
@listfile
```

Все варианты USA описаны здесь:

<http://emboss.sourceforge.net/docs/themes/UniformSequenceAddress.html>

Список поддерживаемых форматов файлов доступен здесь:

<http://emboss.sourceforge.net/docs/themes/SequenceFormats.html>

Список баз данных можно узнать с помощью команды `showdb`. На `kodomo` есть локальная копия Swiss-Prot, и настроено скачивание одиночных записей из ENA/DDBJ и UniProtKB.

В именах файлов и записей можно использовать маски (с помощью символов `*` и `?`). Не забывайте про экранирование!

Аргументы командной строки

- ▶ аргументы называются qualifiers
- ▶ бывают пяти типов: standard, additional, advanced, associated и general
- ▶ всегда задаются в виде опций, начинающихся с *одного* символа -
- ▶ название опции можно сокращать, пока понятно, какая опция имеется в виду
- ▶ нельзя склеивать названия нескольких опций после одного -
- ▶ почти все опции требуют один аргумент
- ▶ у опций типа boolean аргумент можно опускать, имея в виду значение Y

Standard qualifiers

Обязательные аргументы

- ▶ если не заданы, будут запрошены с STDIN в процессе исполнения
- ▶ иногда могут задаваться в виде позиционных аргументов (т.е. без указания названия опции), в этом случае название опции заключено в [] на странице `-help`
- ▶ иногда для них есть значение по умолчанию, которое можно активировать опцией `-auto`

Пример:

```
$ infoseq -sequence 'seq.fasta'
```

или (то же самое):

```
$ infoseq 'seq.fasta'
```

Additional qualifiers

Дополнительные аргументы

- ▶ если не заданы, будут использованы значения по умолчанию (будут запрошены с STDIN в интерактивном режиме, если задана опция `-options`)
- ▶ значения по умолчанию указаны в [] на странице `-help`

Пример:

```
$ infoseq seq.fasta -outfile 'report.txt'
```


Advanced qualifiers

"Расширенные" аргументы

- ▶ предполагается, что они редко потребуются рядовым пользователям
- ▶ отображаются на странице `-help` без опции `-verbose`

Пример:

```
$ infoseq seq.fasta -delimiter ';' 
```

Associated qualifiers

"Ассоциированные" аргументы

- ▶ уточняют значения других аргументов
- ▶ не отображаются на странице `-help` без опции `-verbose`
- ▶ на странице `-help -verbose` указано, какой аргумент они уточняют

Пример:

```
$ infoseq seq.fasta -squick 'Y'
```

General qualifiers

Общие аргументы

- ▶ есть у всех программ EMBOSS
- ▶ не отображаются на странице `-help` без опции `-verbose` (за исключением самой опции `-help`)
- ▶ служат либо для получения служебной информации о программе, либо для переключения режима взаимодействия с программой

Пример:

```
$ infoseq -help 'Y' -verbose 'N'
```

или (то же самое):

```
$ infoseq -help
```

Использование в конвейерах

Программы пакета EMBOSS неудобно использовать в конвейерах, так как они:

- ▶ используют файловый ввод/вывод (а не стандартные потоки);
- ▶ переключаются в интерактивный режим в случае указания не всех обязательных аргументов (даже при наличии подходящих умолчательных значений);
- ▶ выводят бесполезные информационные сообщения.

Но есть общие (general) опции, позволяющие решить некоторые или все проблемы:

- auto** – использовать умолчательные значения даже для пропущенных обязательных аргументов (+ отключить информационные сообщения);
- filter** – превратить программу в нормальную (использовать умолчательные значения для пропущенных аргументов, заменить умолчательные ввод/вывод на стандартные потоки, отключить сообщения).

Советую *всегда* использовать `-filter`, не могу придумать ситуацию, когда она помешает.

Проблемы с выводом сообщений

Все информационные сообщения, в том числе `-help`, программы EMBOSS выводят на `STDERR`, а не на `STDOUT`.

Слить `STDOUT` и `STDERR` и перенаправить в файл:

```
$ seqret -help &> 'seqret_help.txt'
```

Слить `STDOUT` и `STDERR` и передать следующей команде:

```
$ seqret -help -verbose |& less
```

Убить `STDERR` (перенаправить в черную дыру):

```
$ seqret 'seqs.fasta' 'plain::stdout' 2> '/dev/null' | less
```

Отключить сообщения на уровне команды EMBOSS:

```
$ seqret -filter 'seqs.fasta' | less
```

или

```
$ seqret -auto 'seqs.fasta' 'out.fasta'
```

Разбиение fasta на отдельные файлы

Для этого есть `seqretsplit`, вот только задание имен выходных файлов совсем не интуитивное (да еще и глюков полно).

Имя выходных файлов имеет вид `DIR/NAME.FORMAT`

DIR по умолчанию – текущая папка; можно задать с помощью ассоциированной опции `-osdirectory`

NAME идентификатор последовательности (поменять нельзя)

FORMAT всегда `fasta`; причем можно изменить фактический формат выходных файлов (например, с помощью `USA` и `-outseq`), но расширение от этого не изменится 😞

Standalone BLAST

Пакет BLAST+

- ▶ BLAST+ – пакет консольных утилит, позволяющих запускать локальный (standalone) BLAST.
- ▶ Есть версии BLAST+ для Windows, MacOS X и Linux.
- ▶ BLAST+ установлен на kodoMO.
- ▶ Для локального BLAST необходимо создать (или загрузить) базу последовательностей в специальном формате.

Создание/загрузка базы

Создание базы с помощью makeblastdb:

```
$ makeblastdb -in 'seqs.fasta' -dbtype 'nucl'
```

Загрузка готовой базы из NCBI с помощью update_blastdb.

- ▶ Можно загрузить одну из баз, доступную для выбора в веб-интерфейсе.
- ▶ Базы весят десятки и даже сотни гигабайт.
- ▶ Если понадобится, разберетесь сами.

Запуск локального BLAST

Запуск blastn:

```
$ blastn -task 'blastn' -query 'query.fasta' -db 'seqs.fasta'
```

-task тип алгоритма (blastn, megablast, dc-megablast и другие)

-query последовательность-запрос в формате fasta

-db *имя* базы для поиска (это не fasta файл!)

Доступные команды:

- ▶ blastn, blastp, blastx, tblastn, tblastx
- ▶ psiblast
- ▶ rpstblastn, rpsblast+
- ▶ deltablast

Формат вывода

Можно изменить формат выходного файла с помощью опции `-outfmt`. Особенно полезны "табличные" форматы 6, 7 и 10, которые можно дальше кастомизировать, указывая аббревиатуры столбцов.

```
$ blastn -task 'blastn' -query 'query.fasta' -db 'seqs.fasta' \  
-outfmt '7 qseqid sseqid slen evalue' -out 'output.tsv'
```

Entrez Direct (EDirect)

Entrez

Единая поисковая система NCBI, которая объединяет все (или почти все) базы данных.

- ▶ У каждой базы данных в NCBI есть свое имя в системе Entrez.
Пример: nucscore – название базы Nucleotide.
- ▶ Для каждой базы определены поля записей, по которым можно производить поиск.
Пример: TIAB – поле Title/Abstract в PubMed.
- ▶ Между записями в базах данных есть ссылки, каждому типу ссылок присвоено свое имя.
Пример: pubmed_pubmed_refs – ссылка из статьи в PubMed на цитируемую статью.

Entrez API

У системы Entrez есть API, который позволяет использовать возможности Entrez в скриптах.

E-utilities основной Web API, все остальное работает через него.

Не советую использовать напрямую!

Entrez Direct набор консольных утилит для Unix-подобных систем, установлены на kodo.com.

Bio.Entrez модуль Biopython для работы с EUtils.

...

Entrez Direct (EDirect)

Набор консольных утилит:

einfo получение списка названий баз данных, полей и ссылок

esearch поисковые запросы к системе Entrez

elink получение записей по ссылкам из других записей

efetch скачивание записей по идентификаторам

esummary получение основных полей записей

efilter фильтрация результатов поиска

epost отправка идентификаторов записей для дальнейшей обработки

xtract извлечение отдельных полей из выдачи в формате XML

...

Entrez History и конвейеры

Entrez хранит историю запросов, и результатов их исполнения.

- ▶ Каждому запросу присваивается идентификатор WebEnv, по которому можно получить найденные записи.
- ▶ Утилиты EDirect могут работать напрямую с Entrez History, что позволяет производить операции с записями без загрузки их на локальный компьютер.
- ▶ Такая система позволяет объединять вызов утилит в конвейеры: между программами передается идентификатор запроса (и некоторая сопутствующая информация в формате XML), а операции с записями происходят на серверах NCBI.

Entrez History и конвейеры

Пример: поиск таксона по общепринятому названию, получение ссылок на геномные сборки и загрузка их идентификаторов:

```
$ esearch -db 'taxonomy' -query '"White shark"[COMN]' \  
  | elink -target 'assembly' \  
  | efetch -format 'uid'  
9678721  
9678001  
2022931
```

EInfo

Получение информации о базах данных в системе Entrez:

Список баз данных

```
$ einfo -dbs
```

Список полей в базе данных

```
$ einfo -db 'taxonomy' -fields
```

Список названий ссылок на другие базы данных

```
$ einfo -db 'taxonomy' -links
```

Вся доступная информация про базу в формате JSON

```
$ einfo -db 'taxonomy' | transmute -x2j
```

EPost

Отправка списка ID или AC записей в Entrez History.

С этими записями потом можно работать так же, как с результатами других запросов: фильтровать, переходить по ссылкам, скачивать и т.д.

Можно указать список идентификаторов в качестве аргумента

```
$ epost -db 'assembly' -id '9678721,2022931'
```

Или можно указать AC записей

```
$ epost -db 'assembly' -id 'GCF_017639515.1' -format 'acc'
```

Источником идентификаторов может быть файл или STDIN

```
$ epost -db 'protein' -input 'protein.ids'
```

```
$ echo '9678721' | epost -db 'assembly'
```

EFetch и ESummary

Загрузка записей из базы данных:

По списку ID (в некоторых случаях можно AC)

```
$ efetch -db 'protein' -id 'AAC74937.2,BAA15678.1' -format 'fasta'
```

Записи по идентификатору запроса (в виде XML на STDIN)

```
$ epost -db 'protein' -id 'AAC74937.2' | efetch -format 'ft'
```

Список форматов зависит от базы

```
$ efetch -help # список неполный
```

ESummary – это синоним EFetch с опцией `-format 'docsum'`.

ESearch

Поисковые запросы к системе Entrez:

В теории понимает полный синтаксис запросов Entrez:

```
$ esearch -query 'HhaI[TITL] AND Roberts*[AUTH] AND NAR[JOUR]' \  
-db 'pubmed' | efetch -format 'uid'
```

31879785

9207024

7753630

7899082

8506140

ELink

Работа с перекрестными ссылками:

Получение ссылок на записи в других базах данных

```
$ epost -db 'protein' -id 'AAC74937.2,CAB15395.1' \  
  | elink -target 'taxonomy' -cmd 'neighbor' \  
  | xtract -pattern 'LinkSet' -element 'IdList/Id,Link/Id'
```

Переход по ссылкам на другие записи

```
$ epost -db 'protein' -id 'AAC74937.2,CAB15395.1' \  
  | elink -target 'nuccore' | esummary -json
```

Помощь по EDirect

У программ есть справочные страницы в системе man и встроенная помощь:

```
$ man edirect  
$ epost -help | less
```

Подробнее можно почитать в руководствах на сайте NCBI:

EDirect <https://www.ncbi.nlm.nih.gov/books/NBK179288>

EUtils <https://www.ncbi.nlm.nih.gov/books/NBK25501>

Entrez <https://www.ncbi.nlm.nih.gov/books/NBK3837>