Отчет по молекулярной филогенетике № 2

Артем Васильев

17 февраля 2024 года

- Загрузка и настройка библиотек
  - R
  - Python
  - shell
- 1. Код ищет в PubMed статьи по интересному запросу и возвращает абстракты этих статей (N первых статей в списке) в простом текстовом формате
  - R
  - Python
  - shell
- 2. Ищет ID организма по названию в базе taxonomy
  - R
  - Python
  - shell
- 3. Запрашивает в базе нуклеотидных последовательностей по названию гена, после чего возвращает таблицу с UID (в XML это поле называется Id), accession number (в XML это поле называется Caption), длиной последовательности (Slen)
  - R
  - Python
  - shell
- 4. Дает в базу нуклеотидных или белковых последовательностей текстовый запрос, а затем возвращает последовательности в формате fasta, которые записывает в файл
  - R
  - Python
  - shell
- 5. Скачивает белок, соответствующий известному UID нуклеотида
  - R
  - Python
  - shell
- 6. Скачивает все последовательности из работы с PMID … (например, из первого задания) и пишет их в файл fasta.
  - R
  - Python
  - shell

Загрузка и настройка библиотек

R

```
if (!("reutils" %in% installed.packages()))
  install.packages("reutils")
library(reutils)
options(reutils.email = "artem_vasilev_01@list.ru")
```

Python

```
! mamba install -c conda-forge biopython
from Bio import Entrez
Entrez.email = "artem_vasilev_01@list.ru"
```

shell

```
mamba install -c bioconda -c conda-forge entrez-direct
```

1. Код ищет в PubMed статьи по интересному запросу и возвращает абстракты этих статей (N первых статей в списке) в простом текстовом формате

R

(esearch) ищет в PubMed статьи по указанному запросу

```
esearch(db = "pubmed", term = "nxf1")  # вывод UIDs на экран
```

```
## Object of class 'esearch'
## List of UIDs from the 'pubmed' database.
##   [1] "38165804" "37909783" "37679383" "37628773" "37583452" "37540751"
##   [7] "37143720" "37069509" "36920996" "36857431" "36658633" "36040180"
##  [13] "35751040" "35589130" "35581240" "35263598" "34959589" "34793452"
##  [19] "34769195" "34389711" "34096602" "34068524" "33883167" "33771898"
##  [25] "33681726" "33589748" "33547084" "33375634" "33191911" "33172997"
##  [31] "33165929" "33091126" "33054770" "32999004" "32932882" "32917881"
##  [37] "32504555" "32236527" "32169426" "31940815" "31918596" "31917363"
##  [43] "31859577" "31811630" "31525188" "31416967" "31413174" "31398345"
##  [49] "31384064" "31375530" "31263181" "31104896" "30858280" "30819645"
##  [55] "30387240" "30301920" "30290229" "30227889" "30218090" "30194269"
##  [61] "30082412" "30068640" "29916017" "29741478" "29552131" "29545601"
##  [67] "29102717" "29021253" "28990926" "28984244" "28840554" "28831067"
##  [73] "28677678" "28578407" "28515301" "28314893" "28296067" "27929060"
##  [79] "27903772" "27708137" "27679854" "27389771" "27114368" "27070420"
##  [85] "27060156" "27016737" "26944680" "26944675" "26621383" "26476453"
##  [91] "26459599" "26349247" "26343730" "26209800" "26158194" "26144233"
##  [97] "25835743" "25826302" "25802992" "25662211"
```

```
ms <- esearch(db = "pubmed", term = "nxf1")
```

(efetch) возвращает абстракты (rettype = “abstract”) статей в простом текстовом формате

```
# efetch(ms)  # вывод на экран содержимое объекта
abstr <- efetch(ms, rettype = "abstract")
# abstr  # посмотреть на объект
write(content(abstr), "nxf1_abstracts.txt")  # сохранение в файл
```

(esearch) запрашивает в базе нуклеотидных последовательностей все последовательности по запросу по названию гена для организма по названию вида

```
# esearch(db = "nucleotide", term = "nxf1")  # работает
# esearch(db = "nucleotide", term = "nxf1 AND human[orgn]")  # работает
# esearch(db = "nucleotide", term = "nxf1 AND drosophila[orgn]")  # работает
esearch(db = "nucleotide", term = "nxf1 AND Drosophila melanogaster[orgn]")
```

```
## Object of class 'esearch'
## List of UIDs from the 'nucleotide' database.
## [1] "1624698290" "1624698288" "669632474 " "281360686 " "667695275 "
## [6] "14456154  " "14456089  "
```

Python

Этот код позволяет искать статьи, в которых упоминается какой-то термин. После чего мы можем выбрать в аргументе rettype тип скачиваемой части статьи.

```
handle = Entrez.esearch(db="pubmed", term="crustacyanin")
record = Entrez.read(handle)
print(record)
mshandle = Entrez.efetch(
    db="pubmed", id=record["IdList"][0:3], rettype="abstract", retmode="text"
)
print(mshandle.read())
```

shell

```
$ esearch -email artem_vasilev_01@list.ru -db pubmed -query "nxf1"

$ esearch -email artem_vasilev_01@list.ru -db pubmed -query "nxf1 AND human[orgn]" |
efetch -mode text -format abstract

$ esearch -email artem_vasilev_01@list.ru -db nucleotide -query "crustacyanin AND Hom
arus americanus[orgn]" | esummary
```

2. Ищет ID организма по названию в базе taxonomy

R

```
esearch(db = "taxonomy", term = "Drosophila melanogaster")
```

```
## Object of class 'esearch'
## List of UIDs from the 'taxonomy' database.
## [1] "7227"
```

Python

Можно подавать запросы для поиска нуклеотидных последовательностей.

2 БД: nucleotide и nuccore (разница непонятна). Вроде как первая больше.

Сама NCBI создана вроде в 1982. Код в ней написан в 1995-2000, поэтому м.б. много костылей.

```
handle = Entrez.esearch(
    db="nucleotide", term="nxf1 AND Drosophila melanogaster[orgn]"
)  # db="genbank" Error!!
record = Entrez.read(handle)
Entrez.efetch(db="nucleotide", id=record["IdList"])
handle = Entrez.esearch(db="taxonomy", term="Drosophila melanogaster")
record = Entrez.read(handle)
print(record)
print(record["IdList"])
```

shell

```
$ esearch -email artem_vasilev_01@list.ru -db taxonomy -query "Drosophila melanogaste
r" | esummary | grep TaxId
```

3. Запрашивает в базе нуклеотидных последовательностей по названию гена, после чего возвращает таблицу с UID (в XML это поле называется Id), accession number (в XML это поле называется Caption), длиной последовательности (Slen)

R

```
nxf1_prot <- esearch(db = "protein", term = "nxf1 AND Drosophila melanogaster[orgn]")
su <- esummary(nxf1_prot)
cosu <- content(su, "parsed")
as.data.frame(cosu[,c("Id", "Caption", "Slen")])
```

```
##            Id      Caption Slen
## 1  1376014748   A0A0B4K7J2 2718
## 2    74948518       Q9VTL1  395
## 3    20978541       Q9VV73  841
## 4    20978539       Q9U1H9  672
## 5    20455240       Q9VIW3  596
## 6    18203549       Q9V3H8  133
## 7    13124798       Q24168  618
## 8  2282536378       Q9U3V9 2079
## 9    74948793       Q9VYX1  101
## 10 1624698291 NP_001356942  536
## 11 1624698289 NP_001356941  536
## 12   17933682    NP_524660  672
## 13 1622462254     QCD25237  536
## 14 1622462253     QCD25236  536
## 15   22832060     AAF47959  672
## 16 1727108961       6IHJ_D  135
## 17 1727108960       6IHJ_B  135
## 18 1727108959       6IHJ_C  191
## 19 1727108958       6IHJ_A  191
## 20 1700031391       6OPF_F   98
## 21 1700031390       6OPF_D   98
## 22 1700031389       6OPF_B   98
## 23 1700031388       6OPF_A   98
## 24 1700031180       6MRK_V  133
## 25 1700031179       6MRK_B  206
## 26 1700031178       6MRK_U  133
## 27 1700031177       6MRK_A  206
## 28   14456155     CAC41645  672
## 29   14456090     CAC41644  841
```

Python

Вариант с использованием белковой БД. В данном кусочке пример кода с записью некоторой таблички:

ID D белковой стр-ры лина последовательности

```
handle = Entrez.esearch(db="protein", term="nxf1 AND Drosophila melanogaster[orgn]")
record = Entrez.read(handle)
for rec in record["IdList"]:
    temphandle = Entrez.read(Entrez.esummary(db="protein", id=rec, retmode="text"))
    print(
        temphandle[0]["Id"]
        + "\t"
        + temphandle[0]["Caption"]
        + "\t"
        + str(int(temphandle[0]["Length"]))
    )  # +"\n")
```

shell

```
esearch -email artem_vasilev_01@list.ru -db protein -query "nxf1 AND Drosophila melan
ogaster[orgn]" | esummary -mode xml -format docsum | xtract -pattern DocumentSummary
-element Id Caption Slen
```

4. Дает в базу нуклеотидных или белковых последовательностей текстовый запрос, а затем возвращает последовательности в формате fasta, которые записывает в файл

R

```
s <- esearch(db = "protein", term = "nxf1 AND Drosophila melanogaster[orgn]")
f <- efetch(uid = s[1:10], db = "protein", rettype = "fasta", retmode = "text")
write(content(f), "Drmel_nxf1.fa")
fastaf <- readLines("Drmel_nxf1.fa")
head(fastaf)
```

```
## [1] ">sp|A0A0B4K7J2.1|RBP2_DROME RecName: Full=E3 SUMO-protein ligase RanBP2; AltN
ame: Full=358 kDa nucleoporin; AltName: Full=Nuclear pore complex protein Nup358"
## [2] "MFTTRKEVDAHVHKMLGKLQPGRERDIKGLAVARLYMKVQEYPKAIEYLNGYLRVRDDAVGHNMIATCYS"
## [3] "RLNPPDVTEALQHYQRSIQIDPRQSEVVIDACELLVKENNASITECARYWLDQANSLDLSGNKQVFNLRM"
## [4] "RVNLADSNGERDDTSGGDGEQNTLEILMYKELQARPQDVNIRIQLLRSYVEKMKIDQAFNYALKTELESK"
## [5] "NCTSQSNEWYEQIWMVLFKIEMAKDVKKNWRFWHFALHTLDRLVQLSLEGSGLADSSKQLFRLDQYLFKF"
## [6] "STSIERSGDAPQRDLHQACIDHFTGQLLLHAVTLIFKREVLANKNKWMSTLRSALPLLLLGYQVRPIDDS"
```

Python

Здесь показан пример скачивания последовательностей по ID-шникам в .fasta формате.

rettype: fasta, genbank - формат файла

retmode: text, xml - формат данных

```
Entrez.efetch(db="protein", id=record["IdList"], retmode="text", rettype="fasta").rea
d()
with open("Drmel_nxf1.fa", "w") as ouf:
    for rec in record["IdList"]:
        lne = Entrez.efetch(
            db="protein", id=rec, retmode="text", rettype="fasta"
        ).read()
        ouf.write(lne + "\n")
with open("Drmel_nxf1.fa", "r") as fastaf:
    snippet = [next(fastaf) for x in range(5)]
    print(snippet)
```

shell

```
$ esearch -email artem_vasilev_01@list.ru -db protein -query "nxf1 AND Drosophila mel
anogaster[orgn]" | efetch -format fasta -mode text > Dm_nxf1.fa

$ head Dm_nxf1.fa
```

5. Скачивает белок, соответствующий известному UID нуклеотида

R

```
lnk1 <- elink(uid = "2065188392", dbFrom = "nucleotide", dbTo = "protein")
efetch(lnk1, rettype = "fasta", retmode = "text")
```

```
## Object of class 'efetch'
## >XP_042223242.1 crustacyanin-A2 subunit-like [Homarus americanus]
## MGVWYEIQAQPNIFQSIKSCLASSYKRVKTEIHVLSEGLDSSGASTTTKSILKIVDPQNPAHMVTDFVPG
## VEPPFDIVDTDYKTFSCAHSCLSIVGIKTEFVFIYSRNRTLRSNSTQHCLSIFEVSIIGIISFYTNANNY
##
##
## ...
## EFetch query using the 'protein' database.
## Query url: 'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?=efe...'
## Retrieval type: 'fasta', retrieval mode: 'text'
```

Python

Случай, когда у нас есть нуклеотидный ID, а скачать мы хотим белковую последовательность.

```
lhandle = Entrez.elink(dbfrom="nucleotide", db="protein", id="2065188392")
lrecord = Entrez.read(lhandle)
prothandle = lrecord[0]["LinkSetDb"][0]["Link"][0]["Id"]
rrecord = Entrez.efetch(db="protein", id=prothandle, rettype="fasta", retmode="text")
with open("prot_from_nt.fasta", "w") as ouf:
    ouf.write(rrecord.read() + "\n")
```

shell

```
link -id 2065188392 -db nuccore -target protein | efetch -format fasta -mode text
```

6. Скачивает все последовательности из работы с PMID … (например, из первого задания) и

пишет их в файл fasta.

R

```
ms2 <- esearch(db = "pubmed", term = "nxf1")
lnk <- elink(ms2[3], dbFrom = "pubmed", dbTo = "nuccore")
# f2 <- efetch(lnk, rettype = "fasta", retmode = "text")  # данный вариант работает, но очень долго, поэтому я не стал сохранять
# write(content(f2), "all_nxf1.fa")
```

Python

Если в pubmed’е упомянуты последовательности из NCBI, то с помощью ID из PubMed можно скачать эти последовательности. Эта команда может улететь с ошибкой, если в статье на упомянуты никакие данные.

```
lhandle = Entrez.elink(dbfrom="pubmed", db="nucleotide", id="20558169")
lrecord = Entrez.read(lhandle)
ids = []
for el in lrecord[0]["LinkSetDb"][0]["Link"]:
    ids.append(el["Id"])
rrecord = Entrez.efetch(db="nucleotide", id=ids[:4], rettype="fasta", retmode="text")
with open("py_fasta_pmid.fasta", "w") as ouf:
    ouf.write(rrecord.read() + "\n")
```

shell

```
$ elink -db pubmed -target nucleotide -id 20558169 | efetch -format fasta -mode text
> lobster_msp.fasta
```