

Entrez Direct и NCBI Datasets CLI и EMBOSS

EMBOSS

EMBOSS

European Molecular Biology Open Software Suite

Пакет консольных биоинформатических программ.

- ▶ унифицированный интерфейс командной строки
- ▶ общий формат для задания адреса последовательностей (USA)
- ▶ есть программы для 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:

```
kodomo:~$ any-emboss-util -help
```

Описание всех имеющихся опций:

```
kodomo:~$ any-emboss-util -help -verbose
```

Подробное описание программы:

```
kodomo:~$ tfm 'any-emboss-util'
```

Поиск программы по описанию:

```
kodomo:~$ wosname 'alignment'
```

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

```
kodomo:~$ 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

Все варианты USA:

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

Примеры (жирным выделены обязательные части):

Запись entry из базы данных dbname (аббревиатуры баз выдает showdb):

dbname:entry[start:end:reverse]

Запись entry из файла file в формате format (названия форматов [здесь](#)):

format::file:entry[start:end:reverse]

Использовать все USA из файла listfile:

@listfile

В именах файлов и записей можно использовать маски (только * 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:

```
kodomo:~$ infoseq -sequence 'seq.fasta'
```

или (то же самое):

```
kodomo:~$ 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:

```
kodomo:~$ 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:

```
kodomo:~$ 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:

```
kodomo:~$ 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:

```
kodomo:~$ infoseq -help 'Y' -verbose 'N'
```

Использование в конвейерах

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

- ▶ по умолчанию 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.

Использование в конвейерах

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

- auto – использовать умолчательные значения даже для пропущенных обязательных аргументов (+ отключить информационные сообщения);
- filter – то же самое + заменить *умолчательные* ввод и вывод на стандартные потоки.

Опция `-filter` переключает потоки, только если ввод/вывод не указаны в аргументах явно. Советую *всегда* использовать эту опцию, не могу придумать ситуацию, когда она помешает.

В USA можно использовать специальные имена файлов `stdin` и `stdout`.

Изменить формат выдачи в терминал:

```
kodomo:~$ seqret 'seqs.fasta' 'plain::stdout'
```

Транслировать последовательность из потока ввода:

```
kodomo:~$ echo 'ATGC' | seqret -filter 'plain::stdin[::r]'
```

Проблемы с выводом сообщений

Программы EMBOSS выводят справку на `STDERR`, а не на `STDOUT`.

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

```
kodomo:~$ seqret -help &> 'seqret_help.txt'
```

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

```
kodomo:~$ seqret -help -verbose |& less
```

Перед началом работы программы EMBOSS выводят свое краткое описание на `STDERR`.

Отключить сообщения на уровне команды EMBOSS:

```
kodomo:~$ seqret -filter 'seqs.fasta' | less
```

или

```
kodomo:~$ seqret -auto 'seqs.fasta' 'out.fasta'
```

Убить `STDERR` (перенаправить в черную дыру):

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

Программа seqret

от **sequence** **return**

Получает последовательности согласно USA из первого аргумента и записывает их согласно USA из второго аргумента (можно указать только имя файла и его формат).

Изменить формат файла:

```
kodomo:~$ seqret -filter 'seqs.fasta' 'asn1::seqs.txt'
```

Вырезать участок последовательности:

```
kodomo:~$ seqret -filter 'seqs.fasta:*[1:1]' 'first_aa.fasta'
```

Отобрать последовательности по маске имени:

```
kodomo:~$ seqret -filter 'seqs.fasta:abc*' 'ncbi::abc.txt'
```

Объединить разные последовательности в один файл:

```
kodomo:~$ cat usa.list
```

```
seqs.fasta:qwe123[1:30]
```

```
seqs.fasta:qwe123[31:60]
```

```
ncbi::abc.txt
```

```
kodomo:~$ seqret -filter '@usa.list' 'ncbi::all.fasta'
```

Программа entret

от **entry** **return**

Нужна для скачивания записей из баз данных.

Получить последовательность из базы можно с помощью seqret:

```
kodomo:~$ seqret -filter 'sw:ENO_ECOLI' eno_ecoli.fasta
```

Можно даже сохранить её в нативном формате:

```
kodomo:~$ seqret -filter 'sw:ENO_ECOLI' 'swiss::eno_ecoli.seqret'
```

Но seqret не гарантирует сохранение всех аннотаций!

```
kodomo:~$ grep -c '^FT' 'eno_ecoli.seqret'
```

0

Скачивать полные записи следует с помощью entret:

```
kodomo:~$ entret -filter 'sw:ENO_ECOLI' 'eno_ecoli.entret'
```

```
kodomo:~$ grep -c '^FT' 'eno_ecoli.entret'
```

151

Второй аргумент entret – **не** USA, а просто имя файла.

Другие полезные программы

- `infoseq` – получить базовую информацию про последовательности в виде таблицы
- `infoalign` – сравнить последовательности в выравнивании с консенсусом
- `getorf` – найти открытые рамки считывания в нуклеотидной последовательности
- `transeq` – транслировать нуклеотидные последовательности в одной или нескольких рамках
- `wordcount` – посчитать количество вхождений для k-меров заданной длины
- `shuffleseq` – перемешать буквы в последовательностях с сохранением состава
- `makenucseq` – сгенерировать случайные нуклеотидные последовательности
- `makeprotseq` – аналогично для белков
- ... – программы выравниваний, поиска паттернов, работы с аннотациями и т.д.

Полный список программ с разбивкой по категориям:

```
kodomo:~$ wossname -filter | less
```

Полный список программ в алфавитном порядке:

```
kodomo:~$ wossname -filter -alphabetic | less
```

Entrez Direct (EDirect)

Единая поисковая система, которая объединяет многие базы данных NCBI.

- ▶ У каждой базы данных в системе Entrez есть имя, оно не всегда совпадает с именем на сайте.
Пример: nucscore – название в базы NCBI 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.

Функции Entrez

- ▶ В первую очередь – поиск по базам данных NCBI, то есть получение списка записей базы, удовлетворяющих поисковому запросу.
- ▶ Получение списка записей в той же или другой базе, которые каким-то образом связаны с указанными записями.
- ▶ Фильтрация списка записей согласно каким-либо критериям.
- ▶ Загрузка записей с возможностью выбора формата.

Все это вы можете делать на сайте NCBI.

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 kodoמו.

Bio.Entrez модуль Biopython для работы с EUtils.

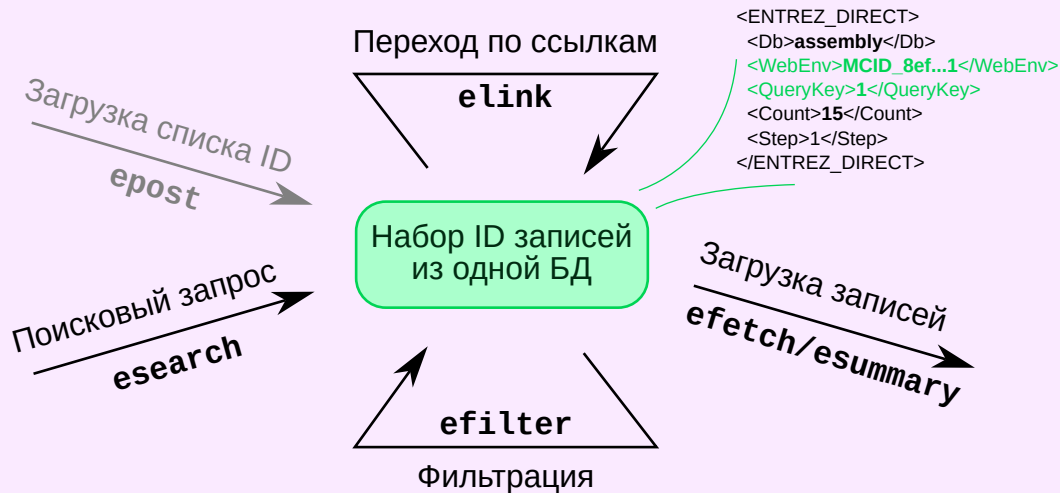
...

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 (не актуальна?)
- xtract** извлечение отдельных полей из выдачи в формате XML
- transmute** изменение формата выдачи и не только

EDirect: общая схема



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

EFilter

Фильтрация записей по дополнительным критериям:

Опция `-query` аналогична `esearch`:

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

31879785

9207024

7753630

7899082

8506140

ELink

Работа с перекрестными ссылками:

Переход по ссылкам на другие записи

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

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'  
... выдача пар белок-taxid
```

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>

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

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

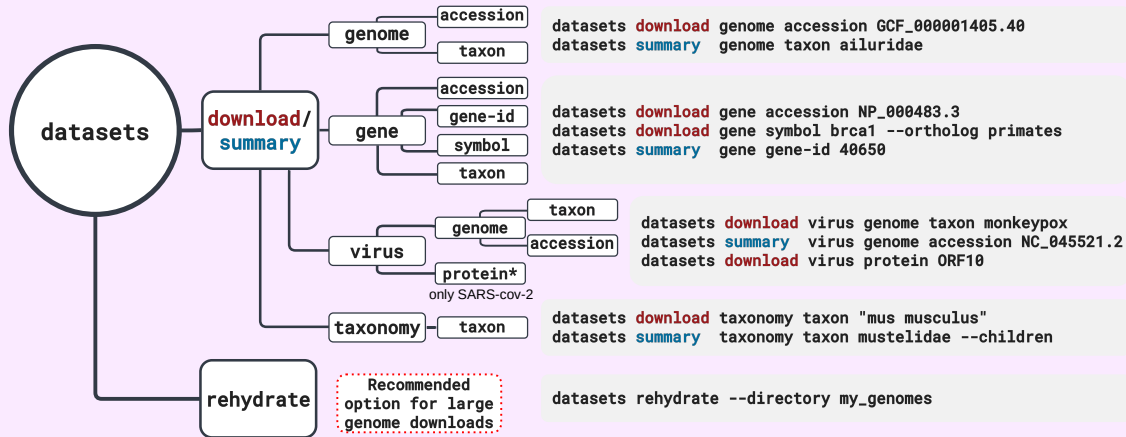
NCBI Datasets CLI

NCBI Datasets

Ребрендинг геномных данных в NCBI.

- ▶ БД Assembly переименовали в Datasets/Genome
- ▶ БД Genome переименовали в Datasets/Taxonomy и существенно доработали
- ▶ Улучшили веб-интерфейс обеих баз, добавили удобный поиск и навигацию
- ▶ Создали новое независимое API – консольные программы `datasets` и `dataformat`

datasets: примеры команд



https://www.ncbi.nlm.nih.gov/datasets/docs/v2/datasets_schema_taxonomy.svg

datasets: используемые форматы

JSON – умолчательный формат для `summary`, можно парсить только сторонними программами (например, `jq`) или самописными скриптами на Python (модуль `json`)

JSON Lines – формат выдачи `summary` с опцией `-as-json-lines` – набор строк, каждая является объектом JSON, можно перевести в таблицу TSV или XLSX с помощью `dataformat`

ZIP – команда `download` всегда загружает архив `.zip`, распаковывать с помощью `unzip`

Единственное предназначение программы `dataformat` – превращать конкретные JSON Lines, которые умеет выдавать `datasets summary`, в таблицы. Она знает конкретные поля и ломается, если объекты отличаются по структуре.