

## Практикум 12

### Задание 1

С помощью программы на Python [1] были определены количества старт-кодонов для каждой из бактерий. Результаты работы программы представлены ниже в виде таблиц.

Таблица 1. Количество старт-кодонов для *Escherichia coli* str. K-12 substr. MG1655, Candidatus “*Gracilibacteria bacterium* 28\_42\_T64” и *Mycoplasma pneumoniae* M29.

| Старт-кодон            | Количество встреч |                                       |                      |
|------------------------|-------------------|---------------------------------------|----------------------|
|                        | <i>E. coli</i>    | Candidatus<br>“ <i>G. bacterium</i> ” | <i>M. pneumoniae</i> |
| AAA                    | 0                 | 0                                     | 1                    |
| ACA                    | 0                 | 1                                     | 0                    |
| ACT                    | 0                 | 0                                     | 1                    |
| ATA                    | 0                 | 0                                     | 1                    |
| ATC                    | 0                 | 0                                     | 1                    |
| ATG                    | 3890              | 1129                                  | 629                  |
| ATT                    | 4                 | 0                                     | 8                    |
| CAA                    | 0                 | 0                                     | 2                    |
| CTC                    | 0                 | 0                                     | 2                    |
| CTG                    | 2                 | 0                                     | 1                    |
| GAA                    | 0                 | 0                                     | 1                    |
| GGA                    | 0                 | 0                                     | 1                    |
| GTG                    | 338               | 41                                    | 60                   |
| GTT                    | 0                 | 0                                     | 1                    |
| TCA                    | 0                 | 1                                     | 0                    |
| TCT                    | 0                 | 0                                     | 1                    |
| TTA                    | 0                 | 0                                     | 3                    |
| TTC                    | 1                 | 1                                     | 0                    |
| TTG                    | 80                | 23                                    | 53                   |
| Всего<br>старт-кодонов | 6                 | 6                                     | 16                   |

Как видно из представленных данных, у всех трёх видов бактерий функцию старт-кодона может выполнять не только стандартный старт-кодон “ATG”. При этом

самыми частыми альтернативными старт-кодонами являются “GTG” и “TTG”, которые, видимо, лучше всего из альтернативных старт-кодонов взаимодействуют с антикодоном fMet-тРНК “CAT”.

Использование альтернативного старт-кодона может уменьшать уровень не только трансляции (за счёт меньшей эффективности взаимодействия fMet-тРНК с кодоном), но и транскрипции (как это было показано при замене стандартного старт-кодона на альтернативный) (Indu S. Panicker et al, 2015). У последовательностей со слабым старт-кодом может быть сильная последовательность Шайна-Дальгарно, компенсирующая низкий уровень трансляции.

Таким образом, последовательность старт-кодон может использоваться живыми организмам для регуляции экспрессии генов. Кроме того, альтернативный старт-кодон может возникать в результате мутаций в псевдогенах, на которые естественный отбор действует с меньшей силой (или наоборот приводить к снижению экспрессии гена и его становлению псевдогеном).

Также стоит отметить разнообразие старт-кодонов у *M. pneumoniae*, которое, возможно, связано с паразитическим образом жизни этой бактерии, при адаптации к которому у нее значительно изменился транскриптом. В данном случае замены старт-кодонов, влияющие на экспрессию генов, как раз могли быть причиной перехода к паразитическому образу жизни.

## Задание 2

По результатам работы программы [2], у *E. coli* имеется четыре кодирующие последовательности, содержащие стоп кодон не в конце. Одна из них является псевдогеном (произошла от гена, кодирующего белок IS911A regulator fragment), а три другие кодируют альфа субъединицы N и O форматдегидрогеназы и H форматдегидрогеназу. Оказалось, что все они содержат стоп-кодон “TGA”. Ниже приведены названия и описания найденных кодирующих последовательностей.

```
lcl|U00096.3_cds_b4587_250      [gene=insN]      [locus_tag=b4587]      [db_xref=ASAP:ABE-0285253,ECOCYC:G6130]
[protein=IS911A regulator fragment] [pseudo=true] [location=join(270278..270540,271764..272190)] [gbkey=CD]
```

```
lcl|U00096.3_cds_AAD13438.1_1459 [gene=fdnG]      [locus_tag=b1474]      [db_xref=UniProtKB/Swiss-Prot:P24183]
[protein=formate dehydrogenase N subunit alpha] [transl_except=(pos:586..588,aa:Sec)] [protein_id=AAD13438.1]
[location=1547401..1550448] [gbkey=CDS]
```

```
lcl|U00096.3_cds_AAD13456.1_3824 [gene=fdoG]      [locus_tag=b3894]      [db_xref=UniProtKB/Swiss-Prot:P32176]
[protein=formate dehydrogenase O subunit alpha] [transl_except=(pos:586..588,aa:Sec)] [protein_id=AAD13456.1]
[location=complement(4082772..4085822)] [gbkey=CDS]
```

```
lcl|U00096.3_cds_AAD13462.1_3997 [gene=fdhF]      [locus_tag=b4079]      [db_xref=UniProtKB/Swiss-Prot:P07658]
[protein=formate dehydrogenase H] [transl_except=(pos:418..420,aa:Sec)] [protein_id=AAD13462.1]
[location=complement(4297219..4299366)] [gbkey=CDS]
```

У псевдогена стоп-кодон мог возникнуть в результате мутации и привести к неспособности синтезировать функциональный белок с гена, или наоборот возникнуть уже после того, как ген-предшественник стал псевдогеном, и сохраниться из-за наименьшего действия отбора на псевдогены.

У генов, кодирующих форматдегидрогеназы, кодон “UGA” кодирует селеноцистеин, о чем можно предположить из-за описания "transl\_except = (...aa:Sec)". Селеноцистеин встраивается в последовательность белка благодаря наличию

специальной последовательности после “UGA”, маркирующей кодирующую функцию этого кодона (F. Zinoni et al, 1987).

### Задание 3

В следующих таблицах представлены количества стоп-кодонов для каждой из бактерий, определенные с помощью программы на Python [3]. Кроме перечисленных, у всех бактерий существуют также другие последние в кодирующей последовательности кодоны, которые встречались от 1 до 4 раз в геноме и реальная функция которых не проверялась (они определялись как последние три нуклеотида кодирующей последовательности).

Таблица 2. Количество стоп-кодонов для *Escherichia coli str. K-12 substr. MG1655*, Candidatus “*Gracilibacteria bacterium 28\_42\_T64*” и *Mycoplasma pneumoniae M29*.

| Стоп-кодон | Количество встреч |                                    |                      |
|------------|-------------------|------------------------------------|----------------------|
|            | <i>E. coli</i>    | Candidatus “ <i>G. bacterium</i> ” | <i>M. pneumoniae</i> |
| TGA        | 1246              | 1                                  | 0                    |
| TAA        | 2761              | 1000                               | 533                  |
| TAG        | 306               | 188                                | 221                  |

Как видно из представленных данных, у Candidatus “*G. bacterium*” и *M. pneumoniae* кодон “TGA” практически не используется в качестве стоп-кодона. Стоп-кодоны “TGA” и “TAA” взаимодействуют с RF2, который отсутствует у этих видов бактерий. Вместо этого стоп-кодон “TGA” кодирует аминокислоты глицин и триптофан у Candidatus “*G. bacterium*” и *M. pneumoniae* соответственно (<https://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/index.cgi?chapter=cgencodes#SG25>, <https://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/index.cgi?chapter=cgencodes#SG4>).

### Задание 4

При помощи программы [4] были рассчитаны частоты кодирующих лейцин кодонов.

Таблица 3. Частоты кодирующих лейцин кодонов для *Escherichia coli str. K-12 substr. MG1655*, Candidatus “*Gracilibacteria bacterium 28\_42\_T64*” и *Mycoplasma pneumoniae M29*, округленные до второго знака после запятой. Жирным шрифтом отмечены кодоны с наибольшей частотой.

| Стоп-кодон | Частота среди кодирующих лейцин кодонов |                                    |                      |
|------------|-----------------------------------------|------------------------------------|----------------------|
|            | <i>E. coli</i>                          | Candidatus “ <i>G. bacterium</i> ” | <i>M. pneumoniae</i> |
| CTA        | 0,04                                    | 0,09                               | 0,11                 |
| CTC        | 0,10                                    | 0,11                               | 0,12                 |

|     |             |             |             |
|-----|-------------|-------------|-------------|
| CTG | <b>0,50</b> | 0,05        | 0,09        |
| CTT | 0,10        | 0,26        | 0,10        |
| TTA | 0,13        | <b>0,41</b> | <b>0,38</b> |
| TTG | 0,13        | 0,09        | 0,21        |

У *E. coli* самый частый кодон лейцина - “CTG”, у *Candidatus “G. bacterium”* и *M. pneumoniae* - “TTA”. Эффективность связывания кодона со своим антикодоном аминоксил-тРНК определяет скорость синтеза белков, а лейцин является одной из самых часто встречающихся аминокислот (примерно 10% для каждого из рассматриваемых видов), поэтому в значительной степени определяет эту скорость. Тогда можно предположить, что самый частый кодон лейцина является наиболее эффективным из возможных, то есть распознающая его аминоксил-тРНК присутствует в клетках в наибольшей концентрации.

Кроме концентрации соответствующей аминоксил-тРНК, на частоты кодонов может также влиять общий GC-состав генома бактерии. Для первой, второй и третьей бактерий он равен 0,51, 0,29 и 0,40 соответственно (Y. Matsumura et al, 2018, Ch. M. K. Sieber et al, 2019, R. Himmelreich et al, 1996). GC-состав кодонов лейцинов также различается для данных бактерий, у *E. coli* он составляет 0,49, а у “*G. bacterium*” и *M. pneumoniae* - 0,25 и 0,28. Видимо, для бактерий с повышенным GC-составом характерно использование кодонов с большим содержанием гуанина и цитозина, а для бактерий с пониженным - с большим содержанием аденина и тимина. Этим объясняется разница частот кодонов и среди кодонов одного вида, и среди разных видов.

## Задание 5

При помощи веб-сервиса webskew (<https://genskew.csb.univie.ac.at/webskew>) при шаге в 1000 нуклеотидов и окне в 100000 нуклеотидов был построен график зависимости GC-skew и кумулятивного GC-skew от координаты нуклеотида (на вход программе была передана полная последовательность генома <https://www.ncbi.nlm.nih.gov/nuccore/U00096.3?report=fasta>).

Gen-skew plot for sequence: sequence.fasta, with stepsize: 1000 and window size: 100000

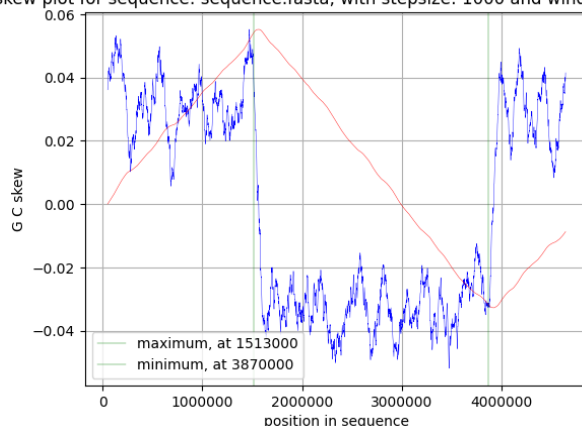


Рисунок 1. График зависимости GC-skew и кумулятивного GC-skew от координаты нуклеотида для *E. coli*

Минимум значения GC-skew соответствует ориджину репликации, максимум - точке терминции репликации. Таким образом, oriC, скорее всего, располагается у 3870000 нуклеотида, а ter - у 1513000. По описанию в genbank oriC находится в 3925744..3925975. Таким образом, данные о местоположении oriC, полученные по значению кумулятивного GC-skew, не сильно отличаются от реальных.

[1]

```
name_fn = input()
with open(name_fn, mode='r') as fn:
    array = fn.readlines()
seqs = list()
for i in range(len(array)-1):
    if '>' in array[i]:
        temp = array[i]
        j = 1
        while (len(array)-i-j > 0) and not('>' in array[i+j]):
            temp += array[i+j]
            j += 1
        seqs.append(temp)
answers = dict()
for seq in seqs:
    temp = seq.split("\n")
    if temp[1][:3] in answers:
        answers[temp[1][:3]] += 1
    else:
        answers[temp[1][:3]] = 1
sorted_answers = sorted(answers.items(), key=lambda k: k[0])
with open('out.txt', mode='w') as out:
    for tpl in sorted_answers:
        out.write(tpl[0]+' '+str(tpl[1])+'\n')
```

[2]

```
name_fn = input()
with open(name_fn, mode='r') as fn:
    array = fn.readlines()
seqs = list()
for i in range(len(array)-1):
    if '>' in array[i]:
        temp = array[i]
        j = 1
        while (len(array)-i-j > 0) and not('>' in array[i+j]):
            temp += array[i+j]
            j += 1
        seqs.append(temp)
stopcods = ['TAA', 'TAG', 'TGA']
ans = []
for seq in seqs:
    temp = seq.split("\n")
    s = str()
    for i in range(1, len(temp)):
        s += temp[i]
    for i in range(0, len(s)-3, 3):
        for stopcod in stopcods:
            if s[i:i+3] == stopcod:
                ans.append(temp[0][1::])
if len(ans) > 1:
    anss = []
```

```

anss.append(ans[0])
for i in range(1, len(ans)):
    if ans[i] != ans[i-1]:
        anss.append(ans[i])
with open('out.txt', mode='w') as out:
    if len(ans) > 1:
        for an in anss:
            out.write(an+'\n')
    else:
        for an in ans:
            out.write(an+'\n')

```

[3]

```

name_fn = input()
with open(name_fn, mode='r') as fn:
    array = fn.readlines()
seqs = list()
for i in range(len(array)-1):
    if '>' in array[i]:
        temp = array[i]
        j = 1
        while (len(array)-i-j > 0) and not('>' in array[i+j]):
            temp += array[i+j]
            j += 1
        seqs.append(temp)
answers = {'TGA':0,
           'TAA':0,
           'TAG':0}
for seq in seqs:
    temp = seq.split('\n')
    s = str()
    for i in range(1, len(temp)):
        s += temp[i]
    if s[-3:] in answers:
        answers[s[-3:]] += 1
    else:
        answers[s[-3:]] = 1
with open('out.txt', mode='w') as out:
    for key, value in answers.items():
        out.write("{} {}{}\n".format(key, value))

```

[4]

```

name_fn = input()
with open(name_fn, mode='r') as fn:
    array = fn.readlines()
seqs = list()
for i in range(len(array)-1):
    if '>' in array[i]:
        temp = array[i]
        j = 1
        while (len(array)-i-j > 0) and not('>' in array[i+j]):
            temp += array[i+j]
            j += 1
        seqs.append(temp)
leucine = {'CTA': 0,
           'CTC': 0,
           'CTG': 0,
           'CTT': 0,

```

```

    'TTA': 0,
    'TTG': 0}
summall = 0
for seq in seqs:
    temp = seq.split("\n")
    s = str()
    for i in range(1, len(temp)):
        s += temp[i]
    summall += len(s)/3
    for i in range(0, len(s)-3, 3):
        if s[i:i+3] in leucine:
            leucine[s[i:i+3]] += 1
summ = 0
for value in leucine.values():
    summ += value
print(summ/summall)
with open('out.txt', mode='w') as out:
    for key, value in leucine.items():
        out.write("{} {} {:.02f}".format(key, value/summ)+"\n")

```

Christian M. K. Sieber, Blair G. Paul, Cindy J. Castelle, Ping Hu, Susannah G. Tringe, David L. Valentine, Gary L. Andersen, and Jillian F. Banfield: Unusual Metabolism and Hypervariation in the Genome of a Gracilibacterium (BD1-5) from an Oil-Degrading Community Published online 2019 Nov 12. *Genome Announc.* doi: 10.1128/mBio.02128-19 PMCID: PMC6851277 PMID: 31719174

F Zinoni, A Birkmann, W Leinfelder, and A Böck: Cotranslational insertion of selenocysteine into formate dehydrogenase from *Escherichia coli* directed by a UGA codon. May 1, 1987 84 (10). *Proc Natl Acad Sci U S A* 3156-3160 <https://doi.org/10.1073/pnas.84.10.3156>

Indu S. Panicker, Glenn F. Browning, Philip F. Markham: The Effect of an Alternate Start Codon on Heterologous Expression of a PhoA Fusion Protein in *Mycoplasma gallisepticum* Published: May 26, 2015. *PloS One* <https://doi.org/10.1371/journal.pone.0127911>

R. Himmelreich, H Hilbert, H Plagens, E Pirkel, B C Li, and R Herrmann: Complete sequence analysis of the genome of the bacterium *Mycoplasma pneumoniae*. 1996 Nov 15; 24(22): 4420–4449. *Nucleic Acids Res.* doi: 10.1093/nar/24.22.4420 PMCID: PMC146264 PMID: 8948633

Yasufumi Matsumura, Masaki Yamamoto, Satoshi Nakano, and Miki Nagao: Complete Genome Sequence of *Escherichia coli* ME8067, an Azide-Resistant Laboratory Strain Used for Conjugation Experiments Published online 2018 Jun 21. *Genome Announc.* doi: 10.1128/genomeA.00515-18 PMCID: PMC6013617 PMID: 29930039