Софийски университет „Св. Климент Охридски”,
Факултет по математика и информатика

Курсов проект

Предмет: **Линукс и езици за програмиране в биоинформатиката**

Зимен семестър, 2020/2021

Преподавател: доц. д-р Д. Василев
Преподавател: Илиян Михайлов

Изготвил: Моника Борисова,
ф.№ 26143, БМИ

Февруари, 2021

София

Цел на проекта:

Да се разработи RESTFul приложение на Python (по желание може и друг език), което да оперира върху данни от ensambl.

Задачи:

1. Да приема id на gene и да връща като резултат цялата секвенция на гена и всички екзони като отделни свойства.

   Извеждане на секвенцията не гена в json формат:

```python
# getting sequence and exons, Get request by default
@app.route('/v1/sequence/gene/id/<id>/')
def get_sequence(id):
    # in json format
    sequence = requests.get(server + gene_id + id,
                            headers={"Content-Type": "application/json"}).json()
    exons = requests.get(server + exon_ext + id + '?expand=1',
                         headers={"Content-Type": "application/json"}).json()

    # from the dictionary sequence, getting the value behind key seq
    seq = sequence["seq"]
```

Seq е String, който съдържа стойността на ключа seq в речника sequence.
За да вземем транскриптите и секвенциите в резултата, append-ваме речници, и резултата ни става списък, който се състой от малки речници с дадените стойности и ключове.

```python
app = Flask(__name__)

server = "https://rest.ensembl.org"
gene_id = "/sequence/id/"
exon_ext = "/lookup/id/"

# Transcripts and Exon function
def get_gene(transcripts, sequence):
    result = list()

    for transcript in transcripts:
        for exon in transcript['Exon']:
            result.append(
                {"start": exon["start"],
                 "end": exon["end"],
                 "id": exon["id"],
                 "seq": sequence[exon["start"]:exon["end"]]
                 })

    return result
```

Output:


```
{"exons":[{"end":140924810,"id":"ENSE00003828230","seq":"","start":140924566},{"end":140850212,"id":"ENSE00003603715","seq":"","start":140850111},{"end":140834872,"id":"ENSE00001035295","seq":"","start":1408
{"end":140808995,"id":"ENSE00001034889","seq":"","start":140808892},{"end":140808062,"id":"ENSE00003603742","seq":"","start":140807960},{"end":140801560,"id":"ENSE00003687908","seq":"","start":140801412},
{"end":140800481,"id":"ENSE00003487759","seq":"","start":140800362},{"end":140794467,"id":"ENSE00003569635","seq":"","start":140794308},{"end":140787584,"id":"ENSE00003680515","seq":"","start":140787548},
{"end":140783157,"id":"ENSE00003685923","seq":"","start":140783021},{"end":140781693,"id":"ENSE00003559218","seq":"","start":140781576},{"end":140778075,"id":"ENSE00003521664","seq":"","start":140777991},
{"end":140777088,"id":"ENSE00003527888","seq":"","start":140776912},{"end":140754233,"id":"ENSE00003649805","seq":"","start":140754187},{"end":140753393,"id":"ENSE00003485507","seq":"","start":140753275},
{"end":140749418,"id":"ENSE00003587655","seq":"","start":140749287},{"end":140739946,"id":"ENSE00003551608","seq":"","start":140739812},{"end":140734770,"id":"ENSE00001907699","seq":"","start":140734617},
{"end":140726516,"id":"ENSE00001847804","seq":"","start":140719327},{"end":140924929,"id":"ENSE00001862791","seq":"","start":140924566},{"end":140850212,"id":"ENSE00003603715","seq":"","start":140850111},
{"end":140834872,"id":"ENSE00001035295","seq":"","start":140834609},{"end":140808995,"id":"ENSE00001034889","seq":"","start":140808892},{"end":140808062,"id":"ENSE00003603742","seq":"","start":140807960},
{"end":140801560,"id":"ENSE00003687908","seq":"","start":140801412},{"end":140800481,"id":"ENSE00003487759","seq":"","start":140800362},{"end":140794467,"id":"ENSE00003569635","seq":"","start":140794308},
{"end":140787584,"id":"ENSE00003680515","seq":"","start":140787548},{"end":140785808,"id":"ENSE00003829283","seq":"","start":140785689},{"end":140783157,"id":"ENSE00003685923","seq":"","start":140783021},
{"end":140781693,"id":"ENSE00003559218","seq":"","start":140781576},{"end":140778075,"id":"ENSE00003521664","seq":"","start":140777991},{"end":140777088,"id":"ENSE00003527888","seq":"","start":140776912},
{"end":140754233,"id":"ENSE00003649805","seq":"","start":140754187},{"end":140753393,"id":"ENSE00003485507","seq":"","start":140753275},{"end":140749418,"id":"ENSE00003587655","seq":"","start":140749287},
{"end":140739946,"id":"ENSE00003551608","seq":"","start":140739812},{"end":140734770,"id":"ENSE00001907699","seq":"","start":140734617},{"end":140726516,"id":"ENSE00003819130","seq":"","start":140719337},
{"end":140824398,"id":"ENSE00003817176","seq":"","start":140823533},{"end":140808995,"id":"ENSE00003815229","seq":"","start":140808892},{"end":140808316,"id":"ENSE00003822896","seq":"","start":140808237},
{"end":140808062,"id":"ENSE00003624175","seq":"","start":140807960},{"end":140801560,"id":"ENSE00003633946","seq":"","start":140801412},{"end":140800481,"id":"ENSE00003558819","seq":"","start":140800362},
{"end":140794415,"id":"ENSE00003831096","seq":"","start":140794308},{"end":140787584,"id":"ENSE00003669794","seq":"","start":140787548},{"end":140783157,"id":"ENSE00003569958","seq":"","start":140783021},
{"end":140781693,"id":"ENSE00003512445","seq":"","start":140781576},{"end":140778075,"id":"ENSE00003545181","seq":"","start":140777991},{"end":140777088,"id":"ENSE00003651618","seq":"","start":140776912},
{"end":140754233,"id":"ENSE00003667578","seq":"","start":140754187},{"end":140753393,"id":"ENSE00003570443","seq":"","start":140753275},{"end":140749418,"id":"ENSE00003667236","seq":"","start":140749287},
{"end":140739946,"id":"ENSE00003664390","seq":"","start":140739812},{"end":140726516,"id":"ENSE00003818766","seq":"","start":140725145},{"end":140808338,"id":"ENSE00003829826","seq":"","start":140808237},
{"end":140808062,"id":"ENSE00003624175","seq":"","start":140807960},{"end":140801560,"id":"ENSE00003633946","seq":"","start":140801412},{"end":140800481,"id":"ENSE00003558819","seq":"","start":140800362},
{"end":140794415,"id":"ENSE00003831096","seq":"","start":140794308},{"end":140787584,"id":"ENSE00003669794","seq":"","start":140787548},{"end":140783157,"id":"ENSE00003569958","seq":"","start":140783021},
{"end":140781693,"id":"ENSE00003512445","seq":"","start":140781576},{"end":140779928,"id":"ENSE00003815659","seq":"","start":140779807},{"end":140778075,"id":"ENSE00003545181","seq":"","start":140777991},
{"end":140777088,"id":"ENSE00003651618","seq":"","start":140776912},{"end":140749418,"id":"ENSE00003667236","seq":"","start":140749287},{"end":140739946,"id":"ENSE00003664390","seq":"","start":140739812},
{"end":140734770,"id":"ENSE00003816893","seq":"","start":140734617},{"end":140726516,"id":"ENSE00003815513","seq":"","start":140725325},{"end":140924928,"id":"ENSE00003824367","seq":"","start":140924566},
{"end":140850212,"id":"ENSE00003603715","seq":"","start":140850111},{"end":140834872,"id":"ENSE00001035295","seq":"","start":140834609},{"end":140808995,"id":"ENSE00001034889","seq":"","start":140808892},
{"end":140808062,"id":"ENSE00003603742","seq":"","start":140807960},{"end":140801560,"id":"ENSE00003687908","seq":"","start":140801412},{"end":140800481,"id":"ENSE00003487759","seq":"","start":140800362},
{"end":140794467,"id":"ENSE00003569635","seq":"","start":140794308},{"end":140787584,"id":"ENSE00003680515","seq":"","start":140787548},{"end":140783157,"id":"ENSE00003685923","seq":"","start":140783021},
{"end":140781693,"id":"ENSE00003559218","seq":"","start":140781576},{"end":140778075,"id":"ENSE00003521664","seq":"","start":140777991},{"end":140777088,"id":"ENSE00003527888","seq":"","start":140776912},
{"end":140754233,"id":"ENSE00003649805","seq":"","start":140754187},{"end":140753393,"id":"ENSE00003485507","seq":"","start":140753275},{"end":140749418,"id":"ENSE00003587655","seq":"","start":140749287},
{"end":140739946,"id":"ENSE00003551608","seq":"","start":140739812},{"end":140734770,"id":"ENSE00003830897","seq":"","start":140730665},{"end":140924712,"id":"ENSE00003830695","seq":"","start":140924566},
```


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.

```python
gcContent = request.args.get("gc_content")
swap = request.args.get("swap")
contentType = request.args.get("content-type")

# if we have ?gc_content=true&swap=A:T
if (gcContent == "true" or swap is not None):
    result = {"seq": seq}

    # check if gcContent is true and return calculated G and C content. The percentage is calculated against the full length. (0-100)
    if (gcContent == "true"):
        result["gc_content"] = GC(seq)

    # check if swap is given and split by : for (A:T), swap the amino acids A with T
    if (swap is not None):
        swap = swap.split(":")
        result["swap_sequence"] = swapFunction(seq, swap[0], swap[1])

    return result

# fasta, x-fasta, multi-fasta
elif contentType:
    return sequenceByContentType(id, sequence, contentType)

# default whole sequence and all the exons
else:
    returnExons = get_gene(exons.get('Transcript'), str(sequence['seq']))
    return {'seq': sequence, 'exons': returnExons}
```

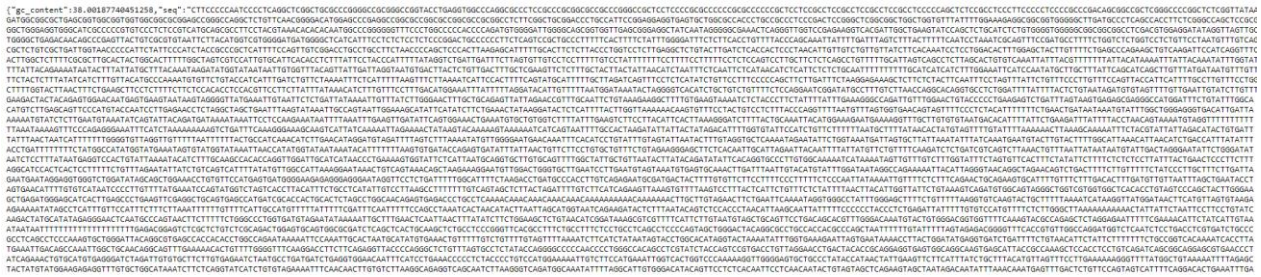
От URL искаме да вземем това, което стой след ?(args), в случая: "gc_content", "swap", "content-type", чрез getter.
След което проверяваме дали gcContent и swap не са None, конкретно искаме gcContent да ни е “true”, за да върнем секвенцията, с два if-а пресмятаме процентното GC съдържание, пресмята се с готов метод от модула Biopython и разменяме базите Аденин с Тимин със следната функция:

Swap функция:

```python
# swap function for A and T
def swapFunction(seq, n1, n2):
    seq = seq.replace(n1, 'S')
    seq = seq.replace(n2, n1)
    seq = seq.replace('S', n2)

    return seq
```

Output:


{"gc_content":38.00187740451258,"seq":"CTTCCCCCAATCCCCTCAGGCTCGGCTGCGCCCGGGGCCGCGGGCCGGTACCTGAGGTGGCCCAGGCGCCCTCCGCCCGCGGCGCCGCCCGGGCCGCTCCTCCCCGCGCCCCCCGCGCCCCCCGCTCCTCCGCCTCCGCCTCCGCCTCCGCCTCCCCCAGCTCTCCGCCTCCCTTCCCCTCCCCCGCCCGACAGCGGCCGCTCGGGCCCCGGCTCTCGGTTATAA
GATGGCGGCGCTGAGCGGTGGCGGTGGTGGCGGCGCGGAGCCGGGCCAGGCTCTGTTCAACGGGGACATGGAGCCCGAGGCCGGCGCCGGCGCCGCGGCCTCTTCGGCTGCGGACCCTGCCATTCCGAGGAGGAGGTGAGTGCTGGCGCCACCCTGCCGCCCTCCCGACTCCGGGCTCGGCGGCTGGCTGGTGTTTATTTTGGAAAGAGGCGGCGGTGGGGGCTTGATGCCCTCAGCCACCTTCTCGGGCCAGCTCGCC
GGCTGGGAGGTGGGCATCGCCCCCGTGTCCCTCTCCGTCATGCAGCGCCTTCCTACGTAAACACACACAATGGCCCGGGGGTTTCCCTGGCCCCCACCCCAGATGTGGGGATTGGGGCGACGCGGTGGTTGAGCGGGAGGCTATCAATAGGGGGCGAAACTCAGGGTTGGTCCGAGAAGGTCACGATTGGCTGAAGTATCCAGCTCTGCATCTCTGTGGGGTGGGGGCGGCGGCGGCCTCGACGTGGAGGATATAGGTTAGTTGC
TGGGGCTGAGACAACAGCCCGAGTTACTGTCGCGTGTAATTCTTACATGGTCGTGGGGATGATGGGGCTCATCATTCCCTCTCTCCTCTCCCGGACTGCCCCCCCTCTCAGTCCGCTGCCCTTTTCACTTTTCTATTTGGGGATTTCTCTTCACCTGTTTACCCAGCAAATTATTTGATTTAGTCTTACTTTTCAATCCTAAATCGACAGTTTCCGATGCCTTTCTGGTCTCTGGTCCTCTGTTCCTAATGTTTGTCAG
CGCTCTGTCGCTGATTGGTAACCCCCATTCTATTCCCATCTACCGCCCGCTCATTTTCCAGTTGTCGGACCTGCCTGCCTTCTAACCCCAGCTCCCACTTAAGAGCATTTTTGCACTTCTCTTACCCTGGTCCTCTTGAGGCTCTGTACTTGATCTCACCACTCCCTAACATTGTTGTCTGTTGTTATCTTCACAAATCCTCCTGGACACTTTGGAGCTACTTGTTTTCTGAGCCCAGAAGCTGTCAAGATTCCATCAGGTTTC
ACTTGGCTCTTTTCGCGCTTGCACTACTGGCACTTTTTGGCTAGTCGTCCATTGTGCATTCACACCCTCTTTATTCCTACCCATTTTATAGGTCTGATTGATTTCTTAGTGTTGTCCTCCTTTTTGTCCTATTTTTTTCCTTTTCCTTTTCCTCTCCAGCTCTGTTTCTCCAGCCTGTTTTGCATTAGTCAGCCTCTTAGCACTGTGTCAAATTATTTACGTTTTATTACATAAAATTTATTACAATAATTTTGGTAT
TTTATTACAGAAAATAATACTTTATTATGCTTTACAAATAAGATATGGTATAATAATTGTGGTTTACAGTTATTGATTAGGTAATGTGACTACTCTGTTGACTTTGCTCGAAGTTCTCTTTGCTACTTACTATTAACATCTAATTTCTCAATTCTCATAAACATCTCATTCTCTCTGCAATTTTTTTTTGGATCATCATCTTTGGAAATTCATCCAATATGCTTGCTTTATTGAATAATGTTTGTT
TTCTACTCTTTATATCATCTTTGTTACATGCCCAAAATGTGTTCTGTACCATCATTTGATCTGTTCTAAAATTTCTCATTTTTAAGTTTCTTAAAATCATTCCACTTTTCAGTATGCATTTTTGCTTAGATCAGTTTCCTCTCATATCTGTTCCTTTCCCCCAGCTTCTTGATTTCTAAGGAGAAAGCTCTTCTCTACTTCAATTTCCTAGTTTATTCTGTTTCCCTTGTTTCCTGC
CTTTTGGTACTTAACTTTCTGAAGCTTCCTCTTTTCTTCTCCCACACCTCCACGTTCCTCTTATTTATAAACATCTTTGTTTCCTTTGACATGGAAATTTATTTTAGGATACATTGTTTTTAATGGATAAATACTAGGGGTCACATCTGCTGTCTGTTTTCTCCAGGAATCGGATATGCCTTTGTCTTAACCAGGCACAGGTGCCTCTGGATTTTATTTTACTCTGTAATAGATGTGTAGTTTTGTTGAATTGTATCTTGTTT
GAAGACTACTACAGAGTGGAACAATGAGTGAAGTAATAAGTAGGGGTTATGAAATTGTAATTCTCTGATTATAAAATTGTTTATCTTGGGAACTTTGCTGCAGAGTTATTAGAACCGTTTGCAATTCTGTAAAGAAGGCTTTTGTGAAGTAAAATCTCTACCCTTCTATTTTATTTGAAAGGGCCAGATTGTTTGGAACTGTACCCCCTGAAGAGTCTGATTTAGTAAGTGAGAGCGAGGGCCATGGATTTCTGTATTTGGCA
CATGTCTTGAGCAGTTCCCATGTACCAATCCTTGAGAACCTCTAGGCTAGCTGAATTTAAGTATAAATTGCCAGTAATTGGAAAGCATATTCATATCTCTGAAACTATAAGGATACTCTCATTTTACTTGGTTAAAAAACAAGTGTTTCCTACTGTCCTCTTTACCCAGGTTTTAATGTTTAGTGGTGAACAGTAGTTTCCCTCTACATTTTTTTCTGAACTGATAATAAATGTATTTGGCTGGGAGGGTGACATTGATTA
AAAAATGTATCTCTTGAATGTAAATATCAGTATTACAGATGATAAAATAAATTCCTCCAAGAAATAATTTTAAATTTGAAGTTGATATTCAGTGGAAACTGAAATGTGCTGTGGTCTTTTATTTGAAGTCTTCCTTACATTCACTTAAGGGATCTTTTACTGCAAATTACATGGAAAGAATGAAAAGGTTTGCTTGTGTGTAATGACACATTTTATTCTGAAGATTTATTTTACCTAACAGTAAAATGTAGGTTTTTTTTTT
TTAAATAAAAGTTTCCCAGAGGGAAATTTCATCTAAAAAAAAAGTCTGATTTCAAAGGGAAAGCAAGTCATTATCAAAAATTAGAAAACTATAAGTACAAAAAGTAAAAAATCATCAGTAATTTTGCCACTAAGATATTATTACTATAGACATTTGGTGTATTCCATCTGTTCTTTTTAATGCTTTATAAACACTATGTGAGTTTTGTATTTTAAAAAACTTAAAGCAAAAATTTCTACGTATTATTAGACATACTGTGATT
TATTTAACTAATCATTTTTTTGGGGTGTTAGGTTGTTTTAATTTTTTACTGCCATCAAACATCTTGAACATAGGATGTAGATTTTAGTCTTTAAAATATGTTGGGGAATGAACAAATTTCACATCCTGTATTTGTAGTATTAATACTTTGTAGGTGCTCAAAATAGAATATTCTGGTAAATGATTAGTGCTTATTAAATATTTATCAAATGAATGTACTTGTACTTTTGGCATTAAACATTAACATCTGACCATTTATATTT
ACCTGATTTTTTTTCTATGGCCATATGGTATGAAATAGTGTATGGATAAATTAACCATATGGTATAATAAATACATTTTTTTAAGTGTGATACCAGAGTGATATTTATTAACTGTTCTTCCTGTGCTGTTTCTGTAGAAGGGAGCTTCTCACAATTGCATTAGAATTACAATTTTATTATGTTCTGTTTTCAAGATCTCTGATCGTCAGTCTTAAACTGTTTAATTATAATAATGTATTGACTAGGGAATATTCTGGGATAT
AATCTCCTTTATAATGAGGTCCACTGTATTAAAATACATCTTTGCAAGCCACACCAGGTTGGATTGCATCATAACCCTGAAAAGTGGTATTCTCATTAATGCAGGTGCTTGTGCAGTTTTGGCTATTGCTGTTAATACTTATACAGATATATTCACAGGTGCCCTTGTGGCAAAAATCATAAAATAGTTGTTTGTCTTTGGTATTTCTAGTGTTCACTTTCTATATTCTTTTCTCTCTCCTTATTTACTGAACTCCCTTCTTT
AGGCATCCACTCACTCCTTTTTCTGTTTAGAATATTATCTGTCAGTCATTTTATATGTTGGCCATTAAAGGAATAAACTGTCAGTAAACAGCTAAGAAAGGAATGTTGGACTGGGTGCTTGAATCCTTGAATGTAGTAAATGTGAGTGCAAACTTGATTTAATTGTACATGTATTTGGATAATAGGCCAGAAAAATTACATTAGGGTAACAGGCTAGAACAGTCTGACTTTTCTTGTTTTCTCATCCCTTGCTTTCTTGATTA
GAATGAATAGGAGGTGGGTCTGGATATAGCAGCTGGAAACCTGTGTTCCATGAGTGATGGGGAAGAGAGGGGAGGGAATAGGTTCCTCTGATTTTTGGCATTTCTAAGACCTGATGCCCACCTTGTCAGAGAATGCGATGACTACTTTGTGTTCTTCCTTTCCCTTTTCTCCCAATTATAAAATTGTTTTCTCTTTCAGAACTGCAGAAGTGCATTTTGTTCTTTGACACTTGATGTTGTTAATTTAGCTGAATACCT
AGTGAACATTTTGTGTCATAATCCCCTTGTTTTATGAAATCCAGTATGGTCTAGTCACCTTACATTTCTGCCTCATATTGTCCTTAAGCCTTTTTTTGTCAGTAGCTCTTACTAGATTTTGTCTTCATCAGAAGTTAAAGTGTTTTAAGTCCTTTACTCATTCTGTTTCTCTATTTTAACTTACATTGGTTATTCTGTAAAGTCAGATGTGGCAGTAGGGCTGGTCGTGGTGGCTCACACCTGTAGTCCCAGCTACTTGGGAA
GCTGAGATGGGAGCATCACTTGAGCCCTGAAGTTCGAGGCTGCAGTGAGCCATGATCGCACCACTGCACTCTAGCCTGGCAACAGAGTGAGACCCTGCCTCAAAACAAACAAACAAACAAACAAAAAAAAACAAAAAACTTGCTTGTAGAAACTTCTGAATTCAAAATAGGTGGCCTATTTGGGAGCTTTTCTGTTTTTAAGGTGTCAAGTACTGCTTTTAAATCATAAGGTTATGGATAACTTCATGTTAGTGTAAGA
AGAAAAATATAGCCTCATTTGTTCCATTTCTTTCTTAAATTTTTTGTTTTCATTGCCATGTTTTTATTTTCGATTTCAATTTTTCCAGCCTAAATCACTAACATACTTAATTAGCATGGTAATCAGAAGATACTCTTTAATACAGTCTCCACCCTAACATTAAGCAATTATTTTCCCCCCTACCCTCTGAGATTATTTTGTGTCCATGTTTTCTCTTGGGCTTAAAAAAAAAAACTATTATTCTAATTCCTTCCTGTATC
AAGACTATGCATATAGAGGGAACTCAATGCCCAGTAACTTCTTTTTCTGGGCCCTGGTGATGTAGAATATAAAAATTGCTTTGAACTCAATTAACTTTATATCTTCTGGAAGCTCTGTAACATCGGATAAAGCGTCGTTTTCATTCTTGTAATGTAGCTGCAGTTCCTGACAGCACGGTTTGGGACAAATGTACTGTGGGACGGTGGTTTTCAAAGTACGCCAGAGCTCTAGGAGAATTTTTCGAAAACATTCTATCATTGTAA
ATAATAATTTTTTTTTTTTTTTTTTTTTGAGACGGAGTCTCGCTCTGTCTCGCAGACTGGAGTGCAGTGGCGCGATCTCAGCTCACTGCAAGCTCTGCCTCCCGGGTTCACGCCTTCTGCCTTTCTCCTGCCTCAGCCTCCCCAGTAGCTGGGACTACAGGCGCCCACCACGCCCAGCTAATTTTTTGTATTTTTAGTAGAGACGGGGTTTCACCGTGTTGGCCAGGATGGTCTCAATCTCCTGACCTCGTGATCTGCCC
GCCTCAGCCTCCCCAAAGTGCTGGGATTACAGGCGTGAGCCACCACACCTGGCCGGAATAAAAATTCCAAATTGCACTAATGCATATGTGAAACTGTTTTGTCTGTTTTTGTAGTTTTAAAATCTTCATCTATAATAGTACCTGGCACATAGGTACTAAAATATTTGGTGAAAGAATTAGTGAATAAAACCTTACTGGATATGAGGTGATCTGATTTTCTGTAACATTCTATTCTTTTTTCTGCCGGTCACAAAATCACCTTA
TGAAATTGACAGCCAAATTGGCTGCAACAGGCAGTTTGAAAAACACTGTTTTGGGGTTTCAAGGACCTTCTTCAGAGGTTACCCAGGGCCTCTGTTTAGTGCCTCTATACCAGGGGCCCCCAACCCCTGGGCCACAGCCTCGTATCTACCAGTCCGTGACCTGTTAGGAACCTGACTACACCGCAGGAGGTGAGTGGCAGGCAAGTGAGCATTACCGCCAAAGCTCCACCTCCTGTCAGATCAGCGGCAGGAGCCGTGAACCCT
ATCAGAAACTGTGCATGTGAGGGATCTAGATTGTGTGCTTCTTGTGAGAATCTAATGCCTGATGATCTGAGGTGGAACAATTTCATCCTGAAACCCCCTCTACCCCTGTCCATGGAAAAATTGTCTTCCATGAAATTGGTCACTGGTCCCAAAAAGGTTGGGGAGTGCTGCCCTATACCATAACTATTGAAGTTCTTCATTTATCTGCTTTACATGTTAGTTTCCTTGAAAAAAGGGTTTTATGGCTGTAAAAATTTTAGAGC
TACTATGTATGGAAGAGAGGTTTGTGCTGGCATAAATCTTCTCAGGTATCATCTGTGTAGAAAATTTCAACAACTTGTGTCTTAAGGCAGAGGTCAGCAATCTTAAGGGTCAGATGGCAAATATTTTAGGCATTGTGGGACATACAGTTCCTCTCACAATTCCTCAACAATACTGTAGTAGCTCAGAAGTAGCTAATAGACAATATTTAAACAAATGAGTTTGACTCTGTTCCAGTAGTCATTTTCAGGACACTGAAATTTGA

3. Да приема ID и да връща секвенция в специфичен формат. Поддържаните формати трябва да са: fasta, multi-fasta, x-fasta.

   Ако искаме определен тип (fasta, x-fasta, multi-fasta) проверяваме дали се съдържа в URL чрез else if, показан по-горе и използваме функция, която приема id, sequence и contentType и връщаме конкретния тип. В случай че нямаме никое от 3те неща, връщаме секвенцията и екзоните.

Функция за различните формати с 2 функции за форматите - x-fasta, multi-fasta:

```
# different formats in query
def sequenceByContentType(id, sequence, contentType):
    if contentType == "fasta":
        return {"seq": sequence, "id": ">" + id}
    elif contentType == "x-fasta":
        return xFasta(id, sequence)
    elif contentType == "multi-fasta":
        return multiFasta(id, sequence)


def xFasta(id, sequence):
    description = ">" + sequence["id"]
    return description + "\n" + sequence["seq"]


def multiFasta(id, sequence):
    result = ""
    for seq in sequence:
        desc = ">" + sequence["id"] + "." + \
            str(sequence["version"]) + " " + sequence["desc"]
        result += desc + "\n" + seq

    return result
```

Fasta:

{"id":">ENSG00000157764","seq":
{"desc":"chromosome:GRCh38:7:140719327:140924929:-1","id":"ENSG00000157764","molecule":"dna","query":"ENSG00000157764","seq":"CTTCCCCCAATCCCCTCAGGCTCGGCTGCGCCCGGGGCCGCGGGCCGGTACCTGAGGTGGCCCAGGCGCCCTCCGCCCGCGGCGCCGCCCGGGCCGCTCCTCCCCGCGCCCCCCGCGCCCCCCGCTCCTCCGCCTCCG
CCTCCGCCTCCGCCTCCCCCAGCTCTCCGCCTCCCTTCCCCCTCCCCGCCCGACAGCGGCCGCTCGGGCCCCGGCTCTCGGTTATAAGATGGCGGCGCTGAGCGGTGGCGGTGGTGGCGGCGCGGAGCCGGGCCAGGCTCTGTTCAACGGGGACATGGAGCCCGAGGCCGGCGCCGGCGCCGGCGCCGCGGCCTCTTCGGCTGCGGACCCTGCCATTCCGGAGGAGGTGAGTGCTGGCGCCACCCTGCCGCCCTCCCGACTCCG
GGCTCGGCGGCTGGCTGGTGTTTATTTTGGAAAGAGGCGGCGGTGGGGGCTTGATGCCCTCAGCCACCTTCTCGGGCCAGCTCCGCGGGCTGGGAGGTGGGCATCGCCCCCGTGTCCCTCTCCGTCATGCAGCGCCTTCCTACGTAAACACACACAATGGCCCGGGGGGTTTCCCTGGCCCCCACCCCAGATGTGGGGATTGGGGCAGCGGTGGTTGAGCGGGAGGCTATCAATAGGGGGCGAAACTCAGGGTTGGTCCGAGAA
GGTCACGATTGGCTGAAGTATCCAGCTCTGCATCTCTGTGGGGTGGGGGCGGCGGCGGCCTCGACGTGGAGGATATAGGTTAGTTGCTGGGGCTGAGACAACAGCCCGAGTTACTGTCGCGTGTAATTCTTACATGGTCGTGGGGATGATGGGGCTCATCATTTCCTCTCTCCTCTCCCGGACTGCCCCCCTTCTCAGTCCGCTGCCCTTTTTCACTTTTCTATTTGGGGATTTCTCTTCACCTGTTTTACCCAGCAAATTATT
TTGATTTAGTCTTTACTTTTTCAATCCTAAATCGCAGTTTCCGATGCCTTTTCTGGTCTCTGGTCCTCTGTTCCTAATGTTTGTCAGCGCTCTGTCGCTGATTGGTAACCCCCATTCTATTCCCATCTACCGCCCGCTCATTTTCCAGTTGTCGGACCTGCCTGCCTTCTAACCCCAGCTCCCACTTAAGAGCATTTTTGCACTTCTCTTACCCTGGTCCTCTTGAGGCTCTGTACTTGATCTCACCACTCCCTAACATTGTTG

X-Fasta:

>ENSG00000157764
CTTCCCCCAATCCCCTCAGGCTCGGCTGCGCCCGGGGCCGCGGGCCGGTACCTGAGGTGGCCCAGGCGCCCTCCGCCCGCGGCGCCGCCCGGGCCGC

Multi-fasta:

>ENSG00000157764.14 chromosome:GRCh38:7:140719327:140924929:-1 query>ENSG00000157764.14 chromosome:GRCh38:7:140719327:140924929:-1 desc>ENSG00000157764.14 chromosome:GRCh38:7:140719327:140924929:-1 molecule>ENSG00000157764.14 chromosome:GRCh38:7:140719327:140924929:-1 version>ENSG00000157764.14 chromosome:GRCh38:7:140719327:140924929:-1 seq>ENSG00000157764.14 chromosome:GRCh38:7:140719327:140924929:-1 id

Тестваме в браузъра след като стартираме програмата:

With debug=True, Flask server will auto-reload when there are code changes

```
if __name__ == '__main__':
    app.run(host='127.0.0.1', port=5000, debug=False)
```

```
# Test:
# http://127.0.0.1:5000/v1/sequence/gene/id/ENSG00000157764/
# http://127.0.0.1:5000/v1/sequence/gene/id/ENSG00000157764/?gc_content=true&swap=A:T
# http://127.0.0.1:5000/v1/sequence/gene/id/ENSG00000157764/?content-type=fasta
# http://127.0.0.1:5000/v1/sequence/gene/id/ENSG00000157764/?content-type=x-fasta
# http://127.0.0.1:5000/v1/sequence/gene/id/ENSG00000157764/?content-type=multi-fasta
```

Terminal:

```
 * Serving Flask app "project" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [05/Feb/2021 12:48:55] "GET /v1/sequence/gene/id/ENSG00000157764/ HTTP/1.1" 200 -
127.0.0.1 - - [05/Feb/2021 12:48:57] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [05/Feb/2021 12:50:24] "GET /v1/sequence/gene/id/ENSG00000157764/?gc_content=true&swap=A:T HTTP/1.1" 200 -
127.0.0.1 - - [05/Feb/2021 12:51:18] "GET /v1/sequence/gene/id/ENSG00000157764/?content-type=fasta HTTP/1.1" 200 -
127.0.0.1 - - [05/Feb/2021 12:52:17] "GET /v1/sequence/gene/id/ENSG00000157764/?content-type=x-fasta HTTP/1.1" 200 -
127.0.0.1 - - [05/Feb/2021 12:52:53] "GET /v1/sequence/gene/id/ENSG00000157764/?content-type=multi-fasta HTTP/1.1" 200 -
```