



# КУРСОВ ПРОЕКТ

*Линукс и езици за програмиране в  
биоинформатиката*

Тема „RESTfull приложение на Python“

*Петър Андреев, Ф№ 26271*

*Специалност: Био- и Медицинска Информатика*

## 1. Извеждане на секвенция на ген

Секвенцията на гена се извежда в текстов вид по следния начин:

```
5  geneID = "ENSG00000157764"
6
7  # provides the sequence in text format
8
9  server = "https://rest.ensembl.org"
10 ext = "/sequence/id/" + geneID + "?"
11
12 r = requests.get(server+ext, headers={ "Content-Type" : "text/plain"})
13
14 if not r.ok:
15     r.raise_for_status()
16     sys.exit()
17
18 print(r.text)
19
20 geneSequence = r.text
```

При задаване на "Content-Type" : "text/plain" е възможно извеждане само на секвенцията в x – fasta формат

## 2. Отпечатване с разменени A -> T и пресмятане на GC content

```
73 # calculates GC content
74
75 elements = Counter(geneSequence)
76
77 a = (((elements["G"] + elements["C"]) / (elements["G"] + elements["C"] + elements["T"] + elements["A"]))*100)
78
79 print("GC content of", geneID, "is", format(a, '.2f'), "%")
80
81 # modifies the sequence
82
83 print (geneSequence.replace('A','T'))
```

### 3. Събиране на идентификаторите на екзоните на гена (заедно с друга информация, включително transcript ID)

Включена expand опция включва в резултата ID на екзони и транскрипти

```
geneID = "ENSG00000157764"

def fetch_endpoint(server, request, content_type):

    r = requests.get(server+request, headers={ "Accept" : content_type})

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

    if content_type == 'application/json':
        return r.json()
    else:
        return r.text

server = "http://rest.ensembl.org/"
ext_gene = "lookup/id/" + geneID + "?expand=1"
con_json = "application/json"
get_gene = fetch_endpoint(server, ext_gene, con_json)
```

## 4. Извличане на ID на екзоните

Два вмъкнати for цикъла обхождат резултата от предходната заявка

```
# for each transcript get a list of the exon ids
for transcript in get_gene['Transcript']:
    print(">", transcript['id'])
    exons = []
    for exon in transcript['Exon']:
        exons.append(exon['id'])

# create a json dump of the exon ids
exon_json = json.dumps({ "ids" : exons })
sequences = fetch_endpoint_POST(server, ext_sequence, exon_json, con_seq)
```

## 5. Списъка с ID на екзоните се подава чрез post заявка и се извеждат секвенциите на екзоните в x – fasta формат

```
def fetch_endpoint_POST(server, request, data, content_type):

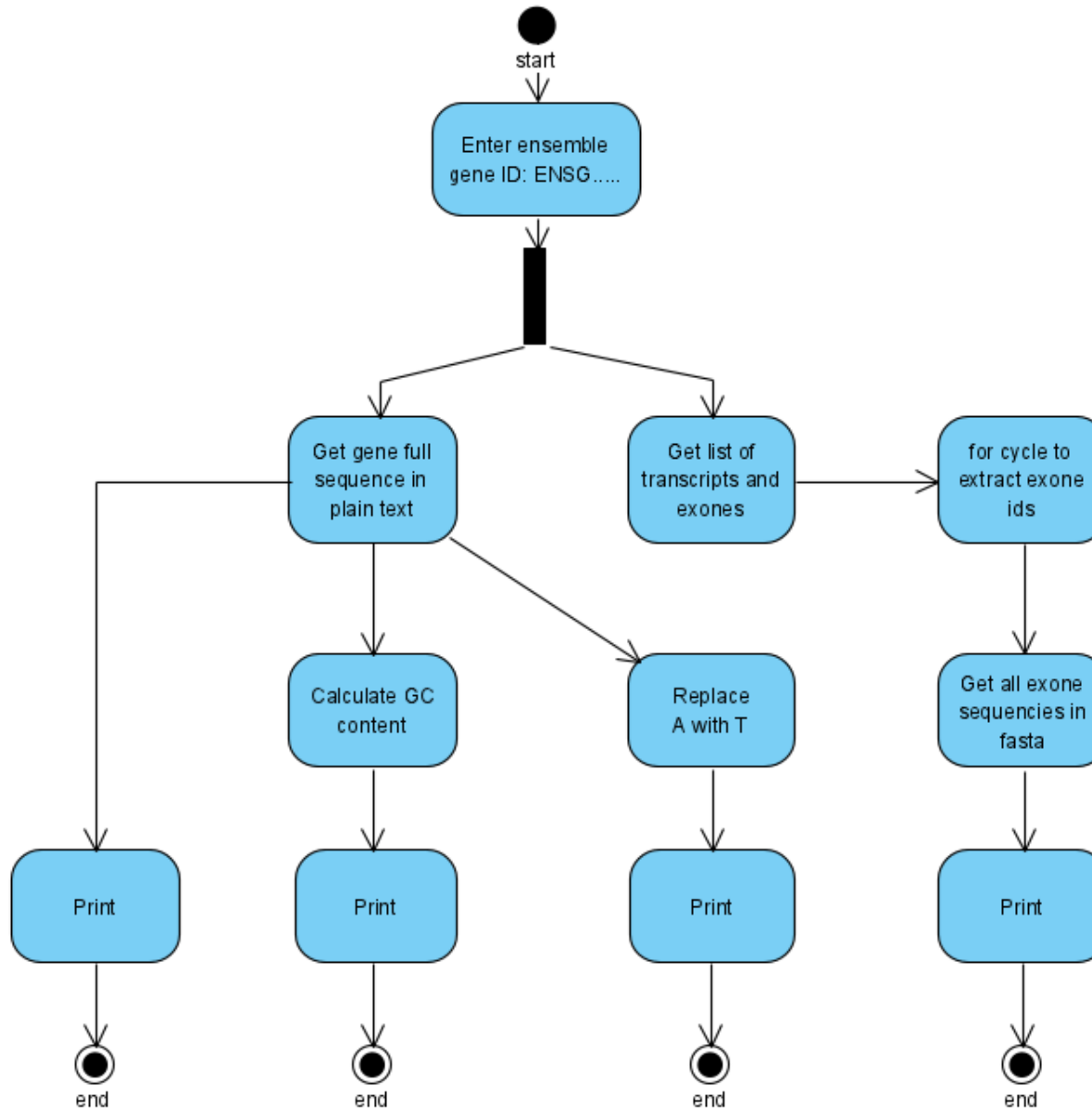
    r = requests.post(server+request,
                      headers={ "Accept" : content_type},
                      data=data )

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

    if content_type == 'application/json':
        return r.json()
    else:
        return r.text

server = "http://rest.ensembl.org/"
ext_gene = "lookup/id/" + geneID + "?expand=1"
con_json = "application/json"
ext_sequence = "/sequence/id/"
con_seq = "text/x-fasta"
get_gene = fetch_endpoint(server, ext_gene, con_json)
```

## 6. Схематично представяне на алгоритъма за работа на приложението



## 6. Примерен изход

```
CTTCCCCCAATCCCCTCAGGCTCGGCTGCGCCTCCGCCTCCCTCCCCCTCCCCGCCGACAG
CGGCCGCTCGGGCCCCGGCTCTCGGTTATAAGATGGCGGCGCTGAGCGGTGGCG.....
.....
> ENST00000496384
>ENSE00003828230.1 chromosome:GRCh38:7:140924566:140924810:-1
CCCCGCTCCTCCGCCTCCGCCTCCGCCTCCGCCTCCCCCAGCTCTCCGCCTCCCCC.....
>ENSE00003603715.1 chromosome:GRCh38:7:140850111:140850212:-1
GTGTGGAATATCAAACAAATGATTAAGTTGACCGCGGGCCGATGATTATCATAGT.....
.....
> ENST00000644969
>ENSE00001862791.3 chromosome:GRCh38:7:140924566:140924929:-1
CTTCCCCCAATCCCCTCAGGCTCGGCTGCGCCCGGGGCCGCGGGCCGGTACCTGAG.....
>ENSE00003603715.1 chromosome:GRCh38:7:140850111:140850212:-1
GTGTGGAATATCAAACAAATGATTAAGTTGACACAGGAACATATAGAGGCCCTATT.....
.....
GC content of ENSG00000157764 is 38.00 %
CTTCCCCCTTTCCCCTCTGGCTCGGCTGCGCCCGGGGCCGCGGGCCGGTTCCTGTGGTGGCCCTG
GCGCCCTCCGCCCGCGGCGCGCCCGGGGCCGCTCCTCCCCGCGCCCCCGCGCCCCC.....
.....
```



**Благодаря за вниманието!**

**Време за въпроси.**