Вердикт: WrongAnswer

Контест: 72730: Финал агротех 9

Участник: Дарья

Задача: 1. Задание 1. "Фермер и плесень" (10 баллов)

Баллы: 0

Компилятор: No compiler

IP пользователя: 85.143.3.38

GlobalID: 10000067-5eba-40d3-49c9-38db56684447

Тесты Логи Исходный код

¶ 0xFF

```
2
```

Вердикт: OK

Контест: 72730: Финал агротех 9

Участник: Дарья

Задача: 2. Задание 2. "Нанотехнологии покоряют мир" (5 баллов)

Баллы: 5

Компилятор: No compiler

IP пользователя: 85.143.3.38

GlobalID: 10000067-5ec6-f916-aaac-432a89618dba

Тесты Логи Исходный код

¶ 0xFF

```
11
```

Вердикт: WrongAnswer

Контест: 72730: Финал агротех 9

Участник: Дарья

Задача: 3. Задание 3. "Волшебная Чага" (10 баллов)

Баллы: 0

Компилятор: No compiler

IP пользователя: 85.143.3.38

GlobalID: 10000067-5ee1-18d0-e285-d3aad96ba5fd

Тесты Логи Исходный код

¶ 0xFF

```
140
```

Вердикт: OK

Контест: 72730: Финал агротех 9

Участник: Дарья

Задача: 4. Задание 4. "Юный натуралист" (10 баллов)

Баллы: 10

Компилятор: No compiler

IP пользователя: 85.143.3.38

GlobalID: 10000067-5ee7-3bf5-9240-ebb4dc20a644

Тесты Логи Исходный код

¶ 0xFF

```
6
```

Вердикт: WrongAnswer

Контест: 72730: Финал агротех 9

Участник: Дарья

Задача: 5. Задание 5. "Амбар" (15 баллов)

Баллы: 0

Компилятор: No compiler

IP пользователя: 85.143.3.38

GlobalID: 10000067-5ed1-827a-9ff9-fb1ccd4582b5

Тесты Логи Исходный код

¶ 0xFF

```
2268
```

Вердикт: WrongAnswer

Контест: 72730: Финал агротех 9

Участник: Дарья

Задача: 6. Задание 6. "Петя" (10 баллов)

Баллы: 7

Компилятор: Python 3.12.3

IP пользователя: 85.143.3.38

GlobalID: 10000067-5ecc-6e8b-5ee3-613ba479b3ab

Тесты Логи Исходный код

0xFF

```
n = int(input())
count = 0
s = input()
s2 = s.split()
smax = s[0]
smin = s[0]
for i in range(n):
    if s2[i] > smax:
        smax = s2[i]
        count += 1
    elif s2[i] < smin:
        smin = s2[i]
        count += 1
if n % 2 == 0:
    print(count - 1)
else:
    print(count)
```

Вердикт: OK

Контест: 72730: Финал агротех 9

Участник: Дарья

Задача: 8. Задание 8. "Юный предприниматель" (10 баллов)

Баллы: 10

Компилятор: Python 3.12.3

IP пользователя: 85.143.3.38

GlobalID: 10000067-5ec6-9f70-9b17-aaba6a98ce07

Тесты Логи Исходный код

¶ 0xFF

```python
x = int(input())
p = int(input())
y = int(input())
count = 0
while x < y:
    x = x * ((p / 100)+1)
    x = round(x)
    count += 1
print(count)
```

Вердикт: RuntimeError

Контест: 72730: Финал агротех 9

Участник: Дарья

Задача: 9. Задание 9. "Грядка" (15 баллов)

Баллы: 0

Компилятор: Python 3.12.3

IP пользователя: 85.143.3.38

GlobalID: 10000067-5ee4-22ef-9adb-9d45b898c728

Тесты Логи Исходный код

¶ 0xFF

```
n = int(input())
k = int(input())
r = k
count = 1
for i in range(k, n+ 1):
    print(k)
    k+= 1
for i in range(0, r- 1):
    print(count)
    count += 1
```

Вердикт: TimeLimitExceeded

Контест: 72730: Финал агротех 9

Участник: Дарья

Задача: 10. Задание 10. "Рис или кукуруза????" (5 баллов)

Баллы: 3

Компилятор: Python 3.12.3

IP пользователя: 85.143.3.38

GlobalID: 10000067-5ec9-36d6-1ebe-5c54c1e62042

Тесты Логи Исходный код

¶ 0xFF

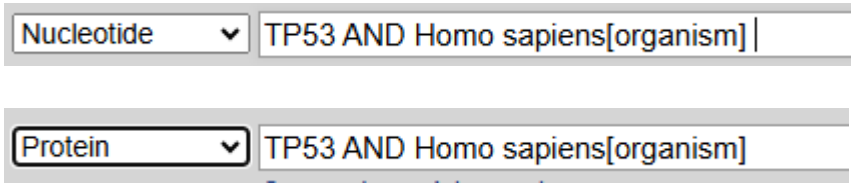
```
n = int(input())
count = 0
for i in range(2, n - 1):
    if n % i == 0:
        count += 1
if count >= 1:
    print("rice")
else:
    print("corn")
```

Панина Дарья

Ген TP53

Задание 1.1

Найдем белковую и нуклеотидную последовательности данного белка из базы данных NCBI.



Нуклеотидная последовательность:

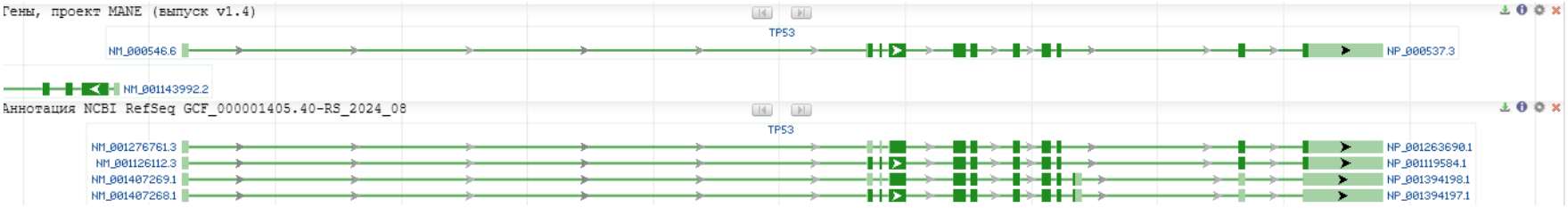
TGGGTTGATTCCACACCCCCGCCC
GGCACCCGCGTCCGCGCCGTGGC
CATCTACAAGCAGTCACAGCACA
TGACGGAGGTTGTGAGGCGCTGC
CCCCACCATGAGCGCTGCTCAGAT
AGCGAT

Белковая последовательность:

WVDSTPPPGTRVRAVAIYKQSQH
MTEVVRRCPHHERCSDSD

Задание 1.2

Ген TP53 имеет 13 экзонов и 10 интронов. Номер хромосомы: 13. Он имеется у эукариотов, млекопитающих, позвоночных и приматов.



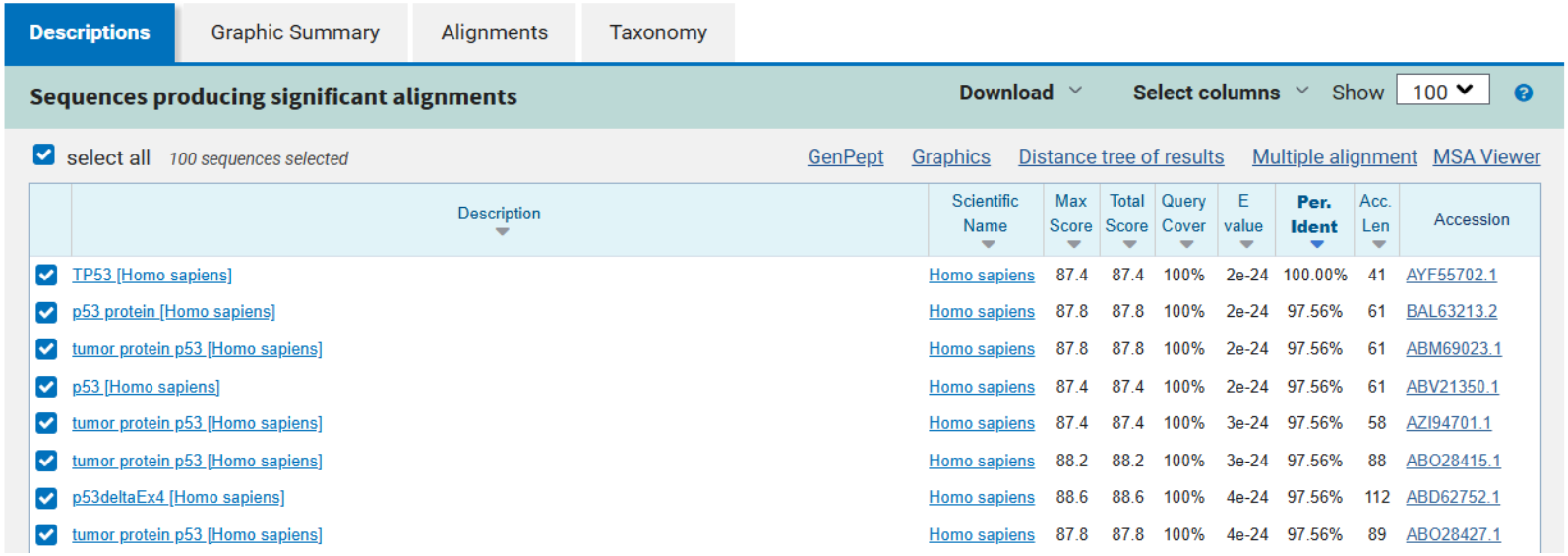
Задание 2

Этот ген кодирует белок-супрессор опухолей. Кодируемый белок реагирует на различные клеточные стрессы, регулируя экспрессию целевых генов, тем самым вызывая остановку клеточного цикла, апоптоз, старение, восстановление ДНК или изменения метаболизма. Мутации в этом гене связаны с различными видами рака у человека.

Взаимодействует с Envelope surface glycoprotein gp120, Nef, Pr55(Gag), Rev, Tat.

Задание 3

Был произведён поиск паралогов с помощью различных матриц (BLOSUM62, BLOSUM80, BLOSUM45, BLOSUM50, BLOSUM90) и алгоритма BLAST. Обнаружить их не удалось, так как процент индетичности везде был слишком большой.



Descriptions | Graphic Summary | Alignments | Taxonomy

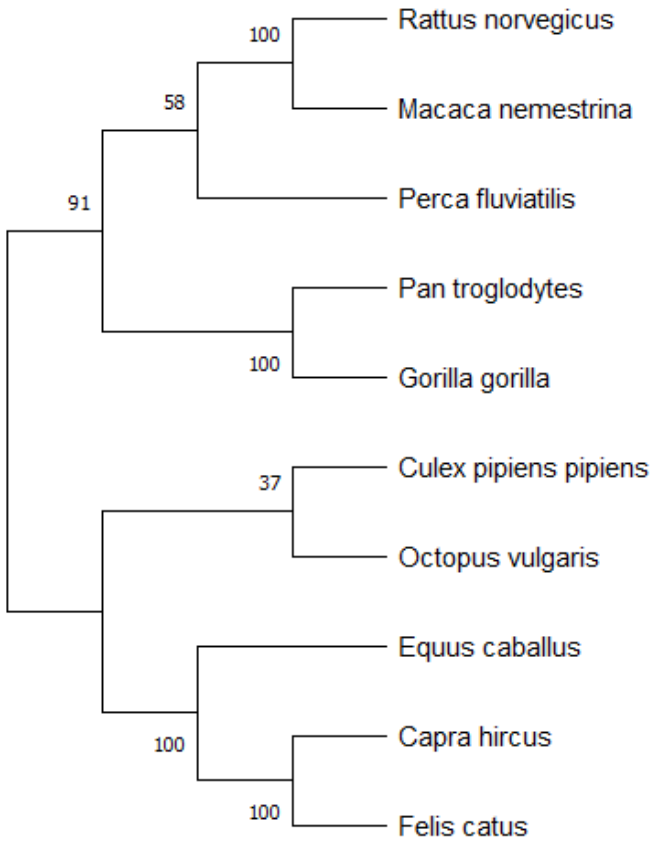
Sequences producing significant alignments — Download — Select columns — Show 100

select all *100 sequences selected* — GenPept | Graphics | Distance tree of results | Multiple alignment | MSA Viewer

Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
TP53 [Homo sapiens]	Homo sapiens	87.4	87.4	100%	2e-24	100.00%	41	AYF55702.1
p53 protein [Homo sapiens]	Homo sapiens	87.8	87.8	100%	2e-24	97.56%	61	BAL63213.2
tumor protein p53 [Homo sapiens]	Homo sapiens	87.8	87.8	100%	2e-24	97.56%	61	ABM69023.1
p53 [Homo sapiens]	Homo sapiens	87.4	87.4	100%	2e-24	97.56%	61	ABV21350.1
tumor protein p53 [Homo sapiens]	Homo sapiens	87.4	87.4	100%	3e-24	97.56%	58	AZI94701.1
tumor protein p53 [Homo sapiens]	Homo sapiens	88.2	88.2	100%	3e-24	97.56%	88	ABO28415.1
p53deltaEx4 [Homo sapiens]	Homo sapiens	88.6	88.6	100%	4e-24	97.56%	112	ABD62752.1
tumor protein p53 [Homo sapiens]	Homo sapiens	87.8	87.8	100%	4e-24	97.56%	89	ABO28427.1

Задание 4

Построим дерево ортологов для TP53(Homo sapiens) из белковой последовательности с помощью программы MEGA. Мы используем именно белковую последовательность, потому что организмы не имеют близкой родословной связи. Воспользуемся алгоритмом Neighbor-Joining, так как дерево получится наиболее точным. Данный ген встречается далеко не у всех организмов(в основном у млекопитающих), а это значит, что он относительно новый.



Вывод

Ген TP53 служит геном-супрессором опухолевого роста. Его мутации являются ранними генетическими изменениями в развитии рака кожи, вызванного УФ-излучением. Преобладает у животных. Я использовала базу данных NCBI, алгоритм BLAST и программу MEGA для построения филогенетических деревьев.

олимпиада иннагрика

2024 Финал

профиль АгроТех

Команда Ц9-013

Наследники Капицы

9-й класс Кейс №Ц9-1

Разработка алгоритмов

для сравнения характеристик
геномных последовательностей

Наследники Капицы

Василиса Шумилина

Оратор, исследователь

Арина Хлюпина

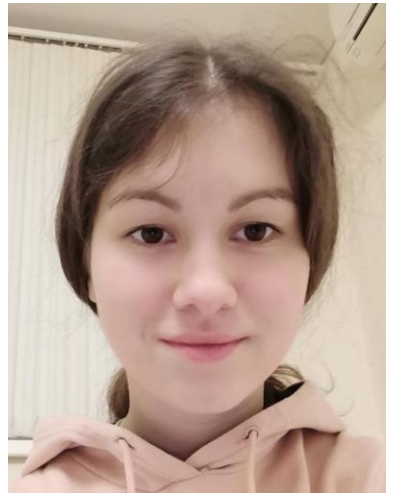
Организатор проектной деятельности

Александра Гамаюнова

Исполнитель

Дарья Панина

Исполнитель



Алиса Прилепина

Генератор идей

Владислав Андреевич Долгих

наш наставник

Проблематика

С точки зрения биологии

Что такое бактериоцины и с чем их едят?

Бактериоцины — это специфичные белки, вырабатываемые определенными бактериями, которые повреждают мембраны других видов бактерий и подавляют их жизнедеятельность.

Могут использоваться в пищевом, сельскохозяйственном и фармацевтическом производстве.

Проблематика

С точки зрения информатики

Как алгоритмы биоинформатики могут помочь в решении проблемы?

Биоинформатика правит миром!!!

Для работы с геномными последовательностями нужно разработать специальные программы.

Данные последовательности очень объёмные, поэтому программы должны работать быстро и эффективно.

Задача: Найти оптимальный вариант кода, выполняющий задания кейса наиболее эффективно и быстро.

Задание 1

Случайные последовательности, их анализ

- Генерируем последовательности в четырёхбуквенном алфавите (A, T, G, C) при помощи библиотеки random
- Разбиваем каждую последовательность на триплеты
- Находим среднее число старт-кодонов ATG

Среднее значение количества старт-кодонов, приходящихся на одну случайную последовательность — 0.786

0.06070494

Задание 1

Случайные последовательности, их анализ. Пример решения задания

- Рассмотрим пример решения, сгенерировав 3 нуклеотидных последовательностей ДНК

>Seq1
TGCAGATCTTCAAAATTTCATCTCTCCTCGATCGTGGTCCTCTAGTACCTTGCAGACA
CCTTGCATAATACCCTTAGGAATGCAACTACACAAGTGCTTACAGTCTGAAGTGTGC
ACTTACGCATCGCCCCGGTGTCCCACTTGGCTTCA

>Seq2
CCGAGTGCCTTCCCAGGGGTCGGTTGCCATTACACTCAGGCTTTCTAGGATCGAGG
AAAGGGTTGTTCGCAACGGGTAATTTTTTCT**ATG**CATTATCGCGACCTGTGTCCGGG
CTGCGGTTAGACATTGCGGACGCTG**ATG**TTACCGTCA

>Seq3
CCCAGCGCGCCGAGAAAGGACAATCGGAGCATAGTGCCACCTCTGCTGTATCTTTT
TAAGCCCTATCGT**ATG**ATCTCCCCTCTTGAAAGTTGCGTATAACCGGCCCTCTCGGG
TTACTACTGAGAGGCCTGGGAGGCTAACAAGTTAGCT

Среднее значение количества старт-кодонов, приходящихся на одну случайную последовательность: 1

```
triple_list=[]
for i in range(len(s1 + s2 + s3)//3):
    a = s3[i*3:i*3+3]
    triple_list.append(a)

def search(triple_list, s='ATG'):
    count  =  0
    for triple    in   triple_list:
        if(triple == 'ATG'):
            count+=1
    return count

ww = search(triple_list, s='ATG')
print(ww)
```

Задание 2

Анализ состава последовательностей бактерицинов

- Проверяем входные данные на соответствие формату FASTA (название последовательности начинается со знака >), чтобы она корректно работала

2.1 Выводим длину каждой строки, которая находится под определенным номером

2.2
- Проверяем строки на наличие отрицательно заряженных аминокислот в бактериоцинах (аспарагиновой N и глутаминовой E)
- Находим процент содержания данных аминокислот (E+N)

0.1364510

```
122
81
162
98
126
```

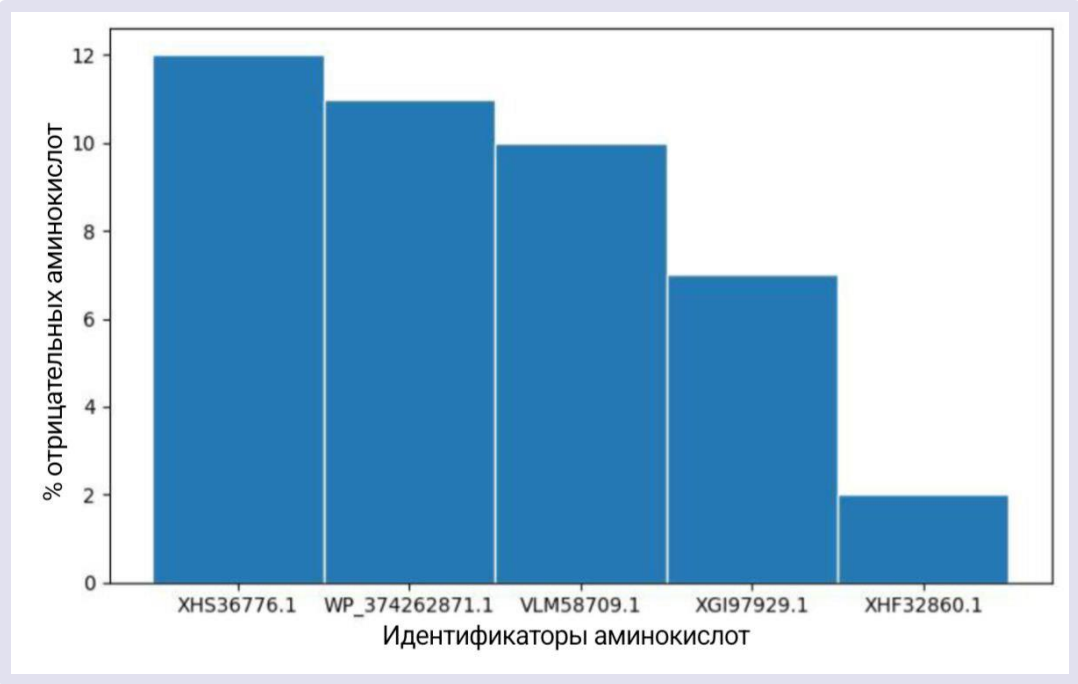
```
XHF32860. 1 bacteriocin [Pseudomonas chlororapis] : 3
XHS36776.1 bacteriocin [Pedobacter sp. UC225_65] : 10
WP_374262871.1 bacteriocin [Campylobacter sp. MRC_CM2] : 18
VLM58709.1 bacteriocin [Streptococcus pneumoniae] : 10
XGI979929.1 bacteriocin [Delftia lacustris] : 10
```

Задание 2

Анализ состава последовательностей бактериоцинов

-Сортируем словарь в обратном порядке основываясь на второй элемент из пары

-Подготавливаем и строим вертикальную гистограмму



Задание 3

Визуализация таксономического состава бактериоцинов

При помощи инструмента BLAST нашли список ближайших гомологов белка WP_374262871.1.

Находим 3 наиболее часто встречающихся таксона и делаем по ним диаграмму (библиотека matplotlib.pyplot.pie)

Находим три самые часто встречающиеся аминокислоты в данной выборке белков

0.12930989

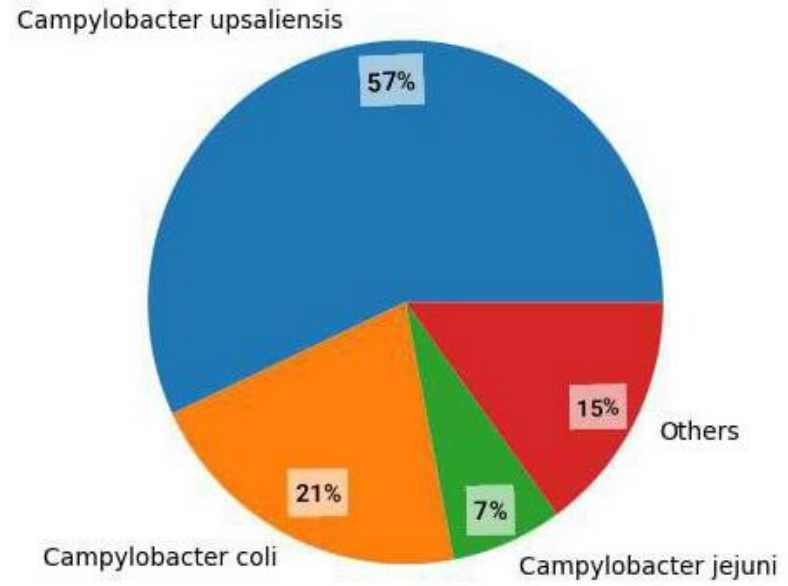
Задание 3

Визуализация таксономического состава бактериоцинов

Ответы:

3.1 Три наиболее часто встречающихся таксона:
[*Campylobacter upsaliensis*] 57%
[*Campylobacter coli*] 21%
[*Campylobacter jejuni*] 7%
3.3 Доли трех наиболее часто встречаемых аминокислот в данной выборке белков:
G 10% (глицин)
N 9% (аспаргин)
V 9% (валин)

3.2 Доли последовательностей трех наиболее представленных таксонов:



Задание 4

Анализ консенсус-последовательности бактериоцинов. Biopython

Скачиваем 100 выравненных при помощи BLAST последовательностей и читаем их с помощью функции AlignIO.read. Сохраняем 20 последовательностей в FASTA-файл.

С помощью функции AlignInfo сравниваем все буквы по вертикали и создаем консенсус-последовательность с порогом 60%.

Консенсус-последовательность:

MKQVKGGYLVGFINIGNNEIG
ALAITTINAHDAGV**X**FG**X**RDG
SEIKYDKNGNYVPSKSGLCGIG
VA**X**CSAPSQN**X**LQEYMQVAG
DPVLNALGYTVKRQVK**X**S**X**LG
QPYVLFTYGVIVYNYSS**X**TAYRI
NS**X**AVLNNNIIIKELAGKYKEQ
MESALGGWYPRR

X - неопределённая аминокислота

0.20532393

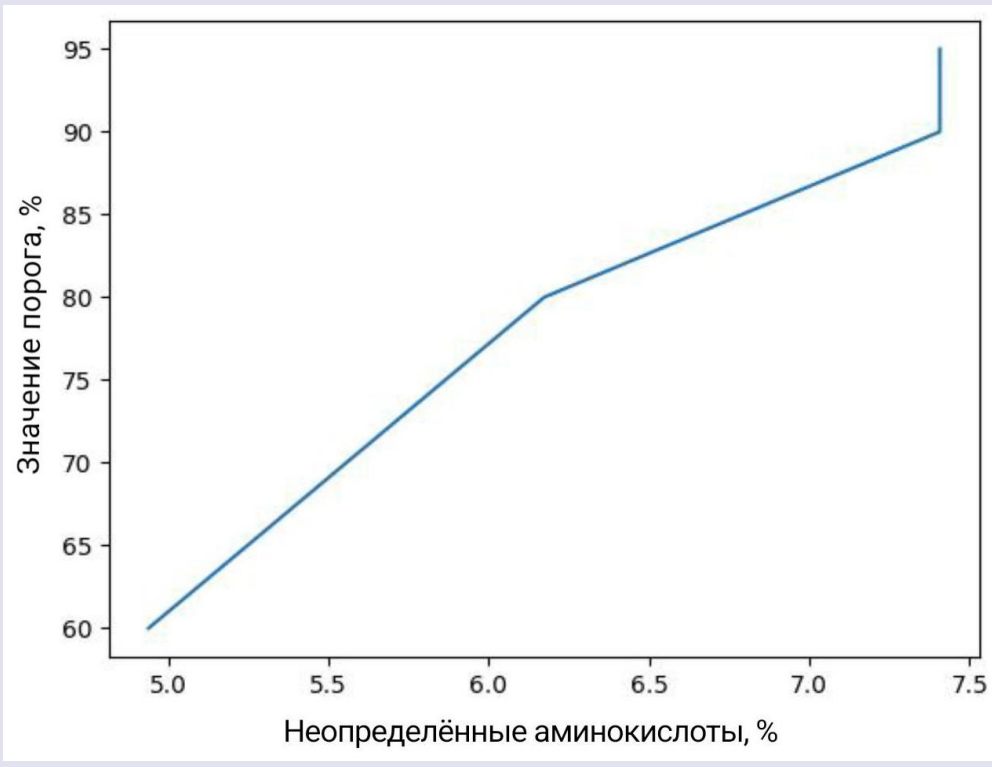
Консенсус-последовательность — искусственная последовательность, содержащая в каждой позиции элемент, наиболее часто встречающийся у нескольких гомологичных последовательностей

Задание 4

Анализ консенсус-последовательности бактериоцинов. Biopython

С помощью функции summary.dumb_consensus находим количество букв X (неопределённых аминокислот) консенсус-последовательности.

Вводим список и вставляем в него новое пороговое значение. Находим количество X в консенсус-последовательностях с различными порогами (70; 80; 90; 95%)



Рисуем полученный график зависимостей на экране. С помощью функции plot

Задание 5

Трансляция. Случайные мутации в гене

Для перевода нуклеотидной последовательности в аминокислотную создадим словарь, где отметим стоп-кодоны “*” и разобьём её на триплеты.

ATG AAT ACA AAA GCA TTT GAA CAA TTT GGT GTC ATG AAC GAA GTA GAA CTT TCA GCG G
TTG AAG GTG GGG TAT AGT GCA ATA GAT TGC CGA AAT GCA ATG CTG ACC GGT ATT GCA AG
T GGA ATA ATT GCT GGA GGC ACG GGA GCA GGA CTT GCA ACA TTG GGT GTT GGA GGA TT
A GCA GGT GCC TTT GTT GGT GCG CAC ATT GGA GCT ATT GGT GGT GGT TTG ACA TGT GTT
GGC GGA ATG ATT GGA AAT AAG TTT **TGA**

MNTKAFEQFGVMNEVELSAVEGGYSAIDCRNAMLTGIASGIIAGGTGAGLATLGVGGLA
GAFVGAHIGAIGGGLTCVGGMIGNKF

В исходную последовательность вносим 10 мутаций без повторения их координат (кроме старт- и стоп-кодонов).
Отметим, что при мутациях может образоваться стоп-кодон, из-за которого последовательность оборвётся.
После запуска программы 100 раз мы посчитали, что возникшие стоп-кодоны TAA, TAG, TGA образовались в среднем 29 раз.

0.00699878

Задание 5

Трансляция. Случайные мутации в гене. Пример решения задания

Внесём 10 случайных мутаций в белок-кодирующую последовательность гена бактериоцина Streptococcus infantarius

ATGAATACAAAAGCATTT**TAA**CAATTTGGTGTCATGAACGAAGTAGAACT**C**TCAGCG
GTTGAAGGTGG**C**TATAGTGCAATAGATTGCCGAAATGCAATGCTGACCGG**A**ATTGCA
AGTGGAATAATTGCTGGAGGCACGGGAGCAGGACTTGCA**T**CATTGGGTGTTGGAGG
ATTAGCA**T**GTGCCTTTGTTGGTGCGCACATT**A**GAGC**A**ATTGGTGGTGGTTTGACATGT
GTTG**A**CG GAATGATTGG**C**AATAAGTTTTGA

В результате образуется стоп-кодон **TAA**, и последовательность обрывается.

ATGAATACAAAAGCATTT**TAA**

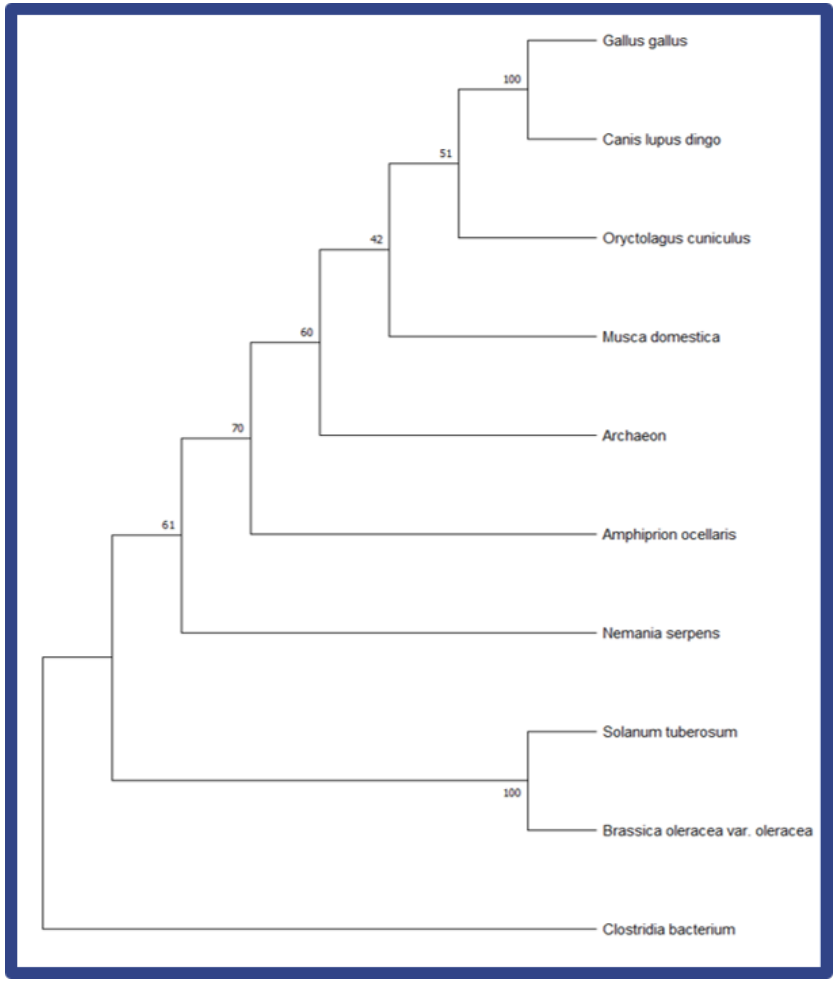
Переводим в аминокислотную последовательность только оставшуюся часть.

MNTKAF

Решение задач практического тура

В рамках практического тура мы:

- **Произвели поиск различных последовательностей в NCBI Genbank**
- **Осуществили поиск гомологов в NCBI BLAST**
- **Выполнили построение филогенетических деревьев в программе MEGA**



Дерево ортологов для гена PAX6

Выводы

К чему нас привели решения данного кейса?

- Написали пакет программ, используя базовые биоинформатические алгоритмы.
- При написании программ выбрали оптимальный по времени и быстродействию вариант их реализации.
- Применили наши программы на реальных последовательностях бактериоцинов, изучив их свойства.

Приложения

Решать данный кейс нам помогали:

Источники информации:

— Python online
— База данных NCBI
— Статья. Аминокислоты в составе ДНК
— Статья. Bacteriocin: A natural approach for food safety and food security
— Описание формата fasta для последовательностей белков и нуклеиновых кислот
— Python 3 для начинающихся
— Функция AlignIO.read из пакета Biopython
— Статья про модуль AlignInfo
— Статья про BioPython
— Алгоритм BLAST

Google-документ с ссылками, представленными на слайде

Ждём Ваши вопросы!)