# EMBOSS

Иван Русинов

# EMBOSS

**E**uropean **M**olecular **B**iology **O**pen **S**oftware **S**uite

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

Описание всех имеющихся опций:
kodomo:~$ any-emboss-util -help -verbose

Подробное описание команды:
kodomo:~$ tfm any-emboss-util

Поиск программы по описанию:
kodomo:~$ wossname "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/ путаный официальный сайт

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

# Унифицированный адрес последовательности (USA)

**U**niform **S**equence **A**ddress

```
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:

```
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:

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

Слить STDOUT и STDERR и передать следующей команде:
kodomo:~$ seqret -help -verbose |& less

Убить STDERR (перенаправить в черную дыру):
kodomo:~$ seqret 'seqs.fasta' 'plain::stdout' 2> '/dev/null' | less

Отключить сообщения на уровне команды EMBOSS:
kodomo:~$ seqret -filter 'seqs.fasta' | less
или
kodomo:~$ 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 ☹