

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таскаев Сергей Валерьевич
Должность: Ректор
Дата подписания: 17.09.2025 10:57:28
Уникальный программный ключ:
04c19ed8bfb98f3b6cb77a486b9a8788b8522525



МИНОБРАЗОВАНИЯ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств по дисциплине «Введение в цифровые биологические исследования» по
направлению подготовки 06.03.01 «Биология» ФГБОУ ВО «ЧелГУ»

стр. 1

Фонд оценочных средств

по дисциплине

Введение в цифровые биологические исследования

Направление подготовки (специальность)

06.03.01 Биология

Направленность (профили)

Микробиология, биоэкология, биофизика, генетика, гистология и
гистологическая техника

Присваиваемая квалификация

Бакалавр

Форма обучения

Очная

Челябинск, 2025



1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Направление подготовки: 06.03.01 «Биология»

Направленности: «Биофизика», «Биоэкология», «Генетика», «Гистология и гистологическая техника», «Микробиология»

Дисциплина: «Введение в цифровые биологические исследования»

Семестр изучения: 7

Форма промежуточной аттестации: зачёт

2. ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ

2.1. Компетенции, закреплённые за дисциплиной

Изучение дисциплины «Введение в цифровые биологические исследования»
направлено на формирование следующих компетенций:

Коды компетенции (по ФГОС)	Содержание компетенций согласно ФГОС	Перечень планируемых результатов обучения по дисциплине
1	2	3
ОПК-7	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	
	ОПК-7.1 Имеет представление об основных существующих информационных технологиях, используемых при решении профессиональных задач;	Обучающийся знает основные биоинформатические базы данных (NCBI, EMBL, UniProt) и способы загрузки этих данных на персональный компьютер, их обработки и представления результатов анализа.



	ОПК-7.2 Демонстрирует умения использовать существующие информационные технологии при решении задач профессиональной деятельности.	Студент умеет писать программы на языках Python и Bash для операций с данными по последовательностям нуклеиновых кислот и белков, способен использовать сетевые ресурсы для обработки больших объемов данных.
	ОПК-7.3 Имеет практический опыт использования существующих информационных технологий при решении задач профессиональной деятельности.	Обучающийся владеет навыками решения практических задач по поиску заданных структур в нуклеотидных последовательностях, оценке различий в экспрессионных профилях, анализу структуры белковых молекул и их лигандов.
ОПК-8	Способен использовать методы сбора, обработки, систематизации и представления полевой и лабораторной информации, применять навыки работы с современным оборудованием, анализировать полученные результаты	
	ОПК-8.1 использует основные типы экспедиционного и лабораторного оборудования, особенности выбранного объекта профессиональной деятельности, условия его содержания и работы с ним с учетом требований биоэтики;	Обучающийся знает особенности основных типов оборудования, используемого для получения геномных и протеомных данных в полевых и лабораторных условиях, понимает пределы применимости данного оборудования, его преимущества и недостатки, степень точности получаемых с его помощью данных.
	ОПК-8.2 анализирует и критически оценивает развитие научных идей, на основе имеющихся ресурсов, составляет план	Обучающийся умеет искать пути решения новых задач, используя руководства к используемому программному обеспечению, а также осуществляя поиск существующих программных решений на сетевых ресурсах



	решения поставленной задачи, выбирает и модифицирует методические приемы;	(GitHub).
	ОПК-8.3 применяет навыки использования современного оборудования в полевых и лабораторных условиях, грамотно обосновывает поставленные задачи в контексте современного состояния проблемы, использует математические методы оценивания гипотез, обработки экспериментальных данных и адекватно оценивает достоверность и значимость полученных результатов, представляет их в широкой аудитории и вести дискуссию.	Студент владеет методами корректной статистической обработки больших объемов биологических экспериментальных данных, способен учитывать при анализе особенности таких данных (редукция многомерных данных, множественная проверка гипотез), владеет методами визуализации данных, оформления отчетов для наглядного представления результатов анализа широкой аудитории.

3. СОДЕРЖАНИЕ ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

3.1 Виды оценочных средств

№ п/п	Код компетенции/планируемые результаты обучения		Контролируемые разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства для промежуточной аттестации /№ задания
1.	ОПК-7	ОПК-7.1 Имеет представление об основных существующих информационных технологиях,	1-8	Ситуационные задачи	Зачётный проект



		используемых при решении профессиональн ых задач;			
		ОПК-7.2 Демонстрирует умения использовать существующие информационные технологии при решении задач профессионально й деятельности.	1-8	Ситуационные задачи	Зачётный проект
		ОПК-7.3 Имеет практический опыт использования существующих информационных технологий при решении задач профессионально й деятельности.	1-8	Ситуационные задачи	Зачётный проект
2.	ОП К-8	ОПК-8.1 использует основные типы экспедиционного и лабораторного оборудования, особенности выбранного объекта профессионально й деятельности, условия его содержания и работы с ним с учетом требований биоэтики;	1-8	Ситуационные задачи	Зачётный проект
		ОПК-8.2	1-8	Ситуационные задачи	Зачётный проект



		анализирует и критически оценивает развитие научных идей, на основе имеющихся ресурсов, составляет план решения поставленной задачи, выбирает и модифицирует методические приемы;			
		ОПК-8.3 применяет навыки использования современного оборудования в полевых и лабораторных условиях, грамотно обосновывает поставленные задачи в контексте современного состояния проблемы, использует математические методы оценивания гипотез, обработки экспериментальных данных и адекватно оценивает достоверность и значимость полученных результатов,	1-8	Ситуационные задачи	Зачётный проект



		представляет их в широкой аудитории и вести дискуссию.			
--	--	--	--	--	--

Примечание: типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины. Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

3.2 Содержание оценочных средств

Оценочные средства представлены ситуационными задачами и итоговым зачётным проектом.

3.2.1 Ситуационные задачи

1. Сравните нуклеотидный состав двух последовательностей, лежащих в fasta-файлах.

Решение:

Для последовательностей, представленных строками seq1 и seq2 в Python нужно выполнить следующие команды:

```
from collections import Counter  
Counter(seq1); Counter(seq2)
```

2. Произведите расчёт GC-состава предложенных нуклеотидных последовательностей.

Решение:

Для интересующей последовательности, представленной строкой seq в Python необходимо выполнить следующие команды:

```
dt = {"A": 0, "G": 0, "C": 0, "T": 0}  
for s in seq:  
    cnt = dt[s]  
    dt[s] = cnt + 1  
print((dt['G'] + dt['C']) / (dt['G'] + dt['C'] + dt['T'] + dt['A']))
```



3. Дан fasta-файл с геномной последовательностью. Найдите в ней предложенный короткий нуклеотидный участок. Сколько раз встречается в геноме этот короткий участок?

Решение:

Допустим, что необходимо найти короткий нуклеотидный участок AGGGCT в FASTA-файле fasta.fa. Тогда соответствующий код будет выглядеть следующим образом:

```
pip install parse
from parse import *
myfasta = open(«fasta.fa», mode=«r»)
list(findall(«AGGGCT», myfasta))
```

4. В предложенном bed-файле найдите, сколько раз встречается каждая хромосома.

Решение:

Если поиск осуществляется по файлу bed_file, то для хромосомы chr1:

```
pip install parse
from parse import *
mybed = open(«bed_file», mode=«r»)
len(list(findall(«chr1\t», mybed)))
```

5. Подсчитайте, сколько в указанном bed-файле белок-кодирующих и некодирующих последовательностей.

Решение:

Если поиск осуществляется по файлу bed_file:

```
pip install parse
from parse import *
mybed = open(«bed_file», mode=«r»)
len(list(findall(«+», mybed)))
len(list(findall(«-», mybed)))
```



6. Дан геном некоторого организма. Требуется определить частоту встречаемости разных кодонов в геноме. Какой кодон самый редкий? Какой – самый частый? Какие аминокислоты они кодируют.

Решение:

Допустим, что последовательность генома некоторого организма представлена в объекте `genome`, тогда:

```
CODON_TABLE = {'AAA': 'K', 'AAC': 'N', 'AAG': 'K', 'AAU': 'N', 'ACA': 'T', 'ACC': 'T',  
'ACG': 'T', 'ACU': 'T', 'AGA': 'R', 'AGC': 'S', 'AGG': 'R', 'AGU': 'S', 'AUA': 'I', 'AUC': 'I', 'AUG':  
'M', 'AUU': 'I', 'CAA': 'Q', 'CAC': 'H', 'CAG': 'Q', 'CAU': 'H', 'CCA': 'P', 'CCC': 'P', 'CCG': 'P',  
'CCU': 'P', 'CGA': 'R', 'CGC': 'R', 'CGG': 'R', 'CGU': 'R', 'CUA': 'L', 'CUC': 'L', 'CUG': 'L',  
'CUU': 'L', 'GAA': 'E', 'GAC': 'D', 'GAG': 'E', 'GAU': 'D', 'GCA': 'A', 'GCC': 'A', 'GCG': 'A',  
'GCU': 'A', 'GGA': 'G', 'GGC': 'G', 'GGG': 'G', 'GGU': 'G', 'GUA': 'V', 'GUC': 'V', 'GUG': 'V',  
'GUU': 'V', 'UAA': '*', 'UAC': 'Y', 'UAG': '*', 'UAU': 'Y', 'UCA': 'S', 'UCC': 'S', 'UCG': 'S',  
'UCU': 'S', 'UGA': '*', 'UGC': 'C', 'UGG': 'W', 'UGU': 'C', 'UUA': 'L', 'UUC': 'F', 'UUG': 'L',  
'UUU': 'F'}  
  
start_codon = 'AUG'  
  
stop_codons = ['UAA', 'UAG', 'UGA']  
  
coding_codons = list(set(CODON_TABLE) - set(stop_codons))  
  
codons = (genome[i:i+3] for i in range(0, len(genome), 3))  
  
aminoacids = (CODON_TABLE[c] for c in codons)  
  
cnt = Counter(aminoacids)  
  
print (cnt.most_common())
```

7. Дана нуклеотидная последовательности. Проведите её трансляцию, т. е. переведите последовательность нуклеотидов в последовательность аминокислот.

Решение:

Если файл с нуклеотидной последовательностью называется `example.fasta`, а таблица кодонов задана аналогично предыдущей задаче, то код для решения следующий:



```
def transcribe(n, tr_table={"A": "A", "T": "U", "G": "G", "C": "C"}):  
    return tr_table[n]  
  
def get_triplets_reader(receiver, ignore_first=0):  
    nucleotides = []  
    for i in range(ignore_first):  
        n = (yield)  
        #print ("Triplet reader", ignore_first, n)  
    while True:  
        n = (yield)  
        #print ("Triplet reader", ignore_first, n)  
        nucleotides.append(n)  
        if len(nucleotides) == 3:  
            #print ("sending")  
            triplet = "".join(nucleotides)  
            nucleotides = []  
            receiver.send(triplet)  
  
def get_proteins_reader(protein_receiver, codons_receiver, name):  
    triplets = None  
    aminoacids = None  
    while True:  
        t = (yield)  
        ac = CODON_TABLE[t]  
        #print ("Protein reader", name, t, ac)  
        if ac == "M" and triplets is None:  
            triplets = [t]  
            aminoacids = [ac]  
        elif ac == "*" and triplets is not None:  
            triplets.append(t)  
            protein_receiver.send("".join(aminoacids))
```



```
codons_receiver.send(triplets)

triplets = None
aminoacids = None

elif triplets is not None:
    triplets.append(t)
    aminoacids.append(ac)

def get_triplets_counter(cnt):
    while True:
        codons = (yield)
        cnt.update(codons)

def get_proteins_writer(lst):
    while True:
        protein = (yield)
        lst.append(protein)

def iter_seqs(in_file_path):
    with open(in_file_path) as in_file:
        in_file.readline()
        letters = chain.from_iterable((line for line in in_file))
        dna_nucleotides = filter(lambda x : x != "\n", letters)
        nucleotides = map(lambda x : transcribe(x.upper()), dna_nucleotides)
        codons_cnt = Counter()
        triplets_counter = get_triplets_counter(codons_cnt)
        proteins_lst = []
        proteins_writer = get_proteins_writer(proteins_lst)
        orfs_readers = [get_triplets_reader(get_proteins_reader(proteins_writer,
        triplets_counter, start_pos),
        ignore_first=start_pos)\
        for start_pos in range(3)]

    for n in nucleotides:
```



```
for reader in orfs_readers:  
    reader.send(n)  
return proteins_lst
```

8. Дано два файла: один содержит уровни экспрессии различных генов, полученные в ходе нескольких отдельных экспериментов, а другой файл содержит метки этих экспериментов. Проведите объединение данных и определите средние уровни экспрессии отдельно по каждому эксперименту.

Решение:

Если файлы лежат в директории /data, файл с уровнями экспрессии называется expression.csv, а файл с метками экспериментов называется metadata.csv, то для решения нужно выполнить следующие команды:

```
import pandas as pd  
express_df = pd.read_csv("/data/expression.csv", index_col=0)  
meta_df = pd.read_csv("/data/metadata.csv", index_col=0)  
express_df["Experiment"]=meta_df["dataset"]  
express_df.groupby("Experiment").mean
```

9. Даны результаты по экспрессии генов из разных экспериментов. Определите, есть ли в данных батч-эффект и в случае его наличия устраните его.

Решение:

Если файлы лежат в директории /data, файл с уровнями экспрессии называется expression.csv, а файл с метками экспериментов называется metadata.csv, то для решения нужно выполнить следующие команды:

```
import pandas as pd  
express_df = pd.read_csv("/data/expression.csv", index_col=0)  
meta_df = pd.read_csv("/data/metadata.csv", index_col=0)  
import numpy as np  
import matplotlib.pyplot as plt
```



```
import seaborn as sns
expresst_df = express_df.transpose()
draw_pca_chart(expresst_df, meta_df["dataset"], normalize=False)
pip install combat
from combat.pycombat import pycombat
expresscor_df = pycombat(express_df, meta_df["dataset"])
draw_pca_chart(expresscor_df.transpose(), meta_df["dataset"], normalize=False)
```

10. В указанной аминокислотной последовательности подсчитайте соотношение полярных и неполярных аминокислот для разных участков последовательности.

Решение:

Для интересующей последовательности, представленной строкой aminoseq, в Python необходимо выполнить следующие команды:

```
dt = {"A": 0, "R": 0, "D": 0, "N": 0, "V": 0, "H": 0, "G": 0, "E": 0, "Q": 0, "L": 0, "I": 0, "K": 0,
      "M": 0, "P": 0, "S": 0, "Y": 0, "T": 0, "W": 0, "F": 0, "C": 0}
for a in aminoseq:
    cnt = dt[a]
    dt[a] = cnt + 1
print(dt['R'] + dt['D'] + dt['N'] + dt['H'] + dt['E'] + dt['Q'] + dt['K'] + dt['S'] + dt['Y'] + dt['T'])
print(dt['A'] + dt['V'] + dt['G'] + dt['L'] + dt['I'] + dt['M'] + dt['P'] + dt['W'] + dt['F'] + dt['C'])
```

11. Дан файл из базы данных аминокислотных последовательностей и файл с последовательностью определённого белка с неизвестной функцией. Проведите поиск белка в базе данных и попробуйте предсказать функцию белка.

Решение:

Для файла с последовательностью белка prot необходимо выполнить следующие команды:

```
from Bio.ExPASy import ScanProsite
search = ScanProsite.scan(seq=protein)
```



```
result = ScanProsite.read(search)  
type(result)
```

3.2.2 Задания зачётного проекта

1. Визуализация результатов транскриптомного исследования с корректировкой на батч-эффект.
2. Анализ используемости кодонов в предложенных геномных последовательностях и сравнение геномов.
3. Создание трёхмерной модели белка по линейной аминокислотной последовательности и оценка термодинамических параметров.

Для выполнения проекта каждый студент должен представить кроме фактического решения также программный код на языке Python.

4. ПОРЯДОК ПРОВЕДЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1 Порядок проведения промежуточной аттестации

Промежуточная аттестация по дисциплине проводится по системе зачёт/незачёт в ходе выполнения итогового зачётного проекта. Проект посвящён одному из трёх заданий на выбор (см. раздел 3.2.2). Каждый студент выбирает один из трёх предложенных проектов индивидуально.

При необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на зачёте.

При проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями. Эти средства могут быть предоставлены ЧелГУ или могут использоваться собственные



технические средства.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

При проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине обеспечивается выполнение следующих дополнительных требований в зависимости от индивидуальных особенностей обучающихся:

а) инструкция по порядку проведения процедуры оценивания предоставляется в доступной форме (устно, в письменной форме, устно с использованием услуг сурдопереводчика);

б) доступная форма предоставления заданий оценочных средств (в печатной форме, в печатной форме увеличенным шрифтом, в форме электронного документа, задания зачитываются ассистентом, задания предоставляются с использованием сурдоперевода);

в) доступная форма предоставления ответов на задания (письменно на бумаге,



набор ответов на компьютере, с использованием услуг ассистента, устно).

При необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Проведение процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья допускается с использованием дистанционных образовательных технологий.

4.2 Критерии оценивания промежуточной аттестации по видам оценочных средств

4.2.1 Критерии оценивания решения ситуационных задач и итогового проекта

Критерии	Балл
Получен верный результат, студент верно понимает и может объяснить ход решения	5
Полученный результат отличается от верного из-за ошибки вычислительного характера, однако принцип решения студент понимает верно	4
Полученный результат отличается от верного из-за методической ошибки, принцип решения студент понимает не полностью	3
Верный результат не получен, студент не может объяснить принцип решения	2

4.3 Результаты промежуточной аттестации и уровни сформированности компетенций

4.3.1 Результаты промежуточной аттестации

«Зачтено» – студент глубоко и полно владеет содержанием учебного материала; умеет связывать теорию с практикой, верно решает задачи, теоретические выводы подтверждает примерами, фактами, данными научных исследований; осуществляет межпредметные связи. Выводы студента логичны и полны. Студент ясно и кратко излагает ответы на поставленные вопросы; умеет обосновывать свои суждения и профессионально-личностную позицию по рассматриваемым вопросам. Допустимо, что

 МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ») Факультет Биологический Кафедра Микробиологии, иммунологии и общей биологии			
Фонд оценочных средств для промежуточной аттестации по дисциплине «Введение в цифровые биологические исследования» по направлению подготовки 06.03.01 «Биология», направленностям «Биофизика», «Биоэкология», «Генетика», «Гистология и гистологическая техника», «Микробиология» ФГБОУ ВО «ЧелГУ»			
Версия документа - 1	стр. 17 из 18	Первый экземпляр _____	КОПИЯ № _____

студент обнаруживает знание и понимание основных положений учебного материала, но излагает его неполно, допускает неточности и ошибки.

«Не зачтено» – студент имеет разрозненные, бессистемные знания: не умеет выделять главное и второстепенное; допускает ошибки в определении понятий, формулировке теоретических положений, искажает их смысл; не умеет соединять теоретические положения с практикой.

4.3.2 Уровни сформированности компетенций:

1. Пороговый уровень: предполагает формирование компетенций на начальном уровне – знание основных техник выделения и очистки нуклеиновых кислот и белков из биологических образцов; знание базовых принципов и особенностей геномной архитектуры для представителей разных доменов и форм жизни.

2. Базовый уровень: предполагает формирование компетенций на более высоком уровне – знание методов изучения структурных особенностей нуклеиновых кислот и белковых молекул; владение навыками поиска молекулярно-генетической информации в открытых базах данных.

3. Продвинутый уровень: предполагает формирование компетенций на высоком уровне, готовность к самостоятельной профессиональной деятельности – владение методами обработки биоинформатических данных; навык планирования геномного и/или п

р

о

т

е

о

м

н

о

г

о

© ФГБОУ ВО «ЧелГУ»

и

с