

Проект за курса

„Линукс и езици за програмиране  
в биоинформатиката“

Софийски университет  
„Св.Климент Охридски“



ФЕВРУАРИ 2021

---

Росена Иванова-Гявурова

Фак.№26488

БМИ

2020-2021

# Проект за RESTful приложение на език Python

## Задание:

Да се разработи RESTful приложение на Python (по желание може и друг език), което да оперира върху данни от Ensembl: Да притежава следните функционалности:

1. Да приема id на gene и да връща като резултат цялата секвенция на гена и всички екзони като отделни свойства.
2. Да приема id на gene и да връща като резултат секвенцията с пресметнато процентното GC съдържание и разменени аминокиселини аденин с тимин в конкретния пример.
3. Да приема ID и да връща секвенция в специфичен формат. Поддържаните формати трябва да са: fasta, multi-fasta, x-fasta.

## Описание на изпълнението:

API е интерфейс за програмиране на приложения, позволява комуникация между две приложения за извличане или изпращане на данните. REST API попада в категорията на заявката-отговор. Използва HTTP методи като GET, POST и PUT за извършване на API операции.

Пример за такава заявка е, когато потребителят изпраща GET метод на уеб услугата (в случая сървър на Ensembl), за да поиска или извлече определен ресурс или колекция от ресурси. След това сървърът изпраща обратно конкретния ресурс или колекция от ресурси обратно на потребителя, който го е поискал.

---

Приложението разработих с Python 3.9, на платформа Visual Studio, Jupiter Notebook , като инсталирах допълнителни библиотеки: Python Bio, requests, Flask (рамка, базирана на Python).

```
import requests, json
from flask import Flask
from Bio.SeqUtils import GC
```

Начало на заявката с адреса на сървъра, към който изпращам заявката

```
app = Flask(__name__)
server = "https://rest.ensembl.org"
```

В кода по-долу съм използвала информацията, подадена в документация на Ensembl, за заявка, която да има функция да извлича секвенция, формат Jason. Използва метода: GET.

```
# ID seq from Ensemb
def get_index(id)
ext = '/v1/sequence/<id>/'
r = requests.get(server + ext, headers={ "Content-Type" :
"application/json"})

if not r.ok:
    r.raise_for_status()
    sys.exit()

decoded = r.json()
return(repr(decoded))
```

Следващата функционалност е насочена към създаване на списък с екзони, за които като резултат да се връщат: секвенцията, старт екзон, краен екзон, id.

```
# ID exons
def get_exon(seq)
exons[]
for e in output["Exon"]:
    exons.append({"start": e["start"],
                  "end": e["end"],
                  "id": e["id"],
                  "seq": seq[e["start"]-output["start"]:e["end"]
-output["start"]]])
```

Тази функционалност е насочена към изчисляване на съдържанието на GS в секвенцията, отново в json формат, връща процентното отношение, както и цялата секвенция. Използвам Bio Python.

```
# GC content

ext = '/sequence/gene/id/?gc_content=true&swap=A:T'
r = requests.get(server + ext, headers={
"Content-Type" : "application/json"})
if(gcContent=="true"):
    result["gc_content"]=GC(seq)
return gc_content
```

Целта на функцията по-долу е да размени базите А с Т и Т с А- използвам for цикъл, които да обходи секвенцията и да направи размяната на базите. Връща променената секвенция.

```
# A-T
def get_reverseA(id)
gc="/sequence/gene/id/+id?gc_content=true&swap=A:T').upper
()
for reverseA in gc:
    if dna=="A":
        dnaSeq.write("T")
    elif dna=="T":
        dnaSeq.write("A")
    else :
        dnaSeq.write(dna)
dnaSeq.close()
return reversed
```

С тази функционалност целя връщане на секвенцията във FASTA формат.  
документация на Ensembl .

```
# Fasta format
def get_fastaA(id)
ext = '/v1/sequence/id/?content-type=fasta'
s = requests.get(server+ext, headers={ "Content-Type" :
"text/x-fasta"})

if not r.ok:
    r.raise_for_status()
    sys.exit()
return fastaA
```

Края на зявката.

```
if __name__ == '__main__':
    app.run(debug=True)
```

Запаметявам файла с име *my project.ipynb*